Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 242 (2009) 161–183

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Calculus for Mobile Ad-hoc Networks with Static Location Binding

Jens Chr. Godskesen [1]

IT University of Copenhagen

Abstract

We present a process calculus for mobile ad hoc networks which is a natural continuation of our earlier work on the process calculus CMAN [6]. Essential to the new calculus is the novel restricted treatment of node mobility imposed by hiding of location names using a *static* binding operator, and we introduce the more general notion of unidirectional links instead of bidirectional links. We define a natural weak reduction semantics and a reduction congruence and prove our weak contextual bisimulation equivalence to be a *sound* and *complete* co-inductive characterization of the reduction congruence.
The two changes to the calculus in [6] yields a much simpler bisimulation semantics, and importantly and in contrast to [6] we manage to provide a *non-contextual* weak bisimulation congruence facilitating ease of proofs and being strictly contained in the contextual bisimulation.

Keywords: Mobile ad-hoc network, statis location binding, process calculi, CMAN

1 Introduction

The widespread use of communicating mobile devices makes mobile and wireless networks become more and more important. The area of application is broad, spanning from ambient intelligence over mobile ad hoc, sensor, and mesh networks, to cellular networks for mobile telephony.

The communication primitive for wireless networks is message broadcast. However in contrast to wired local area networks where broadcasted messages reach every node in the network, for wireless networks broadcast is *local* because messages will only reach the nodes within the communication range of the emitting node. Put differently, in wired networks broadcast scope is *transitive* and *bidirectional* in that if nodes l and m can communicate directly and if m and n can do so also, then in turn l and n can communicate directly whereas this is not necessarily the case for wireless networks.

[1] Supported by grant no. 272-05-0258 from the Danish Research Agency.

[2] Email: jcg@itu.dk

doi:10.1016/j.entcs.2009.06.018

Our work is devoted to a particular kind of wireless networks, i.e. *Mobile Ad Hoc Networks* (MANETs). MANETs are self organizing without centralized control, and they do not contain a pre-deployed infrastructure for routing messages. A MANET may be formed when a collection of nodes join together and agree on how to route messages for each other over possibly multiple hops.

In this paper we present a process calculus for MANETs which is a natural continuation and refinement of our earlier work on the calculus CMAN [6]. Essential to the new calculus is the restricted treatment of node mobility imposed by hiding of location names using a *static* binding operator, this yields a much simpler labelled transition system and bisimulation semantics. To our knowledge no other calculus for MANETs hides nodes and restricts mobility through a calculus operator. Also, we introduce the more general notion of unidirectional links instead of bidirectional links; e.g. because some nodes have larger transmission range than others. We define a natural weak reduction semantics, radically different from the one in [6], and we define a reduction congruence and prove our weak contextual bisimulation equivalence to be a *sound* and *complete* co-inductive characterization of the reduction congruence. Most importantly, and in contrast to [6], we conveniently manage to devise a non-contextual weak bisimulation congruence that is a considerably advantage in many proofs. The non-contextual bisimulation is strictly contained in our reduction congruence.

1.1 Related Work

Despite the widespread use of broadcasting technology it turns out that process calculi for broadcasting systems are not as well-studied as the more common point-to-point calculi like e.g. CCS [9], or in a mobile setting for instance the π-calculus [10,11] and the Ambient Calculus [3]. Moreover, in [5] it is demonstrated that that it is impossible to encode broadcast communication using point-to-point communication uniformly in the π-calculus.

The first study of calculi for broadcasting systems was CBS [15]. Later broadcasting was introduced in a mobile setting in $b\pi$ [4], MBS [16], and HOBS [14]. However, they all let broadcast be transitive and hence are not well suited for local wireless broadcast. More recently local wireless broadcast has been studied in CBS# [13], an extension of CBS. The neighborhood relation between nodes is dealt with letting the semantics be parameterized and quantified over a set of configurations (graphs).

The ω-calculus [17] is an extension of the π-calculus. It is interesting in that the neighborhood relation is modeled by annotating the processes with the *groups* to which they belong. A group is a set of nodes that lie within each others communication range. Local wireless broadcast has also been studied in CMN [7], here the neighborhood relation is taken care of by a metric function that tells if two physical locations are close enough to communicate. [3]

[3] The calculus CWS [8] also studies wireless broadcast but at a much lower level of abstraction, in particular they take the phenomenon of interference into account.

As mentioned above, in [6] we developed CMAN where also the neighborhood relation is explicitly part of the syntax because a node is annotated by the nodes to which it is connected. However, a bit unnaturally, and like in the ω-calculus, communication between nodes is carried out on bidirectional links. Further we assumed, as in CBS#, CMN, and the ω-calculus, that nodes may move and connect arbitrarily, but to be realistic it is easy to envisage that two particular mobile nodes in a MANET can never meet due to physical obstacles (walls, buildings, etc.). Even though we apply many changes to CMAN in this paper the analysis of a cryptographic routing protocol for MANETs carried out in [6] is still valid for our new calculus.

1.2 Motivation

Our goal in this paper is to develop a process calculus for MANETs where communication links are not assumed to be bidirectional, and moreover we want primitives that restrict the otherwise unrestricted mobility of nodes.

A *node*, $\lfloor p \rfloor_l^\sigma$, in our new calculus is modeled as a *process* p located at some logical *location* (or identity) l and connected to other nodes at locations σ. Nodes composed in parallel constitute a *network*, say

$$\lfloor p \rfloor_l^m \parallel \lfloor q \rfloor_m^n \parallel \lfloor r \rfloor_n \ , \tag{1}$$

where the node at location l, $\lfloor p \rfloor_l^m$, is connected to the node at location m, $\lfloor q \rfloor_m^n$, which in turn is connected to the node at location n, $\lfloor r \rfloor_n$.

Mobility is defined by a simple reduction, say that the node at location n in (1) autonomously moves and becomes *connected* to the node at location l,

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

Dually, nodes may arbitrarily *disconnect*, for instance m disconnects from n in

$$\lfloor p \rfloor_l^m \parallel \lfloor q \rfloor_m^n \parallel \lfloor r \rfloor_n^l \searrow \lfloor p \rfloor_l^m \parallel \lfloor q \rfloor_m \parallel \lfloor r \rfloor_n^l \ . \tag{3}$$

A process $\langle t \rangle.p$ can broadcast t and in so doing become p, and a process $(x).q$ can receive a broadcasted message t becoming $q\{t/x\}$, i.e. q with all free occurrences of x replaced by t. *Local synchronous broadcast* is defined by a network *broadcast reduction* labelled by the location of the node containing the emitting process, say

$$\lfloor \langle n \rangle.p \rfloor_l^{mk} \parallel \lfloor (x).q \rfloor_m \parallel \lfloor (x).r \rfloor_k \searrow_l \lfloor p \rfloor_l^{mk} \parallel \lfloor q\{n/x\} \rfloor_m \parallel \lfloor r\{n/x\} \rfloor_k \ , \tag{4}$$

where $\langle n \rangle.p$ broadcasts n to all nodes to which the node at l is connected.

In CMAN one may choose to hide locations in order to let broadcasting be unobservable, the hiding is carried out by a *scope extensible* binder, νl. For instance, the hidden node $\nu l.\lfloor \langle n \rangle.p \rfloor_l$, may connect to other nodes by first *extruding* its location name (through structural congruence, assuming l is fresh),

$$\nu l.\lfloor \langle n \rangle.p \rfloor_l \parallel \lfloor (x).q \rfloor_m \equiv \nu l.(\lfloor \langle n \rangle.p \rfloor_l \parallel \lfloor (x).q \rfloor_m) \searrow \nu l.(\lfloor \langle n \rangle.p \rfloor_l^m \parallel \lfloor (x).q \rfloor_m) \ ,$$

and subsequently send its messages to its new neighbor,

$$\nu l.(\lfloor \langle n \rangle.p \rfloor_l^m \parallel \lfloor (x).q \rfloor_m) \searrow \nu l.(\lfloor p \rfloor_l^m \parallel \lfloor q\{n/x\} \rfloor_m) \ ,$$

the latter carried out as an *unlabelled* and hence unobservable reduction. As a novelty, in this paper we instead introduce a *static* binder for location names, denoted by $\backslash l$, whose scope cannot be extended and we abolish the scope extensible binder mentioned above. Not only will such a binding outside its scope conceal all broadcasting events carried out at l, but also connectivity involving the node at l is restricted within the scope of the binder. For instance, in

$$P \parallel \lfloor r \rfloor_k \ , \quad \text{where} \quad P = (\lfloor \langle n \rangle.p \rfloor_l^m \parallel \lfloor (x).q \rfloor_m) \setminus l \tag{5}$$

the location name l is bound and inaccessible to the node at k, so the two nodes cannot connect and hence not directly receive messages from each other. The hidden node in (5) may broadcast to m as demonstrated by

$$P \searrow (\lfloor p \rfloor_l^m \parallel \lfloor q\{n/x\} \rfloor_m) \setminus l \ , \tag{6}$$

but then the broadcast is carried out as an unobservable unlabelled reduction.

1.3 Structure

The paper is organized as follows: Our new calculus is presented in the next section and in Section 3 we define a reduction semantics and a reduction congruence. In Section 4 we define bisimulation equivalences, one being a contextual co-inductive characterization of the reduction congruence, and one being a non-contextual congruence strictly contained in reduction congruence.

We illustrate the application of the calculus on a simple cryptographic message passing protocol where node mobility is restricted, this example could not have been modeled and analyzed in our previous work [6]. Finally, we end by a conclusion.

2 The Calculus

In this section we outline our calculus defining first terms, then processes, and finally networks.

2.1 Terms

Assume 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. Then the set of *terms* is defined by the grammar below where f is a constructor symbol with arity k. We let $\mathcal{T}$ denote the set of all terms with no variables.

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

2.2 Processes

We assume a set of process variables $\mathcal{Z}$ ranged over by z. The set of *processes* is defined by the grammar

$$p, q ::= 0 \mid \langle t\rangle.p \mid (x).p \mid \mathit{if}\ (t = s)\ \mathit{then}\ p\ \mathit{else}\ q \mid \mathit{let}\, x = t\ \mathit{in}\ p \mid$$

$$\mathit{let}\, x = g(t_1, \ldots, t_i)\ \mathit{in}\ p\ \mathit{else}\ q \mid \nu n.p \mid z \mid \mathit{rec}\ z.p$$

The processes 0, $\nu n.p$, *if* $(t = s)$ *then* p *else* q, *let* $x = t$ *in* p, and *rec* $z.p$ are standard. [4] The process $\langle t\rangle.p$ may broadcast t and in so doing become p, and $(x).p$ binds x in p and may receive a term t and replace all free occurrences of x in p by t. Often we write $\langle t\rangle$ for $\langle t\rangle.p$ when p is 0. The process *let* $x = g(t_1, \ldots, t_k)$ *in* p *else* q also binds x in p, if the destructor application $g(t_1, \ldots, t_k)$ evaluates to a term t then x is bound to t in p, otherwise the process becomes q.

