

Department of Computer Science
University of Aarhus
Ny Munkegade
8000 Aarhus C
Denmark

January 2003

Analysis of GSM Handover using Coloured Petri Nets

A Master's Thesis
by

Jonas Martin Thomsen
and
René Manggaard

Contents

1	Introduction	1
1.1	Naming and typesetting conventions	2
1.2	Acknowledgements	2
1.3	Thesis Structure	2
2	GSM Introduction	5
2.1	Functional view of GSM	5
2.1.1	Call management and call processing	5
2.1.2	Radio management	6
2.1.3	Mobility management	6
2.1.4	Charging	6
2.1.5	Security	6
2.2	Logical Architecture	7
2.2.1	Network Switching Subsystem	7
2.2.2	Base Station System	8
2.2.3	Mobile Station	8
2.3	Physical Architecture	10
2.3.1	Physical layout	10
2.3.2	Knowledge in the network	10
2.4	Summary	11
3	GSM Network and Signalling	13
3.1	Interfaces	13
3.1.1	A-interface	14
3.1.2	Abis-interface	15
3.1.3	Air-interface	16
3.2	Procedures in GSM	18
3.2.1	Power ON	18
3.2.2	IMSI Detach and IMSI Attach	18
3.2.3	Location Update	19
3.2.4	Handover	19
3.3	Summary	22

4	Problem Domain	23
4.1	Details of the intra-MSD handover	23
4.1.1	The successful case	23
4.1.2	Failure cases	27
4.1.3	Timers	28
4.2	Interpretation of the problem domain	32
4.2.1	Discussion of the SDLS	32
4.3	The model design	39
4.3.1	SDL vs. CPN	39
4.3.2	The general model design	39
4.3.3	Messages	40
4.4	Summary	44
5	Description of CPN Model	47
5.1	Modelling aspects	47
5.2	CPN pages	50
5.2.1	GSM	50
5.2.2	MSC	51
5.2.3	OldBSC	54
5.2.4	NewBSC	58
5.2.5	OldBTS	59
5.2.6	NewBTS	60
5.2.7	MS	61
5.3	Summary	62
6	Validation of the Model	63
6.1	Model structure	63
6.2	Simulation scenarios	63
6.2.1	Generation of Message sequence charts	64
6.2.2	The scenarios	64
6.3	Summary	68
7	Verification	71
7.1	Discussion of progress and outcome	71
7.2	Analysis of progress and outcome	72
7.2.1	Progress of handover	72
7.2.2	Outcome of handover	73
7.3	Summary	75
8	Future Work	77
9	Conclusion	79
A	Introduction to SDL	81

B	SDLs from GSM 03.09	83
C	CPN Hierarchy	89
D	Occurence Graph Report	91
E	Terminal Nodes of the state space	99
	E.1 HandoverSucceeded	99
	E.2 FailedBack	100
	E.3 CallReleased	101
	E.4 NoEndState	108
	Bibliography	110

Chapter 1

Introduction

Mobile telephones became very popular in the late nineties and are today an important tool for many people. Our way of life demands more and more mobility and availability.

One of the most important technologies used for mobile telephone networks today is the *Global System for Mobile communication* (GSM) technology. The first GSM networks were rolled out during the early nineties and are therefore quite old today. Several newer and far more advanced technologies has been invented since then and some are almost ready to be rolled out. It is very unlikely that modern mobile telephone networks are going to replace GSM completely within the next decade. A long period of interoperability must be expected. The cost of rolling out a new network is enormous; this requires the new technologies to be able to cooperate with the existing GSM networks in order to achieve an acceptable coverage.

The functionality to ensure acceptable quality of a call, when the person using the mobile phone is mobile, is called *handover*. Handover transfers the call transparently from one stationary antenna to another during the call, when the quality of the transmitted data decreases.

Our work is focused on handovers within GSM networks. We started with a joint project on designing a handover mechanism between GSM and a different radio based network. To be able to design such a handover, we started out with an investigation of handover within GSM networks. This investigation turned out to be far more complex than expected, and we decided to limit our research to GSM exclusively.

Because the GSM equipment required to perform a real handover, is huge and expensive, and because gaining access to real operators networks is impossible, we have decided to base our research on a *model* of a GSM network. Through simulations and analysis of the model, we will be able to investigate the behavior of a GSM handover. The model has the advantage of being as abstract as we need compared to real system. This allows us to concentrate our work on the actual handover and not spend our time on mangling with the bits of a real system.

Our work will focus on building the model of the GSM handover, obtaining validity of the model and determine if the outcome of a handover is consistent throughout the network, i.e. all devices agree on the result of the handover.

1.1 Naming and typesetting conventions

The GSM litterature is inconsistent with respect to naming conventions for the GSM entities. For handovers, the GSM recommendations use 'entity-A' for the entity to be handed over from and 'entity-B' for the entity to be handed to. Heine [3] uses the terms 'old entity' and 'new entity' for the same. We chose to follow the conventions from Heine [3], because it describes the flow in the process; the same is not true for 'A' and 'B' from the recommendations.

Our typesetting conventions are, that we use a **sans serif** font for items in CPN models or SDLs. General GSM terms has not been typeset differently than the rest of the text. Program code and extracts from computer generated reports has been typeset using **Courier**.

1.2 Acknowledgements

Several people has helped us through our work on this thesis. We will thank our supervisor Søren Christensen for the guidance through the work on our thesis. Furthermore we thank Thomas Mailund for reviewing several versions of this thesis. Bo Lindstrøm has been very helpful with CP-Net specific problems. Sam Ravnborg, Kim Jensen-Møller, and Jørgen Karkov has been a great resource within GSM specific details and literature.

1.3 Thesis Structure

The thesis is structured in the following way:

Chapter 1: Introduction introduces the project, we present in this master's thesis. It contains our naming and typesetting conventions, a description of the thesis structure, and finally a readers guide.

Chapter 2: GSM Introduction describes the basics of the GSM networks we look at in the thesis. The chapter introduces the general concepts of the network: Its functionallity, logical, and physical architecture.

Chapter 3: GSM Network and Signalling covers more details of the GSM network: Signalling, interfaces, and procedures.

Chapter 4: Problem Domain gives a thorough walk-through of the problem domain. The chapter includes a detailed description of a successful handover, as well as coverage of the different failure conditions. It specifies our modelling base and presents our model design.

Chapter 5: Description of CPN Model describes our model of the GSM handover in Design/CPN tool. It captures our modelling conventions, as well as the individual pages.

Chapter 6: Validation of the Model includes our validation of the model. It consists of some scenarios to validate the major functionality of the model.

Chapter 7: Verification contains our analysis of some of the properties of a GSM handover. The properties are progress of the protocols and consistency of the outcome.

Chapter 8: Future Work gives examples of interesting related work, to be done in the future.

Chapter 9: Conclusion summarise the results of our work during the modelling, simulation and analysis.

We recommend the reader to start with this chapter, where we introduces our work. If the reader is familiar with GSM networks, he might skip chapter 2 and chapter 3. If the technical details of GSM networks are new to the reader, we give an introduction in the two chapters. Chapter 4 explains most of our limitations of the GSM handover and is therefore important to read. Chapter 5 describes our CPN model and is important to read, in order to understand what we have done. We assume that the reader is familiar with the basics of Coloured Petri Nets and the design/CPN tool. In chapter 6 we argue that our model is valid, which is important in order to trust our results. Chapter 7 is our analysis of the handover. The chapter contains technical details of the Design/CPN tool, and it might be hard to read, if you are unfamiliar with state space analysis of CP-Nets. Consult Jensen [14, 15, 16] for an introduction to CPN and Design/CPN. Chapter 8 gives some recommendations for future work. This is interesting to read if you find the covered topic interesting to work with. Chapter 9 concludes our work. Here we sum up, what we have achieved. This chapter is important both if the hole thesis has been read and if you jump directly from the introduction to the conclusion.

Chapter 2

GSM Introduction

Global System for Mobile communication (GSM) is developed in the working groups associated to *European Telecommunications Standards Institute* (ETSI). Our work is based on the recommendations for GSM phase 2, made by ETSI [4] and not any specific implementation of these.

In the following chapter we give an introduction to the general concepts in GSM networks. The first topic is an informal overview of the functionality of a GSM network. The next topic is the logical architecture of the network. This is a description of the components in the network and their respective roles. Finally, we look into the physical aspects of the network, which includes a discussion of the individual entities' knowledge of the GSM network.

2.1 Functional view of GSM

The primary goal of a mobile telephone network like GSM is that a subscriber having a mobile telephone can make and receive calls anywhere. To achieve this goal, some major functions are required, e.g. call management and call processing, radio management, mobility management, charging, and security. In the following sections each of these functions are described.

2.1.1 Call management and call processing

When a subscriber dials a number on his phone, he expects a response from the network; if a connection to the called subscriber could be established he would expect a dial tone and otherwise an error tone. What implements this behavior is *call management* and *call processing*. Call management deals with setting up and terminating calls. This includes finding a route through the network from the calling party to the called party. Call processing is everything between setting up the call and terminating it, e.g. traffic switching, error handling, and re-routing.

2.1.2 Radio management

The wireless communication path in GSM achieved by radio communication. To be able to communicate by radio, both parties need to know which frequency the other party uses; this is decided by the antennas throughout the countryside. When the MS needs to communicate with an antenna it scans the frequencies in order to find the needed one. All matters related to controlling the radio is called *radio management*.

2.1.3 Mobility management

In order to allow a subscriber to receive calls anywhere, the network needs to know something about the location of the mobile phone. To avoid unnecessary network load in areas far away from the phone, the mobile phone notifies the network with its current location, when moved around; when the phone needs to be contacted by the network, only the nearby antennas try to reach it. Another situation is powering the phone on and off; the network is notified when this happens. All procedures regarding the mobility of the mobile phone is called *mobility management*.

2.1.4 Charging

Charging is the registration and billing of the subscribers' use of the mobile phone. Different charging is done depending on the time and the location of the mobile phone. Usually, network operators have reduced prices during off peak hours compared to peak hours. Also, calls outside the operators network is typically charged at a higher rate than calls within the operators network.

2.1.5 Security

When communication is performed by radiowaves, everyone with a radio receiver is able to listen to the communication. In order to preserve privacy, encryption of the communication is needed. This is just one *security* function in the mobile network. Another example is authentication; to be able to charge the correct subscriber, authentication against the network is needed. This is also needed to prevent fraud. Equipment (e.g. mobile phones) is also checked to ensure that e.g. stolen phones cannot be used.

We have now looked at some of the major functions in a GSM network. They were presented generally here, but will throughout the rest of the chapter be described within their respective contexts.

2.2 Logical Architecture

The GSM networks are divided into two logical parts: *Network Switching Subsystem* (NSS) and *Base Station Subsystem* (BSS). The NSS is responsible for call processing, mobility management, and subscriber related functions such as charging and security. The BSS performs the radio related functions towards the *Mobile Stations* (MS), e.g. a mobile phone.

The following sections describe each part of the network in greater detail. The MS is not a part of the fixed network and is therefore covered in its own section (section 2.2.3). Finally we summarise the logical architecture and show the interconnection of all the entities.

2.2.1 Network Switching Subsystem

The call processing part of the NSS is located in the *Mobile Switching Center* (MSC) and the *Gateway-MSC* (G-MSC). The former connects different BSSs, whereas the latter interworks with other networks, e.g. Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), and the Internet.

The subscriber related functions are located in several components: *Home Location Register* (HLR), *Visitor Location Register* (VLR), *Authentication Center* (AUC), and *Equipment Identity Register* (EIR). The HLR and VLR are databases with subscriber information. The former holds all the subscriber data for a specific operator, whereas the latter holds a copy of the subscriber information from an HLR, for all subscribers being serviced by it; this saves unnecessary communication with the HLR. Copying subscriber information from the HLR to the VLR is a part of the mobility management operations in the network. The AUC is performing all security operations, e.g. authentication and key storage, and is always implemented as a part of the HLR. The EIR is a database, which contains black listed MSs, because they are stolen, defective, or unauthorized. When a MS enters a GSM network, it might be checked against the EIR and if black listed, excluded from the network. The EIR is an optional device, that ensures

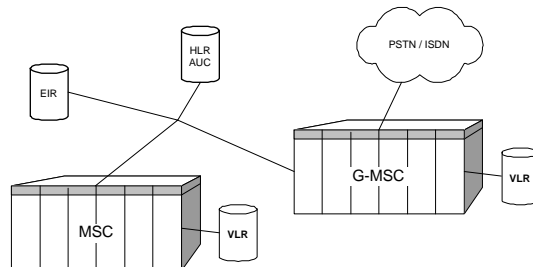


Figure 2.1: The logical structure of the NSS. All entities communicate directly with each other with the exception of the VLRs, which are directly connected to a MSC.

better operation of the network and prevents fraud. Figure 2.1 shows the logical architecture of the NSS.

2.2.2 Base Station System

The primary function of the BSS is to provide connectivity to the MSs and it is implemented as two entities: *Base Station Controller* (BSC) and *Base Transceiver Station* (BTS). The BSC is the controlling unit of the BSS, having several BTSs associated to it. The BSC contains the logic in the BSS and therefore makes all the decisions. An example is handover, where the BSC — assisted by the MS — collects signal quality towards multiple BTSs to determine if a handover is needed and if so which BTS to hand the call to. Handovers are discussed in section 3.2.4.

The BTSs are located around the countryside providing the radio connections to MSs. The BTSs does not contain much logic; they are acting more as a bridge between the radio interface and the backbone network. The logic is placed in either the BSC or the MSC. A single BTS can control several *Transmitter/Receiver* (TRX) modules, each handling a physical antenna. Each TRX defines a cell and can handle up to 8 simultaneous calls. The logical architecture of the NSS is illustrated in figure 2.2.

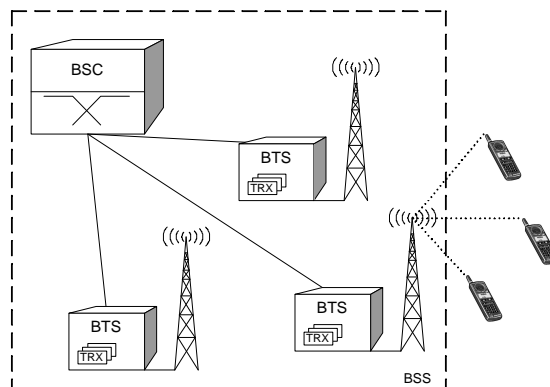


Figure 2.2: The logical structure of the BSS. The BSC controls the sub system, where the BTSs provide the radio link for the MSs.

2.2.3 Mobile Station

The *Mobile Station* (MS) is a device able to communicate with a GSM network. Examples are conventional mobile phones and PCMCIA plug-in cards for a laptop computer. Although the MS is not a part of the wired network, it is important with respect to the functionality of the network. The MS assists the network with measurements of radio signal quality, which are important for handover decisions.

Within wired telephone networks, the telephone represents the subscriber when attached to the network. This is not exactly the case within GSM, where subscriber identity and equipment is separated. The *Subscriber Identity Module* (SIM) inside the MS represents the identity of the subscriber. The MS is useless without a SIM. Authentication keys and encryption algorithms are stored on the SIM together with subscriber information. Because the SIM is pluggable, it is easy to move the identity of the subscriber to another MS.

