A Timed Calculus for Wireless Systems *†

Massimo Merro, Francesco Ballardin, Eleonora Sibilio

Dipartimento di Informatica, Università degli Studi di Verona, Italy

Abstract

We propose a *timed broadcasting process calculus* for wireless systems where *time-consuming communications* are exposed to *collisions*. The operational semantics of our calculus is given in terms of a labelled transition system. The calculus enjoys a number of desirable time properties such as (i) time determinism: the passage of time is deterministic; (ii) patience: devices will wait indefinitely until they can communicate; (iii) maximal progress: data transmissions cannot be delayed, they must occur as soon as a possibility for communication arises. We use our calculus to model and study MAC-layer protocols with a special emphasis on collisions and security. The main behavioural equality of our calculus is a timed variant of barbed congruence, a standard branching-time and contextually-defined program equivalence. As an efficient proof method for timed barbed congruence we define a labelled bisimilarity. We then apply our bisimulation proof-technique to prove a number of algebraic laws.

1 Introduction

Wireless technology spans from user applications such as personal area networks, ambient intelligence and wireless local area networks, to real-time applications, such as cellular, and ad hoc networks. The IEEE 802.11 standard [22] contains a series of specifications for wireless LAN technologies. The basic building block of an 802.11 network is the Basic Service Set (BSS), which is a set of stations that have successfully synchronised and that use radio transceivers to broadcast

*This work was partially supported by the PRIN 2007 project "SOFT".

†An extended abstract appeared in the proceedings of the *3rd International Conference on Fundamentals of Software Engineering* (FSEN'09), volume 5961, pages 228-243, of *Lecture Notes in Computer Science*, Springer, 2010.

messages. In Independent BSS (IBSS), stations communicate with each other without using any distribution system. IBSS networks are sometimes referred to as *ad hoc networks*. In this paper, we propose a formal model for IBSS networks paying particular attention to *communication interferences*. Communication interferences represent one of the main concerns when evaluating the performance of a network in terms of network throughput, i.e. the average rate of successful message delivery over a communication channel.

In concurrent systems, an interference occurs when the activity of a component is damaged or corrupted because of the activities of another component. In Ethernet-like networks communication channels are full-duplex; that is, a node can transmit and receive at the same time. Thus, collisions caused by two simultaneous transmissions are immediately detected and repaired by retransmitting the message after a randomly-chosen period of time. This is not possible in wireless networks where radio signals span over a limited area, called transmission cell, and channels are *half-duplex*: on a given channel, a device can either transmit or receive, but cannot do both at the same time. As a consequence, communication collisions in wireless systems can be only detected at destination.

Many protocols for wireless networks rely on a common notion of *time* among the devices, provided by some clock synchronisation protocol. Most clock synchronisation protocols for ad hoc networks [34, 13, 43, 45, 27, 49] follow the "clock correction" approach correcting the local clock of each node to run in parallel with a global time scale.[1] This approach heavily relies on *network connectivity*. In a connected network all nodes are in touch with each other, although not always directly. Wireless networks are usually assumed to be connected; disconnected devices can be considered as not being part of the network as, in general, they need to re-authenticate to rejoin the network.

In the last twenty-five years, process calculi [30, 8, 31, 10, 21] have been intensively used to study the semantics of concurrent/distributed systems, and to develop verification techniques for such systems. In the literature, there exist a number of process calculi modelling wireless systems [25, 36, 44, 29, 16, 17, 14, 15]. Most of these calculi support message loss to model *communication collisions*. In fact, collisions in wireless systems cannot be avoided, although there are protocols to reduce their occurrences (see, for instance, the IEEE 802.11 CSMA/CA protocol [22] for unicast communications).

In this paper, we propose a *timed broadcasting calculus* for wireless networks, called TCWS, in which all wireless devices are assumed to be *synchronised* (relying on some clock-correction synchronisation protocol). Thus, TCWS is a process calculus with *absolute timing*, where all timing refers to an absolute clock. Time

[1] An excellent survey of existing clock synchronisation protocols for sensor networks (and more generally for ad-hoc networks) can be found in [46].

proceeds in discrete steps represented by occurrences of a simple action σ, in the style of Hennessy and Regan's TPL [20], to denote idling until the next clock cycle. The calculus is value-passing and message transmission is time-consuming. As usual for wireless networks, the communication mechanism is (local) *broadcast*. As in Hennessy and Regan's TPL [20] and Prasad's TCBS [39], our calculus enjoys three basic time properties:

- *time determinism*: the passage of time is deterministic, i.e. a network can reach at most one new state by performing the action σ;
- *patience*: nodes will wait indefinitely until they can communicate;
- *maximal progress*: data transmissions cannot be delayed, they must occur as soon as a possibility for communication arises.

The operational semantics of our calculus is given in terms of a *labelled transition system* (LTS) in the SOS style of Plotkin.

We provide a notion of network well-formedness to take into account node-uniqueness, network connectivity, transmission exposure, and transmission consistency. Then, we prove that our labelled transition semantics preserves network well-formedness.

We use our calculus to model and study MAC-layer protocols, such as CSMA and CSMA/CA [22], and a wireless network security protocol, called MiniSec [28].

A central concern in process calculi is to establish when two terms have the same observable behaviour, that is, they are indistinguishable in any context. *Behavioural equivalences* are fundamental for justifying program transformations. Our program equivalence is a timed variant of (weak) reduction barbed congruence, a branching-time contextually-defined program equivalence. Barbed equivalences [32] are intuitive but difficult to use due to the quantification on all contexts. Simpler proof techniques are based on *labelled bisimilarities* [30], which are co-inductive relations that characterise the behaviour of processes using a labelled transition system. We define a labelled bisimilarity which is a proof method for timed reduction barbed congruence. We then apply our bisimulation proof-technique to prove a number of algebraic laws.

We end this introduction with an outline of the paper. In Section 2, we provide both syntax and operational semantics of our calculus. In the same section we propose a notion of network well-formedness to rule out inconsistent networks. In Section 3, we prove that TCWS enjoys time determinism, maximal progress and patience. In Section 4, we use an extended version of our calculus to specify and study a number of protocols. In Section 5, we equip TCWS with a notion of observational equivalence along the lines of Milner and Sangiorgi's barbed congruence. In Section 6, we propose a labelled bisimilarity as a proof method

Table 1 The Syntax

Networks:			
M, N	$::=$	$\mathbf{0}$	empty network
	$\mid$	$M \mid N$	parallel composition
	$\mid$	$n[W]_t^\nu$	node
Processes:			
W	$::=$	P	inactive process
	$\mid$	A	active process
P, Q	$::=$	nil	termination
	$\mid$	$!\langle u\rangle.P$	broadcast
	$\mid$	$\lfloor ?(x).P \rfloor Q$	receiver with timeout
	$\mid$	$\lfloor \tau.P \rfloor Q$	internal with timeout
	$\mid$	$\sigma.P$	delay
	$\mid$	$[u_1 = u_2]P, Q$	matching
	$\mid$	$H\langle \tilde{u}\rangle$	recursion
A	$::=$	$\langle v\rangle^t.P$	active sender
	$\mid$	$(x)_v.P$	active receiver

for our observations equivalence. More precisely, we prove that our bisimilarity is a congruence and that it implies our observational equivalence. We then use our bisimilarity to prove a number of algebraic laws. Finally, in Section 7 we present, in some detail, future and related works.

2 The Calculus

In Table 1, we define the syntax of TCWS in a two-level structure, a lower one for processes and an upper one for networks. We use letters $a, b, c, \ldots$ for logical names, x, y, z for variables, u for values, and v and w for closed values, i.e. values that do not contain variables. Closed values actually denote messages that are transmitted as TCP/IP packets. We write $\tilde{u}$ to denote a tuple $u_1, \ldots, u_k$ of values.

Networks are collections of nodes (which represent devices) running in parallel and using a unique common channel to communicate with each other. We use the symbol $\mathbf{0}$ to denote the empty network, while $M_1 \mid M_2$ represents the parallel composition of two sub-networks M_1 and M_2. The communication paradigm is *local broadcast*; only nodes located in the range of the transmitter may receive data. We write $n[W]_t^\nu$ for a node named n (the device network address) executing

the sequential process W. The variable t is a semantic tag ranging over positive integers to represent *node exposure*. Thus, a node $n[W]^{\nu}_{t}$, with $t > 0$, is exposed to a transmission (or more transmissions) for the next t instants of time. The tag ν denotes the set of (the names of) the neighbours of n. Said in other words, ν contains all nodes in the transmission cell of n, except for n itself ($n \notin \nu$).[2] Our wireless networks have a fixed topology where nodes cannot be created or destroyed. Furthermore all nodes have the same transmission range.[3]

Processes are sequential and live within the nodes. For convenience, we distinguish between non-active and active processes. An active process is a process which is currently transmitting or receiving. An *active node* is a node with an active process inside. The symbol nil denotes the skip process. The sender process $!\langle v\rangle.P$ allows to broadcast the value v. Once the transmission starts the process evolves into the active sender process $\langle v\rangle^{\delta_v}.P$ which transmits the message v for the next δ_v time units, the time necessary to transmit v. The process $\lfloor ?(x).P \rfloor Q$ denotes a receiver with timeout. Intuitively, this process either starts receiving a value w in the current instant of time, evolving into an active receiver $(x)_w.P$, or it idles for one time unit, and then continues as Q. Notice that only when the reception terminates and the channel becomes free the active receiver does the *Cyclic Redundancy Check* (CRC) to verify the integrity of the received packets. Upon successful reception the variable x of P is instantiated with the transmitted message w. The process $\lfloor \tau.P \rfloor Q$ either performs an internal action, in the current time interval, and then continues as P, or it idles for one time unit, and then continues as Q. The process $\sigma.P$ models sleeping for one time unit. Process $[v_1 = v_2]P, Q$ is the standard "if then else" construct: it behaves as P if $v_1 = v_2$, and as Q otherwise. In processes $\sigma.P$, $\lfloor \tau.P \rfloor Q$, $\lfloor ?(x).P \rfloor Q$, and $!\langle v\rangle.P$ the occurrence of processes P and Q are said to be guarded. We write $H\langle \tilde{v}\rangle$ to denote a process defined by means of an equation of the form $H(\tilde{x}) = P$, with $\mid \tilde{x} \mid = \mid \tilde{v} \mid$, where $\tilde{x}$ contains all variables that appear free in P. Defining equations provide *guarded recursion*, since P may only contain guarded occurrences of process identifiers, such as H itself.

Remark 2.1 *The recursion construct allows us to define a persistent listener, i.e. a receiver which waits indefinitely for an incoming message. With an abuse of notation, we will write* $?(x).P$ *to indicate such listener process, defined via the following recursive equation* $\mathrm{Rcv} = \lfloor ?(x).P \rfloor \mathrm{Rcv}$. *Similarly, we will write* $\tau.P$ *as an abbreviation for the process defined as* $\mathrm{Tau} = \lfloor \tau.P \rfloor \mathrm{Tau}$.

In the terms $\lfloor ?(x).P \rfloor Q$ and $(x)_v.P$ the variable x is bound in P. This gives rise

[2] We could have represented the topology in terms of a restriction operator à la CCS over node names; we preferred our notation to keep at hand the neighbours of a node.

[3] These assumptions are discussed in the last section of the paper.

Table 2 Structural Congruence

$n[[v = v]P, Q]^\nu_t \equiv n[P]^\nu_t$	(Struct Then)
$n[[v_1 = v_2]P, Q]^\nu_t \equiv n[Q]^\nu_t \quad \text{if } v_1 \neq v_2$	(Struct Else)
$n[A\langle\tilde{v}\rangle]^\nu_t \equiv n[\{^{\tilde{v}}/_{\tilde{x}}\}P]^\nu_t \quad \text{if } A(\tilde{x}) = P \ \wedge \ \mid \tilde{x} \mid = \mid \tilde{v} \mid$	(Struct Rec)
$M \mid N \equiv N \mid M$	(Struct Par Comm)
$(M \mid N) \mid M' \equiv M \mid (N \mid M')$	(Struct Par Assoc)
$M \mid \mathbf{0} \equiv M$	(Struct Zero Par)
$M \equiv M$	(Struct Refl)
$M \equiv N$ implies $N \equiv M$	(Struct Symm)
$M \equiv M' \wedge M' \equiv M''$ implies $M \equiv M''$	(Struct Trans)
$M \equiv N$ implies $M \mid M' \equiv N \mid M'$, for all M'	(Struct Ctx Par)

to the standard notion of α-conversion. We identify processes and networks up to α-conversion. We assume there are no free variables in our networks. The absence of free variables in networks is trivially maintained as the network evolves. We write $\{^v/_x\}P$ for the substitution of the variable x with the value v in P. We define *structural congruence*, written $\equiv$, as the smallest congruence induced by the laws in Table 2, which is a commutative monoid with respect to the parallel operator. For convenience, structural congruence includes equalities to deal with matching and recursion. We use a number of notational conventions. $\prod_{i\in I} M_i$ means the parallel composition of all sub-networks M_i, for $i \in I$. We identify $\prod_{i\in I} M_i = \mathbf{0}$ if $I = \emptyset$. We write $!\langle v\rangle$ for $!\langle v\rangle.\mathsf{nil}$, and $\langle v\rangle^\delta$ for $\langle v\rangle^\delta.\mathsf{nil}$. We recall that in the active sender process $\langle v\rangle^t.P$ it holds that $t > 0$. However, sometimes, for convenience, we write $\langle v\rangle^0.P$ assuming the syntactic equality $\langle v\rangle^0.P = P$.

Here are some definitions that will be useful in the remainder of the paper. Given a network M, $\mathsf{nds}(M)$ returns the names of the nodes which constitute the network M. For any network M, $\mathsf{actsnd}(M)$ and $\mathsf{actrcv}(M)$ return the set of active senders and active receivers of M, respectively. Thus, for instance, for $N = m[!\langle w\rangle]^\nu_t \mid n[\langle v\rangle^r.P]^{\nu'}_{t'}$ we have $\mathsf{nds}(N) = \{m, n\}$ and $\mathsf{actsnd}(N) = \{n\}$. Given a network M and an active sender $n \in \mathsf{actsnd}(M)$, the function $\mathsf{active}(n, M)$ says for how long the node n will be transmitting. For instance, if N is the network defined as before, $\mathsf{active}(n, N) = r$. If n is not an active sender then $\mathsf{active}(n, N) = 0$. Finally, given a network M and a node $m \in \mathsf{nds}(M)$, the function $\mathsf{ngh}(m, M)$ returns the set of neighbours of m in M. Thus, for N defined as above $\mathsf{ngh}(m, N) = \nu$.

Table 3 LTS - Synchronisation and internal actions

$$\text{(Snd)}\ \frac{-}{m[!\langle v\rangle.P]^{\nu}_{t} \xrightarrow{m!v} m[\langle v\rangle^{\delta_v}.P]^{\nu}_{t}} \qquad \text{(Rcv)}\ \frac{m \in \nu}{n[\lfloor ?(x).P \rfloor Q]^{\nu}_{0} \xrightarrow{m?v} n[(x)_v.P]^{\nu}_{\delta_v}}$$

$$\text{(RcvPar)}\ \frac{M \xrightarrow{m?v} M' \quad N \xrightarrow{m?v} N'}{M \mid N \xrightarrow{m?v} M' \mid N'} \qquad \text{(Sync)}\ \frac{M \xrightarrow{m!v} M' \quad N \xrightarrow{m?v} N'}{M \mid N \xrightarrow{m!v} M' \mid N'}$$

$$\text{(Coll)}\ \frac{m \in \nu \quad t' := \mathsf{max}(t, \delta_v)}{n[(x)_w.P]^{\nu}_{t} \xrightarrow{m?v} n[(x)_{\perp}.P]^{\nu}_{t'}} \qquad \text{(Exp)}\ \frac{m \in \nu \quad W \neq (x)_w.P \quad t' := \mathsf{max}(t, \delta_v)}{n[W]^{\nu}_{t} \xrightarrow{m?v} n[W]^{\nu}_{t'}}$$

$$\text{(OutRng)}\ \frac{m \notin \nu \quad m \neq n}{n[W]^{\nu}_{t} \xrightarrow{m?v} n[W]^{\nu}_{t}} \qquad \text{(Zero)}\ \frac{-}{\mathbf{0} \xrightarrow{m?v} \mathbf{0}}$$

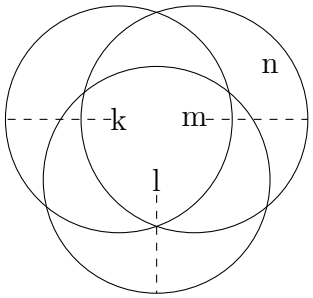
$$\text{(Tau)}\ \frac{-}{m[\lfloor \tau.P \rfloor Q]^{\nu}_{t} \xrightarrow{\tau} m[P]^{\nu}_{t}} \qquad \text{(TauPar)}\ \frac{M \xrightarrow{\tau} M'}{M \mid N \xrightarrow{\tau} M' \mid N}$$

2.1 The Operational Semantics

We have divided our LTS in two sets of rules corresponding to the two main phases of a wireless transmission. Table 3 contains the rules to model both initial synchronisations between a sender and its neighbours, and internal computations within single nodes. Table 4 contains the rules for modelling time passing and transmission ending.

Let us comment on the rules of Table 3. The metavariable λ ranges over the set of labels $\{\tau, m!v, m?v\}$ denoting internal action, broadcasting and reception, respectively. Rule (Snd) models a node starting a broadcast of message v to its neighbours in ν. By maximal progress, a node which is ready to transmit will not be delayed. A transmission fires even if there are no listeners: sending is a *non-blocking* action. Rule (Rcv) models the beginning of the reception of a message v transmitted by a station m. This happens only when the receiver is not exposed to other transmissions i.e. when the exposure indicator is equal to zero. The exposure indicator is then updated because node n will be exposed for the next δ_v instants of time. The reception will finish only when the receiver senses the channel free for a whole time interval (see rule (RcvEnd) of Table 4). Rule (RcvPar) serves to synchronise different receivers on the same transmission originating from a node m. Rule (Sync) serves to synchronise a broadcasting node

Figure 1 Network topology of Example 2.2



m with receivers. In rule (Coll) an active receiver n is exposed to a transmission originating from a node m. This transmission gives rise to a *collision* at n. Rule (Exp) models the exposure of a node n (which is not an active receiver) to a transmission originating from a transmitter m. In this case, n does not take part in the transmission. Notice that a node $n[\lfloor ?(x).P \rfloor Q]_0^\nu$ might execute rule (Exp) instead of (Rcv). This is because a potential (synchronised) receiver might miss the synchronisation with the sender for several reasons (internal misbehaving, radio signals problems, etc). Such a situation will give rise to a failure in reception at n (see rule (σ-Fail) in Table 4). Rule (OutRng) regards nodes which are out of the range of a transmission originating from a node m. Rule (Zero) is similar but regards empty networks. Rule (Tau) models local computations. Rule (TauPar) serves to propagate internal computations on parallel components. Rules (Sync) and (TauPar) have their symmetric counterpart.