We let $p\{t/x\}$ denote p where x is substituted by t. Likewise, $p\{q/z\}$ denotes p where z is substituted by q. The set of *free names* in p is denoted by $\mathit{fn}(p)$, and its *free variables* are denoted by $\mathit{fv}(p)$. A process p is *closed* if $\mathit{fv}(p) = \emptyset$. $\mathbf{P}$ denotes the set of all closed processes and 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 $\mathcal{N} \cap \mathcal{L} = \emptyset$. We let σ range over sets of location names, and we let ϵ denote the empty set. The set of *networks* is defined by the grammar

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

The network 0 denotes the empty network. $\lfloor p\rfloor_l^\sigma$ is a node at location l containing the process p and connected to all nodes in σ. $\nu n.P$ is the network P with a new name n bound by νn, $P\setminus\sigma$ denotes a network with locations in σ bound and hidden, and finally $P \parallel Q$ is the parallel composition of the two networks P and Q. We let the new name operator have higher precedence than the hiding operator which in turn has higher precedence than the left associative parallel composition. We write $\lfloor p\rfloor_l$ instead of $\lfloor p\rfloor_l^\epsilon$. When $\tilde{n} = \{n_1, \ldots, n_i\}$ we write $\tilde{n}n$ for $\tilde{n} \cup \{n\}$ and we write $\nu\tilde{n}$ instead of $\nu n_1 \ldots \nu n_i$. We write σl instead of $\sigma \cup \{l\}$, l for $\{l\}$, and $\sigma\sigma'$ for the union of disjoint sets σ and σ'.

The sets of free names, locations, and variables in P, denoted by $\mathit{fn}(P)$, $\mathit{fl}(P)$, and $\mathit{fv}(P)$ respectively, are defined as expected. We let $P\{t/x\}$ denote P where all free occurrences of x in P are substituted by t. We let $P_{l\oplus k}$ denote network P where k is added to the connections at the (free) location l, taking care that k is not bound in P (using α-conversion if needed), formally we define: $(\lfloor p\rfloor_l^\sigma)_{l\oplus k} = \lfloor p\rfloor_l^{\sigma k}$, $(\lfloor p\rfloor_m^\sigma)_{l\oplus k} = \lfloor p\rfloor_m^\sigma$, if $l \neq m$, $(\nu n.P)_{l\oplus k} = \nu n.(P_{l\oplus k})$, $(P \parallel Q)_{l\oplus k} = P_{l\oplus k} \parallel Q_{l\oplus k}$, and $(P\setminus\sigma)_{l\oplus k} = (P_{l\oplus k})\setminus\sigma$ if $l, k \notin \sigma$. We let $P_{l\oplus\sigma}$ be the obvious generalization of $P_{l\oplus k}$.

[4] We assume all free occurrences of z in p to be either input or output prefixed.

$$let\ x = t\ in\ p\ \equiv_{\mathbf{P}}\ p\{t/x\} \qquad if\ (t = t)\ then\ p\ else\ q \equiv_{\mathbf{P}} p$$

$$if\ (t = s)\ then\ p\ else\ q\ \equiv_{\mathbf{P}}\ q\ ,\quad \text{if } t \neq s \qquad rec\ z.p \equiv_{\mathbf{P}}\ p\{rec\ z.p/z\}$$

$$let\ x = g(t_1, \ldots, t_i)\ in\ p\ else\ q\ \equiv_{\mathbf{P}}\ p\{t/x\}\ ,\quad \text{if } g(t_1, \ldots, t_i) = t$$

$$let\ x = g(t_1, \ldots, t_i)\ in\ p\ else\ q\ \equiv_{\mathbf{P}}\ q\ ,\quad \text{if } g(t_1, \ldots, t_i) \text{ not defined}$$

Table 1
Structural congruence, processes.

We say that a network P is *well-formed* if each node in P is not connected to itself and if each location in P is unique. In the sequel we consider only the set of well-formed networks and we identify networks up to α-equivalence. The set of well-formed and variable closed networks is denoted by $\mathbf{N}$.

3 Reduction Semantics

In this section we provide our calculus with a natural reduction semantics; interestingly and due to the static location binder, the semantics is quite different compared to the one for CMAN.

As in the seminal work on barbed bisimulation [12], and as in [6], we strive to have as simple as possible reduction semantics and to allow an external global observer to have minimal observability, in our case: reductions $\searrow_l$ when the node at the free location l broadcasts, and reductions $\searrow$ for connections, disconnections, and broadcast from hidden nodes. In particular an observer cannot identify the broadcasted message and the receivers of the message. Indistinguishability under these observations gives rise to a natural equivalence which in turn induces a natural congruence over networks, i.e. the equivalence in all contexts closed under structural congruence.

3.1 Reductions

As usual, 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 variable closing 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. We write $C(P)$ for the insertion of P in the hole of a network context C whenever $C(P)$ is well-formed and variable closed. A relation $\mathcal{R}$ on $\mathbf{N}$ is a *congruence* if $P\ \mathcal{R}\ P'$ implies $C(P)\ \mathcal{R}\ C(Q)$ for all $C(P)$. [5] 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. The rules are standard except that new names can be extruded from nodes and pass the scope of statically bound location names.

To assist in the definition of the reduction rules we introduce a family of *ab-*

[5] Notice that any congruence, $\mathcal{R}$, has the property that $P\ \mathcal{R}\ Q$ implies $fl(P) = fl(Q)$ due to the well-formedness criteria.

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

Table 2
Structural congruence, networks.

stractions ranged over by A_σ and defined by:

$$A_\epsilon ::= 0 \qquad A_l ::= \lfloor 0 \rfloor_l^\sigma \mid \lfloor \langle t \rangle.p \rfloor_l^\sigma \mid \lfloor (x).p \rfloor_l^\sigma \qquad A_{\sigma\sigma'} ::= A_\sigma \parallel A_{\sigma'}$$
$$A_\sigma ::= A_\sigma \parallel P \mid \nu n.A_\sigma \mid A_\sigma \setminus \sigma' \ , \text{ if } \sigma \cap \sigma' = \emptyset$$

In A_σ all locations in σ are free and hence may receive messages. Given an abstraction A_σ we define $A_\sigma \circ t$, i.e. a network being the application of a term t on locations σ in A_σ, inductively by the rules in Table 3.

$$\lfloor 0 \rfloor_l^\sigma \circ t = \lfloor 0 \rfloor_l^\sigma \qquad \lfloor \langle t' \rangle.p \rfloor_l^\sigma \circ t = \lfloor \langle t' \rangle.p \rfloor_l^\sigma \qquad \lfloor (x).p \rfloor_l^\sigma \circ t = \lfloor p\{t/x\} \rfloor_l^\sigma$$
$$(A_\sigma \parallel P) \circ t = (A_\sigma \circ t) \parallel P \qquad (\nu n.A_\sigma) \circ t = \nu n.(A_\sigma \circ t) \ , \text{ if } \ n \notin fn(t)$$
$$(A_\sigma \setminus \sigma') \circ t = (A_\sigma \circ t) \setminus \sigma' \qquad (A_\sigma \parallel A_{\sigma'}) \circ t = (A_\sigma \circ t) \parallel (A_{\sigma'} \circ t)$$

Table 3
Abstraction application.

We define $\searrow_{l,t} \subseteq \mathbf{N} \times \mathbf{N}$ as the least relation closed under $\equiv$ and satisfying the rules in Table 4. Intuitively, $P \searrow_{l,t} P'$ means that the node at (the free) location l has completed broadcasting t to all nodes to which it is connected. A reduction due to rule (*emp*) describes that a node may broadcast to the empty set of receivers, whereas rule (*brd*) allows auxiliary nodes σ to be connected to a node l and let the nodes in σ synchronously receive t, whenever l has otherwise completed its broadcast of t. As an example, since $\lfloor \langle n \rangle.p \rfloor_l \searrow_{l,n} \lfloor p \rfloor_l$ we obtain

$$\lfloor \langle n \rangle.p \rfloor_l^m \parallel \lfloor (x).q \rfloor_m \searrow_{l,n} \lfloor p \rfloor_l^m \parallel \lfloor q\{n/x\} \rfloor_m \ , \tag{7}$$

from (*brd*), and from (7) and rule (*brd*) we get

$$\lfloor \langle n \rangle.p \rfloor_l^{mk} \parallel \lfloor (x).q \rfloor_m \parallel \lfloor (x).r \rfloor_k \searrow_{l,n} \lfloor p \rfloor_l^{mk} \parallel \lfloor q\{n/x\} \rfloor_m \parallel \lfloor r\{n/x\} \rfloor_k \ . \tag{8}$$

Rule (hde_1) in Table 4 allows free locations to broadcast a term.

We define $\searrow_l \subseteq \mathbf{N} \times \mathbf{N}$ as the least relation closed under $\equiv$, new name, parallel composition, and satisfying the rules in Table 4. Intuitively, $P \searrow_l P'$ means that the node at location l has completed broadcasting some message as indicated by rule (*cls*). Rule (hde_2) allows broadcast from free locations. As an example, the reduction (4) in the Introduction is inferred from (8) and rule (*cls*), and from (4) we may further infer

$$\nu n.\lfloor \langle n \rangle.p \rfloor_l^{mk} \parallel \lfloor (x).q \rfloor_m \parallel \lfloor (x).r \rfloor_k \searrow_l \nu n.(\lfloor p \rfloor_l^{mk} \parallel \lfloor q\{n/x\} \rfloor_m \parallel \lfloor r\{n/x\} \rfloor_k) \ ,$$

$$(emp)\ \ \lfloor\langle t\rangle.p\rfloor_l \searrow_{l,t} \lfloor p\rfloor_l \qquad (hde_1)\ \ \frac{P \searrow_{l,t} P'}{P \setminus \sigma \searrow_{l,t} P' \setminus \sigma}\ l \notin \sigma$$

$$(brd)\ \ \frac{P \searrow_{l,t} P'}{P_{l\oplus\sigma} \parallel A_\sigma \searrow_{l,t} P'_{l\oplus\sigma} \parallel (A_\sigma \circ t)} \qquad (cls)\ \ \frac{P \searrow_{l,t} P'}{P \searrow_l P'}$$

$$(hde_2)\ \ \frac{P \searrow_l P'}{P \setminus \sigma \searrow_l P' \setminus \sigma}\ l \notin \sigma \qquad (hde_3)\ \ \frac{P \searrow_l P'}{P \setminus \sigma \searrow P' \setminus \sigma}\ l \in \sigma$$

$$(con)\ \ \lfloor p\rfloor_l^\sigma \searrow \lfloor p\rfloor_l^{\sigma k} \qquad (dis)\ \ \lfloor p\rfloor_l^{\sigma k} \searrow \lfloor p\rfloor_l^\sigma$$

Table 4
Reduction rules.

which does not belong to the reductions in $\searrow_{l,n}$.