To summarise the logical architecture of GSM network, we have put together the figures from the previous sections in figure 2.3. The NSS dealing with telecom related functions such as establishing, switching, routing and terminating calls; the BSS handling mobility related functions such as locating phones and handling radio resources; and finally the MS allowing the user to communicate everywhere.

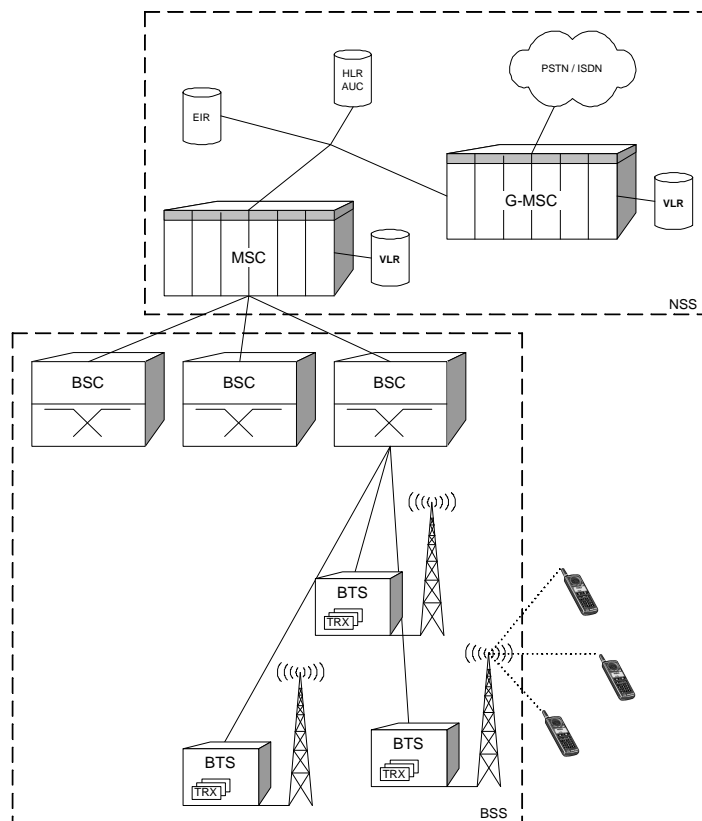


Figure 2.3: Logical architecture of a GSM network: In the top the NSS containing MSC, G-MSC, VLR, HLR, AUC, and EIR. Below the NSS is the BSS with its components: BSC and BTS. Outside the wired network is the MS.

2.3 Physical Architecture

In the previous section we discussed the logical architecture of the GSM network. We talked about the components of the network and their functionality. In this section we discuss the physical architecture of the GSM network. We also discuss what the previously described components are responsible for and what they know.

2.3.1 Physical layout

The GSM recommendations use terms describing the different levels of coverage (i.e. areas), entities are responsible for. In the following we discuss each of these levels.

The lowest level of coverage in the network is the *cell*. A cell is defined as the area covered by a single TRX on a BTS. The radius of a cell depends on the transmission power of the TRX but is typically somewhere between 1 and 30 kilometers. In low populated areas, the transmission power is highest and contains a single TRX. In urban areas, a BTS typically contains at least 3 TRXs — each controlling one sector antenna covering 120 degrees. In highly populated areas a single BTS can control up to 16 TRXs.

The next level of coverage is called a *Location Area (LA)*. A location area is a set of cells with a static border. A BSC typically controls several location areas. When the network needs to contact the MS (e.g. when it is called), all cells within the MS' LA is instructed to contact the MS; therefore the size of the LA is important in order to save signalling bandwidth. The size of the LA mostly depends on the mobility of the users in the area; if users movement is local, it is best to keep the LA large — otherwise it should be kept small.

The level of coverage under the control of a single MSC is called an *MSC Service Area*. It is a set of complete LAs, which means each LA is a part of just one MSC Service Area.

The area a network operator covers is called a *Public Land Mobile Network (PLMN) Service Area*. A network operator has exactly one PLMN Service Area.

The highest level of coverage within GSM is the *GSM Service Area*. This is the part of the earth covered by any GSM network operator.

2.3.2 Knowledge in the network

The logical architecture of a GSM network indicates a hierarchical order of the entities; the NSS controls the BSSs and the BSC controls BTSs. Within the NSS, however, there is no ordering of the entities. All MSCs are equal with respect to control.

In order to control their respective parts of the network, the entities need to know something about the network. In the following sections, the distribution of

knowledge in the network is revealed.

MS

The Mobile Station has no knowledge of any static part of the network. When turned on and authorised it is aware of the current LA and the cell it is in. It does not know anything about BSCs or MSCs.

BTS

The BTS acts as a bridge between the wired part of the network and the radio. It has been configured with some information about the cells it is serving. This information includes cell-id and radio frequencies. The BTS also contains a clock in order to synchronise MS communication.

BSC

The BSC is the lowest entity in the network capable of making decisions, such as when to make handover. It has knowledge about all the BTSs controlled by itself and their physical relations, i.e. neighbouring BTSs. The BSC also knows the neighbouring cells of its area in order to tell the MS which cells to measure radio quality on.

MSC

The MSC is the topmost entity in the GSM network and it has the largest amount of knowledge of the network — still it does not know the entire network. The MSC knows all the cells and BSCs within its service area and their connections. Given a cell-id, the MSC is able to locate the BSC in control of the queried cell if it is inside its service area. Besides the internal knowledge, the MSC also knows which MSC is controlling cells on the border of its service area. This information is needed to hand a call over to a cell on its border.

The physical architecture of a GSM network is separated into levels of coverage — each controlled by different entities. In order to control those levels, some knowledge of the network is necessary. Where this knowledge is located was also discussed.

2.4 Summary

In this chapter we first described the necessary functionality of a GSM network. Next we looked at the logical architecture of a GSM network, where we presented

the entities and their responsibilities. Finally we discussed the physical layout of the network and the network knowledge of the entities.

Chapter 3

GSM Network and Signalling

In this chapter we go into more details of the GSM networks and the signalling interfaces in the network. We give an introduction to some of the interfaces and various procedures in the network, especially the procedures concerned with the handover. We start by introducing the most relevant interfaces: A, Abis, and Air. Next we give an introduction to the procedures in the network that are essential for the mobility of the subscribers.

3.1 Interfaces

A lot of interfaces are introduced in the recommendations, but only a subset of these are relevant in our work. They are presented in a top-down fashion: A-interface, Abis-interface, and finally Air-interface. To give a quick overview of the relevant interfaces in a GSM network we have depicted them on figure 3.1.

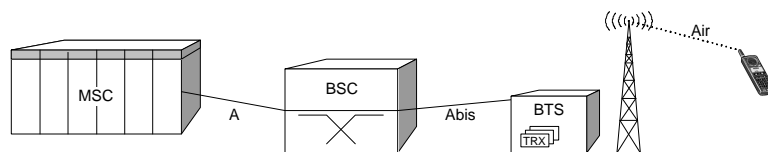


Figure 3.1: The interfaces in a GSM network, relevant to our work. The A-interface connects the MSC with the BSC, the Abis-interface connects the BSC with the BTS, and finally the Air-interface interface connecting the BTS with the MS.

All interfaces follow the Open System Interconnection (OSI) Reference Model [18], which divides the interface into layers to allow interconnection of the different interfaces and easy development of extensions to the specifications. For easy reference, the model is depicted on figure 3.2. All three interfaces utilize only the three lowest layers the OSI stack: physical, data-link, and network.

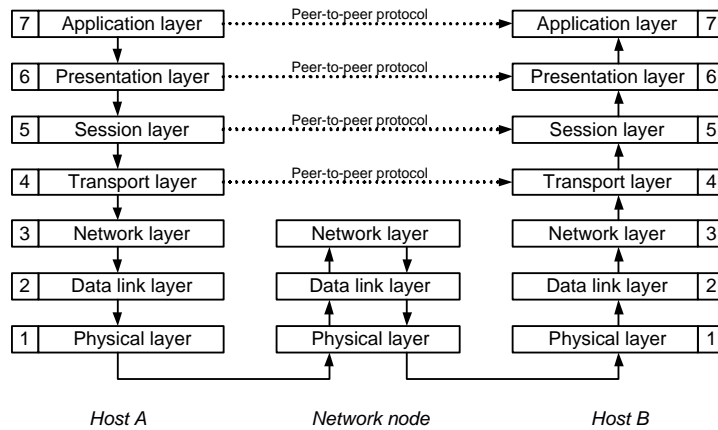


Figure 3.2: The OSI reference model

3.1.1 A-interface

The A-interface is the interface between the BSC and the MSC: It is built on an existing communication standard, *Signalling System 7* (SS7), which is used throughout the entire NSS. This standard is very common within tele communication. The reason for adopting such a standard is obvious: interoperability with existing telecommunication networks (PSTN, ISDN).

The SS7 network is huge and the complete description of it is out of scope for this thesis. The most important parts of the SS7 protocol stack, within the context of GSM, is illustrated on figure 3.3, where only the grayed parts are discussed here.

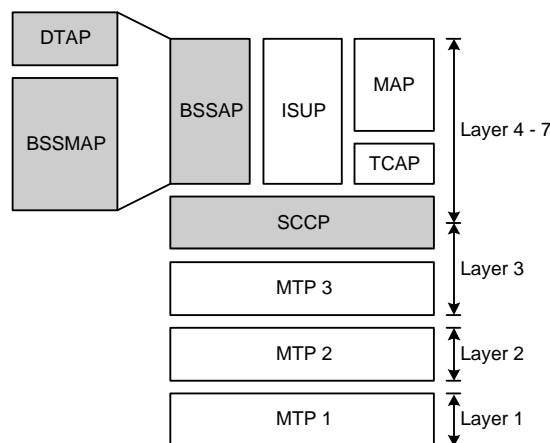


Figure 3.3: A subset of the protocol stack of the SS7 network. The grayed parts are discussed in this thesis. SCCP is part of both layer 3 and 4; BSSAP is separated into two sublayers: BSSMAP and DTAP

The lower levels of the SS7 protocol stack (OSI layer 1–3) are called the *Message Transfer Part* (MTP). The user part of the MTP contains several standards, but only one is interesting in this context, the *Signaling Connection Control Part* (SCCP). The SCCP is considered being the user part of the MTP, but it actually digs a little into layer 3.

The GSM specific signaling on the A-interface is performed by the *Base Station Subsystem Application Part* (BSSAP). This is separated into two layers: *Base Station Subsystem Management Application Part* (BSSMAP) and *Direct Transfer Application Part* (DTAP). The BSSMAP handles RR messages where DTAP handles MM and CC messages. While DTAP maps directly to MM and CC messages, BSSMAP does not map directly to RR: Some RR messages are exchanged exclusively between the MS and the BSS and some BSSMAP messages are exchanged exclusively between the BSS and the MSC. An illustration of this can be seen on figure 3.4.

Further details regarding the A-interface and the SS7 network can be found in [3], chapter 8–10.

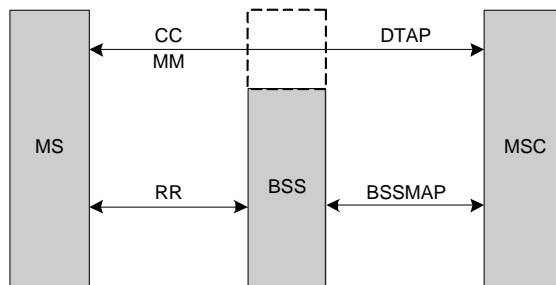


Figure 3.4: The BSSAP message relations to GSM signaling.

3.1.2 Abis-interface

The Abis-interface connects the BTSs with the BSC. The interface is part of the fixed network and communication is performed by conventional cables. The recommendations employ well known and well tested technologies on the fixed interfaces. Typically a PCM 30 (also is known as ISDN30) link is used; providing a bandwidth at 2 Mbit/sec. This allows up to 10 TRXs on the BTS, but in a typical setup a BTS has 1 to 4 TRXs. When using two ISDN30 links, a maximum of 16 TRXs can be installed on a single BTS.

The Abis-interface has never been very well specified. This has lead to the current market situation, where the BTS and the BSC always comes from the same vendor since other combinations would lead to incompatibilities.

Layer 1 of the Abis-interface is the D-channel of the ISDN30 links. Each ISDN30 link contains 30 B-channels for traffic (each giving 64 kbit/sec.) and one

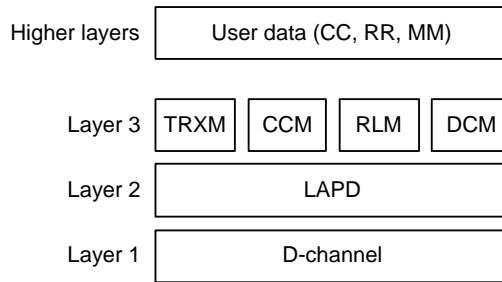


Figure 3.5: The protocol stack of the Abis-interface

D-channel for signalling.

Layer 2 of the ISDN D-channel uses the LAPD protocol for signalling. This is adopted for signalling on the Abis-interface.

Layer 3 is split into four parallel sublayers: *TRX Management* (TRXM), *Common Channel Management* (CCM), *Radio Link Management* (RLM), and *Dedicated Channel Management* (DCM). The TRXM sublayer is used for taking TRXs into and out of service, and controlling their status. CCM is used for broadcast messages for the entire cell, e.g. paging of an MS (the network tries to contact the MS, when it is called or an SMS is received), SMS broadcast, and information about the cell. RLM is for controlling layer 2 of the radio link between the MS and the BTS. This includes establishing and releasing connections. DCM is used for controlling layer 1 of the Air-interface such as handovers, measurements, channel activation/deactivation, and encryption setup. RLM and DCM are only used for active links on the Air-interface, i.e. there is no communication on them in idle mode. On figure 3.5 the protocol stack of the Abis-interface is shown.

On top of layer 3, the payload data is transported. The Abis-interface is mostly used for exchange of RR, CC, and MM messages described in the Air-section (3.1.3). The Abis-interface is covered in greater detail in [3], chapter 6.

3.1.3 Air-interface

The Air-interface is the radio interface between the MS and the fixed network. This interface has a lot of difficulties compared to the other interfaces, because radio communication is far more sensitive to external interference than cabled communication. To compensate for the hostile environment, a great deal of bandwidth is spent on error correction data. This and the age of the technology sets the limitation on the bandwidth of the Air-interface to 9,600 bits/sec. for data communication.

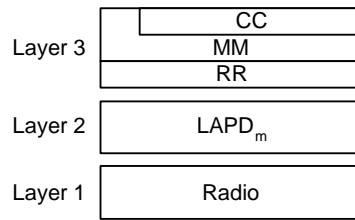


Figure 3.6: The protocol stack of the Air-interface. Users of layer 3 has access to all of CC and limited parts of MM, but RR is not directly accessible.

Layer 1 is concerned with various divisioning schemes and modulation techniques employed to allow multiple access and ensure data quality of the radio. The physical layer is described in section 7.1–7.4 of [3], and will not be covered further in this thesis.