Let us explain the rules in Table 3 with an example.

Example 2.2 *Consider the network*

$$Net \stackrel{\text{def}}{=} k[!\langle v\rangle.?(x).P]_0^{\nu_k} \mid l[?(x).Q]_0^{\nu_l} \mid m[!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n}$$

with the following communication topology: $\nu_k = \{l, m, l'\}$, $\nu_l = \{k, m\}$, $\nu_m = \{k, l, n, l', m'\}$ *and* $\nu_n = \{m\}$ *(see Figure 1). There are two possible broadcast communications originating from stations* k *and* m*, respectively. Let us suppose* k *starts broadcasting. By applying rules (Snd), (Rcv), (Exp), (OutRng), (RcvPar) and (Sync) we have:*

$$\begin{array}{rcl} Net & \xrightarrow{k!v} & k[\langle v\rangle^{\delta_v}.?(x).P]_0^{\nu_k} \mid l[(x)_v.Q]_{\delta_v}^{\nu_l} \mid m[!\langle w\rangle]_{\delta_v}^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & = & Net_1 \ . \end{array}$$

Table 4 LTS - Time passing/End transmission

$$(\sigma\text{-Nil})\ \frac{-}{n[\mathsf{nil}]^{\nu}_{t} \xrightarrow{\sigma} n[\mathsf{nil}]^{\nu}_{t-1}} \qquad (\text{Sleep})\ \frac{-}{n[\sigma.P]^{\nu}_{t} \xrightarrow{\sigma} n[P]^{\nu}_{t-1}}$$

$$(\sigma\text{-Rcv})\ \frac{-}{n[\lfloor ?(x).P \rfloor Q]^{\nu}_{0} \xrightarrow{\sigma} n[Q]^{\nu}_{0}} \qquad (\sigma\text{-Fail})\ \frac{t > 0}{n[\lfloor ?(x).P \rfloor Q]^{\nu}_{t} \xrightarrow{\sigma} n[(x)_{\perp}.P]^{\nu}_{t-1}}$$

$$(\sigma\text{-Tau})\ \frac{-}{n[\lfloor \tau.P \rfloor Q]^{\nu}_{t} \xrightarrow{\sigma} n[Q]^{\nu}_{t-1}} \qquad (\text{ActSnd})\ \frac{r > 0}{n[\langle v \rangle^{r}.P]^{\nu}_{t} \xrightarrow{\sigma} n[\langle v \rangle^{r-1}.P]^{\nu}_{t-1}}$$

$$(\text{ActRcv})\ \frac{t > 0}{n[(x)_{v}.P]^{\nu}_{t} \xrightarrow{\sigma} n[(x)_{v}.P]^{\nu}_{t-1}} \qquad (\text{RcvEnd})\ \frac{-}{n[(x)_{v}.P]^{\nu}_{0} \xrightarrow{\sigma} n[\{^{v}/_{x}\}P]^{\nu}_{0}}$$

$$(\sigma\text{-Zero})\ \frac{-}{\mathbf{0} \xrightarrow{\sigma} \mathbf{0}} \qquad (\sigma\text{-Par})\ \frac{M \xrightarrow{\sigma} M' \quad N \xrightarrow{\sigma} N'}{M \mid N \xrightarrow{\sigma} M' \mid N'}$$

By maximal progress, m can not delay its transmission. Supposing $\delta_v < \delta_w$ we have:

$$\begin{array}{rcl} Net_1 & \xrightarrow{m!w} & k[\langle v \rangle^{\delta_v}.?(x).P]^{\nu_k}_{\delta_w} \mid l[(x)_{\perp}.Q]^{\nu_l}_{\delta_w} \mid m[\langle w \rangle^{\delta_w}]^{\nu_m}_{\delta_v} \mid n[(y)_{w}.R]^{\nu_n}_{\delta_w} \\ & = & Net_2 \ . \end{array}$$

Now, node l is exposed to a collision and its reception is doomed to fail. Notice that, although node m was already exposed when it started transmitting, node n will receive correctly the message w from m.

Let us comment on rules of Table 4. Rule (σ-Nil) is straightforward: it simply decreases the exposure tag of the node. This updating of the exposure tag appears in all rules of Table 4, where we assume an arithmetic for positive integers such that $0 - 1 = 0$. Rule (Sleep) models sleeping for one time unit. In rule (σ-Rcv) a timeout fires if no reception has started. Rule (σ-Fail) models a failure of an exposed receiver. This may happen, for instance, when a receiver wakes up in the middle of an ongoing transmission. In rule (σ-Tau) a timeout can fire if no internal actions are executed. Rules (ActSnd) and (ActRcv) represent the passage of time for active senders and active receivers, respectively. When the transmission is over, active senders simple evolve to the next state (we recall that, by convention, $\langle v \rangle^{0}.P = P$). On the other hand, active receivers stop receiving only when the

Table 5 LTS - Matching and recursion

$$\text{(Then)}\ \frac{n[P]^{\nu}_{t} \xrightarrow{\lambda} n[P']^{\nu}_{t'}}{n[[v=v]P,Q]^{\nu}_{t} \xrightarrow{\lambda} n[P']^{\nu}_{t'}} \qquad \text{(Else)}\ \frac{n[Q]^{\nu}_{t} \xrightarrow{\lambda} n[Q']^{\nu}_{t'} \quad v_1 \neq v_2}{n[[v_1=v_2]P,Q]^{\nu}_{t} \xrightarrow{\lambda} n[Q']^{\nu}_{t'}}$$

$$\text{(Rec)}\ \frac{n[\{\tilde{v}/\tilde{x}\}P]^{\nu}_{t} \xrightarrow{\lambda} n[P']^{\nu}_{t'} \quad H(\tilde{x}) \stackrel{\text{def}}{=} P}{n[H\langle\tilde{v}\rangle]^{\nu}_{t} \xrightarrow{\lambda} n[P']^{\nu}_{t'}}$$

channel becomes idle. The end of a reception of a message v is modelled in rule (RcvEnd). As the communication is half-duplex this happens when the receiver senses the channel idle for one time unit. Rule (σ-Zero) is straightforward. Rule (σ-Par) models time synchronisation among the devices.

Example 2.3 *Let us continue with the previous example. Let us show how the system evolves after δ_v and δ_w time units. We recall that $0 < \delta_v < \delta_w$. For simplicity let us define $\delta := \delta_w - \delta_v$:*

$$\begin{array}{lll}
Net_2 & (\xrightarrow{\sigma})^{\delta_v} & k[?(x).P]^{\nu_k}_{\delta} \mid l[(x)_{\perp}.Q]^{\nu_l}_{\delta} \mid m[\langle w\rangle^{\delta}]^{\nu_m}_{0} \mid n[(y)_w.R]^{\nu_n}_{\delta} \\
 & \xrightarrow{\sigma} & k[(x)_{\perp}.P]^{\nu_k}_{\delta-1} \mid l[(x)_{\perp}.Q]^{\nu_l}_{\delta-1} \mid m[\langle w\rangle^{\delta-1}]^{\nu_m}_{0} \mid n[(y)_w.R]^{\nu_n}_{\delta-1} \\
 & (\xrightarrow{\sigma})^{\delta-1} & k[(x)_{\perp}.P]^{\nu_k}_{0} \mid l[(x)_{\perp}.Q]^{\nu_l}_{0} \mid m[\mathsf{nil}]^{\nu_m}_{0} \mid n[(y)_w.R]^{\nu_n}_{0} \\
 & \xrightarrow{\sigma} & k[\{^{\perp}/_{x}\}P]^{\nu_k}_{0} \mid l[\{^{\perp}/_{x}\}Q]^{\nu_l}_{0} \mid m[\mathsf{nil}]^{\nu_m}_{0} \mid n[\{^{w}/_{y}\}R]^{\nu_n}_{0} \ .
\end{array}$$

Notice that, after δ_v instants of time, node k will start a reception in the middle of an ongoing transmission (the transmitter being m). This will lead to a failure at k.

In Table 5 we report the obvious rules for nodes containing matching and recursive processes (we recall that only guarded recursion is allowed).

In the remainder of this article we will use the notion of execution trace. A *trace* is a sequence of labelled transitions. If Λ is a sequence of labels $\lambda_1\lambda_2\ldots\lambda_n$, with $\lambda_i \neq \tau$ for $1 \leq i \leq n$, we write $M \stackrel{\Lambda}{\Longrightarrow} N$ to mean

$$M(\xrightarrow{\tau})^* \xrightarrow{\lambda_1} (\xrightarrow{\tau})^* \ldots (\xrightarrow{\tau})^* \xrightarrow{\lambda_n} (\xrightarrow{\tau})^* N$$

where $(\xrightarrow{\tau})^*$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$.

Below, we report a number of basic properties of our LTS.

Proposition 2.4 *Let M, M_1 and M_2 be networks.*

1. *$m \notin \mathsf{nds}(M)$ if and only if $M \xrightarrow{m?v} M'$, for some network M'.*

2. *$M_1 \mid M_2 \xrightarrow{m?v} N$ if and only if there are N_1 and N_2 such that $M_1 \xrightarrow{m?v} N_1$, $M_2 \xrightarrow{m?v} N_2$ and $N = N_1 \mid N_2$.*

3. *If $M \xrightarrow{m!v} M'$ then $M \equiv m[!\langle v\rangle.P]^{\nu}_t \mid N$, for some ν, t, P and N, and there is N' such that $m[!\langle v\rangle.P]^{\nu}_t \xrightarrow{m!v} m[\langle v\rangle^{\delta v}.P]^{\nu}_t$, $N \xrightarrow{m?v} N'$ and $M' \equiv m[\langle v\rangle^{\delta v}.P]^{\nu}_t \mid N'$.*

4. *If $M \xrightarrow{\tau} M'$ then $M \equiv m[\lfloor \tau.P \rfloor Q]^{\nu}_t \mid N$, for some m, ν, t, P, Q and N such that $m[\lfloor \tau.P \rfloor Q]^{\nu}_t \xrightarrow{\tau} m[P]^{\nu}_t$ and $M' \equiv m[P]^{\nu}_t \mid N$.*

5. *$M_1 \mid M_2 \xrightarrow{\sigma} N$ if and only if there are N_1 and N_2 such that $M_1 \xrightarrow{\sigma} N_1$, $M_2 \xrightarrow{\sigma} N_2$ and $N = N_1 \mid N_2$.*

Proof See the Appendix. □

2.2 Well-formedness

The syntax presented in Table 1 allows us to derive inconsistent networks, i.e. networks that do not have a realistic counterpart. Below we give a number of definitions to rule out ill-formed networks. We recall that $\equiv$ denotes structural congruence.

As network addresses are unique, we assume that there cannot be two nodes with the same name in the same network.

Definition 2.5 (Node uniqueness) *A network M is said to be* node-unique *if whenever $M \equiv M_1 \mid m[W_1]^{\nu}_t \mid n[W_2]^{\nu'}_{t'}$ it holds that $m \neq n$.*

We also assume network connectivity, i.e. all nodes are connected to each other, although not always directly. This is because time synchronisation can be achieved only in connected networks. Moreover, in our networks, all nodes have the same transmission range. Formally,

Definition 2.6 (Network connectivity) *A network M is said to be* connected *if*

- *whenever $M \equiv N \mid m[W_1]^{\nu}_t \mid n[W_2]^{\nu'}_{t'}$ with $m \in \nu'$ it holds that $n \in \nu$;*
- *for all $m, n \in \mathsf{nds}(M)$ there is a sequence of nodes $m_1, \ldots, m_k \in \mathsf{nds}(M)$, with neighbouring $\nu_1, \ldots, \nu_k$, respectively, such that $m{=}m_1$, $n{=}m_k$ and $m_i \in \nu_{i+1}$, for $1 \leq i \leq k{-}1$.*

The next definition is about the consistency of exposure indicators of nodes. Intuitively, the exposure indicators of active senders and active receivers must be consistent with their current activity (transmission/reception). Moreover, the neighbours of active senders must have their exposure indicators consistent with the duration of the transmission.

Definition 2.7 (Exposure consistency) *A network M is said to be* exposure-consistent *if the following conditions are satisfied.*

1. *If $M \equiv N \mid m[(x)_v.P]_t^\nu$, with $v \neq \bot$, then $0 \leq t \leq \delta_v$.*
2. *If $M \equiv N \mid m[\langle v \rangle^r.P]_t^\nu$, then $r \leq \delta_v$.*
3. *If $M \equiv N \mid m[\langle v \rangle^r.P]_t^\nu \mid n[W]_{t'}^{\nu'}$, with $m \in \nu'$, then $0 < r \leq t'$.*
4. *Let $M \equiv N \mid n[W]_t^\nu$ with $t{>}0$. If $\mathsf{active}(k, N) \neq t$ for all k in $\nu \cap \mathsf{actsnd}(N)$, then there is k' in $\nu \backslash \mathsf{nds}(N)$ such that whenever $N \equiv N' \mid l[W']_{t'}^{\nu'}$, with $k' \in \nu'$, then $t' \geq t$.*

The next definition is about the consistency of transmitting stations. The first and the second part are about successful transmissions, while the third part is about collisions.

Definition 2.8 (Transmission consistency) *A network M is said to be* transmission-consistent *if the following conditions are satisfied.*

1. *If $M \equiv N \mid n[(x)_v.Q]_t^\nu$ and $v \neq \bot$, then $\mid \mathsf{actsnd}(N) \cap \nu \mid \leq 1$.*
2. *If $M \equiv N \mid m[\langle w \rangle^r.P]_t^\nu \mid n[(x)_v.Q]_{t'}^{\nu'}$, with $m \in \nu'$ and $v \neq \bot$, then (i) $v = w$, and (ii) $r = t'$.*
3. *If $M \equiv N \mid n[(x)_v.P]_t^\nu$, with $\mid \mathsf{actsnd}(N) \cap \nu \mid > 1$, then $v = \bot$.*

Definition 2.9 (Well-formedness) *A network M is said to be* well-formed *if it is node-unique, connected, exposure-consistent and transmission-consistent.*

We prove that network well-formedness is preserved at runtime. In particular, the preservation of exposure- and transmission-consistency are the more interesting and delicate results.

Theorem 2.10 (Subject reduction) *If M is a well-formed network, and $M \xrightarrow{\lambda} M'$ for some label λ and network M', then M' is well-formed as well.*

Proof By transition induction. □

3 Time Properties

We start proving three desirable time properties of TCWS: time determinism, patience and maximal progress.

Theorem 3.1 formalises the deterministic nature of time passing: a network can reach at most one new state by executing the action σ.

Theorem 3.1 (Time Determinism) *Let M be a well-formed network. If $M \xrightarrow{\sigma} M'$ and $M \xrightarrow{\sigma} M''$ then M' and M'' are syntactically the same.*

Proof By induction on the length of the proof of $M \xrightarrow{\sigma} M'$. □

In [20, 39], the maximal progress property says that processes communicate as soon as a possibility of communication arises. However, unlike [20, 39], in our calculus message transmission requires a positive amount of time. So, we generalise the property saying that transmissions cannot be delayed.

Theorem 3.2 (Maximal Progress) *Let M be a well-formed network. If there is N such that $M \xrightarrow{m!v} N$ then $M \xrightarrow{\sigma} M'$ for no network M'.*

Proof Because sender nodes cannot perform σ-actions. □

The last time property is patience. In [20, 39] patience guarantees that a process will wait indefinitely until it can communicate. In our setting, this means that if no transmission can start then it must be possible to execute a σ-action to let time pass.

Theorem 3.3 (Patience) *Let M be a well-formed network. If $M \xrightarrow{m!v} M'$ for no network M' then there is a network N such that $M \xrightarrow{\sigma} N$.*

Proof By contradiction and then by induction on the structure of M. □

4 Case studies

The calculus defined in Section 2 should be considered as a core language for the specification of wireless systems. As many other process calculi, TCWS can be extended with useful constructs (basically, syntactic sugar) which do not introduce new concepts. We report below the extensions we are interested in. For commodity, values are extended with functions.[4] We adopt a polyadic version of the calculus where tuple of values are transmitted. Thus, for instance, the process $!\langle v, v', w\rangle.P$ denotes the broadcast of a tuple containing three values. We

[4]Functions are already implicitly used in the core calculus when writing δ_v to denote the time necessary to transmit value v: $\delta()$ is a function that given a data value v returns an integer.

assume standard tuple destructors fst(), snd(), etc. returning the corresponding component of a tuple, if it exists, and the value $\perp$ otherwise. The matching construct $[u = w]P, Q$ is extended to check the conjunction and/or disjunction of more equalities. Last but not least, in the process definition we assume also process variables; this is not a big extension as values in TCWS represent data packets, so they can also contain code. We call *extended* TCWS the calculus obtained by extending the syntax of TCWS with the just mentioned constructs. The operational semantics of these constructs is completely standard.

The goal of this section is to show the expressiveness of our extended TCWS by defining a number protocols/applications. We start with MAC-layer protocols, such as CSMA and CSMA/CA then we pass to study a sensor network link layer security protocol, called MiniSec.

4.1 Carrier Sense Multiple Access

The *Carrier Sense Multiple Access* (CSMA) scheme [22] 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 (i.e. in the next instant of time [5]); if the channel is busy (i.e. some other station is transmitting) the device keeps listening the channel until it becomes idle and then starts transmitting immediately. This strategy is called *1-persistent* CSMA. More generally, in a *p-persistent* CSMA strategy (where p is a probability) the sender transmits with probability p, and waits for the next available time slot, with probability $1 - p$.