Finally, we define $\searrow \subseteq \mathbf{N} \times \mathbf{N}$ as the least relation closed under $\equiv$, new name, parallel composition, and location hiding, and satisfying the rules in Table 4. $P \searrow P'$ is either the result of a hidden broadcast, i.e. rule (hde_3), or a connection or disconnection as defined by the rules (con) and (dis) respectively. For instance, the reduction (6) in the Introduction is inferred from (7) and rule (hde_3), and (2) and (3) are inferred from (con) and (dis) respectively.

3.2 Reduction Congruence

Next we introduce a natural weak congruence in which reductions $\searrow_l$ are our only observables. Let $\searrow^*$ be the reflexive and transitive closure of $\searrow$. We say that a binary relation $\mathcal{R}$ on $\mathbf{N}$ is *weakly reduction-closed* if whenever $P\ \mathcal{R}\ Q$ then $P \searrow_l P'$ ($P \searrow P'$) implies the existence of some Q' such that $Q \searrow^* \searrow_l \searrow^* Q'$ ($Q \searrow^* Q'$) and $P'\ \mathcal{R}\ Q'$.

Definition 3.1 A symmetric relation $\mathcal{R}$ on $\mathbf{N}$ is a *weak reduction congruence* if it is weakly reduction-closed and a congruence.

Let $\cong$ be the largest weak reduction congruence. As an example, $P \cong Q$ if $fl(P) = fl(Q)$ and if neither P nor Q can ever broadcast since no context can distinguish them apart, in particular $0 \cong P$ if $fl(P) = \emptyset$.

4 Bisimulation Semantics

In this section we first provide a labelled transition system; interestingly the network semantics turns out much simpler than the one for CMAN. Next, we give the definition of a weak bisimulation, $\approx$, a sound and complete co-inductive characterization of $\cong$. Also this definition is quite different from the corresponding weak bisimulation for CMAN, but it is still contextual. Therefore, as a novelty compared to [6], we define a non-contextual weak bisimulation that is strictly contained in $\approx$, and we demonstrate its convenience in our examples.

4.1 Labeled Transition System Semantics

We begin with the process semantics and continue with semantics for networks.

4.1.1 Process Semantics

$$(out)\ \ \langle t\rangle.p \xrightarrow{\langle t\rangle} p \qquad (open)\ \ \frac{p \xrightarrow{\nu\tilde{n}\langle t\rangle} p'}{\nu n.p \xrightarrow{\nu\tilde{n}n\langle t\rangle} p'}\ n \in fn(t)\setminus\tilde{n}$$

$$(in_1)\ \ (x).p \xrightarrow{(t)} p\{t/x\} \qquad (in_2)\ \ \langle t'\rangle.p \xrightarrow{(t)} \langle t'\rangle.p \qquad (in_3)\ \ 0 \xrightarrow{(t)} 0$$

$$(new)\ \ \frac{p \xrightarrow{\lambda} p'}{\nu n.p \xrightarrow{\lambda} \nu n.p'}\ n \notin fn(\lambda)\cup bn(\lambda)$$

Table 5
Transition Rules, Processes.

Let the set of *process actions*, $\mathcal{A}_{\mathbf{P}}$, where $t \in \mathcal{T}$, be defined by:

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

The action (t) describes that 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 free (bound) names in λ.

The processes semantics is defined by $(\mathbf{P}, \mathcal{A}_{\mathbf{P}}, \rightarrow)$ where $\rightarrow \subseteq \mathbf{P}\times\mathcal{A}_{\mathbf{P}}\times\mathbf{P}$ is the least relation defined by the rules in Table 5 and closed by $\equiv_{\mathbf{P}}$. The rules (out) and (in_1) are immediate, and (in_2) and (in_3) state that processes may lose messages. The rule (new) is standard and the rule $(open)$ takes care of extrusion of new names.

4.1.2 Networks Semantics

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

$$\alpha ::= \bar{l}\sigma\nu\tilde{n}\langle t\rangle \mid \sigma(t) \mid \beta \qquad \beta ::= \bar{l} \mid \tau$$

where $t \in \mathcal{T}$. The action $\bar{l}\sigma\nu\tilde{n}\langle t\rangle$ means that the node at location l broadcasts t to nodes in σ where the names in $\tilde{n}$ are bound. $\sigma(t)$ means that t is received by the nodes in σ. $\bar{l}$ denotes that the broadcast session for the node at l has completed. As usual τ denotes an internal computation. We let $bn(\alpha)$ ($fn(\alpha)$) denote the bound (free) names in α, and we let $fl(\alpha)$ denote the free locations in α.

The semantics for networks is defined by $(\mathbf{N}, \mathcal{A}, \rightarrow)$ where $\rightarrow \subseteq \mathbf{N}\times\mathcal{A}\times\mathbf{N}$ is the least relation satisfying the rules in Table 6, omitting the symmetric counter parts of rules (syn) and (par). The rules (new), (hde_1), and (par) are as expected. The rule (con) deals with connectivity, and so does (dis). As an example, consider the network

$$P = \nu n.(Q \setminus m) \parallel \lfloor (x).p\rfloor_k\ , \quad Q = \lfloor\langle n\rangle.q\rfloor_l \parallel R\ , \quad R = \lfloor (x).r\rfloor_m \parallel \lfloor (x).r'\rfloor_{m'}\ .$$

$$(brd)\ \frac{p \xrightarrow{\nu\tilde{n}\langle t\rangle} p'}{\lfloor p\rfloor_l^\sigma \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} \lfloor p'\rfloor_l^\sigma} \qquad (rec_1)\ \frac{p \xrightarrow{(t)} p'}{\lfloor p\rfloor_l^\sigma \xrightarrow{l(t)} \lfloor p'\rfloor_l^\sigma} \qquad (rec_2)\ \frac{}{P \xrightarrow{\epsilon(t)} P}$$

$$(rec_3)\ \frac{P \xrightarrow{\sigma(t)} P' \quad Q \xrightarrow{\sigma'(t)} Q'}{P \parallel Q \xrightarrow{\sigma\sigma'(t)} P' \parallel Q'} \qquad (opn)\ \frac{P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'}{\nu n.P \xrightarrow{\bar{l}\sigma\nu\tilde{n}n\langle t\rangle} P'}\ n \in fn(t)\setminus\tilde{n}$$

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

$$(cls)\ \frac{P \xrightarrow{\bar{l}\epsilon\nu\tilde{n}\langle t\rangle} P'}{P \xrightarrow{\bar{l}} \nu\tilde{n}.P'} \qquad (new)\ \frac{P \xrightarrow{\alpha} P'}{\nu n.P \xrightarrow{\alpha} \nu n.P'}\ n \notin fn(\alpha)\cup bn(\alpha)$$

$$(hde_1)\ \frac{P \xrightarrow{\alpha} P'}{P\setminus\sigma \xrightarrow{\alpha} P'\setminus\sigma}\ fl(\alpha)\cap\sigma = \emptyset \qquad (hde_2)\ \frac{P \xrightarrow{\bar{l}} P'}{P\setminus\sigma \xrightarrow{\tau} P'\setminus\sigma}\ l \in \sigma$$

$$(con)\ \lfloor p\rfloor_l^\sigma \xrightarrow{\tau} \lfloor p\rfloor_l^{\sigma k} \qquad (dis)\ \lfloor p\rfloor_l^{\sigma k} \xrightarrow{\tau} \lfloor p\rfloor_l^\sigma$$

$$(par)\ \frac{P \xrightarrow{\beta} P'}{P \parallel Q \xrightarrow{\beta} P' \parallel Q}\ fl(\beta)\cap fl(Q) = \emptyset$$

Table 6
Transition Rules, Networks.

Using rules (con), (par), (hde_1), and (new) we may get

$$P \xrightarrow{\tau} \nu n.(Q_{l\oplus k}\setminus m) \parallel \lfloor (x).p\rfloor_k = P_{l\oplus k}$$

The rule (brd) states that a node may broadcast to all those nodes to which it is currently connected, (rec_1) defines when a single node can receive a message, and (rec_3) defines when multiple nodes can receive a message. Not all nodes in a parallel compostion are required to receive because of (rec_2), for instance $R \xrightarrow{m'(t)} \lfloor (x).r\rfloor_m \parallel \lfloor r'\{t/x\}\rfloor_{m'}$. The rule (syn) defines synchronization of broadcasting enforcing no name clash. For instance, assuming $n \notin fn(r)\cup fn(r')$,

$$Q_{l\oplus\{m,m'\}} \xrightarrow{\bar{l}\epsilon\langle n\rangle} \lfloor q\rfloor_l^{mm'} \parallel (\lfloor r\{n/x\}\rfloor_m \parallel \lfloor r'\{n/x\}\rfloor_{m'}) \ ,$$

so due to (cls), which closes a broadcast session, we get

$$Q_{l\oplus\{m,m'\}} \xrightarrow{\bar{l}} \lfloor q\rfloor_l^{mm'} \parallel (\lfloor r\{n/x\}\rfloor_m \parallel \lfloor r'\{n/x\}\rfloor_{m'}) \ .$$

Observe that $Q_{l\oplus m} \not\xrightarrow{\bar{l}m\langle n\rangle}$ because $m \in fl(R)$, and notice also that rule (rec_2) will allow locations m and m' to be bypassed in $Q \xrightarrow{\bar{l}} \lfloor q\rfloor_l \parallel R$. The rule (hde_2) conceals the emitter of the broadcasted message, so e.g.

$$Q_{l\oplus\{m,m'\}}\setminus l \xrightarrow{\tau} (\lfloor q\rfloor_l^{m,m'} \parallel (\lfloor r\{n/x\}\rfloor_m \parallel \lfloor r'\{n/x\}\rfloor_{m'}))\setminus l \ .$$

$$C^{\epsilon}_{l,\epsilon} \circ t = C^{\epsilon}_{l,\epsilon} \qquad C_{l,\sigma}((-) \setminus \sigma') \circ t = (C_{l,\sigma} \circ t)((-) \setminus \sigma')$$
$$C_{l,\sigma}((-) \parallel A_{\sigma'}) \circ t = (C_{l,\sigma} \circ t)((-) \parallel A_{\sigma'} \circ t)$$

Table 7
Network context application, $C_{l,\sigma}$.

Observe that $Q_{l\oplus k} \setminus k \stackrel{\bar{l}k\langle n\rangle}{\not\longrightarrow}$ because of (hde_2). The rule (opn) takes care of extrusion of bound (term) names, hence

$$P_{l\oplus k} \stackrel{\bar{l}}{\longrightarrow} \nu n.((\lfloor q \rfloor^k_l \parallel R) \setminus m \parallel \lfloor p\{n/x\} \rfloor_k) \ .$$

4.1.3 Correspondence

The correspondence between the transition semantics and structural equivalence is demonstrated by the lemma below.

Lemma 4.1 *If $P \stackrel{\alpha}{\longrightarrow} P'$ and $P \equiv Q$ then there exists Q' such that $Q \stackrel{\alpha}{\longrightarrow} Q'$ and $P' \equiv Q'$.*

and the correspondence between the transition and the reduction semantics is demonstrated by Lemma 4.2 and 4.3.