Layer 2 controls the transmission and has knowledge about the layout of the various logical channels on top of the physical channels. The data-link layer offers both unacknowledged and acknowledged data transfer as well as mechanisms to prioritise the data transfer. The protocol for signaling on this layer is the *LAPD_m* — a modified version of *Link Access Protocol for the D-channel* (LAPD) used in for example ISDN networks [13]. The modification takes into account the limited resources on the radio interface; all the dispensable parts of LAPD are therefore removed, resulting in a light version of LAPD.

Layer 3 is divided into three sublayers, each concerned with different tasks in the network. The sublayers are *Radio Resource* (RR), *Mobility Management* (MM), and *Call Connection Management* (CC). The task of the RR sublayer is to ensure that the upper sublayers, i.e. MM and CC, are able to transmit transparently of the radio path used. The tasks are channel setup and release, handover, and various radio related procedures when there are active channels. The MM sublayer handles the procedures ensuring the reachability while being mobile, authentication of the subscriber towards the network as well as initialization of chipering (encryption) before call setup. The CC sublayer is responsible for setup and release of calls, and various things happening during the call. RR, MM, and CC are sublayers and not three individual protocols implementing network services. RR offers reliable radio services to MM and CC by taking care of the low level radio layers. On figure 3.6 the protocol stack of the Air-interface is shown. For further information regarding the Air-interface, please consult chapter 7 of [3] and [6].

We have now presented the interfaces connecting the devices in a GSM network. We also presented the layers and the tasks each of them are responsible for.

3.2 Procedures in GSM

A lot of the challenges in the design of mobile networks are concerned with the mobility of the MSs. In this section we give an informal description of some of the most important procedures concerning mobility in the network. The procedures are presented in the following order: Power ON, IMSI Attach and Detach, Location Update, and Handover. The presentation of handovers includes more details than the rest of the procedures.

3.2.1 Power ON

When an MS is turned on it will try to connect to the network. This procedure contains several substeps. The first substep is a frequency scan within the GSM allocated frequency spectrum. This is done to achieve knowledge of the surrounding cells. From the prioritized list of allowed PLMNs in the SIM, the MS selects the one with the highest priority available. If the home network is available it is selected; otherwise the MS select a foreign PLMN. This is called roaming and it requires an agreement between the foreign PLMN and the home PLMN. After the PLMN selection the MS performs a Location Update (LU) as described in section 3.2.3. After the LU the MS is attached to the network and operational. The Power On procedure is described on page 157 in [1], section 5.1 in [5], and section 4.1 in [10].

3.2.2 IMSI Detach and IMSI Attach

When the MS is turned off the *IMSI (International Mobile Subscriber Identity) Detach* procedure might be executed. From the MS point of view, the IMSI Detach procedure is performed by sending an unacknowledged IMSI Detach message to the BSS. If the message is received, the MS is marked unreachable in the HLR. When the MS is called there is no need to contact the last known BTS (from the LA knowledge in the HLR) just to find out that the MS is unreachable. The IMSI Detach procedure can also be performed implicit by the network if the MS fails a periodic location update (discussed in section 3.2.3). It is not mandatory to use the IMSI Detach procedure but most operators choose to do so. The procedure can also be executed in other situations than power off. A hybrid telephone might use this procedure to leave the GSM network when entering another kind of network.

The *IMSI Attach* procedure is used to inform the network, that the MS is available again. The actual signaling is just a location update and it is usually not considered a real procedure. The IMSI Attach is described in [1], page 192; IMSI Detach on page 195. Further informations can be found in [3], page 357 and in [10, 11].

3.2.3 Location Update

The mobility of the MSs requires procedures to monitor and maintain the current location, in order to route incoming calls to the MS. The procedure to ensure this is *Location Update* (LU). The LU is performed in different situations: one when the MS attaches to the network; another when the MS moves from one LA to another. If the MS just changes cell within the same LA no LU is performed. In most networks, the MS has to perform LU periodically. If such a periodic LU fails, the network will register the MS as unreachable. This is called implicit IMSI Detach as discussed in section 3.2.2.

The Location Update procedure is discussed in a greater detail in [1], page 193–194 and in [3], section 12.1 (scenarios).

3.2.4 Handover

The *Handover* procedure is probably the most important procedure to ensure the mobility of the MS during calls. The purpose of the procedure is to preserve ongoing calls, when moving from one cell to another. The presence of an ongoing call gives rise to time criticality of the processing.

The decision whether to perform the handover, is made by the serving BSC, which has no direct knowledge of the radio quality. In order to decide whether to initiate a handover, the BSC receives information about the radio link quality from the BTS and the MS. During a call, the MS periodically sends *measurement results* to the BTS. The measurement results contain measurements of the radio signal quality of the downlink (from the BTS to the MS) of the call and up to five neighbouring cells. The serving BTS measures the uplink (from the MS to the BTS) radio signal quality of the call and forwards the measurement result from the MS, together with its own measurements, to the BSC in a *measurement report*. From the information in the measurement reports, the BSC is able to decide whether a handover to another cell is needed. The algorithm to decide whether to perform a handover or not is not specified in the recommendations — it is considered to be operator dependant. This algorithm is not investigated in our work, because our work start when the decision has been made.

There are different kinds of handovers, which involves different parts of the network. Changing cells within the same BTS is not as complex as changing cells belonging to different MSCs. In the following sections the different kinds of handover are discussed. They are listed in increasing complexity: Intra-cell/BTS, Intra-BSC, Intra-MSC and finally Inter-MSC.

The various handovers are discussed in section 3.4.4–3.4.5 of [10], section 3.1.5–3.1.7 of [9], [8], and section 12.5 of [3].

Intra-cell/BTS

The terms intra-cell and intra-BTS handover are both used in the literature to describe the same situation: a frequency change. Because a BTS can control several cells the the intra-BTS handover is a little more advanced (logically) than the intra-cell handover. The reason they are considered the same is, that frequency reuse within the same BTS is impossible.

The intra-cell handover is actually not a “real” handover because its only purpose is to change the frequency of an ongoing call. The frequency change is performed when the quality of the link is degrading and the measurements on the neighboring cells shows nothing better. In this case the BSC, which controls the BTS serving the MS, orders the MS and BTS to retune to another frequency, which hopefully offers better quality to the link. The link degradation is caused by interference with other calls in nearby cells using the same frequencies, and therefore the solution is to try another channel.

The intra-BTS handover is simpler than the rest of the handovers, because the cells involved are *synchronized*, i.e. the MS knows when to communicate with the new cell. This saves a great deal of signalling during the handover and is therefore faster the more complex non-synchronized handovers.

Intra-BSC

The intra-BSC handover is performed when the target cell is controlled by a BTS different from the source cell and both BTSs are controlled by the same BSC. The MSC is not involved in the handover, but is notified when the handover has taken place. If the target cell is located in another LA, the MS needs to perform the location update procedure after finishing the call; otherwise the MS is unreachable by the network. Figure 3.7 illustrates the situation.

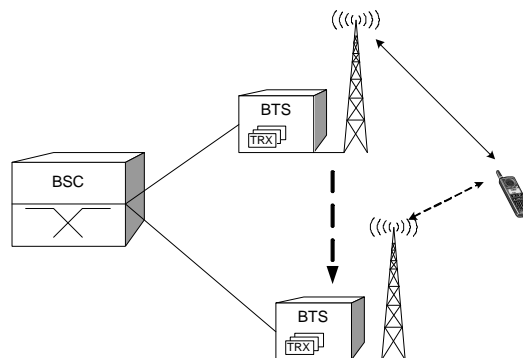


Figure 3.7: Intra-BSC handover.

Intra-MS

When the BSC decides that a handover is required, but the target cell is not controlled by itself, it needs assistance from the connected MSC. The result could be either an Intra-MS or Inter-MS handover.

In the Intra-MS handover case, the target cell is located in another BSS controlled by the same MSC. When contacted by the source BSS, the MSC contacts the target BSS for allocation of required resources and informs the source BSS when ready. After a successful resource allocation, the MS is instructed to access the new channel and the call is switched to the new BSS. This is illustrated in figure 3.8.

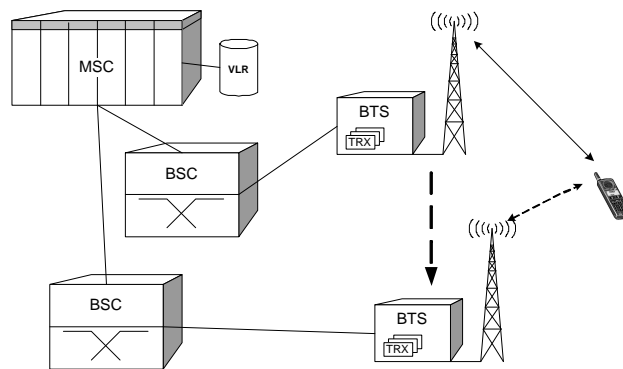


Figure 3.8: Intra-MS handover.

Inter-MS

The inter-MS handover procedure is performed, when the target cell is connected to another MSC (MSC-B) than the one currently serving the call (MSC-A). MSC-A contacts MSC-B with a handover request, from which MSC-B allocates resources for as in the intra-MS case. When the resources are allocated within MSC-B and its BSS, the call is switched as in the intra-MS case. Even though MSC-B has received the call, MSC-A remains in control of the call for the rest of its duration — even if a subsequent handover is performed. The reason for this is, that MSC-A has all informations about the subscriber in its VLR. The information is only moved when an LU is performed. Because of this, an LU is required at the end of the call, when an inter-MS handover has been performed.

The inter-MS handover is depicted in figure 3.9

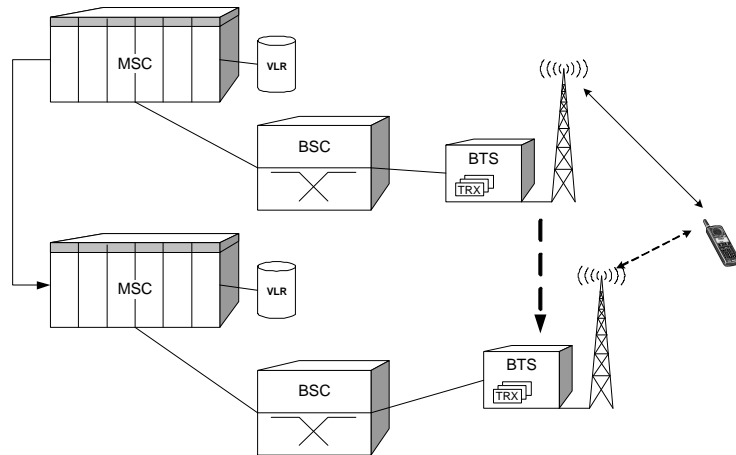


Figure 3.9: Inter-MSC handover.

3.3 Summary

In this chapter we first described the interfaces and their layers. Most important are layer 3 of the interfaces, because our work is focused on this layer. Next we presented the most important procedures to ensure mobility in GSM networks. Most important are the handovers, but without the rest, the overall functionality of the network would decrease to an unacceptable level.

Chapter 4

Problem Domain

In this chapter we discuss the problem domain of our work, the GSM intra-MSC handover. First we go into a detailed discussion of the intra-MSC handover, based on our interpretation of the GSM recommendations [4] and Heine [3]. Next we specify our limitations of the problem domain. Finally we present our model design and abstractions.

4.1 Details of the intra-MSC handover

In section 3.2.4 we described the handover procedures in GSM networks. In our work we have focused on the intra-MSC handover. In this section we discuss this procedure in a detailed manner which includes a description of the messages sent between the involved entities. First we look at the successful handover, then we describe the cases involved with handling failures in the network.

Almost all of our work is concentrated on layer 3 of the involved interfaces; all communication shown in the description of the handover is layer 3 messages, except two layer 2 messages that are important for the overall functionality of the handover.

In order to avoid errors as a result of lost messages, messages are retransmitted until a response is received or a timer times out. In our description these are filtered out to make it easier to read. For the full description, please see [9] section 3.1.5, [8], [10] section 3.4.4, and [3] section 12.5.

4.1.1 The successful case

The intra-MSC handover happens when the cell to be handed over to is controlled by the same MSC, but another BSC than the one currently serving the call.

As previously described, all handovers are initiated by the serving BSC; when receiving a MEAS_RES (MEASurement RESult) message indicating a handover

is required, the BSC sends a HND_RQD (HaNDover ReQuireD) message to the MSC.

The intra-MS-C handover can be split into four phases: Decision of handover, channel allocation, handover execution, and resource deallocation. We discuss each of the four phases in the following section. After the individual discussion of the phases, we put them together in order to see the successful intra-MS-C handover as a complete process.

The figures 4.1–4.4 illustrate the four phases of a successful intra-MS-C handover. Each step on the figures is numbered in order to see their position in the complete handover on figure 4.5. The numbering is taken from the complete handover, why the first step of figure 4.2–4.4 is numbered different from 1.

Decision of handover

The decision of whether to perform a handover is made by the serving BSC from the MEAS_RES (MEASurement RESult) messages received from the BTS. When the decision is made, the handover is initiated by sending the HND_RQD (HaNDover ReQuireD) message to the MSC. Figure 4.1 shows the messages involved with the decision phase of the handover. The MEAS_RES messages are sent periodically during a call, but we only look at a single one of them.

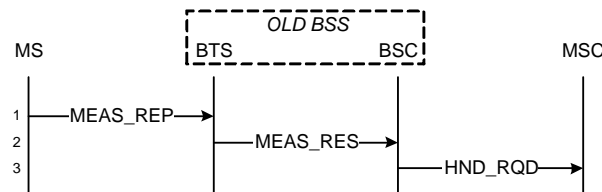


Figure 4.1: Message exchange for deciding to perform an intra-MS-C handover.

Channel allocation

The channel allocation phase of the handover is concerned with allocation of radio resources in the new BSS. The MSC requests the new BSS to allocate a channel for the call with the HND_REQ (HaNDover REQuest) message, which is acknowledged with the HND_REQ_ACK (HaNDover REQuest ACKnowledgement) when the channel has been activated. The actual allocation of the channel is done by the message CHAN_ACT (CHANnel ACTivation) and its acknowledgement CHAN_ACT_ACK (CHANnel ACTivation ACKnowledgement). The messages on figure 4.2 are involved with the channel allocation phase.

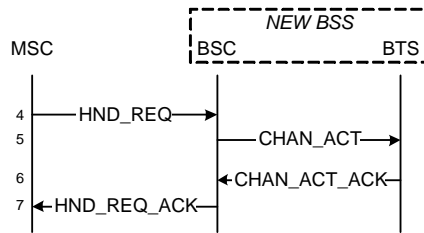


Figure 4.2: The channel allocation phase of an intra-MS handover.

Handover execution

When the resources for the call have been allocated in the new BSS, the MS is instructed to access the new radio channel. The new BSS generates the HND_CMD (HaNDover CoMmanD) message, which contains information about the new radio channel. This message is forwarded through the old BSS to the MS (message 8–10). After reception of the HND_CMD, the MS tries to access the new channel with a HND_ACC (HaNDover ACCess) message while it is listening for a PHYS_INFO (PHYSical INFOrmation) from the new BSS, containing synchronization information for the MS. The HND_ACC message is a special message, a so called *access burst*, because no signalling channel exists. A signalling channel has been set up, when the PHYS_INFO is received by the MS. In order to set up acknowledged communication between the MS and the new BSS, the MS sends a SABM (Set Asynchronous Balance Mode) message, which is a layer 2 message. This is acknowledged by a UA (Unnumbered Acknowledgement) which also is a layer 2 message. The reason for bringing the two layer 2 messages into the discussion is, that a timer depends on them (discussed in section 4.1.3). When acknowledged mode is up, the BSC is notified with an EST_IND (ESTablish INDication).