In our calculus, we can easily model the 1-persistent CSMA scheme using receivers with timeout where the sender process $!\langle v\rangle.P$ is replaced by the process defined below:

$$!!\langle v\rangle.P \stackrel{\text{def}}{=} \lfloor ?(x).!\langle v\rangle.P \rfloor !\langle v\rangle.P \ .$$

The next example shows how 1-persistent CSMA affects the behaviour of a wireless system. Let us consider the network:

$$Net \stackrel{\text{def}}{=} k[!!\langle v\rangle.?(x).P]_0^{\nu_k} \mid l[?(x).Q]_0^{\nu_l} \mid m[\sigma.!!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n}$$

with the following communication topology: $\nu_k = \{l, m, l'\}$, $\nu_l = \{k, m\}$, $\nu_m = \{k, l, n, l', m'\}$ and $\nu_n = \{m\}$ (see Figure 1 at page 8). Here, node k senses the channel free and, according to the CSMA scheme, in the next instant of time, it will start transmitting. Thus,

$$\begin{array}{rcl} Net & \stackrel{\sigma}{\longrightarrow} & k[!\langle v\rangle.?(x).P]_0^{\nu_k} \mid l[?(x).Q]_0^{\nu_l} \mid m[!!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & = & Net_1 \ . \end{array}$$

[5] We recall that in wireless systems channels are half-duplex.

In Net_1, node m is currently listening the channel to check whether it is free. By applying rules (Snd), (Rcv), (Exp), (OutRng), (RcvPar) and (Sync) node k can start transmitting:

$$\begin{array}{rcl} Net_1 & \xrightarrow{k!v} & k[\langle v\rangle^{\delta_v}.?(x).P]_0^{\nu_k} \mid l[(x)_v.Q]_{\delta_v}^{\nu_l} \mid m[(x)_v.!\langle w\rangle]_{\delta_v}^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & = & Net_2 \ . \end{array}$$

Now, since k has already started its transmission, node m senses the channel busy and it must wait until the channel becomes free. Notice that in this manner there are no collisions at l and/or k. In fact, after δ_v instants of time we have:

$$\begin{array}{rcl} Net_2 & (\xrightarrow{\sigma})^{\delta_v} & k[?(x).P]_0^{\nu_k} \mid l[(x)_v.Q]_0^{\nu_l} \mid m[(x)_v.!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & \xrightarrow{\sigma} & k[?(x).P]_0^{\nu_k} \mid l[\{^v/_x\}Q]_0^{\nu_l} \mid m[!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & = & Net_3 \end{array}$$

where node l has successfully received value v from k. Notice that after δ_v instants of time node m senses the channel free, and by maximal progress it will start transmitting in the next instant of time.

However, using a CSMA scheme, there is always a chance of stations starting transmitting at exactly the same time, caused by the fact that different stations sensed the medium free and decided to transmit at once. As an example, consider the network:

$$Net' \stackrel{\text{def}}{=} k[!!\langle v\rangle.?(x).P]_0^{\nu_k} \mid l[?(x).Q]_0^{\nu_l} \mid m[!!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n}$$

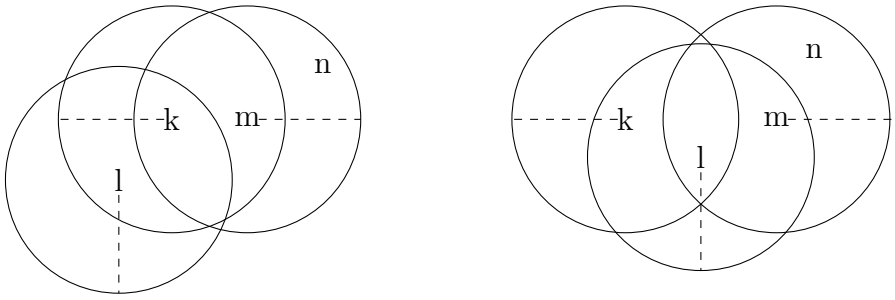
with the same communication topology as before. In this scenario, both nodes k and m want to start transmitting. And since both of them sense the channel free, they will start transmitting in the next instant of time. Thus, assuming $\delta_v < \delta_w$, we have:

$$\begin{array}{rcl} Net' & \xrightarrow{\sigma} & k[!\langle v\rangle.?(x).P]_0^{\nu_k} \mid l[?(x).Q]_0^{\nu_l} \mid m[!\langle w\rangle]_0^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & \xrightarrow{k!v} & k[\langle v\rangle^{\delta_v}.?(x).P]_0^{\nu_k} \mid l[(x)_v.Q]_{\delta_v}^{\nu_l} \mid m[!\langle w\rangle]_{\delta_v}^{\nu_m} \mid n[?(y).R]_0^{\nu_n} \\ & \xrightarrow{m!w} & k[\langle v\rangle^{\delta_v}.?(x).P]_{\delta_w}^{\nu_k} \mid l[(x)_\perp.Q]_{\delta_w}^{\nu_l} \mid m[\langle w\rangle^{\delta_w}]_{\delta_v}^{\nu_m} \mid n[(y)_w.R]_{\delta_w}^{\nu_n} \ . \end{array}$$

In this situation, node l is exposed to a collision caused by the two transmissions.

It should be pointed out that the CSMA scheme is not always a good idea. Let us consider, for instance, the previous network Net where nodes l and m are not neighbours anymore, that is $\nu_l = \{k\}$ and $\nu_m = \{k, n, l', m'\}$ (see the first

Figure 2 Exposed and Hidden terminal problem



picture in Figure 2). Now, suppose that m wants to send a message to n. Then, the CSMA scheme delays the transmission without any reason, only because m is exposed to the transmission originating from k. This is a well-known problem, introduced by CSMA, called *exposed terminal problem*.

The CSMA scheme suffers another well-known problem called *hidden terminal problem*. This happens when two transmitters sense the channel free, because they are not in each other's transmission cell, and start transmitting causing a collision to a third node lying in the transmission cells of both. As an example, you can consider, for instance, the previous network Net with the following communication topology: $\nu_k = \{l, l'\}$, $\nu_l = \{k, m\}$, $\nu_m = \{l, n, l', m'\}$ and $\nu_n = \{m\}$ (see the second picture in Figure 2). In this case, both transmissions at k and m will fire causing (after two instants of time) an interference at l.

4.1.1 Collision Avoidance

In *unicast communications*, to reduce the number of collisions due to the hidden terminal problem, the CSMA scheme may be used with a *Collision Avoidance* (CA) mechanism together with a *Positive Acknowledgement Scheme*. With the latter, receivers check the integrity of the data frame and if no errors occur they send an acknowledgement (ack) to the sender. Reception of the ack ensures the transmitter that the data frame has been successfully received. If the sender does not receive the ack frame, then the receiver might have not received the data. In this case, the sender will try to retransmit the data frame for a given number of times.

The Collision Avoidance mechanism is achieved by distributing reservation information announcing the impending use of the medium. This mechanism is also called *virtual carrier sense*. A device wishing to transmit a data frame will first transmit a short control packet RTS (request to send), which will include the

source, the destination, and the duration of the whole transaction (i.e. the transmission of the data frame together with the returning ack frame). If the medium is free, the destination station will respond with a control packet called CTS (clear to send), which will include the same duration information. All stations receiving either the RTS (from the sender) and/or the CTS (from the receiver), will learn of the medium reservation. More precisely, they will set their *network allocation vector* (NAV) register to the maximum among the current value stored in their NAV and the duration time carried in the RTS/CTS frame. The NAV may be thought of as a counter, which counts down to zero at a uniform rate. When the NAV is zero, the virtual carrier sense indication is that the medium is idle; when nonzero, the medium is supposed to be busy and the station must remain silent. Upon receiving an RTS, a node returns a CTS frame only if its NAV value is zero, otherwise no CTS is sent. Thus, a sender will see no CTS if its RTS packet has collided with another transmission at the receiver, or if the receiver's NAV indicates that the network is not available. In this case, the transmitter will repeat the process according to some *backoff algorithm*.

The goal of the RTS/CTS mechanism is to reduce the probability of a collision at the receiver to the short duration of the RTS transmission. In fact, if a station hears the CTS then it "reserves" the medium as busy until the end of the transmission. The duration field in the RTS frame also protects the transmitter area from potential collisions during the reception of the ack (by stations that are out of range from the acknowledging station).

Notice that the virtual carrier sense cannot be applied to multicast and broadcast packets because there would be multiple recipients for the RTS, and thus potentially multiple concurrent senders of the corresponding CTS. Notice also that the virtual carrier sense works correctly under the assumption that all devices have the same transmission range. In order to understand that, think of two nodes m and n such that n is in the transmission range of m but not vice versa. Suppose that n receives a RTS frame from another node and sends it back the CTS frame starting the reception of the data frame. In this scenario, since the node m did not hear the CTS frame it could start transmitting causing an interference at n.

In Table 6, we provide an encoding of a sender and a receiver process, written in our extended TCWS, and respecting the CSMA/CA protocol. For brevity, in sub-terms of the form $?(x).P$, instead of using the standard tuple destructors, we write x_i in P to mean the i-th component of the tuple that will be received at x, if this component is defined, and $\perp$ otherwise.

The process $\mathtt{SND}(m, v, n, P)$ runs at node m and tries to transmit the value v to node n, being P the continuation. The sending takes into account both physical and virtual carrier sense. If the channel is sensed free the process sends an RTS

Table 6 CSMA/CA

Sender at m:

$\mathtt{SND}(m,v,n,P) \stackrel{\text{def}}{=}$	$\lfloor ?(x).$	Do physical carrier sense,
	$\quad [x_1 = \mathsf{rts} \vee x_1 = \mathsf{cts}]$	If a RTS/CTS is received
	$\quad\quad \mathtt{NAV}\langle x_4, \mathtt{SND}\langle m,v,n,P\rangle\rangle,$	update the NAV,
	$\quad\quad \mathtt{SND}\langle m,v,n,P\rangle \rfloor$	otherwise, restart.
	$!\langle \mathsf{rts}, m, n, \delta\rangle.\sigma.\mathtt{CTS}\langle m,v,n,P\rangle$	If channel is free send RTS.
$\mathtt{CTS}(m,v,n,P) \stackrel{\text{def}}{=}$	$\lfloor ?(x).$	
	$\quad [x = (\mathsf{cts}, m, n, \cdot)]$	If the right CTS is received
	$\quad\quad !\langle v\rangle.\sigma.\mathtt{ACK}\langle m,v,n,P\rangle,$	then start transmitting v,
	$\quad\quad \mathtt{NAV}\langle \mathrm{bo}(m), \mathtt{SND}\langle m,v,n,P\rangle\rangle \rfloor$	if not then wait for $\mathrm{bo}(m)$
	$\mathtt{NAV}\langle \mathrm{bo}(m), \mathtt{SND}\langle m,v,n,P\rangle\rangle$	If timeout wait for $\mathrm{bo}(m)$.
$\mathtt{ACK}(m,v,n,P) \stackrel{\text{def}}{=}$	$\lfloor ?(x).[x = (\mathsf{ack}, m, n)]P,$	If ACK is received then P
	$\quad\quad \mathtt{SND}\langle m,v,n,P\rangle \rfloor$	else, restart transmission.
	$\mathtt{SND}\langle m,v,n,P\rangle$	If timeout restart transmission.
$\mathtt{NAV}(\delta,Q) \stackrel{\text{def}}{=}$	$[\delta = 0]Q, \lfloor ?(x).$	If NAV is zero then Q, else
	$\quad [x_1 = \mathsf{rts} \vee x_1 = \mathsf{cts}]$	if a RTS/CTS is received,
	$\quad\quad \mathtt{NAV}\langle \max(x_4,\delta) - \delta_x - 1, Q\rangle,$	then update the NAV
	$\quad\quad \mathtt{NAV}\langle \delta - \delta_x - 1, Q\rangle \rfloor$	else decrease the NAV.
	$\mathtt{NAV}\langle \delta - 1, Q\rangle$	If timeout decrease the NAV.

Receiver at n:

$\mathtt{RCV}(n,y,R) \stackrel{\text{def}}{=}$	$\lfloor ?(x).$	
	$\quad [x_1 = \mathsf{rts}]$	If a RTS packet is received
	$\quad\quad [x_3 = n]$	with destination n
	$\quad\quad\quad !\langle \mathsf{cts}, x_2, n, x_4\rangle.\sigma.R',$	then reply with a CTS pkt
	$\quad\quad\quad \mathtt{NAV}\langle x_4, \mathtt{RCV}\langle n,y,R\rangle\rangle,$	otherwise, update the NAV
	$\quad [x_1 = \mathsf{cts}]$	if a CTS is received
	$\quad\quad \mathtt{NAV}\langle x_4, \mathtt{RCV}\langle n,y,R\rangle\rangle,$	update the NAV,
	$\quad\quad \mathtt{RCV}\langle n,y,R\rangle \rfloor$	otherwise restart.
	$\mathtt{RCV}\langle n,y,R\rangle$	If timeout then restart.
$R' \stackrel{\text{def}}{=}$	$\lfloor ?(y).[y = \bot]$	Receive data and check it
	$\quad\quad \mathtt{RCV}\langle n,y,R\rangle,$	if there is a collision restart
	$\quad\quad !\langle \mathsf{ack}, x_4, n\rangle.\sigma.R \rfloor$	otherwise, ack and continue.
	$\mathtt{RCV}\langle n,y,R\rangle$	If timeout restart reception.

packet and then, in the next time interval, move to process $\texttt{CTS}\langle m, v, n, P\rangle$ to wait for the CTS packet. If the CTS packet is not received in the current instant of time, the process will remain silent for an amount of time calculated by means of a backoff algorithm/function bo(). For simplicity, our backoff function depends on the logical address of the node; thus, different nodes have different backoff periods. When the CTS is received the value v is transmitted. Then, the sender moves to state $\texttt{ACK}\langle m, v, n, P\rangle$ and waits for an ACK. The process $\texttt{NAV}(\delta, Q)$ takes care of the virtual carrier sense by updating the NAV register with the delays contained in the control packets RTS/CTS. In this process, δ_x denotes the time required to receive the current packet (we recall that $\delta()$ is a function). Thus, when x is instantiated with v, δ_x becomes δ_v. We recall that since the communication is half-duplex the time required to receive a packet v is actually $\delta_v + 1$.

The receiver process $\texttt{RCV}(n, y, R)$ is supposed to run at node n waiting for a control packet. If a CTS packet is received then the NAV register is updated. Otherwise, if a RTS packet is received, with destination n, the receiver replies with a CTS packet and then waits for the data. If the data is received correctly an ACK is sent to the transmitter.

Let us write down some simple systems where nodes adopt the CSMA/CA protocol. The goal of these examples is to show that the CSMA/CA protocol may fail in avoiding communication collisions in different ways. For simplicity, we assume that the RTS/CTS packets require one time interval for their transmission.

Let us start with the following system:

$$Sys \stackrel{\text{def}}{=} l[\texttt{SND}\langle l, w, n, Q\rangle]_0^{\nu_l} \mid m[\texttt{SND}\langle m, v, n, P\rangle]_0^{\nu_m} \mid n[\texttt{RCV}\langle n, y, R\rangle]_0^{\nu_n}$$

with $n \in \nu_l$, $n \in \nu_m$ and $\{l, m\} \subseteq \nu_n$. Here, the following execution trace

$$Sys \xrightarrow{\sigma} \xrightarrow{l!\mathsf{rts}} \xrightarrow{m!\mathsf{rts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \ldots\ldots$$

denotes a collision at n caused by the transmission of two different RTS packets, transmitted by l and m, respectively. As a consequence, according to the protocol, the two RTS packets will be resend in two different instants of time by using the backoff algorithm and relying on the fact that $\text{bo}(l) \neq \text{bo}(m)$. The same problem would occur if the process running at l would be $\sigma.\texttt{SND}\langle l, w, n, Q\rangle$, with l not in the transmission range of m, i.e. $m \notin \nu_l$ and $l \notin \nu_m$. In this case, the physical carrier sense at l could not hear the RTS packet of m.

Let us consider now a slightly different system:

$$Sys_1 \stackrel{\text{def}}{=} l[\sigma.\sigma.\texttt{SND}\langle l, w, n, Q\rangle]_0^{\nu_l} \mid m[\texttt{SND}\langle m, v, n, P\rangle]_0^{\nu_m} \mid n[\texttt{RCV}\langle n, y, R\rangle]_0^{\nu_n}$$

where $\{m,n\} \subseteq \nu_l$, $\{l,n\} \subseteq \nu_m$ and $\{l,m\} \subseteq \nu_n$. Here, the following execution trace

$$Sys_1 \xrightarrow{\sigma} \xrightarrow{m!\mathsf{rts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \xrightarrow{n!\mathsf{cts}} \xrightarrow{l!\mathsf{rts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \ldots\ldots$$

describes a collision at m caused by the simultaneous transmissions of the CTS packet of n and the RTS packet of l. Also in this case the protocol will restart by relying on the backoff algorithm. The same problem would occur if the process running at l would be $\sigma.\sigma.\sigma.\mathtt{SND}\langle l,w,n,Q\rangle$, with l in the transmission range of m but not in that of n. In this case, the physical carrier sense at l would not help in hearing the CTS packet of n.

A different situation emerges in the following system:

$$Sys_2 \stackrel{\text{def}}{=} l[\sigma.\sigma.\sigma.\sigma.\mathtt{SND}\langle l,w,n,Q\rangle]_0^{\nu_l} \mid m[\mathtt{SND}\langle m,v,n,P\rangle]_0^{\nu_m} \mid n[\mathtt{RCV}\langle n,y,R\rangle]_0^{\nu_n}$$

with again $\{m,n\} \subseteq \nu_l$, $\{l,n\} \subseteq \nu_m$ and $\{l,m\} \subseteq \nu_n$. Here, the following execution trace is possible:

$$Sys_2 \xrightarrow{\sigma} \xrightarrow{m!\mathsf{rts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \xrightarrow{n!\mathsf{cts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \xrightarrow{m!v} \xrightarrow{l!\mathsf{rts}} \ldots\ldots$$

This denotes a *more serious collision* at n on the transmission of data v. This collision is more problematic as it requires at least δ_v+1 time units to be detected at destination. The collision is due to the fact that the node l sleeps while the RTS/CTS packets are exchanged. Then, after four time intervals, l wakes up, senses the channel free, and starts transmitting its RTS packet in the next time interval. Notice that if l would have slept for a longer period then the physical carrier sense at l would have heard the transmission originating from m, thus preventing the collision.