Lemma 4.2 *$P \stackrel{\bar{l}}{\longrightarrow} \equiv P'$ iff $P \searrow_l P'$.*

Lemma 4.3 *$P \stackrel{\tau}{\longrightarrow} \equiv P'$ iff $P \searrow P'$.*

4.2 Weak Contextual Bisimulation

Based on the network semantics given in the preceding section below we define our weak contextual bisimulation. First we introduce a subset of network contexts ranged over by $C^{\sigma'}_{l,\sigma}$ and defined by the grammar

$$C^{\epsilon}_{l,\epsilon} ::= (-)$$
$$C^{\sigma''}_{l,\sigma\sigma'} ::= C^{\sigma''}_{l,\sigma}((-) \parallel A_{\sigma'}) \ , \quad \text{if } l \notin fl(A_{\sigma'})$$
$$C^{\sigma'\sigma''}_{l,\sigma} ::= C^{\sigma'}_{l,\sigma}((-) \setminus \sigma'') \ , \quad \text{if } \sigma l \cap \sigma'' = \epsilon$$

Notice that σ' binds free locations of P in $C^{\sigma'}_{l,\sigma}(P)$. We write $C_{l,\sigma}$ instead of $C^{\sigma'}_{l,\sigma}$ if σ' is not important. Given $C_{l,\sigma}$ we write $C_{l,\sigma} \circ t$ for the network context being the application of t on all locations σ in $C_{l,\sigma}$ as defined in Table 7. We write $C_{l,\sigma} \circ (\tilde{n}, t, P)$ for $\nu\tilde{n}.((C_{l,\sigma} \circ t)(P))$ assuming that $\tilde{n}$ does not overlap with the free names in $C_{l,\sigma}$.

Intuitively, for all $C_{l,\sigma}(P)$, if $l \in fl(P)$ then the node at location l in P may broadcast messages to all nodes in $C_{l,\sigma}$ with locations in σ as demonstrated by the Lemma below:

Lemma 4.4 *For all $C_{l,\sigma}(P)$, if $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ then $C_{l,\sigma}(P) \xrightarrow{\bar{l}} C_{l,\sigma} \circ (\tilde{n}, t, P')$.*

$$(-) \parallel C_{l,\sigma'}(\lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma'}) \circ t = (-) \parallel (C_{l,\sigma'} \circ t)(\lfloor p\rfloor_l^{\sigma\sigma'})$$
$$\langle t\rangle D_{l,\sigma\sigma'}((-) \parallel A_{\sigma'}) \circ t = (D_{l,\sigma\sigma'} \circ t)((-) \parallel (A_{\sigma'} \circ t))$$
$$\langle t\rangle D_{l,\sigma}((-) \setminus \sigma') \circ t = (D_{l,\sigma} \circ t)((-) \setminus \sigma')$$

Table 8
Network context application, $\langle t\rangle D_{l,\sigma}$.

Also we define a set of network contexts ranged over by $\langle t\rangle D_{l,\sigma}$ and defined by the grammar:

$$\langle t\rangle D_{l,\sigma} ::= (-) \parallel C_{l,\sigma'}^{\sigma''}(\lfloor\langle t\rangle.p\rfloor_l^{\sigma\sigma'}) \ , \text{ if } \sigma \cap (\sigma'' \cup fl(C_{l,\sigma'}^{\sigma''})) = \epsilon$$

$$\langle t\rangle D_{l,\sigma} ::= \langle t\rangle D_{l,\sigma\sigma'}((-) \parallel A_{\sigma'})$$

$$\langle t\rangle D_{l,\sigma} ::= \langle t\rangle D_{l,\sigma}((-) \setminus \sigma') \ , \text{ if } \sigma l \cap \sigma' = \epsilon$$

Moreover, for a context $\langle t\rangle D_{l,\sigma}$ we write $D_{l,\sigma} \circ t$ for the context defined by the rules in Table 8. To clarify, for any $\langle t\rangle D_{l,\sigma}(P)$ if $\sigma \subseteq fl(P)$ then all nodes at locations σ in P may receive t broadcasted by the node at location l in $\langle t\rangle D_{l,\sigma}$ as illustrated by:

Lemma 4.5 *For all $\langle t\rangle D_{l,\sigma}(P)$, if $P \xrightarrow{\sigma(t)} P'$ then $\langle t\rangle D_{l,\sigma}(P) \xrightarrow{\bar{l}} (D_{l,\sigma} \circ t)(P')$.*

4.2.1 Weak Contextual Bisimulation

Making use of the two types of contexts outlined above we next define *weak contextual bisimulation*. As usual we let $\stackrel{\tau}{\Longrightarrow}$ be the reflexive and transitive closure of $\xrightarrow{\tau}$ and we define $\stackrel{\bar{l}}{\Longrightarrow}$ by $\stackrel{\tau}{\Longrightarrow}\xrightarrow{\bar{l}}\stackrel{\tau}{\Longrightarrow}$.

Definition 4.6 A symmetric relation $\mathcal{R}$ on $\mathbf{N}$ is a *weak contextual bisimulation* if $P \mathcal{R} Q$ implies

if $P \xrightarrow{\tau} P'$ *then* $\exists Q'.\ Q \stackrel{\tau}{\Longrightarrow} Q'$ *and* $P' \mathcal{R} Q'$

if $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ *then* $\forall C_{l,\sigma}(Q).\ \exists Q'.\ C_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ *and* $C_{l,\sigma} \circ (\tilde{n}, t, P') \mathcal{R} Q'$

if $P \xrightarrow{\sigma(t)} P'$ *then* $\forall \langle t\rangle D_{l,\sigma}(Q).\ \exists Q'.\langle t\rangle D_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ *and* $(D_{l,\sigma} \circ t)(P') \mathcal{R} Q'$

We let $\approx$ denote the largest weak contextual bisimulaiton.

Theorem 4.7 $\approx$ *is an equivalence relation and a congruence.*

Example 4.8 It is obvious that $\lfloor\langle t\rangle.\langle s\rangle\rfloor_l \not\approx \lfloor\langle s\rangle.\langle t\rangle\rfloor_l$ if $t \neq s$. However, similar to what is shown in [7] the order of infinite broadcast sequences may be interchanged,

i.e. whenever C binds l then

$$C(\lfloor rec\ z.\langle t\rangle.\langle s\rangle.z\rfloor_l) \approx C(\lfloor rec\ z.\langle s\rangle.\langle t\rangle.z\rfloor_l) \tag{9}$$

Intuitively, the reason why (9) holds is that receivers may disconnect from l before a term is broadcasted and connect again in order to receive next.

The first clause in Definition 4.6 is standard. The second clause says that whenever node l in P is able to broadcast to nodes σ in the environment, then when Q is placed in any such environment l in Q must complete a broadcast, but we do not know the receiving nodes. Dually, the third clause states that whenever nodes σ in P synchronously may receive a broadcasted message from the environment then when Q is placed in any such environment the emitting node must complete a broadcast, but again we may not know the actual receiving nodes. The giving up of knowing the broadcast receivers in the matching part of the two latter clauses in Definition 4.6 is related to the fact that in the observables of our reduction semantics we only know the broadcasting node, but we have no means of telling which nodes actually received the broadcasted message.

A major and non-trivial result of this paper is that weak bisimulation is a sound and complete characterization of reduction congruence.

Theorem 4.9 $\approx = \cong$.

4.3 Weak Non-Contextual Bisimulation

Because weak contextual bisimulation uses quantification over all contexts it may be hard to show equivalence between two networks, hence we provide a standard non-contextual weak bisimulation letting γ be a network action defined by the grammar:

$$\gamma ::= \bar{l}\sigma\nu\tilde{n}\langle t\rangle \ \mid\ \sigma(t) \ \mid\ \tau$$

Definition 4.10 A symmetric relation $\mathcal{R}$ on $\mathbf{N}$ is a *weak bisimulation* if $P\ \mathcal{R}\ Q$ implies

$$\textit{if } P \xrightarrow{\gamma} P' \textit{ then } \exists Q'.\ Q \stackrel{\gamma}{\Longrightarrow} Q' \textit{ and } P'\ \mathcal{R}\ Q'$$

The largest weak bisimulation, $\dot{\approx}$, is an equivalence relation and a congruence, and

Theorem 4.11 $\dot{\approx} \subset \approx$.

Notice that in contrast to weak contextual bisimulation in a weak bisimulation a matching network must output exactly the same term and also let exactly the same nodes synchronously receive a term. For instance, if f and g are two unary constructors with no destructors then $\lfloor \nu n.\nu m.\langle n\rangle.\langle m\rangle\rfloor_l$ and $\lfloor \nu n.\langle g(n)\rangle.\langle f(n)\rangle\rfloor_l$ are weak contextual bisimilar because for both two unrelated values are broadcasted that are different from any value any context can build, but clearly the two nodes are not weak bisimilar.

Weak bisimulation abstracts from connectivity in that $P_{l\oplus k} \dot{\approx} P$ because $P_{l\oplus k} \stackrel{\tau}{\Longrightarrow} P$ and $P \stackrel{\tau}{\Longrightarrow} P_{l\oplus k}$. This property is a characteristic of MANETs in that connection to any reachable node may be obtained and also it turns out useful in many proofs. The adequacy of weak bisimulation is further illustrated by Example 4.12 below which would have been quite hard to show in case of just weak contextual bisimulation. The example illustrates the use of the new feature with restricted mobility and could not have been modelled by the calculus in [6].

$$p \stackrel{\text{def}}{=} \nu n.\langle enc(pair(msg, n), key)\rangle.p' \qquad r \stackrel{\text{def}}{=} (x).\langle x\rangle.r$$
$$p' \stackrel{\text{def}}{=} (x).let\ x' = dec(x, key)\ in\ if\ (x' = n)\ then\ p\ else\ p'\ else\ p'$$
$$q \stackrel{\text{def}}{=} (x).let\ x' = dec(x, key)\ in\ let\ x'' = snd(x')\ in\ \langle enc(x'', key)\rangle.q\ else\ q$$

Table 9
A simple cryptographic message passing protocol.

