Formal Aspects of Computing (1996) 3: 1–000

An Algebraic Verification of a Mobile Network[1]

Fredrik Orava and Joachim Parrow

Swedish Institute of Computer Science, Box 1263, S-16428 Kista, Sweden
and
Department of Computer Systems, Uppsala University
email: fredrik@sics.se, joachim@sics.se

Keywords: Mobile networks; Algebraic verification; Process algebra; Communication protocols

Abstract. In a mobile communication network some nodes change locations, and are therefore connected to different other nodes at different points in time. We show how some important aspects of such a network can be formally defined and verified using the π-calculus, which is a development of CCS (Calculus of Communicating Systems) allowing port names to be sent as parameters in communication events. As an example of a mobile network we consider the Public Land Mobile Network currently being developed by the European Telecommunication Standards Institute and concentrate on the handover procedure which controls the dynamic topology of the network.

1. Introduction

The need for mathematically rigorous definitions of communication protocol standards is today widely acknowledged. Such definitions are needed to specify protocols and services, and to verify that the protocols fulfil their services. Unfortunately most protocols still rely on informal definitions. One reason for this is that many aspects of protocols are not described in a precise and yet abstract enough way. The purpose of this paper is to demonstrate how one such aspect, *mobility* of nodes, can be directly expressed and verified in the rigorous

[1] An early version of part of this material appeared in *Protocol Specification, Testing, and Verification X*, pages 275–291, North-Holland 1990.
Correspondence and offprint requests to: Fredrik Orava, Swedish Institute of Computer Science, Box 1263, S-16428 Kista, Sweden, email: fredrik@sics.se

mathematical framework of the π-calculus, a process calculus where mobility is one of the basic concepts.

A typical process calculus such as CCS [Mil89] contains an algebraic language for descriptions of processes in terms of the *communication actions* they can perform. A communication action occurs on a specific port, and may carry a data value from one process to another; typical constructs in the language are

$$\overline{a}v \,.\, P \qquad \text{and} \qquad a(x) \,.\, P$$

meaning "output the value v on port a" and "input a value to the variable x from port a" respectively. With additional constructs for choice and recursion it is possible to describe the behaviour of the entities of many protocols, and also to describe the expected services. A formal protocol verification amounts to proving that the parallel composition of the entities is *equivalent* to the service. The relevant equivalence can be defined operationally by interpreting the language into labelled transition systems, or axiomatically by postulating the appropriate algebraic laws. The LOTOS specification language [vEVD89] uses CCS as a semantic basis, and several examples of protocol verifications within this or closely related theories exist [BK84, Koo85, SFD85, LM87, Par88, BSV88, BA90, Bae90, EFJ91], some of which used automated tools.

One limitation of traditional process algebras is that the linkage between processes cannot change as the processes execute. The linkage is determined by the choice of port names (an output action can only combine with an input action on the same port), and such names cannot move between processes. This limitation is striking when modelling mobile networks. Consider a mobile node, perhaps in a car, approaching a stationary node and establishing communication with it. To do this the mobile node must be told *how* to contact the stationary node, perhaps by obtaining the correct radio frequency and low-level calling mechanisms. An abstract view of this phenomenon is that the mobile node acquires a new link, or port name, for communication.

This abstract view can be fully captured in the π-calculus [MPW89a] which extends CCS in one important way: communication actions can carry names of ports from one process to another. Thus, in an output action $\overline{a}v$ the value v may represent a port name. A simple example of an expression in the π-calculus is the following:

$$\overline{a}b \,.\, P \mid a(x) \,.\, \overline{x}v \,.\, Q$$

Here the left hand side transmits, along a, the port name b. The right hand side will receive this name and use it to transmit the value v.

As an example of mobility we will consider a part of a protocol intended to be used in the *Public Land Mobile Network* (PLMN) proposed by *European Telecommunication Standards Institute* (ETSI). In this network some nodes are mobile and may move out of (and in to) the communicating range of stationary nodes. The protocol should ensure uninterrupted communications between mobile nodes even as they move around. We focus on the *handover* procedure in the protocol. This procedure is activated when a mobile node moves out of the range of a stationary node with which it is communicating. The handover procedure relieves the stationary node from the responsibility to communicate with the mobile node, and assigns this responsibility to another stationary node. Clearly, the precise location of a mobile node is unimportant for the handover procedure; it is only necessary to know with which other nodes the mobile node can communicate and how these communication capabilities can change. These facts are easily represented in the π-calculus.

In Section 2 below we give an overview of the structure of the Public Land Mobile Network and explain informally the handover procedure. Section 3 is an informal introduction to the π-calculus. We concentrate on the relevant aspects and explain them mainly by examples; an interested reader is referred to the reports [MPW89a, MPW89b]. The formal definition of a simplified version of the handover procedure is contained in Section 4, where we also use the definition to explain how a handover is accomplished and prove that after a handover the system returns to its initial state. In Section 5 we motivate a more abstract view of the protocol and prove that this view is equivalent. The proof is conducted wholly within the π-calculus by using the appropriate algebraic laws. To our knowledge it is the first substantial proof within this formalism; we therefore include some details of it in an appendix. In Section 6 we indicate how the formal description can be generalized to a more realistic protocol. Finally in Section 7 some concluding remarks are drawn and we comment on alternative ways to formulate the handover procedure.

2. The Handover Procedure in GSM Protocols

In 1982, CEPT (*Conférence Européenne des Administrations des Postes et des Télécommunications*) formed the GSM (*Group Spécial Mobile*) with the task to make a recommendation for a Public Land Mobile Network. The major characteristic of such a network is that some nodes are *mobile* and may be connected to different nodes at different points in time. The requirements on the PLMN were that it should be usable in all European countries, the system should provide telephone services as well as supplementary services according to ISDN, the system should be able to coexist — in the same radio band — with other mobile systems, and it should not require any major modifications to existing public networks. It was soon recognized that the PLMN would need a new set of communication protocols to manage routing to mobile nodes. The development of these protocols is now being continued by ETSI. The following account of the PLMN and its protocols is based on personal experience (one of us, Fredrik Orava, collaborated with Swedish Telecom in structuring the protocols) and on the recommendation [ETS89].

2.1. PLMN System Architecture

The PLMN consists of Mobile Stations, Base Stations, Mobile Switching Centers, and Location Registers (Fig. 1 left). A *Mobile Station* (MS) consists of a *Mobile Station Control Unit* (MSCU) and a terminal equipment (e.g. a telephone unit) providing service to an end user. A *Base Station* (BS) manages the interface between the Mobile Stations and a stationary network. Each Base Station controls radio communication within a designated geographical area, called a *cell*. All communication with Mobile Stations in a cell is routed through the Base Station responsible for the cell.

A *Mobile Switching Center* (MSC) manages a set of Base Stations. Communication between two Mobile Switching Centers, and between an MSC and an external network, uses an existing stationary public network. Communication

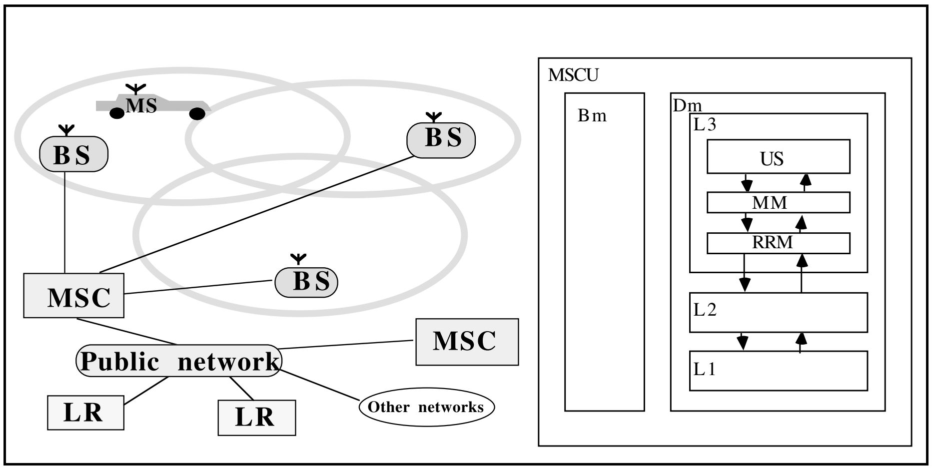


Fig. 1. Left: PLMN System Overview. Right: The architecture of a Mobile Station.

between an MSC and a BS may also employ this public network, or any other fixed network. The *Location Registers* (LR) record the status of the PLMN. Each Mobile Station is associated with a *Home Location Register* to which all incoming calls to the station are routed. This register contains information about where to find the associated Mobile Station.

Let us zoom in on the architecture of a Mobile Station (Fig. 1 right). The Mobile Station Control Unit consists of two modules: the *Bm-module* which handles the circuit switched data channel, and the *Dm-module* which handles the control channels used for call establishment and release. The Dm-module contains the protocols recommended by GSM. The sole purpose of these protocols is to connect, maintain, and disconnect data channels between the MS and a BS (not necessarily the same during a call), and to do end-to-end call setup and release. The protocol for the data channel (if any) resides in the Bm module.

The Dm-module is subdivided into three layers. The Physical Layer (L1) handles the physical communication and most of the details concerning the radio channels. The Logical Link Layer (L2) consists of a modified version of a link access protocol [CCI85] called LAPDm. The Network Layer (L3) consists of all other functions needed to manage the radio channels as well as protocols for allocation and deallocation of data channels. The Network Layer is further subdivided into three sublayers: the Radio Resource Management (RRM) sublayer, the Mobility Management (MM) sublayer, and the User Services (US) sublayer. The RRM sublayer handles the radio management and provides MM with an uninterrupted link service to a Base Station, even if the Mobile Station moves to another cell. The MM sublayer performs bookkeeping such as updating the Home Location Register associated with the station. The US sublayer provides the terminal equipment in the Mobile Station with services such as call establishment and release.

2.2. The Handover Procedure

The handover procedure resides in the RRM sublayer of the Mobile Station. The purpose of this procedure is to move data and control channels from the Base Station currently communicating with the Mobile Station (in the following called the *old* Base Station) to another Base Station (called the *new* Base Station) in another cell. This procedure is needed to ensure the continuous operation of a data channel allocated to a Mobile Station moving across a cell boundary; without such a handover the communication on the data channels could be disrupted because of the restricted range of the radio equipment.

The network (MSC) initiates the handover by transmitting a *handover command* message to the Mobile Station via the old Base Station. The handover command message contains parameters enabling the Mobile Station to locate the radio channels of the new Base Station. When transmitting this message the network (MSC) suspends transmission of all messages except for RRM sublayer messages. Upon receipt of a handover command message the Mobile Station disconnects the old radio channels and initiates the establishment of lower layer connections on the new radio channels. In order to establish these connections the Mobile Station sends *handover access* messages to the new Base Station. The purpose of the access messages is to synchronize the Mobile Station with the new Base Station. When the lower layer connections are successfully established, the Mobile Station sends a *handover complete* message to the network via the new Base Station. When the handover complete message has been received the network resumes normal operations and releases the old radio channels; these are now free and can be allocated to another Mobile Station.

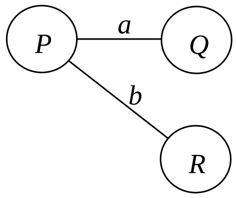
There are a number of ways in which the handover procedure may fail, we mention here only a few. If the Mobile Station is unable to establish the lower layer connections on the new radio channels it will try to reestablish the connections on the old channels. If this succeeds the Mobile Station sends a *handover failure* message to the network via the old Base Station, and resumes operation on the old channels as if no handover attempt had been made. If the Mobile Station does not succeed in reestablishing the connections on the old channels the Mobile Station is isolated and recovery procedures are invoked. The network may also time out before it receives a response (handover complete or handover failure) to a handover message. The old channels are then released and all connections with the Mobile Station are cleared.

3. The π-calculus

3.1. Mobile Processes

In this section we will briefly present the π-calculus, a calculus for descriptions of communicating processes with a changing interconnection structure. This calculus is based on ideas of Engberg and Nielsen [EN86] and is in many respects reminiscent of CCS [Mil89]. We will only mention aspects of the π-calculus which are relevant for the understanding of this paper, and we will not treat any aspect in depth — an interested reader is referred to the reports [MPW89a, MPW89b]. In the following we assume the reader to be familiar with the basic concepts of CCS.

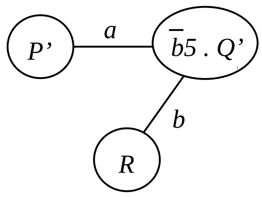
As in CCS, processes in the π-calculus are called *agents* and are built from a small set of operators including a binary parallel composition written $|$. The important way in which the π-calculus extends CCS is that agents may communicate port names (and not only data values) to each other. As an example consider three agents P, Q, and R. Assume that P has ports $\overline{a}$ and $\overline{b}$, Q has port a and R has port b. The bar over a port name indicates that it is used as an output port while non-barred ports are used as input ports. In the parallel composition $P|Q|R$ communications are possible between P and Q (through the a ports) and between P and R (through the b ports), but there is no communication possible between Q and R.



In CCS this linking structure between agents would be static in the sense that no agent can acquire a new port as a result of a communication. But in the π-calculus the agent P can send (along a) its b port to Q. For example, P could be $\overline{a}b.P'$ and Q could be $a(x).\overline{x}5.Q'$. There is then a communication

$$P|Q|R \xrightarrow{\tau} P'|\overline{b}5.Q'|R$$

where Q has evolved into $\overline{b}5.Q'$; this agent has a b port which will be used to send the value 5 to R. Assuming that P' does not further use the b port the resulting system can be displayed as



Such a reconfiguration of the system could perhaps be triggered by the fact that R moves from P to the vicinity of Q. Of course, geographic locations do not enter the calculus at all, but their consequences for the interconnection structure do. These are often the *only* interesting consequences, therefore we may say that the π-calculus is appropriate for descriptions of mobile agents.

It may appear that the π-calculus would have to be very complicated since it has to talk about ports and port variables (x in the example above) as well as data values, functions on data, and data variables. Fortunately this is not the case: the calculus treats ports, values, and variables uniformly, so there is no need to distinguish between them formally. There is just one syntactic class for these, the class of *names* written $\mathcal{N}$. In the example above a, b, x, and 5 are all just names. Furthermore, there is no need to introduce functions on data values since these can be encoded as agents within the calculus.

3.2. Syntax and Informal Semantics

In the following we let $x, y, z, u, v, \ldots$ range over $\mathcal{N}$. We assume a set of *agent identifiers* ranged over by $A, B, \ldots$; each identifier has a nonnegative *arity*. We let $P, Q, \ldots$ range over the *agents*, which are of the following kinds:

$\mathbf{0}$ is as in CCS an agent which can do nothing.

$\overline{y}x.P$ is an agent whose first action will be to output the name x on port $\overline{y}$; thereafter it behaves as P. In this action y is called the *subject* and x the *object*.

$y(x).P$ is an agent whose first action is to receive an input on port y; thereafter it behaves as P but with the newly received name in place of x. We say that y is the *subject* of the action and that x is *bound* in $y(x).P$; this x is just a placeholder for a new name to be received.

$\tau.P$ is an agent whose first action will be the *silent* action τ; thereafter it behaves as P.

$P + Q$ is as in CCS an agent which behaves like either P or Q.

$P|Q$ is as in CCS an agent representing the parallel composition of P and Q. This agent can do anything that P or Q can do, and moreover communications between P and Q can occur if one agent outputs a name and the other inputs a name on the same port; as in CCS these communications result in τ actions.

$(x)P$ is an agent which acts like P but the name x is *restricted*, i.e. it cannot be used as a port in communications with the environment of the agent. In CCS this would be written $P \backslash x$; in the π-calculus the notation $(x)P$ is used to emphasize the fact that x is *bound* in this agent: the name x is a local name whose scope is P.

$[x = y]P$ is an agent which behaves like P if x and y are the same name; otherwise it does nothing.

$A(y_1, \ldots, y_n)$ is an agent if A is an identifier of arity n; for any such identifier there is a *defining equation* written $A(x_1, \ldots, x_n) \stackrel{\text{def}}{=} P$, where the names $x_1, \ldots, x_n$ are distinct and are the only names which may occur free in P. The agent $A(y_1, \ldots, y_n)$ behaves like P where y_i is substituted for x_i for all $i = 1, \ldots, n$. The x_i:s may be considered formal parameters of A, while the y_i:s are actual parameters in $A(y_1, \ldots, y_n)$. As in CCS, agent identifiers provide recursion since the defining equation for A may contain A itself.

By convention, the unary operators bind tighter than parallel composition which binds tighter than summation. We will write $(x_1 \ldots x_n)P$ for $(x_1) \ldots (x_n)P$. The formal operational semantics of agents is defined and explained in the report [MPW89b]; hopefully this paper can be understood without it.

3.3. Free and Bound Names

The *bound names* in an agent P are the names which occur bound by an input or restriction operator in P. We write $P \equiv Q$ to mean that P and Q are alpha-equivalent, i.e. only differ in the choice of bound names. In the following we will not distinguish between alpha-equivalent agents. The *free names* in P are the names which occur not bound in P. We write $P\{y/x\}$ to mean the result of substituting y for all free occurrences of x in P, with a change of bound names if necessary to prevent y from becoming bound in P. For example,

$$((y)\overline{x}y.\mathbf{0} + y(x).\overline{x}y.\mathbf{0})\{y/x\} \equiv (y')\overline{y}y'.\mathbf{0} + y(x).\overline{x}y.\mathbf{0}$$

where y' is a name distinct from y. Note that the bar over a name in the output prefix is *not* part of the name — it just signifies that the name will be used as an output port.

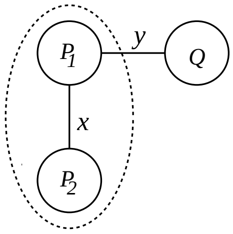
The free names of an agent represent its knowledge of, or linkage to, other agents. Dually, they represent what the environment knows about the agent. The bound names of an agent, in contrast, are unimportant for its environment. In the case of an input binding $y(x).P$ the bound x in P just acts as a placeholder; this binding is reminiscent of λ-abstraction. The binding in a restriction $(x)P$ is of a different kind: here x can be thought of as a local name or a name which is guaranteed to be different from any name the environment may know. This x can be used in "private" links between components of P, but cannot be used as a vehicle for communication between P and its environment. Most programming languages provide such local declarations and many operating systems provide generators of "magic numbers" which are guaranteed to be unique in this sense.

Since x is local in $(x)P$, any name (even x itself) which occurs outside this agent is unrelated to the local x. Consider the agent $\overline{x}y\,.\,P|(x)x(z)\,.\,Q$. The two x:es on the different sides of the parallel composition do not represent a link between $\overline{x}y.P$ and $(x)x(z).Q$. In particular these agents cannot communicate along x. Even if x is sent as an object to an agent with a local x (a phenomenon called *scope intrusion*) the two x:es remain distinct, as evidenced by the transition

$$\overline{y}x.P|y(z).(x)Q \;\xrightarrow{\tau}\; P|((x)Q)\{x/z\} \equiv P|(x')Q\{x'/x\}\{x/z\}$$

The renaming of the local x to x' is a consequence of the definition of substitution: bound names are changed as necessary to avoid captures.

