IT University of Copenhagen

A Calculus for Mobile Ad Hoc Networks

Jens Chr. Godskesen

IT University Technical Report Series **TR-2007-98**

ISSN 1600–6100 **May 2007**

ISSN 1600–6100

ISBN 978-87-7949-154-0

Copies may be obtained by contacting:

IT University of Copenhagen
Rued Langgaards Vej 7
DK-2300 Copenhagen S
Denmark

Telephone: **+45 72 18 50 00**
Telefax: **+45 72 18 50 01**
Web `www.itu.dk`

A Calculus for Mobile Ad Hoc Networks

Jens Chr. Godskesen ⋆

IT University of Copenhagen
Rued Langgaards Vej 7
DK-2300 Copenhagen S, Denmark
jcg@itu.dk

Abstract. We suggest a *Calculus for Mobile Ad Hoc Networks*, CMAN. A node in a network is a process equipped with a location, it may communicate with other nodes using *synchronous spatially oriented broadcast* where only the current neighbors receive the message. Nodes may autonomously change their neighbor relationship and thereby change the network topology. We define a natural reduction semantics and strong and weak reduction congruences as well as a labeled transition semantics and prove strong and weak contextual bisimulation respectively to be *sound* and *complete* co-inductive characterizations of the corresponding reduction congruences. For the subset of *connection closed* networks we show a significantly simpler co-inductive characterization. Finally, we apply CMAN on a small example of a cryptographic routing protocol.

1 Introduction

The use of wireless networks is becoming more and more important due to the increasing and widespread use of communicating mobile devices. The application area for wireless networks is broad, spanning from ambient intelligence, wireless local area networks, sensor networks, and cellular networks for mobile telephony.

Our work is devoted to a particular kind of wireless networks, the so called *Mobile Ad Hoc Networks* (MANETS). MANETS are self organizing wireless networks without centralized access points or any other central control components. Hence they do not contain a pre-deployed infrastructure for routing messages. An ad hoc network may be formed when a collection of mobile nodes join together and agree on how to route messages for each other over possibly multiple hops.

The communication primitive for wireless devices is message broadcast. However in contrast to the conventional technology in wired local area networks, say the Ethernet, where broadcasted messages reach every node in the network, then for wireless networks broadcast is *spatially oriented* meaning that messages will only reach those nodes within the communication range (the cell) of the emitting node. Another difference between wired and wireless network technology is that interference is a much harder and severe problem in wireless systems. Also, in wireless networks communication links between entities cannot always be considered bidirectional.

Calculi for broadcast systems were first studied by Prasad in the work on the CBS calculus [16] and later in a mobile setting by Ene and Muntean in the $b\pi$ calculus [5], and by Ostrovsky, Prasad, and Taha in HOBS [15]. Recently wireless broadcast systems have been studied by Nanz and Hankin in CBS# [13] and by Merro in CMN [8]. In the former calculi broadcast scope is *transitive* in that if two nodes P and Q both can communicate with a third node then P and Q can also communicate with each other whereas this is not necessarily the case for CBS# and CMN. The calculus CWS [9] by Mezzetti and Sangiorgi also studies wireless broadcast but at a much lower level of abstraction, in particular they take the phenomenon of interference into account.

Another characteristic of MANETS is that nodes may be mobile, not only do they enter and leave the network, but also they autonomously change localities and thereby change their connections and hence the topology of the network. Mobility of processes has been addressed by many calculi, like π [11], Mobile Ambients [3], Seal [4], and Homer [7], and some even take the notion of spatially oriented communication into account like Mobile Ambients and Hennessy and Riely's $D\pi$ [17]. However only very limited work has so far been devoted to calculi for broadcast and mobility, like $b\pi$ and HOBS, and to our knowledge the only reported work on calculi for spatially oriented broadcast and mobility is CBS# and CMN.

The goal of our work is to define a *Calculus for Mobile Ad Hoc Networks* (CMAN) that facilitates mobility and spatially oriented broadcast. As in CMN we adopt that communication between nodes in a network is carried out on

⋆ Supported by grant no. 272-05-0258 from the Danish Research Agency.

bidirectional links, and further we assume that nodes in a network may move arbitrarily as in both CBS# and CMN. We shall refrain from dealing with interference in this paper.

The neighborhood relation in CBS# is dealt with at the semantic level, the semantics is parameterized and quantified over a set of configurations (graphs). In CMN and CWS the neighborhood relation is taken care of by a metric function that tells if two physical locations are close enough to communicate. Here instead we choose *logical locations* and follow to some extent the ideas by De Nicola et al. [14] letting the topology be explicitly part of the network syntax and letting the topology change as a consequence of computational steps. We choose as a key design principle of our calculus that the specification of a node's control behaviour must be independent of and not intermixed with its neighborhood coordination as this would render models in the calculus unnecessarily complex.

We follow the approach from CBS# and CMN (and CWS) letting broadcast be spatially oriented, but in contrast to CBS#, where broadcast messages may be received after a change of network topology, we let broadcast as in CMN be *atomic* in the sense that all neighbors at the time of the broadcast, and only those, can listen to and receive the broadcasted message. Another similarity with CMN is that we allow broadcasted messages to be lost for some potential recipients. However, opposite to CMN where broadcast is carried out on channels that may be restricted, we let broadcasted messages be transmitted on an unrestricted medium.

One important factor of motivation is that we want to be able to model cryptographic routing protocols for MANETS, like ARAN [18]. For that reason we choose to adopt a data (term) language as the one known from the Applied π-Calculus [2].

A *node*, $\lfloor p \rfloor_l^\sigma$, in our calculus is modeled as a (sequential) *process* p located at some (logical) *location* l and connected to other nodes at locations σ. A location is an abstract name that cannot be referred by the node's process. Nodes put together in parallel constitute a *network*, say

$$P = \lfloor p \rfloor_l^m \parallel \lfloor q \rfloor_m^l \parallel \lfloor r \rfloor_n \ ,$$

where the current topology is that the node at location l, $\lfloor p \rfloor_l^m$, is connected to the node at location m, $\lfloor q \rfloor_m^l$, (and vice versa). The node at location n is disconnected from any other node. Mobility is obtained by a simple reduction, say that the node at location n autonomously moves and becomes (bidirectionally) *connected* to the node at location l,

$$\lfloor p \rfloor_l^m \parallel \lfloor q \rfloor_m^l \parallel \lfloor r \rfloor_n \searrow \lfloor p \rfloor_l^{mn} \parallel \lfloor q \rfloor_m^l \parallel \lfloor r \rfloor_n^l \ . \tag{1}$$

Similarly, nodes may arbitrarily *disconnect*, say

$$\lfloor p \rfloor_l^{mn} \parallel \lfloor q \rfloor_m^l \parallel \lfloor r \rfloor_n^l \searrow \lfloor p \rfloor_l^n \parallel \lfloor q \rfloor_m \parallel \lfloor r \rfloor_n^l \ . \tag{2}$$

A node containing a process $\langle t \rangle.p$ may broadcast t and a node with $(x).q$ can receive a broadcasted message. *Synchronous spatially oriented broadcast* is realized by a *broadcast reduction* labelled by the location of the emitting node, say

$$\lfloor \langle t \rangle.p \rfloor_l^{nm} \parallel \lfloor (x).q \rfloor_m^l \parallel \lfloor (x).r \rfloor_n^l \searrow_l \lfloor p \rfloor_l^{nm} \parallel \lfloor q\{t/x\} \rfloor_m^l \parallel \lfloor r\{t/x\} \rfloor_n^l \ , \tag{3}$$

where the node at location l broadcasts to all nodes to which it is connected in the current topology, or similarly

$$\lfloor \langle t \rangle.p \rfloor_l^{nm} \parallel \lfloor (x).q \rfloor_m^l \parallel \lfloor (x).r \rfloor_n^l \searrow_l \lfloor p \rfloor_l^{nm} \parallel \lfloor (x).q \rfloor_m^l \parallel \lfloor r\{t/x\} \rfloor_n^l \ , \tag{4}$$

where the broadcasted message to one of l's neighbors, in this case the node at location m, is lost. As a special case, a disconnected node in a network may broadcast without anyone listening

$$\lfloor \langle t \rangle.p \rfloor_l \parallel \lfloor (x).q \rfloor_m \parallel \lfloor (x).r \rfloor_n \searrow_l \lfloor p \rfloor_l \parallel \lfloor (x).q \rfloor_m \parallel \lfloor (x).r \rfloor_n \ . \tag{5}$$

A novel contribution of our work is that we choose to work with a family of broadcast reductions, one for each locality in the network. This allows an external observer to observe the locality (node) in charge of the synchronous broadcast.

However, since it may be unrealistic for an observer to cover the whole network we introduce the notion of a *hidden node*, i.e. a node with the location name restricted. A hidden node, say $\nu k.\lfloor \langle t \rangle.r \rfloor_k$, may connect to other nodes extruding its location name,

$$\lfloor (x).p \rfloor_l^m \parallel \lfloor q \rfloor_m^l \parallel \nu k.\lfloor \langle t \rangle.r \rfloor_k \searrow \nu k.(\lfloor (x).p \rfloor_l^{mk} \parallel \lfloor q \rfloor_m^l \parallel \lfloor \langle t \rangle.r \rfloor_k^l) \ , \tag{6}$$

and subsequently send (receive) messages to (from) its neighbors, e.g.

$$\nu k.(\lfloor (x).p \rfloor_l^{mk} \parallel \lfloor q \rfloor_m^l \parallel \lfloor \langle t \rangle.r \rfloor_k^l) \searrow \nu k.(\lfloor p\{t/x\} \rfloor_l^{mk} \parallel \lfloor q \rfloor_m^l \parallel \lfloor r \rfloor_k^l) \ , \tag{7}$$

but the emission from a hidden node cannot be observed by an external observer, hence the reduction (7) is not a broadcast reduction.

As in the seminal work on barbed bisimulation [12, 10] we strive to have an as simple as possible reduction semantics and to allow an external global observer to have minimal observability, in our case: reductions $\searrow_l$ for broadcast, and reductions $\searrow$ for connections, disconnections, and broadcast from hidden nodes. Similar to the semantics of CMN and CBS# we choose to abstract from observability of node mobility. Indistinguishability under these observations gives rise to natural strong and weak equivalences which in turn induces natural strong and weak congruences over networks, i.e. the strong and weak equivalences in all contexts closed under structural congruence. In the present paper we show how to obtain a labeled transition semantics such that (early contextual) strong and weak bisimulation are *sound* and *complete* co-inductive characterizations of the the strong and weak reduction congruences respectively.

The paper is organized as follows: The language of CMAN is presented in Section 2. The reduction semantics and the natural reduction congruences follows in Section 3. In Section 4 we provide the labeled transition system semantics and give the co-inductive characterizations of the reduction congruences. Then, for a sub-calculus of CMAN, in Section 5 we demonstrate a considerably simpler characterization of the reduction congruence. We end the paper with a simple example of a cryptographic routing protocol and a conclusion. Proofs are to be found in the appendix.

2 Syntax

As already touched upon above a network in CMAN consists of nodes composed in parallel, some nodes may be hidden, and each node is a sequential process at some abstract location connected to other locations.

Our process definition is similar to the one in [1], a variant of the *Applied π-Calculus* ($A\pi$) [2]. $A\pi$ is a simple extension of the π-Calculus [11] with value passing, primitive functions, and term equations.

2.1 Terms

Terms are defined relative to an infinite set of *names* $\mathcal{N}$ ranged over by n, an infinite set of *variables* $\mathcal{X}$ ranged over by x, and two disjoint finite sets, $\mathcal{F}$ and $\mathcal{G}$, of *constructor* and *destructor* symbols ranged over by f and g respectively. Formally, destructors are defined to be partial functions, i.e. the application of a destructor to a tuple of terms is only defined in case the tuple matches one of the destructors defining equations (we refer the reader to [1]).

Then the set of terms is defined as follows:

$$s, t ::= n \mid x \mid f(t_1, \ldots, t_k) \mid (t_1, \ldots, t_i) \ ,$$

where f is a constructor symbol with arity k. We let $\mathcal{T}$ denote the set of all terms with no variables.

2.2 Processes

As mentioned above, processes in CMAN are based on the process constructs from $A\pi$. We choose although to omit the notion of a *channel*, letting everyone able to listen be a potential receiver of the broadcasted message.[1] We assume a set of process variables $\mathcal{Z}$ ranged over by z. The set of processes is defined by the grammar:

$$\begin{aligned} p, q ::= 0 \mid \langle t \rangle.p \mid (x).p \mid \mathit{if}\ (t = s)\ \mathit{then}\ p\ \mathit{else}\ q \mid \mathit{let}\ x = t\ \mathit{in}\ p \mid \\ \mathit{let}\ x = g(t_1, \ldots, t_i)\ \mathit{in}\ p\ \mathit{else}\ q \mid \nu n.p \mid z \mid \mathit{rec}\ z.p \ . \end{aligned}$$

The process 0 is the inactive process. $\langle t \rangle.p$ may output t and in so doing become p. The process $(x).p$ binds x in p and may input a term t and replace all free occurrences of x in p by t. The process $\mathit{if}\ t = s\ \ \mathit{then}\ p\ \mathit{else}\ q$ is a standard conditional. The local definition $\mathit{let}\ x = t\ \mathit{in}\ p$ binds the variable x in p and executes p with all free occurrences of x replaced by t. The process $\mathit{let}\ x = g(t_1, \ldots, t_k)\ \mathit{in}\ p\ \mathit{else}\ q$ also binds x in p, if the destructor application

[1] Another approach would be to broadcast on a given channel as in CMN and $b\pi$.

$g(t_1, \ldots, t_k)$ evaluates to a term t then x is bound to t in p, otherwise the process becomes q. The process $\nu n.p$ binds the name n in p and restricts n to p. Finally, $rec\ z.p$ is a recursively defined process where $rec\ z$ binds z in p. [2]

We let $p\{t/x\}$ denote the process p where any free occurrence of x is substituted by t (taking care that names in t are not bound in p by the use of α-conversion if needed). Likewise, $p\{q/z\}$ denotes the process p where z is substituted by q. The set of *free names* in a process p is denoted by $fn(p)$, and its *free variables* are denoted by $fv(p)$. A process p is (variable) *closed* if $fv(p) = \emptyset$. $\mathbf{P}$ denotes the set of all closed processes and as usual we identify processes up to α-equivalence.

2.3 Networks

Assume a finite set of *location* names $\mathcal{L}$ ranged over by l and k. We assume $\mathcal{N} \cap \mathcal{L} = \emptyset$ and let m range over $\mathcal{N} \cup \mathcal{L}$. We let σ range over sets of location names and let ϵ denote the empty set. The set of networks is defined by the grammar:

$$P, Q, R ::= 0 \mid \lfloor p \rfloor_l^\sigma \mid \nu m.P \mid P \parallel Q \ .$$

The network 0 denotes the empty network. $\lfloor p \rfloor_l^\sigma$ is a singleton network with the node at location l containing the process p and connected to nodes in σ. $\nu m.P$ is the network P with the (location or term) name m hidden, and finally $P \parallel Q$ is the parallel composition of the two networks P and Q. [3] As a shorthand we allow to write $\Pi_{i \in I} P_i$ for the parallel composition of all networks P_i, $i \in I$.

We let the hiding operator have higher precedence than parallel composition. We write $\lfloor p \rfloor_l$ instead of $\lfloor p \rfloor_l^\epsilon$. When $\tilde{m} = \{m_1, \ldots, m_i\}$ we write $\tilde{m}m$ for $\tilde{m} \cup \{m\}$ and we write $\nu\tilde{m}$ instead of $\nu m_1 \ldots \nu m_i$. We write σl instead of $\sigma \cup \{l\}$ and let $\sigma\sigma'$ denote the union of disjoint sets σ and σ'.

The set of *free names* in a network P, denoted by $fn(P)$, is defined as expected and so is the set of *free variables* $fv(P)$. We let $P\{t/x\}$ denote the network P where all free occurrences of x in P is substituted by t (taking care that names in t are not bound in P using α-conversion if needed). The set of *free locations* in a network P, denoted by $fl(P)$, is inductively defined by: $fl(\lfloor p \rfloor_l^\sigma) = \{l\}$, $fl(\nu m.P) = fl(P) \setminus \{m\}$, and $fl(P \parallel Q) = fl(P) \cup fl(Q)$. The set of *free connections* in a network P, denoted by $fc(P)$, is inductively defined by: $fc(\lfloor p \rfloor_l^\sigma) = \sigma$, $fc(\nu m.P) = fc(P) \setminus \{m\}$, and $fc(P \parallel Q) = fc(P) \cup fc(Q)$. Finally, the set of free locations and connections in a network P is denoted by $flc(P) = fl(P) \cup fc(P)$.