When the MS has received the UA message, it informs the network, that the handover is completed. This is done by the HND_COM (HaNDover CoMplete) message sent to the new BTS, which forwards it to the BSC that ends the handover with the HND_CMP (HaNDover CoMpleted) message for the MSC. At this point of time, the call is switched through the new BSS.

The handover execution phase is illustrated on figure 4.3 as an Message Sequence Chart.

Resource deallocation

When the call has been switched to the new BSS, the actual handover is completed, but radio resources are still occupied in the old BSS. The MSC sends a CLR_CMD (CLear CoMmanD) to the old BSC, which orders the old BTS to release the radio resources allocated for the call with the RF_CHAN_REL (Radio Frequency CHANnel RELEase). The old BSC acknowledges the CLR_CMD

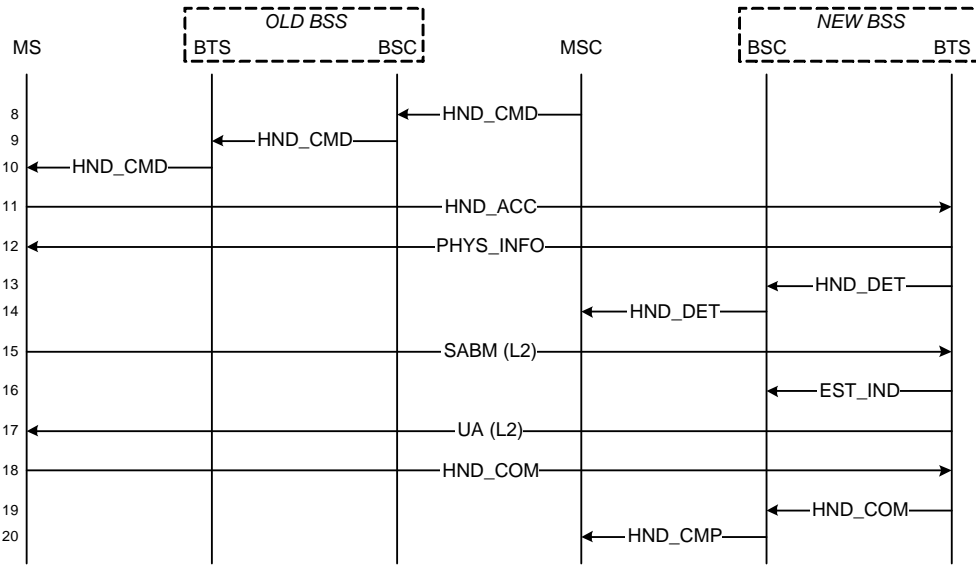


Figure 4.3: Messages exchanged during the handover execution phase of an intra-MSC handover.

before it receives the acknowledgement from the BTS, confirming that the resources are released. This is because the BSC is in complete control over the BTS and that it knows resources are released eventually. If a failure occurs within the BSS, the resources are marked unuseable. The messages involved with the resource deallocation phase are illustrated on figure 4.4.

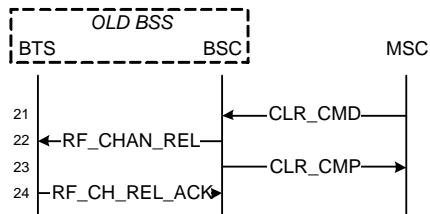


Figure 4.4: Messages exchanged during the resource deallocation phase of an intra-MSC handover.

The complete intra-MSC handover

The last four sections described the phases of a successful intra-MSC handover. In this section we put the phases together in order to show the complete handover. Figure 4.5 shows all messages involved with the successful intra-MSC handover.

The gray boxes on figure 4.5 with the text saying “Call switched...” are the two places where the call can be switched to the new BSS. The first is called *early switching* since it allows the MSC to switch the call after reception of the HND_DET

message. Early call switching is an optional feature of a GSM network, but most modern networks use it. It shortens the *fall out time* (the time from the traffic stops on the old BSS to it resumes on the new BSS) during a handover. *Late switching* is the standard point of time, where the call is switched.

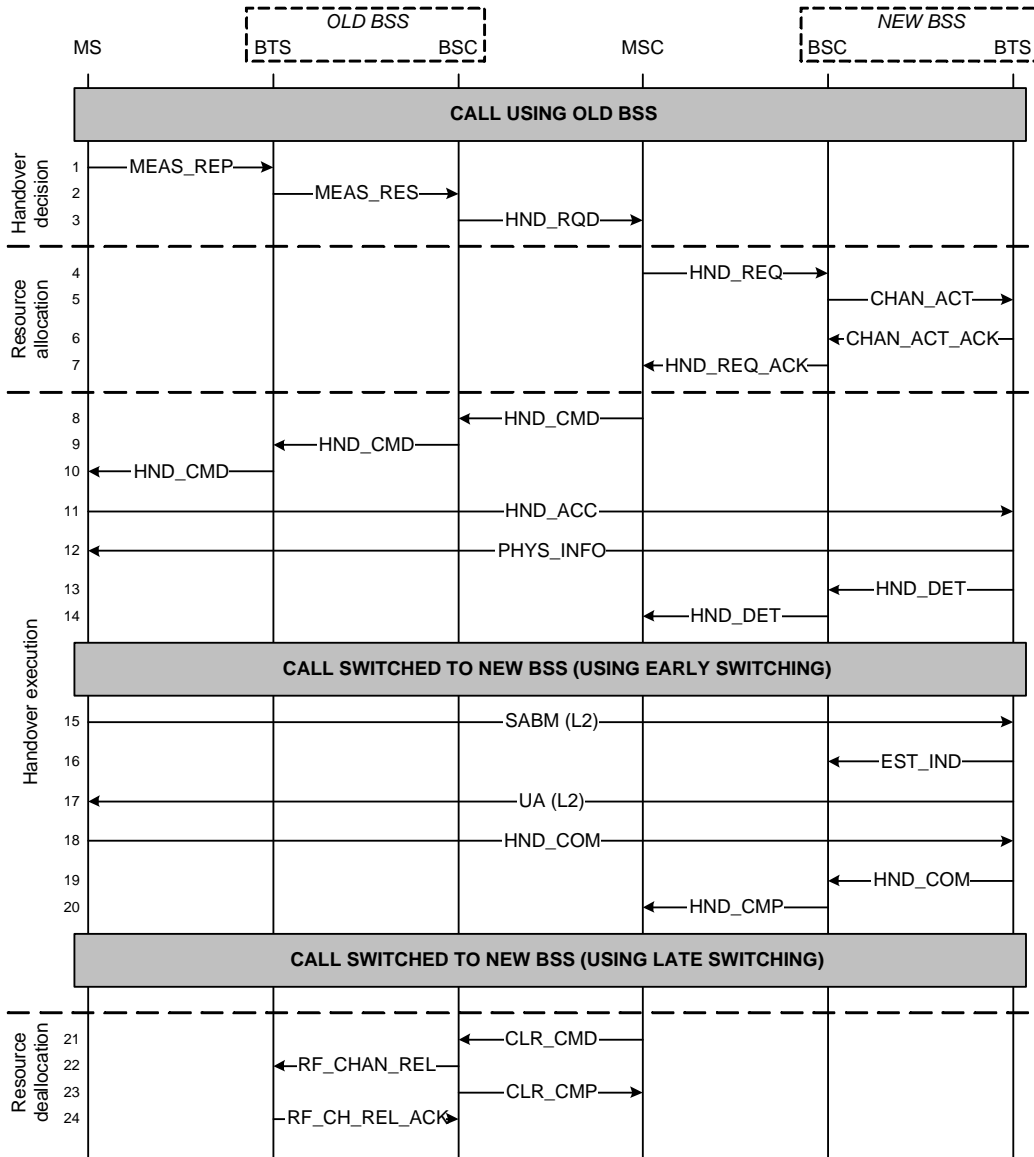


Figure 4.5: The complete successful intra-MSC handover.

4.1.2 Failure cases

A GSM network is a distributed system, since logic is placed at several nodes in the network. This gives rise to a number of distributed error cases, e.g. a device

crashing or a failing communication link. In this section we look at the major failure situations during an intra-MS handover.

Call lost

During, or even before a handover the call could be lost because of the radio link quality decreases to a level, where it is impossible to maintain the link. There is not much the network can do in this situation, but it is important that all allocated resources are released. The call could be lost after resources are allocated in the new BSS and before the call has been switched. In this case, resources should be deallocated in both the new and the old BSS. If the call ends by one party hanging up during a handover, the same resource deallocation must take place.

MS fails to access the new BSS

When the HND_CMD has been sent to the MS, it tries to access the new BSS. The MS is not synchronized with the new BSS and is therefore not aware of when the BSS is listening for its HND_ACC message. This synchronization phase could fail, in which case the MS goes back to the old BSS and indicates the failure.

Incompatible equipment

For a handover to succeed the new BSS must support the same features as the old BSS utilizes for the call. If the new BSS for example not supports the used ciphering algorithm, it will not be able to continue the call. In such a situation, the new BSS indicates the failure to the MSC.

4.1.3 Timers

Within GSM most of the distributed error cases are handled with timers. In this section we discuss the timers involved with the intra-MS handover.

T7 The T7 timer is located in the old BSS and is started when the HND_RQD message towards the MSC (step 3 on figure 4.5) has been sent. When T7 times out, the HND_RQD message is retransmitted. The recommendations does not say anything about a maximum number of times this can happen. Four situations exists where the T7 timer is stopped:

- The BSC receives a HND_CMD from the MSC.
- A RESET message is received from the MSC, indicating a fatal error within the communication data on the link. The RESET message resets the A interface between the MSC and the BSC.

- The radio link quality towards the MS improves so a handover is not required anymore.
- The call is terminated — either because of one of the parties ends the call or because the radio link is lost.

The recommendations have no description of the case where no response is received from the MSC or where the handover resource allocation fails. The HND_RQD_REJ (HaNDOver ReQuireD REJected) message is an optional message sent from the MSC to the old BSS telling it that the new BSS was unable to allocate the required resources.

We have decided not to include the T7 timer in our work because its only purpose is to repeat a message transmission; we do not consider resending messages, as stated earlier.

The T7 timer is described in [9], section 3.1.5.1.1.

T8 The T8 timer is also placed in the old BSC. The timer is started when the HND_CMD is sent to the MS (at step 9 of figure 4.5). The reason for having the timer, is to keep radio resources on the old BSS long enough, to let the MS fall back if needed. There are two ways to stop the T8 timer:

- The BSC receives a CLR_CMD from the MSC, which tells the BSC that its part of the call is completed and that its resources are to be deallocated.
- The BSC receives a HND_FAI (HaNDOver FAILED) from the MS, telling the BSC that the MS could not reach the new BSS, and that it has fallen back to the old BSS.

If T8 times out, the BSC releases all resources allocated for the call and sends a CLR_REQ message to the MSC. The MSC replies with a CLR_CMD which is also sent to the new BSS. The result of T8 timing out is that all radio resources allocated to the call are released. This means that the call is lost. Figure 4.5 illustrates the effect of T8 timing out.

The T8 timer is described in [9], section 3.1.5.3.1 and 3.1.5.3.3.

T3103 The T3103 is imprecisely described in the recommendations. It is located within the network, but its precise location is not stated. From its message involvement, we have deduced that it must be somewhere in the old BSS.

The timer is started by the sending of a HND_CMD to the MS, which is done by the old BSS. There are two cases where the T3103 timer is stopped:

- The network (i.e. the new BSS) receives the HND_COM; this is step 18 on figure 4.5. This message does not end up where the timer is, so our interpretation is that the meaning is CLR_CMD, which is the indication to the old BSS, that the handover has completed successfully.

T3124 The timer T3124 is located in the MS. T3124 is started when the MS sends the first HND_ACC to the new BSS; it is stopped when the MS receives a PHYS_INFO from the new BSS. If T3124 times out, the MS tries to switch back to the old BSS, sends a HND_FAI message, and resumes normal operation as if no handover has happened. This is of course only possible if the old radio link is good enough for communication; otherwise the call is lost.

The T3124 timer is described in [10] section 3.4.4.2.2 and 3.4.4.4.

T101 The T101 timer is located in the MSC and protects the MSC from waiting infinitely for the new BSS to allocate resources for the handover. The timer is started when the MSC sends the HND_REQ to the new BSS (step 4 on figure 4.5). The T101 timer is stopped in two situations:

- A HND_REQ_ACK is received from the new BSS (step 7 on figure 4.5).
- A HND_FAIL (HaNDover FAILED on the A interface) message is received from the new BSS. This could happen if the new BSS is unable to allocate the requested resources or if the current ciphering algorithm is unsupported by the new BSS. Further details regarding the failure conditions in the resource allocation phase can be found in [9], section 3.1.5.2.2.

If T101 times out or if the HND_FAIL is received, the resources in the new BSS are deallocated by sending a CLR_CMD; the call is supposed to continue on the old BSS. The recommendations allow the MSC to send a HND_RQD_REJ (HaNDover ReQuireD REJect) to the old BSS in order to inform that it was impossible to allocate resources for the handover. This message is not required because the old BSS is protected by the T7 timer. It is considered a gesture to inform of the failure.

The T101 timer is not described in prose in the recommendations, but it can be found on the SDLs describing the MSC during a handover: [8], figure 13, sheet 2.

T102 The T102 timer is also located in the MSC guarding it from waiting infinitely for the actual handover to complete or fail. The timer is started when the HND_CMD message is sent to the old BSS (step 8 on figure 4.5). The T102 timer is stopped in three situations:

- The MSC receives a HND_CMP message from the new BSS indicating that the handover has completed successfully (step 20 on figure 4.5).
- The MSC receives a HND_FAIL from the old BSS meaning the MS was unable to reach the new BSS.
- The MSC receives a CLR_REQ from the old BSS because the call was lost.

If T102 times out, the resources allocated for the call in both the old and the new BSS are released and the call is lost. If the CLR_REQ is received from the new BSS, the same outcome is specified, which is the expected behavior because the call already has been lost. In the case of the HND_FAIL message the call is resumed on the old BSS if possible.

The T102 timer is also not described in prose in the recommendations; it can be found on the SDLs describing the MSC during a handover in [8], figure 13, sheet 2–4.

Throughout section 4.1 we have discussed message exchange of the intra-MSC handover in a very detailed manner. We have also discussed the failure situations in the network as a result of the distributed layout. Finally we have seen how timers are able to discover and handle the distributed failures.

4.2 Interpretation of the problem domain

The GSM recommendations are from time to time unclear and related information is often placed in different parts. In order to model the intra-MSC handover we need a clear specification; this section is considered to be this specification. Our starting point with this specification is the SDLs in [8] figure 13, sheet 1–4, which can be found in appendix B for easy reference. They specify the behavior of the MSC during an intra-MSC handover. The SDLs are imprecise and ambiguous, which is why we made our own: figure 4.7–4.10. Certain limitations and changes exist in our SDLs compared to the originals: The naming is changed to conform with our conventions; several selections on our SDLs are only allowed to give one outcome, where the originals contain more ways; and cases irrelevant to our work is removed. However the structure of our SDLs is identical to the originals. In the following sections we discuss our work with the SDLs and finally conclude that results obtained from our SDLs are valid.