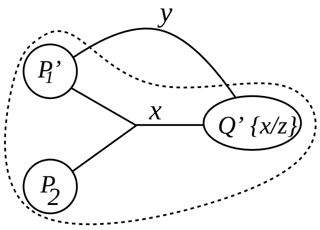
The only situation when the environment of $(x)P$ gets to know about the local x is when P sends it this x as an *object* in a communication. The scope of x then grows to include the recipient (but no other agent) in the communication. This phenomenon is called *scope extrusion* and may be exemplified as follows. Assume that P_1 and P_2 both use the name x (i.e. have x as a free name). Then the name x in $P \equiv (x)(P_1|P_2)$ represents a private link; P has no free x. Now let Q be another agent and assume that P_1 and Q communicate over y. The situation is illustrated graphically below, where a dotted curve is used to indicate the scope of the local x.



For example P_1 may be $\overline{y}x.P_1'$ and Q may be $y(z).Q'$. This means that x will be sent along y to Q, and since x is local its scope will extend — think of x as being trapped in a balloon (its scope); whenever x is sent to a place outside the balloon it will extend the balloon to this place. This communication can be written

$$(x)(P_1|P_2)\,|\,Q \;\xrightarrow{\tau}\; (x)(P_1'|P_2|Q'\{x/z\})$$

The result of the transition can be illustrated:



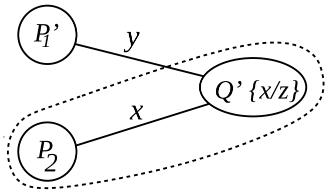
If additionally P_1' does not further use x, i.e. x is not in the free names of P_1', then we may use the *scope extension* law which says that scope boundaries may move freely across agents which do not use the scoped name:

If x is not free in P then $(x)(P|Q) \sim P|(x)Q$.

Here $\sim$ is the *strong equivalence* which will be introduced in Section 5.2. Through an application of this law the result of the transition may be rewritten:

$$(x)(P_1'|P_2|Q'\{x/z\}) \sim \; P_1' \,|\, (x)(P_2|Q'\{x/z\})$$

Pictorially this means:



We can here say that the scope of x has "migrated" from P_1 to Q. In our communication protocol such migrations will be triggered by physical movement (P_2 moving from P_1 towards Q in this example).

3.4. Encoding Data Values

If we are only interested in a finite set of data values then we can simply designate a finite set of names to represent them. We will refer to such names as *constants*. Formally a constant is just a name which never occurs bound by an input or restriction operator. As an example let the constants `t` and `f` represent the boolean values. An agent $Inv(u, v)$ which repeatedly receives a value on u and outputs its inverse on v can then be defined as

$$Inv(u, v, \mathtt{t}, \mathtt{f}) \stackrel{\mathrm{def}}{=} u(x).\Big([x = \mathtt{t}]\overline{v}\mathtt{f}.Inv(u, v, \mathtt{t}, \mathtt{f}) \;+\; [x = \mathtt{f}]\overline{v}\mathtt{t}.Inv(u, v, \mathtt{t}, \mathtt{f})\Big)$$

Obviously any function on finitely many values can be represented as an agent in this way. Notice that matching and summation are combined into a "case" construct which activates different branches depending on the value received. For convenience we introduce an abbreviation for this:

$$u : [y_1 \Rightarrow P_1, \ldots, y_n \Rightarrow P_n]$$

abbreviates

$$u(x).\Big([x = y_1]P_1 + \cdots + [x = y_n]P_n\Big)$$

if x does not occur free in any P_i. For the special case where $n = 1$ we write $u[y].P$ to mean $u : [y \Rightarrow P]$; this agent inputs a name on u and if that name is y the agent proceeds as P, otherwise it does nothing. As another abbreviation, whenever a particular set of constants is implied by context, we will omit the constants in the formal and actual parameters to identifiers. With these conventions the agent computing the inverse can be more compactly rendered as

$$Inv(u, v) \stackrel{\mathrm{def}}{=} u : [\mathtt{t} \Rightarrow \overline{v}\mathtt{f}.Inv(u, v),\ \mathtt{f} \Rightarrow \overline{v}\mathtt{t}.Inv(u, v)]$$

The encoding of arbitrary data types and functions over data types is an interesting topic; the report [MPW89a] contains further examples. In the present paper we will be satisfied with the simplistic strategy outlined above.

4. A Formal Description

4.1. Definition of Handover

In this section we formally specify the handover procedure described in Section 2 using the π-calculus presented in Section 3. We will concentrate on the mobile aspect of the system and therefore we make a number of simplifications. We consider a system consisting of only one Mobile Station and two Base Stations controlled by the same Mobile Switching Center. Each Base Station possesses one radio channel for communication with the Mobile Station. We only consider one way in which the procedure can fail, and assume that the Mobile Station never becomes isolated. Finally, we only consider transfer of user data in one direction, from the network to the Mobile Station. These simplifications are all made in order to obtain a formal specification which can easily be comprehended and proved; in Section 6 we will discuss how more realistic models can be made. For a more complete specification of the handover procedure see [Naz89].

The formal specifications of the stations are given in Fig. 2. The intended behaviour of $System$ is to act as a buffer: whatever is received on port `in` will be transmitted on port `out`. We regard these ports as constants; they will never occur bound and will remain fixed throughout the execution of $System$. The `in` port is presumably used by the stationary part of the network, while the `out` port interfaces to the terminal equipment in the Mobile Station. The initial configuration of the system is depicted in Fig. 3. The dotted curves represent scope restrictions: a link which lies completely within an area enclosed by a dotted curve is local to the agents in that area.

The system consists of four stations: one Mobile Switching Center MSC, one *active* Base Station BS_a, one *passive* Base Station BS_p, and one Mobile Station MS. The MSC and the BS_p forms the P *part* of the system, and the BS_a and the MS forms the Q *part*. The link f_a connects MSC with BS_a, while the link f_p connects MSC with BS_p. These links are fixed in the sense that the names will not be transmitted as objects in communications; intuitively they are part of a stationary network. They are not accessible by anyone outside $System$, so we consider them local for (i.e. restricted in) $System$.

In addition, each BS has one mobile link m representing a radio channel. The active BS_a shares this m with the MS, while BS_p shares its m with the MSC. This represents the fact that BS_a uses its m to communicate with the MS, while the m in BS_p is unused and held in reserve by the MSC. Note that

$$
\begin{array}{lcl}
CC(f_a, f_p, l) & \stackrel{\text{def}}{=} & \quad \mathtt{in}(v) \,.\, \overline{f_a}\, \mathtt{data} \,.\, \overline{f_a}\, v \,.\, CC(f_a, f_p, l) \\
 & & + \; l(m_{new}) \,.\, \overline{f_a}\, \mathtt{ho\_cmd} \,.\, \overline{f_a}\, m_{new} \,. \\
 & & \qquad \Big(\quad f_p[\mathtt{ho\_com}] \,.\, \overline{f_a}\, \mathtt{ch\_rel} \,.\, f_a(m_{old}) \,. \\
 & & \qquad\qquad \overline{l}\, m_{old} \,.\, CC(f_p, f_a, l) \\
 & & \qquad + \; f_a[\mathtt{ho\_fail}] \,.\, \overline{l}\, m_{new} \,.\, CC(f_a, f_p, l) \Big) \\
HC(l, m) & \stackrel{\text{def}}{=} & \overline{l}\, m \,.\, l(m) \,.\, HC(l, m) \\
MSC(f_a, f_p, m) & \equiv & (l)\Big(HC(l, m) \mid CC(f_a, f_p, l) \Big) \\
BS_a(f, m) & \stackrel{\text{def}}{=} & f : [\; \mathtt{data} \Rightarrow f(v) \,.\, \overline{m}\, \mathtt{data} \,.\, \overline{m}\, v \,.\, BS_a(f, m), \\
 & & \quad \mathtt{ho\_cmd} \Rightarrow f(v) \,.\, \overline{m}\, \mathtt{ho\_cmd} \,.\, \overline{m}\, v \,. \\
 & & \qquad \Big(\quad f[\mathtt{ch\_rel}] \,.\, \overline{f}\, m \,.\, BS_p(f, m) \\
 & & \qquad + \; m[\mathtt{ho\_fail}] \,.\, \overline{f}\, \mathtt{ho\_fail} \,.\, BS_a(f, m) \Big)] \\
BS_p(f, m) & \stackrel{\text{def}}{=} & m[\mathtt{ho\_acc}] \,.\, \overline{f}\, \mathtt{ho\_com} \,.\, BS_a(f, m) \\
MS(m) & \stackrel{\text{def}}{=} & m : [\mathtt{data} \Rightarrow m(v) \,.\, \overline{\mathtt{out}}\, v \,.\, MS(m), \\
 & & \quad \mathtt{ho\_cmd} \Rightarrow m(m_{new}) \,. \\
 & & \qquad \Big(\quad \overline{m_{new}}\, \mathtt{ho\_acc} \,.\, MS(m_{new}) \\
 & & \qquad + \; \overline{m}\, \mathtt{ho\_fail} \,.\, MS(m) \Big)] \\
P(f_a, f_p) & \equiv & (m)\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m) \Big) \\
Q(f_a) & \equiv & (m)\Big(BS_a(f_a, m) \mid MS(m) \Big) \\
System & \equiv & (f_a)(f_p)\Big(P(f_a, f_p) \mid Q(f_a) \Big)
\end{array}
$$

Fig. 2. A formal specification of the handover procedure.

"m" is used to represent both mobile links. This is possible since these m:s are local names with disjoint scopes (one m is local in P and the other is local in Q).

Finally, the MSC consists of two parts. One is the *Communication Controller* CC. This part performs all communications over the fixed network (links `in`, f_a, and f_p). The other part is the *Handover Controller* HC, which stores all unused radio channels (in our simple example there will be just one such, namely m in BS_p) and which may at any time tell the CC to initiate a handover; this it does by transmitting on a private link l the channel into which the handover should be made. In reality HC would be quite complex, containing many channels for many Base Stations, and it would be equipped to determine when and to which Base Station a handover should be made (perhaps by comparing the quality of reception at different bases).

The operation of each agent is as follows. The CC accepts messages from the network on port `in`. An object received in this way is interpreted as data

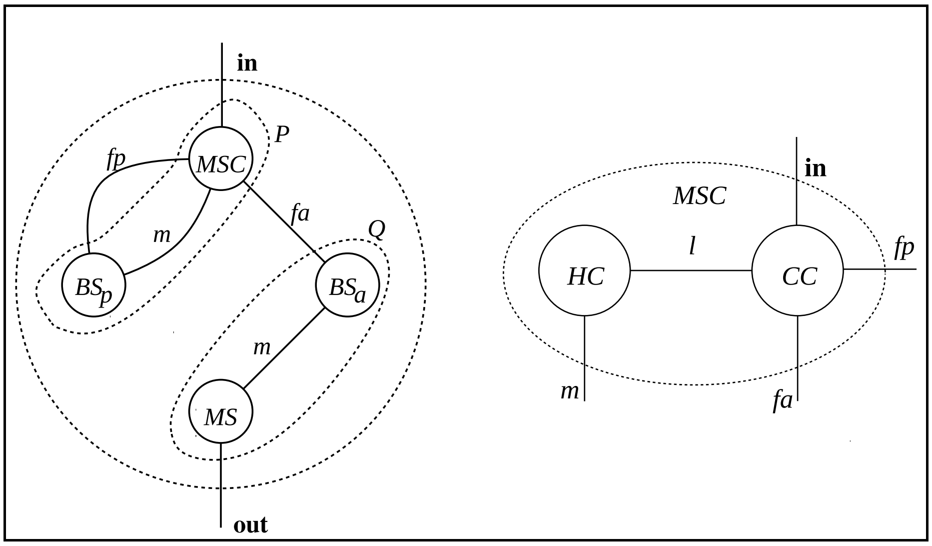


Fig. 3. *System* in its initial state.

to be transmitted to MS. The CC thus transmits a `data` message, followed by the data, to BS_a using f_a. In addition CC can engage in a handover. The handover begins when CC receives an object m_{new} on l; this object is the new radio channel which should be used by MS. The CC then transmits a `ho_cmd` message (*handover command*) followed by m_{new} on f_a.

Thereafter one of two things can happen. Either CC receives a `ho_com` message (*handover complete*) from the (formerly) passive station indicating a successful handover. Then CC requests the now unused radio channel from the (formerly) active Base Station with a `ch_rel` message (*channel release*), accepts this channel on f_a, forwards it on l for storage, and resumes normal operations. Notice that f_a now connects to the passive Base Station and f_p to the active Base Station since the stations have exchanged rôles. Hence the corresponding recursive call in CC exchanges the order of the fixed links in the parameter list.

Alternatively CC receives a `ho_fail` message (*handover failure*) from the active Base Station. Then CC returns the link m_{new} on l and resumes normal operations — in this case the base stations have not exchanged rôles.

The Handover Control HC initially stores one free radio channel represented by m, and a handover is initiated by sending this m to CC. The HC then contains no more channels and waits for a channel to be delivered on l; it then reverts to its original state storing the newly received channel.

A Base Station has a fixed link f to MSC, and a mobile link m representing a radio channel. An active Base Station BS_a repeatedly receives messages from MSC on f and forwards them to MS on m. If the message was a `ho_cmd` message then the Base Station, after transmitting the messages to MS, waits for one of two things to happen. One possibility is that the MSC requests the now unused channel, then this channel (m) is delivered to MSC. In this case the handover has succeeded and the Base Station is no longer in contact with the Mobile Station, hence the Base Station continues as a passive station BS_p. The other possibility is that the Base Station receives a `ho_fail` from MS; this message is then forwarded to MSC and the station continues as an active Base Station.

A passive Base Station BS_p awaits a `ho_acc` message (*handover access*) from MS on its radio channel m. Upon reception of this message a `ho_com` message to MSC signals that a handover has been completed, and the station continues as an active station BS_a.

The Mobile Station repeatedly accepts data messages from a Base Station on a radio channel m; these messages are forwarded to the terminal equipment on `out`. The MS also accepts `ho_cmd` messages followed by a new radio channel m_{new}. The MS then activates the new Base Station by sending a `ho_acc` message via m_{new}, and continues to use m_{new}. If this is not possible an alternative is to send a `ho_fail` on the old channel m, and continue to use m.

Note that the factors which determine whether a handover will be successful or not are not represented in our model, therefore the choice between failure and success is nondeterministic: both alternatives are possible. A more detailed treatment of this choice is beyond the scope of this paper.

4.2. Executing a Handover

We will now examine in detail how a handover is accomplished. Our point of departure will be a system in its initial state as in Fig. 3. We will call the leftmost Base Station BS_2 and the rightmost Base Station BS_1; initially BS_1 is an instance of BS_a and BS_2 is an instance of BS_p but this will change as the handover executes. The first action of the handover is an internal action within MSC, namely a communication along l where HC sends to CC the name m. This implies no changes in the figure other than that the local m connecting BS_2 and MSC now relates to the CC part of the MSC. The HC part becomes $l(m).HC(l,m)$ and this agent has no free m.

The next two actions are communications between P and Q along f_a, where MSC (in P) sends the local m in P to BS_1 (in Q). This results in an extrusion of the scope of m. Furthermore, since Q already has a local name m (representing the link between BS_1 and MS) this local m will automatically be renamed to m'. This is an instance of scope intrusion as explained in Section 3. The situation now looks as in Fig. 4, and the corresponding agent is

$$
\begin{array}{ll}
(f_a)(f_p)(m)\Big(& MSC'(f_a,f_p,m)|BS_p(f_p,m) \\
& | \\
& (m')\Big(BS'(f_a,m',m)|MS(m')\Big)\Big)
\end{array}
$$

where

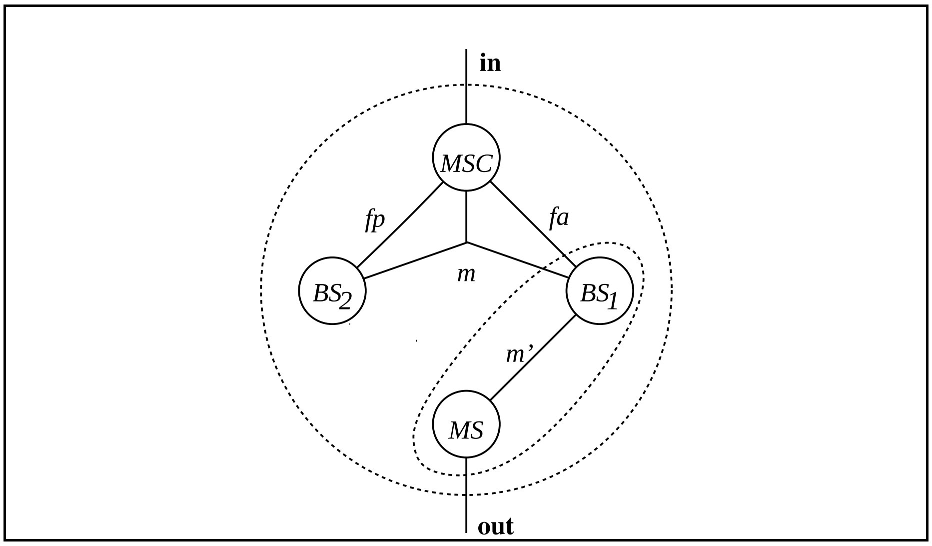


Fig. 4. BS_1 has just received a new channel.

$$\begin{array}{rcl} MSC'(f_a, f_p, m) & \equiv & (l)\Big(\; l(m) \,.\, HC(l, m) \\ & & \quad \mid \\ & & \quad \Big(\quad f_p[\mathtt{ho\_com}] \,.\, \overline{f_a}\,\mathtt{ch\_rel} \,. \\ & & \qquad\quad f_a(m_{old}) \,.\, \overline{l}\, m_{old} \,.\, CC(f_p, f_a, l) \\ & & \qquad + \; f_a[\mathtt{ho\_fail}] \,.\, \overline{l}\, m \,.\, CC(f_a, f_p, l)\Big)\Big) \end{array}$$

$$BS'(f, m', m) \quad \equiv \quad \overline{m'}\,\mathtt{ho\_cmd} \,.\, \overline{m'}\, m \,.\, BS''(f, m')$$

$$\begin{array}{rcl} BS''(f, m') & \equiv & f[\mathtt{ch\_rel}] \,.\, \overline{f}\, m' \,.\, BS_p(f, m') \\ & & + \; m'[\mathtt{ho\_fail}] \,.\, \overline{f}\,\mathtt{ho\_fail} \,.\, BS_a(f, m') \end{array}$$

BS_1 may now, after sending `ho_cmd` to MS, transmit the name m on m'. The situation now looks as in Fig. 5 (left), and the corresponding agent is

$$\begin{array}{l} (f_a)(f_p)(m)\Big(\quad MSC'(f_a, f_p, m) | BS_p(f_p, m) \\ \qquad\qquad\quad \mid \\ \qquad\qquad\quad (m')\Big(BS''(f_a, m') | MS'(m, m')\Big)\Big) \end{array}$$

where

$$MS'(m, m') \quad \equiv \quad \overline{m}\,\mathtt{ho\_acc} \,.\, MS(m) \quad + \quad \overline{m'}\,\mathtt{ho\_fail} \,.\, MS(m')$$

Now one of two things can happen: either MS fails to establish contact with BS_2 or it succeeds. If MS fails then it communicates this fact through a `ho_fail` message to BS_1; this message is forwarded to CC which returns the apparently unsatisfactory radio channel m to HC. The whole system is then back in its initial state.