Example 4.12 Suppose a node, $\lfloor p \rfloor_{l_0}$, that repetitively sends a message, *msg*, to a node, $\lfloor q \rfloor_{l_1}$. The message *msg* is re-broadcasted by p only when the reception of the previous *msg* has been acknowledged. A simple example with only one intermediary node, $\lfloor r \rfloor_{l_2}$, that can communicate with both l_0 and l_1, and where l_0 and l_1 are outside reach of each other, so they must communicate via l_2, may be defined by:

$$P = \nu\, key.(\lfloor p \rfloor_{l_0} \parallel (\lfloor r \rfloor_{l_2} \parallel \lfloor q \rfloor_{l_1}) \setminus l_1) \setminus l_2$$

where *key* is a secret symmetric key shared between p and q. Notice that only q can return the encrypted acknowledge expected by p. Further, let $pair(x, y)$ be a constructor for pairs and let *snd* be the destructor returning the second element of a pair. Also, let $enc(x, y)$ be a constructor denoting the symmetric encryption of a message x by the key y and let *dec* be the corresponding decryption destructor defined by: $dec(enc(x, y), y) = x$. We define p, q, and r in Table 9 using equations instead of recursion. Despite the risk of having a copy of *msg* forward broadcasted by each of two intermediary nodes one may show that one or two intermediary nodes will not make any observational difference, i.e. $P \dot{\approx} Q$ where

$$Q = \nu\, key.(\lfloor p \rfloor_{l_0} \parallel (\lfloor r \rfloor_{l_2} \parallel \lfloor r \rfloor_{l_3} \parallel \lfloor q \rfloor_{l_1}) \setminus l_1) \setminus \{l_2, l_3\}$$

5 Conclusion

The main contribution of this paper is the refinement of CMAN [6] to allow for restricted node mobility through the novel introduction of a *static location binder*, and also we imposed the more realistic use of *unidirectional* instead of bidirectional links. Importantly the refinement gives rise to a much simpler labelled transition system and bisimulation semantics than in [6]. Moreover, we have developed a natural reduction semantics and congruence, $\cong$, for which the largest weak contextual bisimulation, $\approx$, is a co-inductive sound and complete characterization. Most significantly and in contrast to [6] we manage to define a *non-contextual* weak bisimulation

where the largest bisimulation, $\dot{\approx}$, is strictly contained in $\approx$ and which turned out adequate in the proofs of our examples.

Several further developments of our calculus are immediate: For instance the process language could easily be extended with concurrency, and one may consider extending the language with *active substitutions* as in [1] in order to have a less contextual characterization of $\cong$. Moreover, instead of just restricting mobility of nodes we could enforce explicit mobility models as described in [2]. Also, we plan to investigate other equivalences, in particular we want to consider equivalences where the observer is mobile and has only a limited and not a global view of the whole network, and we want to investigate equivalences suitable to help reason about MANETs, and in particular routing and secure routing.

References

[1] 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.

[2] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Comm. & Mobile Comp.: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[3] L. Cardelli and A.D. Gordon. Mobile ambients. In *Foundations of Software Science and Computation Structures: First International Conference, FOSSACS '98*. Springer-Verlag, Berlin Germany, 1998.

[4] 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.

[5] Cristian Ene and Traian Muntean. Expressiveness of point-to-point versus broadcast communications. In *Fundamentals of Comp. Theory*, pages 258–268, 1999.

[6] J.C. Godskesen. A calculus for mobile ad hoc networks. In *Proceedings of the 9th International Conference, COORDINATION 2007*, volume 4467 of *LNCS*, pages 132–150, Paphos, Cyprus, June 2007. Springer–Verlag.

[7] M. Merro. An observational theory for mobile ad hoc networks. *Electron. Notes Theor. Comput. Sci.*, 173:275–293, 2007.

[8] N. Mezzetti and D. Sangiorgi. Towards a calculus for wireless systems. *Electr. Notes Theor. Comput. Sci.*, 158:331–353, 2006.

[9] R. Milner. *Communication and Concurrency*. Series in Computer Science. Prentice–Hall International, 1989.

[10] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 100:1–77, September 1992.

[11] Robin Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, May 1999.

[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. *Theoretical Computer Science*, 367(1):203–227, 2006.

[14] K. Ostrovsky, K. V. S. Prasad, and W. 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.

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

[16] K. V. S. Prasad. A prospectus for mobile broadcasting systems. *Electr. Notes Theor. Comput. Sci.*, 162:295–300, 2006.

[17] A. Singh, C.R. Ramakrishnan, and S.A. Smolka. A process calculus for mobile ad hoc networks. www.lmc.cs.sunysb.edu/~cram/Papers/SRS_OmegaCalc2006/.

6 Appendix

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

6.1 Proof of Lemma 4.1

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 [6] 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 (well-formed) parallel composition, by new name generation, and by restriction of location names (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 4.1 it is immediate that:

Corollary 6.1 *$\equiv$ is a weak bisimulation.*

6.2 Proof of Lemma 4.2 and 4.3

Below is a series of lemmas that show how the reduction and the labeled transition system semantics relate. Lemma 4.2 follows from Lemma 6.13 and 6.14, and Lemma 4.3 follows due to Lemma 6.11 and 6.12.

Lemma 6.2 *$p \xrightarrow{(t)} \equiv_{\mathbf{P}} p'$ iff for some $\tilde{n}$ where $\tilde{n} \cap fn(t) = \emptyset$ either*

i) $p \equiv_{\mathbf{P}} p' \equiv_{\mathbf{P}} \nu\tilde{n}.0$,

ii) $p \equiv_{\mathbf{P}} p' \equiv_{\mathbf{P}} \nu\tilde{n}.\langle t' \rangle.q$ *for some t' and q , or*

iii) $p \equiv_{\mathbf{P}} \nu\tilde{n}.(x).q$ *and* $p' \equiv_{\mathbf{P}} q\{t/x\}$

Proof. The 'only if' direction follows by induction in the derivation of $p \xrightarrow{(t)} p'$, and the 'if' direction follows because $\xrightarrow{\lambda}$ is closed by $\equiv_{\mathbf{P}}$ and because $\nu\tilde{n}.0 \xrightarrow{(t)} \nu\tilde{n}.0$, $\nu\tilde{n}.\langle t' \rangle.q \xrightarrow{(t)} \nu\tilde{n}.\langle t' \rangle.q$, and $\nu\tilde{n}.(x).q \xrightarrow{(t)} q\{t/x\}$ when $\tilde{n} \cap fn(t) = \emptyset$. □

Lemma 6.3 *$p \xrightarrow{\nu\tilde{n}\langle t \rangle} \equiv_{\mathbf{P}} p'$ iff $p \equiv_{\mathbf{P}} \nu\tilde{n}\tilde{n}'.\langle t \rangle.q$ and $p' \equiv_{\mathbf{P}} \nu\tilde{n}'.q$ 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 6.4 *$P \equiv Q$ implies $P_{l \oplus \sigma} \equiv Q_{l \oplus \sigma}$.*

Proof. Immediate from the rules in Table 2 and the fact that $\equiv$ is an equivalence relation, a congruence, and closed under α-conversion. □

Lemma 6.5 *$P \xrightarrow{\sigma(t)} \equiv P'$ iff $P \equiv A_\sigma$ for some A_σ and $P' \equiv A_\sigma \circ t$.*

Proof. ('only if') The proof is by induction in the inference of $P \xrightarrow{\sigma(t)} P'$. ('if') By induction on the structure of A_σ we show $A_\sigma \xrightarrow{\sigma(t)} A_\sigma \circ t$, then the rest follows due to Lemma 4.1. □

Lemma 6.6 *$P \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t \rangle} P'$ and $\sigma' \cap fl(P) = \emptyset$ implies $P_{l \oplus \sigma'} \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}\langle t \rangle} P'_{l \oplus \sigma'}$.*

Proof. By induction in the inference of $P \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t \rangle} P'$. □

Let $P_{l \ominus \sigma}$ be P where all connections at l in σ are removed.

Lemma 6.7 *$P \xrightarrow{\overline{l}\sigma\sigma'\nu\tilde{n}\langle t \rangle} P'$ implies $P_{l \ominus \sigma'} \xrightarrow{\overline{l}\sigma\nu\tilde{n}\langle t \rangle} P'_{l \ominus \sigma'}$.*

[6] We identify processes up to α-conversion.

Proof. By induction in the inference of $P \xrightarrow{\bar{l}\sigma\sigma'\nu\tilde{n}\langle t\rangle} P'$. □

Lemma 6.8 *$P \xrightarrow{\bar{l}\epsilon\langle t\rangle} P'$ implies $P \searrow_{l,t} P'$ if P contains no bound names.*

Proof. Suppose $P \xrightarrow{\bar{l}\epsilon\langle t\rangle} P'$, the proof is by induction in the inference of $P \xrightarrow{\bar{l}\epsilon\langle t\rangle} P'$.

Case Assume $\lfloor p\rfloor_l \xrightarrow{\bar{l}\epsilon\langle t\rangle} \lfloor p'\rfloor_l^\sigma$ because $p \xrightarrow{\langle t\rangle} p'$. Then, since p contains no bound names, due to Lemma 6.3, $p \equiv_{\mathbf{P}} \langle t\rangle.q$ and $p' \equiv_{\mathbf{P}} q$ for some q. Also, $\lfloor p\rfloor_l^\sigma \equiv \lfloor\langle t\rangle.q\rfloor_l$. Next, from reduction rule (*emp*) it follows that $\lfloor\langle t\rangle.q\rfloor_l \searrow_{l,t} \lfloor q\rfloor_l$. Hence, $\lfloor p\rfloor_l \searrow_{l,t} \lfloor p'\rfloor_l^\sigma$.

Case The cases where $P \xrightarrow{\bar{l}\epsilon\langle t\rangle} P'$ is due to rule (hde_1) follows by induction.

Case Suppose $P \parallel Q \xrightarrow{\bar{l}\epsilon\langle t\rangle} P' \parallel Q'$ because $P \xrightarrow{\bar{l}\sigma\langle t\rangle} P'$ and $Q \xrightarrow{\sigma(t)} Q'$. Due to Lemma 6.5, $Q \equiv A_\sigma$ and $Q' \equiv A_\sigma \circ t$ for some A_σ. Also, by Lemma 6.7, we obtain $P_{l\ominus\sigma} \xrightarrow{\bar{l}\epsilon\langle t\rangle} P'_{l\ominus\sigma}$ and hence by induction $P_{l\ominus\sigma} \searrow_{l,t} P'_{l\ominus\sigma}$. Finally, by (*brd*) and because $\searrow_{l,t}$ is closed by $\equiv$, we get $P \parallel Q \searrow_{l,t} P' \parallel Q'$.