We have included a short introduction to SDL in appendix A.

4.2.1 Discussion of the SDLs

In the following section we discuss our SDLs in order to describe what parts of the intra-MSC handover we are concerned with. Most of the steps in the SDLs are already discussed in the previous sections — they will not be discussed again. This section gives an overview and illustrates some of our abstractions. The flow of the SDLs present in this section, is always top–down.

Sheet 1 — figure B.1 The starting condition of the handover procedure is an ongoing call handled by **Old BSS**, which is going to be handed over to **New BSS**; the initial state of figure 4.7 assumes the ongoing call on **Old BSS**. After

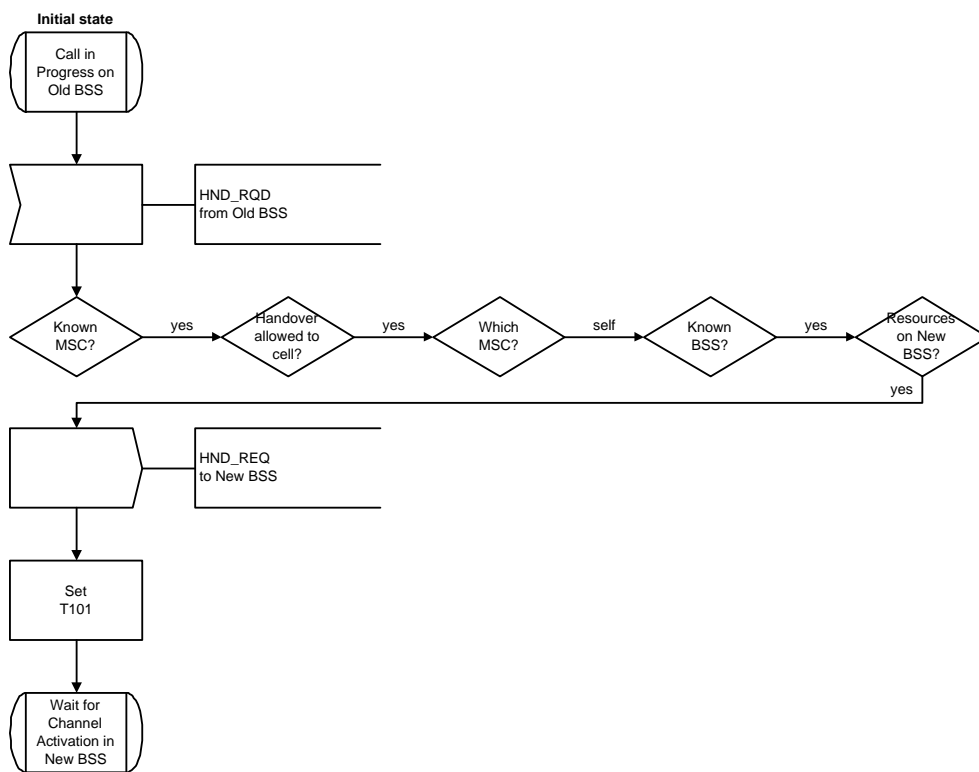


Figure 4.7: SDL describing the behavior of the MSC during an intra-MSC handover — part 1 of 4.

the reception of the HND_RQD message, a number of selections are passed. As illustrated, they all return **yes** or **self**; the rest of the possible outcomes are removed. We keep the selections in order to show what we assume the network is capable of. The rest of figure 4.7 is already explained.

We have made two limitations on our version of sheet 1 (figure 4.7). The first is leaving out the case called **Call Release** and the second is leaving out other outcomes of the selections. The **Call Release** case is where the call ends before a handover is initiated. The cause is either one of the parties hanging up or loss of the radio path. In both cases the handover has not been initiated and no signaling regarding the handover has occurred. In other words, no handover. This case is therefore not considered a part of our work. The reason for leaving out most of the outcomes of the selections are, that the left out outcomes contradict our assumptions of the network, which are: The handover is intra-MS, so the MSC in control of the call, also controls the new BSS; a handover to the new cell is allowed; the MSC knows the BSS to be handed over to (obviously because it is in control of it); and that resources are available in the new BSS.

The lower left corner of sheet 1 (figure B.1) starting with reference point 3, has in our version been moved to the next part, figure 4.8.

Sheet 2 — figure B.2 The SDL on figure 4.8 starts in the state where the MSC is waiting for **New BSS** to allocate resource for the handover. Three outcomes are possible in this state: the successful case (left flow), fall back to **Old BSS** (middle flow), and too late — call lost (right flow). The successful case contains some functions not previously mentioned: **Queue Messages for MS** and **Set Up Handover Device**. The **Queue Messages for MS** is a function for queueing message while no signalling link is present to the MS. This part is not considered relevant to our work, but is obviously necessary in a real system. The **Set Up Handover Device** function is an internal function within the MSC for switching the call to **New BSS**. We assume that the MSC is able to perform call switching, which is why we are not concerned with the the **Handover Device** and thereby the **Set Up Handover Device** function. The successful case continues on figure 4.8 through the state **Wait for access by MS on New BSS**. The Fall Back case is described earlier except for the two selections: **Retry Handover Attempt** and **Send Reject**; both only able to return **no** and both included for showing our assumptions or limitations: We do not try to do a second handover if the first fails and we do not support the optional HND_RQD_REJ message for **Old BSS**, indicating that the requested resources could not be allocated on **New BSS**. The Too Late case is initiated by **Old BSS** sending a CLR_REQ (CLear REQuest) because the call has been lost. The first step hereafter is informing the network, and thereby the other party of the call, that it has been lost. After this, the allocated resources in **Old BSS** and **New BSS** can be released. **Old BSS** is allowed to start deallocating its resources at the same time as the CLR_REQ has been sent to the MSC. The

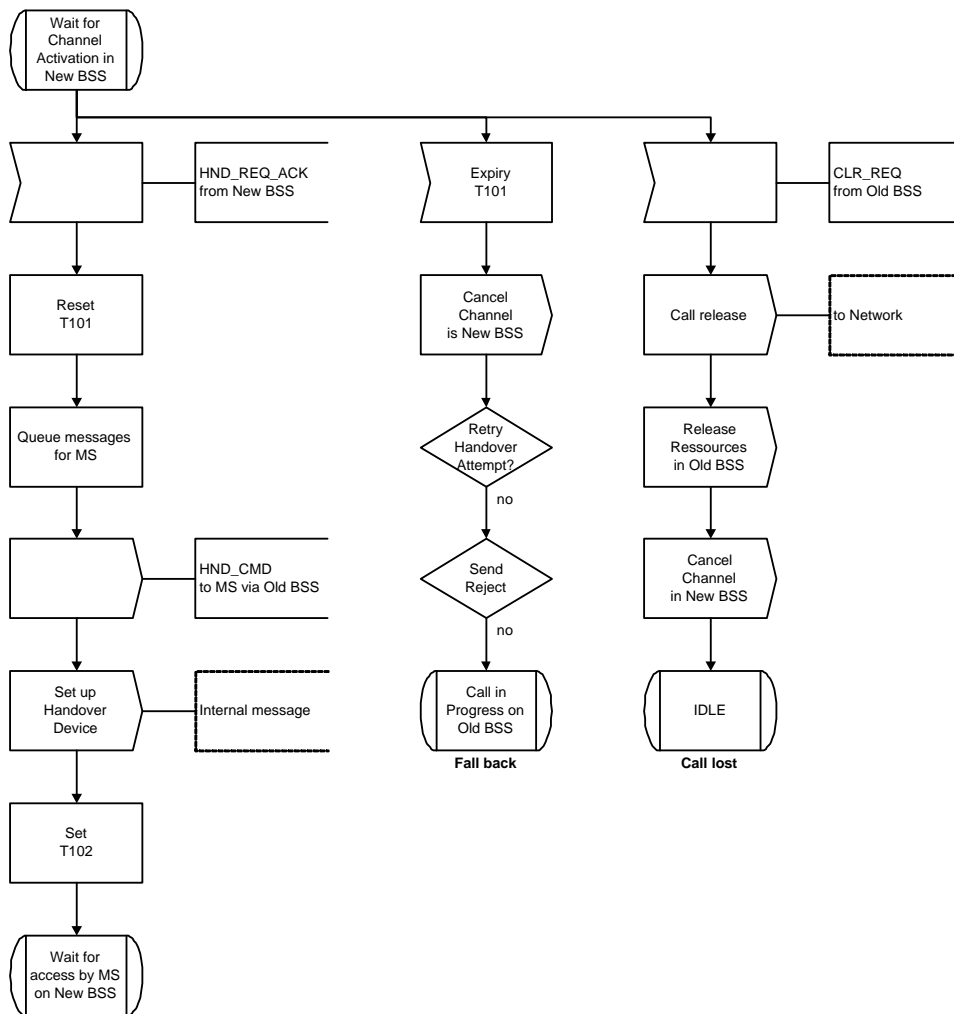


Figure 4.8: SDL describing the behavior of the MSC during an intra-MSC handover — part 2 of 4.

MSC sends a CLR_CMD to both of the BSSs, which performs the deallocation of the resources.

The parts of sheet 2 being left out are the Call Release case and the A-HANDOVER FAILURE from BSS-B case from Wait for Channel Activation Intra-MS, and the yes outcome of the selection Retry Handover Attempt. The Call Release case is the case where one of the parties of the call hangs up. The outcome is releasing all resources in both the old BSS and the new BSS. This is also the case, if the call ends because of losing the radio path. We chose to collapse the two cases, because their difference lies on the CC sublayer of the Air-interface, which is above the level we are working with (we are not working with higher layers than RR on the Air-interface). The reason for leaving out the HND_FAIL from New BSS is that the resources needed in the new BSS are assumed to be available and the two BSSs are assumed to support all the features utilized for the call. These assumption ensures that this case never occurs. The last part of sheet 2 being left out is the Retry Handover; we do not perform a secondary handover if the first fails. Real system do try again if possible, but since our interest is the result of the handovers, the case where the handover has failed is as interesting as the case of a successful handover.

Sheet 3 Figure 4.9 shows the first three outcomes of the state Wait for access by MS on New BSS. The first two flows (left and middle) shows the successful cases — first the HND_DET and the early switching (left) and next the HND_CMP finalising the handover and late switching (middle). The function Forward queued messages for MS via New BSS is the place where the messages queued for the MS while no signalling link was available, is sent to the MS via New BSS. The last flow (right) shows the case where the MS is unable to access New BSS and therefore falls back to Old BSS and informs the network, that the handover failed. This flow resumes communication through Old BSS and deallocates all resources allocated for the handover — both in New BSS and internally (Handover Device).

Sheet 4 The last SDL (figure 4.10) shows the remaining outcomes of the Wait for access by MS on New BSS state. The first outcome (left flow) shows that New BSS is unable to continue the handover and therefore sends a CLR_REQ. In this case the MS might be able to fall back to Old BSS. The two remaining outcomes are Old BSS loses the connection because of a timeout of T3103 (middle) and timeout of T102 described earlier. The selection Wait for MS in New BSS can only choose the no way. This choice is based on an ambiguity in the recommendations. Timeout of T3103 should release the call but the SDL allows the new BSS to wait for the MS to access it. We chose to follow the T3103.

One case has been left out from sheet 4, the Call Release from network, meaning the call has ended because of the other party (not the MS) has hung up. This informations is queued for the MS, because no signalling link is present at this

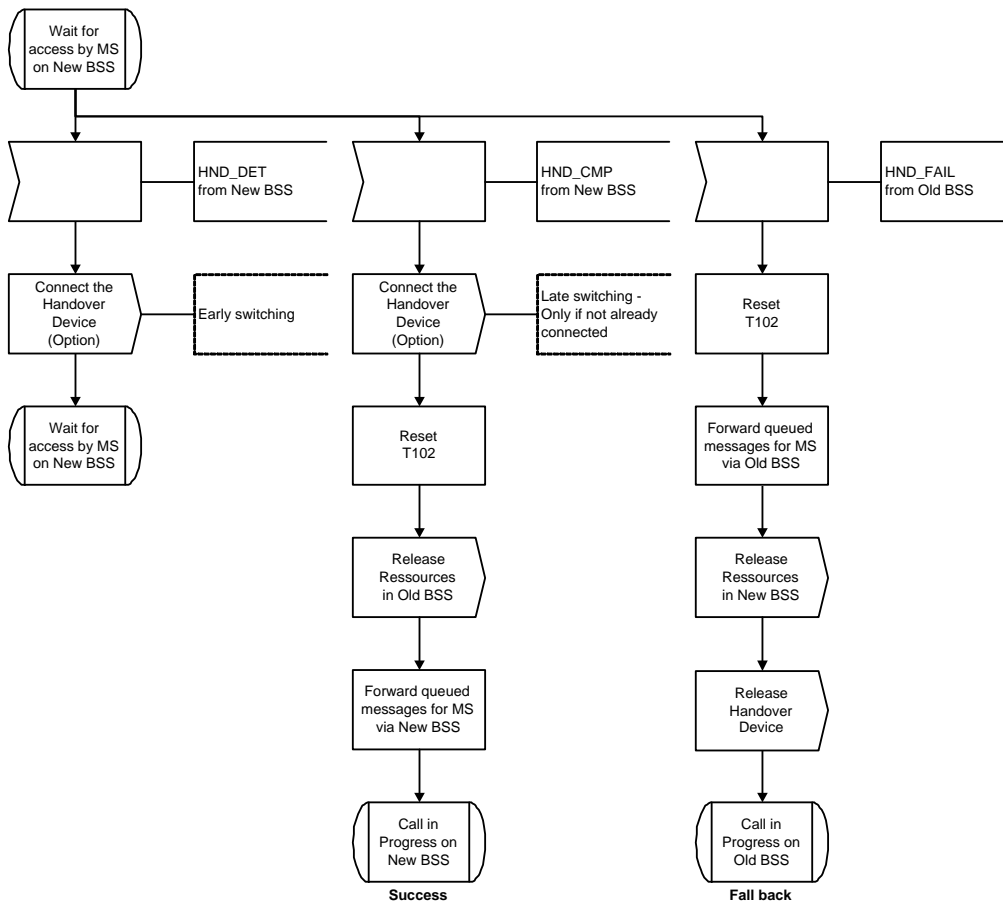


Figure 4.9: SDL describing the behavior of the MSC during an intra-MSC handover — part 3 of 4.