As a syntactical convention we allow to write $P_{l \oplus k}$ meaning that the node in P (if any) with location name l is connected to a node with location name k, and symmetrically node k in P (if any) is connected to l. Formally we define $P_{l \oplus k}$ inductively by: $0_{l \oplus k} = 0$, and $(\lfloor p \rfloor_l^\sigma)_{l \oplus k} = \lfloor p \rfloor_l^{\sigma k}$, $(\lfloor p \rfloor_k^\sigma)_{l \oplus k} = \lfloor p \rfloor_k^{\sigma l}$, and $(\lfloor p \rfloor_m^\sigma)_{l \oplus k} = \lfloor p \rfloor_m^\sigma$ if $m \notin \{l, k\}$, $(P \parallel P')_{l \oplus k} = P_{l \oplus k} \parallel P'_{l \oplus k}$, $(\nu m.P)_{l \oplus k} = \nu m.(P_{l \oplus k})$ if $m \notin \{l, k\}$. Similarly, we let $P_{l \ominus k}$ denote the network where k is not connected to node l, and vice versa. We let $l \oplus k$ and $l \ominus k$ have higher precedence than the hiding operator.

2.4 Well-formedness

We say that a network P is *well-formed* if each node in P is not connected to itself and if each free location in P is unique. Formally, well-formedness is inductively defined by:

- $\lfloor p \rfloor_l^\sigma$ is well-formed if $l \notin \sigma$.
- $P \parallel Q$ is well-formed if P and Q are well-formed and if $fl(P) \cap fl(Q) = \emptyset$.
- $\nu m.P$ is well-formed if P is well-formed.

In the sequel we consider only the set of well-formed networks and we identify networks up to $alpha$-equivalence. The set of well-formed and variable closed networks is denoted by $\mathbf{N}$.

[2] Notice, that in the present version of CMAN we have left out parallel composition and replication of processes.
[3] As in [14] we have no operator for having an unbounded number of network nodes.

$let\ x = t\ in\ p\ \equiv_{\mathbf{P}}\ p\{t/x\}$	$if\ (t = t)\ then\ p\ else\ q \equiv_{\mathbf{P}} p$
$if\ (t = s)\ then\ p\ else\ q\ \equiv_{\mathbf{P}}\ q$, if $t \neq s$	$rec\ z.p \equiv_{\mathbf{P}}\ p\{rec\ z.p/z\}$
$let\ x = g(t_1, \ldots, t_i)\ in\ p\ else\ q\ \equiv_{\mathbf{P}}\ p\{t/x\}$, if $g(t_1, \ldots, t_i) = t$	
$let\ x = g(t_1, \ldots, t_i)\ in\ p\ else\ q\ \equiv_{\mathbf{P}}\ q$, if $g(t_1, \ldots, t_i)$ not defined	

Table 1. Structural congruence, processes.

$P \parallel 0 \equiv P$	$P \parallel Q \equiv Q \parallel P$	$(P \parallel P') \parallel P'' \equiv P \parallel (P' \parallel P'')$
	$\lfloor p \rfloor_l^\sigma \equiv \lfloor q \rfloor_l^\sigma$, if $p \equiv_{\mathbf{P}} q$	$\lfloor \nu n.p \rfloor_l^\sigma \equiv \nu n.\lfloor p \rfloor_l^\sigma$
$\nu m.\nu m'.P \equiv \nu m'.\nu m.P$	$\nu m.P \parallel Q \equiv \nu m.(P \parallel Q)$, if $m \notin fn(Q) \cup flc(Q)$	

Table 2. Structural congruence, networks.

3 Reduction Semantics

We provide our calculus with a reduction semantics defined through the use of evaluation contexts, structural congruence, and reduction rules.

As usual we say that a binary relation $\mathcal{R}$ on $\mathbf{P}$ is a *congruence* if $p\ \mathcal{R}\ q$ implies $C(p)\ \mathcal{R}\ C(q)$ for any process context C. Structural congruence on $\mathbf{P}$, $\equiv_{\mathbf{P}}$, is the least congruence and equivalence relation that is closed under α-conversion and the rules in Table 1. Likewise, we say that a binary relation $\mathcal{R}$ on $\mathbf{N}$ is a *congruence* if $P\ \mathcal{R}\ P'$ implies $\nu m.P\ \mathcal{R}\ \nu m.P'$ for all m, and $P \parallel Q\ \mathcal{R}\ P' \parallel Q$ for all Q with $fl(Q) \cap (fl(P) \cup fl(P')) = \emptyset$. Structural congruence on $\mathbf{N}$, $\equiv$, is the least congruence and equivalence relation that is closed under α-conversion and the rules in Table 2.

3.1 Reduction Rules

We define a reduction $\searrow_l \subseteq \mathbf{N} \times \mathbf{N}$ for each $l \in \mathcal{L}$ as the least relation closed under structural congruence, parallel composition, and satisfying the rules in Table 3. Also, we define $\searrow \subseteq \mathbf{N} \times \mathbf{N}$ as the least relation closed under structural congruence, parallel composition, and restriction, and satisfying the rules in Table 3. We let $\searrow^*$ denote the reflexive and transitive closure of $\searrow$.

A reduction due to rule (con) in Table 3 signifies that a bidirectional connection within the network has taken place, and likewise a reduction due to (dis) means that a disconnection has happened.

A reduction due to rule (brd) means that the node at location l synchronously broadcasts a message to neighbors to which it is currently connected and which are capable of listening. Notice that the rule (brd) captures that broadcast is an atomic step, hence no node outside the range of the emitting node at the time of transmission can ever receive the broadcasted message. Also note that broadcasted messages may be lost, i.e. not only will neighbors to which l is connected but which are not listening for sure lose the message, but also connected neighbors that are listening are not guaranteed to receive the emitted message as demonstrated by reduction (4) in the Introduction.

Rule (res) allows broadcasting from non-hidden localities to be observable, and dually rule $(hide)$ makes emission from hidden nodes unobservable. For reduction examples we refer the reader to (1) – (7) in the Introduction.

3.2 Reduction Congruences

Based on the reductions above we introduce a natural strong and a also a weak congruence for CMAN.

We say that a binary relation $\mathcal{R}$ on $\mathbf{N}$ is *strong reduction closed* if whenever $P\ \mathcal{R}\ Q$ then $P \searrow P'$ implies the existence of some Q' such that $Q \searrow Q'$ and $P'\ \mathcal{R}\ Q'$, and $P \searrow_l P'$ implies the existence of some Q' such $Q \searrow_l Q'$ and $P'\ \mathcal{R}\ Q'$. Likewise, we say that a binary relation $\mathcal{R}$ on $\mathbf{N}$ is *weak reduction closed* if whenever $P\ \mathcal{R}\ Q$ then $P \searrow P'$ implies the existence of some Q' such that $Q \searrow^* Q'$ and $P'\ \mathcal{R}\ Q'$, and $P \searrow_l P'$ implies the existence of some Q' such $Q \searrow^* \searrow_l \searrow^* Q'$ and $P'\ \mathcal{R}\ Q'$.

$$(con)\ \frac{}{\lfloor p \rfloor_l^{\sigma} \parallel \lfloor q \rfloor_k^{\sigma'} \searrow \lfloor p \rfloor_l^{\sigma k} \parallel \lfloor q \rfloor_k^{\sigma' l}} \qquad (dis)\ \frac{}{\lfloor p \rfloor_l^{\sigma k} \parallel \lfloor q \rfloor_k^{\sigma' l} \searrow \lfloor p \rfloor_l^{\sigma} \parallel \lfloor q \rfloor_k^{\sigma'}}$$

$$(brd)\ \frac{}{\lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'} \parallel \Pi_{m \in \sigma} \lfloor (x).p_m \rfloor_m^{\sigma_m l} \searrow_l \lfloor p \rfloor_l^{\sigma\sigma'} \parallel \Pi_{m \in \sigma} \lfloor p_m\{t/x\} \rfloor_m^{\sigma_m l}}$$

$$(res)\frac{P \searrow_l P'}{\nu m.P \searrow_l \nu m.P'}\ \ m \neq l \qquad (hide)\frac{P \searrow_l P'}{\nu l.P \searrow \nu l.P'}$$

Table 3. Reduction rules.

Definition 1. *A symmetric relation $\mathcal{R}$ on $\mathbf{N}$ is a* strong reduction bisimulation *if it is strong reduction closed and if $P\ \mathcal{R}\ Q$ implies $fl(P) = fl(Q)$.*

Definition 2. *A symmetric relation $\mathcal{R}$ on $\mathbf{N}$ is a* weak reduction bisimulation *if it is weak reduction closed and if $P\ \mathcal{R}\ Q$ implies $fl(P) = fl(Q)$.*

Strong and weak reduction bisimulation are equivalence relations.

Notice that in reduction bisimulations the location name of the (non-hidden) broadcasting location is observable, however we do not use barbs as for instance in [12], but instead make broadcasting from a node a location l say be observable through reductions of type $\searrow_l$. As usual weak reduction bisimulation abstracts from internal computation, in our case change of connectivity and broadcast from hidden nodes.

Definition 3. *A relation $\mathcal{R}$ on $\mathbf{N}$ is a* strong reduction congruence *if it is a strong reduction bisimulation and a congruence.*

We let $\simeq$ denote the largest strong reduction congruence.

Definition 4. *A relation $\mathcal{R}$ on $\mathbf{N}$ is a* weak reduction congruence *if it is a weak reduction bisimulation and a congruence.*

We let $\cong$ denote the largest weak reduction congruence.

4 Labeled Transition System Semantics

In order to give an alternative co-inductive characterization of the weak reduction congruence, $\cong$, we provide a labeled transition system semantics of our calculus. We begin with the semantics for plain processes and proceed with the semantics for networks.

4.1 Process Semantics

Let the set of *process actions*, $\mathcal{A}_{\mathbf{P}}$, ranged over by λ be defined by:

$$\lambda ::= (t) \mid \nu\tilde{n}\langle t \rangle$$

where $t \in \mathcal{T}$. The action (t) describes that the term t is received by a process and the action $\nu\tilde{n}\langle t \rangle$ denotes the emission of the term t with names in $\tilde{n}$ bound. If $\tilde{n} = \emptyset$ we write $\langle t \rangle$ instead of $\nu\emptyset\langle t \rangle$. We let $fn(\lambda)$ ($bn(\lambda)$) denote the bound (free) names in λ.

The operational semantics for processes is defined as a labeled transition system $(\mathbf{P}, \mathcal{A}_{\mathbf{P}}, \rightarrow)$ where $\rightarrow \subseteq \mathbf{P} \times \mathcal{A}_{\mathbf{P}} \times \mathbf{P}$ is the least set defined by the rules in Table 4 and closed by $\equiv_{\mathbf{P}}$. The rule (out) states that the process $\langle t \rangle.p$ can broadcast the term t. (in) states that $(x).p$ can receive any term t and let it be substituted for any free occurrence of x in p. The rule (res) is the usual rule for restriction. The rule $(open)$ takes care of extrusion of restricted names.

$$(out)\frac{}{\langle t\rangle.p \xrightarrow{\langle t\rangle} p} \qquad (in)\frac{}{(x).p \xrightarrow{(t)} p\{t/x\}}$$

$$(res)\frac{p \xrightarrow{\lambda} p'}{\nu n.p \xrightarrow{\lambda} \nu n.p'} \quad n \notin fn(\lambda) \cup bn(\lambda) \qquad (open)\frac{p \xrightarrow{\nu\tilde{n}\langle t\rangle} p'}{\nu n.p \xrightarrow{\nu\tilde{n}n\langle t\rangle} p'} \quad n \in fn(t) \setminus \tilde{n}$$

Table 4. Transition Rules, Processes.

4.2 Networks Semantics

The set of *network actions* $\mathcal{A}$ ranged over by α is defined by:

$$\alpha ::= \beta \mid \gamma \qquad \beta ::= \overline{l} \mid \overline{l}\sigma\nu\tilde{n}\langle t\rangle \mid l\overline{\sigma}(t) \mid \tau \qquad \gamma ::= l\triangleright \mid \nu l.l\triangleright \mid l \triangleleft k \mid \tau$$

where $t \in \mathcal{T}$. Actions are grouped into broadcast and mobility actions ranged over by β and γ respectively. The action $\overline{l}$ denotes that the node at location l has completed a broadcast computation. The action $\overline{l}\sigma\nu\tilde{n}\langle t\rangle$ is an output action, it means that the node at location l may broadcast the message t with names in $\tilde{n}$ bound to the nodes with locations in σ. The action $l\overline{\sigma}(t)$ is an input action, meaning that t may be received from the node at location l by the nodes with locations in σ. The action $l\triangleright$ ($\nu l.l\triangleright$) means that the (hidden) node at location l may move. Finally, the action $l \triangleleft k$ indicates that the two nodes at locations l and k respectively are disconnecting. As usual τ denotes an internal computation.

For convenience we write $\nu\tilde{m}.l\triangleright$ for $l\triangleright$ if $\tilde{m} = \emptyset$, likewise if $\tilde{m} = \{l\}$ we write $\nu\tilde{m}.l\triangleright$ for $\nu l.l\triangleright$. We let $bn(\alpha)$ ($fn(\alpha)$) denote the bound (free) names in α, and we let $bl(\alpha)$ ($fl(\alpha)$) denote the bound (free) locations in α.

The operational semantics for networks is defined by a labeled transition system $(\mathbf{N}, \mathcal{A}, \rightarrow)$ where $\rightarrow \subseteq \mathbf{N} \times \mathcal{A} \times \mathbf{N}$ is the least relation satisfying the rules in Table 5 and 6, omitting the symmetric counterparts of the three rules $(synch)$, (par_1), and (par_2).

The rule (brd) in Table 5 states that a node at location l may broadcast its message to any node with location in σ. Rule ($lose$) represents that broadcast messages may be arbitrarily lost, nodes with locations in σ' will not receive the message broadcasted by l. Hence we may have:

$$P_1 = \lfloor\langle n\rangle.p\rfloor_k^{lm} \xrightarrow{\overline{k}lm\langle n\rangle} \lfloor p\rfloor_k^{lm} \quad \text{and} \quad P_1 \xrightarrow{\overline{k}l\langle n\rangle} \lfloor p\rfloor_k^{lm} \ .$$

The two rules (rec_1) and (rec_2) show how broadcasted terms may be received by nodes, e.g. we may have:

$$Q_1 = \lfloor(x).q\rfloor_l^k \parallel \lfloor(x).r\rfloor_m^k \xrightarrow{k\overline{lm}(n)} \lfloor q\{n/x\}\rfloor_l^k \parallel \lfloor r\{n/x\}\rfloor_m^k = Q_2 \ .$$

The actual synchronization between broadcast and reception of messages is shown in ($synch$), for instance:

$$P_1 \parallel Q_1 \xrightarrow{\overline{k}\epsilon\langle n\rangle} \lfloor p\rfloor_k^{lm} \parallel Q_2 \quad \text{and} \quad P_1 \parallel \lfloor(x).r\rfloor_m^k \xrightarrow{\overline{k}l\langle n\rangle} \lfloor p\rfloor_k^{lm} \parallel \lfloor r\{n/x\}\rfloor_m^k \ .$$

The rules ($open_1$) and ($close$) make sure that extrusion of bound names is treated properly, where ($close$) signals the completion of a broadcasting session. As an example of the result of an application of rule ($open_1$) we may take:

$$P_1{}' = \nu n.P_1 \xrightarrow{\overline{k}lm\nu n\langle n\rangle} \lfloor p\rfloor_k^{lm} \ ,$$

and assuming $n \notin fn(Q_1)$, taking care to avoid name clashing in the ($synch$) rule, we may apply the rule ($close$) to obtain:

$$P_1{}' \parallel Q_1 \xrightarrow{\overline{k}} \nu n.(\lfloor p\rfloor_k^{lm} \parallel Q_2) \ .$$