If MS succeeds it sends a `ho_acc` message to BS_2 along the newly received

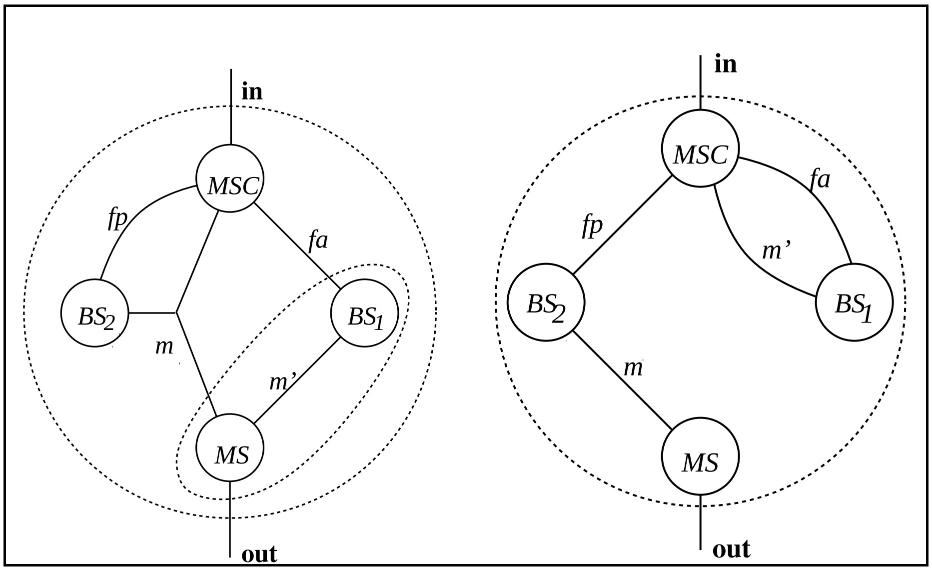


Fig. 5. Left: MS has just received a new channel. Right: the configuration after a successful handover.

link m. This message is forwarded as a `ho_com` message to CC. A reception of this means that CC proceeds to collect the now unused radio channel m' from BS_1; this channel is forwarded to HC. As a result the scope of this local m' extrudes to the former P part, i.e. MSC and BS_2. The system has now reached a configuration where transmission of user data is again possible. The resulting agent is depicted in Fig. 5 (right) and can be written

$$(f_a)(f_p)(m)(m')\Big(\begin{array}{l} (l)(HC(l,m')|CC(f_p,f_a,l)) \\ | \\ BS_a(f_p,m)|BS_p(f_a,m')|MS(m)\Big) \end{array}$$

Call this agent $System'$. It is of interest to compare this agent, which represents the result of a successful handover, with the original agent $System$. As is evident from the definitions of these agents (and from the figures illustrating the interconnection structures) the agents are not identical. But they can be proven strongly equivalent:

$$
\begin{array}{lll}
\multicolumn{2}{l}{System'} & \\
\sim & (f_p)(f_a)(m)(m')\Big(\; MSC(f_a, f_p, m') \mid BS_a(f_a, m) & (1)\\
& \qquad \mid & \\
& \qquad BS_p(f_p, m') \mid MS(m)\Big) & \\
\sim & (f_p)(f_a)\Big(\; (m')\Big(MSC(f_a, f_p, m') \mid BS_p(f_p, m')\Big) & (2)\\
& \qquad \mid & \\
& \qquad (m)\Big(BS_a(f_a, m) \mid MS(m)\Big)\Big) & \\
\sim & (f_p)(f_a)\Big(P(f_a, f_p) \mid Q(f_a)\Big) & (3)\\
\sim & System & (4)
\end{array}
$$

The equivalences are justified by laws in the π-calculus (see Section 5.2) as follows: (1) α-converting f_a to f_p and vice versa; (2) $\mid$ is associative and commutative, scope extension (laws **P1**–**P3**); (3) α-converting m' to m; (4) restriction commutes (law **R1**). Thus, we formally proved that a successful handover in the initial configuration has no effect on the externally observable behaviour.

5. Verification

5.1. An External View of *System*

In this section we will analyze the observable behaviour of our protocol *System* as perceived on the external ports `in` and `out`. Intuitively, *System* should function as a buffer, receiving items (i.e. names) on `in` and transmitting them on `out`. Each of *MSC*, *BS* and *MS* can hold at most one item in transit, so *System* should correspond to a buffer which can hold at most three items. Furthermore, *System* can spontaneously decide to engage in a handover (as described in the previous section).

A handover does not in itself result in an observable action. However, once a handover has begun, all messages in transit within *System* (if any) must be delivered on `out` before a new message is accepted on `in` (this is because the handover requires participation of all parts of *System*, and these parts cannot hold a message while engaging in a handover). Hence the handover has an indirect consequence for the externally observable behaviour: it preempts further actions on `in` until all messages in transit have been delivered. Also, note that a handover can begin only when *MSC* holds no item in transit, i.e. when *System* holds at most two items.

These considerations suggest that *System* in its initial state should behave as the agent S_0 where:

$$
\begin{array}{lcl}
S_0 & \stackrel{\mathrm{def}}{=} & \mathtt{in}(v) \,.\, S_1(v) + \tau \,.\, S_0 \\
S_1(v_1) & \stackrel{\mathrm{def}}{=} & \mathtt{in}(v) \,.\, S_2(v_1, v) + \overline{\mathtt{out}} v_1 \,.\, S_0 + \tau \,.\, \overline{\mathtt{out}} v_1 \,.\, S_0 \\
S_2(v_1, v_2) & \stackrel{\mathrm{def}}{=} & \mathtt{in}(v) \,.\, S_3(v_1, v_2, v) + \overline{\mathtt{out}} v_1 \,.\, S_1(v_2) \\
& & + \; \tau \,.\, \overline{\mathtt{out}} v_1 \,.\, \overline{\mathtt{out}} v_2 \,.\, S_0 \\
S_3(v_1, v_2, v_3) & \stackrel{\mathrm{def}}{=} & \overline{\mathtt{out}} v_1 \,.\, S_2(v_2, v_3)
\end{array}
$$

Here S_i for $i = 0, \ldots, 3$ corresponds to a buffer currently storing i items; these items are supplied as parameters (v_1, v_2, v_3). Each S_i for $i < 3$ has the possibility to become S_{i+1} by receiving an item on `in`. Each S_i for $i > 0$ also has the possibility to become S_{i-1} by transmitting an item on `out`. These possibilities correspond to the normal operation of the protocol, when no handover occurs. In addition each S_i for $i < 3$ may, through an unobservable action τ, reach a state where it continues by transmitting (onto `out`) all items currently stored and then becomes the empty buffer S_0. This unobservable action corresponds to the beginning of a handover. Note in particular that S_0 can do this action; the result will again be S_0, i.e. the buffer has a τ-loop (or *divergence*).

The goal in this section will be to formally prove that *System* and S_0 are behaviourally equivalent. The importance of such a result is that a designer using *System* as a building block when constructing a more complex system will not ever have to consider the detailed specification of how *System* is implemented through the different parts *MSC*, *BS* and *MS*. When dealing with the externally observable behaviour it will suffice to instead consider the simpler specification S_0.

Of course, the equivalence proof does not really mean that the protocol is "totally correct". It does, however, imply some desirable properties such as deadlock freedom and that data will never be lost. But since S_0 does not explicitly mention handovers it gives no guarantee that a handover will ever take place. One way to remedy this would be to make the τ representing a handover an observable "probe" action. Even so, our specification would only imply that handovers are possible and not that they must occur.

5.2. Algebraic Laws in the π-calculus

In CCS, the notion of behavioural equivalences is made mathematically precise by using *bisimulations*, and it has been proven that the equivalences satisfy many natural algebraic laws. Similar notions are currently being developed for the π-calculus, but we will not go into them in this paper. Instead, we will merely postulate the relevant algebraic laws and use them in our proof. In the course of the proof we will use two equivalences on agents:

$\dot{\sim}$ *strong ground equivalence*
$\dot{\simeq}$ *weak ground equivalence*

The strong ground equivalence corresponds to behaviour equivalence where the precise amount of internal action is significant — compare the strong equivalence of CCS. For example $\tau \,.\, \tau \,.\, \mathbf{0} \not\dot{\sim} \tau \,.\, \mathbf{0}$. In contrast, the weak ground equivalence identifies these two agents; the internal actions are significant only insofar as they preempt other actions — compare observation congruence in CCS.

The algebraic laws for strong ground equivalence (as stated in [MPW89b]) are summarized in Fig. 6. To state the laws in a compact way we define the *derived* prefix $\overline{x}(y) \,.\, P$ to mean $(y)\overline{x}y \,.\, P$ when $x \neq y$ and let α, β range over ordinary and derived prefixes. Note in particular that we do not assume that the equivalence is a congruence: in order to infer $x(y) \,.\, P = x(y) \,.\, Q$ we must prove (by **C1**) that P and Q are equivalent for *all* instantiations of y. Most laws are straightforward generalizations from the corresponding laws in CCS, only the expansion law **E** is complicated by the possibilities of scope migration.

A	$P \equiv Q$ implies $P = Q$
C0	All operators except input prefix preserve $=$
C1	If $P\{z/y\} = Q\{z/y\}$ for all z then $x(y) . P = x(y) . Q$
S0	$P + \mathbf{0} = P$
S1	$P + P = P$
S2	$P + Q = Q + P$
S3	$(P + Q) + R = P + (Q + R)$
I	$A(\widetilde{y}) = P\{\widetilde{y}/\widetilde{x}\}$ If $A(\widetilde{x}) \stackrel{\mathrm{def}}{=} P$
M0	$[x = y]P = \mathbf{0}$ If $x \neq y$
M1	$[x = x]P = P$
R0	$(x)P = P$ If $x \notin \mathrm{fn}(P)$
R1	$(x)(y)P = (y)(x)P$
R2	$(x)(P + Q) = (x)P + (x)Q$
R3	$(x)\alpha . P = \alpha . (x)P$ If x is not in α
R4	$(x)\alpha . P = \mathbf{0}$ If x is the subject of α
P0	$P \mid \mathbf{0} = P$
P1	$P \mid Q = Q \mid P$
P2	$(x)(P \mid Q) = P \mid (x)Q$ If $x \notin \mathrm{fn}(P)$
P3	$P \mid (Q \mid R) = (P \mid Q) \mid R$
E	Let $P \equiv \sum_i \alpha_i . P_i$ and $Q \equiv \sum_j \beta_j . Q_j$, where $\mathrm{bn}(\alpha_i) \cap \mathrm{fn}(Q) = \emptyset$ for all i, and $\mathrm{bn}(\beta_j) \cap \mathrm{fn}(P) = \emptyset$ for all j. Then $P \mid Q = \sum_i \alpha_i . (P_i \mid Q) + \sum_j \beta_j . (P \mid Q_j) + \sum_{\alpha_i \,\mathrm{comp}\, \beta_j} \tau . R_{ij}$ where the relation $\alpha_i \,\mathrm{comp}\, \beta_j$ (α_i *complements* β_j) holds in the following four cases, which also define R_{ij}: 1. α_i is $\overline{x}u$ and β_j is $x(v)$; then R_{ij} is $P_i \mid Q_j\{u/v\}$. 2. α_i is $\overline{x}(u)$ and β_j is $x(v)$; then R_{ij} is $(w)(P_i\{w/u\} \mid Q_j\{w/v\})$, where w is not free in $(u)P_i$ or in $(v)Q_j$. 3. α_i is $x(v)$ and β_j is $\overline{x}u$; then R_{ij} is $P_i\{u/v\} \mid Q_j$. 4. α_i is $x(v)$ and β_j is $\overline{x}(u)$; then R_{ij} is $(w)(P_i\{w/v\} \mid Q_j\{w/u\})$, where w is not free in $(v)P_i$ or in $(u)Q_j$.

Fig. 6. Algebraic laws (fn and bn stand for free and bound names respectively).

The weak ground equivalence is strictly weaker than strong ground equivalence and also satisfies the laws in Fig. 6. In addition it satisfies the well known τ-laws ([Mil89]), these are:

$$\begin{array}{ll} \mathbf{T0} & \alpha\,.\,\tau\,.\,P \,\dot{\simeq}\, \alpha\,.\,P \\ \mathbf{T1} & P+\tau\,.\,P \,\dot{\simeq}\, \tau\,.\,P \\ \mathbf{T2} & \alpha\,.\,(P+\tau\,.\,Q)+\alpha\,.\,Q \,\dot{\simeq}\, \alpha\,.\,(P+\tau\,.\,Q) \end{array}$$

In order to eliminate τ-loops from recursively defined agents we postulate the law

$$\mathbf{L}\quad \text{If } A \stackrel{\text{def}}{=} P+\tau\,.\,A \text{ and } B \stackrel{\text{def}}{=} \tau\,.\,P \text{ then } A \,\dot{\simeq}\, B$$

A similar law in CCS is called **R4** ([Mil89], page 166).

We call $\dot{\sim}$ and $\dot{\simeq}$ "ground" equivalences since they are not in general preserved by substitution of names. For example,

$$\overline{x}u\,.\,\mathbf{0} \,|\, y(w)\,.\,\mathbf{0} \;\dot{\sim}\; \overline{x}u\,.\,y(w)\,.\,\mathbf{0}+y(w)\,.\,\overline{x}u\,.\,\mathbf{0}$$

But if y is substituted for x this no longer holds, instead we have

$$\overline{y}u\,.\,\mathbf{0} \,|\, y(w)\,.\,\mathbf{0} \;\dot{\sim}\; \overline{y}u\,.\,y(w)\,.\,\mathbf{0}+y(w)\,.\,\overline{y}u\,.\,\mathbf{0}+\tau\,.\,\mathbf{0}$$

This explains the perhaps unexpectedly strong premise in the law for input (**C1**) in Fig. 6, since an input prefix means that *any* name may be received and replace the bound name. We define *strong (non-ground) equivalence* $\sim$ as strong ground equivalence under all substitutions of non-constant names, i.e., $P \sim Q$ iff $P\sigma \,\dot{\sim}\, Q\sigma$ for all substitutions σ from non-constant names to names, and similarly *weak (non-ground) equivalence* $\simeq$ as weak ground equivalence under all substitutions of non-constant names. For example, the two equations above are captured by the following non-ground equivalence

$$\overline{x}u\,.\,\mathbf{0} \,|\, y(w)\,.\,\mathbf{0} \;\sim\; \overline{x}u\,.\,y(w)\,.\,\mathbf{0}+y(w)\,.\,\overline{x}u\,.\,\mathbf{0}+[x=y]\tau\,.\,\mathbf{0}$$

The main use of the non-ground equivalences is in the laws for recursively defined agents which we adopt from [MPW89b]. These laws are the same as in ordinary CCS, and to formulate them we need some additional notation. Let $E, F, \ldots$ represent *agent expressions*; these are like agents with "holes" where agents or agent identifiers can be inserted, and write $E(P_1, \ldots, P_n)$ for the agent which is the result of inserting $P_1, \ldots, P_n$ into E. Two agent expressions E and F are *(strongly/weakly) equivalent* if $E(\widetilde{P})$ is (strongly/weakly) equivalent to $F(\widetilde{P})$ for all $\widetilde{P} = P_1, \ldots, P_n$.

Our first law for recursion means that if the right-hand sides of definitions are transformed, respecting equivalence, then the agent defined is the same up to equivalence:

U0 Suppose that $E_1, \ldots, E_n$ and $F_1, \ldots, F_n$ are expressions and $A_1, \ldots, A_n$ and $B_1, \ldots B_n$ identifiers such that for all i:

$$E_i = F_i$$
$$A_i(\widetilde{x}_i) \stackrel{\mathrm{def}}{=} E_i(A_1, \ldots, A_n)$$
$$B_i(\widetilde{x}_i) \stackrel{\mathrm{def}}{=} F_i(B_1, \ldots, B_n)$$

Then $A_i(\widetilde{x}_i) = B_i(\widetilde{x}_i)$ for all i.

The law **U0** holds for strong and weak non-ground equivalence (but fails for the ground equivalences!). The second law means that if two agents satisfy the same set of recursive equations, then the agents are equivalent:

U1 Suppose $E_1, \ldots, E_n$ are expressions and $P_1, \ldots, P_n$ and $Q_1, \ldots, Q_n$ are agents such that for all i:

$$P_i = E_i(P_1, \ldots, P_n)$$
$$Q_i = E_i(Q_i, \ldots, Q_n)$$

Then $P_i = Q_i$ for all i.

The law **U1** holds for strong non-ground equivalence provided $E_1, \ldots, E_n$ are *weakly guarded*, i.e., all occurences of P_j in $E_i(P_1, \ldots, P_n)$ are within a prefix operator. Furthermore, **U1** holds for weak non-ground equivalence provided $E_1, \ldots E_n$ are *guarded*, i.e., all occurences of P_j in $E_i(P_1, \ldots, P_n)$ are within an output or input prefix operator, and *sequential*, i.e., no E_i contains a parallel composition.

5.3. Outline of the Proof

The normal procedure to prove that a protocol is equivalent to a more abstract specification is as follows: repeatedly apply the expansion law to the protocol and prove that it satisfies the defining equation of the abstract specification, then apply **U1**. Our proof will follow this procedure but is complicated by a τ-loop in S_0 and in the protocol; thus **U1** cannot be applied since the relevant equations are not guarded. For S_0 we have

$$S_0 \stackrel{\mathrm{def}}{=} \cdots + \tau \,.\, S_0$$

This is comparatively unproblematic since **L** can be applied and will immediately remove the τ-loop and make the equations guarded. In the protocol the τ-loop is less explicit; it arises from a handover within *System*, and some preparatory work is needed to eliminate it. Note that the only law which can possibly remove such a τ-loop is **L**, which is only applicable to agent identifiers with particular defining equations. Our approach is therefore to work with strong equivalence until *System* has been rewritten in a form where the τ-loop can be removed.

The proof is mostly straightforward but quite long and requires some attention to detail. Since it is, to our knowledge, the first example of a nontrivial proof from the algebraic laws in the π-calculus we include extracts of the proof in an appendix; this will give the reader a taste of the techniques involved. In the rest of this section we will give an overview of the proof.