□

Lemma 6.9 *$P \searrow_{l,t} P'$ implies $P \xrightarrow{\bar{l}\epsilon\langle t\rangle} Q$ for some Q such that $Q \equiv P'$.*

Proof. The proof is by induction in the inference of $P \searrow_{l,t} P'$. The case where $P \searrow_{l,t} P'$ is due to the rule (*empty*) is trivial. If $P \searrow_{l,t} P'$ is due to rule (*new*) or rule (res_1) the result follows by induction. Suppose, due to rule (*brd*), that

$$P_{l\oplus\sigma} \parallel A_\sigma \searrow_{l,t} P'_{l\oplus\sigma} \parallel A_\sigma \circ t$$

because $P \searrow_{l,t} P'$. By induction, $P \xrightarrow{\bar{l}\epsilon\langle t\rangle} \equiv P'$, and therefore due to Lemma 6.6 and 6.4, $P_{l\oplus\sigma} \xrightarrow{\bar{l}\sigma\langle t\rangle} \equiv P'_{l\oplus\sigma}$. Now, because $A_\sigma \xrightarrow{\sigma(t)} A_\sigma \circ t$, due to Lemma 6.5, we obtain

$$P_{l\oplus\sigma} \parallel A_\sigma \xrightarrow{\bar{l}\epsilon\langle t\rangle} \equiv P'_{l\oplus\sigma} \parallel A_\sigma \circ t$$

Finally, if $P \searrow_{l,t} P'$ is due to closing by structural congruence the result follows by induction and Lemma 4.1. □

Lemma 6.10 *$P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ implies $P \equiv \nu\tilde{n}.Q$ for some Q such that $Q \xrightarrow{\bar{l}\sigma\langle t\rangle} Q'$ and $Q' \equiv P'$ and $\tilde{n} \subseteq fn(t)$.*

Proof. By induction in the inference of $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ using Lemma 6.3. □

Lemma 6.11 *$P \xrightarrow{\bar{l}} P'$ implies $P \searrow_l P'$.*

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

The case where $P \xrightarrow{\bar{l}} P'$ is due to the rule (*par*) [7], (*new*), or (hde_1) follows by induction. Suppose $P \xrightarrow{\bar{l}} \nu\tilde{n}.P'$ because $P \xrightarrow{\bar{l}\epsilon\nu\tilde{n}\langle t\rangle} P'$. Then by Lemma 6.10 $P \equiv \tilde{n}.Q$ for some Q such that $Q \xrightarrow{\bar{l}\epsilon\langle t\rangle} Q'$ for some Q' with $P' \equiv Q'$. Next, let $Q \equiv \nu\tilde{m}.R$ for some $\tilde{m}$ and R where R contains no bound names. Then $R \xrightarrow{\bar{l}\epsilon\langle t\rangle} R'$ for some R' such that $Q' \equiv \nu\tilde{m}.R'$. Hence, due to Lemma 6.8, $R \searrow_{l,t} R'$ so also $R \searrow_l R'$, therefore $Q \searrow_l Q'$ and also $P \searrow_l \nu\tilde{n}.P'$. □

Lemma 6.12 *$P \searrow_l P'$ implies $P \xrightarrow{\bar{l}} Q$ for some Q such that $Q \equiv P'$.*

Proof. Suppose $P \searrow_l P'$. The proof is by induction in the inference of $P \searrow_l P'$.

The case where the reduction is due to closing by parallel composition or by new name generation follows by induction. If the reduction is due to closing by structural congruence the result holds by induction and because of Lemma 4.1. If $P \setminus \sigma \searrow_l P' \setminus \sigma$ because $P \searrow_l P'$ and $l \notin \sigma$ the result follows by induction. Finally, if $P \searrow_l P'$ because $P \searrow_{l,t} P'$ the lemma holds due to Lemma 6.9. □

[7] or its symmetric counterpart

Lemma 6.13 *$P \xrightarrow{\tau} P'$ 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 one of the rules (*con*) and (*dis*) is immediate. If $P \xrightarrow{\tau} P'$ is due to rule (hde_2) the result follows due to Lemma 6.11 and the reduction rule (hde_3). Finally, if $P \xrightarrow{\tau} P'$ follows by one of the rules (*par*) (or its symmetric counterpart), (*new*) or (hde_1) the lemma holds because $\searrow$ is closed by parallel composition, new names, and hiding of location names. □

Lemma 6.14 *$P \searrow P'$ implies $P \xrightarrow{\tau} Q$ for some Q such that $Q \equiv P'$.*

Proof. Suppose $P \searrow P'$. The proof is by induction in the inference of $P \searrow P'$.

The case where $P \searrow P'$ is due to rule (*con*) or (*dis*) is immediate. If $P \searrow P'$ is because of the rule (hde_3) the result follows due to Lemma 6.12 and the lts-rule (hde_2). The closing by parallel composition, new names, and hiding of location names follows by induction and the lts-rules (*par*) and its symmetric counterpart, (*new*), and (hde_1) respectively. Finally, the closing by $\equiv$ follows by Lemma 4.1. □

6.3 Proof of Lemma 4.4 and 4.5

The proof of Lemma 4.4 is by induction on the structure of $C_{l,\sigma}$, and the proof of Lemma 4.5 follows by structural induction on $\langle t\rangle D_{l,\sigma}$ with the help of Lemma 4.4.

6.4 Bisimulation up to

Definition 6.15 Let $\mathcal{R}$ be binary relations on $\mathbf{N}$. Then $\mathcal{R}$ is a *weak simulation up to* $\equiv$ if $P\ \mathcal{R}\ Q$ implies

if $P \xrightarrow{\tau} P'$ *then* $\exists Q'.\ Q \stackrel{\tau}{\Longrightarrow} Q'$ *and* $P' \equiv\mathcal{R}\equiv Q'$

if $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ *then* $\forall C_{l,\sigma}(Q).\ \exists Q'.\ C_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ *and* $C_{l,\sigma} \circ (\tilde{n}, t, P') \equiv\mathcal{R}\equiv Q'$

if $P \xrightarrow{\sigma(t)} P'$ *then* $\forall \langle t\rangle D_{l,\sigma}(Q).\ \exists Q'.\langle t\rangle D_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ and $(D_{l,\sigma} \circ t)(P') \equiv\mathcal{R}\equiv Q'$

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

Lemma 6.16 *If $\mathcal{R}$ is a weak bisimulation up to $\equiv$ then $\equiv\mathcal{R}\equiv$ is a weak bisimulation.*

Proof. Suppose $\mathcal{R}$ is a weak bisimulation up to $\equiv$. We only show that $\equiv\mathcal{R}\equiv$ is a weak simulation [8], the proof of $(\equiv\mathcal{R}\equiv)^{-1}$ being a weak simulation is similar.

Let $P \equiv P_1\ \mathcal{R}\ Q_1 \equiv Q$. Suppose $P \xrightarrow{\alpha} P'$. We only consider the case where $\alpha = \bar{l}\sigma\nu\tilde{n}\langle t\rangle$, the other cases are immediate or similar.

If $P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P'$ then, due to Lemma 4.1, there exists $P_1{}'$ such that $P_1 \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P_1{}'$ and $P' \equiv P_1{}'$. Then, since $P_1\ \mathcal{R}\ Q_1$, for all $C_{l,\sigma}$ there exists $Q_1{}'$ such that $C_{l,\sigma}(Q_1) \stackrel{\bar{l}}{\Longrightarrow} Q_1{}'$ and $.C_{l,\sigma} \circ (\nu\tilde{n}, t, P_1{}') \equiv\mathcal{R}\equiv Q_1{}'$. Because $\equiv$ is a congruence we have $C_{l,\sigma} \circ (\nu\tilde{n}, t, P') \equiv C_{l,\sigma} \circ (\nu\tilde{n}, t, P_1{}')$ and $C_{l,\sigma}(Q) \equiv C_{l,\sigma}(Q_1)$. From Lemma 4.1 we infer that there exists Q' such that $C_{l,\sigma}(Q) \stackrel{\tau}{\Longrightarrow} Q'$ and $Q_1{}' \equiv Q'$. Hence, since $\equiv$ is transitive, $C_{l,\sigma} \circ (\nu\tilde{n}, t, P') \equiv\mathcal{R}\equiv Q'$. □

6.5 Proof of Theorem 4.7

That $\approx$ is a congruence follows from the lemmas below.

Lemma 6.17 *For any $C^{\sigma_0}_{l,\sigma}(P \parallel C^{\sigma_1}_{l,\sigma'}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'\sigma''}_l))$ there exists $\langle t\rangle_l D_{\sigma''}$ such that*

$$C^{\sigma_0}_{l,\sigma}(P \parallel C^{\sigma_1}_{l,\sigma'}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'\sigma''}_l)) = \langle t\rangle_l D_{\sigma''}(P)$$

and

$$(C^{\sigma_0}_{l,\sigma} \circ t)(P \parallel (C^{\sigma_1}_{l,\sigma'} \circ t)(\lfloor p\rfloor^{\sigma\sigma'\sigma''}_l)) = (D_{\sigma''} \circ t)(P)$$

if $\sigma\sigma'' \cap (\sigma_1 \cup fl(C^{\sigma_1}_{l,\sigma'}) = \emptyset$.

Proof. The proof is by induction on the structure of $C^{\sigma_0}_{l,\sigma}$. □

[8] We here assume a definition of weak bisimulation, similar to the one for weak bisimulation up to $\equiv$, such that $\mathcal{R}$ is a weak bisimulation if both $\mathcal{R}$ and $\mathcal{R}^{-1}$ is a weak simulation.

Lemma 6.18 *$P \xrightarrow{\bar{l}\sigma\langle t\rangle} P'$ implies $P \equiv \nu\tilde{n}.C^{\sigma''}_{l,\sigma'}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'}_l)$ for some p, $\tilde{n}$, and $C^{\sigma''}_{l,\sigma'}$ with $\sigma \cap (\sigma'' \cup fl(P)) = \emptyset$, $\tilde{n} \cap fn(t) = \emptyset$, and $P' \equiv \nu\tilde{n}.(C^{\sigma''}_{l,\sigma'} \circ t)(\lfloor p\rfloor^{\sigma\sigma'}_l)$.*

Proof. By induction in the inference of $P \xrightarrow{\bar{l}\sigma\langle t\rangle} P'$. □

Lemma 6.19 *$C^{\sigma''}_{l,\sigma'}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'}_l) \xrightarrow{\bar{l}\sigma\langle t\rangle} (C^{\sigma''}_{l,\sigma'} \circ t)(\lfloor p\rfloor^{\sigma\sigma'}_l)$ if $\sigma \cap (\sigma'' \cup fl(C^{\sigma''}_{l,\sigma'})) = \emptyset$.*

Proof. By induction on the structure of $C^{\sigma''}_{l,\sigma'}$. □

Lemma 6.20 *$P \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} \equiv P'$ iff*

$$P \equiv \nu\tilde{n}\tilde{n}'.C^{\sigma''}_{l,\sigma'}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'}_l)$$

and

$$P' \equiv \nu\tilde{n}'.(C^{\sigma''}_{l,\sigma'} \circ t)(\lfloor p\rfloor^{\sigma\sigma'}_l)$$

for some p, $\tilde{n}'$, and $C^{\sigma''}_{l,\sigma'}$ with $\sigma \cap (\sigma'' \cup fl(P)) = \emptyset$, $\tilde{n} \subseteq fn(t)$, $\tilde{n}' \cap fn(t) = \emptyset$.

Proof. The 'only if' direction follows from Lemma 6.10 and 6.18, and the 'if' direction follows due to Lemma 4.1 and 6.19. □

Lemma 6.21 *$P \approx Q$ implies $C(P) \approx C(Q)$*

Proof. Let $\mathcal{R} = \{(C(P), C(Q)) \mid P \approx Q\}$. It is sufficient, due to Lemma 6.16, to prove that $\mathcal{R}$ is a weak bisimulation up to $\equiv$. We only show $\mathcal{R}$ is a weak simulation up to $\equiv$. The proof of $\mathcal{R}^{-1}$ being a weak simulation up to $\equiv$ is similar. Let $C(P)\ \mathcal{R}\ C(Q)$, the proof is by induction on the structure of C.