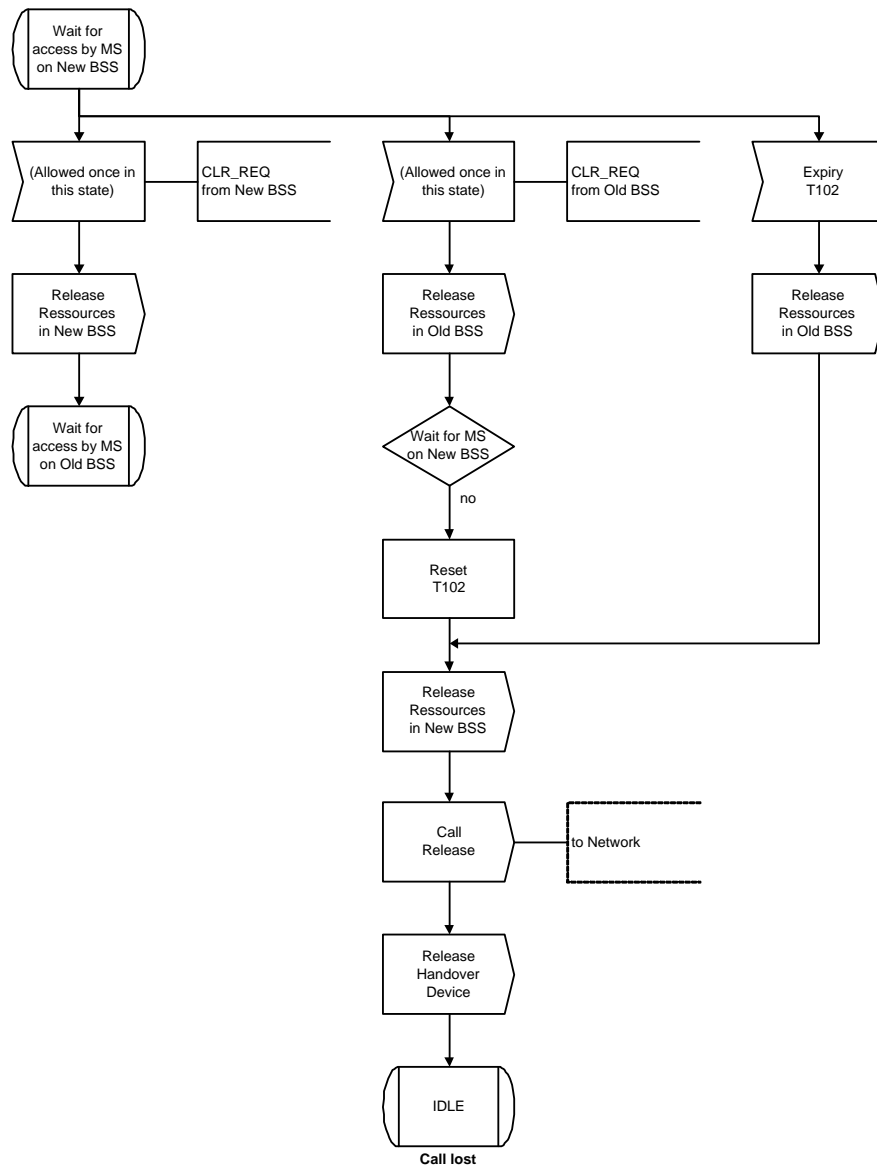


Figure 4.10: SDL describing the behavior of the MSC during an intra-MSC handover — part 4 of 4.

time. The handover must complete or fall back in order to inform the MS of the termination of the call. The outcome of this event has no influence on the handover and has therefore been left out.

Conclusion on the SDLs Throughout the discussion of the SDLs, we have argued that our limitations and changes do not limit the validity of any results achieved from analysing the intra-MSC handover with our SDLs as a base.

4.3 The model design

In this section we summarize our design choices for our model. First we discuss the decision of using Coloured Petri Nets for our modelling. Next we discuss the general aspects of our design, i.e. what we model, how we model it, what we assume, etc. Finally we go into details on our datatypes — how we map GSM messages into CPN color sets and what details we include.

4.3.1 SDL vs. CPN

Within telecommunication people usually use SDLs for describing their systems. This is also the case for GSM. However, the SDLs available in the GSM recommendations are incomplete and ambiguous. Incomplete because the only entity specified in SDL is the MSC. Ambiguous because the entity to communicate with is specified as *BSS* — not BSS-A and BSS-B or old/new BSS. Since the SDLs could not be used as they are, we had to model everything from scratch and decided to use a tool and a formalism we are familiar with. We also wanted to be able to do formal analysis of our model in order to investigate certain properties of the GSM intra-MSC handover. This led to CPN.

We are aware that research has been done and tools exist (e.g. Maria [17]) being able to automatically transform SDLs to Petri Nets. Since only small parts of the entities within GSM have been specified by SDLs, we decided to not look further into such tools and techniques.

4.3.2 The general model design

As previously stated, our goal with the model is to *understand and explain how intra-MSC handovers are performed within GSM networks*. To achieve this, we need to limit the problem domain, i.e. the GSM recommendations. We look at a limited part of a GSM network, containing one MSC having two BSSs connected. Each BSS contains one BSC and one BTS. Finally we have a single MS, which is “connected” to the first BSS (Old BSS), having an ongoing call. The setup of the model is illustrated on figure 4.11. What we want to do with this setup is performing a handover from the old BSS to the new BSS, requiring the MSC to

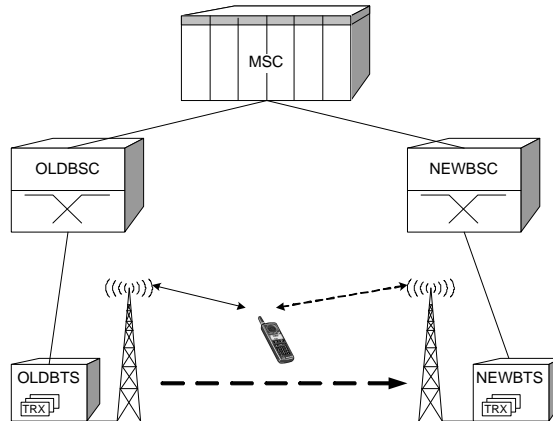


Figure 4.11: The entities of our model design.

take part. The decision whether to perform a handover or not, lies in the old BSC. The algorithm for deciding to perform a handover is not a part of our work — we assume that the decision has been made. We have limited the functionality of all the entities to the minimum needed, in order to perform the handover. The two BSSs are limited even more: The old BSS only contains functionality to hand over the call and the new BSS only contains functionality for receiving the call being handed over. The choice to limit the functionality is made in order to make the handover itself stand out clear and not mixed with other functions. In a real system, however, the functionality is not separated.

Going into a deeper level of technical details, we look at our abstractions on the protocol level. Almost all our work is concentrated on layer 3 of the protocols involved. As mentioned in section 4.1, only two layer 2 messages is inspected and modelled. Leaving out the lower layers of the protocols is the result of lots of discussions and thoughts. Our conclusion was, that the lower layers provide reliable communication for layer 3 and up, and that errors here can be thought of as a messages coming through or not. Because of this decision, our model is abstract enough to understand and still detailed enough to be interesting.

4.3.3 Messages

Within a GSM network, the actual data packets contains a lot of information needed by the network. Most of this information is irrelevant to our work and has therefore been left out. In this section we discuss our modelling of the messages. The discussion covers a general message design, then each interface independently, starting with the A-interface, next the Abis-interface, and finally the Air-interface.

General message design

Our first design of messages was a very real-system-like approach, in which we included all the fields and represented them in the bitwise manner. This approach gave us some quite unreadable messages in our model, which is why we removed the parts having no relevance to our work and represented the remaining parts in a more human readable manner.

Where the same message name exist on more than one interface, it has been suffixed with a '2' on the last interface it arrives on.

A-interface

The messages on the A-interface is either DTAP or BSSMAP messages — both SCCP messages; this was described in section 3.1.1. Also described in the same section, is the fact that SCCP is both layer 3 and layer 4. When DTAP and BSSMAP are built upon SCCP, it would be incorrect to call their messages layer 3, but as the other entities the MSC communicates with, treat the messages as layer 3, so do we.

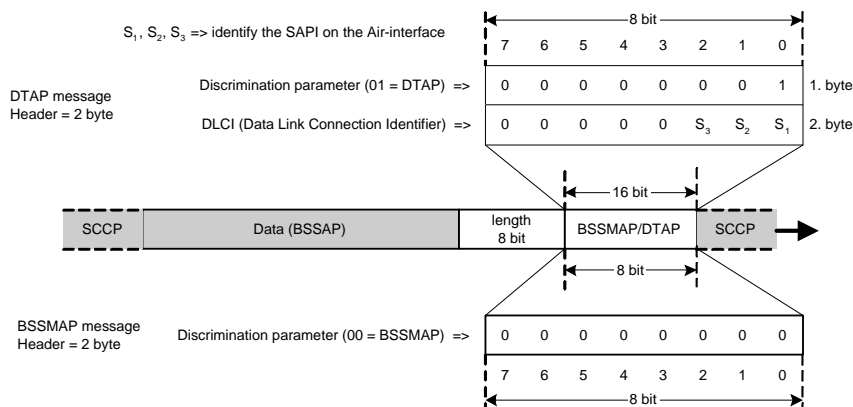


Figure 4.12: The DTAP and BSSMAP messages on the A interface.

The layout of DTAP and BSSMAP messages is illustrated on figure 4.12; their content is fairly simple. Most of the header is statically determined from the message kind (DTAP or BSSMAP), but one might wonder where the message type is (e.g. HND_RQD). DTAP messages are transparent to the BSS and does therefore not contain any message type visible to this link; the message for the MS is the data part. In the case of BSSMAP, the first 8 bit of the data part contains the message type field. The details of the DTAP/BSSMAP messages can be found in [3] section 10.2.2–10.2.3.

The color declaration for A messages is as follows:

```
color AMsg = union HND_RQD      : EntityID +
```

```

HND_REQ      : EntityID +
HND_REQ_ACK  : AirMsg +
HND_CMD      : AirMsg +
HND_DET2 +
HND_CMP +
CLR_CMD +
CLR_CMP +
HND_FAIL +
CLR_REQ;

```

The first message to discuss is the HND_DET2, which just is the HND_DET on the A-interface, having the '2' suffixed because it is the second interface it arrives on. Two messages contains an AirMsg as payload data (HND_REQ_ACK and HND_CMD); both messages are carrying the HND_CMD message for the MS. The HND_REQ and HND_CMD also carries payload data; in this case because of the model — not anything related to the handover functionality. This is discussed in section 5.1.

Abis-interface

On the Abis-interface, the layer 3 message is as illustrated on figure 4.13. Most of the fields of the message header are not relevant to our work and are therefore not discussed here. The values available for the Message Discriminator has previously been seen in section 3.1.2; the field itself is relevant when routing messages internally in the BSS for the parts responsible for each service. The Message Type field is the most important field in our context; it specifies what message is being sent, e.g. the HND_DET message has the value 27_{hex} . Details of the Abis layer 3 message header fields are found in [3] section 6.3.3.1.

The color declaration for Abis messages is the following:

```

color AbisMsg = union CHAN_ACT +
                      CHAN_ACT_ACK +
                      HND_DET +
                      EST_IND +
                      DATA_REQ : AirMsg +
                      DATA_IND : AirMsg +
                      RF_CHAN_REL +
                      RF_CH_REL_ACK +
                      CONN_FAIL;

```

Two messages above are new to the reader: DATA_REQ (DATA REQuest) and DATA_IND (DATA INDication). Their purpose is transporting data transparently through the BSS between the MSC and the MS, e.g. the HND_CMD message for the MS created by the new BSS. DATA_REQ is used when the BSC

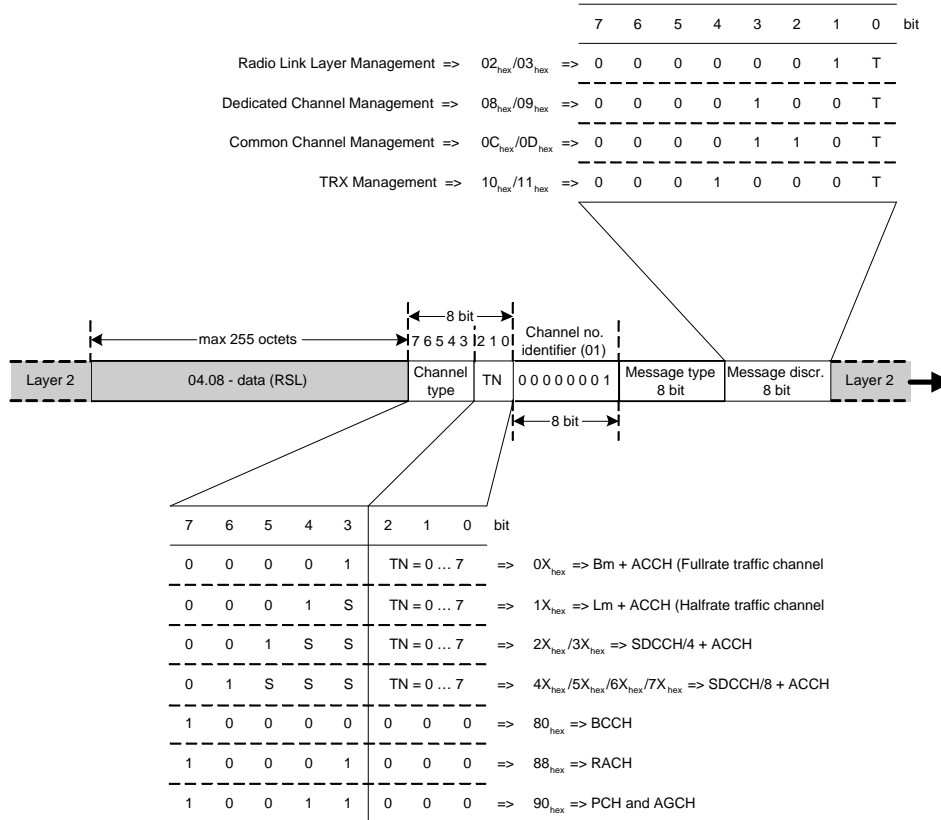


Figure 4.13: The layer 3 message on the Abis-interface.

wants to send data to the MS; DATA_IND is sent from the BTS to the BSC, to indicate that the message is received by the MS. The payload data part of the two messages, also shows that they transport Air messages. The rest of the modelled Abis messages does not carry any payload data, relevant to our work.

Air-interface

The layer 3 message of the Air-interface is shown on figure 4.14; it contains three parts: Type ID, Message type, and Data. The Type ID and the Message type part is the header of the message. The details of the header fields are very well covered in [3] section 7.5.2; here we just look at the 6 bit Message type field, which distinguish the Air messages from each other, e.g. the HND_CMD sent from the BTS to the MS has the value 2B_{hex} in the Message type field. In the Data part of the real Air message, data is represented as a bit-stream, which is very hard for humans to read. In most cases our work is not concerned with the actual payload data of the message, which makes the data field obsolete. The final design is a union data type, which can contain the needed type of payload data for messages where it is necessary and just the message type information

in the rest of the cases. The CPN color declaration for the Air message is the following:

```
color AirMsg = union HND_CMD2 : EntityID +
                    HND_ACC +
                    HND_FAI +
                    PHYS_INFO +
                    SABM +
                    UA +
                    HND_COM;
```

An important decision, regarding the two level 2 messages in the color declaration (SABM and UA), is that we chose to treat them as they were layer 3 messages in order to get a simple message flow without need for separating the different layers from each other. However, we are not consistent bringing in all layer 2 messages and doing the usual wrapping of layer 3 messages in layer 2 messages. Our reasons for this approach are, that only the necessary messages are included and are all considered being first order messages that either arrive or not. The actual layer they belong to is not important within this context.