Finally, let us consider the system

$$Sys_3 \stackrel{\text{def}}{=} l[\sigma^i.\mathtt{SND}\langle l,w,n,Q\rangle]_0^{\nu_l} \mid m[\mathtt{SND}\langle m,v,n,P\rangle]_0^{\nu_m} \mid n[\mathtt{RCV}\langle n,y,R\rangle]_0^{\nu_n}$$

with $4 \leq i \leq \delta_v$+6 and l not in the transmission range of m, hence $n \in \nu_l$, $m \notin \nu_l$, $n \in \nu_m$, $l \notin \nu_m$, and $\{l,m\} \subseteq \nu_n$. In this case, l misses the RTS/CTS packets because it is sleeping. When it wakes up, it senses the channel free and it starts transmitting its RTS by causing a collision at n on the transmission of the data v (assuming $\delta_v > 1$). As an example, for $i = 5$, the execution trace is:

$$Sys_3 \xrightarrow{\sigma} \xrightarrow{m!\mathsf{rts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \xrightarrow{n!\mathsf{cts}} \xrightarrow{\sigma} \xrightarrow{\sigma} \xrightarrow{m!v} \xrightarrow{\sigma} \xrightarrow{l!\mathsf{rts}} \ldots\ldots$$

4.2 The MiniSec protocol

MiniSec [28] is a secure network layer protocol for wireless sensor networks. Basically, it improves on two well-known sensor network link layer protocols such

as TinySec [23] and ZigBee [3]. MiniSec obtains the best of both protocols by achieving three basic goals: *data secrecy*, *authentication* and *protection against replay attacks* (when the attacker replay packets at a later time).

The first two goals are obtained by using pre-deployed shared keys: a receiver can always authenticate (and thus access) a packet using a group-key k_{G}. More precisely, in order to achieve secrecy and authentication, MiniSec adopts *Offset CodeBook* (OCB) [42], a block-cipher operation mode well-suited for sensor networks. OCB operates as follows. Let v be an arbitrary message that needs to be encrypted and authenticated, k be the encryption key (which is the key used by the underlying block cipher), and N be a non-repeating nonce. Then, OCB takes in v, k, and N and generates the ciphertext core C. After that, by using the plaintext v, and the ciphertext C, OCB generates a tag. Thus, the final output of $\mathrm{OCB}(k, v, N)$ is the pair (C, tag). To decrypt a ciphertext C, the receiver performs the reverse process $\mathrm{OCB}^{-1}(k, C, N)$ trying to obtain the plaintext v. If the receiver computes a pair (v', tag') with $tag' = tag$ then $v' = v$, otherwise the ciphertext is considered to be invalid.

Protection against replay attacks is achieved by adopting a *loosely time synchronisation* between sender and receiver(s) following a *sliding-windows approach*. The nodes of the network agree on the passage of time intervals called epochs: when a sender builds a packet it includes as a nonce the current epoch, so that a receiver can know at which epoch the received packet was sent. An epoch is defined as the maximum time required to complete a local broadcast. If ΔN represents the maximum network latency and ΔT the maximum clock synchronisation error, then the length of each epoch is exactly $E = 2\Delta T + \Delta N$. By using the current epoch number as the nonce for OCB-encryption, the protocol defends against replay attacks from older epochs. Unfortunately, because of time synchronisation errors and network latency, such a scheme experiences many false positives at epoch transitions, as legitimate packets sent from the previous epoch will be discarded. The solution proposed by MiniSec is to perform decryption with two possible candidate epoch values for the nonce. Thus, if a valid packet had been sent at the beginning of an epoch, an attacker can replay that packet for at most the remainder of the epoch as well as $\Delta T + \Delta N$ time units of the next epoch. As a consequence, the maximum *window of vulnerability* for replay attacks is $3\Delta T + 2\Delta N$, which intuitively represents the maximum packet delay between its dispatch and its retrieval.

MiniSec has two operating modes: unicast and broadcast, henceforth known as MiniSec-U and MiniSec-B. In this article, we focus on the latter because the unicast variant does not present particular modelling interest.

In Table 7, we provide a specification of MiniSec in TCWS with some simplifications. We extend the values of our calculus with a few simple functions.

Table 7 MiniSec specification

Sender:

S_i^j	$= \lfloor \tau.$	Encrypt payload and epoch
	$\quad !\langle \mathrm{OCB}(k_\mathrm{G}, p, i)\rangle.OK\_SND \rfloor$	broadcast ciphertext,
	$\quad [j+1 = E]S_{i+1}^0, S_i^{j+1}$	if timeout go to next state.

Receiver:

R_i^j	$= \lfloor ?(x).[j+\delta_x+1 \geq E]P_{i+1}^{j+\delta_x+1-E},\ P_i^{j+\delta_x+1} \rfloor$	Start reception
	$\quad [j+1 = E]R_{i+1}^0, R_i^{j+1}$	if timeout then restart.
P_i^j	$= [\mathrm{snd}(x) = \mathrm{snd}(\mathrm{OCB}^{-1}(k_\mathrm{G}, \mathrm{fst}(x), i))]$	If msg comes from epoch i
	$\quad !\langle \mathsf{auth}_i\rangle.OK\_RCV,$	signal msg authentication
	$\quad [\mathrm{snd}(x) = \mathrm{snd}(\mathrm{OCB}^{-1}(k_\mathrm{G}, \mathrm{fst}(x), i-1))]$	if msg comes from epoch $i-1$
	$\quad !\langle \mathsf{auth}_{i-1}\rangle.OK\_RCV,$	signal msg authentication
	$\quad R_i^j$	otherwise restart.

In particular, OCB() and $\mathrm{OCB}^{-1}()$ represent the OCB encoding and its reverse, respectively. We split an epoch in E time intervals. The protocol contemplates a sender and a receiver process. The sender process S_i^j is very simple: it builds a tuple with the payload p, the current epoch i, which acts as a nonce, and then encrypts this information using the OCB algorithm and the group-key k_G. The resulting ciphertext is broadcast. The transmission of a packet v takes δ_v instants of time, with $1 \leq \delta_v \leq \Delta N$. Moreover, message loss can affect a transmission for at most ΔT instants of time, which represents the maximum clock synchronisation error. In process S_i^j the variable j denotes the offset counting σ-actions within an epoch. As our epochs consists of E time intervals, we have $0 \leq j \leq E-1$. The receiver is a bit more complicated. Upon successful reception, a receiver decrypts the information using the group-key k_G, and then proceeds to verify the epoch of the received packet. As said above, a receiver accepts packets sent during the current epoch or the previous one. This is indirectly done by checking the tags returned by encryption and decryption. If a packet is accepted then an authentication message is sent and the receiver process restart by updating the epoch counter and the offset.

For simplicity our specification of MiniSec considers only two nodes, to yield an easier to read model:

$$\mathrm{MiniSec} \stackrel{\mathrm{def}}{=} m[S_0^0]_0^{\nu_m} \mid n[R_0^0]_0^{\nu_n}$$

where m is the sender and n is the receiver, with $m \in \nu_n$ and $n \in \nu_m$. This does not lose any generality with respect to the case where there are more receivers.

Let us formalise now a few properties on the behaviour of MiniSec. Let $\#^{\sigma}(\Lambda)$ be the occurrences of σ actions in the execution trace Λ. Let p_i be an abbreviation for $\mathrm{OCB}(k_{\mathrm{G}}, p, i)$, the packet sent by the sender m at epoch i. We assume standard Dolev-Yao assumptions for the attacker.

The next result says that only fresh packets are authenticated.

Proposition 4.1 (Packet freshness) *If node n authenticates a packet p_i in epoch k then p_i was sent by m either in the current epoch k or in the previous one, i.e. $i \leq k \leq i+1$.*

Proof See the Appendix. □

The following proposition says that if there is a replay attack then it must occur within a vulnerability window. In fact, if node n receives the packet p_i a number of instants of time later than it was originally transmitted by m, then p_i has been replayed by the attacker. In this case, p_i will be authenticated only if the attack occurred in the vulnerability window.

Proposition 4.2 (Vulnerability window) *If the protocol evolves as*

$$\text{MiniSec} \stackrel{\Lambda_1}{\Longrightarrow} \xrightarrow{m!p_i} \stackrel{\Lambda_2}{\Longrightarrow} \xrightarrow{n?p_i} \stackrel{\Lambda_3}{\Longrightarrow} \xrightarrow{n!\mathsf{auth}_i}$$

then $\#^{\sigma}(\Lambda_2) \leq 2\Delta N + 3\Delta T$.

Proof See the Appendix. □

5 Observational semantics

In this section we propose a notion of timed behavioural equivalence for our wireless networks. Our starting point is Milner and Sangiorgi's barbed congruence [32], a standard contextually-defined program equivalence. Intuitively, two terms are barbed congruent if they have the same *observables*, in all possible contexts, under all possible *evolutions*. The definition of barbed congruence strongly relies on two crucial concepts: a reduction semantics to describe how a system evolves, and a notion of observable which says what the environment can observe in a system.

From the operational semantics given in Section 2.1 it should be clear that the evolution of our wireless networks depends on message transmission and internal actions within nodes. Thus, we can define the reduction relation $\twoheadrightarrow$ between networks using the following inference rule:

$$\text{(Red1)}\ \frac{M \xrightarrow{m!v} N}{M \twoheadrightarrow N} \qquad\qquad \text{(Red2)}\ \frac{M \xrightarrow{\tau} N}{M \twoheadrightarrow N}$$

We write $\twoheadrightarrow^*$ for the reflexive and transitive closure of $\twoheadrightarrow$.

Now, let us focus on the definition of an appropriate notion of observable. In our calculus, as in CCS [30] and in π-calculus [31], we have both transmission and reception of messages. However, in broadcast calculi only the transmission of messages may be observed [40, 29]. In fact, an observer cannot detect whether a given node actually receives a broadcast value. In particular, if the node $m[!\langle v\rangle.P]_t^\nu$ evolves into $m[\langle v\rangle^r.P]_t^\nu$ we do not know whether some of the neighbours have actually synchronised for receiving the message v. On the other hand, if a *non-exposed* node $n[\lfloor ?(x).P \rfloor Q]_0^\nu$ evolves into $n[(x)_v.P]_t^\nu$, then we can be sure that some node in ν has started transmitting. Notice that a node n can certify the reception of a message v only if it receives the whole message without collisions.

Following Milner and Sangiorgi [32] we use the term "barb" as synonymous of observable.

Definition 5.1 (Barbs) *Let M be a well-formed network. We write $M \downarrow_n$, if $M \equiv N \mid m[\langle v\rangle^r.P]_t^\nu$, for some m, v, r, P, t and ν, such that $n \in \nu$ and $n \notin \mathsf{nds}(N)$. We write $M \Downarrow_n$ if there is M' such that $M \twoheadrightarrow^* M' \downarrow_n$.*

The barb $M \Downarrow_n$ says that there is an ongoing transmission at M reaching the node n of the environment. The observer can easily detect such a transmission placing a receiver with timeout at n of the form $n[\lfloor ?(x).\mathbf{0} \rfloor !\langle w\rangle.\mathbf{0}]_t^\nu$ where the system $M \mid n[\lfloor ?(x).\mathbf{0} \rfloor !\langle w\rangle.\mathbf{0}]_t^\nu$ is well-formed, and $f \in \nu$, for some fresh node f. In this manner, if n is currently exposed to a transmission then, after a σ-action, the fresh barb at f is definitely lost. One may wonder whether the barb should mention the name m of the transmitter. Notice that, in general, due to communication collisions, the observer may receive incomprehensible packets without being able to identify the transmitter. In fact, if $M \downarrow_n$ there might be several nodes in M which are currently transmitting to n. So, in our setting, it does not make sense to put the name of the transmitter in the barb.

Now, everything is in place to define our timed notion of barbed congruence. In the sequel, we write $\mathcal{R}$ to denote binary relations over well-formed networks.

Definition 5.2 (Barb preserving) *A relation $\mathcal{R}$ is said to be* barb preserving *if whenever $M\ \mathcal{R}\ N$ it holds that $M \downarrow_n$ implies $N \Downarrow_n$.*

Definition 5.3 (Reduction closure) *A relation $\mathcal{R}$ is said to be* reduction-closed *if $M\ \mathcal{R}\ N$ and $M \twoheadrightarrow M'$ imply there is N' such that $N \twoheadrightarrow^* N'$ and $M'\ \mathcal{R}\ N'$.*

As we are interested in weak behavioural equivalences, the definition of reduction closure is given in terms of weak reductions.

Definition 5.4 (σ-closure) *A relation $\mathcal{R}$ is said to be* σ-closed *if $M\ \mathcal{R}\ N$ and $M \xrightarrow{\sigma} M'$ imply there is a network N' such that $N \twoheadrightarrow^* \xrightarrow{\sigma} \twoheadrightarrow^* N'$ and $M'\ \mathcal{R}\ N'$.*

When comparing two networks M and N, time must pass in the same manner for M and N.

Definition 5.5 (Contextuality) *A relation $\mathcal{R}$ is said* contextual *if $M\ \mathcal{R}\ N$, for M and N well-formed, implies $M \mid O\ \mathcal{R}\ N \mid O$ for all networks O such that $M \mid O$ and $N \mid O$ are well-formed.*

Finally, everything is in place to define timed reduction barbed congruence.

Definition 5.6 (Timed reduction barbed congruence) Timed reduction barbed congruence*, written $\cong$, is the largest symmetric relation over well-formed networks which is barb preserving, reduction-closed, σ-closed and contextual.*

6 A bisimulation proof method

The definition of timed reduction barbed congruence is simple and intuitive. However, due to the universal quantification on parallel contexts, it may be quite difficult to prove that two terms are barbed congruent. Simpler proof techniques are based on labelled bisimilarities. In this section, we propose an appropriate notion of bisimulation between networks. As a main result, we prove that our labelled bisimilarity is a proof-technique for timed reduction barbed congruence.

First of all we have to distinguish between transmissions which may be observed and transmissions which may not be observed by the environment. Thus, we extend the set of labelled transitions with the following two rules:

$$\text{(Shh)}\ \frac{M \xrightarrow{m!v} N \quad \mathsf{ngh}(m,M)\subseteq\mathsf{nds}(M)}{M \xrightarrow{\tau} N} \qquad \text{(Out)}\ \frac{M \xrightarrow{m!v} N \quad \nu:=\mathsf{ngh}(m,M)\setminus\mathsf{nds}(M)\neq\emptyset}{M \xrightarrow{!v\triangleright\nu} N}$$

Rule (Shh) models transmissions that cannot be detected by the environment. This happens if none of the potential receivers is in the environment. Rule (Out) models a transmission of a message that may be potentially received by the nodes ν of the environment. Notice that this transmission can be really observed at some node $n \in \nu$ only if no collisions arise at n during the transmission of v. In rule (Out) the name of the transmitter is removed from the action. This is motivated by the fact that nodes may refuse to reveal their identities, e.g., for security reasons, or limited sensory capabilities in perceiving these identities. Actually, in a hostile scenario the identity of the transmitter can only be ensured by using appropriate authentication protocols.

In the sequel, we use the metavariable α to range over the following actions: τ, σ, $m?v$ and $!v\triangleright\nu$. Since we are interested in *weak behavioural equivalences*, that abstract over τ-actions, we introduce a standard notion of weak action: $\Rightarrow$ denotes the reflexive and transitive closure of $\stackrel{\tau}{\longrightarrow}$; $\stackrel{\alpha}{\Longrightarrow}$ denotes $\Rightarrow\stackrel{\alpha}{\longrightarrow}\Rightarrow$; $\stackrel{\hat{\alpha}}{\Longrightarrow}$ denotes $\Rightarrow$ if $\alpha=\tau$ and $\stackrel{\alpha}{\Longrightarrow}$ otherwise.

Definition 6.1 (Bisimilarity) *A relation $\mathcal{R}$ over well-formed networks is a* simulation *if $M\ \mathcal{R}\ N$ implies that whenever $M\stackrel{\alpha}{\longrightarrow}M'$ there is N' such that $N\stackrel{\hat{\alpha}}{\Longrightarrow}N'$ and $M'\ \mathcal{R}\ N'$. A relation $\mathcal{R}$ is called* bisimulation *if both $\mathcal{R}$ and its converse are simulations. We say that M and N are* bisimilar*, written $M\approx N$, if there is some bisimulation $\mathcal{R}$ such that $M\ \mathcal{R}\ N$.*

It is worth noticing that whenever two networks are bisimilar then they must have the same set of nodes.

Proposition 6.2 *If $M\approx N$ then $\mathsf{nds}(M)=\mathsf{nds}(M)$.*

Proof By contradiction. Suppose there is a node m such that $m\in\mathsf{nds}(M)$ and $m\notin\mathsf{nds}(N)$. Then, by Proposition 2.4(1) (implication left to right) there is N' such that $N\stackrel{m?v}{\longrightarrow}N'$. Since $M\approx N$ there must be M' such that $M\stackrel{m?v}{\Longrightarrow}M'$ with $M'\approx N'$. However, since $m\in\mathsf{nds}(M)$, by Proposition 2.4(1) (implication right to left) there is no way to deduce a weak transition of the form $M\stackrel{m?v}{\Longrightarrow}M'$, as a node $m[W]^{\nu}_{t}$ can not perform an action $m?v$. □

In order to prove that our labelled bisimilarity implies timed reduction barbed congruence we have to show its contextuality.

Theorem 6.3 ($\approx$ is contextual) *Let M and N be two well-formed networks such that $M\approx N$. Then $M\mid O\approx N\mid O$ for all networks O such that $M\mid O$ and $N\mid O$ are well-formed.*

Proof See the Appendix. □

Theorem 6.4 (Soundness) *Let M and N be two well-formed networks such that $M\approx N$. Then $M\cong N$.*

Proof We have to prove that the labelled bisimilarity is contextual, barb preserving, reduction- and σ-closed. Contextuality follows from Theorem 6.3. Reduction and σ-closure follow by definition. As to barb preservation we reason by contradiction, if $M\downarrow_n$ we can choose $O\stackrel{\text{def}}{=}n[\lfloor?(x).\mathbf{0}\rfloor!\langle w\rangle.\mathbf{0}]^{\nu}_{t}$ such that $M\mid O$ and $N\mid O$ are well-formed, and $f\in\nu$, for some fresh name f. Since $M\downarrow_n$ the network $M\mid O$ will never (even in the future) perform an output action $!w\triangleright\nu$. On the other hand, if $N\not\downarrow_n$ by Theorem 3.3 we would have $N\mid O\rightarrowtriangle^{*}\stackrel{\sigma}{\longrightarrow}N'$,

for some network N'. This implies $N \mid O \rightarrow^* \xrightarrow{\sigma} \rightarrow^* \xrightarrow{!w \triangleright \nu}$. However, by Theorem 6.3 it follows that $M \mid O \approx N \mid O$. So, it must be $N \Downarrow_n$. □

In Theorem 6.5, we report a number of algebraic laws on well-formed networks that can be proved using our bisimulation proof-technique. The first and the second law show different but equivalent nodes that do not interact with the rest of the network. The third law is about exposed sleeping nodes. The fourth law is about successful reception. Here, node n will receive correctly because all its neighbours will not interfere during the current transmission. The fifth and the sixth law are about collisions: in both cases the transmission at m will cause a collision at n. The seventh law tells about the blindness of receivers exposed to collisions. In particular, if all neighbours of a transmitter are exposed, then the content of the transmission is irrelevant as all recipients will fail. Only the duration of the transmission may be important as the exposure indicators of the neighbours may change.