Case 1 ($C = (-)$) : Immediate.

Case 2 ($C = C' \parallel R$) : The proof is by induction in the derivation of $C'(P) \parallel R \xrightarrow{\alpha} P' \parallel R'$.

Case 2.1 ($\alpha = \tau$)**:** Assume

$$C'(P) \parallel R \xrightarrow{\tau} P' \parallel R' \tag{11}$$

- The case where $C'(P) \xrightarrow{\tau} P'$ and $R = R'$ follows by induction.
- The case where $R \xrightarrow{\tau} R'$ and $C'(P) = P'$ is trivial.

Case 2.2 ($\alpha = \bar{l}\sigma\nu\tilde{n}\langle t\rangle$)**:** Assume

$$C'(P) \parallel R \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P' \parallel R' \tag{12}$$

- Suppose (12) is due to $C'(P) \xrightarrow{\bar{l}\sigma\sigma'\nu\tilde{n}\langle t\rangle} P'$ and $R \xrightarrow{\sigma'(t)} R'$ where $\tilde{n} \cap fn(R) = \emptyset$ and $\sigma \cap fl(R) = \emptyset$. Due to Lemma 6.5, $R \equiv A_{\sigma'}$ and $R' \equiv A_{\sigma'} \circ t$ for some $A_{\sigma'}$. For all $C_{l,\sigma}(C'(Q) \parallel A_{\sigma'})$ there exists $C_{l,\sigma\sigma'}$ such that

  $$C_{l,\sigma\sigma'}(C'(Q)) = C_{l,\sigma}(C'(Q) \parallel A_{\sigma'})$$

  Then, by induction $C'(P)\ \mathcal{R}\ C'(Q)$ and there exists Q' such that $C_{l,\sigma\sigma'}(C'(Q)) \stackrel{\bar{l}}{\Longrightarrow} Q'$ and $C_{l,\sigma\sigma'} \circ (\tilde{n}, t, P') \equiv \mathcal{R} \equiv Q'$. Because $\equiv$ is a congruence we obtain $C_{l,\sigma}(C'(Q) \parallel R) \stackrel{\bar{l}}{\Longrightarrow} Q''$ for some Q'' where $Q'' \equiv Q'$ from Lemma 4.1. Finally, since

  $$(C_{l,\sigma\sigma'} \circ t)(P') = (C_{l,\sigma} \circ t)(P' \parallel A_{\sigma'} \circ t)$$

  we have $C_{l,\sigma} \circ (\tilde{n}, t, P' \parallel R') \equiv \mathcal{R} \equiv Q''$.

- Suppose (12) is due to $C'(P) \xrightarrow{\sigma'(t)} P'$ and $R \xrightarrow{\bar{l}\sigma\sigma'\nu\tilde{n}\langle t\rangle} R'$ where $\tilde{n} \cap fn(C'(P)) = \emptyset$ and $\sigma \cap fl(C'(P)) = \emptyset$. From Lemma 6.20 we infer,

  $$R \equiv \nu\tilde{n}\tilde{n}'.C^{\sigma_0}_{l,\sigma''}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'\sigma''}_l)$$

  and

  $$R' \equiv \nu\tilde{n}'.(C^{\sigma_0}_{l,\sigma''} \circ t)(\lfloor p\rfloor^{\sigma\sigma'\sigma''}_l)$$

  for some p, $\tilde{n}'$, and $C^{\sigma_0}_{l,\sigma''}$ with $\sigma\sigma' \cap fl(C^{\sigma_0}_{l,\sigma''}) = \emptyset$, $\sigma\sigma' \cap \sigma_0 = \emptyset$, and $\tilde{n}' \cap fn(t) = \emptyset$. For any $C^{\sigma_1}_{l,\sigma}(C'(Q) \parallel R)$, assuming (using α-conversion if needed) $\tilde{n}\tilde{n}' \cap fn(C^{\sigma_1}_{l,\sigma}(C'(Q))) = \emptyset$, we have

  $$C^{\sigma_1}_{l,\sigma}(C'(Q) \parallel R) \equiv \nu\tilde{n}\tilde{n}'.C^{\sigma_1}_{l,\sigma}(C'(Q) \parallel C^{\sigma_0}_{l,\sigma''}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'\sigma''}_l))$$

From Lemma 6.17 we infer

$$C^{\sigma_1}_{l,\sigma}(C'(Q) \parallel C^{\sigma_0}_{l,\sigma''}(\lfloor\langle t\rangle.p\rfloor^{\sigma\sigma'\sigma''}_l)) = \langle t\rangle D_{l,\sigma'}(C'(Q))$$

for some $\langle t\rangle D_{l,\sigma'}$. By induction there exists Q' such that

$$\langle t\rangle D_{l,\sigma'}(C'(Q)) \stackrel{\bar{l}}{\Longrightarrow} Q'$$

and $(D_{l,\sigma'} \circ t)(P') \equiv \mathcal{R} \equiv Q'$. Hence, because

$$\nu\tilde{n}\tilde{n}'.\langle t\rangle D_{l,\sigma'}(C'(Q)) \equiv C^{\sigma_1}_{l,\sigma}(C'(Q) \parallel R)$$

we obtain, due to Lemma 4.1, $C^{\sigma_1}_{l,\sigma}(C'(Q) \parallel R) \stackrel{\bar{l}}{\Longrightarrow} Q''$ for some Q'' with $Q'' \equiv \nu\tilde{n}\tilde{n}'.Q'$. From Lemma 6.17 we have

$$(C^{\sigma_1}_{l,\sigma} \circ t)(P' \parallel (C^{\sigma_0}_{l,\sigma''} \circ t)(\lfloor p\rfloor^{\sigma\sigma'\sigma''}_l)) = (D_{\sigma'} \circ t)(P')$$

and finally we obtain $C^{\sigma_1}_{l,\sigma} \circ (\tilde{n}, t, P' \parallel R') \equiv \mathcal{R} \equiv Q''$.

Case 2.3 ($\alpha = \sigma(t)$)**:** Suppose

$$C'(P) \parallel R \stackrel{\sigma(t)}{\longrightarrow} P' \parallel R'$$

is due to $C'(P) \stackrel{\sigma_1(t)}{\longrightarrow} P'$, $R \stackrel{\sigma_2(t)}{\longrightarrow} R'$, where $\sigma = \sigma_1\sigma_2$. From Lemma 6.5 it follows that $R \equiv A_{\sigma_2}$ and $R' \equiv A_{\sigma_2} \circ t$ for some A_{σ_2}. For any $\langle t\rangle D_{l,\sigma}(C'(Q) \parallel A_{\sigma_2})$ there exists $\langle t\rangle D_{l,\sigma_1}$ such that

$$\langle t\rangle D_{l,\sigma}(C'(Q) \parallel A_{\sigma_2}) = \langle t\rangle D_{l,\sigma_1}(C'(Q))$$

By induction there exists Q' such that $\langle t\rangle D_{l,\sigma_1}(C'(Q)) \stackrel{\bar{l}}{\Longrightarrow} Q'$ and

$$(D_{\sigma_1} \circ t)(P') \equiv \mathcal{R} \equiv Q'$$

But then since

$$\langle t\rangle D_{l,\sigma_1}(C'(Q)) \equiv \langle t\rangle D_{l,\sigma}(C'(Q) \parallel R)$$

we obtain $\langle t\rangle D_{l,\sigma}(C'(Q) \parallel R) \stackrel{\bar{l}}{\Longrightarrow} Q''$ for some Q'' with $Q'' \equiv Q'$ due to Lemma 4.1 and because

$$(D_{l,\sigma} \circ t)(P' \parallel R') \equiv (D_{l,\sigma_1} \circ t)(P')$$

finally we get $(D_{l,\sigma} \circ t)(P' \parallel R') \equiv \mathcal{R} \equiv Q''$.

Case 3 ($C = C' \setminus \sigma$) : Similar to the case above

Case 4 ($C = \nu n.C'$) : Follows easily by induction.

□

Lemma 6.22 *(Process Normal Form) For any process $p \in \mathcal{P}$, $p \equiv_{\mathcal{P}} \nu\tilde{n}.q$ where $q = 0$, $q = (x).q'$, or $q = \langle t\rangle.q'$ for some $\tilde{n}$, q', x, and t.*

Proof. The proof is by induction on the structure of p. □

Lemma 6.23 *(Network Normal Form) For all P and for all $\sigma \subseteq fl(P)$ there exists A_σ such that $P \equiv A_\sigma$.*

Proof. The proof is by induction on the structure of P with help of Lemma 6.22 □

Lemma 6.24 *$P \approx Q$ implies $fl(P) = fl(Q)$.*

Proof. Assume $fl(P) \neq fl(Q)$. Without loss of generality, $l \in fl(P) \setminus fl(Q)$ for some l. Due to Lemma 6.23 and 6.5, $P \stackrel{\sigma(t)}{\longrightarrow} P'$ for some σ and P' with $l \notin \sigma$. Suppose, in order to obtain a contradiction, that $P \approx Q$. Observe that there exists some $\langle t\rangle D_{l,\sigma}$ such that $\langle t\rangle D_{l,\sigma}(Q)$ is well-defined. Then since for all $\langle t\rangle D_{l,\sigma}(Q)$ there must exists some Q' such that $\langle t\rangle D_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ and $(D_{l,\sigma} \circ t)(P') \approx Q'$ we obtain a contradiction because no $(D_{l,\sigma} \circ t)(P')$ is well-defined as $l \in fl(P')$. Consequently it must be that $P \not\approx Q$. □

Theorem 6.25 *$\approx$ is a congruence.*

Proof. Suppose $P \approx Q$ and $C(P)$ is well-defined for some context C. Then, because of Lemma 6.24, we know $fl(P) = fl(Q)$ and hence also $C(Q)$ is well-defined. Then remaining part of the proof follows due to Lemma 6.21. □

Theorem 6.26 *$\approx$ is an equivalence relation*

Proof. Reflexivity holds due to Lemma 4.4 and 4.5, symmetry follows by definition of $\approx$, and transitivity holds due to Theorem 6.25. □

6.6 Proof of Theorem 4.9

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 6.11, 6.12, 6.13, and 6.14 and Corollary 6.1.

The remaining part of the proof establishes that $\cong \subseteq \approx$. It's sufficient to show that $\cong$ is a weak bisimulation. Let $P_1 \cong P_2$.

Case 1 ($\alpha = \tau$)**:** The case where $P_1 \xrightarrow{\tau} P_1{}'$ is immediate due to Lemma 6.13 and 6.14.