The rule (par_1) is a standard rule for concurrency, say:

$$Q_1 \xrightarrow{k\overline{l}(n)} \lfloor q\{n/x\}\rfloor_l^k \parallel \lfloor(x).r\rfloor_m^k \ ,$$

$$(brd)\frac{p \xrightarrow{\nu\tilde{n}\langle t\rangle} p'}{\lfloor p\rfloor_l^\sigma \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t\rangle} \lfloor p'\rfloor_l^\sigma} \qquad (lose)\frac{P \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}\langle t\rangle} P'}{P \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t\rangle} P'} \qquad (rec_1)\frac{p \xrightarrow{(t)} p'}{\lfloor p\rfloor_l^{\sigma k} \xrightarrow{k\overline{l}(t)} \lfloor p'\rfloor_l^\sigma}$$

$$(rec_2)\frac{P \xrightarrow{l\overline{\sigma}(t)} P' \quad Q \xrightarrow{l\overline{\sigma'}(t)} Q'}{P \parallel Q \xrightarrow{l\overline{\sigma\sigma'}(t)} P' \parallel Q'} \qquad (close)\frac{P \xrightarrow{\overline{l}\epsilon\nu\tilde{n}\langle t\rangle} P'}{P \xrightarrow{\overline{l}} \nu\tilde{n}.P'}$$

$$(hide)\frac{P \xrightarrow{\overline{l}} P'}{\nu l.P \xrightarrow{\tau} \nu l.P'} \qquad (open_1)\frac{P \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t\rangle} P'}{\nu n.P \xrightarrow{\overline{l}\sigma\nu\tilde{n}n\langle t\rangle} P'} \quad n \in fn(t) \setminus \tilde{n}$$

$$(synch)\frac{P \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}\langle t\rangle} P' \qquad Q \xrightarrow{l\overline{\sigma'}(t)} Q'}{P \parallel Q \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t\rangle} P' \parallel Q'} \quad \tilde{n} \cap fn(Q) = \sigma \cap fl(Q) = \emptyset$$

$$(par_1)\frac{P \xrightarrow{\beta} P'}{P \parallel Q \xrightarrow{\beta} P' \parallel Q} \quad fl(\beta) \cap fl(Q) = bn(\beta) \cap fn(Q) = \emptyset$$

$$(res_1)\frac{P \xrightarrow{\beta} P'}{\nu m.P \xrightarrow{\beta} \nu m.P'} \quad m \notin fl(\beta) \cup fn(\beta) \cup bn(\beta)$$

Table 5. Transition Rules, Network Broadcast.

and beyond taking care to avoid name clash it implies for instance:

$$P_1 \parallel \lfloor (x).r\rfloor_m^k \not\xrightarrow{\overline{k}lm\langle n\rangle} \ ,$$

because m is a free location in both $\overline{k}lm\langle n\rangle$ and $\lfloor (x).r\rfloor_m^k$, hence the side condition in (par_1) enforces networks not to externally broadcast messages to nodes it already contains. Likewise, $(synch)$ enforces:

$$P_1 \parallel Q_1 \not\xrightarrow{\overline{k}m\langle n\rangle} \ ,$$

because $m \in fl(Q_1)$.

The rule (res_1) is defined as usual, but $(hide)$ is a new special rule added for the same reason as the rule with the same name in Table 3, i.e. to hide broadcast from hidden nodes. Hence for instance:

$$\nu k.({P_1}' \parallel Q_1) \xrightarrow{\tau} \nu k.\nu n.(\lfloor p\rfloor_k^{lm} \parallel Q_2) \ ,$$

is the result of letting the hidden node at location k in ${P_1}'$ complete a broadcast communication.

Mobility of nodes is obtained through the rules (con_1) and (dis_1) and their respective synchronization rules (con_2) and (dis_2) in Table 6. The rule (con_1) states that a node at a free location l may connect to any other node as demonstrated by the rule (con_2). As an example:

$$\lfloor p\rfloor_l \xrightarrow{l\triangleright} \lfloor p\rfloor_l \ , \quad \lfloor q\rfloor_k \xrightarrow{k\triangleright} \lfloor q\rfloor_k \ , \text{ and } \ \lfloor p\rfloor_l \parallel \lfloor q\rfloor_k \xrightarrow{\tau} \lfloor p\rfloor_l^k \parallel \lfloor q\rfloor_k^l \ .$$

Dually, (dis_1) states that a node at location l with a neighbor at location k may disconnect from k and in so doing remove k from the set of connections of the node. The mutual disconnection of bidirectionally connected nodes is taken care of by the rule (dis_2). For instance,

$$\lfloor p\rfloor_l^k \xrightarrow{l\triangleleft k} \lfloor p\rfloor_l \ , \quad \lfloor q\rfloor_k^l \xrightarrow{k\triangleleft l} \lfloor q\rfloor_k \ , \text{ and } \ \lfloor p\rfloor_l^k \parallel \lfloor q\rfloor_k^l \xrightarrow{\tau} \lfloor p\rfloor_l \parallel \lfloor q\rfloor_k \ .$$

$$(con_1)\frac{}{\lfloor p\rfloor_l^{\sigma} \xrightarrow{l\triangleright} \lfloor p\rfloor_l^{\sigma}} \qquad (dis_1)\frac{}{\lfloor p\rfloor_l^{\sigma k} \xrightarrow{l\triangleleft k} \lfloor p\rfloor_l^{\sigma}}$$

$$(con_2)\frac{P \xrightarrow{\nu\tilde{m}.l\triangleright} P' \qquad Q \xrightarrow{\nu\tilde{m}'.k\triangleright} Q'}{P \parallel Q \xrightarrow{\tau} \nu\tilde{m}'\tilde{m}.(P' \parallel Q')_{l\oplus k}} \quad \tilde{m}\cap\tilde{m}' = \tilde{m}\cap \mathit{flc}(Q) = \tilde{m}'\cap \mathit{flc}(P) = \emptyset$$

$$(dis_2)\frac{P \xrightarrow{l\triangleleft k} P' \qquad Q \xrightarrow{k\triangleleft l} Q'}{P \parallel Q \xrightarrow{\tau} P' \parallel Q'}$$

$$(open_2)\frac{P \xrightarrow{l\triangleright} P'}{\nu l.P \xrightarrow{\nu l.l\triangleright} P'} \qquad (res_2)\frac{P \xrightarrow{\gamma} P'}{\nu m.P \xrightarrow{\gamma} \nu m.P'} \quad m \notin \mathit{fl}(\gamma)\cup \mathit{bl}(\gamma)$$

$$(par_2)\frac{P \xrightarrow{\gamma} P'}{P \parallel Q \xrightarrow{\gamma} P' \parallel Q} \quad \mathit{bl}(\gamma)\cap \mathit{flc}(Q) = \emptyset$$

Table 6. Transition Rules, Network Mobility.

Special care must be given to hidden nodes. The rules ($open_2$), (par_2), and (con_2) allow the location names for hidden nodes to be properly extruded, in particular taking care to avoid clashes between bound location names and free locations and connections. As an example, assuming $l \neq k$,

$$\nu l.\lfloor p\rfloor_l \xrightarrow{\nu l.l\triangleright} \lfloor p\rfloor_l \quad \text{and} \quad \nu l.\lfloor p\rfloor_l \parallel \lfloor q\rfloor_k \xrightarrow{\tau} \nu l.(\lfloor p\rfloor_l^k \parallel \lfloor q\rfloor_k^l) \ .$$

Illustrating the use of (par_2) we may have:

$$\nu l.\lfloor p\rfloor_l \parallel \lfloor q\rfloor_k \xrightarrow{\nu l.l\triangleright} \lfloor p\rfloor_l \parallel \lfloor q\rfloor_k \ ,$$

and from (con_2) we may then get if $m \notin \{l,k\}$,

$$\nu l.\lfloor p\rfloor_l \parallel \lfloor q\rfloor_k \parallel \nu m.\lfloor r\rfloor_m \xrightarrow{\tau} \nu m.\nu l.(\lfloor p\rfloor_l^m \parallel \lfloor q\rfloor_k \parallel \lfloor r\rfloor_m^l) \ .$$

The rule (res_2) is defined as usual.

The close correspondence between the reduction semantics and the labeled transition system semantics is demonstrated by the lemmas below.

Lemma 1. $P \xrightarrow{\tau}\equiv P'$ *iff* $P \searrow P'$.

Lemma 2. $P \xrightarrow{\bar{l}}\equiv P'$ *iff* $P \searrow_l P'$.

4.3 Bisimulation Semantics

Below we give co-induction characterizations, a strong and a weak bisimulation, of the strong and weak reduction congruences respectively. Our characterizations follows the contextual style as found in e.g. [4, 14].

To assist in the definitions below we introduce a shorthand, $(x)A_{\sigma\oplus l}$, for a family of variable closed networks defined by the grammar:

$$(x)A_{\sigma\oplus l} ::= \Pi_{m\in\sigma}\lfloor (x).p_m\rfloor_m^{\sigma_m l}$$

For any $(x)A_{\sigma\oplus l}$ with $(x)A_{\sigma\oplus l} = \Pi_{m\in\sigma}\lfloor (x).p_m\rfloor_m^{\sigma_m l}$ we write $A_{\sigma\oplus l}\{t/x\}$ for the network $\Pi_{m\in\sigma}\lfloor p_m\{t/x\}\rfloor_m^{\sigma_m l}$.

Strong bisimulation is defined as follows.

Definition 5. *A binary relation $\mathcal{R}$ on* $\mathbf{N}$ *is a* strong simulation *if $P \mathcal{R} Q$ implies $fl(P) = fl(Q)$ and for all $p \in \mathbf{P}$,*

1. *if* $P \xrightarrow{\tau} P'$ *then* $\exists Q'.\, Q \xrightarrow{\tau} Q'$ *and* $P' \mathcal{R} Q'$
2. *if* $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ *then* $\forall \sigma'.\, \sigma' \subseteq \sigma.\, \forall (x)A_{\sigma'\oplus l}.\, \tilde{n} \cap fn((x)A_{\sigma'\oplus l}) = \emptyset.\, \exists Q'.$
   $Q \parallel (x)A_{\sigma'\oplus l} \xrightarrow{\bar{l}} Q'$ *and* $\nu\tilde{n}.(P' \parallel A_{\sigma'\oplus l}\{t/x\}) \mathcal{R} Q'$
3. *if* $P \xrightarrow{l\overline{\sigma}(t)} P'$ *then* $\forall \sigma'.\, \sigma' \cap \sigma l = \emptyset,\, \exists Q'.$
   $Q \parallel \lfloor \langle t\rangle.p \rfloor_l^{\sigma\sigma'} \xrightarrow{\bar{l}} Q'$ *and* $P' \parallel \lfloor p \rfloor_l^{\sigma\sigma'} \mathcal{R} Q'$
4. *if* $P \xrightarrow{\nu\tilde{m}.l\triangleright} P'$ *then* $\forall\, k.\, k \notin fl(P) \cup \tilde{m}.\, \forall \sigma.\, \sigma \cap \tilde{m}k = \emptyset.\, \exists Q'.$
   $Q \parallel \lfloor p \rfloor_k^{\sigma} \xrightarrow{\tau} Q'$ *and* $\nu\tilde{m}.(P' \parallel \lfloor p \rfloor_k^{\sigma})_{l\oplus k} \mathcal{R} Q'$
5. *if* $P \xrightarrow{l\triangleleft k} P'$ *then* $\forall \sigma.\, k \notin \sigma.\, \exists Q'.$
   $Q \parallel \lfloor p \rfloor_k^{\sigma l} \xrightarrow{\tau} Q'$ *and* $P' \parallel \lfloor p \rfloor_k^{\sigma} \mathcal{R} Q'$

$\mathcal{R}$ is a strong bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are strong simulations.

Let $\sim$ be the largest strong bisimulation.

The notion of strong bisimulation in a broadcasting framework as defined by Definition 5 is a key contribution of this paper and deserves some comments. Requirement 1 in the definition is standard. Requirement 2 demands that an open broadcast communication to nodes at locations σ in the environment by a node at some visible location l in P must be matched by a completed broadcast communication by a node at the same location l in Q, when Q is put in parallel with any potential receivers. Requirement 3 states that if nodes at locations σ in P may receive t from a broadcasting node at location l in the environment, then Q composed with any such node may let the node emit t and complete a broadcast communication with Q, and in so doing Q and the node together become a network that can match the reception of t by the nodes at σ in P. Requirement 4 states that if a (possibly hidden) node in P (bidirectionally) connects to an external node at some fresh location k then Q an the new external node can make an internal computation and then match P being connected to the node at location k. Finally, requirement 5 demands that if the node at location l in P is about to disconnect from location k in its environment, then Q in an environment with a single node at location k that is connected to l can make an internal computation, and match P and the node at location k together in parallel when the two are disconnected.

Notice that all but the first requirements in Definition 5 are contextual because they are demands on the network execution environment to receive broadcasted messages, to provide external input of data terms, to connect with new fresh localities, and to disconnect from environmental locations respectively.

Theorem 1. *$\sim$ is a congruence.*

Let $\stackrel{\tau}{\Longrightarrow}$ be the reflexive and transitive closure of $\xrightarrow{\tau}$ and define $\stackrel{\bar{l}}{\Longrightarrow}$ by $\stackrel{\tau}{\Longrightarrow}\xrightarrow{\bar{l}}\stackrel{\tau}{\Longrightarrow}$.

Weak bisimulation is defined as below.

Definition 6. *A binary relation $\mathcal{R}$ on* $\mathbf{N}$ *is a* weak simulation *if $P \mathcal{R} Q$ implies $fl(P) = fl(Q)$ and for all $p \in \mathbf{P}$,*

1. *if* $P \xrightarrow{\tau} P'$ *then* $\exists Q'.\, Q \stackrel{\tau}{\Longrightarrow} Q'$ *and* $P' \mathcal{R} Q'$
2. *if* $P \xrightarrow{\bar{l}} P'$ *then* $\exists Q'.\, Q \stackrel{\bar{l}}{\Longrightarrow} Q'$ *and* $P' \mathcal{R} Q'$
3. *if* $P \xrightarrow{l\overline{\sigma}(t)} P'$ *then* $\forall \sigma'.\, \sigma' \cap \sigma l = \emptyset,\, \exists Q'.$
   $Q \parallel \lfloor \langle t\rangle.p \rfloor_l^{\sigma\sigma'} \stackrel{\bar{l}}{\Longrightarrow} Q'$ *and* $P' \parallel \lfloor p \rfloor_l^{\sigma\sigma'} \mathcal{R} Q'$
4. *if* $P \xrightarrow{\nu\tilde{m}.l\triangleright} P'$ *then* $\forall\, k.\, k \notin fl(P) \cup \tilde{m}.\, \forall \sigma.\, \sigma \cap \tilde{m}k = \emptyset.\, \exists Q'.$
   $Q \parallel \lfloor p \rfloor_k^{\sigma} \stackrel{\tau}{\Longrightarrow} Q'$ *and* $\nu\tilde{m}.(P' \parallel \lfloor p \rfloor_k^{\sigma})_{l\oplus k} \mathcal{R} Q'$
5. *if* $P \xrightarrow{l\triangleleft k} P'$ *then* $\forall \sigma.\, k \notin \sigma.\, \exists Q'.$
   $Q \parallel \lfloor p \rfloor_k^{\sigma l} \stackrel{\tau}{\Longrightarrow} Q'$ *and* $P' \parallel \lfloor p \rfloor_k^{\sigma} \mathcal{R} Q'$

$\mathcal{R}$ is a weak bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are weak simulations.

Let $\approx$ be the largest weak bisimulation.

The requirements in Definition 6 are obvious weak generalizations of the requirements in Definition 5. However, one exception being Requirement 2 since it only considers completed broadcast transitions and not messages broadcasted to the environment as in e.g.

$$2.\ \textit{if}\ P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'\ \textit{then}\ \forall\sigma'.\,\sigma' \subseteq \sigma.\,\forall(x)A_{\sigma'\oplus l}.\,\tilde{n} \cap \mathit{fn}((x)A_{\sigma'\oplus l}) = \emptyset.\,\exists Q'.$$
$$Q \parallel (x)A_{\sigma'\oplus l} \stackrel{\bar{l}}{\Longrightarrow} Q' \text{ and } \nu\tilde{n}.(P' \parallel A_{\sigma'\oplus l}\{t/x\})\ \mathcal{R}\ Q'$$

In the appendix we show that Requirement 2 in Definition 6 can be interchanged with the stronger requirement above leading to an equivalent definition of weak bisimulation.

Theorem 2. *$\approx$ is a congruence.*

Because $\sim$ ($\approx$) is a congruence it is sufficient to establish that a strong (weak) bisimulation is strong (weak) reduction closed in order to show $\sim \subseteq \simeq$ ($\approx \subseteq \cong$), this follows from Lemma 1 and 2. Then in order to show $\sim = \simeq$ ($\approx = \cong$) it just remains showing $\simeq$ ($\cong$) to be a strong (weak) bisimulation. For details we refer the reader to the appendix.