Theorem 6.5 *All networks below are assumed to be well-formed.*

1. $n[\mathsf{nil}]_t^\nu \approx n[Sleep]_t^\nu$, *where* $Sleep \stackrel{\text{def}}{=} \sigma.Sleep$.
2. $n[\mathsf{nil}]_t^\nu \approx n[P]_t^\nu$, *if* P *does not contain sender processes.*
3. $n[\sigma^r.P]_s^\nu \approx n[\sigma^r.P]_t^\nu$ *if* $s \leq r$ *and* $t \leq r$.
4. $m[\langle v \rangle^r.P]_t^\nu \mid n[(x)_v.Q]_r^{\nu'} \mid M \approx m[\langle v \rangle^r.P]_t^\nu \mid n[\sigma^r.\{^v/_x\}Q]_r^{\nu'} \mid M$, *if* $m \in \nu' \subseteq \mathsf{nds}(M)$ *and whenever* $M \stackrel{A}{\Longrightarrow} \xrightarrow{!w \triangleright \mu}$, *with* $n \in \mu$, *then* $\#^\sigma(A) \geq r{+}1$.
5. $m[!\langle v \rangle.P]_s^\nu \mid n[(x)_w.Q]_t^{\nu'} \approx m[!\langle v \rangle.P]_s^\nu \mid n[(x)_\perp.Q]_t^{\nu'}$, *if* $m \in \nu'$.
6. $m[\langle v_1 \rangle^r.!\langle v_2 \rangle.P]_s^\nu \mid n[(x)_w.Q]_t^{\nu'} \approx m[\langle v_1 \rangle^r.!\langle v_2 \rangle.P]_s^\nu \mid n[(x)_\perp.Q]_t^{\nu'}$, *if* $m \in \nu'$.
7. $m[!\langle v \rangle.P]_t^\nu \mid N \approx m[!\langle w \rangle.P]_t^\nu \mid N$, *if* $\delta_v = \delta_w$, *and for all* $n \in \nu$ *it holds that* $N \equiv n[W]_{t'}^{\nu'} \mid N'$, *with* $t' > 0$.

Proof By exhibiting the appropriate bisimulations. Let us prove, for instance, Laws 5 and 7. Let us start with Law 5. For convenience, let us define:

- $A \stackrel{\text{def}}{=} m[!\langle v \rangle.P]_s^\nu \mid n[(x)_w.Q]_t^{\nu'}$
- $B \stackrel{\text{def}}{=} m[!\langle v \rangle.P]_s^\nu \mid n[(x)_\perp.Q]_t^{\nu'}$.

Let

$$\mathcal{S} \stackrel{\text{def}}{=} \{(A, B) \mid \text{ for all } s \text{ and } t\} \cup Id$$

where Id is the identity relation over network terms. We prove that $\mathcal{S}$ is a bisimulation up to $\equiv$. We proceed by case analysis on the possible transitions of A. Notice that by maximal progress, no σ-actions may be performed.

- If $A \xrightarrow{h?v'} A'$. The most interesting case is when $h \in \nu \cap \nu'$. In this case, by an application of rules (Coll), (Exp) and (RcvPar) we have $A' = m[!\langle v\rangle.P]^{\nu}_{s'} \mid n[(x)_{\perp}.Q]^{\nu'}_{t'}$, where $t' = \max(t, \delta_{v'})$ and $s' = \max(s, \delta_{v'})$. Similarly, we have $B \xrightarrow{h?v'} A'$ and we are done.
- If $A \xrightarrow{!v\triangleright\hat{\nu}} A'$, with $\hat{\nu} = \nu \setminus \{n\} \neq \emptyset$, then since $m \in \nu'$, by an application of rules (Snd), (Coll), (Sync) and (Out) it follows that $A' = m[\langle v\rangle^{\delta_v}.P]^{\nu}_{s} \mid n[(x)_{\perp}.Q]^{\nu'}_{t'}$ with $t' = \max(t, \delta_v)$. Similarly, we have $B \xrightarrow{!v\triangleright\hat{\nu}} A'$ and we are done.
- If $A \xrightarrow{\tau} A'$, because $A \xrightarrow{m!v} A'$ and $\nu = \{n\}$. This case is similar to the previous one.

As regards the proof of Law 7, let us define:

- $A_1 \stackrel{\text{def}}{=} m[!\langle v\rangle.P]^{\nu}_{t} \mid N$, where for all $n \in \nu$ it holds that $N \equiv n[W]^{\nu'}_{t'} \mid N'$, with $t' > 0$
- $B_1 \stackrel{\text{def}}{=} m[!\langle w\rangle.P]^{\nu}_{t} \mid N$, where for all $n \in \nu$ it holds that $N \equiv n[W]^{\nu'}_{t'} \mid N'$, with $t' > 0$
- $A_2 \stackrel{\text{def}}{=} m[\langle v\rangle^{r}.P]^{\nu}_{t} \mid N$, with $r \leq \delta_v$, where for all $n \in \nu$ it holds that $N \equiv n[W]^{\nu'}_{t'} \mid N'$, with $t' \geq r$
- $B_2 \stackrel{\text{def}}{=} m[\langle v\rangle^{r}.P]^{\nu}_{t} \mid N$, with $r \leq \delta_w$, where for all $n \in \nu$ it holds that $N \equiv n[W]^{\nu'}_{t'} \mid N'$, with $t' \geq r$

where $\delta_v = \delta_w$. Now, let

$$\mathcal{S} \stackrel{\text{def}}{=} \{(A_1, B_1) : \text{ for all } P, t, \nu, \ldots\} \cup \{(A_2, B_2) : \text{ for all } P, t, \nu, \ldots)\} \cup Id$$

where Id is the identity relation between network terms. We prove that $\mathcal{S}$ is a bisimulation. We proceed by case analysis on the possible transitions.

- Let us examine the most interesting transitions of A_1. The reasoning for the other transitions of A_1 is simpler. Notice that by maximal progress the term A_1 cannot perform σ-actions.

– Let $A_1 \xrightarrow{\tau} A_2 = m[\langle v\rangle^{\delta_v}.P]^{\nu}_t \mid \hat{N}$, because $A_1 \xrightarrow{m!v} A_2$ by an application of rule (Shh). This is possible only by an application of rule (Sync) with

  * $m[!\langle v\rangle.P]^{\nu}_t \xrightarrow{m!v} m[\langle v\rangle^{\delta_v}.P]^{\nu}_t$
  * $N \xrightarrow{m?v} \hat{N}$, where if $n \in \nu$ then $\hat{N} \equiv n[\hat{W}]^{\nu'}_{\hat{t}} \mid \hat{N}'$, with $\hat{t} \geq \delta_v$ (by definition of rules (Coll) and (Exp)).

  Notice that since all nodes in $\nu \cap \mathsf{nds}(N)$ are exposed, it follows that if $\hat{W}$ is an active receiver then it will be of the form $(x)_{\perp}.P$, for some P. Now, $A_2 \xrightarrow{\tau} B_2 = m[\langle w\rangle^{\delta_w}.P]^{\nu}_t \mid \hat{\hat{N}}$, because $A_2 \xrightarrow{m!v} B_2$ by an application of rule (Shh). This is possible only by an application of rule (Sync) with

  * $m[!\langle w\rangle.P]^{\nu}_t \xrightarrow{m!w} m[\langle w\rangle^{\delta_w}.P]^{\nu}_t$
  * $N \xrightarrow{m?w} \hat{\hat{N}}$, where if $n \in \nu$ then $\hat{\hat{N}} \equiv n[\hat{\hat{W}}]^{\nu'}_{\hat{\hat{t}}} \mid \hat{\hat{N}}'$, with $\hat{\hat{t}} \geq \delta_v$ (by definition of rules (Coll) and (Exp)).

  Again, since all nodes in $\nu \cap \mathsf{nds}(N)$ are exposed, it follows that if $\hat{\hat{W}}$ is an active receiver then it will be of the form $(x)_{\perp}.P$, for some P. Moreover, since $\delta_v = \delta_w$ it follows $\hat{t} = \hat{\hat{t}}$. As a consequence, $\hat{N} = \hat{\hat{N}}$ and $(A_2, B_2) \in \mathcal{S}$.

- Let us examine the most interesting transitions of A_2. The reasoning for the other transitions is simpler.

  – Let $A_2 \xrightarrow{\sigma} A'_2 = m[\langle v\rangle^{r-1}.P]^{\nu}_{t-1} \mid \hat{N}$ by an application of rule (Par-σ) because

    * $m[\langle v\rangle^{r}.P]^{\nu}_t \xrightarrow{\sigma} m[\langle v\rangle^{r-1}.P]^{\nu}_{t-1}$
    * $N \xrightarrow{\sigma} \hat{N}$.

    In this case we have $B_2 \xrightarrow{\sigma} B'_2 = m[\langle w\rangle^{r-1}.P]^{\nu}_{t-1} \mid \hat{N}$. Now, independent of whether $r > 1$ or not we have $(A'_2, B'_2) \in \mathcal{S}$.

□

7 Conclusions, future and relate work

We have proposed a broadcasting timed process calculus for wireless networks with time-consuming transmissions. We have equipped our calculus with a formal operational semantics which has been used to formally analyse communication collisions. We have used an extended version of TCWS to describe a

number of protocols at different levels of abstraction. Then, we have developed a bisimulation-based semantic theory which has been used to prove a number of algebraic laws.

In TCWS we have modelled wireless networks with a unique channel. However, new techniques have been developed in the last years to provide several virtual channels. The most known techniques are *Frequency Division Multiplexing* (FDM), which involves assigning non-overlapping frequency ranges to different signals, and *Time Division Multiplexing* (TDM), in which the time domain is divided into several recurrent time slots of fixed length, one for each sub-channel.

Frequency Division Multiplexing, where a channel is divided into several non-interfering sub-channels, can be easily implemented in a generalisation of TCWS with multiple channels (à la CCS).

Time Division Multiplexing can be represented in TCWS as well. For instance, *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. TDMA is easy to model in our TCWS calculus. To have an idea let us consider a simple system in which the channel is divided in two sub-channels taking Δ time units each:

$$Sys \stackrel{\text{def}}{=} m_1[P]_0^{\nu_{m_1}} \mid n_1[R]_0^{\nu_{n_1}} \mid m_2[\sigma^{\Delta}.P]_0^{\nu_{m_2}} \mid n_2[\sigma^{\Delta}.R]_0^{\nu_{m_1}}$$

where $P \stackrel{\text{def}}{=} \ldots.\sigma^{\Delta}.P$ and $R \stackrel{\text{def}}{=} \ldots.\sigma^{\Delta}.R$ are two synchronised processes. Intuitively, the pair of nodes m_1 and n_1 use the channel in odd time slots, while m_2 and n_2 use the channel in even time slots.

In TCWS we have assumed that all nodes have the same transmission range; this is a quite common assumption in models for ad hoc networks [33] and actually it is required in some MAC-layer protocols such as CSMA/CA (see Section 4.1.1 for details).

As in Lanese and Sangiorgi's CWS [25], our calculus does not deal with node mobility, i.e. nodes are assumed to be immobile. This is mainly for two reasons. First, as noticed in [25] node mobility is an orthogonal issue, which does not affect the formulation of our semantics and the treatment of interference (the main topic of this paper). Second, movement is not relevant in important classes of wireless systems, most notably *sensor networks* [2] (not all sensor networks are stationary, but the stationary case is predominant). Nevertheless, it is possible

to adopt in our calculus some techniques developed in [14, 15] to allow disciplined forms of mobility, where neighbouring relations may change provided that network connectivity is maintained.

In Section 4.1, we have seen that the CSMA scheme (even in its p-persistent form) suffers of several problems such as the exposed terminal problem and the hidden terminal problem. Clearly, in a broadcast environment where there is no direct way to infer the loss of information owing to collisions, it is important to indirectly and accurately determine the *probability* of packet collisions. We believe that our calculus represents a solid basis to develop a probabilistic calculus where transmitters start transmitting with a certain probability p (independently whether the channel is free) and with probability $1 - p$ waits before transmitting. The goal would be that of developing verification techniques such as *probabilistic model checking* [24] to guarantee the absence of collisions with a certain probability. This will be one of the directions of our research.

Last but not least, we believe that our timed calculus can be used as a basis to develop *trust models* for wireless systems. Trust establishment in ad hoc networks is an open and challenging field. In fact, without a centralised trusted authority it is not obvious how to build and maintain trust. Nevertheless, the notion of time seems to be important to represent credentials' expiration.

Let us examine now the most relevant related works.

We start with the literature on process calculi for wireless systems. Nanz and Hankin [36] have introduced a calculus for Mobile Wireless Networks (CBS$^\sharp$), relying on graph representation of node localities. The main goal of the paper is to present a framework for specification and security analysis of communication protocols for mobile wireless networks. Merro [29] has proposed a process calculus for Mobile Ad Hoc Networks with a labelled characterisation of reduction barbed congruence. Godskesen [16] has proposed a calculus for mobile ad hoc networks (CMAN). The paper proves a characterisation of reduction barbed congruence 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. Ghassemi et al. [14] have proposed a process algebra for mobile ad hoc networks (RBPT) where, topology changes are implicitly modelled in the (operational) semantics rather than in the syntax. The authors propose a notion of bisimulation for networks parameterised on a set of topology invariants that must be respected by equivalent networks. This work in then refined in [15] where the

authors propose an equational theory for an extension of RBPT. All the previous calculi abstract from the presence of interferences. Lanese and Sangiorgi [25] have instead proposed the CWS calculus, a lower level untimed calculus to describe interferences in wireless systems. In their LTS there is a separation between transmission beginning and transmission ending. Our work is definitely inspired by [25]. More recently, Godskesen and Nanz [18] have proposed a simple timed calculus for wireless systems to express a wide range of mobility models.

None of the calculi mentioned above, except for [18], deals with time, although there is an extensive literature on timed process algebra. From a purely syntactic point of view, the earliest proposals are extensions of the three main process algebras, ACP, CSP and CCS. For example, [4] presents a real-time extension of ACP, [41] contains a denotational model for a timed extension of CSP, while CCS is the starting point for [35]. In [4] and [41] time is real-valued, and at least semantically, associated directly with actions. The other major approach to representing time is to introduce special actions to model the passage of time, which the current paper shares with [19, 7, 35, 37] and [47, 48], although the basis for all those proposals may be found in [9]. The current paper shares many of the assumptions of the languages presented in these papers. For example, all the papers above assume that actions are instantaneous and only the extension of ACP presented in [19] does not incorporate time determinism; however maximal progress is less popular and patience is even rarer.

More recent works on timed process algebra include the following papers. Aceto and Hennessy [1] have presented a simple process algebra where time emerges in the definition of a *timed observational equivalence*, assuming that beginning and termination of actions are distinct events which can be observed. Hennessy and Regan [20] have proposed a timed version of CCS enjoying time determinism, maximal progress and patience. Our action σ takes inspiration from theirs. The authors have developed a semantic theory based on testing and characterised in terms of a particular kind of ready traces. Prasad [39] has proposed a timed variant of his CBS [38], called TCBS. In TCBS a time out 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. [11] 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 [12] 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 [5] have proposed several timed process algebras treated

in a common framework, and 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 [6] Baeten and Reniers extend the framework of [5] to model successful termination for the relative-time case. Laneve and Zavattaro [26] 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.

Acknowledgements The referees have provided many useful suggestions. We thank Sebastian Nanz for a preliminary discussion on timed calculi for wireless networks. Davide Quaglia for insightful discussions on the IEEE 802.11 standard. Many thanks to Matthew Hennessy for his precious comments on an early draft of the paper. We thank Andrea Cerone for suggesting us to use receivers with timeout to model the CSMA protocol.

References

[1] L. Aceto and M. Hennessy. Towards action-refinement in process algebras. *Information and Computation*, 103(2):204–269, 1993.

[2] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4), 2002.

[3] ZigBee Alliance. Zigbee specification, version 1.0. Technical Report 053474r06, ZigBee Alliance, 2005.

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

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

[6] 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.

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

[8] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Computation*, 60:109–137, 1984.

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

[10] L. Cardelli and A. Gordon. Mobile ambients. *Theoretical Computer Science*, 240(1):177–213, 2000.

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

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

[13] S. Ganeriwal, R. Kumar, and M. Srivastava. Timing-Sync Protocol for Sensor Networks. In *SenSys*, pages 138–149. ACM Press, 2003.

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

[15] F. Ghassemi, W. Fokkink, and A. Movaghar. Equational Reasoning on Ad Hoc networks. In *FSEN*, To appear in Lecture Notes in Computer Science. Springer, 2009.

[16] 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.

[17] J.C. Godskesen. A Calculus for Mobile Ad-hoc Networks with Static Location Binding. To appear in the Prooceedings of EXPRESS, 2008.

[18] 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.

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

[20] M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 117(2):221–239, 1995.

[21] M. Hennessy and J. Riely. A typed language for distributed mobile processes. In *Proc. 25th POPL*. ACM Press, 1998.

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

[23] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of SenSys'04*, pages 162–175. ACM, 2004.

[24] M. Kwiatkowska, G. Norman, and D. Parker. Prism: Probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.

[25] Ivan Lanese and Davide Sangiorgi. An operational semantics for a calculus for wireless systems. *Theoretical Computer Science*, 411(19):1928–1948, 2010.

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

[27] Q. Li and D. Rus. Global Clock Synchronization in Sensor Networks. *IEEE Transactions on Computers*, 55(2):214–226, 2006.

[28] M. Luk, G. Mezzour, A. Perrig, and V.D. Gligor. Minisec: a secure sensor network communication architecture. In *IPSN*, pages 479–488, 2007.

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

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

[31] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, (Parts I and II). *Information and Computation*, 100:1–77, 1992.

[32] R. Milner and D. Sangiorgi. Barbed bisimulation. In *ICALP*, volume 623 of *Lecture Notes in Computer Science*, pages 685–695. Springer Verlag, 1992.

[33] S Misra and I Woungag. *Guide to Wireless Ad Hoc Networks*. Computer Communications and Networks. Springer London, 2009.