Case 2 ($\alpha = \bar{l}\sigma\nu\tilde{n}\langle t\rangle$)**:** Suppose $P_1 \xrightarrow{\bar{l}\sigma\nu\tilde{n}\langle t\rangle} P_1{}'$. Due to Lemma 4.4,

$$C_{l,\sigma}(P_1) \xrightarrow{\bar{l}} C_{l,\sigma} \circ (\tilde{n}, t, P_1{}')$$

for all $C_{l,\sigma}(P_1)$. Hence, because of Lemma 6.11,

$$C_{l,\sigma}(P_1) \searrow_l C_{l,\sigma} \circ (\tilde{n}, t, P_1{}')$$

Now, because $\cong$ is a congruence and weak reduction closed there exists $P_2{}'$ such that

$$C_{l,\sigma}(P_2) \searrow^* \searrow_l \searrow^* P_2{}'$$

and $C_{l,\sigma} \circ (\tilde{n}, t, P_1{}') \cong P_2{}'$. Then due to Lemma 4.1, 6.12, and 6.14 it follows that $C_{l,\sigma}(P_2) \stackrel{\bar{l}}{\Longrightarrow} \equiv P_2{}'$. The remaining part of the proof follows since $\equiv \subseteq \cong$.

Case 3 ($\alpha = \sigma(t)$)**:** Suppose $P_1 \xrightarrow{\sigma(t)} P_1{}'$. Due to Lemma 4.5,

$$\langle t\rangle D_{l,\sigma}(P_1) \xrightarrow{\bar{l}} (D_{l,\sigma} \circ t)(P_1{}')$$

for all $\langle t\rangle D_{l,\sigma}(P_1)$. Hence, because of Lemma 6.11,

$$\langle t\rangle D_{l,\sigma}(P_1) \searrow_l (D_{l,\sigma} \circ t)(P_1{}')$$

Now, because $\cong$ is a congruence and weak reduction closed there exists $P_2{}'$ such that

$$\langle t\rangle D_{l,\sigma}(P_2) \searrow^* \searrow_l \searrow^* P_2{}'$$

and $(D_{l,\sigma} \circ t)(P_1{}') \cong P_2{}'$. Then due to Lemma 4.1, 6.12, and 6.14 it follows that $\langle t\rangle D_{l,\sigma}(P_2) \stackrel{\bar{l}}{\Longrightarrow} \equiv P_2{}'$. The remaining part of the proof follows since $\equiv \subseteq \cong$.

6.7 Proofs for $\dot{\approx}$ being an equivalence relation and a congruence

Theorem 6.27 $\dot{\approx}$ *is an equivalence relation.*

Proof. Standard. □

Lemma 6.28 *$P \dot{\approx} Q$ implies $C(P) \dot{\approx} C(Q)$*

Proof. Let $\mathcal{R} = \{(C(P), C(Q)) \mid P \dot{\approx} Q\}$. It is sufficient to prove that $\mathcal{R}$ is a weak bisimulation. To show that $\mathcal{R}$ is a weak simulation, let $C(P)\ \mathcal{R}\ C(Q)$ and suppose $C(P) \xrightarrow{\alpha} P'$. The proof is a straightforward outer induction on the structure of C and a inner induction in the derivation of $C(P) \xrightarrow{\alpha} P'$. The proof of $\mathcal{R}^{-1}$ being a weak simulation is similar. □

Theorem 6.29 $\dot{\approx}$ *is a congruence.*

Proof. Due to the clause about input in the definition of weak bisimulation it is immediate that $P \dot{\approx} Q$ implies $fl(P) = fl(Q)$. Hence whenever $P \dot{\approx} Q$ and $C(P)$ is well-defined then also $C(Q)$ is well-defined. The remaining part of the proof then follows from Lemma 6.28. □

6.8 Proof of Theorem 4.11

In order to show $\dot{\approx} \subseteq \approx$ it is sufficient to prove that $\dot{\approx}$ is a weak bisimulation. We show $\dot{\approx}$ to be a weak simulation. The proof of $(\dot{\approx})^{-1}$ being a weak simulation is similar.

Suppose $P \dot{\approx} Q$.

Suppose $P \xrightarrow{\bar{l}\sigma\tilde{n}\langle t\rangle} P'$ for some l, σ, $\tilde{n}$, t, and P'. In that case we must show that for all $C_{l,\sigma}(Q)$ there exists Q' such that $C_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ and $C_{l,\sigma} \circ (\tilde{n}, t, P') \dot{\approx} Q'$. The proof follows due to Lemma 4.4 and because $P \dot{\approx} Q$ implies $Q \stackrel{\bar{l}\sigma\tilde{n}\langle t\rangle}{\Longrightarrow} Q''$ for some Q'' such that $P' \dot{\approx} Q''$.

Suppose $P \xrightarrow{\sigma(t)} P'$ for some σ and t. In that case we must show that for all $\langle t\rangle D_{l,\sigma}(Q)$ there exists Q' such that $\langle t\rangle D_{l,\sigma}(Q) \stackrel{\bar{l}}{\Longrightarrow} Q'$ and $(D_{l,\sigma} \circ t)(P') \dot{\approx} Q'$. The proof follows due to Lemma 4.5 and because $P \dot{\approx} Q$ implies $Q \stackrel{\sigma(t)}{\Longrightarrow} Q''$ for some Q'' such that $P' \dot{\approx} Q''$.

The case where $P \xrightarrow{\tau} P'$ is trivial.

In order to show $\dot{\approx} \subset \approx$ let f and g be two unary constructors with no destructors and let $P = \nu n.\nu m.\lfloor\langle n\rangle.\langle m\rangle\rfloor_l$ and $Q = \nu n.\lfloor\langle g(n)\rangle.\langle f(n)\rangle\rfloor_l$ then $P \approx Q$ because in both P and Q the outputs are two unrelated values that are different from any value any context can build, but clearly $P \not\dot{\approx} Q$.

6.9 Proof of Example 4.8

In order to show (9) define the family of (parameterized) processes

$$P^{\sigma}_{t,s} \stackrel{\text{def}}{=} \lfloor rec\ x.\langle t\rangle.\langle s\rangle.x\rfloor^{\sigma}_{l} \qquad Q^{\sigma}_{t,s} \stackrel{\text{def}}{=} \lfloor\langle t\rangle.rec\ x.\langle s\rangle.\langle t\rangle.x\rfloor^{\sigma}_{l}$$

and let

$$\mathcal{R}^{\sigma}_{s,t} = \{(C(P^{\sigma}_{t,s}), C(P^{\sigma}_{s,t})), (C(Q^{\sigma}_{s,t}), C(P^{\sigma}_{s,t})), (C(P^{\sigma}_{t,s}), C(Q^{\sigma}_{t,s})) \mid C \text{ binds } l\}$$

Then $\mathcal{R}_{s,t} = \bigcup_{\sigma} \mathcal{R}^{\sigma}_{s,t}$ is a weak contextual bisimulation. Consider for instance $C(P^{\sigma}_{t,s})\ \mathcal{R}\ C(P^{\sigma}_{s,t})$. If $C(P^{\sigma}_{t,s}) \xrightarrow{\tau} C'(Q^{\sigma}_{s,t})$ then intuitively, we let the process at location l in $C(P^{\sigma}_{s,t})$ completely disconnect from other nodes and then let it broadcast s to an empty set of receivers, afterwards we then let l connect to all nodes in σ again after which it can broadcast t.

6.10 Proof of Example 4.12

$\mathcal{R}$, the least relation on $\mathbf{N}$ such that for all $\sigma_0, \sigma_1, \sigma_1{}', \sigma_2$:

$$\begin{array}{llll} P_0^{\sigma_0,\sigma_1,\sigma_2}\ \mathcal{R}\ Q_{0i}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} & \text{and} & P_4^{\sigma_0,\sigma_1,\sigma_2}\ \mathcal{R}\ Q_{4i}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} & \text{for } i = 0,\ldots,4 \\ P_1^{\sigma_0,\sigma_1,\sigma_2}\ \mathcal{R}\ Q_{1i}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} & \text{and} & P_3^{\sigma_0,\sigma_1,\sigma_2}\ \mathcal{R}\ Q_{3i}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} & \text{for } i = 0,\ldots,5 \\ P_2^{\sigma_0,\sigma_1,\sigma_2}\ \mathcal{R}\ Q_{2i}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} & \text{for } i = 0,\ldots,2 & & \end{array}$$

is a weak bisimulation containing $P \dot{\approx} Q$ where

$$\begin{aligned} P_0^{\sigma_0,\sigma_1,\sigma_2} &= (\lfloor p\rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r\rfloor_{l_2}^{\sigma_1} \parallel \lfloor q\rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2\} \\ P_1^{\sigma_0,\sigma_1,\sigma_2} &= (\lfloor p'\rfloor_{l_0}^{\sigma_0} \parallel (\lfloor\langle t_1\rangle.r\rfloor_{l_2}^{\sigma_1} \parallel \lfloor q\rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2\} \\ P_2^{\sigma_0,\sigma_1,\sigma_2} &= (\lfloor p'\rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r\rfloor_{l_2}^{\sigma_1} \parallel \lfloor\langle t_2\rangle.q\rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2\} \\ P_3^{\sigma_0,\sigma_1,\sigma_2} &= (\lfloor p'\rfloor_{l_0}^{\sigma_0} \parallel (\lfloor\langle t_2\rangle.r\rfloor_{l_2}^{\sigma_1} \parallel \lfloor q\rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2\} \\ P_4^{\sigma_0,\sigma_1,\sigma_2} &= (\lfloor p'\rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r\rfloor_{l_2}^{\sigma_1} \parallel \lfloor q\rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2\} \end{aligned}$$

with $t_1 = enc((msg, n), key)$, $t_2 = enc(n, key)$, and let, where $n \notin fn(t, t')$,

$$
\begin{aligned}
Q_{00}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{01}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{02}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{03}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t' \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{04}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t' \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\}
\end{aligned}
$$

$$
\begin{aligned}
Q_{10}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{11}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t_1 \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{12}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{13}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{14}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t_1 \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{15}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t' \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\}
\end{aligned}
$$

$$
\begin{aligned}
Q_{20}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t_2 \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{21}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_1 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t_2 \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{22}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t_2 \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\}
\end{aligned}
$$

$$
\begin{aligned}
Q_{30}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_2 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{31}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_2 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t_1 \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{32}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_2 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t_2 \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{33}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_2 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t_2 \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{34}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_2 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{35}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t_2 \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\}
\end{aligned}
$$

$$
\begin{aligned}
Q_{40}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{41}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{42}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{43}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor \langle t' \rangle.q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\} \\
Q_{44}^{\sigma_0,\sigma_1,\sigma_1{}',\sigma_2} &= (\lfloor p' \rfloor_{l_0}^{\sigma_0} \parallel (\lfloor \langle t \rangle.r \rfloor_{l_2}^{\sigma_1} \parallel \lfloor \langle t' \rangle.r \rfloor_{l_3}^{\sigma_1{}'} \parallel \lfloor q \rfloor_{l_1}^{\sigma_2}) \setminus \{l_1\}) \setminus \{l_2, l_3\}
\end{aligned}
$$