Theorem 3. $\sim = \simeq$.

Theorem 4. $\approx = \cong$.

Because the establishment of bidirectional connections and deconnections are unobservable in a weak bisimulation semantics the following lemma holds:

Lemma 3. *If $l, k \in \mathit{fl}(P)$ then $P_{l\oplus k} \approx P_{l\ominus k}$.*

It is not difficult to show that $\equiv$ is a weak bisimulation, and as an example we may show that the inactive network is weak bisimilar to a hidden node with an inactive process, i.e. $0 \approx \nu k.\lfloor 0\rfloor_k$, because $\mathcal{R} \cup \mathcal{R}^{-1}$ is a weak bisimulation up to $\equiv$ where

$$\mathcal{R} = \{(\nu\tilde{m}.(0 \parallel P), \nu\tilde{m}k.(\lfloor 0\rfloor_k \parallel P)_{\sigma\oplus k}) \mid \tilde{m} \cup \sigma \subseteq \mathit{fl}(P),\ k \notin \mathit{fl}(P)\}$$

letting $P_{\sigma\oplus k}$ be defined by $(\ldots(P_{l_1\oplus k})\ldots)_{l_i\oplus k}$ whenever $\sigma = \{l_1, \ldots, l_i\}$.

5 Connection Closed Networks

The definitions of strong and weak bisimulation are contextual and therefore it is hard to prove bisimulation equivalence between networks. For the class of *connection closed* networks however it turns out that our framework becomes significantly simpler.

We say that a network P is connection closed if each node in P is connected only to other nodes within P, i.e. if $\mathit{fc}(P) \subseteq \mathit{fl}(P)$. For instance, all networks in the examples (1) – (7) in the Introduction are connection closed, but $\lfloor 0\rfloor_l^m$ is not. We let $\mathbf{N}_c$ denote the subset of $\mathbf{N}$ of connection closed networks.

Let a binary relation on $\mathbf{N}_c$ be a *c-congruence* if it is closed by hiding and by (well-formed) parallel composition of networks in $\mathbf{N}_c$, and let structural c-congruence be the least c-congruence and equivalence relation on $\mathbf{N}_c$ closed under α-conversion and the rules in Table 2.

Similar to Definition 3 we define a strong congruence over connection closed networks.

Definition 7. *A symmetric relation $\mathcal{R}$ on $\mathbf{N}_c$ is a* strong reduction c-congruence *if it is strong reduction closed, a c-congruence, and if $P\ \mathcal{R}\ Q$ implies $\mathit{fl}(P) = \mathit{fl}(Q)$.*

We let $\simeq_c$ be the largest strong reduction c-congruence.

As in Definition 4 we define a weak congruence abstracting from internal computation, but now only over connection closed networks.

Definition 8. *A symmetric relation $\mathcal{R}$ on $\mathbf{N}_c$ is a* weak reduction c-congruence *if it is weak reduction closed, a c-congruence, and if $P\ \mathcal{R}\ Q$ implies $\mathit{fl}(P) = \mathit{fl}(Q)$.*

We let $\cong_c$ be the largest weak reduction c-congruence.

Like strong and weak reduction congruences was characterized by strong and weak bisimulation respectively we may also characterize strong and weak reduction c-congruence by a co-inductively defined bisimulation.

Let R_k range over networks in $\mathbf{N}_c$ where $k \in fl(R_k)$.

Definition 9. *A binary relation $\mathcal{R}$ on $\mathbf{N}_c$ is a* strong c-simulation *if $P \mathcal{R} Q$ implies $fl(P) = fl(Q)$ and*

$$\begin{array}{l} \textit{if } \ P \xrightarrow{\tau} P' \textit{ then } \ \exists Q'.\, Q \xrightarrow{\tau} Q' \textit{ and } P' \mathcal{R} Q' \\ \textit{if } \ P \xrightarrow{\bar{l}} P' \textit{ then } \ \exists Q'.\, Q \xrightarrow{\bar{l}} Q' \textit{ and } P' \mathcal{R} Q' \\ \textit{if } \ P \xrightarrow{\nu\tilde{m}.l\triangleright} P' \textit{ then } \ \forall R_k \in \mathbf{N}_c. fl(R_k) \cap (fl(P) \cup \tilde{m}) = \emptyset.\ \exists Q'. \\ \qquad Q \parallel R_k \xrightarrow{\tau} Q' \textit{ and } \ \nu\tilde{m}.(P' \parallel R_k)_{l\oplus k} \mathcal{R} Q' \end{array}$$

$\mathcal{R}$ is a strong c-bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are strong c-simulations.

Let $\sim_c$ be the largest strong c-bisimulation.

Definition 10. *A binary relation $\mathcal{R}$ on $\mathbf{N}_c$ is a* weak c-simulation *if $P \mathcal{R} Q$ implies $fl(P) = fl(Q)$ and*

$$\begin{array}{l} \textit{if } \ P \xrightarrow{\tau} P' \textit{ then } \ \exists Q'.\, Q \stackrel{\tau}{\Longrightarrow} Q' \textit{ and } P' \mathcal{R} Q' \\ \textit{if } \ P \xrightarrow{\bar{l}} P' \textit{ then } \ \exists Q'.\, Q \stackrel{\bar{l}}{\Longrightarrow} Q' \textit{ and } P' \mathcal{R} Q' \\ \textit{if } \ P \xrightarrow{\nu\tilde{m}.l\triangleright} P' \textit{ then } \ \forall R_k \in \mathbf{N}_c. fl(R_k) \cap (fl(P) \cup \tilde{m}) = \emptyset.\ \exists Q'. \\ \qquad Q \parallel R_k \stackrel{\tau}{\Longrightarrow} Q' \textit{ and } \ \nu\tilde{m}.(P' \parallel R_k)_{l\oplus k} \mathcal{R} Q' \end{array}$$

$\mathcal{R}$ is a weak c-bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are weak c-simulations.

Let $\approx_c$ be the largest weak c-bisimulation.

One may show that $\sim_c$ and $\approx_c$ are c-congruences and that

Theorem 5. $\simeq_c = \sim_c$

Theorem 6. $\cong_c = \approx_c$

As an example, we may then (writing $\langle t \rangle$ for $\langle t \rangle.0$) show $\nu k.\lfloor \langle n \rangle.\langle n \rangle \rfloor_k \approx_c \nu k.\lfloor \langle n \rangle \rfloor_k \parallel \nu l.\lfloor \langle n \rangle \rfloor_l$ because

$$\begin{array}{l} \{\ (\nu\tilde{m}k.(\lfloor \langle n \rangle.\langle n \rangle \rfloor_k \parallel Q)_{\sigma\oplus k}, \nu\tilde{m}kl.(\lfloor \langle n \rangle \rfloor_k \parallel (\lfloor \langle n \rangle \rfloor_l \parallel Q)_{\sigma_1\oplus l})_{\sigma_2\oplus k}), \\ \quad (\nu\tilde{m}k.(\lfloor \langle n \rangle \rfloor_k \parallel Q)_{\sigma\oplus k}, \nu\tilde{m}kl.(\lfloor \langle n \rangle \rfloor_k \parallel (\lfloor 0 \rfloor_l \parallel Q)_{\sigma_1\oplus l})_{\sigma_2\oplus k}), \\ \quad (\nu\tilde{m}k.(\lfloor 0 \rfloor_k \parallel Q)_{\sigma\oplus k}, \nu\tilde{m}kl.(\lfloor 0 \rfloor_k \parallel (\lfloor 0 \rfloor_l \parallel Q)_{\sigma_1\oplus l})_{\sigma_2\oplus k}) \\ \quad \mid \sigma \cup \sigma_1 \cup \sigma_2 \subseteq fl(Q)\ \} \end{array}$$

is a weak c-bisimulation up to structural c-congruence.

6 An Example: ARAN

As mentioned in the Introduction a key motivation for our work is to establish a framework that allows to reason about security properties for MANETS. In [6] an attack on the cryptographic routing protocol ARAN [18] was identified and below we recapture the principles of this attack.

The goal of ARAN is to ensure secure requests for routing in ad hoc networks by making requests and replies be signed and checked in every hop, hence messages cannot be altered and therefore the protocol is claimed to be safe in that no false routing information can be imposed by malicious nodes. The basic idea of the protocol is that a receiver of a message is obliged to check its signature and if the message is correctly signed the signature is removed and signed

$$
\begin{array}{l}
p_0 \stackrel{\text{def}}{=} \nu n_1.let\ x_{cert} = sign(pk(n_1), sk(n_0))\ in\ p_1 \\
p_1 \stackrel{\text{def}}{=} let\ x_{sreq} = sign(rdp, sk(n_1))\ in\ \langle (x_{sreq}, x_{cert})\rangle.(x).p_2 \\
p_2 \stackrel{\text{def}}{=} let\ x_1 = fst(x)\ in\ \ let\ x_2 = snd(x)\ in\ \ let\ x_3 = get(x_1)\ in\ p_3 \\
p_3 \stackrel{\text{def}}{=} if\ rep = x_3\ then\ let\ x_{key} = get(x_2)\ in\ p_4 \\
p_4 \stackrel{\text{def}}{=} let\ x_4 = check(x_{key}, x_2, pk(n_0))\ in\ if\ x_4 = ok\ \ then\ p_5 \\
p_5 \stackrel{\text{def}}{=} let\ x_5 = check(rep, x_1, x_{key})\ in\ if\ x_5 = ok\ \ then\ \langle success\rangle \\
\\
q_0 \stackrel{\text{def}}{=} \nu n_3.let\ x_{cert} = sign(pk(n_3), sk(n_0))\ in\ (x).q_1 \\
q_1 \stackrel{\text{def}}{=} let\ x_1 = fst(x)\ in\ \ let\ x_2 = snd(x)\ in\ \ q_2 \\
q_2 \stackrel{\text{def}}{=} let\ x_3 = get(x_1)\ in\ \ if\ rdp = x_3\ then\ let\ x_{key} = get(x_2)\ in\ q_3 \\
q_3 \stackrel{\text{def}}{=} let\ x_4 = check(x_{key}, x_2, pk(n_0))\ in\ if\ x_4 = ok\ then\ \langle (sign(rep, sk(n_3)), x_{cert})\rangle
\end{array}
$$

Table 7. ARAN processes.

by the node itself before the new message is forwarded. It is assumed that all valid nodes in the network a priori have a private public key pair and a certificate and also that the public key of the certificate authority is known to every node.

In order to illustrate the attack it is sufficient to consider only a network consisting of three nodes: the initiator of a route request, the destination of the request, and an attacker. The attacker is not a valid node and hence it has not been authorized by the certificate authority.

The simplified ARAN protocol we consider goes as follows: The initiator broadcasts a signed request rdp to its neighbors and awaits a signed reply rep in return, if the reply is successfully returned the initiator broadcasts $success$. Hence the destination must be an immediate neighbor in order for a route to exist. The destination of the route request on the other hand waits for a signed route request, checks that it is properly signed and if so returns a signed reply to the initiator. Upon reception of the reply the initiator validates the signed message.

To model the cryptographic primitives, let $\{ok, pk, sk, sign\}$ be a set of constructor symbols and let $\{check, get\}$ be a set of destructor symbols where ok has arity 0, where get, pk, and sk have arity 1, where $sign$ has arity 2, and where $check$ has arity 3. We let $pk(n)$ be the constructor for a public key based on some seed n, and we let $sk(m)$ be a private (secret) key based on the seed m. The application of the constructor $sign$

$$sign(pk(n), sk(m)) \quad ,$$

then denotes the signing of the public key $pk(n)$ with the secret key $sk(m)$. We let the destructors $check$ and get be defined by:

$$check(t, sign(t, sk(s)), pk(s)) = ok \quad , \quad get(sign(t, sk(s))) = t \quad .$$

That is, checking the signature of a message t with the public key matching the private key by which the message was signed yields the result ok. The destructor get simply returns the contents of a signed message. By convention we introduce two auxiliary destructors, fst and snd, that returns the first and second element of a pair respectively.

As shorthands for the process expressions, whenever q is 0, we abbreviate $if\ \ t = s\ \ then\ \ p\ \ else\ \ q$ by $if\ \ t = s\ \ then\ \ p$, we write $let\ \ x = g(t_1, \ldots, t_k)\ \ in\ \ p$ instead of $let\ \ x = g(t_1, \ldots, t_k)\ \ in\ \ p\ \ else\ \ q$, and also, as before we write $\langle t\rangle$ for $\langle t\rangle.q$.

The simplified one shot version of the ARAN protocol is defined by:

$$A = \nu n_0.(\lfloor p_0 \rfloor_l \parallel \lfloor q_0 \rfloor_k) \ ,$$

where p_0 and q_0 are defined in Table 7. The process p_0 defines the behaviour of the initiator of the protocol, and q_0 defines the behaviour of the destination.

The intruder, which in this example can only relay messages, is defined as a hidden node by:

$$I = \nu m.\lfloor rec\ z.(x)\langle x\rangle.z \rfloor_m \ . \tag{8}$$

Observe, that since the intruder is a hidden node broadcasting of messages from I cannot be observed.

A correctness criterion for the ARAN protocol is as stated above that the routing messages must be validated in each and every hop, in that each hop should always be between certified nodes only. For instance, it must not be possible for a non-certified node (an intruder) to be part of a valid route in ARAN. This criterion may be checked by verifying as to whether the protocol is unaffected by running together with an intruder doing relays as defined by (8).

The composition of A and the intruder can do the following computation:

$$A \parallel I \stackrel{\bar{l}}{\Longrightarrow} \nu n_0.\nu n_1.\nu m.(\lfloor (x).p_2 \rfloor_l^m \parallel \lfloor q_0 \rfloor_k \parallel \lfloor \langle t \rangle.rec\ z.(x)\langle x \rangle.z \rfloor_m^l) = P \ , \tag{9}$$

where

$$t = (sign(rdp, sk(n_1)), sign(pk(n_1), sk(n_0))) \ .$$

We argue $A \not\approx A \parallel I$, and hence demonstrate that the simple version of the ARAN protocol is not robust and therefore subject of attack from an intruder doing relays. A can match the weak output transition (9) above by the four moves:

$$A \stackrel{\bar{l}}{\Longrightarrow} \nu n_0.\nu n_1.(\lfloor (x).p_2 \rfloor_l \parallel \lfloor q_0 \rfloor_k) = Q \ ,$$

$$A \stackrel{\bar{l}}{\Longrightarrow} Q_{l \oplus k} \ ,$$

$$A \stackrel{\bar{l}}{\Longrightarrow} \nu n_0.\nu n_1.(\lfloor (x).p_2 \rfloor_l \parallel \lfloor \nu n_3.\langle t' \rangle \rfloor_k) = Q' \ ,$$

$$A \stackrel{\bar{l}}{\Longrightarrow} Q'_{l \oplus k} \ ,$$

where $t' = (sign(rdp, sk(n_3)), sign(pk(n_3), sk(n_0)))$.

Clearly $P \not\approx Q$ because $P \stackrel{\bar{k}}{\Longrightarrow}$ which cannot be matched by Q. Notice that the in-equivalence follows due to computations by P where the intruder is part of a route from the initiator to the destination whereas Q is a state where the request has been lost. Because $P \not\approx Q$ also $P \not\approx Q_{l \oplus k}$ due to Lemma 3 since $Q = Q_{l \ominus k}$.

The final part of the proof is due to the fact that the state P where the intruder got the request can be followed by a computation in which the message is lost when the intruder performs a (hidden) broadcast, [4] i.e.

$$P \stackrel{\tau}{\longrightarrow} \nu n_0.\nu n_1.(\nu m.(\lfloor (x).p_2 \rfloor_l^m \parallel \lfloor q_0 \rfloor_k \parallel \lfloor rec\ z.(x)\langle x \rangle.z \rfloor_m^l)) = P' \ .$$

Then, since in Q' the destination cannot escape being able to broadcast, because for all $R \in \{R \mid Q' \stackrel{\tau}{\Longrightarrow} R\} = \{Q', Q'_{l \oplus k}\}$ it holds that $R \stackrel{\bar{k}}{\longrightarrow}$, and since $P' \stackrel{\bar{k}}{\not\Longrightarrow}$ it turns out that $P \not\approx Q'$. It then follows from Lemma 3 that also $P \not\approx Q'_{l \oplus k}$ because $Q' = Q'_{l \ominus k}$.