Our point of departure is the formal specification in Fig. 2, containing the defining equations for *System* and its components. We will repeatedly apply the expansion law **E** to parts of *System*, thereby generating new (recursive) equations for strong ground equivalence. These equations will in turn be applied to larger parts of *System*. This technique has the advantage of being modular: we never have to consider the whole of Fig. 2 at the same time. As a result we will get the following recursive equations for *System* (in the following "τ^n" abbreviates $\underbrace{\tau \,.\, \tau \,.\, \ldots \,.\, \tau}_{n \text{ times}}$):

$$
\begin{aligned}
&System\\
&\quad \dot\sim \quad \mathtt{in}(v) \,.\, A(v) + \tau^5 \,.\, \Big(\tau^5 \,.\, System + \tau^3 \,.\, System\Big)\\
&A(v_1)\\
&\quad \dot\sim \quad \tau^2 \,.\, A_{11}(v_1)\\
&A_{11}(v_1)\\
&\quad \dot\sim \quad \mathtt{in}(v) \,.\, \tau^2 \,.\, A_1(v_1, v) + \tau^3 \,.\, A_2(v_1)\\
&\qquad\quad + \tau \,.\, \Big(\mathtt{in}(v) \,.\, \tau \,.\, A_1(v_1, v) + \tau^2 \,.\, A_2(v_1)\\
&\qquad\qquad + \tau \,.\, \Big(\mathtt{in}(v) \,.\, A_1(v_1, v) + \tau \,.\, A_2(v_1)\\
&\qquad\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, System\Big)\Big)\\
&A_1(v_1, v_2)\\
&\quad \dot\sim \quad \overline{\mathtt{out}}\, v_1 \,.\, A(v_2) + \tau \,.\, A_1'(v_1, v_2)\\
&A_2(v_1)\\
&\quad \dot\sim \quad \overline{\mathtt{out}}\, v_1 \,.\, B + \tau \,.\, B_1'(v_1)\\
&A_1'(v_1, v_2)\\
&\quad \dot\sim \quad \overline{\mathtt{out}}\, v_1 \,.\, \tau \,.\, A_{11}(v_2)\\
&\qquad\quad + \tau \,.\, \Big(\mathtt{in}(v) \,.\, A_{12}(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \Big(\tau \,.\, B_1'(v_2) + \overline{\mathtt{out}}\, v_2 \,.\, B\Big)\\
&\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, A_{11}(v_2)\Big)\\
&A_{12}(v_1, v_2, v_3)\\
&\quad \dot\sim \quad \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \Big(\overline{\mathtt{out}}\, v_2 \,.\, A(v_3) + \tau \,.\, A_1'(v_2, v_3)\Big)\\
&B\\
&\quad \dot\sim \quad \tau^4 \,.\, \Big(\tau^5 \,.\, System + \tau^3 \,.\, System\Big)\\
&B_1'(v_1)\\
&\quad \dot\sim \quad \overline{\mathtt{out}}\, v_1 \,.\, \tau^3 \,.\, \Big(\tau^5 \,.\, System + \tau^3 \,.\, System\Big)\\
&\qquad\quad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \Big(\tau^5 \,.\, System + \tau^3 \,.\, System\Big)
\end{aligned}
$$

We write these equations as

$$\widetilde{System} \mathrel{\dot\sim} \widetilde{E}(\widetilde{System})$$

Note that v_1, v_2, v_3 are the only non-constant names occurring free in the equations. In fact we will prove the equations for all substitutions of v_1, v_2, v_3; thus we can infer the non-ground equivalence

$$\widetilde{System} \sim \widetilde{E}(\widetilde{System})$$

Now introduce new agent identifiers $\widetilde{D} = D_0, \ldots, D_8$ with the defining equations determined by $\widetilde{E}$:

$$\widetilde{D} \stackrel{\text{def}}{=} \widetilde{E}(\widetilde{D})$$

By **I** we have that $\widetilde{D}$ satisfy the same equations as $\widetilde{System}$. Moreover the equations are weakly guarded, so by **U1** for strong equivalence we conclude

$$\widetilde{System} \sim \widetilde{D}$$

Thus in particular

$$System \sim D_0$$

We continue by simplifying $\widetilde{D}$. The equation for D_0 is

$$D_0 \stackrel{\text{def}}{=} \mathtt{in}(v) \,.\, D_1(v) + \tau^5 \,.\, \left(\tau^5 \,.\, D_0 + \tau^3 \,.\, D_0\right)$$

First, by use of **U0** for weak equivalence it is safe to transform the right-hand side by using the τ-laws. This will result in a new definition of D_0 where the defining equation is

$$D_0 \stackrel{\text{def}}{=} \mathtt{in}(v) \,.\, D_1(v) + \tau \,.\, D_0$$

By applying **L** to this definition we obtain the (weakly) equivalent definition

$$D_0 \stackrel{\text{def}}{=} \tau \,.\, \mathtt{in}(v) \,.\, D_1(v)$$

Similarly the definition of S_0 can be transformed by **L** to yield a weakly equivalent definition beginning

$$S_0 \stackrel{\text{def}}{=} \tau \,.\, \mathtt{in}(v) \,.\, S_1(v)$$

The defining equations for S_i are now guarded and sequential, and we finally show that D_0 satisfies the same equation as S_0. By **U1** for weak equivalence it then follows that $D_0 \simeq S_0$.

6. A Generalized Specification of Handover

In this section we generalize the specification given in Section 4 by allowing more than one Mobile Station, more than two Base Stations, and several radio channels in each Base Station. This means that the Mobile Switching Center must perform a certain amount of bookkeeping. These aspects can, as we will see, easily be modeled in the π-calculus.

We give two generalized specifications of the MSC, one abstract and one more concrete. In both these specifications the MSC consists of one Communication Controller (CC) for each active connection to a Mobile Station and one Handover Controller (HC) for each inactive radio channel. In the abstract specification the CC and the HC agents communicate via a link l which is internal to the MSC. In the more concrete specification we introduce a new agent, H_n^m, which models the intelligence of the system, i.e., H_n^m makes the decision when and to which new radio channel an MS shall switch. A (generalized) Base Station (BS) consists of one *Radio Channel Controller* (RCC) for each radio channel controlled by the base station. Each RCC has a fixed link for communication

$$HC(l,f,m) \stackrel{\mathrm{def}}{=} (k)\Big(\bar{l}\,k\,.\,\bar{k}\,f\,.\,\bar{k}\,m\,.\,k(f_{new})\,.\,k(m_{new})\,.\,HC(l,f_{new},m_{new})\Big)$$

$$HCE(l,\widetilde{f_1^p},\widetilde{m_1^p}) \equiv HC(l,f_1,m_1) \mid \cdots \mid HC(l,f_p,m_p)$$

$$\begin{aligned} CC(in,l,f) \stackrel{\mathrm{def}}{=}\; & in(v)\,.\,\bar{f}\,\mathtt{data}\,.\,\bar{f}\,v\,.\,CC(in,l,f) \\ & + l(k)\,.\,k(f_{new})\,.\,k(m_{new})\,.\,\bar{f}\,\mathtt{ho\_cmd}\,.\,\bar{f}\,m_{new}\,. \\ & \Big(\; f_{new}[\mathtt{ho\_com}]\,.\,\bar{f}\,\mathtt{ch\_rel}\,.\,f(m)\,. \\ & \quad \bar{k}\,f\,.\,\bar{k}\,m\,.\,CC(in,l,f_{new}) \\ & + f[\mathtt{ho\_fail}]\,.\,\bar{k}\,f_{new}\,.\,\bar{k}\,m_{new}\,.\,CC(in,l,f)\Big) \end{aligned}$$

$$CCE(\widetilde{in_1^a},l,\widetilde{f_1^a}) \equiv CC(in_1,l,f_1) \mid \cdots \mid CC(in_a,l,f_a)$$

$$MSC(\widetilde{in_1^a},\widetilde{f_1^a},\widetilde{f'^p_1},\widetilde{m_1^p}) \equiv (l)\Big(\; HCE(l,\widetilde{f'^p_1},\widetilde{m_1^p}) \mid CCE(\widetilde{in_1^a},l,\widetilde{f_1^a})\Big)$$

Fig. 7. An abstract specification of the MSC.

with the MSC. Fig. 13 shows the structure of an abstract system consisting of two Mobile Stations, two Base Stations and five radio channels. Fig. 14 shows the more concrete version of the MSC for the system. We will in this section give specifications of the (generalized) CC, HC, RCC and MS. We will then, by an example, demonstrate how to build specifications with any (fixed) number of Base Stations, radio channels, and Mobile Stations. In the specifications we will write $\widetilde{x_1^n}$ for $x_1,\ldots,x_n$ and $\widehat{x_1^n}$ for $x_1\ldots x_n$.

6.1. The Mobile Switching Center

6.1.1. *An Abstract Generalization*

As before, the MSC is the coordinator of the system: it directs input to the Mobile Stations and initiates handovers. In the following, assume that there are a mobile stations, each with one active radio channel, and p passive radio channels available for handovers. The MSC is composed of two different kinds of agents (see Fig. 7): for each active connection to a Mobile Station there is a CC agent which coordinates the communication between the MSC and the Mobile Station and for each inactive radio channel there is an HC agent which initiates handovers to that channel.

A handover starts when an HC agent sends its private name k via the com-

$$
\begin{array}{l}
HC(k,f,m) \\
\quad \stackrel{\mathrm{def}}{=} \quad k(l)\,.\,\overline{l}\,k\,.\,\overline{k}\,f\,.\,\overline{k}\,m\,.\,k(f_{new})\,.\,k(m_{new})\,.\,HC(k,f_{new},m_{new}) \\
\\
HCE(\widetilde{k_1^p},\widetilde{f_1^p},\widetilde{m_1^p}) \\
\quad \equiv \quad HC(k_1,f_1,m_1) \mid \cdots \mid HC(k_p,f_p,m_p) \\
\\
CC(in,l,f,c) \\
\quad \stackrel{\mathrm{def}}{=} \qquad in(v)\,.\,\overline{f}\,\mathtt{data}\,.\,\overline{f}\,v\,.\,CC(in,l,f,c) \\
\qquad\qquad + \; l(k)\,.\,k(f_{new})\,.\,k(m_{new})\,.\,\overline{f}\,\mathtt{ho\_cmd}\,.\,\overline{f}\,m_{new}\,. \\
\qquad\qquad\qquad \Big(\quad f_{new}[\mathtt{ho\_com}]\,.\,\overline{f}\,\mathtt{ch\_rel}\,.\,f(m)\,. \\
\qquad\qquad\qquad\qquad\qquad \overline{k}\,f\,.\,\overline{k}\,m\,.\,\overline{c}\,l\,.\,\overline{l}\,k\,.\,CC(in,l,f_{new},c) \\
\qquad\qquad\qquad\quad + \; f[\mathtt{ho\_fail}]\,.\,\overline{k}\,f_{new}\,.\,\overline{k}\,m_{new}\,.\,\overline{c}\,l\,.\,\overline{l}\,k\,.\,CC(in,l,f,c)\Big) \\
\\
CCE(\widetilde{in_1^a},\widetilde{l_1^a},\widetilde{f_1^a},c) \\
\quad \equiv \quad CC(in_1,l_1,f_1,c) \mid \cdots \mid CC(in_a,l_a,f_a,c) \\
\\
H_n^m(\widetilde{k_1^n},\widetilde{l_1^m},c) \\
\quad \stackrel{\mathrm{def}}{=} \qquad \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{m}\overline{k_i}\,l_j\,.\,H_{n-1}^{m-1}(\widetilde{k_1^{i-1}},\widetilde{k_{i+1}^{n}},\widetilde{l_1^{j-1}},\widetilde{l_{j+1}^{m}},c) \\
\qquad\qquad + \; c(l')\,.\,l'(k')\,.\,H_{n+1}^{m+1}(\widetilde{k_1^n},k',\widetilde{l_1^m},l',c) \qquad\qquad\qquad n,m \geq 0 \\
\\
MSC(\widetilde{in_1^a},\widetilde{f_1^a},\widetilde{f'^p_1},\widetilde{m_1^p}) \\
\quad \equiv \quad (\widehat{k_1^p})\Big(\quad HCE(\widetilde{k_1^p},\widetilde{f'^p_1},\widetilde{m_1^p}) \\
\qquad\qquad\qquad \mid \\
\qquad\qquad\qquad (\widehat{l_1^a}\,c)\Big(\quad H_p^a(\widetilde{k_1^p},\widetilde{l_1^a},c) \\
\qquad\qquad\qquad\qquad\qquad \mid \\
\qquad\qquad\qquad\qquad\qquad CCE(\widetilde{in_1^a},\widetilde{l_1^a},\widetilde{f_1^a},c)\Big)\Big)
\end{array}
$$

Fig. 8. A more concrete specification of the MSC.

mon port l to a CC agent. The name k is thereafter used for communication between the CC agent and the HC agent. The handover proceeds with the HC sending the names f and m to the CC via the link k. The name f is interpreted as the name of the fixed link to the RCC controlling the new radio channel and the name m is interpreted as the name of the new radio channel. The CC then transmits a `ho_cmd` message followed by the name of new radio channel, m, to the RCC currently in contact with the MS (via the old fixed link). The handover is completed as in Section 4. If it succeeds (a `ho_com` message is received on the new fixed link), the CC requests the now unused radio channel from the old BS and transmits it together with the old fixed link to the HC via the link k. If the handover fails (a `ho_fail` message is received on the old fixed link) the newly received links f and m are returned to the HC via the link k.

$$
\begin{aligned}
&RCC_a(f,m) \\
&\quad \stackrel{\text{def}}{=} \quad f : [\,\texttt{data} \Rightarrow f(v) \,.\, \overline{m}\,\texttt{data} \,.\, \overline{m}\,v \,.\, RCC_a(f,m), \\
&\qquad\qquad \texttt{ho\_cmd} \Rightarrow f(v) \,.\, \overline{m}\,\texttt{ho\_cmd} \,.\, \overline{m}\,v \,. \\
&\qquad\qquad\qquad \Big(\quad f[\texttt{ch\_rel}] \,.\, \overline{f}\,m \,.\, RCC_p(f,m) \\
&\qquad\qquad\qquad\quad + m[\texttt{ho\_fail}] \,.\, \overline{f}\,\texttt{ho\_fail} \,.\, RCC_a(f,m)\Big)\,] \\[1em]
&RCC_p(f,m) \\
&\quad \stackrel{\text{def}}{=} \quad m[\texttt{ho\_acc}] \,.\, \overline{f}\,\texttt{ho\_com} \,.\, RCC_a(f,m) \\[1em]
&MS(m,out) \\
&\quad \stackrel{\text{def}}{=} \quad m : [\texttt{data} \Rightarrow m(v) \,.\, \overline{out}\,v \,.\, MS(m,out), \\
&\qquad\qquad \texttt{ho\_cmd} \Rightarrow m(m_{new}) \,. \\
&\qquad\qquad\qquad \Big(\quad \overline{m_{new}}\,\texttt{ho\_acc} \,.\, MS(m_{new},out) \\
&\qquad\qquad\qquad\quad + \overline{m}\,\texttt{ho\_fail} \,.\, MS(m,out)\Big)\,]
\end{aligned}
$$

Fig. 9. Generalized specifications of the BS and the MS.

By using k in the communications between HC and CC we guarantee that the same CC receives both names f and m from the HC in the initialization of the handover. We call this method of exporting a local name and then sending a sequence of data (f and m) over it a *molecular* action [MPW89a]. In this case this maneuver also guarantees that the same HC receives the names of the unused channels after the handover.

The main difference compared to our earlier specification in Section 4 is that also the (names of the) fixed links are stored in, and received from, the Handover Controllers. The collection of fixed links connecting a Base Station with the Mobile Switching Center (one for each RCC in the BS) may, in an implementation of the system, be multiplexed on the same physical medium. Observe also that each CC in the MSC administrating an active connection to a Mobile Station has its own input channel. All these input channels can be multiplexed onto a single input channel. To avoid the extra addressing necessary, which is irrelevant to the handover procedure, we have omitted this multiplexing/demultiplexing.

6.1.2. A More Concrete Generalization

In the concrete specification of the MSC (see Fig. 8) we introduce the agent H_n^m. This agent has a link k_i to each HC_i and a link l_j to each CC_j. Furthermore it has a link c common to all CC. Intuitively, H_n^m replaces the single link l in the abstract specification above. We can think of the agent H_n^m as representing the intelligence in the system; it initiates a handover by selecting a link k_i (to a HC) and a link l_j (to a CC) and transmits the name l_j via the link k_i. The handover then proceeds almost as in the previous case. The HC agent receiving the name l_j transmits via this name its link k_i to the selected CC agent. When the handover is completed (success or failure) the CC agent reports this fact to the H_n^m agent by transmitting, via the common link c, its name l_j after which the

$$
\begin{aligned}
&HCE(l, f_2, f_3, f_4, m_2, m_3, m_4) \\
&\quad \equiv \quad HC(l, f_2, m_2) \mid HC(l, f_3, m_3) \mid HC(l, f_4, m_4) \\[1ex]
&CCE(in_1, in_2, l, f_1, f_5) \\
&\quad \equiv \quad CC(in_1, l, f_1) \mid CC(in_2, l, f_5) \\[1ex]
&MSC(in_1, in_2, f_1, f_5, f_2, f_3, f_4, m_2, m_3, m_4) \\
&\quad \equiv \quad (l)\Big(HCE(l, f_2, f_3, f_4, m_2, m_3, m_4) \mid CCE(in_1, in_2, l, f_1, f_5)\Big)
\end{aligned}
$$

Fig. 10. A specification of an abstract MSC handling two active and three passive radio channels.

$$
\begin{aligned}
&HCE(k_1, k_2, k_3, f_2, f_3, f_4, m_2, m_3, m_4) \\
&\quad \equiv \quad HC(k_1, f_2, m_2) \mid HC(k_2, f_3, m_3) \mid HC(k_3, f_4, m_4) \\[1ex]
&CCE(in_1, in_2, l_1, l_2, f_1, f_5, c) \\
&\quad \equiv \quad CC(in_1, l_1, f_1, c) \mid CC(in_2, l_2, f_5, c) \\[1ex]
&MSC(in_1, in_2, f_1, f_5, f_2, f_3, f_4, m_2, m_3, m_4) \\
&\quad \equiv \quad (k_1\, k_2\, k_3\, l_1\, l_2\, c)\Big(HCE(k_1, k_2, k_3, f_2, f_3, f_4, m_2, m_3, m_4) \\
&\qquad\qquad\qquad\qquad\quad \mid \\
&\qquad\qquad\qquad\qquad\quad H_3^2(k_1, k_2, k_3, l_1, l_2, c) \\
&\qquad\qquad\qquad\qquad\quad \mid \\
&\qquad\qquad\qquad\qquad\quad CCE(in_1, in_2, l_1, l_2, f_1, f_5, c)\Big)
\end{aligned}
$$

Fig. 11. A specification of a more concrete MSC handling two active and three passive radio channels.

CC agent returns the name k_i to H_n^m via the link l_j. This is again a molecular action since more than one handover can be completed at the same time.

6.2. A Generalized System

In the generalized specification a Base Station (see Fig. 9) consists of a number of Radio Channel Controllers executing in parallel. Each RCC is connected to the MSC via a fixed link f and controls one radio channel m. The operation of an RCC is precisely the operation of the Base Station in our earlier example. A Base Station in Section 4 only handles one radio channel each and hence consists of only one RCC.