[34] M. Mock, R. Frings, E. Nett, and S. Trikaliotis. Continuous Clock Synchronization in Wireless Real-Time Applications. In *SRDS*, pages 125–133. IEEE Computer Society, 2000.

[35] 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.

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

[37] X. Nicollin and J. Sifakis. The Algebra of Timed Processes, ATP: Theory and Application. *Information and Computation*, 114(1):131–178, 1994.

[38] K.V.S. Prasad. A Calculus of Broadcasting Systems. *Science of Computer Programming*, 25(2-3), 1995.

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

[40] J. Rathke, V. Sassone, and P. Sobocinski. Semantic Barbs and Biorthogonality. In *FoSSaCS*, volume 4423 of *Lecture Notes in Computer Science*, pages 302–316. Springer, 2007.

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

[42] Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In *ACM Conference on Computer and Communications Security*, pages 196–205, 2001.

[43] M. L. Sichitiu and C. Veerarittiphan. Simple, Accurate Time Synchronization for Wireless Sensor Networks. In *WCNC*, pages 1266–1273. IEEE Computer Society, 2003.

[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] W. Su and I. Akyildiz. Time-Diffusion Synchronization Protocols for Sensor Networks. *IEEE/ACM Transactions on Networking*, 13(2):384–397, 2005.

[46] B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

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

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

[49] S. Yoon, C. Veerarittiphan, and M. L. Sichitiu. Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Transactions on Sensor Networks*, 3(2):81–118, 2007.

A Proofs

Proof of Proposition 2.4

Let us prove the single items of the proposition.

1. Let us prove first the implication from left to right. *If* $m \notin \mathsf{nds}(M)$ *then* $M \xrightarrow{m?v} M'$*, for some network* M'.

Let us proceed by induction on the structure of M.

- Let $M = \mathbf{0}$. By an application of rule (Zero) we have $M \xrightarrow{m?v} M$.
- Let $M = n[W]^{\nu}_{t}$. Let us proceed by induction on the structure of W.
  - Let $W = \mathsf{nil}$. There are two cases.
    - If $m \notin \nu$ then by an application of rule (OutRng) we have $M \xrightarrow{m?v} M$.
    - If $m \in \nu$ then by an application of rule (Exp) we have $M \xrightarrow{m?v} M'$ with $M' = n[\mathsf{nil}]^{\nu}_{t'}$, where $t' = \max(t, \delta_v)$.
  - Let $W = !\langle v \rangle.P$. This case is similar to the previous one.
  - $W = \sigma.P$. This case is similar to the previous one.
  - $W = \langle v \rangle^{r}.P$. This case is similar to the previous one.
  - Let $W = \lfloor \tau.P \rfloor Q$. This case is similar to the previous one.
  - Let $W = \lfloor ?(x).P \rfloor Q$. There are three sub-cases.

* If $m \notin \nu$ then by an application of rule (OutRng) we have $M \xrightarrow{m?v} M$.
* If $m \in \nu$ and $t = 0$ then there are two possibilities:
  - by an application of rule (Rcv) we can derive $M \xrightarrow{m?v} M'$, with $M' = n[(x)_v.P]^{\nu}_{\delta_v}$;
  - by an application of rule (Exp) we can derive $M \xrightarrow{m?v} M'$, with $M' = n[\lfloor ?(x).P \rfloor Q]^{\nu}_{\delta_v}$.
* If $m \in \nu$ and $t > 0$ then by an application of rule (Exp) we have $M \xrightarrow{m?v} M'$, with $M' = n[\lfloor ?(x).P \rfloor Q]^{\nu}_{t'}$ and $t' = \max(t, \delta_v)$.

– Let $W = (x)_w.P$. There are two sub-cases.
* If $m \in \nu$ then by an application of rule (Coll), it holds that $M \xrightarrow{m?v} M' = n[(x)_{\perp}.P]^{\nu}_{t'}$ with $t' := \mathsf{max}(t,\delta_v)$.
* If $m \notin \nu$ then by an application of rule (OutRng) we have $M \xrightarrow{m?v} M$.

– Let $W = [v = v]P_1, P_2$. By an application of rule (Then) we can apply the inductive hypothesis to conclude that the statement holds.

– Let $W = [v_1 = v_2]P_1, P_2$, with $v_1 \neq v_2$. By an application of rule (Else), this case is similar to the previous one.

– Let $W = H\langle \tilde{v} \rangle$. The constraint on guarded recursion ensures us that by an application of rule (Rec) we can apply the inductive hypothesis to conclude that the statement holds.

- Let $M = M_1 \mid M_2$. By inductive hypothesis it holds that $M_1 \xrightarrow{m?v} M_1'$ and $M_2 \xrightarrow{m?v} M_2'$, for some M_1', M_2'. By an application of rule (RcvPar) it holds that $M \xrightarrow{m?v} M'$, for $M' = M_1' \mid M_2'$.

The implication from right to left says the following: if $M \xrightarrow{m?v} M'$ for some M', then $m \notin \mathsf{nds}(M)$. The proof is by straightforward rule induction.

2. Let us consider first the implication from left to right. *If* $M_1 \mid M_2 \xrightarrow{m?v} N$ *then there are* N_1 *and* N_2 *such that* $M_1 \xrightarrow{m?v} N_1$, $M_2 \xrightarrow{m?v} N_2$ *and* $N = N_1 \mid N_2$. Here, the proof follows by noticing that the only rule for deriving the action $m?v$ from $M_1 \mid M_2$ is (RcvPar). In its premises this rule requires exactly that the two parallel components M_1 and M_2 must perform an action $m?v$. The other implication is an easy application of rule (RcvPar).

3. If $M \xrightarrow{m!v} M'$ then $M \equiv m[!\langle v\rangle.P]^{\nu}_{t} \mid N$, for some ν, t, P and N, and there is N' such that $m[!\langle v\rangle.P]^{\nu}_{t} \xrightarrow{m!v} m[\langle v\rangle^{\delta_v}.P]^{\nu}_{t}$, $N \xrightarrow{m?v} N'$ and $M' \equiv m[\langle v\rangle^{\delta_v}.P]^{\nu}_{t} \mid N'$. The proof of this result follows by a straightforward induction on why $M \xrightarrow{m!v} M'$.

4. If $M \xrightarrow{\tau} M'$ then $M \equiv m[\lfloor \tau.P \rfloor Q]^{\nu}_{t} \mid N$, for some m, ν, t, P, Q and N such that $m[\lfloor \tau.P \rfloor Q]^{\nu}_{t} \xrightarrow{\tau} m[P]^{\nu}_{t}$ and $M' \equiv m[P]^{\nu}_{t} \mid N$. Again, the proof is by a straightforward transition induction.

5. Let us consider first the implication from left to right. *If* $M_1 \mid M_2 \xrightarrow{\sigma} N$ *then there are* N_1 *and* N_2 *such that* $M_1 \xrightarrow{\sigma} N_1$, $M_2 \xrightarrow{\sigma} N_2$ *and* $N = N_1 \mid N_2$. Here, the proof follows by noticing that the only rule for deriving the action σ from $M_1 \mid M_2$ is (σ-Par). In its premises this rule requires exactly that the two parallel components M_1 and M_2 must perform an action σ. The other implication is an easy application of rule (σ-Par). □

Now, we prove that our operational semantics preserves network well-formedness.

Proposition A.1 *Let M be a node-unique network. If $M \xrightarrow{\lambda} M'$ then M' is node-unique.*

Proof By transition induction. □

Proposition A.2 *Let M be a connected network. If $M \xrightarrow{\lambda} M'$ then M' is connected.*

Proof By transition induction. Notice that no inference rule changes the network topology. □

Next, we prove that our labelled transition semantics preserves exposure consistency. For that we need the two following technical lemmas.

Lemma A.3 *Let $M \xrightarrow{\lambda} M'$ with $\lambda \in \{m!v, m?v\}$ such that $M \equiv \prod_{i\in} n_i[W_i]^{\nu_i}_{t_i}$ and $M' \equiv \prod_{i\in I} n_i[W'_i]^{\nu_i}_{t'_i}$.*

1. *If $\lambda = m?v$ then $m \neq n_i$, for all i.*
2. *If $\lambda = m!v$ then there is $i \in I$ such that $m = n_i$, $W_i = !\langle v\rangle.P_i$ and $W'_i = \langle v\rangle^{\delta_v}.P_i$.*
3. *If $m \notin \nu_i$, for some i, then $t'_i = t_i$; if also $m \neq n_i$, then $W'_i = W_i$.*
4. *If $m \in \nu_i$, for some i, then $t'_i = \max(t_i, \delta_v)$.*

5. *If* $m \in \nu_i$ *and* $W_i'=(x)_w.P_i$, *for some* i, *and* $w\neq\perp$, *then* $w = v$, $t_i = 0$, $t_i' = \delta_v$, *and* W_i *is not an active sender process,*

6. *If* $W_i = \langle w\rangle^r.P_i$, *for some* i, *then* $W_i' = W_i$.

7. *If* $m \neq n_i$ *and* $W_i' = \langle w\rangle^r.P_i$, *for some* i, *then* $W_i' = W_i$.

Proof By transition induction. □

Lemma A.4 *Let* $M \xrightarrow{\sigma} M'$ *such that* $M \equiv \prod_{i\in} n_i[W_i]^{\nu_i}_{t_i}$ *and* $M' \equiv \prod_{i\in I} n_i[W_i']^{\nu_i}_{t_i'}$.

1. *For all* i, $t_i' = t_i - 1$, *if* $t_i > 0$ *and* 0 *otherwise.*

2. *If* $W_i' = (x)_v.P$, *for some* i, *then*

   - *either* $W_i = W_i'$
   - *or* W_i *is not an active receiver and* $v = \perp$

3. *If* $W_i' = \langle w\rangle^r.P$, *for some* i, *then* $W_i = \langle w\rangle^{r+1}.P$.

Proof By transition induction. □

Now, we can prove the preservation of exposure consistency.

Proposition A.5 (Exposure consistency) *Let* M *be an exposure consistent network. If* $M \xrightarrow{\lambda} M'$ *then* M' *is exposure consistent.*

Proof The proof proceeds by transition induction on the derivation of $M \xrightarrow{\lambda} M'$, for $\lambda \in \{m!v, m?v, \sigma, \tau\}$. We show the most significant cases, derived by an application of rules (Sync), (RcvPar) and (σ-Par). The other cases are straightforward.

- Let $M \xrightarrow{m!v} M'$ by an application of rule (Sync) with $M = M_1 \mid M_2$, $M_1 \xrightarrow{m!v} M_1'$ and $M_2 \xrightarrow{m?v} M_2'$, and $M' = M_1' \mid M_2'$, where M_1' and M_2' are exposure consistent by inductive hypothesis. We have to prove that M' respects the clauses of Definition 2.7.

  - Clauses 1-2. In these cases the result follows directly by inductive hypothesis.
  - Clause 3. Let $M' \equiv \prod_i n_i[W_i']^{\nu_i}_{t_i'} \mid h[\langle v\rangle^r.P]^{\nu_h}_{t_h'} \mid n[W']^{\nu_n}_{t_n'}$, with $h \in \nu_n$. We have to prove that $r \leq t_n'$. We only consider the case when $h \in \mathsf{nds}(M_1)$ and $n \in \mathsf{nds}(M_2)$ (or vice versa). The other cases are easier. There are two possibilities.

* $h \neq m$. By Lemma A.3(7) we have

$$M \equiv \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid h[\langle v \rangle^r.P]^{\nu_h}_{t_h} \mid n[W]^{\nu_n}_{t_n}$$

for appropriate processes and tags. Now, if $m \in \nu_n$ by Lemma A.3(4) we have $t'_n = \max(t_n, \delta_v)$. As M is exposure consistent it holds that $r \leq t_n$ and hence also $r \leq t'_n$. On the other hand, if $m \notin \nu_n$ by an application of Lemma A.3(3) we have $t'_n = t_n$. As M is exposure consistent it follows that $r \leq t_n = t'_n$.

* $h = m$. By Lemma A.3(2) it follows that

$$M \equiv \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid h[!\langle v \rangle.P]^{\nu_h}_{t_h} \mid n[W]^{\nu_n}_{t_n}$$

for appropriate processes and tags, with $r = \delta_v$. Since $h \in \nu_n$, by Lemma A.3(4) we have $t'_n = \max(t_n, \delta_v)$. As a consequence, $r \leq t'_n$.

– Clause 4. Let

$$M \equiv N \mid n[W]^{\nu}_{t} = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[W]^{\nu}_{t}$$

and

$$M' \equiv N' \mid n[W']^{\nu}_{t'} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[W']^{\nu}_{t'}$$

with $t' > 0$ and $\mathsf{active}(k', N') \neq t'$ for all $k' \in \nu \cap \mathsf{actsnd}(N')$. We have to prove that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t'_i \geq t'$. We can distinguish two cases:

* If $m \notin \nu$ by Lemma A.3(3) we have $t' = t$. By Lemma A.3(6), it follows that $\mathsf{actsnd}(N) \subseteq \mathsf{actsnd}(N')$. As a consequence, $\nu \cap \mathsf{actsnd}(N) \subseteq \nu \cap \mathsf{actsnd}(N')$. Since $t' = t$ we can derive that for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds that $\mathsf{active}(k, N') \neq t$. By Lemma A.3(6) and Lemma A.3(7) if $k \neq m$ then $\mathsf{active}(k, N) = \mathsf{active}(k, N')$. Since $m \notin \nu$ it follows that for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds $\mathsf{active}(k, N) \neq t$. Since M is exposure consistent it follows that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N)$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i \geq t$. Notice that $\nu \setminus \mathsf{nds}(N) = \nu \setminus \mathsf{nds}(N')$. Moreover, by Lemma A.3(3) and A.3(4) we have $t_i \leq t'_i$, for all i. This allows us to derive that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t'_i \geq t_i \geq t = t'$.

* If $m \in \nu$ then by Lemma A.3(4) we have $t' = \max(t, \delta_v)$. By definition of neighbouring of a node $m \in \nu$ implies $m \neq n$. By Lemma A.3(2) it follows that $m \notin \mathsf{actsnd}(N)$, $m \in \mathsf{actsnd}(N')$ and $\mathsf{active}(m, N') = \delta_v$. Since $\mathsf{active}(k', N') \neq t'$ for all $k' \in \nu \cap \mathsf{actsnd}(N')$, and $m \in \nu \cap \mathsf{actsnd}(N')$, it follows that $t' \neq \delta_v$. Since $t' = \max(t, \delta_v)$, it follows that $t' = t$. By Lemma A.3(6), it follows that $\mathsf{actsnd}(N) \subseteq \mathsf{actsnd}(N')$. As a consequence, $\nu \cap \mathsf{actsnd}(N) \subseteq \nu \cap \mathsf{actsnd}(N')$. Since $t' = t$ we can derive that for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds that $\mathsf{active}(k, N') \neq t$. By Lemma A.3(6) and Lemma A.3(7) if $k \neq m$ then $\mathsf{active}(k, N) = \mathsf{active}(k, N')$. Since $m \notin \mathsf{actsnd}(N)$ it follows that for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds $\mathsf{active}(k, N) \neq t$. Since M is exposure consistent it follows that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N)$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i \geq t$. Notice that $\nu \setminus \mathsf{nds}(N) = \nu \setminus \mathsf{nds}(N')$. Moreover, by Lemmas A.3(3) and A.3(4) we have $t_i \leq t'_i$, for all i. This allows us to derive that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t'_i \geq t_i \geq t = t'$.

- Let $M \xrightarrow{m?v} M'$ by an application of rule (RcvPar) with $M = M_1 \mid M_2$, $M_1 \xrightarrow{m?v} M'_1$, $M_2 \xrightarrow{m?v} M'_2$, and $M' = M'_1 \mid M'_2$, where both M'_1 and M'_2 are exposure consistent by inductive hypothesis. We have to prove that M' respects the clauses of Definition 2.7.
  - Clauses 1-2. We reason as in the case of the the sending action $m!v$ examined above.
  - Clause 3. Let $M' \equiv \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid h[\langle v \rangle^r.P]^{\nu_h}_{t'_h} \mid n[W']^{\nu_n}_{t'_n}$, with $h \in \nu_n$. We have to prove that $r \leq t'_n$. We only consider the case when $h \in \mathsf{nds}(M_1)$ and $n \in \mathsf{nds}(M_2)$ (or vice versa). The other cases are easier. By Lemma A.3(1) it holds that $m \notin \mathsf{nds}(M')$. By A.3(7) it follows:

    $$M \equiv \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid h[\langle v \rangle^r.P]^{\nu_h}_{t_h} \mid n[W]^{\nu_n}_{t_n}$$

    for appropriate processes and tags. Now, if $m \in \nu_n$ by Lemma A.3(4) we have $t'_n = \max(t_n, \delta_v)$. As M is exposure consistent it holds that $r \leq t_n$ and hence also $r \leq t'_n$. On the other hand, if $m \notin \nu_n$ by an application of Lemma A.3(3) we have $t'_n = t_n$; as M is exposure consistent it follows that $r \leq t_n = t'_n$.
  - Clause 4. Let

    $$M \equiv N \mid n[W]^{\nu}_{t} = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[W]^{\nu}_{t}$$

and

$$M' \equiv N' \mid n[W']^{\nu}_{t'} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[W']^{\nu}_{t'}$$

with $t' > 0$ and $\mathsf{active}(k', N') \neq t'$ for all $k' \in \nu \cap \mathsf{actsnd}(N')$. We have to prove that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t'_i \geq t'$. There are two cases.