7 Conclusion

We have defined a broadcasting calculus, CMAN, for MANETS that supports synchronous spatially oriented broadcast and dynamic changes of the network topology. CMAN is equipped with a natural reduction semantics and congruence, and a co-inductive sound and complete bisimulation characterization. The characterization is shown to be particularly simple for connection closed networks. CMAN has been applied on a small example of a cryptographic routing protocol. A major advantage of CMAN is that it permits direct description of features of MANETS that would be hard do describe in classical calculi.

In the future the process language of CMAN should be extended with concurrency, and we consider also extending the network language with a replication like construct that allows to reason about infinitely many (copies of) instances of nodes. Also, it would be of interest to understand how the semantics should be altered to cater for unidirectional communication links.

As of now nodes are allowed to move around arbitrarily connecting to any other node, however that freedom may seem to be too liberal for many applications, and hence the mobility capabilities may be restricted in our future work by imposing more structure on the networks.

Finally, a challenging topic would be to continue the work of how to formalize and reason about security properties for MANETS, and in particular to investigate to what extent the current behavioural equivalences are sufficient to cater for more extensive security analysis.

[4] Alternatively the intruder could disconnect from the initiator and then make the broadcast to an empty set of receivers.

Acknowledgments Thanks to the anonymous reviewers for valuable comments on earlier versions of this paper.

References

1. Martín Abadi and Bruno Blanchet. Analyzing Security Protocols with Secrecy Types and Logic Programs. *Journal of the ACM*, 52(1):102–146, January 2005.
2. Martin Abadi and Cedric Fournet. Mobile vales, new names, and secure communication. In Hanne Riis Nielson, editor, *28th ACM Symposium on Principles of Programming Languages*, pages 104–115, London, UK, January 2001. ACM.
3. Luca Cardelli and Andrew D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures: First International Conference, FOSSACS '98*. Springer-Verlag, Berlin Germany, 1998.
4. Giuseppe Castagna, Jan Vitek, and Francesco Zappa Nardelli. The seal calculus. *Information and Computation*, 201(1):1–51, 2005.
5. C. Ene and T. Muntean. A broadcast-based calculus for communicating systems. In *6th International Workshop on formal Methods for Parallel Programming: Theory and Applications*, San Francisco, 2001.
6. Jens Chr. Godskesen. Formal verification of the ARAN protocol using the applied π-calculus. In *Proceedings of Sixth International IFIP WG 1.7 Workshop on Issuses in the Theory of Security, (WITS)*, pages 99–113, Vienna, Austria, March 2006.
7. Jens Chr. Godskesen and Thomas Hildebrandt. Extending Howe's method to early bisimulations for typed mobile embedded resources with local names. In *Proceedings of FSTTCS'2005*, volume 3821, pages 140–151, Hyderabad, India, December 2005.
8. Massimo Merro. An observational theory for mobile ad hoc networks. *Electron. Notes Theor. Comput. Sci.*, 173:275–293, 2007.
9. Nicola Mezzetti and Davide Sangiorgi. Towards a calculus for wireless systems. *Electr. Notes Theor. Comput. Sci.*, 158:331–353, 2006.
10. Robin Milner. Functions as processes. In *Proceedings of the seventeenth international colloquium on Automata, languages and programming*, pages 167–180, New York, NY, USA, 1990. Springer-Verlag New York, Inc.
11. Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 100:1–77, September 1992.
12. Robin Milner and Davide Sangiorgi. Barbed bisimulation. In *Proceedings ICALP '92*, volume 623, pages 685–695, Vienna, 1992. Springer-Verlag.
13. Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. *Theor. Comput. Sci.*, 367(1):203–227, 2006.
14. R. De Nicola, D. Gorla, and R. Pugliese. Basic observables for a calculus for global computing. In *Proceedings of ICALP'05*, volume 3580 of *Lecture Notes in Computer Science*, pages 1226–1238. Springer, 2005.
15. Karol Ostrovsky, K. V. S. Prasad, and Walid Taha. Towards a primitive higher order calculus of broadcasting systems. In *PPDP '02: Proceedings of the 4th ACM SIGPLAN international conference on Principles and practice of declarative programming*, pages 2–13, New York, NY, USA, 2002. ACM Press.
16. K. V. S. Prasad. A calculus of broadcasting systems. *Sci. Comput. Program.*, 25(2-3):285–327, 1995.
17. James Riely and Matthew Hennessy. A typed language for distributed mobile processes (extended abstract). In *Conference Record of POPL '98: The 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 378–390, San Diego, California, 1998.
18. Kimaya Sanzgiri, Daniel LaFlamme, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. Authenticated routing for ad hoc networks. *IEEE Journal on Selected Areas in Communication, special issue on Wireless Ad hoc Networks*, 23(3):598–610, March 2005.

A Appendix

This appendix contains the proofs of the Theorems and Lemmas of our theory.

A.1 Proof of Lemma 1 and Lemma 2

Below is a series of lemmas that show how the reduction and the labeled transition system semantics relate.

Lemma 1 follows from Lemma 15 and 16, and Lemma 2 follows from Lemma 13 and 14.

Lemma 4. *For any process $p \in \mathbf{P}$, $p \equiv_{\mathcal{P}} \nu\tilde{n}.q$ where $q = 0$, $q = (x).q'$, or $q = \langle t \rangle.q'$ for some $\tilde{n}$, q', x, and t.*

Proof By induction on the structure of $p \in \mathbf{P}$. □

Lemma 5. *$p \xrightarrow{\nu\tilde{n}\langle t \rangle} p'$ iff $p \equiv_{\mathbf{P}} \nu\tilde{n}\tilde{n}'.\langle t \rangle.q$ and $p' \equiv_{\mathbf{P}} \nu\tilde{n}'.\langle t \rangle.q$ for some q and $\tilde{n}'$ with $\tilde{n} \subseteq fn(t)$ and $\tilde{n}' \cap fn(t) = \emptyset$.*

Proof The 'only if' direction follows by induction in the derivation of $p \xrightarrow{\nu\tilde{n}\langle t \rangle} p'$, and the 'if' direction follows because $\nu\tilde{n}\tilde{n}'.\langle t \rangle.q \xrightarrow{\nu\tilde{n}\langle t \rangle} \nu\tilde{n}'.\langle t \rangle.q$ and since $\xrightarrow{\lambda}$ is closed by $\equiv_{\mathbf{P}}$. □

Lemma 6. *$p \xrightarrow{(t)} p'$ iff $p \equiv_{\mathbf{P}} \nu\tilde{n}.(x).q$ and $p' \equiv_{\mathbf{P}} q\{t/x\}$ for some $\tilde{n}$ where $\tilde{n} \cap fn(t) = \emptyset$.*

Proof The 'only if' direction follows by induction in the derivation of $p \xrightarrow{(x)} p'$, and the 'if' direction follows because $\xrightarrow{\lambda}$ is closed by $\equiv_{\mathbf{P}}$ and because $\nu\tilde{n}.(x).q \xrightarrow{(t)} q\{t/x\}$ when $\tilde{n} \cap fn(t) = \emptyset$. □

Lemma 7. *If $P \xrightarrow{\alpha} P'$ and $P \equiv Q$ then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' \equiv Q'$.*

Proof Suppose $P \equiv Q$. We must show the property

$$P \xrightarrow{\alpha} P' \text{ implies } \exists Q'.\ Q \xrightarrow{\alpha} Q' \text{ and } P' \equiv Q' \tag{10}$$

It's obvious that (10) is preserved by α-conversion and also by reflexivity, symmetry, and transitivity (recall $\equiv$ is closed by α-conversion and it is an equivalence relation).

One may show by induction in the depth of the inference of $P \xrightarrow{\alpha} P'$ that (10) is closed by parallel composition and by restriction (recall $\equiv$ is defined to be a congruence).

Finally we show (10) is closed by the rules in Table 2, also by induction in the depth of the inference of $P \xrightarrow{\alpha} P'$. □

From Lemma 7 it is immediate that:

Corollary 1. *$\equiv$ is a strong bisimulation.*

Lemma 8. *$P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t \rangle} \equiv P'$ iff*

$$P \equiv \nu\tilde{m}.(\lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'\sigma''} \parallel (x)A_{\sigma'\oplus l} \parallel Q)$$

and

$$P' \equiv \nu\tilde{m}'.(\lfloor p \rfloor_l^{\sigma\sigma'\sigma''} \parallel A_{\sigma'\oplus l}\{t/x\} \parallel Q)$$

for some $\tilde{m}$, p, σ', σ'', $(x)A_{\sigma'\oplus l}$, and Q where $\tilde{n} = fn(t) \cap \tilde{m}$, $\tilde{m}' = \tilde{m} \setminus \tilde{n}$, $\tilde{m} \cap \sigma l = \emptyset$, and $\sigma \cap fl(P) = \emptyset$.

Proof: The 'only if' direction follows by induction in the inference of $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t \rangle} P'$, Lemma 5 is used in the base case. The 'if' direction follows from Lemma 7 since

$$\nu\tilde{m}.(\lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'\sigma''} \parallel (x)A_{\sigma'\oplus l} \parallel Q) \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t \rangle} \nu\tilde{m}'.(\lfloor p \rfloor_l^{\sigma\sigma'\sigma''} \parallel A_{\sigma'\oplus l}\{t/x\} \parallel Q)$$

□

Lemma 9. *$P \xrightarrow{l\overline{\sigma}(t)}\equiv P'$ iff $P \equiv \nu\tilde{m}.((x)A_{\sigma\oplus l} \parallel Q)$ and $P' \equiv \nu\tilde{m}.(A_{\sigma\oplus l}\{t/x\} \parallel Q)$ for some $\tilde{m}$, $(x)A_{\sigma\oplus l}$, and Q where $\sigma l \cap \tilde{m} = \emptyset$ and $l \notin fl(P)$.*

Proof: Similar to the proof of Lemma 8. □

Lemma 10. *$P \xrightarrow{l\triangleright} P'$ iff $P = P'$ and $P \equiv \nu\tilde{m}.(\lfloor p \rfloor_l^\sigma \parallel Q)$ for some $\tilde{m}$, p, σ, and Q where $l \notin \tilde{m}$.*

Proof Similar to the proof of Lemma 8, but makes use of Lemma 6 instead of Lemma 5. □

Lemma 11. *$P \xrightarrow{\nu l.l\triangleright}\equiv P'$ iff $P \equiv \nu l.P'$ and $P' \equiv \nu\tilde{m}.(\lfloor p \rfloor_l^\sigma \parallel Q)$ for some $\tilde{m}$, p, σ, and Q where $l \notin \tilde{m}$.*

Proof: Similar to the proof of Lemma 8. □

Lemma 12. *$P \xrightarrow{l\triangleleft k}\equiv P'$ iff $P \equiv \nu\tilde{m}.(\lfloor p \rfloor_l^{\sigma k} \parallel Q)$ and $P' \equiv \nu\tilde{m}.(\lfloor p \rfloor_l^\sigma \parallel Q)$ for some $\tilde{m}$, p, σ, and Q where $lk \cap \tilde{m} = \emptyset$, and $k \notin fl(P)$.*

Proof: Similar to the proof of Lemma 8. □

Lemma 13. *$P \xrightarrow{\overline{l}} P'$ implies $P \searrow_l P'$.*

Proof: Suppose $P \xrightarrow{\overline{l}} P'$. The proof is by induction in the derivation of the transition $P \xrightarrow{\overline{l}} P'$.

The case where $P \xrightarrow{\overline{l}} P'$ is inferred from the rule $(close)$ follows from Lemma 8, the remaining cases follows by induction. □

Lemma 14. *$P \searrow_l P'$ implies $P \xrightarrow{\overline{l}} Q$ for some Q such that $Q \equiv P'$.*

Proof: Suppose $P \searrow_l P'$. The proof is by induction in the derivation of $P \searrow_l P'$, and making use of Lemma 7 in case $P \searrow_l P'$ is obtained closing by $\equiv$.

If $P \searrow_l P'$ is due to the rule (brd) the result follows due to Lemma 8 and the lts-rule $(close)$. The remaining cases follows by induction. □

Lemma 15. *$P \xrightarrow{\tau} P'$ implies $P \searrow P'$.*

Proof Suppose $P \xrightarrow{\tau} P'$. The proof is by induction in the derivation of the transition $P \xrightarrow{\tau} P'$.

The case where $P \xrightarrow{\tau} P'$ is inferred by the rule $(hide)$ follows due to Lemma 13, and when $P \xrightarrow{\tau} P'$ is inferred by rules (con_2) the result follows due to Lemma 10 and 11. If $P \xrightarrow{\tau} P'$ is inferred from rule (dis_2) the result follows from Lemma 12.

Finally, if the transition $P \xrightarrow{\tau} P'$ follows by one of the rules (par_1) and (par_2) (or their symmetric counter parts), or by one of the rules (res_1) and (res_2) the lemma holds by induction because $\searrow$ is closed by restriction and parallel composition. □

Lemma 16. *$P \searrow P'$ implies $P \xrightarrow{\tau} Q$ for some Q such that $Q \equiv P'$.*

Proof Suppose $P \searrow P'$. If $P \searrow P'$ is because of the rule $(hide)$ the result follows due to Lemma 14. The case where $P \searrow P'$ is due to rule (dis) follows due to the lts-rules (dis_1) and (dis_2). If $P \searrow P'$ is due to rule (con) the result follows from the lts-rules (con_1) and (con_2).

The closing by parallel composition and restriction follows by the lts-rules (par_1) and (par_2) (and their symmetric counter parts) and by (res_1) and (res_2). The closing by $\equiv$ follows due to Lemma 7. □

A.2 Proof of Lemma 3

In order to prove Lemma 3 we show that $\mathcal{R}$ is a weak bisimulation where

$$\mathcal{R}= \{(P_{l\oplus k}, P_{l\ominus k}) \mid P \in \mathbf{N}, \text{ and } l, k \in fl(P)\} \cup \approx$$

The result follows because $P_{l\oplus k} \xrightarrow{\tau} P_{l\ominus k}$ and also $P_{l\ominus k} \xrightarrow{\tau} P_{l\oplus k}$ whenever $l, k \in fl(P)$. □

A.3 Open output bisimulation

In order to show that $\approx$ is a congruence we give an alternative characterization of $\approx$ that is more adequate in our proofs. Alternatively to consider only completed broadcast transitions, i.e. transitions on the form $P \xrightarrow{\bar{l}} P'$, we define instead an *open output bisimulation* depending on data broadcast to the environment, i.e. we take transitions of the type $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ into account.

Definition 11. *A binary relation $\mathcal{R}$ on $\mathbf{N}$ is a weak* open output *simulation if $P \mathcal{R} Q$ implies $fl(P) = fl(Q)$ and for all $p \in \mathbf{P}$,*

1. *if* $P \xrightarrow{\tau} P'$ *then* $\exists Q'.\, Q \Longrightarrow Q'$ *and* $P' \mathcal{R} Q'$
2. *if* $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ *then* $\forall \sigma'.\, \sigma' \subseteq \sigma.\, \forall (x)A_{\sigma'\oplus l}.\, \tilde{n} \cap fn((x)A_{\sigma'\oplus l}) = \emptyset.\, \exists Q'.$
   $Q \parallel (x)A_{\sigma'\oplus l} \overset{\bar{l}}{\Longrightarrow} Q'$ *and* $\nu\tilde{n}.(P' \parallel A_{\sigma'\oplus l}\{t/x\}) \mathcal{R} Q'$
3. *if* $P \xrightarrow{l\overline{\sigma}(t)} P'$ *then* $\forall \sigma'.\, \sigma' \cap \sigma l = \emptyset,\, \exists Q'.$
   $Q \parallel \lfloor \langle t\rangle.p\rfloor_l^{\sigma\sigma'} \overset{\bar{l}}{\Longrightarrow} Q'$ *and* $P' \parallel \lfloor p\rfloor_l^{\sigma\sigma'} \mathcal{R} Q'$
4. *if* $P \xrightarrow{\nu\tilde{m}.l\triangleright} P'$ *then* $\forall\, k.\, k \notin fl(P) \cup \tilde{m}.\, \forall \sigma.\, \sigma \cap \tilde{m}k = \emptyset.\, \exists Q'.$
   $Q \parallel \lfloor p\rfloor_k^{\sigma} \overset{\tau}{\Longrightarrow} Q'$ *and* $\nu\tilde{m}.(P' \parallel \lfloor p\rfloor_k^{\sigma})_{l\oplus k} \mathcal{R} Q'$
5. *if* $P \xrightarrow{l\triangleleft k} P'$ *then* $\forall \sigma.\, k \notin \sigma.\, \exists Q'.$
   $Q \parallel \lfloor p\rfloor_k^{\sigma l} \overset{\tau}{\Longrightarrow} Q'$ *and* $P' \parallel \lfloor p\rfloor_k^{\sigma} \mathcal{R} Q'$

$\mathcal{R}$ is a weak open output bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are weak open output simulations.