The Mobile Stations of the generalized system are given by the same specifications as in Section 4.

In Fig. 10 and Fig. 11 we specify an abstract and a concrete MSC handling five radio channels, two active and three passive, and in Fig. 12 we specify a generalized system consisting of two Base Stations, five radio channels and two

$$
\begin{aligned}
&BS_1(f_1, f_2, m_1, m_2)\\
&\quad \equiv \quad RCC_a(f_1, m_1) \mid RCC_p(f_2, m_2)\\[1ex]
&BS_2(f_3, f_4, f_5, m_3, m_4, m_5)\\
&\quad \equiv \quad RCC_p(f_3, m_3) \mid RCC_p(f_4, m_4) \mid RCC_a(f_5, m_5)\\[1ex]
&System\\
&\quad \equiv \quad (\widehat{f_1^5}\,\widehat{m_1^5})\Big(MSC(\mathtt{in}_1, \mathtt{in}_2, f_1, f_5, f_2, f_3, f_4, m_2, m_3, m_4)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad BS_1(f_1, f_2, m_1, m_2)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad BS_2(f_3, f_4, f_5, m_3, m_4, m_5)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad MS(m_1, \mathtt{out}_1)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad MS(m_5, \mathtt{out}_2)\Big)
\end{aligned}
$$

Fig. 12. A specification of a system consisting of two Base Stations, five radio channels and two Mobile Stations.

Mobile Stations. The structure of the abstract system is given in Fig. 13 and the structure of the more concrete MSC is given in Fig. 14.

Note that the specification in Fig. 12 is not structured into an active and a passive part as our earlier specification in Section 4. In our generalized specification some of the Radio Channel Controllers in a Base Station may belong to an active part and other may belong a passive part. Hence the BS cannot be associated with either part. However, we could have structured the system into an active and a passive part by not grouping together the Radio Channel Controllers into Base Stations. Note also that the specifications can obviously be generalized to any number of components.

The external behaviour of the system is independent of the choice of MSC. In general, a system consists of a mobile stations, each with a port $\mathtt{out}_i$, and each connected with a Communication Controller with a corresponding port $\mathtt{in}_i$, for $1 \leq i \leq a$. There are also p unused radio channels ($p \geq 0$) which are available for handovers. We expect the system to behave as a parallel buffers S_0 as explained in Section 5. However, during a handover a mobile station temporarily uses two radio channels, so if $p < a$ the handovers cannot occur independently. We can model this situation by introducing p semaphores, one for each free radio channel. In order to accomplish a handover, a buffer must have access to a semaphore. We conjecture that the external behaviour of the system is given by the agent SS in Fig. 15.

7. Discussion

We have formally specified the handover procedure in the π-calculus. A link in the network is represented by two agents sharing a name; the mobility of nodes, as manifested by their changing capabilities to communicate with different Base

Here goes figure 13.

Fig. 13. The structure of the abstract generalized system.

Here goes figure 14

Fig. 14. The structure of the more concrete MSC.

$$
\begin{aligned}
&Buf_0(in, out, l)\\
&\quad \stackrel{\mathrm{def}}{=} \quad in(v) \,.\, Buf_1(in, out, l, v)\\
&\qquad\quad + l(s) \,.\, \overline{s}\, s \,.\, Buf_0(in, out, l)\\[1ex]
&Buf_1(in, out, l, v_1)\\
&\quad \stackrel{\mathrm{def}}{=} \quad in(v) \,.\, Buf_2(in, out, l, v_1, v)\\
&\qquad\quad + \overline{out}\, v_1 \,.\, Buf_0(in, out, l)\\
&\qquad\quad + l(s) \,.\, \overline{out}\, v_1 \,.\, \overline{s}\, s \,.\, Buf_0(in, out, l)\\[1ex]
&Buf_2(in, out, l, v_1, v_2)\\
&\quad \stackrel{\mathrm{def}}{=} \quad in(v) \,.\, Buf_3(in, out, l, v_1, v_2, v)\\
&\qquad\quad + \overline{out}\, v_1 \,.\, Buf_1(in, out, l, v_2)\\
&\qquad\quad + l(s) \,.\, \overline{out}\, v_1 \,.\, \overline{out}\, v_2 \,.\, \overline{s}\, s \,.\, Buf_0(in, out, l)\\[1ex]
&Buf_3(in, out, l, v_1, v_2, v_3)\\
&\quad \stackrel{\mathrm{def}}{=} \quad \overline{out}\, v_1 \,.\, Buf_2(in, out, l, v_2, v_3)\\[1ex]
&Sem(l)\\
&\quad \stackrel{\mathrm{def}}{=} \quad (s)\Big(\overline{l}\, s \,.\, s(s) \,.\, Sem(l)\Big)\\[1ex]
&SS\\
&\quad \equiv \quad (l)\Big(Buf_0(\mathtt{in}_1, \mathtt{out}_1, l) \mid \cdots \mid Buf_0(\mathtt{in}_a, \mathtt{out}_a, l)\\
&\qquad\qquad \mid\\
&\qquad\qquad \underbrace{Sem(l) \mid \cdots \mid Sem(l)}_{p \text{ times}}\Big)
\end{aligned}
$$

Fig. 15. A high level view of the generalized system.

Stations, is represented by the transmissions of such names between agents. The formal specification provides a precise description of how the stations interact to accomplish a handover. Since each interaction is described in an abstract way, the formal specification gives a comprehensible high-level definition.

Our specification may serve as a basis for an abstract understanding of the handover procedure and for experiments through simulation. Moreover, the formal semantics enable mathematically precise analyses of the behaviour. One example of this was presented at the end of Section 4, where it was shown that a successful handover does not change the network as perceived by the end users. Another example was presented in Section 5 where we presented a high-level view of the protocol, and proved it behaviourally equivalent with our specification. Yet another example would be to formulate a service in a modal logic and prove that the protocol fulfils that service. Such methods have been used with CCS and are currently being developed for the π-calculus [MPW91].

Although the mobility of nodes cannot be directly represented in other description formalisms, it can sometimes be "coded" into them. For example, the transmission $\overline{x}y$ of y on x can be coded as a transmission on a universal ether;

this transmission contains two fields x and y. A reception $x(y)$ is correspondingly coded as a reception on the ether which discards the message unless the first field is x. In this way the handover procedure has been coded into LOTOS [FO91]. We find the resulting specification less clear to work with, partly because scope migration cannot be modelled. The advantage of using LOTOS is that automatic tools are available for analysis; with these tools our manual verification has been confirmed.

The efforts of GSM until now are concentrated on describing the Mobile Station in great detail, including characteristics of the radio channels. There has been no attempt to fully define the handover procedure nor to provide a high-level definition of the mobile network. It is our opinion that this concentration may have undesirable consequences in that low-level design decisions are made too early.

Our description of the handover procedure can be extended in many ways. For examples, other aspects such as call setup and termination can be described. There are various additional ways in which a handover may fail, and there are various recovery procedures associated with failures. It is probably straightforward to define these failures and recoveries in the π-calculus.

Acknowledgements

We are grateful to Hans Hansson, Bengt Jonsson, Björn Victor and the anonymous referees for valuable comments on the manuscript. This work was partially supported by the Swedish Board for Technical Development 89-01218P (ESPRIT/BRA project 3006, CONCUR), and the Swedish Telecommunication Administration (project: PROCOM).

References

[BA90] G. Bruns and S. Anderson. The formalization and analysis of a communications protocol. Technical report, University of Edinburgh, Dept. of Computer Science, 1990.

[Bae90] J.C.M. Baeten, editor. *Applications of Process Algebra.* Cambridge University Press, 1990.

[BK84] J. Bergstra and J. Klop. Verification of an alternating bit protocol by means of process algebra. Technical report, Centrum voor Wiskunde en Informatica, 1984.

[BSV88] G. Boudol, R. de Simone, and D. Vergamini. Experiment with AUTO and AUTOGRAPH on a simple case of sliding window protocol. Technical Report 870, INRIA, 1988.

[CCI85] CCITT. *CCITT Recommendation I400 (Q920), I441 (Q921)*, 1985.

[EFJ91] P. Ernberg, L. Fredlund, and B. Jonsson. Specification and validation of a simple overtaking protocol in LOTOS. In *Proc. 4^{th} International Conference on Formal Description Techniques*, 1991.

[EN86] U. Engberg and M. Nielsen. A calculus of communicating systems with label-passing. Technical Report DAIMI PB-208, Comp. Sc. Department, Univ. of Aarhus, Denmark, 1986.

[ETS89] ETSI/GSM. *GSM Recommendation 04.08, Mobile Radio Interface Layer 3 Specification*, 1989.

[FO91] L. Fredlund and F. Orava. Modeling dynamic communication structures in LOTOS. In *Proc. 4^{th} International Conference on Formal Description Techniques*, 1991.

[Koo85] C. Koomen. Algebraic specification and verification of protocols. *Science of Computer Programming*, 5(1):1–36, 1985.

[LM87] K. Larsen and R. Milner. A complete protocol verification using relativized bisimulation. In *Proceedings of the 14:th Colloquium on Automata, Languages and Programming*, 1987. Springer Verlag Lecture Notes in Computer Science 267.

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

[MPW89a] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, part I. Technical Report ECS-LFCS-89-85, Department of Computer Science, University of Edinburgh, 1989. Accepted for publication in *Information and Computation*.

[MPW89b] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, part II. Technical Report ECS-LFCS-89-86, Department of Computer Science, University of Edinburgh, 1989. Accepted for publication in *Information and Computation*.

[MPW91] R. Milner, J. Parrow, and D. Walker. Modal logics for mobile processes. Technical Report R91:03, SICS, 1991. Accepted for publication in *Theoretical Computer Science*.

[Naz89] A. Nazari. Formell beskrivning av kommunikationssystem med dynamisk topologi (in Swedish). Technical report, Dept. of Computer Systems, Uppsala University, 1989. MSc Thesis.

[Par88] J. Parrow. Verifying a CSMA/CD-protocol with CCS. In *Protocol Specification, Testing, and Verification VIII*, pages 373–384, 1988. North-Holland.

[SFD85] S.A. Smolka, A.J. Frank, and S.K. Debray. Testing protocol robustness the CCS way. In *Protocol Specification, Testing, and Verification IV (1984)*, pages 93–108, 1985. North-Holland.

[vEVD89] P.H.J. van Eijk, C.A. Vissers, and M. Diaz, editors. *The Formal Description Technique LOTOS*. North-Holland, 1989.

Appendix

In this appendix we will show that the protocol *System* is weakly equivalent to S_0. The proof is quite long and we will omit some uninteresting parts. We will frequently collect the results obtained so far; in this way the presentation becomes longer but easier to follow. All our results will hold for *all* instances of v_1, v_2, and v_3. We will throughout this appendix use the convention of writing τ^n for $\underbrace{\tau\,.\,\tau\,.\,\ldots\,.\,\tau}_{n \text{ times}}$.

We will start with the expansion of $MSC(f_a, f_p, m)$:

$$\begin{array}{ll}
MSC(f_a,f_p,m) & \\
\quad \equiv \ (l)\Big(HC(l,m) \mid CC(f_a,f_p,l)\Big) & \\
\quad \equiv \ (l)\Big(\overline{l}\,m\,.\,l(m)\,.\,HC(l,m) \mid \mathtt{in}(v)\,.\,\cdots + l(m_{new})\,.\,\cdots\Big) & \\
\quad \dot{\sim} \ \quad \mathtt{in}(v)\,.\,(l)\Big(\overline{l}\,m\,.\,l(m)\,.\,HC(l,m) \mid \overline{f_a}\,\mathtt{data}\,.\,\cdots\Big) & \text{(A)} \\
\qquad + \ \tau\,.\,(l)\Big(l(m)\,.\,HC(l,m) \mid \overline{f_a}\,\mathtt{ho_cmd}\,.\,\overline{f_a}\,m\,.\,\cdots\Big) & \text{(B)}
\end{array}$$

Consider the A part:

$$\begin{array}{l}
\mathtt{in}(v)\,.\,(l)\Big(\overline{l}\,m\,.\,l(m)\,.\,HC(l,m) \mid \overline{f_a}\,\mathtt{data}\,.\,\cdots\Big) \\
\quad \dot{\sim} \ \mathtt{in}(v)\,.\,\overline{f_a}\,\mathtt{data}\,.\,(l)\Big(\overline{l}\,m\,.\,l(m)\,.\,HC(l,m) \mid \overline{f_a}\,v\,.\,\cdots\Big) \\
\quad \dot{\sim} \ \mathtt{in}(v)\,.\,\overline{f_a}\,\mathtt{data}\,.\,\overline{f_a}\,v\,.\,(l)\Big(\overline{l}\,m\,.\,l(m)\,.\,HC(l,m) \mid CC(f_a,f_p,l)\Big) \\
\quad \dot{\sim} \ \mathtt{in}(v)\,.\,\overline{f_a}\,\mathtt{data}\,.\,\overline{f_a}\,v\,.\,(l)\Big(HC(l,m) \mid CC(f_a,f_p,l)\Big) \\
\quad \dot{\sim} \ \mathtt{in}(v)\,.\,\overline{f_a}\,\mathtt{data}\,.\,\overline{f_a}\,v\,.\,MSC(f_a,f_p,m)
\end{array}$$

Consider the B part:

$$\begin{array}{l}
\tau\,.\,(l)\Big(l(m)\,.\,HC(l,m) \mid \overline{f_a}\,\mathtt{ho_cmd}\,.\,\overline{f_a}\,m\,.\,\cdots\Big) \\
\quad \dot{\sim} \ \tau\,.\,\overline{f_a}\,\mathtt{ho_cmd}\,.\,\overline{f_a}\,m\,.\,(l)\Big(l(m)\,.\,HC(l,m) \\
\qquad\qquad\qquad\qquad \mid \\
\qquad\qquad\qquad\qquad (\quad f_p[\mathtt{ho_com}]\cdots \\
\qquad\qquad\qquad\qquad\quad + f_a[\mathtt{ho_fail}]\cdots)\Big) \\
\quad \dot{\sim} \ \tau\,.\,\overline{f_a}\,\mathtt{ho_cmd}\,.\,\overline{f_a}\,m\,.\,\Big(\quad f_p(x)\,.\,B_1(f_a,f_p,m,x) \\
\qquad\qquad\qquad\qquad + f_a(x)\,.\,B_2(f_a,f_p,m,x)\Big)
\end{array}$$

where

$$\begin{array}{l}
B_1(f_a,f_p,m,x) \\
\quad \stackrel{\mathrm{def}}{=} \ (l)\Big([x=\mathtt{ho_com}]\overline{f_a}\,\mathtt{ch_rel}\,.\,\cdots \mid l(m)\,.\,HC(l,m)\Big) \\
B_2(f_a,f_p,m,x) \\
\quad \stackrel{\mathrm{def}}{=} \ (l)\Big([x=\mathtt{ho_fail}]\overline{l}\,m\,.\,CC(f_a,f_p,l) \mid l(m)\,.\,HC(l,m)\Big)
\end{array}$$

Consider $B_1(f_a, f_p, m, x)$. In the next steps we will move the matching operator outside of the scope of l.

$$B_1(f_a, f_p, m, x)\{y/x\}$$
$$= \begin{cases} (l)\Big(\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid l(m)\,.\,HC(l,m)\Big) & \text{if } y = \texttt{ho\_com} \\ (l)\Big(\quad [y=\texttt{ho\_com}]\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \\ \qquad\quad \mid \\ \qquad l(m)\,.\,HC(l,m)\Big) \\ \dot\sim \quad (l)\Big(\mathbf{0} \mid l(m)\,.\,HC(l,m)\Big) \\ \dot\sim \quad \mathbf{0} & \text{if } y \neq \texttt{ho\_com} \end{cases}$$

So, for *all* y it holds:

$$B_1(f_a, f_p, m, x)\{y/x\}$$
$$= \Big([x=\texttt{ho\_com}](l)\Big(\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid l(m)\,.\,HC(l,m)\Big)\Big)\{y/x\}$$

So, by **C1**:

$$f_p(x)\,.\,B_1(f_a, f_p, l, m, x)$$
$$\dot\sim \quad f_p(x)\,.\,\Big([x=\texttt{ho\_com}](l)\Big(\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid l(m)\,.\,HC(l,m)\Big)\Big)$$

Which can be further expanded:

$$\begin{aligned}
&\dot\sim \quad f_p[\texttt{ho_com}](l)\Big(\overline{f_a}\,\texttt{ch_rel}\,.\,\cdots \mid l(m)\,.\,HC(l,m)\Big) \\
&\dot\sim \quad f_p[\texttt{ho_com}]\,.\,\overline{f_a}\,\texttt{ch_rel}\,.\,(l)\Big(f_a(m_{old})\,.\,\cdots \mid l(m)\,.\,HC(l,m)\Big) \\
&\dot\sim \quad f_p[\texttt{ho_com}]\,.\,\overline{f_a}\,\texttt{ch_rel}\,.\,f_a(m_{old})\,.\,(l)\Big(\quad \bar{l}\,m_{old}\,.\,CC(f_p, f_a, l) \\
&\qquad\qquad \mid \\
&\qquad\qquad l(m)\,.\,HC(l,m)\Big) \\
&\dot\sim \quad f_p[\texttt{ho_com}]\,.\,\overline{f_a}\,\texttt{ch_rel}\,.\,f_a(m_{old})\,.\,\tau\,.\,(l)\Big(CC(f_p, f_a, l) \\
&\qquad\qquad \mid \\
&\qquad\qquad HC(l, m_{old})\Big) \\
&\dot\sim \quad f_p[\texttt{ho_com}]\,.\,\overline{f_a}\,\texttt{ch_rel}\,.\,f_a(m_{old})\,.\,\tau\,.\,MSC(f_p, f_a, m_{old})
\end{aligned}$$

The agent $B_2(f_a, f_p, m, x)$ is expanded in the same way as above and the result is:

$$f_a(x)\,.\,B_2(f_a, f_p, m, x) \dot\sim f_a[\texttt{ho\_fail}]\,.\,\tau\,.\,MSC(f_a, f_p, m)$$

So, the B part of the expansion can be expressed as:

$$\begin{aligned}
\tau\,.\,\overline{f_a}\,\texttt{ho_cmd}\,.\,\overline{f_a}\,m\,.\,\Big(\quad & f_p[\texttt{ho_com}]\,.\,\overline{f_a}\,\texttt{ch_rel}\,.\,f_a(m_{old})\,. \\
& \quad \tau\,.\,MSC(f_p, f_a, m_{old}) \\
+ \; & f_a[\texttt{ho_fail}]\,.\,\tau\,.\,MSC(f_a, f_p, m)\Big)
\end{aligned}$$