- Let $m \notin \nu$. By Lemma A.3(3) we have $t' = t$. By Lemma A.3(6), it follows that $\mathsf{actsnd}(N) \subseteq \mathsf{actsnd}(N')$. As a consequence, $\nu \cap \mathsf{actsnd}(N) \subseteq \nu \cap \mathsf{actsnd}(N')$. Since $t' = t$ we can derive that for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds that $\mathsf{active}(k, N') \neq t$. By Lemma A.3(6) and Lemma A.3(7) if $k \neq m$ then $\mathsf{active}(k, N) = \mathsf{active}(k, N')$. Since $m \notin \nu$ it follows that for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds $\mathsf{active}(k, N) \neq t$. Since M is exposure consistent it follows that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N)$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i \geq t$. Notice that $\nu \setminus \mathsf{nds}(N) = \nu \setminus \mathsf{nds}(N')$. Moreover, by Lemma A.3(3) and A.3(4) we have $t_i \leq t'_i$, for all i. This allows us to derive that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t'_i \geq t_i \geq t = t'$.
- Let $m \in \nu$. By Lemma A.3(1) we have $m \notin \mathsf{nds}(M)$. By Lemmas A.3(6) and A.3(7) for all $k \in \mathsf{nds}(N)$ it holds that $\mathsf{active}(k, N) = \mathsf{active}(K, N')$. As a consequence, $\mathsf{actsnd}(N) = \mathsf{actsnd}(N')$, and hence $\nu \cap \mathsf{actsnd}(N) = \nu \cap \mathsf{actsnd}(N')$. By Lemma A.3(4) we have $t' = \max(t, \delta_v)$. So, there are two cases.
  - Let $\delta_v \leq t$. Then $t' = t$ and for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds $\mathsf{active}(k, N) \neq t$. Since M is exposure consistent it follows that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N)$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i \geq t$. Notice that $\nu \setminus \mathsf{nds}(N) = \nu \setminus \mathsf{nds}(N')$. Moreover, by Lemmas A.3(3) and A.3(4) we have $t_i \leq t'_i$, for all i. This allows us to derive that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t'_i \geq t_i \geq t = t'$.
  - Let $\delta_v > t$. Then $t' = \delta_v$. In this case, there is $m \in \nu \setminus \mathsf{nds}(N')$ such that if $m \in \nu_i$, for some i, then by Lemma A.3(4) it holds that $t'_i = \max(t_i, \delta_v)$. Thus, $t'_i \geq \delta_v = t'$.

- Let $M \xrightarrow{\sigma} M'$ by an application of rule (σ-Par) with $M = M_1 \mid M_2$, $M_1 \xrightarrow{\sigma} M'_1$ and $M_2 \xrightarrow{\sigma} M'_2$, and $M' = M'_1 \mid M'_2$, where both M'_1 and M'_2 are exposure consistent by inductive hypothesis. We have to prove that M' respects the clauses of Definition 2.7.

– Clauses 1-2. It is easy to show that M' is exposure consistent. The results follow by inductive hypothesis.

– Clause 3. Let

$$M' \equiv \prod_i n_i[W_i']_{t_i'}^{\nu_i} \mid h[\langle v \rangle^r.P]_{t_h'}^{\nu_h} \mid n[W']_{t_n'}^{\nu_n}$$

with $h \in \nu_n$. We have to prove that $r \leq t_n'$. We only consider the case when $h \in \mathsf{nds}(M_1)$ and $n \in \mathsf{nds}(M_2)$ (or vice versa). The other cases are easier. By Lemma A.4(1) and A.4(3) we have

$$M \equiv \prod_i n_i[W_i]_{t_i}^{\nu_i} \mid h[\langle v \rangle^{r+1}.P]_{t_h}^{\nu_h} \mid n[W]_{t_n'+1}^{\nu_n}$$

for appropriate processes and tags. As M is exposure consistent, it follows that $r \leq t_n'$.

– Clause 4. Let

$$M \equiv N \mid n[W]_t^{\nu} = \prod_i n_i[W_i]_{t_i}^{\nu_i} \mid n[W]_t^{\nu}$$

and

$$M' \equiv N' \mid n[W']_{t'}^{\nu} = \prod_i n_i[W_i']_{t_i'}^{\nu_i} \mid n[W']_{t'}^{\nu}$$

with $t' > 0$ and $\mathsf{active}(k', N') \neq t'$ for all $k' \in \nu \cap \mathsf{actsnd}(N')$. We have to prove that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i' \geq t'$. By Lemma A.4(1) we have $t' = t - 1$. Since $t' > 0$ it follows that $t > 1$. Moreover, by Lemma A.4(3), if $W_i' = \langle w \rangle^{r'}.Q$, for some i, then $W_i = \langle w \rangle^r.Q$, with $r' = r - 1$. As a consequence, $\mathsf{actsnd}(N') \subseteq \mathsf{actsnd}(N)$. By Lemma A.4(3) if $\mathsf{active}(k', N') \neq t'$ then $\mathsf{active}(k', N) \neq t' + 1 = t$. Notice also that $\mathsf{active}(k, N) = 1$ for all $k \in \mathsf{actsnd}(N) \setminus \mathsf{actsnd}(N')$. Thus, since $t > 1$ for all $k \in \nu \cap \mathsf{actsnd}(N)$ it holds that $\mathsf{active}(k, N) \neq t$. Since M is exposure consistent it follows that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N)$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i \geq t$. Notice that $\nu \setminus \mathsf{nds}(M) = \nu \setminus \mathsf{nds}(M')$. Moreover, by Lemma A.4(1) we have $t_i' = t_i - 1$, for all i. This allows us to derive that there is $\hat{k} \in \nu \setminus \mathsf{nds}(N')$ such that if $\hat{k} \in \nu_i$, for some i, then $t_i' \geq t'$.

- Let $M \xrightarrow{\tau} M'$ by an application of rule (τ-Par). It follows immediately by an application of the inductive hypothesis.

□

Let us prove now that our LTS preserves transmission consistency.

Proposition A.6 (Transmission consistency) *Let M be both an exposure consistent and a transmission consistent network. If $M \xrightarrow{\lambda} M'$ then M' is transmission consistent.*

Proof Let us consider all the possible values of λ.

- Let $M \xrightarrow{m!v} M'$. We have to prove that M' respects the clauses of Definition 2.8. Let examine the three clauses one by one.
  - Clause 1. Let
    $$M' \equiv N' \mid n[(x)_w.Q]^{\nu_n}_{t'_n} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[(x)_w.Q]^{\nu_n}_{t'_n}$$
    with $w \neq \bot$. We have to prove that $\mid \mathsf{actsnd}(N') \cap \nu \mid \leq 1$. By Lemma A.3(2) we have
    $$M' \equiv N' \mid n[(x)_w.Q]^{\nu_n}_{t'_n} \equiv \prod_j n_j[W'_j]^{\nu_j}_{t'_j} \mid m[\langle v \rangle^{\delta_v}.P]^{\nu_m}_{t'_m} \mid n[(x)_w.Q]^{\nu_n}_{t'_n}$$
    and
    $$M \equiv N \mid n[W]^{\nu_n}_{t'_n} = \prod_j n_j[W_j]^{\nu_j}_{t_j} \mid m[!\langle v \rangle.P]^{\nu_m}_{t_m} \mid n[W]^{\nu_n}_{t_n}$$
    for appropriate processes and tags.
    There are two possibilities.
    - If $m \notin \nu_n$ then by Lemma A.3(3) we have $W = (x)_w.Q$. By Lemmas A.3(6) and A.3(7) we have $\mathsf{actsnd}(N') = \mathsf{actsnd}(N) \cup \{m\}$. Since M is transmission consistent, we have $\mid \mathsf{actsnd}(N) \cap \nu_n \mid \leq 1$. Since $m \notin \nu_n$ it follows that $\mid \mathsf{actsnd}(N') \cap \nu_n \mid \leq 1$.
    - If $m \in \nu$ then by Lemma A.3(5) it follows that W is not active sender and $t_n = 0$. By Lemmas A.3(6) and A.3(7) we have $\mathsf{actsnd}(N') = \mathsf{actsnd}(N) \cup \{m\}$. Since $t_n = 0$, $m \in \nu_n$, and M is exposure consistent, clause 3 of Definition 2.7 allows to derive that $\mathsf{actsnd}(N') \cap \nu_n = \{m\}$. Hence, $\mid \mathsf{actsnd}(N') \cap \nu_n \mid \leq 1$.
  - Clause 2. Let
    $$M' \equiv \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid h[\langle w_1 \rangle^r.P]^{\nu_h}_{t'_h} \mid n[(x)_{w_2}.Q]^{\nu_n}_{t'_n}$$
    with $h \in \nu_n$ and $w_2 \neq \bot$. We have to show that $w_2 = w_1$ and $r = t'_n$. There are two cases.

1. Suppose $h \neq m$. In this case, by Lemma A.3(2) we have the following situation:

$$M' \equiv \prod_j n_j[W'_j]^{\nu_j}_{t'_j} \mid m[\langle v\rangle^{\delta_v}.R]^{\nu_m}_{t'_m} \mid h[\langle w_1\rangle^r.P]^{\nu_h}_{t'_h} \mid n[(x)_{w_2}.Q]^{\nu_n}_{t'_n}$$

and

$$M \equiv \prod_j n_j[W_j]^{\nu_j}_{t_j} \mid m[!\langle v\rangle.R]^{\nu_m}_{t_m} \mid h[\langle w_1\rangle^r.P]^{\nu_h}_{t_h} \mid n[W]^{\nu_n}_{t_n}$$

for appropriate processes and tags.
Now, there are two sub-cases.

   (a) If $m \notin \nu_n$ then by Lemma A.3(3) we have $W = (x)_{w_2}.Q$ and $t'_n = t_n$. Since M is transmission consistent we derive $w_2 = w_1$ and $r = t'_n$.
   (b) If $m \in \nu_n$ then by Lemma A.3(5) we have $t_n = 0$. However, since M is exposure consistent by clause 3 of Definition 2.7 it must be $t_n > 0$. So, this case is not possible.

2. Suppose $h = m$. This case easily follows by an application of Lemma A.3(2) and Lemma A.3(5).

– Clause 3. Let

$$M' \equiv N' \mid n[(x)_w.P]^{\nu_n}_{t'_n} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[(x)_w.P]^{\nu_n}_{t'_n}$$

with $\mid \mathsf{actsnd}(N') \cap \nu_n \mid > 1$. We want to show that $w = \bot$. By an application of Lemma A.3(2) it holds that

$$M' \equiv \prod_j n_j[W'_j]^{\nu_j}_{t'_j} \mid m[\langle v\rangle^{\delta_v}.Q]^{\nu_m}_{t'_m} \mid n[(x)_w.P]^{\nu_n}_{t'_n}$$

and

$$M \equiv \prod_j n_j[W_j]^{\nu_j}_{t_j} \mid m[!\langle v\rangle.Q]^{\nu_m}_{t_m} \mid n[W]^{\nu_n}_{t_n}$$

for appropriate processes and tags. Since $\mid \mathsf{actsnd}(N') \cap \nu_n \mid > 1$, it must be $W'_j = \langle w_j\rangle^r.P_j$, for some j. By Lemma A.3(6) we derive that $W_j = W'_j$. At this point we reason by contradiction. Suppose $w \neq \bot$. Then, by Lemma A.3(5) we have $t_n = 0$. However, since M is exposure consistent, by clause 3 of Definition 2.7 it must be $t_n > 0$. This contradiction allows us to conclude that $w = \bot$.

- Let $M \xrightarrow{m?v} M'$. We have to prove that M' respect the clauses of Definition 2.8.
  - Clause 1. Let
$$M' \equiv N' \mid n[(x)_w.Q]^{\nu_n}_{t'_n} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[(x)_w.Q]^{\nu_n}_{t'_n}$$
with $w \neq \bot$. We have to prove that $\mid \mathsf{actsnd}(N') \cap \nu_n \mid \leq 1$. There are two possibilities.
    - If $m \notin \nu_n$ then by Lemma A.3(3) we have
$$M \equiv N \mid n[(x)_w.Q]^{\nu_n}_{t_n} = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[(x)_w.Q]^{\nu_n}_{t_n} \ .$$
Since M is transmission consistent, we have $\mid \mathsf{actsnd}(N) \cap \nu_n \mid \leq 1$. By Lemmas A.3(6) and A.3(7) we derive $\mathsf{actsnd}(N') = \mathsf{actsnd}(N)$. This allows us to derive that $\mid \mathsf{actsnd}(N') \cap \nu_n \mid \leq 1$.
    - If $m \in \nu_n$, since $w \neq \bot$, by Lemma A.3(5) we have
$$M \equiv N \mid n[W]^{\nu_n}_0 = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[W]^{\nu_n}_0$$
where $t'_n = \delta_v$ and W is not an active receiver. By Lemmas A.3(6) and A.3(7) we derive $\mathsf{actsnd}(N') = \mathsf{actsnd}(N)$. Since M is exposure consistent, by clause 3 of Definition 2.7 we derive $\mid \mathsf{actsnd}(N) \cap \nu_n \mid = 0$. As a consequence, $\mid \mathsf{actsnd}(N') \cap \nu_n \mid = 0$.
  - Clause 2. Let
$$M' \equiv \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid h[\langle w_1 \rangle^r.P]^{\nu_h}_{t'_h} \mid n[(x)_{w_2}.Q]^{\nu_n}_{t'_n}$$
with $h \in \nu_n$ and $w_2 \neq \bot$. We have to show that $w_2 = w_1$ and $r = t'_n$. By Lemma A.3(1) we have $h \neq m$. By Lemmas A.3(7) we have
$$M \equiv \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid h[\langle w_1 \rangle^r.P]^{\nu_h}_{t_h} \mid n[W]^{\nu_n}_{t_n}$$
for appropriate processes and tags. Now, there are two cases.
    - If $m \notin \nu_n$ then by Lemma A.3(3) we have $W = (x)_{w_2}.Q$ and $t'_n = t_n$. Since M is transmission consistent it follows that $w_2 = w_1$ and $t'_n = r$.

* If $m \in \nu_n$ then by Lemma A.3(5) we have $t_n = 0$. Since M is exposure consistent, by clause 3 of Definition 2.7 it should be $t_n > 0$. This contradiction shows that this case is not possible.

– Clause 3. Let

$$M' \equiv N' \mid n[(x)_w.P]^{\nu_n}_{t'_n} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[(x)_w.P]^{\nu_n}_{t'_n}$$

with $\mid \mathsf{actsnd}(N') \cap \nu_n \mid > 1$. We have to show that $w = \bot$. By Lemma A.3 we have

$$M \equiv \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[W]^{\nu_n}_{t_n}$$

for appropriate processes and tags. Since $\mid \mathsf{actsnd}(N') \cap \nu_n \mid > 1$ it follows that $W'_j = \langle w_j \rangle^{r_j}.P_j$ and $W'_k = \langle w_k \rangle^{r_k}.P_k$, for some j and k such that $\{n_j, n_k\} \subseteq \nu_n$. By Lemma A.3(1) and Lemma A.3(7) we have $W_j = W'_j$ and $W_k = W'_k$. At this point we reason by contradiction. Suppose $w \neq \bot$. Then, by Lemma A.3(5) we have $t_n = 0$. However, since M is exposure consistent, by clause 3 of Definition 2.7 it must be $t_n > 0$. This contradiction allows us to derive that $w = \bot$.

- Let $M \xrightarrow{\sigma} M'$. We have to prove that M' respects the clauses of Definition 2.8. Let us examine the three clauses one by one.

– Clause 1. Let

$$M' \equiv N' \mid n[(x)_w.Q]^{\nu_n}_{t'_n} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[(x)_w.Q]^{\nu_n}_{t'_n}$$

with $w \neq \bot$. We have to prove that $\mid \mathsf{actsnd}(N') \cap \nu_n \mid \leq 1$. By Lemma A.4(2), since $w \neq \bot$, it must be

$$M \equiv N \mid n[(x)_w.Q]^{\nu_n}_{t_n} = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[(x)_w.Q]^{\nu_n}_{t_n}$$

Since M is transmission consistent it follows that $\mid \mathsf{actsnd}(N) \cap \nu \mid \leq 1$. By Lemma A.4(3) it follows that $\mathsf{actsnd}(N') \subseteq \mathsf{actsnd}(N)$. This implies $\mid \mathsf{actsnd}(N') \cap \nu \mid \leq 1$.

– Clause 2. Let

$$M' \equiv \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid h[\langle w_1 \rangle^r.P]^{\nu_h}_{t'_h} \mid n[(x)_{w_2}.Q]^{\nu_n}_{t'_n}$$

with $h \in \nu_n$ and $w_2 \neq \bot$. We have to show that $w_2 = w_1$ and $r = t'_n$. Since $w_2 \neq \bot$, by Lemmas A.4(1), A.4(2) and A.4(3)

$$M \equiv \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid h[\langle w_1 \rangle^{r+1}.P]^{\nu_h}_{t_h} \mid n[(x)_{w_2}.Q]^{\nu_n}_{t'_n+1} \ .$$

Since M is transmission consistent we have $w_2 = w_1$ and $r+1 = t'_n+1$. As a consequence, $r = t'_n$.

– Clause 3. Let

$$M' \equiv N' \mid n[(x)_w.P]^{\nu_n}_{t'_n} = \prod_i n_i[W'_i]^{\nu_i}_{t'_i} \mid n[(x)_w.P]^{\nu_n}_{t'_n}$$

with $\mid \mathsf{actsnd}(N') \cap \nu_n \mid > 1$. We have to show that $w = \bot$. By an application of Lemma A.4(2) there are two possibilities:

∗ Either

$$M \equiv N \mid n[(x)_w.P]^{\nu_n}_{t_n} = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[(x)_w.P]^{\nu_n}_{t_n} \ .$$

In this case, by Lemma A.4(3) it follows that $\mathsf{actsnd}(N') \subseteq \mathsf{actsnd}(N)$. Thus $\mid \mathsf{actsnd}(N') \cap \nu_n \mid > 1$ implies $\mid \mathsf{actsnd}(N) \cap \nu_n \mid > 1$. Since M is transmission consistent it follows that $w = \bot$.

∗ Or

$$M \equiv N \mid n[W]^{\nu_n}_{t_n} = \prod_i n_i[W_i]^{\nu_i}_{t_i} \mid n[W]^{\nu_n}_{t_n}$$

where W is not an active receiver and $w = \bot$.

- Let $M \xrightarrow{\tau} M'$ by an application of rule (τ-Par). It follows immediately by an application of Proposition 2.4(4).

□

Finally, every thing is in place to prove that network well-formedness is preserved at run time.