Let $\approx_o$ be the largest weak output open bisimulation.
Weak output open bisimulation up to $\equiv$ is used in the proof of Theorem 2.

Definition 12. *A binary relation $\mathcal{R}$ on $\mathbf{N}$ is a weak open output simulation up to $\equiv$ if $P \mathcal{R} Q$ implies $fl(P) = fl(Q)$ and for all $p \in \mathbf{P}$,*

1. *if* $P \xrightarrow{\tau} P'$ *then* $\exists Q'.\, Q \Longrightarrow Q'$ *and* $P' \equiv\mathcal{R}\equiv Q'$
2. *if* $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ *then* $\forall \sigma'.\, \sigma' \subseteq \sigma.\, \forall (x)A_{\sigma'\oplus l}.\, \tilde{n} \cap fn((x)A_{\sigma'\oplus l}) = \emptyset.\, \exists Q'.$
   $Q \parallel (x)A_{\sigma'\oplus l} \overset{\bar{l}}{\Longrightarrow} Q'$ *and* $\nu\tilde{n}.(P' \parallel A'_{\sigma}\{t/x\}) \equiv\mathcal{R}\equiv Q'$
3. *if* $P \xrightarrow{l\overline{\sigma}(t)} P'$ *then* $\forall \sigma'.\, \sigma' \cap \sigma l = \emptyset,\, \exists Q'.$
   $Q \parallel \lfloor \langle t\rangle.p\rfloor_l^{\sigma\sigma'} \overset{\bar{l}}{\Longrightarrow} Q'$ *and* $P' \parallel \lfloor p\rfloor_l^{\sigma\sigma'} \equiv\mathcal{R}\equiv Q'$
4. *if* $P \xrightarrow{\nu\tilde{m}.l\triangleright} P'$ *then* $\forall\, k.\, k \notin fl(P) \cup \tilde{m}.\, \forall \sigma.\, \sigma \cap \tilde{m}k = \emptyset.\, \exists Q'.$
   $Q \parallel \lfloor p\rfloor_k^{\sigma} \overset{\tau}{\Longrightarrow} Q'$ *and* $\nu\tilde{m}.(P' \parallel \lfloor p\rfloor_k^{\sigma})_{l\oplus k} \equiv\mathcal{R}\equiv Q'$
5. *if* $P \xrightarrow{l\triangleleft k} P'$ *then* $\forall \sigma.\, k \notin \sigma.\, \exists Q'.$
   $Q \parallel \lfloor p\rfloor_k^{\sigma l} \overset{\tau}{\Longrightarrow} Q'$ *and* $P' \parallel \lfloor p\rfloor_k^{\sigma} \equiv\mathcal{R}\equiv Q'$

$\mathcal{R}$ is a weak open output bisimulation up to $\equiv$ if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ are weak open output simulations up to $\equiv$.

Lemma 17. *If $\mathcal{R}$ is a weak open output bisimulation up to $\equiv$ then $\equiv\mathcal{R}\equiv$ is a weak open output bisimulation.*

Proof Suppose $\mathcal{R}$ is a weak open output bisimulation up to $\equiv$. We only show that $\equiv\mathcal{R}\equiv$ is a weak open output simulation, the proof of $(\equiv\mathcal{R}\equiv)^{-1}$ being a weak open output simulation is similar.

Let $P \equiv P_1 \mathcal{R} Q_1 \equiv Q$. Suppose $P \xrightarrow{\alpha} P'$. We only consider one of the cases where $\alpha = \nu\tilde{m}.l\triangleright$, the other cases are immediate or similar.

If $P \xrightarrow{\nu\tilde{m}.l\triangleright} P'$ then, due to Lemma 7, there exists $P_1{}'$ such that $P_1 \xrightarrow{\nu\tilde{m}.l\triangleright} P_1{}'$ and $P' \equiv P_1{}'$. Then, since $P_1 \mathcal{R} Q_1$, for all $p \in \mathbf{P}$, for all $k \notin fl(P) \cup \tilde{m}$, and for all σ with $\sigma \cap \tilde{m}k = \emptyset$ there exists $Q_1{}'$ such that $Q_1 \parallel \lfloor p \rfloor_k^\sigma \stackrel{\tau}{\Longrightarrow} Q_1{}'$ and

$$\nu\tilde{m}.(P_1{}' \parallel \lfloor p \rfloor_k^\sigma)_{l\oplus k} \equiv\mathcal{R}\equiv Q_1{}'$$

Because $\equiv$ is a congruence we have

$$\nu\tilde{m}.(P' \parallel \lfloor p \rfloor_k^\sigma)_{l\oplus k} \equiv \nu\tilde{m}.(P_1{}' \parallel \lfloor p \rfloor_k^\sigma)_{l\oplus k}$$

and

$$Q \parallel \lfloor p \rfloor_k^\sigma \equiv Q_1 \parallel \lfloor p \rfloor_k^\sigma$$

From Lemma 7 we infer that there exists Q' such that

$$Q \parallel \lfloor p \rfloor_k^\sigma \stackrel{\tau}{\Longrightarrow} Q'$$

and $Q_1{}' \equiv Q'$. Hence, since $\equiv$ is transitive,

$$\nu\tilde{m}.(P' \parallel \lfloor p \rfloor_k^\sigma)_{l\oplus k} \equiv Q'$$

□

Likewise we may define the notion of a strong (weak) bisimulation (or c-bisimulation) up to $\equiv$ and show that whenever $\mathcal{R}$ is a strong (weak) bisimulation (or a c-bisimulation) up to $\equiv$ then $\equiv\mathcal{R}\equiv$ is a strong (weak) bisimulation (or c-bisimulation).

Theorem 7. $\approx = \approx_o$.

Proof We show $\approx \subseteq \approx_o$ and $\approx_o \subseteq \approx$.

Case ($\approx_o \subseteq \approx$) To obtain $\approx_o \subseteq \approx$ we show that $\approx_o$ is a weak bisimulation up to $\equiv$. Suppose $P \approx_o Q$. It's enough to show only that if $P \xrightarrow{\bar{l}} P'$ then there exists $Q \stackrel{\bar{l}}{\Longrightarrow} Q'$ such that $P' \equiv \approx_o \equiv Q'$.

Assume $P \xrightarrow{\bar{l}} P'$, then there exists $P \xrightarrow{\bar{l}\epsilon\nu\tilde{n}\langle t\rangle} P''$ with $P' = \nu\tilde{n}.P''$. Hence, because $P \approx_o Q$, there exists $Q \parallel (x)A_{\epsilon\oplus l} \stackrel{\bar{l}}{\Longrightarrow} Q'$ such that

$$\nu\tilde{n}.(P'' \parallel A_{\epsilon\oplus l}\{t/x\}) \approx_o Q'$$

Since $Q \parallel (x)A_{\epsilon\oplus l} \equiv Q$ there exists, due to Lemma 7, $Q \stackrel{\bar{l}}{\Longrightarrow} Q''$ such that $Q' \equiv Q''$. Then we obtain as desired because

$$\nu\tilde{n}.P'' \equiv \nu\tilde{n}.(P'' \parallel A_{\epsilon\oplus l}\{t/x\}) \approx_o Q' \equiv Q''$$

Case ($\approx \subseteq \approx_o$) In order to show $\approx \subseteq \approx_o$ we prove that $\approx$ is a weak open output bisimulation up to $\equiv$. Suppose $P \approx Q$. It is sufficient to show only that if $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}.\langle t\rangle} P'$ then for all σ' with $\sigma' \subseteq \sigma$ and for all $(x)A_{\sigma'\oplus l}$ with $\tilde{n} \cap fn((x)A_{\sigma'\oplus l}) = \emptyset$ there exists Q' such that $Q \parallel (x)A_{\sigma'\oplus l} \stackrel{\bar{l}}{\Longrightarrow} Q'$ and $\nu\tilde{n}.(P' \parallel A_{\sigma'\oplus l}\{t/x\}) \equiv \approx \equiv Q'$.

Assume $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$. Then $\sigma \cap fl(P) = \emptyset$ due to Lemma 8. Let $\sigma' \subseteq \sigma$ and let

$$(x)A_{\sigma'\oplus l} = \lfloor (x).p_1 \rfloor_{l_1}^{\sigma_1 l} \parallel \ldots \lfloor (x).p_k \rfloor_{l_k}^{\sigma_k l}$$

be such that $\tilde{n} \cap fn((x)A_{\sigma'\oplus l}) = \emptyset$. Since $l \in fl(P)$ it follows due to Lemma 10 that $P \xrightarrow{l\triangleright} P$. Then because $P \approx Q$ and since $l_1 \notin fl(P)$ there exists Q_1 such that $Q \parallel \lfloor (x).p_1 \rfloor_{l_1}^{\sigma_1 l} \stackrel{\tau}{\Longrightarrow} Q_1$ and

$$P_1 = (P \parallel \lfloor (x).p_1 \rfloor_{l_1}^{\sigma_1 l})_{l\oplus l_1} \approx Q_1$$

(Observe that $P_1 = P \parallel \lfloor (x).p_1 \rfloor_{l_1}^{\sigma_1 l}$ because $P = P_{l\oplus l_1}$.) Likewise, $P_i \xrightarrow{l\triangleright} P_i$ so there exists Q_{i+1} such that

$$Q_i \parallel \lfloor (x).p_{i+1} \rfloor_{l_{i+1}}^{\sigma_{i+1} l} \stackrel{\tau}{\Longrightarrow} Q_{i+1}$$

and

$$P_{i+1} = P_i \parallel \lfloor (x).p_{i+1} \rfloor_{l_{i+1}}^{\sigma_{i+1} l} \approx Q_{i+1}$$

for $i = 1, \ldots, k-1$. From the steps above it follows that

$$P_k \equiv P \parallel (x) A_{\sigma' \oplus l}$$

Then because

$$P \parallel (x) A_{\sigma' \oplus l} \xrightarrow{\bar{l}} \nu \tilde{n}.(P' \parallel A_{\sigma' \oplus l}\{t/x\})$$

also $P_k \xrightarrow{\bar{l}} {P_k}'$ for some ${P_k}'$ with $\nu \tilde{n}.(P' \parallel A_{\sigma' \oplus l}\{t/x\}) \equiv {P_k}'$ due to Lemma 7. Then, since $P_k \approx Q_k$ there exists ${Q_k}'$ such that $Q_k \stackrel{\bar{l}}{\Longrightarrow} {Q_k}'$ and

$$\nu \tilde{n}.(P' \parallel A_{\sigma' \oplus l}\{t/x\}) \equiv \approx \ {Q_k}'$$

The final part of the proof is to observe that, due to Lemma 7,

$$Q \parallel (x) A_{\sigma' \oplus l} \stackrel{\bar{l}}{\Longrightarrow} Q'$$

for some Q' with $Q' \equiv {Q_k}'$ because

$$Q \parallel (x) A_{\sigma' \oplus l} \equiv Q \parallel \lfloor (x).p_1 \rfloor_{l_1}^{\sigma_1 l} \parallel \ldots \lfloor (x).p_k \rfloor_{l_k}^{\sigma_k l}$$

and

$$\begin{aligned}
& Q \parallel \lfloor (x).p_1 \rfloor_{l_1}^{\sigma_1 l} \parallel \ldots \lfloor (x).p_k \rfloor_{l_k}^{\sigma_k l} \\
& \stackrel{\tau}{\Longrightarrow} Q_1 \parallel \lfloor (x).p_2 \rfloor_{l_2}^{\sigma_2 l} \parallel \ldots \lfloor (x).p_k \rfloor_{l_k}^{\sigma_k l} \\
& \stackrel{\tau}{\Longrightarrow} Q_2 \parallel \lfloor (x).p_3 \rfloor_{l_3}^{\sigma_3 l} \parallel \ldots \lfloor (x).p_k \rfloor_{l_k}^{\sigma_k l} \\
& \stackrel{\tau}{\Longrightarrow} \ldots \\
& \stackrel{\tau}{\Longrightarrow} Q_k \\
& \stackrel{\bar{l}}{\Longrightarrow} {Q_k}'
\end{aligned}$$

from which we get:

$$\nu \tilde{n}.(P' \parallel A_{\sigma'}\{t/x\}) \equiv \approx \equiv Q'$$

□

A.4 Proof of Theorem 1 and 2

Below we only show the proof of Theorem 2. The proof of Theorem 1 is similar and simpler.

Because of Theorem 7 we only need to show that $\approx_o$ is a congruence. In order to do so it's sufficient, due to Lemma 17, to show that $\mathcal{R}$ is a weak open output bisimulation up to $\equiv$ where

$$\mathcal{R} = \{(\nu \tilde{m}.(P \parallel Q), \nu \tilde{m}.(P' \parallel Q)) \mid P \approx_o P' \text{ and } fl(P) \cap fl(Q) = \emptyset\}$$

We only show here that $\mathcal{R}$ is a weak open output simulation up to $\equiv$, the proof of $\mathcal{R}^{-1}$ being a weak open output simulation up to $\equiv$ is similar.

Let $\nu \tilde{m}.(P_1 \parallel Q) \ \mathcal{R} \ \nu \tilde{m}.(P_2 \parallel Q)$. Suppose $\nu \tilde{m}.(P_1 \parallel Q) \xrightarrow{\alpha} R$. The proof proceeds by induction in the derivation of $\nu \tilde{m}.(P_1 \parallel Q) \xrightarrow{\alpha} R$.

Case 1 ($\alpha = \tau$) Suppose $\nu \tilde{m}.(P_1 \parallel Q) \xrightarrow{\tau} R$.

Case 1.1 (*hide*) The case where $\nu \tilde{m}.(P_1 \parallel Q) \xrightarrow{\tau} R$ because $P_1 \parallel Q \xrightarrow{\bar{l}} R'$, $l \in \tilde{m}$, and $R = \nu \tilde{m}.R'$ follows by induction because then there exists some $P_1 \parallel Q \xrightarrow{\bar{l} \epsilon \nu \tilde{n} \langle t \rangle} R''$ with $R' = \nu \tilde{n}.R''$.

Case 1.2 (res_1) and (res_2) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\tau} R$ because $P_1 \parallel Q \xrightarrow{\tau} R'$, and $R = \nu\tilde{m}.R'$.