The payload data of the messages is, as described, mostly non existing, except for the HND_CMD2 (which is the HND_CMD message arriving on its second interface), which contains the entity of the BTS to be handed over to.

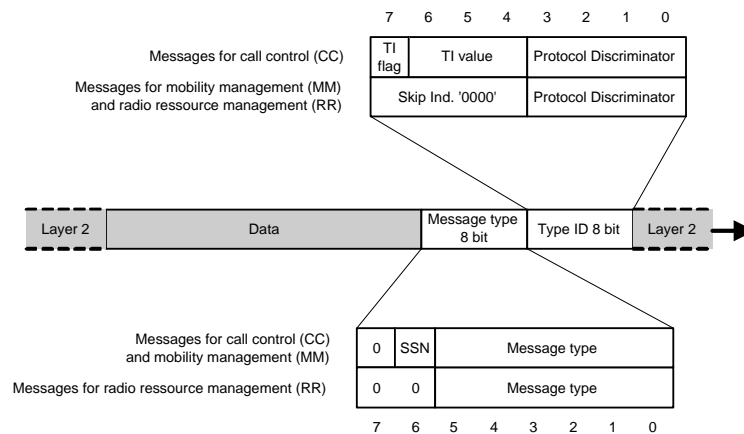


Figure 4.14: The layer 3 message on the Air-interface.

4.4 Summary

In this chapter we started out with a detailed discussion of the message exchange during a successful intra-MSC handover. Next we looked at the failure conditions

in the network. We discussed our choice of making our own SDLs a base of our modelling and the decision of using CP-Nets in stead of using SDLs as is common praksis within telecommunication. In the last part of the chapter, we described our model design.

Chapter 5

Description of CPN Model

In this chapter we give a description of our CPN model of the GSM handover functionality. We start out with a discussion of our specific modelling aspects. This includes categorization of places and transitions, initial state, naming conventions, token flow within the pages, modelling of GSM timers, and color sets. After these general aspects we look into each of the individual pages, discussing their correspondance to the specifications discussed in chapter 4. The discussion follows the structure of the network starting at the MSC, next the BSC, the BTS, and finally the MS. The individual nodes have some of their implementation deferred to subpages; these will be described after the page, on which they are used for the first time. The hierarchy page of the model can be found in appendix C.

5.1 Modelling aspects

In this section we describe the choices of modelling conventions we have made throughout our work with the CPN model. It includes references to the CP-Net, which serves as examples and will be described in detail in the sections describing the individual pages.

Categorization of places In our model we have different roles of the places, and we categorize these in the following: entity state, message exchange, and setup places. We have mapped these categories to some layout conventions, which are shown on figure 5.1 (a)–(c). The figures also illustrate the positioning conventions we have used for the color sets and initial markings of the places.

The entity state places model the state of a single entity and should contain at most one token. Nearly all our entity state places have the color set E, indicating that no data is needed to represent the state of the device — only the location of the state-token.

Message exchange places are used to exchange messages between the involved entities, e.g. between the MSC and the BSC. These places have the most complex

color sets of our model. They model the different messages exchanged on the specific interfaces. We model message exchange places with multisets instead of queues. The advantage with this approach is, that messages can overtake each other and that the entities are able to prioritize the incoming messages in the way they want.

The last category of the places is setup places. Most of the setup places are needed to generate message sequence charts, because some entities share model implementation, i.e. use the same subpage; therefore we need to provide the identity of the entity, to be able to distinguish between them on the subpage. In the figures of the model these places will be grayed, since they do not affect the logic being modelled.

The categories will be used in the descriptions of the pages to give a better understanding of the meaning of the places.

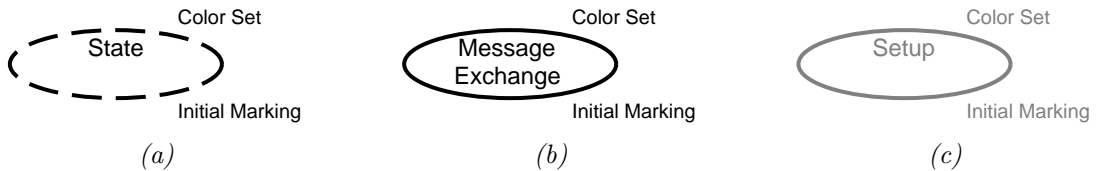


Figure 5.1: The layout of the places: (a) Entity state with the dashed border, (b) Message Exchange with the solid border, and (c) Setup being grayed.

Categorization of transitions In our model we have two kinds of transitions, distinguished by the actions they model: message transfer and time outs. The transitions follow the layout conventions found in figure 5.2 (a)–(b).

Message transfer transitions model the participation in a message exchange. They require a specific message from another entity to be enabled, and their occurrence changes the state of the entity. Sometimes they also send a message to another device. A result of this is, that the internal decisions taking place in a real system, before a message is sent, is modelled as an atomic operation. We have made this decision because we want to investigate the message flow between the entities, not the possible internal error cases of the involved devices.

In the problem domain we have several timers. We want to investigate all the possible executions of the handover functionality, and therefore we modelled time out of timers with the non-deterministic formalism of CPNs. The timers



Figure 5.2: The layout of the transitions: (a) Message transfer with the solid border and (b) Timeout with the dotted/dashed border.

are running in specified periods of the handover. We model this by enabling a timeout transition during the part of the the protocols, in which the timer is running.

Substitution transitions The substitution transition is a feature of the Design/CPN which allows splitting up model parts to several pages. We use the mechanism to divide the possible outcomes of the handover into comprehensible parts, from which some are reused in several pages. The layout of substitution transitions is shown on figure 5.3.

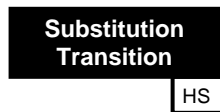


Figure 5.3: Substitution transitions are fill black boxes. The analogy to the black fill color is the *black box* — a box handling some functionality without showing how.

Naming conventions We use some naming conventions in our model, many of these contain the abbreviations from the literature, tying the model as close to the references as possible.

The transitions involved with sending and receiving messages are named by the abbreviation of the message used in chapter 4 and the action performed, e.g. `recCLR_CMD` means the reception of the message `CLR_CMD`.

The places are, as previously mentioned, divided into three categories; the naming conventions of these follow their categories. The entity state places are typically called something like `WaitForHND_CMD`, when we are expecting a `HND_CMD` message. Again we use the abbreviations from chapter 4. The message exchange places are named by their interface, direction, and the prefix 'old' or 'new', whenever ambiguity is present, e.g. `Abis RX` means that the interface is `Abis` and the place is for reception (`RX`); `TX` on the other hand is short for transmission. The setup places hold identity information and are therefore suffixed with 'ID'.

Layout of the pages Most of the pages in our model are build upon the same layout conventions, to give a consistent structure. The top and bottom of the pages usually contain message exchange places. Most of the communication interfaces are modelled by both a place for up- and downlink; the only exception is the Air-interface, which only consist of one place for communication.

In the middle of the page we have the protocol flow. The state tokens propagate through the model from left to right, with only minor exceptions; these will be covered as they come up. The main flow shows the successful handover case,

and when we have alternative actions from a state, these are placed underneath the main flow. The main flow is illustrated on the pages with thick arcs.

The way substitution transitions are placed and sized, shows their place in the protocol flow. Imagine a timeline going horizontally from left to right on the pages. The substitution transitions are placed on this timeline, from which their relative start, duration, and end, can be read.

Color sets The color sets of our model follow the categories of the places. The entity state places are modelled by the color `E`, which is used to model a token without any extra color information. The color sets of the message exchange places models the messages we exchange on the interfaces, as discussed in section 4.3.3. The color set of the setup places is `EntityId`, which ranges over the possible types of entities, and are used to distinguish the different entities on the shared subpages. Compositions of the above color sets are also present. An example is the color set `MsgIdxAbisMsg`, which is a composition of the two color sets, `MsgId` and `AbisMsg`. The `MsgId` part is used in the generation of message sequence charts, and the `AbisMsg` ranges over the possible messages on the Abis-interface. The declaration node of the model is included in appendix C.

Initial state The model is initialized with the MS having an ongoing call. During the call, the measurement reports sent to the serving BSC (`OLDBSC`), indicates that a handover to a cell in another BSS (`NEWBSC`) is required. The only enabled transition in the initial state is `sendHND_RQD`. After occurring, a `HND_RQD` message is sent to the serving MSC. This initiates the execution of the model. The initial state of the model resembles the first *Call in progress on Old BSS* from the SDL on figure 4.7.

5.2 CPN pages

We are now ready to look at the individual pages of the CPN model. We will not go into all the details of the individual pages, but describe the functionality they model.

5.2.1 GSM

In the top of our hierarchy we have the `GSM` page. It contains the overall setup of our model and the involved entities of an Intra-MSC handover: `MSC`, `OLDBSC`, `NEWBSC`, `OLDBTS`, `NEWBTS`, and `MS`; they are all modelled as separate subpages described in later sections. Their role in the handover scenario is prefixed if any ambiguity is present, e.g. `OLDBTS` is the `BTS` initially serving the `MS`. The `GSM` page is shown on figure 5.4.

Communication between entities The communication between the entities in the network is modelled in two different ways. For all communication interface we have a port-socket solution in order to separate entities in subpages. The fixed connections from the real world (the cables, which are all interfaces except Air), we have two places: one for uplink and one for downlink. We chose this solution to make the direction of the message flow more obvious. The places involved in these connection are: A Downlink, A Uplink, Abis Downlink, and Abis Uplink.

The communication over the Air-interface is modelled by one place called AIR, i.e. we do not separate up- and downlink communication. The reason is that communication between the MS and the BTSs is wireless and hereby very dynamical and interference canceling messages is possible.

The location of the interfaces on this page illustrates the convention we use for positioning the interfaces on the subpages. The page is shown on figure 5.4.

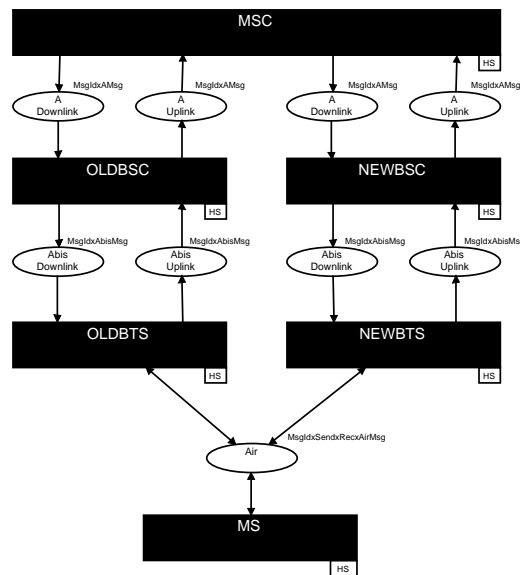


Figure 5.4: The GSM page

5.2.2 MSC

The MSC page models the handover functionality located at the MSC. Most of the modelled functionality is implemented on the three subpages of the page. The partitioning of the functionality has been made on the possible outcomes of the handover: `HandoverPossible`, `FallBackToOldBSS` and `ReleaseCallNecessary`. The page contains interfaces to the two BSCs at the bottom; the suffices `old` and `new` indicate the roles of the connected BSCs. The three places on the right: `HandoverSuccess`, `FallBack`, and `CallReleased`, indicates the outcome of the handover from the MSC's point of view.

The two places on the page, `WaitChanAlloc T101running` and `WaitMSAccess T102running`, are states picked directly from the description of the MSC. These are the major states, *Wait for channel allocation in New BSS* and *Wait for access by MS on New BSS*, from the SDLs on figure 4.8 and 4.9, indicating the phases of the handover: channel allocation and MS access on the new BSS.

The only real transition on the MSC page is `rechHND_RQD sendHND_REQ`, which initiates the MSC's role in the handover. It is enabled when the MSC receives a `HND_RQD` from the old BSS. When occurring it sends a `HND_REQ` to the new BSS. The two messages are the MSC's contribution to messages 3 and 4 on figure 4.5. The positioning of the elements illustrate the flow of the execution. The main flow, which is the successful handover scenario, is located at the top. The MSC page is shown on figure 5.5.

In the following three sections we give a more detailed description of the `HandoverPossible`, `FallBackToOldBSS`, and `ReleaseCallNecessary` subpages.

HandoverPossible

The `HandoverPossible` page models the situations, where a handover is still possible from the MSC's point of view. We follow the convention from the MSC page with the interfaces to the BSSs at the bottom; note that we only *receive* messages on the new interface, illustrated by the absence of the place for transmission. The other four places, `T101started`, `T102started`, `HND_CMPrec CLR_CMDsent(A)`, and `CallInProgress(B)` are all entity state places, capturing the progress of the handover in a successful attempt. Note that the place `T102started` is an `Input/Output` port place, i.e. tokens from this place are used in bindings on pages above this in the hierarchy.

All transitions on the page, except `rechHND_DET`, make state changes by moving the state token to another entity state place. The reason `rechHND_DET` is not altering the state of the entity, modelled by putting the state token back on input place, follows from the discussion of early and late switching. This modelling allows both early and late switching, because the MSC does not demand receiving the `HND_DET` before receiving `HND_CMP`. The page is shown on figure 5.6.

FallBackToOldBSS

The second possible outcome of the handover is that the call falls back to the old BSS and continues as if nothing had happened; this is modelled at the `FallBackToOldBSS` page. We have the communication interfaces at the bottom. In the top we have the entity states that can result in the *fall back to old BSS* scenario: `WaitChanAlloc T101running` and `WaitMSAccess T102running`. The two transitions `T101Timeout sendCLR_CMD` and `rechHND_FAIL sendCLR_CMD` are the possible actions initiating a fall back. The result of both of these are the sending of a `CLR_CMD` message to the new BSC. The scenarios end by reception

of a CLR_CMP, and the call is continued on the old BSS. The page is shown on figure 5.7.

ReleaseCallNecessary

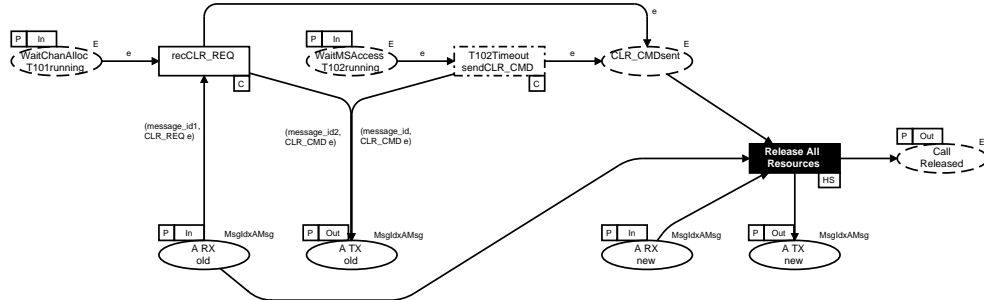


Figure 5.8: The ReleaseCallNecessary page

The last possible outcome of the handover is that we have to release the call due to lack of a radio path. The communication interfaces are located at the bottom. In the top we have the places modelling the states, where the scenario is possible. The two transitions, `recCLR_REQ sendCLR_CMD` and `T102Timeout sendCLR_CMD`, represent the events leading to a complete release of the call. Both events will initiate the release with a `CLR_CMD` to the old BSS. The substitution transition `Release All Resources` finishes the release in the new BSS, and the call is released, when a token is put on `CallReleased`. The page is shown on figure 5.9.