Finally, putting the parts together yields:

$$
\begin{aligned}
&MSC(f_a, f_p, m)\\
&\quad\dot\sim\quad \mathtt{in}(v)\,.\,\overline{f_a}\,\mathtt{data}\,.\,\overline{f_a}\,v\,.\,MSC(f_a, f_p, m)\\
&\qquad + \tau\,.\,\overline{f_a}\,\mathtt{ho\_cmd}\,.\,\overline{f_a}\,m\,.\Big(\quad f_p[\mathtt{ho\_com}]\,.\,\overline{f_a}\,\mathtt{ch\_rel}\,.\,f_a(m_{old})\,.\\
&\qquad\qquad\qquad\qquad\qquad \tau\,.\,MSC(f_p, f_a, m_{old})\\
&\qquad\qquad\qquad\quad + f_a[\mathtt{ho\_fail}]\,.\,\tau\,.\,MSC(f_a, f_p, m)\Big)
\end{aligned}
$$

Next we will expand the P part of the system. The MSC is part of the P part, so we will use the expansion above:

$$
\begin{aligned}
&P(f_a, f_p)\\
&\quad\equiv\quad (m)\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big)\\
&\quad\equiv\quad (m)\Big(\mathtt{in}(v)\,.\,\cdots + \tau\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&\quad\dot\sim\quad \mathtt{in}(v)\,.\,(m)\Big(\overline{f_a}\,\mathtt{data}\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big) \qquad (P_A)\\
&\qquad + \tau\,.\,(m)\Big(\overline{f_a}\,\mathtt{ho\_cmd}\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big) \qquad (P_B)
\end{aligned}
$$

The expansion of the P_A part is simple and the result is:

$$
\begin{aligned}
&\mathtt{in}(v)\,.\,(m)\Big(\overline{f_a}\,\mathtt{data}\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&\quad\dot\sim\quad \mathtt{in}(v)\,.\,\overline{f_a}\,\mathtt{data}\,.\,\overline{f_a}\,v\,.\,P(f_a, f_p)
\end{aligned}
$$

Next, we expand the P_B part:

$$
\begin{aligned}
&\tau\,.\,(m)\Big(\overline{f_a}\,\mathtt{ho\_cmd}\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&\quad\dot\sim\quad \tau\,.\,\overline{f_a}\,\mathtt{ho\_cmd}\,.\,(m)\Big(\overline{f_a}\,m\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&\quad\dot\sim\quad \tau\,.\,\overline{f_a}\,\mathtt{ho\_cmd}\,.\\
&\qquad\qquad (m)\overline{f_a}\,m\,.\Big((f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&\quad\dot\sim\quad \tau\,.\,\overline{f_a}\,\mathtt{ho\_cmd}\,.\,\overline{f_a}\,(m)\,.\\
&\qquad\Big(\quad f_p(x)\,.\,(\,[x = \mathtt{ho\_com}]\overline{f_a}\,\mathtt{ch\_rel}\,.\,\cdots\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad m[\mathtt{ho\_acc}]\,.\,\cdots)\\
&\qquad + f_a(x)\,.\,(\,[x = \mathtt{ho\_fail}]\tau\,.\,MSC(f_a, f_p, m)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad m[\mathtt{ho\_acc}]\,.\,\cdots)\\
&\qquad + m(x)\,.\,\Big((f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad [x = \mathtt{ho\_acc}]\overline{f_p}\,\mathtt{ho\_com}\,.\,\cdots\Big)\Big)\\
&\quad\dot\sim\quad \tau\,.\,\overline{f_a}\,\mathtt{ho\_cmd}\,.\,\overline{f_a}\,(m)\,.\\
&\qquad\Big(\quad f_p(x)\,.\,P_B^1(f_a, f_p, m, x)\\
&\qquad + f_a(x)\,.\,P_B^2(f_a, f_p, m, x)\\
&\qquad + m(x)\,.\,P_B^3(f_a, f_p, m, x)\Big)
\end{aligned}
$$

Where

$$
\begin{aligned}
&P_B^1(f_a, f_p, m, x)\\
&\quad\stackrel{\text{def}}{=}\ \Big([x=\mathtt{ho\_com}]\overline{f_a}\,\mathtt{ch\_rel}\,.\,\cdots \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&P_B^2(f_a, f_p, m, x)\\
&\quad\stackrel{\text{def}}{=}\ \Big([x=\mathtt{ho\_fail}]\tau\,.\,MSC(f_a, f_p, m) \mid m[\mathtt{ho\_acc}]\,.\,\cdots\Big)\\
&P_B^3(f_a, f_p, m, x)\\
&\quad\stackrel{\text{def}}{=}\ \Big(\ (f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots)\\
&\qquad\qquad \mid\\
&\qquad\qquad [x=\mathtt{ho\_acc}]\overline{f_p}\,\mathtt{ho\_com}\,.\,\cdots\Big)
\end{aligned}
$$

The interesting parts to consider are $P_B^2(f_a, f_p, m, x)$ and $P_B^3(f_a, f_p, m, x)$. This is because $P_B^1(f_a, f_p, m, x)$ will never appear in the complete system (this will be shown when we expand $System$). Simplification of $P_B^2(f_a, f_p, m, x)$ yields:

$$
\begin{aligned}
&P_B^2(f_a, f_p, m, x)\{y/x\}\\
&\dot\sim\ \begin{cases}
\tau\,.\,\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big) & \\
+\ m(y)\,.\,\Big(\tau\,.\,MSC(f_a, f_p, m) & \\
\qquad\mid & \\
\qquad [y=\mathtt{ho\_acc}]\,.\,\cdots\Big) & \text{if } y = \mathtt{ho\_fail}\\
BS_p(f_p, m) & \text{if } y \neq \mathtt{ho\_fail}
\end{cases}
\end{aligned}
$$

For $P_B^3(f_a, f_p, m, x)$ we have two possibilities, namely $x = \mathtt{ho\_acc}$ and $x \neq \mathtt{ho\_acc}$. We start with the second case:

$$
\begin{aligned}
&P_B^3(f_a, f_p, m, x)\{y/x\}\\
&\quad\dot\sim\ \Big((f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots) \mid [y=\mathtt{ho\_acc}]\cdots\Big)\\
&(y \neq \mathtt{ho\_acc})\\
&\quad\dot\sim\ \Big((f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots) \mid \mathbf{0}\Big)\\
&\quad\dot\sim\ \Big(f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots\Big)
\end{aligned}
$$

Next we consider the case where $x = \mathtt{ho\_acc}$:

$$
\begin{aligned}
&P_B^3(f_a, f_p, m, x)\{\mathtt{ho\_acc}/x\}\\
&\quad\dot\sim\ \Big((f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots) \mid \overline{f_p}\,\mathtt{ho\_com}\,.\,\cdots\Big)\\
&\quad\dot\sim\ \quad f_p(y)\,.\,\Big([y=\mathtt{ho\_com}]\overline{f_a}\,\mathtt{ch\_rel}\,.\,\cdots \mid \overline{f_p}\,\mathtt{ho\_com}\,.\,\cdots\Big)\\
&\qquad\ +\ f_a(y)\,.\,\Big([y=\mathtt{ho\_fail}]\tau\,.\,MSC(f_a, f_p, m) \mid \overline{f_p}\,\mathtt{ho\_com}\,.\,\cdots\Big)\\
&\qquad\ +\ \overline{f_p}\,\mathtt{ho\_com}\,.\,\Big((f_p[\mathtt{ho\_com}]\,.\,\cdots + f_a[\mathtt{ho\_fail}]\,.\,\cdots)\\
&\qquad\qquad\qquad \mid\\
&\qquad\qquad\qquad BS_a(f_p, m)\Big)\\
&\qquad\ +\ \tau\,.\,\Big([y=\mathtt{ho\_com}]\overline{f_a}\,\mathtt{ch\_rel}\,.\,\cdots \mid BS_a(f_p, m)\Big)\{\mathtt{ho\_com}/y\}\\
&\quad\dot\sim\ \quad f_p(y)\,.\,P_B^{31}(f_a, f_p, m, y) + f_a(y)\,.\,P_B^{32}(f_a, f_p, m, y)\\
&\qquad\ +\ \overline{f_p}\,\mathtt{ho\_com}\,.\,P_B^{33}(f_a, f_p, m) + \tau\,.\,P_B^{34}(f_a, f_p, m)
\end{aligned}
$$

where:

$$
\begin{aligned}
&P_B^{31}(f_a, f_p, m, y)\\
&\quad\stackrel{\text{def}}{=}\quad \Big([y = \texttt{ho\_com}]\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid \overline{f_p}\,\texttt{ho\_com}\,.\,\cdots\Big)\\
&P_B^{32}(f_a, f_p, m, y)\\
&\quad\stackrel{\text{def}}{=}\quad \Big([y = \texttt{ho\_fail}]\tau\,.\,MSC(f_a, f_p, m, in) \mid \overline{f_p}\,\texttt{ho\_com}\,.\,\cdots\Big)\\
&P_B^{33}(f_a, f_p, m)\\
&\quad\stackrel{\text{def}}{=}\quad \Big((f_p[\texttt{ho\_com}]\,.\,\cdots + f_a[\texttt{ho\_fail}]\,.\,\cdots) \mid BS_a(f_p, m)\Big)\\
&P_B^{34}(f_a, f_p, m)\\
&\quad\stackrel{\text{def}}{=}\quad \Big([y = \texttt{ho\_com}]\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid BS_a(f_p, m)\Big)\{\texttt{ho\_com}/y\}
\end{aligned}
$$

The agents $P_B^{31}(f_a, f_p, m, y)$, $P_B^{32}(f_a, f_p, m, y)$ and $P_B^{33}(f_a, f_p, m)$ will never appear in the expansion of $System$ and we will not expand them further.

$$
\begin{aligned}
&P_B^{34}(f_a, f_p, m)\\
&\quad\stackrel{\text{def}}{=}\quad \Big([y = \texttt{ho\_com}]\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid BS_a(f_p, m)\Big)\{\texttt{ho\_com}/y\}\\
&\quad\dot\sim\quad \Big(\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid BS_a(f_p, m)\Big)\\
&\quad\dot\sim\quad \overline{f_a}\,\texttt{ch\_rel}\,.\,\Big(f_a(m_{old})\,.\,\cdots \mid BS_a(f_p, m)\Big)\\
&\qquad\quad + f_p(z)\,.\,\Big(\overline{f_a}\,\texttt{ch\_rel}\,.\,\cdots \mid ([z = \texttt{data}]\cdots + [z = \texttt{ho\_cmd}]\cdots)\Big)\\
&\quad\dot\sim\quad \overline{f_a}\,\texttt{ch\_rel}\,.\,\Big(\ f_a(m_{old})\,.\,\Big(\ \tau\,.\,(MSC(f_p, f_a, m_{old}) \mid BS_a(f_p, m))\\
&\qquad\qquad\qquad\qquad\qquad\qquad + f_p(z)\,.\,P_B^{34'''}(f_a, f_p, m, m_{old}, z)\Big)\\
&\qquad\qquad\qquad\qquad + f_p(z)\,.\,P_B^{34'}(f_a, f_p, m, z)\Big)\\
&\qquad\quad + f_p(z)\,.\,P_B^{34''}(f_a, f_p, m, z)
\end{aligned}
$$

where

$$
\begin{aligned}
&P_B^{34'}(f_a, f_p, m, z)\\
&\quad\stackrel{\text{def}}{=}\quad \Big(\ f_a(m_{old})\,.\,\tau\,.\,MSC(f_p, f_a, m_{old}) \mid ([z = \texttt{data}]\,.\,\cdots + [z = \texttt{ho\_cmd}]\,.\,\cdots)\Big)\\
&P_B^{34''}(f_a, f_p, m, z)\\
&\quad\stackrel{\text{def}}{=}\quad \Big(\ \overline{f_a}\,\texttt{ch\_rel}\,.\,f_a(m_{old})\,.\,\tau\,.\,MSC(f_p, f_a, m_{old}) \mid ([z = \texttt{data}]\,.\,\cdots + [z = \texttt{ho\_cmd}]\,.\,\cdots)\Big)\\
&P_B^{34'''}(f_a, f_p, m, m_{old}, z)\\
&\quad\stackrel{\text{def}}{=}\quad \Big(\tau\,.\,MSC(f_p, f_a, m_{old}) \mid ([z = \texttt{data}]\,.\,\cdots + [z = \texttt{ho\_cmd}]\,.\,\cdots)\Big)
\end{aligned}
$$

The agents $P_B^{34'}(f_a, f_p, m, z)$, $P_B^{34''}(f_a, f_p, m, z)$, and $P_B^{34'''}(f_a, f_p, m, m_{old}, z)$ will never occur in the expansion of $System$, thus we do not expand them further. Finally, putting all parts together yields:

$$
\begin{array}{ll}
P(f_a, f_p) & \\
\quad \dot\sim & \texttt{in}(v) \,.\, \overline{f_a}\,\texttt{data} \,.\, \overline{f_a}\, v \,.\, P(f_a, f_p) \\
& + \tau \,.\, \overline{f_a}\,\texttt{ho\_cmd} \,.\, \overline{f_a}\,(m) \,.\, \Big(\quad f_p(x) \,.\, P_B^1(f_a, f_p, m, x) \\
& \qquad\qquad\qquad\qquad + f_a(x) \,.\, P_B^2(f_a, f_p, m, x) \\
& \qquad\qquad\qquad\qquad + m(x) \,.\, P_B^3(f_a, f_p, m, x) \Big)
\end{array}
$$

where $P_B^1(f_a, f_p, m, x)$ is uninteresting and $P_B^2(f_a, f_p, m, x)$ and $P_B^3(f_a, f_p, m, x)$ are given by:

$$
\begin{array}{l}
P_B^2(f_a, f_p, m, x)\{y/x\} \\
\quad \dot\sim \begin{cases}
\quad \tau \,.\, \Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m) \Big) & \\
+ m(y) \,.\, \Big(\tau \,.\, MSC(f_a, f_p, m) & \\
\qquad\qquad \mid & \\
\qquad\qquad [y = \texttt{ho\_acc}] \,.\, \cdots \Big) & \text{if } y = \texttt{ho\_fail} \\
BS_p(f_p, m) & \text{if } y \neq \texttt{ho\_fail}
\end{cases}
\end{array}
$$

and

$$
\begin{array}{l}
P_B^3(f_a, f_p, m, x)\{z/x\} \\
\quad \dot\sim \begin{cases}
\quad\; f_p(y) \,.\, P_B^{31}(f_a, f_p, m, y) & \\
+ \;\; f_a(y) \,.\, P_B^{32}(f_a, f_p, m, y) & \\
+ \;\; \overline{f_p}\,\texttt{ho\_com} \,.\, P_B^{33}(f_a, f_p, m) & \\
+ \; \tau \,.\, P_B^{34}(f_a, f_p, m) & \text{if } z = \texttt{ho\_acc} \\
\Big(f_p[\texttt{ho\_com}] \,.\, \cdots + f_a[\texttt{ho\_fail}] \,.\, \cdots \Big) & \text{if } z \neq \texttt{ho\_acc}
\end{cases}
\end{array}
$$

where $P_B^{31}(f_a, f_p, m, y)$, $P_B^{32}(f_a, f_p, m, y)$ and $P_B^{33}(f_a, f_p, m)$ will never occur in the expansion of $System$, and $P_B^{34}(f_a, f_p, m)$ is given by:

$$
\begin{array}{l}
P_B^{34}(f_a, f_p, m) \\
\quad \dot\sim \quad \overline{f_a}\,\texttt{ch\_rel} \,.\, \Big(\quad f_a(m_{old}) \,.\, \Big(\quad \tau \,.\, (MSC(f_p, f_a, m_{old}) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mid \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad BS_a(f_p, m)) \\
\qquad\qquad\qquad\qquad\qquad\qquad + f_p(z) \,.\, P_B^{34'''}(f_a, f_p, m, m_{old}, z) \Big) \\
\qquad\qquad\qquad\quad + f_p(z) \,.\, P_B^{34'}(f_a, f_p, m, z) \Big) \\
\qquad + \quad f_p(z) \,.\, P_B^{34''}(f_a, f_p, m, z)
\end{array}
$$

where $P_B^{34'}(f_a, f_p, m, z)$, $P_B^{34''}(f_a, f_p, m, z)$, $P_B^{34'''}(f_a, f_p, m, m_{old}, z)$ will never occur in the expansion of $System$.

The Q part of the system can be expanded in the same manner as above. We will only state here the result of this expansion.

$$
\begin{array}{l}
Q(f_a) \\
\quad \dot\sim \quad f_a : [\begin{array}{lll} \texttt{data} & \Rightarrow & f_a(v_1) \,.\, Q'_A(f_a, v_1) \\ \texttt{ho\_cmd} & \Rightarrow & f_a(ch) \,.\, Q'_B(f_a, ch)] \end{array}
\end{array}
$$

where

$$
Q'_A(f_a, v_1) \dot\sim \tau^2 \,.\, \Big(\overline{\texttt{out}}\, v_1 \,.\, Q(f_a) + f_a(x) . Q''_A(f_a, v_1, x) \Big)
$$

and

$$
\begin{aligned}
&Q''_A(f_a, v_1, x)\{y/x\} \\
&\quad \dot\sim \begin{cases}
\begin{aligned} &\quad f_a(v) \,.\, \overline{\mathtt{out}}\, v_1 \,.\, Q'_A(f_a, v) \\ &+ \overline{\mathtt{out}}\, v_1 \,.\, f_a(v) \,.\, Q'_A(f_a, v) \end{aligned} & \text{if } y = \mathtt{data} \\[1ex]
\begin{aligned} &\quad f_a(v) \,.\, \overline{\mathtt{out}}\, v_1 \,.\, Q'_B(f_a, v) \\ &+ \overline{\mathtt{out}}\, v_1 \,.\, f_a(v) \,.\, Q'_B(f_a, v) \end{aligned} & \text{if } y = \mathtt{ho\_cmd} \\[1ex]
(m)\Big(\overline{\mathtt{out}}\, v_1 \,.\, MS(m)\Big) & \text{if } y \notin \{\mathtt{data}, \mathtt{ho\_cmd}\}
\end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&Q'_B(f_a, ch) \\
&\quad \dot\sim \quad \tau^2 \,.\, \Big(\quad \overline{ch}\, \mathtt{ho\_acc} \,.\, Q^3_B(f_a, ch) \\
&\qquad\qquad + \tau \,.\, Q^4_B(f_a) \\
&\qquad\qquad + f_a(w) \,.\, Q''_B(f_a, ch, w)\Big)
\end{aligned}
$$

$$
\begin{aligned}
&Q^3_B(f_a, ch) \\
&\quad \dot\sim \quad f_a(w) \,.\, Q^{31}_B(f_a, ch, w) + ch(w) \,.\, Q^{32}_B(f_a, ch, w)
\end{aligned}
$$

$$
\begin{aligned}
&Q^{31}_B(f_a, ch, w)\{x/w\} \\
&\quad \dot\sim \begin{cases}
\begin{aligned} &\quad ch(u) \,.\, (m)\Big(\overline{f_a}\, m \,.\, BS_p(f_a, m) \\ &\qquad\qquad\qquad \mid \\ &\qquad\qquad\qquad ([u = \mathtt{data}] \,.\, \cdots \mid [u = \mathtt{ho\_cmd}] \,.\, \cdots)\Big) \\ &+ \overline{f_a}\,(m) \,.\, \Big(BS_p(f_a, m) \mid MS(ch)\Big) \end{aligned} & \text{if } x = \mathtt{ch\_rel} \\[1ex]
(m)MS(ch) & \text{if } x \neq \mathtt{ch\_rel}
\end{cases}
\end{aligned}
$$

$$
Q^4_B(f_a) \dot\sim \overline{f_a}\, \mathtt{ho\_fail} \,.\, Q(f_a)
$$

The agents $Q''_B(f_a, ch, w)$, $Q'_B(f_a, ch)$, and $Q^{32}_B(f_a, ch, w)$ will never occur in the expansion of $System$ and will thus not be given here.