Case 1.2.1 (con_2) Suppose $P_1 \parallel Q \xrightarrow{\tau} R'$ because

$$P_1 \xrightarrow{\nu\tilde{m}'.l\triangleright} {P_1}' \quad \text{and} \quad Q \xrightarrow{\nu\tilde{m}''.m\triangleright} Q'$$

$\tilde{m}' \cap \tilde{m}'' = \tilde{m}' \cap \mathit{flc}(Q) = \tilde{m}'' \cap \mathit{flc}(P_1) = \emptyset$, and

$$R' = \nu\tilde{m}''\tilde{m}'.({P_1}' \parallel Q')_{m\oplus l}$$

From Lemma 10 and 11 we infer $Q \equiv \nu\tilde{m}''.Q'$ with either $\tilde{m}'' = \{m\}$ or $\tilde{m}'' = \emptyset$, and

$$Q' \equiv \nu\tilde{m}'''.(\lfloor p \rfloor^{\sigma}_{m} \parallel Q_0)$$

for some $\tilde{m}'''$, p, σ, and Q_0 where $m \notin \tilde{m}'''$. Since $P_1 \approx_o P_2$ there exists ${P_2}'$ such that

$$P_2 \parallel \lfloor p \rfloor^{\sigma}_{m} \stackrel{\tau}{\Longrightarrow} {P_2}'$$

and

$$\nu\tilde{m}'.({P_1}' \parallel \lfloor p \rfloor^{\sigma}_{m})_{l\oplus m} \approx_o {P_2}'$$

From (res_2) and (par_2) we infer

$$\nu\tilde{m}.\nu\tilde{m}''.\nu\tilde{m}'''.(P_2 \parallel \lfloor p \rfloor^{\sigma}_{m} \parallel Q_0) \stackrel{\tau}{\Longrightarrow} \nu\tilde{m}.\nu\tilde{m}''.\nu\tilde{m}'''.({P_2}' \parallel Q_0)$$

Assuming $\tilde{m}'' \cap \mathit{flc}(P_2) = \emptyset$ and $\tilde{m}''' \cap (\mathit{flc}(P_2) \cup \mathit{fn}(P_2)) = \emptyset$ (using α-conversion if needed) we get

$$\nu\tilde{m}.(P_2 \parallel Q) \equiv \nu\tilde{m}.\nu\tilde{m}''.\nu\tilde{m}'''.(P_2 \parallel \lfloor p \rfloor^{\sigma}_{m} \parallel Q_0)$$

Hence, because of Lemma 7, there exists ${P_2}''$ such that

$$\nu\tilde{m}.(P_2 \parallel Q) \stackrel{\tau}{\Longrightarrow} {P_2}''$$

and

$$\nu\tilde{m}.\nu\tilde{m}''.\nu\tilde{m}'''.({P_2}' \parallel Q_0) \equiv {P_2}''$$

Finally, since

$$R \equiv \nu\tilde{m}.\nu\tilde{m}''.\nu\tilde{m}'''.(\nu\tilde{m}'.({P_1}' \parallel \lfloor p \rfloor^{\sigma}_{m})_{l\oplus m} \parallel Q_0)$$

assuming $\tilde{m}''' \cap (\mathit{flc}(P_1) \cup \mathit{fn}(P_1)) = \emptyset$ and $\tilde{m}' \cap (\tilde{m}''' \cup \mathit{fl}(Q_0)) = \emptyset$ (using α-conversion if needed) we obtain

$$R \equiv\mathcal{R}\equiv {P_2}'''$$

Case 1.2.2 (dis_2) The case where $P_1 \parallel Q \xrightarrow{\tau} R'$ because

$$P_1 \xrightarrow{l\triangleleft k} {P_1}' \quad \text{and} \quad Q \xrightarrow{k\triangleleft l} Q'$$

and $R' = {P_1}' \parallel Q'$ is similar to case 1.2.1 above.

Case 1.2.3 The case where $P_1 \parallel Q \xrightarrow{\tau} R'$ because $P_1 \xrightarrow{\tau} {P_1}'$ (or $Q \xrightarrow{\tau} Q'$) and $R' = {P_1}' \parallel Q$ (or $R' = P_1 \parallel Q'$) is immediate.

Case 2 ($\alpha = \overline{l}\sigma\nu\tilde{n}\langle t\rangle$) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t\rangle} R$ because $P_1 \parallel Q \xrightarrow{\overline{l}\sigma\nu\tilde{n}'\langle t\rangle} R'$ and $R = \nu\tilde{m}'.R'$ where, due to multiple applications of the rules (res_1 and ($open_1$), $\tilde{m} \cap \tilde{n}' = \emptyset$, $\tilde{m}' = \tilde{m} \setminus \tilde{n}_1$ for $\tilde{n}_1 = \tilde{m} \cap \mathit{fn}(t)$, and $\tilde{n} = \tilde{n}' \cup \tilde{n}_1$. Also, $\sigma l \cap \tilde{m} = \emptyset$.

Case 2.1 ($lose$) The case where $P_1 \parallel Q \xrightarrow{\overline{l}\sigma\nu\tilde{n}'\langle t\rangle} R'$ because of a transition $P_1 \parallel Q \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}'\langle t\rangle} R'$ follows by induction.

Case 2.2 ($synch$) Suppose $P_1 \parallel Q \xrightarrow{\overline{l}\sigma\nu\tilde{n}'\langle t\rangle} R'$ because $P_1 \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}'\langle t\rangle} P_1{}'$, $Q \xrightarrow{l\overline{\sigma}'(t)} Q'$, and $R' = P_1{}' \parallel Q'$ where $\tilde{n}' \cap fn(Q) = \emptyset$ and $\sigma \cap fl(Q) = \emptyset$.

From Lemma 9 we infer

$$Q \equiv \nu\tilde{m}_0.((x)A_{\sigma'\oplus l} \parallel Q_0)$$

for some $\tilde{m}_0$, $(x)A_{\sigma'\oplus l}$, and Q_0 where $\sigma' \cap \tilde{m}_0 = \emptyset$. We assume (using α-conversion if needed) that $\tilde{n}' \cap \tilde{m}_0 = \emptyset$.

Because $P_1 \approx_o P_2$, for any σ'' where $\sigma'' \subseteq \sigma$ and for any $(x)A_{\sigma''\oplus l}$ where $\tilde{n}' \cap fn((x)A_{\sigma''\oplus l}) = \emptyset$ there exists $P_2{}'$ such that

$$P_2 \parallel (x)A_{\sigma''\oplus l} \parallel (x)A_{\sigma'\oplus l} \stackrel{\overline{l}}{\Longrightarrow} P_2{}'$$

with

$$\nu\tilde{n}'.(P_1{}' \parallel A_{\sigma''\oplus l}\{t/x\} \parallel A_{\sigma'\oplus l}\{t/x\}) \approx_o P_2{}'$$

Hence,

$$\nu\tilde{m}.\nu\tilde{m}_0.(P_2 \parallel (x)A_{\sigma''\oplus l} \parallel (x)A_{\sigma'\oplus l} \parallel Q_0) \stackrel{\overline{l}}{\Longrightarrow} \nu\tilde{m}.\nu\tilde{m}_0.(P_2{}' \parallel Q_0)$$

Assuming (using α-conversion if needed) $\tilde{m}_0 \cap (flc(P_2) \cup fn(P_2)) = \emptyset$ and $\tilde{m}_0 \cap (flc((x)A_{\sigma''\oplus l}) \cup fn((x)A_{\sigma''\oplus l})) = \emptyset$ we have

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel (x)A_{\sigma''\oplus l} \equiv \nu\tilde{m}.\nu\tilde{m}_0.(P_2 \parallel (x)A_{\sigma''\oplus l} \parallel (x)A_{\sigma'\oplus l} \parallel Q_0)$$

Then, due to Lemma 7, there exists $P_2{}''$ such that

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel (x)A_{\sigma''\oplus l} \stackrel{\overline{l}}{\Longrightarrow} P_2{}''$$

and $\nu\tilde{m}.\nu\tilde{m}_0.(P_2{}' \parallel Q_0) \equiv P_2{}''$.

Assuming (using α-conversion if needed) $\tilde{m}_0 \cap (fn(P_1) \cup flc(P_1) \cup \tilde{n}') = \emptyset$ and that $\tilde{m} \cup (flc((x)A_{\sigma''\oplus l}) \cup fn((x)A_{\sigma''\oplus l})) = \emptyset$, then since

$$\nu\tilde{n}.(R \parallel A_{\sigma''\oplus l}\{t/x\}) \equiv \nu\tilde{m}\tilde{m}_0.(\nu\tilde{n}'.(P_1{}' \parallel A_{\sigma''\oplus l}\{t/x\} \parallel A_{\sigma'\oplus l}\{t/x\}) \parallel Q_0)$$

we get as desired

$$\nu\tilde{n}.(R \parallel A_{\sigma''\oplus l}\{t/x\}) \equiv\mathcal{R}\equiv P_2{}''$$

Case 2.3 This case covers the rule symmetric to the rule ($synch$).

Suppose $P_1 \parallel Q \xrightarrow{\overline{l}\sigma\nu\tilde{n}'\langle t\rangle} R'$ because $P_1 \xrightarrow{l\overline{\sigma}'(t)} P_1{}'$, $Q \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}'\langle t\rangle} Q'$, $\tilde{n}' \cap fn(P_1) = \emptyset$, $\sigma \cap fl(P_1) = \emptyset$, and $R' = P_1{}' \parallel Q'$. From Lemma 8 it follows that

$$Q \equiv \nu\tilde{m}_0.(\lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \parallel (x)A_{\sigma''\oplus l} \parallel Q_0)$$

and

$$Q' \equiv \nu\tilde{m}_1.(\lfloor p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \parallel A_{\sigma''\oplus l}\{t/x\} \parallel Q_0)$$

for some $\tilde{m}_0$, p, σ'', σ''', $(x)A_{\sigma''\oplus l}$, and Q_0 where $\tilde{n} = \tilde{m}_0 \cap fn(t)$, $\tilde{m}_1 = \tilde{m}_0 \setminus \tilde{n}$, $\tilde{m}_0 \cap \sigma\sigma' l = \emptyset$, and $\sigma\sigma' \cap fl(Q) = \emptyset$. Because $P_1 \approx_o P_2$ there exists $P_2{}'$ such that

$$P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \stackrel{\overline{l}}{\Longrightarrow} P_2{}' \tag{11}$$

and

$$P_1{}' \parallel \lfloor p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \approx_o P_2{}'$$

Since $\sigma\sigma'' \cap fl(P_2) = \emptyset$, then from (11) we infer

$$P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \stackrel{\tau}{\Longrightarrow} P_2{}^1 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma''\sigma_0} \xrightarrow{\overline{l}} P_2{}^2 \parallel \lfloor p\rfloor_l^{\sigma\sigma''\sigma_0} \stackrel{\tau}{\Longrightarrow} P_2{}'$$

for some $P_2{}^1$, $P_2{}^2$, and some σ_0 with (due to rule dis_2) $\sigma_0 \subseteq \sigma''' \cup fl(P_2)$. Let $\sigma_1 \subseteq \sigma$. For any $(x)A_{\sigma_1\oplus l}$ with $\tilde{n} \cap fn((x)A_{\sigma_1\oplus l}) = \emptyset$ let

$$P_2{}'' = \nu\tilde{m}.\nu\tilde{m}_0.(P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \parallel (x)A_{\sigma''\oplus l} \parallel (x)A_{\sigma_1\oplus l} \parallel Q_0)$$

Due to

$$P_2{}^1 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma''\sigma_0} \xrightarrow{\bar{l}} P_2{}^2 \parallel \lfloor p\rfloor_l^{\sigma\sigma''\sigma_0}$$

and because

$$(x)A_{\sigma''\oplus l} \parallel (x)A_{\sigma_1\oplus l} \xrightarrow{l\overline{\sigma''\sigma_1}} A_{\sigma''\oplus l}\{t/x\} \parallel A_{\sigma_1\oplus l}\{t/x\}$$

we infer,

$$P_2{}'' \stackrel{\bar{l}}{\Longrightarrow} \nu\tilde{m}.\nu\tilde{m}_0.(P_2{}' \parallel A_{\sigma''\oplus l}\{t/x\} \parallel A_{\sigma_1\oplus l}\{t/x\} \parallel Q_0) = P_2{}'''$$

Assuming $\tilde{m}_0 \cap (fn(P_2) \cup flc(P_2)) = \emptyset$ and $\tilde{m}_0 \cap (fn((x)A_{\sigma_1\oplus l}) \cup flc((x)A_{\sigma_1\oplus l})) = \emptyset$ (using α-conversion if needed), we get

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel (x)A_{\sigma_1\oplus l} \equiv P_2{}''$$

Then there exists $P_2{}''''$, due to Lemma 7, such that

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel (x)A_{\sigma_1\oplus l} \stackrel{\bar{l}}{\Longrightarrow} P_2{}''''$$

and $P_2{}''' \equiv P_2{}''''$. Then, letting $Q_1 = A_{\sigma''\oplus l}\{t/x\} \parallel A_{\sigma_1\oplus l}\{t/x\} \parallel Q_0$, we have

$$\nu\tilde{n}.(R \parallel A_{\sigma_1\oplus l}\{t/x\}) \equiv \nu\tilde{m}.\nu\tilde{m}_0.(P_1{}' \parallel \lfloor p\rfloor_l^{\sigma\sigma'\sigma''\sigma'''} \parallel Q_1)$$

assuming $\tilde{m}_0 \cap (fn(P_1) \cup flc(P_1)) = \emptyset$ and $\tilde{m}_0 \cap (fn((x)A_{\sigma_1\oplus l}) \cup flc((x)A_{\sigma_1\oplus l})) = \emptyset$ (using α-conversion if needed) we get

$$\nu\tilde{n}.(R \parallel A_{\sigma_1\oplus l}\{t/x\}) \equiv\mathcal{R}\equiv P_2{}''''$$

Case 2.4 (par_1) The case where $P_1 \parallel Q \xrightarrow{\bar{l}\sigma\nu\tilde{n}'\langle t\rangle} R'$ because $P_1 \xrightarrow{\bar{l}\sigma\nu\tilde{n}'\langle t\rangle} P_1{}'$, and $R' = P_1{}' \parallel Q$ where $\tilde{n}' \cap fn(Q) = \emptyset$ and $\sigma \cap fl(Q) = \emptyset$ is similar to the cases above.

Case 2.5 The case where $P_1 \parallel Q \xrightarrow{\bar{l}\sigma\nu\tilde{n}'\langle t\rangle} R'$ because $Q \xrightarrow{\bar{l}\sigma\nu\tilde{n}'\langle t\rangle} Q'$, and $R' = P_1 \parallel Q'$ where $\tilde{n}' \cap fn(P_1) = \emptyset$ and $\sigma \cap fl(P_1) = \emptyset$ is similar to the cases above.

Case 3 ($\alpha = l\overline{\sigma}(t)$) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{l\overline{\sigma}(t)} R$. Then $P_1 \parallel Q \xrightarrow{l\overline{\sigma}(t)} R'$, $\tilde{m} \cap (\sigma l \cup fn(t)) = \emptyset$, and $R = \nu\tilde{m}.R'$. Let $p \in \mathcal{P}$ and let σ' be such that $\sigma' \cap \sigma l = \emptyset$. We assume $\tilde{m} \cap (fn(p) \cup \sigma') = \emptyset$ (using α-conversion if needed).

Case 3.1 (rec_2) Suppose $P_1 \parallel Q \xrightarrow{l\overline{\sigma}(t)} R'$ because $P_1 \xrightarrow{l\overline{\sigma}_1(t)} P_1{}'$, $Q \xrightarrow{l\overline{\sigma}_2(t)} Q'$, $\sigma = \sigma_1\sigma_2$, and $R' = P_1{}' \parallel Q'$. Because $P_1 \approx_o P_2$, for any p, for any σ' with $\sigma' \cap \sigma l = \emptyset$, we have

$$P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_1\sigma_2\sigma'} \stackrel{\bar{l}}{\Longrightarrow} P_2{}' \tag{12}$$

and

$$P_1{}' \parallel \lfloor p\rfloor_l^{\sigma_1\sigma_2\sigma'} \approx_o P_2{}'$$

From (12) we infer, since $\sigma_2 \cap fl(P_2) = \emptyset$,

$$P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_1\sigma_2\sigma'} \stackrel{\tau}{\Longrightarrow} P_2{}^1 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_2\sigma''} \xrightarrow{\bar{l}} P_2{}^2 \parallel \lfloor p\rfloor_l^{\sigma_2\sigma''} \stackrel{\tau}{\Longrightarrow} P_2{}'$$

for some $P_2{}^1$, $P_2{}^2$, and some σ'' with (due to rule dis_2) $\sigma'' \subseteq \sigma' \cup fl(P_2)$. Due to

$$P_2{}^1 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_2\sigma''} \xrightarrow{\bar{l}} P_2{}^2 \parallel \lfloor p\rfloor_l^{\sigma_2\sigma''}$$

and because $Q \xrightarrow{l\overline{\sigma}_2(t)} Q'$ we conclude

$$P_2{}^1 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_2\sigma''} \parallel Q \xrightarrow{\bar{l}} P_2{}^2 \parallel \lfloor p\rfloor_l^{\sigma_2\sigma''} \parallel Q'$$

and hence

$$\nu\tilde{m}.(P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_1\sigma_2\sigma'} \parallel Q) \stackrel{\bar{l}}{\Longrightarrow} \nu\tilde{m}.(P_2{}' \parallel Q')$$

Since

$$\nu\tilde{m}.(P_2 \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_1\sigma_2\sigma'} \parallel Q) \equiv \nu\tilde{m}.(P_2 \parallel Q) \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_1\sigma_2\sigma'}$$

there exists, due to due to Lemma 7, some $P_2{}''$ such that

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel \lfloor\langle t\rangle.p\rfloor_l^{\sigma_1\sigma_2\sigma'} \stackrel{l}{\Longrightarrow} P_2{}''$$

and $P_2{}'' \equiv P_2{}'$ and therefore

$$\nu\tilde{m}.(P_1{}' \parallel Q') \parallel \lfloor p\rfloor_l^{\sigma_1\sigma_2\sigma'} \equiv\mathcal{R}\equiv P_2{}''$$

Case 3.2 (par_1) The case where $P_1 \parallel Q \stackrel{l\overline{\sigma}(t)}{\longrightarrow} R'$ because $P_1 \stackrel{l\overline{\sigma}(t)}{\longrightarrow} P_1{}'$, $R' = P_1{}' \parallel Q$, and $\sigma \cap fl(Q) = \emptyset$ is immediate.