Proof of Theorem 2.10
The proof follows by an application of Propositions A.1, A.2, A.5, and A.6. □

Proof of Theorem 3.1
By induction on the length of the proof of $M \xrightarrow{\sigma} M'$. The base cases are when the transition is derived by the application of one of the rules of Table 4 but rule (σ-Par). It is straightforward to prove that the statement holds for these rules. As to the inductive case, let $M \xrightarrow{\sigma} M'$ by an application of rule (σ-Par). This implies that $M = M_1 \mid M_2$, for some M_1 and M_2, with $M_1 \xrightarrow{\sigma} M_1'$, $M_2 \xrightarrow{\sigma} M_2'$ and $M' = M_1' \mid M_2'$. As $M = M_1 \mid M_2$, the transition $M \xrightarrow{\sigma} M''$ can be derived only by applying rule (σ-Par) where $M_1 \xrightarrow{\sigma} M_1''$, $M_2 \xrightarrow{\sigma} M_2''$ and $M'' = M_1'' \mid M_2''$. By inductive hypothesis it holds that M_i' and M_i'' are syntactically the same, for $i \in \{1, 2\}$. This implies that M' and M'' are syntactically the same. □

Proof of Theorem 3.2
By induction on the structure of M. If $M = \mathbf{0}$ the statement does not apply. Let M be composed by only one node with $M \xrightarrow{m!v} N$. In this case the transition can only be derived by an application of rule (Snd) where $M = m[!\langle v\rangle.P]_t^\nu$, for some P, ν and t, and $N = m[\langle v\rangle^{\delta_v}.P]_t^\nu$. Because sender nodes cannot perform σ-actions, there is no network M' such that $M \xrightarrow{\sigma} M'$. Let M be composed by at least two nodes. If $M \xrightarrow{m!v} N$ then by an application of rule (Sync) we have $M = M_1 \mid M_2$ for some M_1 and M_2, with $M_1 \xrightarrow{m!v} M_1'$, $M_2 \xrightarrow{m?v} M_2'$ and $N = M_1' \mid M_2'$ (the converse is similar). In this case the only rule for deriving a σ-transition from M is (σ-Par). However, the inductive hypothesis guarantees that $M_1 \xrightarrow{\sigma} \widehat{M}$ for no network $\widehat{M}$; thus $M \xrightarrow{\sigma} M'$ for no network M'. □

In order to prove Theorem 3.3 on the Patience property, we use the following auxiliary lemma.

Lemma A.7 *Let M be a well-formed network. If $M \xrightarrow{m!v} M'$ then for all network N such that $M \mid N$ is a well-formed network it holds that $M \mid N \xrightarrow{m!v} M' \mid N'$ for some network N'.*

Proof The result follows by Proposition 2.4(1) and an application of rule (Sync).

□

Proof of Theorem 3.3
By contradiction and then by induction on the structure of M. We prove that if $M \xrightarrow{\sigma} N$ for no network N then $M \xrightarrow{m!v} M'$ for some network M'. Let us proceed by induction on the structure of M.

- Let $M = \mathbf{0}$. Then $M \xrightarrow{\sigma} M$ by an application of rule (σ-Zero). So, the statement does not apply.

- Let $M = n[W]_t^\nu$. We proceed by induction on the structure of W.
  - If $W = \mathsf{nil}$ then $M \xrightarrow{\sigma} n[\mathsf{nil}]_{t-1}^\nu$ by an application of rule (σ-Nil). Thus, the statement does not apply.
  - If $W = \sigma.P$ then $M \xrightarrow{\sigma} n[P]_{t-1}^\nu$ by an application of rules (Sleep). Then, the statement does not apply.
  - If $W = !\langle v\rangle.P$ then by inspection on the rules of Table 4 $M \xrightarrow{\sigma} N$ for no network N. However, $M \xrightarrow{m!v} m[\langle v\rangle^{\delta_v}.P]_t^\nu$, by an application of rule (Snd), as expected.
  - If $W = \lfloor ?(x).P \rfloor Q$ and $t = 0$ then $M \xrightarrow{\sigma} n[Q]_0^\nu$, by an application of rule (σ-Rcv). Then, the statement does not apply.
  - If $W = \lfloor ?(x).P \rfloor Q$ and $t > 0$ then $M \xrightarrow{\sigma} n[(x)_\perp.P]_{t-1}^\nu$, by an application of rule (σ-Fail). Then, the statement does not apply.
  - If $W = \lfloor \tau.P \rfloor Q$ then $M \xrightarrow{\sigma} n[Q]_{t-1}^\nu$, by an application of rule (σ-Tau). Then, the statement does not apply.
  - If $W = [v = v]P_1, P_2$ then by an application of rule (Then) we can apply the inductive hypothesis to conclude that we fall in one of the previous cases.
  - If $W = [v_1 = v_2]P_1, P_2$, with $v_1 \neq v_2$, by an application of rule (Else) we can apply the inductive hypothesis to conclude that we fall in one of the previous cases.
  - If $W = H\langle\tilde{v}\rangle$ the constraint of guarded recursion ensures us that by an application of rule (Rec) we can apply the inductive hypothesis and we fall in one of the previous cases.
  - If $W = \langle v\rangle^r.P$ (by definition $r > 0$) then by an application of rule (ActSnd) we have $M \xrightarrow{\sigma} n[\langle v\rangle^{r-1}.P]_{t-1}^\nu$ and the statement does not apply.
  - If $W = (x)_v.P$, with $t > 0$, then by an application of rule (ActRcv) we have $M \xrightarrow{\sigma} n[(x)_v.P]_{t-1}^\nu$ and the statement does not apply.
  - If $W = (x)_v.P$, with $t = 0$, then by an application of rule (RcvEnd) we have $M \xrightarrow{\sigma} n[\{^v/_x\}P]_0^\nu$ and the statement does not apply.
- Let $M = M_1 \mid M_2$. A transition of the form $M \xrightarrow{\sigma} M'$ can be derived only by an application of rule (σ-Par). Thus if M cannot perform a σ-action then at least one of the premises of rule (σ-Par) does not hold:

– If $M_1 \xrightarrow{\sigma} M_1'$ for no network M_1', then by inductive hypothesis we have $M_1 \xrightarrow{m!v} M_1'$, for some M_1'. As $M = M_1 \mid M_2$ is a well-formed network, by Lemma A.7 it holds that $M \xrightarrow{m!v} M_1' \mid M_2'$, for some M_2', in contradiction with the hypothesis.

– If $M_2 \xrightarrow{\sigma} M_2'$ for no network M_2', then we can reason as in the previous sub-case.

□

Proof of Proposition 4.1
By inspection of the code, if the receiver node $n[P_k^j]_t^{\nu_n}$ sends a message auth_i to authenticate packet p_i in a generic epoch k then either $k = i$ or $k = 1 + i$. By definition, the packet p_i is (originally) sent by the sender m at epoch i (the attacker may replay the same packet later on). This suffices to conclude the proof. □

Proof of Proposition 4.2
The proof proceeds by contradiction. We show that if $\#^\sigma(\Lambda_2) > 2\Delta N + 3\Delta T$ then the packet p_i sent by m can not be authenticated by n. Suppose the sender node $m[S_i^j]_t^{\nu_m}$ sends its ciphertext p_i at epoch i and offset j. Without loss of generality, we can suppose $1 \leq \Delta T \leq \delta_{p_i} \leq \Delta N$. Since $\#^\sigma(\Lambda_2) > 2\Delta N + 3\Delta T$, the receiver process R_i^j must miss the synchronisation with S_i^j. Then, after $\#^\sigma(\Lambda_2)$ instants of time the receiver node $n[R_{i'}^{j'}]_{t'}^{\nu_n}$ starts receiving the packet p_i (from the attacker), with $i' = i + 1$ and $j' = j + \#^\sigma(\Lambda_2) - E$, if $j + \#^\sigma(\Lambda_2) - E < E$, and $i' = i + 2$ and $j' = j + \#^\sigma(\Lambda_2) - 2E$, if $j + \#^\sigma(\Lambda_2) - E \geq E$. The receiver node terminates receiving the packet p_i, in some state of the form $P_{i''}^{j''}$. Since $\#^\sigma(\Lambda_2) > 2\Delta N + 3\Delta T$ and $1 \leq \Delta T \leq \delta_{p_i}$. It follows that $i'' \geq i + 2$. However, by inspection of the code, in an epoch greater than or equal as $i + 2$ the receiver node can not send the message auth_i to authenticate a packet of epoch i. This concludes the proof. □

Proof of Theorem 6.3
We prove that the relation

$$\mathcal{S} \stackrel{\text{def}}{=} \{\big(M \mid O\,,\, N \mid O\big) : M \approx N,\ \ M \mid O \text{ and } N \mid O \text{ well-formed } \}$$

is a bisimulation. We proceed by case analysis on why $M \mid O \xrightarrow{\alpha} Z$. The interesting cases are when the transition is due to an interaction between M and O. The remaining cases are simpler.

- Let $M \mid O \xrightarrow{!v \triangleright \nu} \widehat{M}$, by an application of rule (Out) because $M \mid O \xrightarrow{m!v} \widehat{M}$, with $\nu = \mathsf{ngh}(m, M \mid O) \setminus \mathsf{nds}(M \mid O)$ and $\nu \neq \emptyset$. There are two possible cases:

– $M \mid O \xrightarrow{m!v} \widehat{M}$ is derived by an application of rule (Sync) because $M \xrightarrow{m!v} M'$ and $O \xrightarrow{m?v} O'$, with $\widehat{M} = M' \mid O'$. Since $M \xrightarrow{m!v} M'$, by Proposition 2.4(3) it follows that $m \in \mathsf{nds}(M)$. As $M \mid O$ is well-formed, by node-uniqueness, it follows that $m \notin \mathsf{nds}(O)$, and hence $\mathsf{ngh}(m, M \mid O) = \mathsf{ngh}(m, M)$. As $\mathsf{nds}(M \mid O) = \mathsf{nds}(M) \cup \mathsf{nds}(O)$, it follows that $\nu = (\mathsf{ngh}(m, M) \setminus \mathsf{nds}(M)) \setminus \mathsf{nds}(O)$. Let $\nu' = \mathsf{ngh}(m, M) \setminus \mathsf{nds}(M)$. Since $\nu \neq \emptyset$ it follows that $\nu' \neq \emptyset$. Since $\nu' \neq \emptyset$ and $M \xrightarrow{m!v} M'$, by an application of rule (Out) we have $M \xrightarrow{!v \triangleright \nu'} M'$. Now, since $M \approx N$ there is N' such that $N \xRightarrow{!v \triangleright \nu'} N'$ with $M' \approx N'$. Since the action $!v \triangleright \nu'$ can be generated only by an application of rule (Out), there is $h \in \mathsf{nds}(N)$ such that $N \xRightarrow{h!v} N'$ and $\nu' = \mathsf{ngh}(h, N) \setminus \mathsf{nds}(N) \neq \emptyset$. We recall that $N \mid O$ is well-formed. This implies:

  * $h \notin \mathsf{nds}(O)$, by node-uniqueness;
  * $\nu' \subseteq \mathsf{ngh}(h, N)$;
  * If $k \in \nu' \cap \mathsf{nds}(O)$ then $h \in \mathsf{ngh}(k, O)$, because the neighbouring relation is symmetric (by Definition 2.6).

  This implies that $O \xrightarrow{h?v} O'$. By an application of rule (Sync) and several applications of rule (TauPar) we have $N \mid O \xRightarrow{h!v} N' \mid O'$. Since $h \notin \mathsf{nds}(O)$ it follows that $\mathsf{ngh}(h, N \mid O) = \mathsf{ngh}(h, N)$. We recall that $\nu' = \mathsf{ngh}(h, M) \setminus \mathsf{nds}(M) = \mathsf{ngh}(h, N) \setminus \mathsf{nds}(N)$. Thus, we have the following sequence of equalities:

$$\begin{array}{rcl} \nu & = & \mathsf{ngh}(m, M \mid O) \setminus \mathsf{nds}(M \mid O) \\ & = & \mathsf{ngh}(m, M) \setminus \mathsf{nds}(M \mid O) \\ & = & (\mathsf{ngh}(m, M) \setminus \mathsf{nds}(M)) \setminus \mathsf{nds}(O) \\ & = & (\mathsf{ngh}(h, N) \setminus \mathsf{nds}(N)) \setminus \mathsf{nds}(O) \\ & = & \mathsf{ngh}(h, N \mid O) \setminus \mathsf{nds}(N \mid O) \\ & \neq & \emptyset \ . \end{array}$$

  As $\nu \neq \emptyset$, by an application of rule (Out) we have $N \mid O \xRightarrow{!v \triangleright \nu} N' \mid O'$. By Theorem 2.10, both $M' \mid O'$ and $N' \mid O'$ are well-formed. As $M' \approx N'$ it follows that $\big(M' \mid O' \,,\, N' \mid O'\big) \in \mathcal{S}$.

– $M \mid O \xrightarrow{m!v} \widehat{M}$, by an application of rule (Sync), because $M \xrightarrow{m?v} M'$ and $O \xrightarrow{m!v} O'$, with $\widehat{M} = M' \mid O'$. Since $O \xrightarrow{m!v} O'$, by Proposition 2.4(3) it follows that $m \in \mathsf{nds}(O)$, and hence $\nu = (\mathsf{ngh}(m, O) \setminus \mathsf{nds}(O)) \setminus \mathsf{nds}(M)$. Since $M \approx N$ there is N' such that $N \xRightarrow{m?v} N'$ with $M' \approx N'$. By several applications of rule (TauPar) and one

application of rule (Sync) (in its symmetric version) it follows that $N \mid O \stackrel{m!v}{\Longrightarrow} N' \mid O'$. By Proposition 6.2, $M \approx N$ implies that $\mathsf{nds}(M) = \mathsf{nds}(N)$. Moreover, since $M \mid O$ and $N \mid O$ are well-formed and $m \in \mathsf{nds}(O)$, by node uniqueness it follows that $m \notin \mathsf{nds}(M)$ and $m \notin \mathsf{nds}(N)$. Thus,

$$\begin{array}{rcl} \mathsf{ngh}(m, N \mid O) \setminus \mathsf{nds}(N \mid O) & = & (\mathsf{ngh}(m, O) \setminus \mathsf{nds}(O)) \setminus \mathsf{nds}(N) \\ & = & (\mathsf{ngh}(m, O) \setminus \mathsf{nds}(O)) \setminus \mathsf{nds}(M) \\ & = & \mathsf{ngh}(m, M \mid O) \setminus \mathsf{nds}(M \mid O) \\ & = & \nu \\ & \neq & \emptyset \ . \end{array}$$

With this premise, by an application of rule (Out) we can derive $N \mid O \stackrel{!v \triangleright \nu}{\Longrightarrow} N' \mid O'$. By Theorem 2.10, both $M' \mid O'$ and $N' \mid O'$ are well-formed. As $M' \approx N'$ it follows that $\left(M' \mid O' \, , \, N' \mid O'\right) \in \mathcal{S}$.

- Let $M \mid O \stackrel{\tau}{\longrightarrow} \widehat{M}$, by an application of rule (Shh), because $M \mid O \stackrel{m!v}{\longrightarrow} \widehat{M}$ and $\mathsf{ngh}(m, M \mid O) \subseteq \mathsf{nds}(M \mid O)$. There are two possible cases:

  - Let $M \mid O \stackrel{m!v}{\longrightarrow} \widehat{M}$, by an application of rule (Sync), because $M \stackrel{m!v}{\longrightarrow} M'$ and $O \stackrel{m?v}{\longrightarrow} O'$, with $\widehat{M} = M' \mid O'$. As $m \in \mathsf{nds}(M)$ and $M \mid O$ is well-formed it follows that $m \notin \mathsf{nds}(O)$. Thus,

    $$\mathsf{ngh}(m, M \mid O) \setminus \mathsf{nds}(M \mid O) = (\mathsf{ngh}(m, M) \setminus \mathsf{nds}(M)) \setminus \mathsf{nds}(O) = \emptyset \ .$$

    Again there are two possibilities:

    * Let $\mathsf{ngh}(m, M) \setminus \mathsf{nds}(M) = \emptyset$. Then, since $M \stackrel{m!v}{\longrightarrow} M'$, by an application of rule (Shh) we have $M \stackrel{\tau}{\longrightarrow} M'$. Since $M \approx N$ there is N' such that $N \Rightarrow N'$ and $M' \approx N'$. We know that $O \stackrel{m?v}{\longrightarrow} O'$. Let us assume $O \neq \mathbf{0}$ (the case when $O = \mathbf{0}$ is simple). By definition of our networks there are n_i, W_i, ν_i and t_i, for $1 \leq i \leq k$, such that $O = \prod_{i=1}^{k} n_i[W_i]_{t_i}^{\nu_i}$. Since $O \stackrel{m?v}{\longrightarrow} O'$ by Proposition 2.4(2), for all i, $1 \leq i \leq k$, there are W_i', ν_i', and t_i' such that

      $$n_i[W_i]_{t_i}^{\nu_i} \stackrel{m?v}{\longrightarrow} n_i[W_i']_{t_i'}^{\nu_i'}$$

      and $O' = \prod_{i=1}^{k} n_i[W_i']_{t_i'}^{\nu_i'}$. Since $M \mid O$ is well-formed, by node-uniqueness it follows that $n_i \notin \mathsf{nds}(M)$ for all i, $1 \leq i \leq k$. Now, since

· $\mathsf{ngh}(m, M) \setminus \mathsf{nds}(M) = \emptyset$

· $n_i \notin \mathsf{nds}(M)$, for all i

· $M \mid O$ is connected and the neighbouring relation is symmetric (see clause 2 of Definition 2.6)

it follows that $m \notin \nu_i$, for all i, $1 \leq i \leq k$. This implies that the transitions

$$n_i[W_i]^{\nu_i}_{t_i} \xrightarrow{m?v} n_i[W_i']^{\nu_i'}_{t_i'}$$

can only be derived by applying rule (OutRng) with $W_i' = W_i$, $\nu_i' = \nu_i$ and $t_i' = t_i$. This implies $O' = O$. Now, since $N \Rightarrow N'$, by several applications of rule (TauPar) it follows that $N \mid O \Rightarrow N' \mid O = N' \mid O'$. By Theorem 2.10, both $M' \mid O'$ and $N' \mid O'$ are well-formed. As $M' \approx N'$ it follows that $\left(M' \mid O' \,,\, N' \mid O'\right) \in \mathcal{S}$.

∗ Let $\nu' = \mathsf{ngh}(m, M) \setminus \mathsf{nds}(M) \neq \emptyset$. The reasoning in this sub-case is very similar to that done in the first case (when $\alpha = !v \triangleright \nu$) of this proof.

– Let $M \mid O \xrightarrow{m!v} \widehat{M}$, by an application of rule (Sync) because $M \xrightarrow{m?v} M'$ and $O \xrightarrow{m!v} O'$, with $\widehat{M} = M' \mid O'$. This case is similar to a previous one.

- Let $M \mid O \xrightarrow{m?v} \widehat{M}$, by an application of rule (RcvPar), because $M \xrightarrow{m?v} M'$, $O \xrightarrow{m?v} O'$ and $\widehat{M} = M' \mid O'$. This case is easy.

- Let $M \mid O \xrightarrow{\sigma} \widehat{M}$ by an application of rule (σ-Par) because $M \xrightarrow{\sigma} M'$, $O \xrightarrow{\sigma} O'$ and $\widehat{M} = M' \mid O'$. This case is easy.

□