Now we are going to expand the initial $System$ using these expansions of P and Q. While expanding, we will skip through some uninteresting parts (these parts will be indicated by dots ($\vdots$)).

$$
\begin{aligned}
&System \\
&\quad \stackrel{\mathrm{def}}{=} \quad (f_a)(f_p)\Big(P(f_a, f_p) \mid Q(f_a)\Big) \\
&\quad \vdots \\
&\quad \dot\sim \qquad \mathtt{in}(v) \,.\, (f_a)(f_p)\Big(\overline{f_a}\, \mathtt{data} \,.\, \cdots \\
&\qquad\qquad\qquad\qquad \mid \\
&\qquad\qquad\qquad\qquad f_a(w)(\quad [w = \mathtt{data}] \,.\, \cdots \\
&\qquad\qquad\qquad\qquad\qquad + [w = \mathtt{ho\_cmd}] \,.\, \cdots)\Big) \\
&\qquad\quad + \quad \tau \,.\, (f_a)(f_p)\Big(\overline{f_a}\, \mathtt{ho\_cmd} \,.\, \cdots \\
&\qquad\qquad\qquad\qquad \mid \\
&\qquad\qquad\qquad\qquad f_a(w)([w = \mathtt{data}] \,.\, \cdots + [w = \mathtt{ho\_cmd}] \,.\, \cdots)\Big) \\
&\quad \dot\sim \quad \mathtt{in}(v) \,.\, A(v) + \tau \,.\, B
\end{aligned}
$$

The A part represents the system after receiving one message to transmit and the B part represents the system after deciding that a handover should take place and no messages are in transit. Consider the A part:

$$\begin{aligned}
&A(v_1)\\
&\quad\stackrel{\text{def}}{=}\quad (f_a)(f_p)\Big(\;\overline{f_a}\,\mathtt{data}\,.\,\cdots\\
&\qquad\qquad\qquad\quad\mid\\
&\qquad\qquad\qquad\quad f_a(w)([w=\mathtt{data}]\,.\,\cdots+[w=\mathtt{ho_cmd}]\,.\,\cdots)\Big)\\
&\quad\vdots\\
&\quad\dot\sim\quad \tau^2\,.\,(f_a)(f_p)\Big(P(f_a,f_p)\mid Q'_A(f_a,v_1)\Big)\\
&\quad\dot\sim\quad \tau^2\,.\,A_{11}(v_1)
\end{aligned}$$

where

$$\begin{aligned}
&A_{11}(v_1)\\
&\quad\stackrel{\text{def}}{=}\quad (f_a)(f_p)\Big(P(f_a,f_p)\mid Q'_A(f_a,v_1)\Big)\\
&\quad\vdots\\
&\quad\dot\sim\quad\;\; \mathtt{in}(v)\,.\,\tau^2\,.\,A_1(v_1,v)\\
&\qquad\quad + \tau^3\,.\,A_2(v_1)\\
&\qquad\quad + \tau\,.\Big(\;\; \mathtt{in}(v)\,.\,\tau\,.\,A_1(v_1,v)\\
&\qquad\qquad\quad + \tau^2\,.\,A_2(v_1)\\
&\qquad\qquad\quad + \tau\,.\Big(\;\; \mathtt{in}(v)\,.\,A_1(v_1,v)\\
&\qquad\qquad\qquad\quad + \tau\,.\,A_2(v_1)\\
&\qquad\qquad\qquad\quad + \overline{\mathtt{out}}\,v_1\,.\,System\Big)\Big)
\end{aligned}$$

The A_1 part represents the system after receiving a second message. We will only state the result of expanding $A_1(v_1,v)$. The A_2 part represents the system after deciding that a handover should take place and there is one message in transit.

$$\begin{aligned}
&A_1(v_1,v_2)\\
&\quad\stackrel{\text{def}}{=}\quad (f_a)(f_p)\Big(\;\overline{f_a}\,\mathtt{data}\,.\,\overline{f_a}\,v_2\,.\,P(f_a,f_p)\\
&\qquad\qquad\qquad\quad\mid\\
&\qquad\qquad\qquad\quad (\overline{\mathtt{out}}\,v_1\,.\,Q(f_a)+f_a(w)\,.\,Q''_A(f_a,v_1,w))\Big)\\
&\quad\vdots\\
&\quad\dot\sim\quad \overline{\mathtt{out}}\,v_1\,.\,A(v_2)+\tau\,.\,A'_1(v_1,v_2)
\end{aligned}$$

where

$$
\begin{aligned}
&A'_1(v_1, v_2)\\
&\quad \dot{\sim} \quad \overline{\texttt{out}}\, v_1 \,.\, \tau \,.\, A_{11}(v_2)\\
&\qquad + \tau \,.\, \Big(\quad \texttt{in}(v) \,.\, A_{12}(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\texttt{out}}\, v_1 \,.\, \tau^2 \,.\, \Big(\tau \,.\, B'_1(v_2) + \overline{\texttt{out}}\, v_2 \,.\, B\Big)\\
&\qquad\qquad + \overline{\texttt{out}}\, v_1 \,.\, A_{11}(v_2)\Big)\\
&A_{12}(v_1, v_2, v_3)\\
&\quad \dot{\sim} \quad \overline{\texttt{out}}\, v_1 \,.\, \tau^2 \,.\, \Big(\overline{\texttt{out}}\, v_2 \,.\, A(v_3) + \tau \,.\, A'_1(v_2, v_3)\Big)
\end{aligned}
$$

The agent $A_{12}(v_1, v_2, v_3)$ represents the system after receiving three messages. The agent $B'_1(v_2)$ represents the system after a decision to perform a handover with one message in transit. Next we expand $A_2(v_1)$:

$$
\begin{aligned}
&A_2(v_1)\\
&\quad \stackrel{\text{def}}{=} \quad (f_a)(f_p)\Big(\overline{f_a}\, \texttt{ho\_cmd} \,.\, \overline{f_a}\,(m) \,.\, \cdots\\
&\qquad\qquad\qquad |\\
&\qquad\qquad\qquad (\overline{\texttt{out}}\, v_1 \,.\, Q(f_a) + f_a(w) \,.\, Q''_A(f_a, v_1, w))\Big)\\
&\quad \dot{\sim} \quad \overline{\texttt{out}}\, v_1 \,.\, (f_a)(f_p)\Big(\overline{f_a}\, \texttt{ho\_cmd} \,.\, \cdots \mid Q(f_a)\Big)\\
&\qquad + \tau \,.\, \Big(\overline{f_a}\,(m) \,.\, \cdots \mid Q''_A(f_a, v_1, \texttt{ho\_cmd})\Big)\\
&\quad \dot{\sim} \quad \overline{\texttt{out}}\, v_1 \,.\, B + \tau \,.\, B'_1(v_1)
\end{aligned}
$$

Simplifying $B'_1(v_1)$:

$$
\begin{aligned}
&B'_1(v_1)\\
&\quad \stackrel{\text{def}}{=} \quad (f_a)(f_p)\Big(\overline{f_a}\,(m) \,.\, \Big(\quad f_p(x) \,.\, P^1_B(f_a, f_p, m, x)\\
&\qquad\qquad\qquad + f_a(x) \,.\, P^2_B(f_a, f_p, m, x)\\
&\qquad\qquad\qquad + m(x) \,.\, P^3_B(f_a, f_p, m, x)\Big)\\
&\qquad\qquad |\\
&\qquad\qquad Q''_A(f_a, v_1, \texttt{ho\_cmd})\Big)\\
&\quad \dot{\sim} \quad (f_a)(f_p)\Big(\overline{f_a}\,(m) \,.\, \cdots\\
&\qquad\qquad |\\
&\qquad\qquad (\quad f_a(v) \,.\, \overline{\texttt{out}}\, v_1 \,.\, Q'_B(f_a, v)\\
&\qquad\qquad\quad + \overline{\texttt{out}}\, v_1 \,.\, f_a(v) \,.\, Q'_B(f_a, v))\Big)\\
&\quad \dot{\sim} \quad \overline{\texttt{out}}\, v_1 \,.\, (f_a)(f_p)\Big(\overline{f_a}\,(m) \,.\, \cdots \mid f_a(v) \,.\, Q'_B(f_a, v)\Big)\\
&\qquad + \tau \,.\, (f_a)(f_p)(m)\Big((f_p(x) \,.\, \cdots + f_a(x) \,.\, \cdots + m(x) \,.\, \cdots)\\
&\qquad\qquad |\\
&\qquad\qquad \overline{\texttt{out}}\, v_1 \,.\, Q'_B(f_a, m)\Big)
\end{aligned}
$$

$$
\begin{aligned}
\dot\sim\quad & \overline{\texttt{out}}\,v_1\,.\,\tau\,.\,(f_a)(f_p)(m)\Big(\;(\;\; f_p(x)\,.\,\cdots + f_a(x)\,.\,\cdots \\
&\qquad\qquad + m(x)\,.\,\cdots) \\
&\qquad\qquad \mid \\
&\qquad\qquad Q'_B(f_a,m)\Big) \\
& +\,\tau\,.\,\overline{\texttt{out}}\,v_1\,.\,(f_a)(f_p)(m)\Big(\;(\;\; f_p(x)\,.\,\cdots + f_a(x)\,.\,\cdots \\
&\qquad\qquad + m(x)\,.\,\cdots) \\
&\qquad\qquad \mid \\
&\qquad\qquad Q'_B(f_a,m)\Big) \\
\dot\sim\quad & \overline{\texttt{out}}\,v_1\,.\,\tau\,.\,B''_1 + \tau\,.\,\overline{\texttt{out}}\,v_1\,.\,B''_1
\end{aligned}
$$

The agent B''_1 represents the system when a handover will take place. The following expansion essentially follows the execution in Section 4. Simplifying B''_1:

$$
\begin{aligned}
& B''_1 \\
\equiv\quad & (f_a)(f_p)(m)\Big((f_p(x)\,.\,\cdots + f_a(x)\,.\,\cdots + m(x)\,.\,\cdots) \\
&\qquad \mid \\
&\qquad Q'_B(f_a,m)\Big) \\
\dot\sim\quad & \tau^2\,.\,\Big((f_a)(f_p)(m)\Big((f_p(x)\,.\,\cdots + f_a(x)\,.\,\cdots + m(x)\,.\,\cdots) \\
&\qquad \mid \\
&\qquad (\;\;\; \overline{m}\,\texttt{ho\_acc}\,.\,Q^3_B(f_a,m) \\
&\qquad\quad + \tau\,.\,Q^4_B(f_a) \\
&\qquad\quad + f_a(x)\,.\,Q''_B(f_a,m,x))\Big)\Big) \\
\dot\sim\quad & \tau^2\,.\,\Big(\;\;\; \tau\,.\,(f_a)(f_p)(m)\Big(P^3_B(f_a,f_p,m,\texttt{ho\_acc}) \\
&\qquad \mid \\
&\qquad Q^3_B(f_a,m)\Big) \\
&\quad + \tau\,.\,(f_a)(f_p)(m)\Big((f_p(x)\,.\,\cdots + f_a(x)\,.\,\cdots + m(x)\,.\,\cdots) \\
&\qquad \mid \\
&\qquad Q^4_B(f_a)\Big)\Big) \\
\dot\sim\quad & \tau^2\,.\,\Big(\;\;\; \tau\,.\,(f_a)(f_p)(m)\Big((\;\;\; f_p(x)\,.\,P^{31}_B(f_a,f_p,m,x) \\
&\qquad + \underline{f_a}(x)\,.\,P^{32}_B(f_a,f_p,m,x) \\
&\qquad + \overline{f_p}\,\texttt{ho\_com}\,.\,P^{33}_B(f_a,f_p,m) \\
&\qquad + \tau\,.\,P^{34}_B(f_a,f_p,m)) \\
&\qquad \mid \\
&\qquad (\;\;\; f_a(x)\,.\,Q^{31}_B(f_a,m,x) \\
&\qquad + m(x)\,.\,Q^{32}_B(f_a,m,x))\Big) \\
&\quad + \tau\,.\,(f_a)(f_p)(m)\Big((\;\;\; f_p(x)\,.\,P^1_B(f_a,f_p,m,x) \\
&\qquad + f_a(x)\,.\,P^2_B(f_a,f_p,m,x) \\
&\qquad + m(x)\,.\,P^3_B(f_a,f_p,m,x)) \\
&\qquad \mid \\
&\qquad (\,\overline{f_a}\,\texttt{ho\_fail}\,.\,Q(f_a))\Big)\Big)
\end{aligned}
$$

$$
\begin{array}{ll}
\dot{\sim} & \tau^2 . \Big(\quad \tau^2 . (f_a)(f_p)(m)\Big((P_B^{34}(f_a, f_p, m)) \\
& \qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad (\quad f_a(x) . Q_B^{31}(f_a, m, x) \\
& \qquad\qquad\qquad\qquad\qquad\quad + m(x) . Q_B^{32}(f_a, m, x))\Big) \\
& \qquad + \tau^2 . (f_a)(f_p)(m)\Big(P_B^2(f_a, f_p, m, \mathtt{ho\_fail}) \mid Q(f_a)\Big)\Big)
\end{array}
$$

$$
\begin{array}{ll}
\dot{\sim} & \tau^2 . \Big(\quad \tau^2 . (f_a)(f_p)(m)\Big((\overline{f_a}\, \mathtt{ch\_rel} . \cdots + f_p(x) . \cdots) \\
& \qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad (\quad f_a(x) . Q_B^{31}(f_a, m, x) \\
& \qquad\qquad\qquad\qquad\qquad\quad + m(x) . Q_B^{32}(f_a, m, x))\Big) \\
& \qquad + \tau^2 . (f_a)(f_p)(m)\Big(\tau . \Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad (m)\Big(BS_a(f_a, m) \mid MS(m)\Big)\Big)\Big)
\end{array}
$$

$$
\begin{array}{ll}
\dot{\sim} & \tau^2 . \Big(\tau^3 . (f_a)(f_p)(m)\Big(\Big(f_a(m_{old}) . \Big(\tau . \Big(MSC(f_p, f_a, m_{old}) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad BS_a(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad + \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad f_p(z) . \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad P_B^{34'''}(f_a, f_p, m, m_{old}, z)\Big) \\
& \qquad\qquad\qquad\qquad\qquad + \\
& \qquad\qquad\qquad\qquad\qquad f_p(z) . B'_{34}(f_a, f_p, m, z)\Big) \\
& \qquad\qquad\qquad\qquad\quad \mid \\
& \qquad\qquad\qquad\qquad\quad Q_B^{31}(f_a, m, \mathtt{ch\_rel})\Big) \\
& \quad + \\
& \quad \tau^3 . (f_a)(f_p)(m)\Big(\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad\quad \mid \\
& \qquad\qquad\qquad\qquad\quad (m)\Big(BS_a(f_a, m) \mid MS(m)\Big)\Big)\Big)
\end{array}
$$

$$
\begin{array}{ll}
\dot{\sim} & \tau^2 . \Big(\quad \tau^4 . (f_a)(f_p)(m)(m')\Big(\Big(\quad \tau . \Big(MSC(f_p, f_a, m') \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad BS_a(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\quad + f_p(z) . \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad P_B^{34'''}(f_a, f_p, m, m', z)\Big) \\
& \qquad\qquad\qquad\qquad\qquad\quad \mid \\
& \qquad\qquad\qquad\qquad\qquad\quad \Big(BS_p(f_a, m') \mid MS(m)\Big)\Big) \\
& \qquad + \tau^3 . (f_a)(f_p)(m)\Big(\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad (m)\Big(BS_a(f_a, m) \mid MS(m)\Big)\Big)\Big)
\end{array}
$$

$$
\begin{aligned}
\dot{\sim}\quad & \tau^2 . \Big(\quad \tau^5 . (f_a)(f_p)(m)(m')\Big(\Big(MSC(f_p, f_a, m') \\
& \qquad\qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad\qquad BS_a(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad\qquad \Big(BS_p(f_a, m') \mid MS(m)\Big)\Big) \\
& \qquad + \tau^3 . (f_a)(f_p)(m)\Big(\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad (m)\Big(BS_a(f_a, m) \mid MS(m)\Big)\Big)\Big) \\
\dot{\sim}\quad & \tau^2 . \Big(\tau^5 . System' + \tau^3 . System''\Big)
\end{aligned}
$$

where

$$
\begin{aligned}
& System' \\
& \quad \stackrel{\mathrm{def}}{=} \quad (f_a)(f_p)(m)(m')\Big(\Big(MSC(f_p, f_a, m') \mid BS_a(f_p, m)\Big) \\
& \qquad\qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad\qquad \Big(BS_p(f_a, m') \mid MS(m)\Big)\Big) \\
& System'' \\
& \quad \stackrel{\mathrm{def}}{=} \quad (f_a)(f_p)(m)\Big(\Big(MSC(f_a, f_p, m) \mid BS_p(f_p, m)\Big) \\
& \qquad\qquad\qquad \mid \\
& \qquad\qquad\qquad (m)\Big(BS_a(f_a, m) \mid MS(m)\Big)\Big)
\end{aligned}
$$

Observe that $System'$ is exactly the result of a successful handover as is shown in Section 4. Furthermore, we have proven (see Section 4) that $System'$ is strongly equivalent to $System$. In the same manner it can be proven that $System''$ is strongly equivalent to $System$. Hence,

$$B_1'' \dot{\sim} \tau^2 . \Big(\tau^5 . System + \tau^3 . System\Big)$$

Thus we have the following result:

$$
\begin{aligned}
& B_1'(v_1) \\
& \quad \dot{\sim} \quad \overline{\mathtt{out}}\, v_1 . \tau^3 . \Big(\tau^5 . System + \tau^3 . System\Big) \\
& \qquad + \tau . \overline{\mathtt{out}}\, v_1 . \tau^2 . \Big(\tau^5 . System + \tau^3 . System\Big)
\end{aligned}
$$

Finally, expansion of B yields:

$$B \quad \dot{\sim} \quad \tau^4 . \Big(\tau^5 . System + \tau^3 . System\Big)$$