Case 3.3 The case where $P_1 \parallel Q \stackrel{l\overline{\sigma}(t)}{\longrightarrow} R'$ because $Q \stackrel{l\overline{\sigma}(t)}{\longrightarrow} Q'$, $R' = P_1 \parallel Q'$, and $\sigma \cap fl(P_1) = \emptyset$ is trivial.

Case 4 ($\alpha = \nu\tilde{m}'.l\triangleright$) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} R$.

Case 4.1 (res_2) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} R$ because $P_1 \parallel Q \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} R'$, $\tilde{m}\cap\tilde{m}' = \emptyset$, $l \notin \tilde{m}$, and $R = \nu\tilde{m}.R'$.

Case 4.1.1 (par_2) Suppose $P_1 \parallel Q \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} R'$ because $P_1 \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} P_1{}'$ and $R' = P_1{}' \parallel Q$. Let $k \notin fl(P_1) \cup fl(Q) \cup \tilde{m}'$. Because $P_1 \approx_o P_2$, for all $p \in \mathbf{P}$, and for all σ with $\sigma \cap \tilde{m}'k = \emptyset$ there exists $P_2{}'$ such that

$$P_2 \parallel \lfloor p\rfloor_k^\sigma \stackrel{\tau}{\Longrightarrow} P_2{}'$$

and $\nu\tilde{m}'.(P_1{}' \parallel \lfloor p\rfloor_k^\sigma)_{l\oplus k} \approx_o P_2{}'$. Hence,

$$\nu\tilde{m}.(P_2 \parallel \lfloor p\rfloor_k^\sigma \parallel Q) \stackrel{\tau}{\Longrightarrow} \nu\tilde{m}.(P_2{}' \parallel Q)$$

and since

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel \lfloor p\rfloor_k^\sigma \equiv \nu\tilde{m}.(P_2 \parallel \lfloor p\rfloor_k^\sigma \parallel Q)$$

assuming $\tilde{m} \cap (fn(p) \cup \sigma k) = \emptyset$ (using α-conversion if needed) then, because of Lemma 7, there exists $P_2{}''$ such that

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel \lfloor p\rfloor_k^\sigma \stackrel{\tau}{\Longrightarrow} P_2{}''$$

and $\nu\tilde{m}.(P_2{}' \parallel Q) \equiv P_2{}''$. Finally, since

$$\nu\tilde{m}'.(R \parallel \lfloor p\rfloor_k^\sigma)_{l\oplus k} \equiv \nu\tilde{m}.(\nu\tilde{m}'.(P_1{}' \parallel \lfloor p\rfloor_k^\sigma)_{l\oplus k} \parallel Q)$$

we conclude that

$$\nu\tilde{m}'.(R \parallel \lfloor p\rfloor_k^\sigma)_{l\oplus k} \equiv\mathcal{R}\equiv P_2{}''$$

Case 4.1.2 The case where $P_1 \parallel Q \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} R'$ because $Q \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} Q'$ and $R' = P_1 \parallel Q'$ is immediate.

Case 4.2 ($open_2$) The case where $\nu\tilde{m}.(P_1 \parallel Q) \stackrel{\nu\tilde{m}'.l\triangleright}{\longrightarrow} R$ because $P_1 \parallel Q \stackrel{l\triangleright}{\longrightarrow} R'$, $l \in \tilde{m}$, and $R = \nu\tilde{m}'.R'$ where $\tilde{m}' = \tilde{m} \setminus \{l\}$ is similar to case 4.1 above.

Case 5 ($\alpha = l \triangleleft k$) Similar to Case 4.1 above.

□

A.5 Proof of Theorem 3 and 4

Below we only give the proof of Theorem 4. The proof of Theorem 3 is similar.

In order to show $\approx \subseteq \cong$ it is sufficient to show that $\approx$ is weak reduction closed because from Theorem 1 we know $\approx$ is a congruence. That $\approx$ is weak reduction closed follows from Lemma 13, 14, 15, and 16.

The remaining part of the proof establishes that $\cong \subseteq \approx$. It's sufficient to show that $\cong$ is a weak bisimulation.

Lemma 18. *$\cong$ is a weak bisimulation.*

Proof We only prove $\cong$ to be a weak simulation, the proof of $\cong^{-1}$ being a simulation is similar.

Let $P_1 \cong P_2$. Suppose $P_1 \stackrel{\alpha}{\longrightarrow} P_1{}'$.

Case 1 ($\alpha = \tau$): The case where $P_1 \xrightarrow{\tau} P_1{}'$ is immediate due to Lemma 15 and 16.
Case 2 ($\alpha = \overline{l}$): The case where $P_1 \xrightarrow{\overline{l}} P_1{}'$ is immediate due to Lemma 13, 14, and 16.
Case 3 ($\alpha = l\overline{\sigma}(t)$): Suppose $P_1 \xrightarrow{l\overline{\sigma}(t)} P_1{}'$. Due to Lemma 9,

$$P_1 \equiv \nu\tilde{m}.((x)A_{\sigma\oplus l} \parallel Q)$$

and

$$P_1{}' \equiv \nu\tilde{m}.(A_{\sigma\oplus}\{t/x\} \parallel Q)$$

for some $\tilde{m}$, $(x)A_{\sigma\oplus l}$, and Q where $\sigma l \cap \tilde{m} = \emptyset$ and $l \notin fl(P_1)$. Because $\cong$ is a congruence, for any $\lfloor p \rfloor_l^{\sigma\sigma'}$,

$$P_1 \parallel \lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'} \cong P_2 \parallel \lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'}$$

Assuming $\tilde{m} \cap (fn(t) \cup \sigma' \cup fn(p)) = \emptyset$ (using α-conversion if needed), then because

$$P_1 \parallel \lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'} \searrow_l P_1{}' \parallel \lfloor p \rfloor_l^{\sigma\sigma'}$$

there exists $P_2{}'$ such that

$$P_2 \parallel \lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'} \searrow^* \searrow_l \searrow^* P_2{}'$$

and

$$P_1{}' \parallel \lfloor p \rfloor_l^{\sigma\sigma'} \cong P_2{}'$$

Then, due to Lemma 16 and 14, there exists $P_2{}''$ such that

$$P_2 \parallel \lfloor \langle t \rangle.p \rfloor_l^{\sigma\sigma'} \overset{\overline{l}}{\Longrightarrow} P_2{}''$$

and $P_2{}'' \equiv P_2{}'$. Hence

$$P_1{}' \parallel \lfloor p \rfloor_l^{\sigma\sigma'} \cong P_2{}'$$

because $\equiv \subseteq \cong$.
Case 4 ($\alpha = \nu\tilde{m}.l\triangleright$): Suppose $P_1 \xrightarrow{\nu\tilde{m}.l\triangleright} P_1{}'$. Let $k \notin fl(P_1)$, let $\sigma k \cap \tilde{m} = \emptyset$. For any p we have

$$P_1 \parallel \lfloor p \rfloor_k^{\sigma} \xrightarrow{\tau} \nu\tilde{m}.(P_1{}' \parallel \lfloor p \rfloor_k^{\sigma})_{l\oplus k}$$

and therefore

$$P_1 \parallel \lfloor p \rfloor_k^{\sigma} \searrow \nu\tilde{m}.(P_1{}' \parallel \lfloor p \rfloor_k^{\sigma})_{l\oplus k}$$

due to Lemma 15. Since $P_1 \parallel \lfloor p \rfloor_k^{\sigma} \cong P_2 \parallel \lfloor p \rfloor_k^{\sigma}$ there exists

$$P_2 \parallel \lfloor p \rfloor_k^{\sigma} \searrow^* Q$$

with $\nu\tilde{m}.(P_1{}' \parallel \lfloor p \rfloor_k^{\sigma})_{l\oplus k} \cong Q$. From Lemma 16, there exists Q' such that $P_2 \parallel \lfloor p \rfloor_k^{\sigma l} \overset{\tau}{\Longrightarrow} Q'$ and $Q \equiv Q'$. Hence, because $\equiv \subseteq \cong$,

$$\nu\tilde{m}.(P_1{}' \parallel \lfloor p \rfloor_k^{\sigma})_{l\oplus k} \cong Q'$$

Case 5 ($\alpha = l \triangleleft k$) Similar to Case 4.

□

A.6 Proof of Theorem 5 and 6

Below we only show the proof of Theorem 6. The proof of Theorem 5 is similar and simpler.

Lemma 19. *$\approx_c$ is a c-congruence.*

Proof We must show that $\approx_c$ is closed by restriction and by well-formed parallel composition of networks in $\mathbf{N}_c$. Hence, let

$$\mathcal{R} = \{(\nu\tilde{m}.(P \parallel Q), \nu\tilde{m}.(P' \parallel Q)) \mid P \approx_c P',\ Q \in \mathbf{N}_c,\ \text{and } fl(P) \cap fl(Q) = \emptyset\}$$

We prove $\mathcal{R}$ to be a weak c-bisimulation up to $\equiv$. Here we just show that $\mathcal{R}$ is a weak c-simulation, the proof of $\mathcal{R}^{-1}$ being a weak c-simulation is similar.

Let $\nu\tilde{m}.(P_1 \parallel Q)\mathcal{R}\nu\tilde{m}.(P_2 \parallel Q)$, and suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\alpha} R$. The proof proceeds by induction on the derivation of $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\alpha} R$ and is similar to but simpler than the proof of Theorem 2.

Case 1 ($\alpha = \tau$): Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\tau} R$.

Case 1.1 ($hide$): The case where $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\tau} R$ because $P_1 \parallel Q \xrightarrow{\bar{l}} R'$, $l \in \tilde{m}$, and $R = \nu\tilde{m}.R'$ follows by induction.

Case 1.2 (res_1) and (res_2) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\tau} R$ because $P_1 \parallel Q \xrightarrow{\tau} R'$ and $R = \nu\tilde{m}.R'$.

Case 1.2.1 (con_2) Suppose $P_1 \parallel Q \xrightarrow{\tau} R'$ because

$$P_1 \xrightarrow{\nu\tilde{m}'.l\triangleright} {P_1}' \quad \text{and} \quad Q \xrightarrow{\nu\tilde{m}''.m\triangleright} Q'$$

$\tilde{m}' \cap \tilde{m}'' = \tilde{m}' \cap flc(Q) = \tilde{m}'' \cap flc(P_1) = \emptyset$, and

$$R' = \nu\tilde{m}''\tilde{m}'.({P_1}' \parallel Q')_{m\oplus l}$$

From Lemma 10 and 11 we infer $m \in fl(Q')$ and $Q \equiv \nu\tilde{m}''.Q'$. Hence from above, $fl(Q') \cap (fl(P_1) \cup \tilde{m}') = \emptyset$. Since $P_1 \approx_c P_2$ there exists ${P_2}'$ such that

$$P_2 \parallel Q' \Longrightarrow {P_2}'$$

and

$$\nu\tilde{m}'.({P_1}' \parallel Q')_{l\oplus m} \approx_c {P_2}'$$

From (res_2) and (par_2) we infer

$$\nu\tilde{m}.\nu\tilde{m}''.(P_2 \parallel Q') \Longrightarrow \nu\tilde{m}.\nu\tilde{m}''.{P_2}'$$

Since $\tilde{m}'' \cap fl(P_2) = \emptyset$ we get

$$\nu\tilde{m}.(P_2 \parallel Q) \equiv \nu\tilde{m}.\nu\tilde{m}''.(P_2 \parallel Q')$$

Hence, because of Lemma 7, there exists ${P_2}''$ such that

$$\nu\tilde{m}.(P_2 \parallel Q) \Longrightarrow {P_2}''$$

and

$$\nu\tilde{m}.\nu\tilde{m}''.({P_2}' \parallel Q') \equiv {P_2}''$$

Finally, since

$$R \equiv \nu\tilde{m}.\nu\tilde{m}''.(\nu\tilde{m}'.({P_1}' \parallel Q')_{l\oplus m})$$

we obtain

$$R \equiv\mathcal{R}\equiv {P_2}'''$$

Case 1.2.2 (dis_2) The case isn't an issue since $fc(P_1) \cap fc(Q) = \emptyset$.

Case 1.2.3 The case where $P_1 \parallel Q \xrightarrow{\tau} R'$ because $P_1 \xrightarrow{\tau} {P_1}'$ (or $Q \xrightarrow{\tau} Q'$) and $R' = {P_1}' \parallel Q$ (or $R' = P_1 \parallel Q'$) is immediate.

Case 2 ($\alpha = \bar{l}$) The case is immediate because $flc(P_1) \cap flc(Q) = \emptyset$.

Case 3 ($\alpha = \nu\tilde{m}'.l\triangleright$) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\nu\tilde{m}'.l\triangleright} R$. Let $R_k \in \mathbf{N}_c$ such that $fl(R_k) \cap (fl(\nu\tilde{m}.(P_1 \parallel Q)) \cup \tilde{m}') = \emptyset$.

Case 3.1 (res_2) Suppose $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\nu\tilde{m}'.l\triangleright} R$ because $P_1 \parallel Q \xrightarrow{\nu\tilde{m}'.l\triangleright} R'$, $\tilde{m} \cap \tilde{m}' = \emptyset$, $l \notin \tilde{m}$, and $R = \nu\tilde{m}.R'$.

Case 3.1.1 (par_2) Suppose $P_1 \parallel Q \xrightarrow{\nu\tilde{m}'.l\triangleright} R'$ because $P_1 \xrightarrow{\nu\tilde{m}'.l\triangleright} {P_1}'$ and $R' = {P_1}' \parallel Q$. Because $P_1 \approx_c P_2$ there exists ${P_2}'$ such that

$$P_2 \parallel R_k \stackrel{\tau}{\Longrightarrow} {P_2}'$$

and $\nu\tilde{m}'.({P_1}' \parallel R_k)_{l\oplus k} \approx_c {P_2}'$. Hence,

$$\nu\tilde{m}.(P_2 \parallel R_k \parallel Q) \stackrel{\tau}{\Longrightarrow} \nu\tilde{m}.({P_2}' \parallel Q)$$

and since

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel R_k \equiv \nu\tilde{m}.(P_2 \parallel R_k \parallel Q)$$

then, because of Lemma 7, there exists ${P_2}''$ such that

$$\nu\tilde{m}.(P_2 \parallel Q) \parallel R_k \stackrel{\tau}{\Longrightarrow} {P_2}''$$

and $\nu\tilde{m}.({P_2}' \parallel Q) \equiv {P_2}''$. Finally, since

$$\nu\tilde{m}'.(R \parallel R_k)_{l\oplus k} \equiv \nu\tilde{m}.(\nu\tilde{m}'.({P_1}' \parallel R_k)_{l\oplus k} \parallel Q)$$

we conclude that

$$\nu\tilde{m}'.(R \parallel R_k)_{l\oplus k} \equiv\mathcal{R}\equiv {P_2}''$$

Case 3.1.2 The case where $P_1 \parallel Q \xrightarrow{\nu\tilde{m}'.l\triangleright} R'$ because $Q \xrightarrow{\nu\tilde{m}'.l\triangleright} Q'$ and $R' = P_1 \parallel Q'$ is immediate.

Case 3.2 ($open_2$) The case where $\nu\tilde{m}.(P_1 \parallel Q) \xrightarrow{\nu\tilde{m}'.l\triangleright} R$ because $P_1 \parallel Q \xrightarrow{l\triangleright} R'$, $l \in \tilde{m}$, and $R = \nu\tilde{m}'.R'$ where $\tilde{m}' = \tilde{m} \setminus \{l\}$ is similar to case 3.1 above.

□

Because $\sim_c$ ($\approx_c$) is a c-congruence and also strong (weak) reduction closed due to Lemma 13, 14, 15, and 16 it follows that $\sim_c \subseteq \simeq_c$ ($\approx_c \subseteq \cong_c$).

In order to prove $\simeq_c \subseteq \sim_c$ ($\cong_c \subseteq \approx_c$) it's sufficient to show that $\simeq_c$ ($\cong_c$) is a strong (weak) c-bisimulation.

Lemma 20. *$\cong_c$ is a weak c-bisimulation.*

Proof Similar to the proof of Lemma 18. □