In summary we now have the following:

$$
\begin{aligned}
&System\\
&\quad \dot{\sim}\quad \mathtt{in}(v)\,.\,A(v) + \tau^5\,.\,\Big(\tau^5\,.\,System + \tau^3\,.\,System\Big)\\
&A(v_1)\\
&\quad \dot{\sim}\quad \tau^2\,.\,A_{11}(v_1)\\
&A_{11}(v_1)\\
&\quad \dot{\sim}\quad \quad \mathtt{in}(v)\,.\,\tau^2\,.\,A_1(v_1,v) + \tau^3\,.\,A_2(v_1)\\
&\qquad\quad + \tau\,.\,\Big(\quad \mathtt{in}(v)\,.\,\tau\,.\,A_1(v_1,v) + \tau^2\,.\,A_2(v_1)\\
&\qquad\qquad + \tau\,.\,\Big(\quad \mathtt{in}(v)\,.\,A_1(v_1,v) + \tau\,.\,A_2(v_1)\\
&\qquad\qquad\qquad + \overline{\mathtt{out}}\,v_1\,.\,System\Big)\Big)
\end{aligned}
$$

$$
\begin{aligned}
&A_1(v_1,v_2)\\
&\quad \dot{\sim}\quad \overline{\mathtt{out}}\,v_1\,.\,A(v_2) + \tau\,.\,A'_1(v_1,v_2)\\
&A_2(v_1)\\
&\quad \dot{\sim}\quad \overline{\mathtt{out}}\,v_1\,.\,B + \tau\,.\,B'_1(v_1)\\
&A'_1(v_1,v_2)\\
&\quad \dot{\sim}\quad \quad \overline{\mathtt{out}}\,v_1\,.\,\tau\,.\,A_{11}(v_2)\\
&\qquad\quad + \tau\,.\,\Big(\quad \mathtt{in}(v)\,.\,A_{12}(v_1,v_2,v)\\
&\qquad\qquad + \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,\tau^2\,.\,\Big(\tau\,.\,B'_1(v_2) + \overline{\mathtt{out}}\,v_2\,.\,B\Big)\\
&\qquad\qquad + \overline{\mathtt{out}}\,v_1\,.\,A_{11}(v_2)\Big)\\
&A_{12}(v_1,v_2,v_3)\\
&\quad \dot{\sim}\quad \overline{\mathtt{out}}\,v_1\,.\,\tau^2\,.\,\Big(\overline{\mathtt{out}}\,v_2\,.\,A(v_3) + \tau\,.\,A'_1(v_2,v_3)\Big)\\
&B\\
&\quad \dot{\sim}\quad \tau^4\,.\,\Big(\tau^5\,.\,System + \tau^3\,.\,System\Big)\\
&B'_1(v_1)\\
&\quad \dot{\sim}\quad \quad \overline{\mathtt{out}}\,v_1\,.\,\tau^3\,.\,\Big(\tau^5\,.\,System + \tau^3\,.\,System\Big)\\
&\qquad\quad + \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,\tau^2\,.\,\Big(\tau^5\,.\,System + \tau^3\,.\,System\Big)
\end{aligned}
$$

The equations above hold for *all* v_1, v_2, v_3, thus they also hold for strong equivalence. Put

$$
\begin{aligned}
\widetilde{System} \quad = \quad \langle \quad & System, A(v_1), A_{11}(v_1), A_1(v_1,v_2), A_2(v_1), A'_1(v_1,v_2),\\
& A_{12}(v_1,v_2,v_3), B, B'_1(v_1)\rangle
\end{aligned}
$$

and let

$$
\begin{aligned}
\widetilde{D} \quad = \quad \langle \quad & D_0, D_1(v_1), D_2(v_1), D_3(v_1,v_2), D_4(v_1),\\
& D_5(v_1,v_2), D_6(v_1,v_2,v_3), D_7, D_8(v_1)\rangle
\end{aligned}
$$

be agent constants. Define the expressions $E_0(\widetilde{D}), \ldots, E_8(\widetilde{D})$ by:

$$
\begin{aligned}
E_0(\widetilde{D}) &= \mathtt{in}(v) \,.\, D_1(v) + \tau^5 \,.\, \left(\tau^5 \,.\, D_0 + \tau^3 \,.\, D_0\right) \\
E_1(\widetilde{D}) &= \tau^2 \,.\, D_2(v_1) \\
E_2(\widetilde{D}) &= \quad \mathtt{in}(v) \,.\, \tau^2 \,.\, D_3(v_1, v) + \tau^3 \,.\, D_4(v_1) \\
&\quad + \tau \,.\, \Big(\quad \mathtt{in}(v) \,.\, \tau \,.\, D_3(v_1, v) + \tau^2 \,.\, D_4(v_1) \\
&\qquad\quad + \tau \,.\, \Big(\quad \mathtt{in}(v) \,.\, D_3(v_1, v) + \tau \,.\, D_4(v_1) \\
&\qquad\qquad\quad + \overline{\mathtt{out}}\, v_1 \,.\, D_0\Big)\Big) \\
E_3(\widetilde{D}) &= \overline{\mathtt{out}}\, v_1 \,.\, D_1(v_2) + \tau \,.\, D_5(v_1, v_2) \\
E_4(\widetilde{D}) &= \overline{\mathtt{out}}\, v_1 \,.\, D_7 + \tau \,.\, D_8(v_1) \\
E_5(\widetilde{D}) &= \quad \overline{\mathtt{out}}\, v_1 \,.\, \tau \,.\, D_2(v_2) \\
&\quad + \tau \,.\, \Big(\quad \mathtt{in}(v) \,.\, D_6(v_1, v_2, v) \\
&\qquad\quad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \left(\tau \,.\, D_8(v_2) + \overline{\mathtt{out}}\, v_2 \,.\, D_7\right) \\
&\qquad\quad + \overline{\mathtt{out}}\, v_1 \,.\, D_2(v_2)\Big) \\
E_6(\widetilde{D}) &= \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \left(\overline{\mathtt{out}}\, v_2 \,.\, D_1(v_3) + \tau \,.\, D_5(v_2, v_3)\right) \\
E_7(\widetilde{D}) &= \tau^4 \,.\, \left(\tau^5 \,.\, D_0 + \tau^3 \,.\, D_0\right) \\
E_8(\widetilde{D}) &= \quad \overline{\mathtt{out}}\, v_1 \,.\, \tau^3 \,.\, \left(\tau^5 \,.\, D_0 + \tau^3 \,.\, D_0\right) \\
&\quad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \left(\tau^5 \,.\, D_0 + \tau^3 \,.\, D_0\right)
\end{aligned}
$$

Now define $D_i \stackrel{\mathrm{def}}{=} E_i(\widetilde{D})$ for $i = 0, \ldots, 8$. As each D_i is weakly guarded we know that the set of equations, $\widetilde{P} \sim \widetilde{E}(\widetilde{P})$, has unique solutions up to strong equivalence. We also know that $\widetilde{System} \sim \widetilde{E}(\widetilde{System})$ and hence it follows that $\widetilde{System} \sim \widetilde{D}$ and in particular $System \sim D_0$.

We will now prove D_0 observation congruent to the service specification. The first step is to remove the τ-loop in D_0. For this purpose we first simplify the above set of equations under observational congruence by observing that:

$$E_0(\widetilde{D}) \;\simeq\; \mathtt{in}(v_1) \,.\, D_1(v_1) + \tau \,.\, D_0$$

Next we define a new set of equations by:

$$E'_i(\widetilde{D}) = \begin{cases} \mathtt{in}(v_1) \,.\, D_1(v_1) + \tau \,.\, D_0 & \text{if } i = 0 \\ E_i(\widetilde{D}) & \text{if } i \in \{1, \ldots, 8\} \end{cases}$$

Let D'_i, $i = 0, \ldots, 8$ be new agent constants and let $\widetilde{D'} \stackrel{\mathrm{def}}{=} \widetilde{E'}(\widetilde{D'})$. We now know that

$$
\begin{aligned}
\widetilde{E} &\simeq \widetilde{E'} \\
\widetilde{D} &\stackrel{\mathrm{def}}{=} \widetilde{E}(\widetilde{D}) \\
\widetilde{D'} &\stackrel{\mathrm{def}}{=} \widetilde{E'}(\widetilde{D'})
\end{aligned}
$$

and by **U0** we conclude $\widetilde{D} \simeq \widetilde{D'}$ and in particular $D_0 \simeq D'_0$.

Next we are going to remove the τ-loop from D'_0. Define the set of equations $\widetilde{E''}$ by:

$$E''_i(\widetilde{D}) = \begin{cases} \tau\,.\,\mathtt{in}(v_1)\,.\,D_1(v_1) & \text{if } i = 0 \\ E_i(\widetilde{D}) & \text{if } i \in \{1,\ldots,8\} \end{cases}$$

Let D''_i, $i = 0,\ldots,8$ be new agent constants and let $\widetilde{D''} \stackrel{\text{def}}{=} \widetilde{E''}(\widetilde{D''})$. According to **L** we now have $\widetilde{D'} \simeq \widetilde{D''}$ and in particular $D'_0 \simeq D''_0$.

Next let $\widetilde{S} = \langle S_0, S_1(v_1), S_2(v_1,v_2), S_3(v_1,v_2,v_3)\rangle$ be agent constants. Define the expressions $F_0(\widetilde{S}),\ldots,F_3(\widetilde{S})$ by

$$\begin{array}{rcl} F_0(\widetilde{S}) & = & \mathtt{in}(v)\,.\,S_1(v) + \tau\,.\,S_0 \\ F_1(\widetilde{S}) & = & \mathtt{in}(v)\,.\,S_2(v_1,v) + \overline{\mathtt{out}}\,v_1\,.\,S_0 + \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,S_0 \\ F_2(\widetilde{S}) & = & \mathtt{in}(v)\,.\,S_3(v_1,v_2,v) + \overline{\mathtt{out}}\,v_1\,.\,S_1(v_2) + \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,\overline{\mathtt{out}}\,v_2\,.\,S_0 \\ F_3(\widetilde{S}) & = & \overline{\mathtt{out}}\,v_1\,.\,S_2(v_2,v_3) \end{array}$$

So, the specification of S_i in Section 5 can be written $S_i \stackrel{\text{def}}{=} F_i(\widetilde{S})$ for $i = 0,\ldots,3$. In the same manner as above we are now going to remove the τ-loop from the service specification. Let S'_i, $i = 0,\ldots,3$ be new agent constants. We define $\widetilde{F'}$ from $\widetilde{F}$ as

$$F'_i(\widetilde{S}) = \begin{cases} \tau\,.\,\mathtt{in}(v)\,.\,S_1(v) & \text{if } i = 0 \\ F_i(\widetilde{S}) & \text{if } i \in \{1,\ldots,3\} \end{cases}$$

Note that $\widetilde{F'}$ are guarded and sequential. Let $\widetilde{S'} \stackrel{\text{def}}{=} \widetilde{F'}(S')$. From **L** we now know that $\widetilde{S'} \simeq \widetilde{S}$ and in particular $S'_0 \simeq S_0$.

We are finally going to show that D''_0 satisfies the defining equation of S'_0. The first step is to simplify the defining equations for D''_i, $i = 0,\ldots,8$ using the τ-laws of the calculus and identifying and substituting equivalent expressions for an identifier.

$$\begin{array}{ll} D''_7 & \\ \quad\stackrel{\text{def}}{=} & \tau^4\,.\left(\tau^5\,.\,D''_0 + \tau^3\,.\,D''_0\right) \\ \quad\simeq & \tau\,.\,D''_0 \\ D''_8(v_1) & \\ \quad\stackrel{\text{def}}{=} & \quad\overline{\mathtt{out}}\,v_1\,.\,\tau^3\,.\left(\tau^5\,.\,D''_0 + \tau^3\,.\,D''_0\right) \\ & + \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,\tau^2\,.\left(\tau^5\,.\,D''_0 + \tau^3\,.\,D''_0\right) \\ \quad\simeq & \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,D''_0 \\ D''_4(v_1) & \\ \quad\stackrel{\text{def}}{=} & \overline{\mathtt{out}}\,v_1\,.\,D''_7 + \tau\,.\,D''_8(v_1) \\ \quad\simeq & \tau\,.\,\overline{\mathtt{out}}\,v_1\,.\,D''_0 \\ \quad\simeq & D''_8(v_1) \end{array}$$

$$
\begin{aligned}
&D_1''(v_1)\\
&\quad\stackrel{\text{def}}{=}\quad \tau^2 \,.\, D_2''(v_1)\\
&\quad\simeq\quad \tau \,.\, D_2''(v_1)\\
&D_0''\\
&\quad\stackrel{\text{def}}{=}\quad \tau \,.\, \mathtt{in}(v) \,.\, D_1''(v)\\
&\quad\simeq\quad \tau \,.\, \mathtt{in}(v) \,.\, D_2''(v)\\
&D_2''(v_1)\\
&\quad\stackrel{\text{def}}{=}\quad \mathtt{in}(v) \,.\, \tau^2 \,.\, D_3''(v_1, v) + \tau^3 \,.\, D_4''(v_1)\\
&\qquad\quad + \tau \,.\, \Big(\mathtt{in}(v) \,.\, \tau \,.\, D_3''(v_1, v) + \tau^2 \,.\, D_4''(v_1)\\
&\qquad\qquad + \tau \,.\, \big(\mathtt{in}(v) \,.\, D_3''(v_1, v) + \tau \,.\, D_4''(v_1) + \overline{\mathtt{out}}\, v_1 \,.\, D_0''\big)\Big)\\
&\quad\simeq\quad \tau \,.\, \Big(\mathtt{in}(v) \,.\, D_3''(v_1, v) + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, D_0'' + \overline{\mathtt{out}}\, v_1 \,.\, D_0''\Big)
\end{aligned}
$$

$$
\begin{aligned}
&D_5''(v_1, v_2)\\
&\quad\stackrel{\text{def}}{=}\quad \overline{\mathtt{out}}\, v_1 \,.\, \tau \,.\, D_2''(v_2)\\
&\qquad + \tau \,.\, \Big(\mathtt{in}(v) \,.\, D_6''(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \tau^2 \,.\, \big(\tau \,.\, D_8''(v_2) + \overline{\mathtt{out}}\, v_2 \,.\, D_7''\big)\\
&\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, D_2''(v_2)\Big)\\
&\quad\simeq\quad \tau \,.\, \Big(\mathtt{in}(v) \,.\, D_6''(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \underbrace{\big(\tau \,.\, D_8''(v_2) + \overline{\mathtt{out}}\, v_2 \,.\, D_7''\big)}_{\tau \,.\, \overline{\mathtt{out}}\, v_2 \,.\, D_0''}\\
&\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, D_2''(v_2)\Big)\\
&\quad\simeq\quad \tau \,.\, \Big(\mathtt{in}(v) \,.\, D_6''(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \overline{\mathtt{out}}\, v_2 \,.\, D_0''\\
&\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, D_2''(v_2)\Big)
\end{aligned}
$$

$$
\begin{aligned}
&D_3''(v_1, v_2)\\
&\quad\stackrel{\text{def}}{=}\quad \overline{\mathtt{out}}\, v_1 \,.\, D_1''(v_2) + \tau \,.\, D_5''(v_1, v_2)\\
&\quad\simeq\quad \overline{\mathtt{out}}\, v_1 \,.\, D_2''(v_2) + \tau \,.\, \Big(\mathtt{in}(v) \,.\, D_6''(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \overline{\mathtt{out}}\, v_2 \,.\, D_0''\\
&\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, D_2''(v_2)\Big)\\
&\quad\simeq\quad \tau \,.\, \Big(\mathtt{in}(v) \,.\, D_6''(v_1, v_2, v)\\
&\qquad\qquad + \tau \,.\, \overline{\mathtt{out}}\, v_1 \,.\, \overline{\mathtt{out}}\, v_2 \,.\, D_0''\\
&\qquad\qquad + \overline{\mathtt{out}}\, v_1 \,.\, D_2''(v_2)\Big)
\end{aligned}
$$

$$
\begin{aligned}
&D''_6(v_1,v_2,v_3)\\
&\quad\stackrel{\mathrm{def}}{=}\quad \overline{\texttt{out}}\,v_1\,.\,\tau^2\,.\,\Big(\overline{\texttt{out}}\,v_2\,.\,D''_1(v_3)+\tau\,.\,D''_5(v_2,v_3)\Big)\\
&\quad\simeq\quad \overline{\texttt{out}}\,v_1\,.\,\Big(\quad \overline{\texttt{out}}\,v_2\,.\,D''_1(v_3)\\
&\qquad\qquad +\tau\,.\,\Big(\quad \texttt{in}(v)\,.\,D''_6(v_2,v_3,v)\\
&\qquad\qquad\qquad +\tau\,.\,\overline{\texttt{out}}\,v_2\,.\,\overline{\texttt{out}}\,v_3\,.\,D''_0\\
&\qquad\qquad\qquad +\overline{\texttt{out}}\,v_2\,.\,D''_2(v_3)\Big)\\
&\quad\simeq\quad \overline{\texttt{out}}\,v_1\,.\,\underbrace{\tau\,.\,\Big(\quad \texttt{in}(v)\,.\,D''_6(v_2,v_3,v)+\tau\,.\,\overline{\texttt{out}}\,v_2\,.\,\overline{\texttt{out}}\,v_3\,.\,D''_0+\overline{\texttt{out}}\,v_2\,.\,D''_2(v_3)\Big)}_{D''_3(v_2,v_3)}\\
&\quad\simeq\quad \overline{\texttt{out}}\,v_1\,.\,D''_3(v_2,v_3)
\end{aligned}
$$

The result is:

$$
\begin{aligned}
&D''_0\\
&\qquad\simeq\quad \tau\,.\,\texttt{in}(v)\,.\,D''_2(v)\\
&D''_2(v_1)\\
&\qquad\simeq\quad \tau\,.\,\Big(\texttt{in}(v)\,.\,D''_3(v_1,v)+\tau\,.\,\overline{\texttt{out}}\,v_1\,.\,D''_0+\overline{\texttt{out}}\,v_1\,.\,D''_0\Big)\\
&D''_3(v_1,v_2)\\
&\qquad\simeq\quad \tau\,.\,\Big(\quad \texttt{in}(v)\,.\,D''_6(v_1,v_2,v)\\
&\qquad\qquad +\tau\,.\,\overline{\texttt{out}}\,v_1\,.\,\overline{\texttt{out}}\,v_2\,.\,D''_0\\
&\qquad\qquad +\overline{\texttt{out}}\,v_1\,.\,D''_2(v_2)\Big)\\
&D''_6(v_1,v_2,v_3)\\
&\qquad\simeq\quad \overline{\texttt{out}}\,v_1\,.\,D''_3(v_2,v_3)
\end{aligned}
$$

Observe that $D''_2(v_1)$ and $D''_3(v_1,v_2)$ always occur immediately after a prefix in the above expressions. Thus the above set of expressions,

$$\widetilde{D'''} = \langle D''_0, \tau\,.\,D''_2(v_1), \tau\,.\,D''_3(v_1,v_2), D''_6(v_1,v_2,v_3)\rangle$$

satisfies the equations $\widetilde{D'''} \simeq \widetilde{F'}(\widetilde{D'''})$. So we conclude that $\widetilde{D'''} \simeq \widetilde{S'}$ and in particular $D''_0 \simeq S'_0$. Furthermore, we already know that $D''_0 \simeq D'_0 \simeq D_0 \sim$ *System* and $S'_0 \simeq S_0$ and thus we have proven $\mathit{System} \simeq S_0$.