ReleaseAllResources

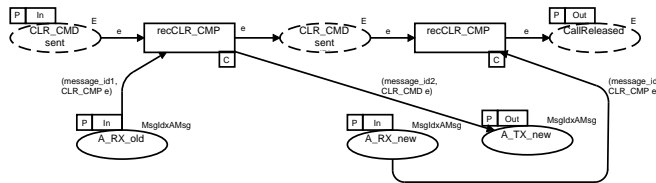


Figure 5.9: The ReleaseAllResources page

The `ReleaseAllResources` page models the release of resources in both BSSs. It releases the resources sequentially starting with the old BSS. The page is shown on figure 5.9.

5.2.3 OldBSC

The `OldBSC` page combines the possible involvement of the old BSC in the handover. The two entity states, `WaitForHND_CMD` and `WaitForCLR_CMD`, be-

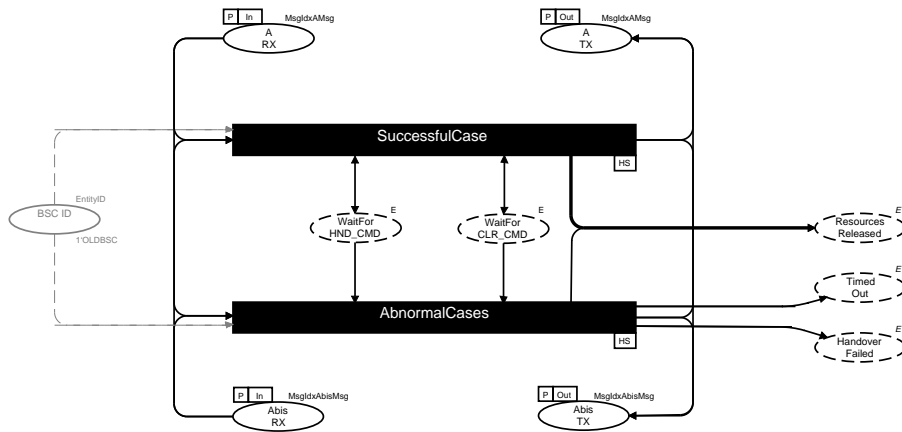


Figure 5.10: The OldBSC page

tween the substitution transitions handle the states that are shared between the subpages. Their specific roles in the scenarios are explained on the subpages. The three entity state places on the right are terminal states of the old BSC, i.e. states where no more execution is possible. The page is shown on figure 5.10.

SuccessfulOldBSC

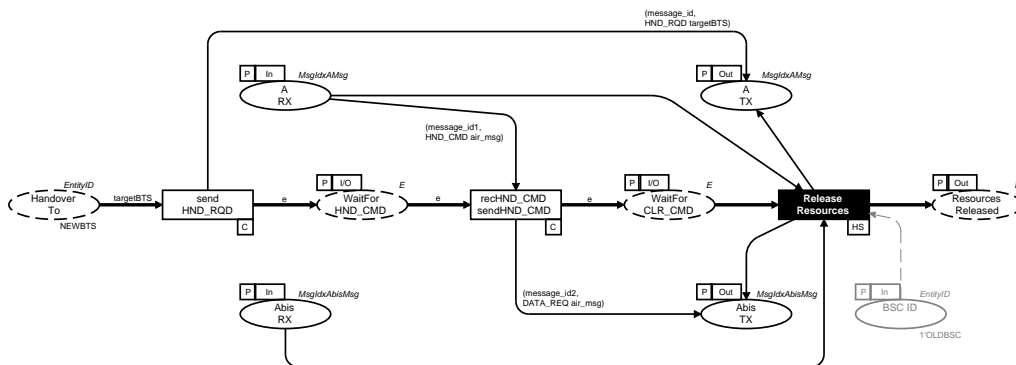


Figure 5.11: The SuccessfulOldBSC page

The SuccessfulOldBSC page models the old BSC's involvement in a successful handover. The flow is straight forward and illustrated by the thick arcs. The transition `sendHND_RQD` initiates the handover, and is the only enabled transition in the initial state. The page is shown on figure 5.11.

ReleaseResourcesBSC

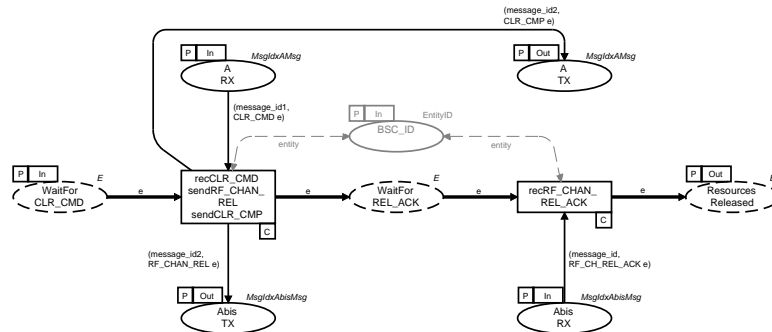


Figure 5.12: The ReleaseResourcesBSC page

The ReleaseResourcesBSC page is shared between the new and the old BSC, and reused heavily in subpages of these. The logic of the release is rather simple, and models the BSC’s involvement in the messages 22, 23, and 24 from figure 4.5. The page is shown on figure 5.12.

AbnormalCasesOldBSC

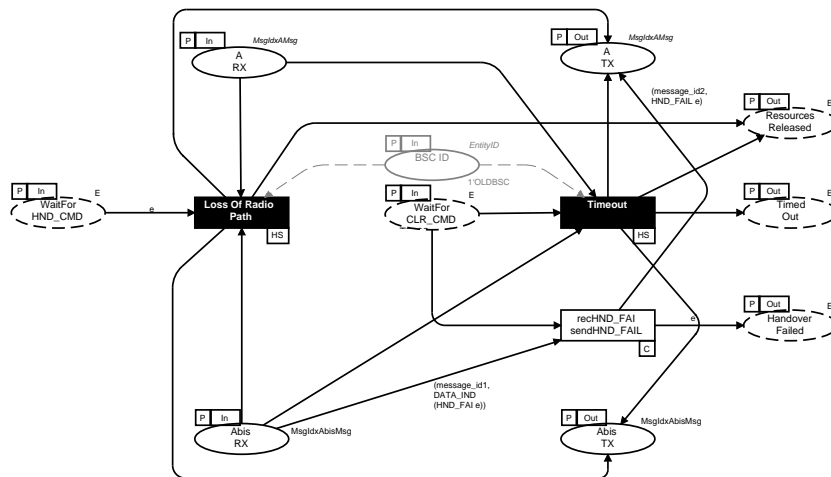


Figure 5.13: The AbnormalCasesOldBSC page

The AbnormalCasesOldBSC subpage models all the abnormal cases of the handover, within the old BSC. These are losing the radio path to the MS, receiving a HND_FAI message from the MS, and timeout of T3103 timer. The ‘loss of radio path’ condition is modelled on the Loss Of Radio Path subpage and the

timeout is modelled on the TimeoutT3103. The detected failure of the handover is modelled by the transition `rechND_FAIL sendHND_FAIL`. Figure 5.13 shows the `AbnormalCasesOldBSC` page.

TimeoutT3103

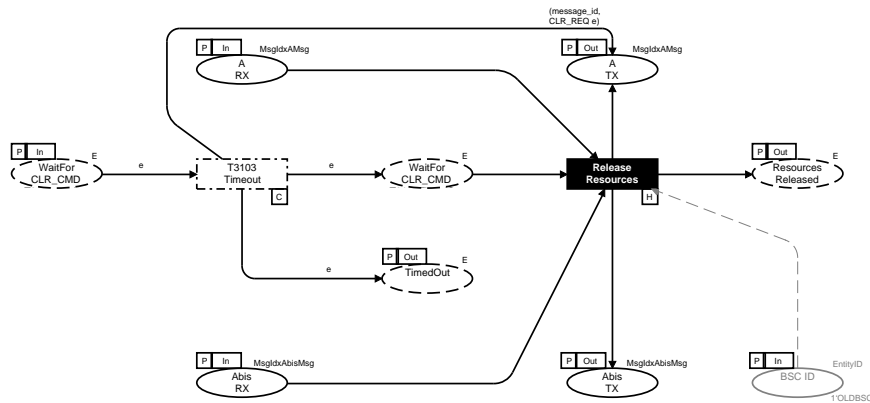


Figure 5.14: The TimeoutT3103 page

The TimeoutT3103 page is a subpage of `AbnormalCasesOldBSC` and models a timeout of the T3103 timer, as described in section 4.1.3. The timeout can occur when the old BSC is waiting for a `CLR_CMD` from the MSC. When T3103 times out, the allocated resources are released. This is modelled on the `ReleaseResourcesBSC` page (represented by the `Release Resources` substitution transition). As previously described, the BSC is allowed to release its resources when the `CLR_REQ` message has been sent to the MSC, but we chose to wait for the `CLR_CMD` from the MSC before releasing the resources. The reason for this, is reuse of the `ReleaseResourcesBSC` page; when we wait for the `CLR_CMD`, the release is identical to the release initiated by the MSC. However, our choice is in accordance with the recommendations because they say, it is *allowed* to start releasing before receiving `CLR_CMD`, but not *required*.

The entity state place `TimedOut` is used for storing the state of the BSC, when the simulation is completed in order to inspect the outcome. The TimeoutT3103 page is shown on figure 5.14.

LossOfRadioPath

The `LossOfRadioPath` page is almost identical to the TimeoutT3103 page. One difference is, that we do not save the state of the device on a special entity state place. The transition `RadioPathLost sendCLR_REQ` is layouted as it is a timeout transition, but it is also a message transfer transition. We chose the

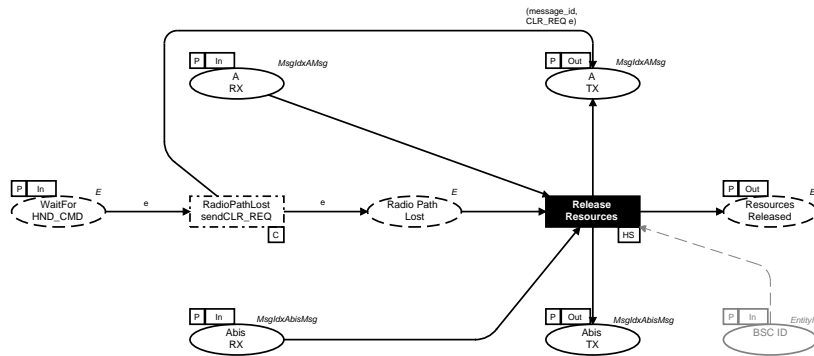


Figure 5.15: The LossOfRadioPath page

timeout layout to clarify that the transition handles a failure condition. The LossOfRadioPath page is shown on figure 5.15.

5.2.4 NewBSC

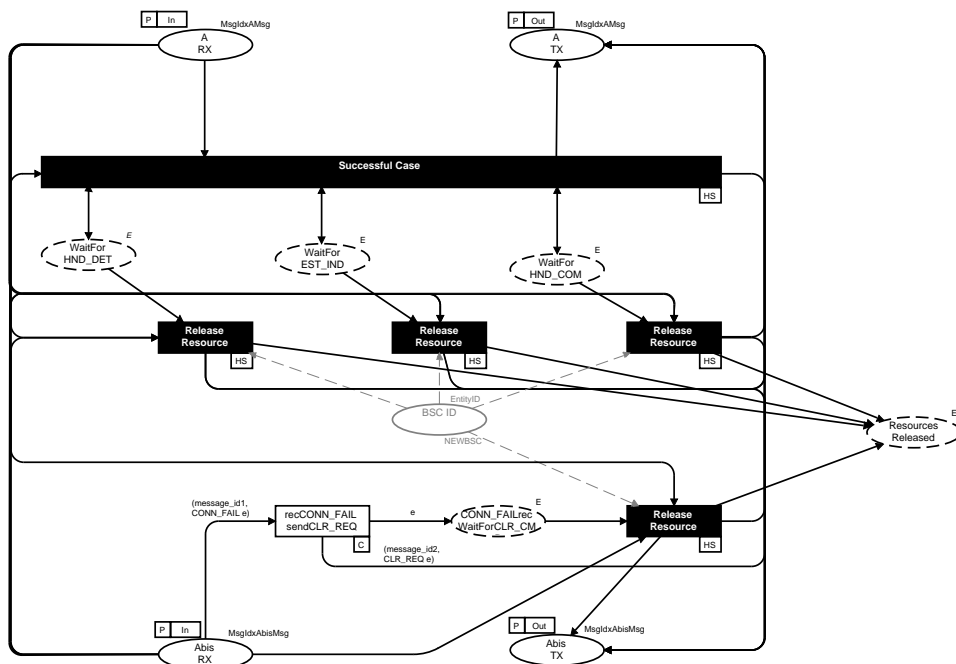


Figure 5.16: The NewBSC page

The NewBSC page models the behavior of the new BSC during the intra-MSC handover. The interface towards the MSC (A-interface) is located in the top of the page and the interface towards the BTS (Abis-interface) is located in the bottom. The wide substitution transition below the A-interface, Successful Case,

models the successful handover. During the handover, a failure or a timeout in either the MSC or the old BSS might occur. Such a failure is indicated to the new BSS by the MSC ordering it to release its resources with a CLR_CMD message. The handling of such failures is modelled by the Release Ressources substitution transitions, referring to the ReleaseResourcesBSC page.

If the MS fails to synchronize with the new BSS (timeout of T3105 Ny1 times), the new BTS sends a CONN_FAIL to the new BSC. The new BSC initiates a release of its resources and awaits an acknowledgement from the MSC. This is modelled in the bottom of the page by 'recCONN_FAIL sendCLR_REQ' and 'CONN_FAILrec WaitForCLR_CMD'. Figure 5.16 shows the NewBSC page.

SuccessfulNewBSC

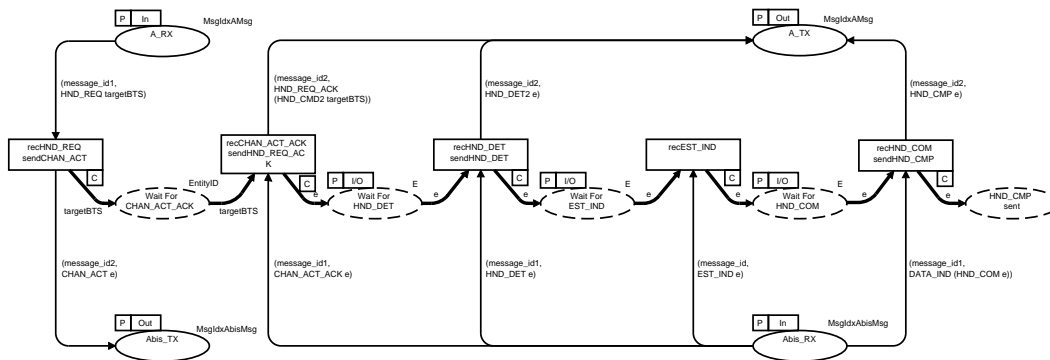


Figure 5.17: The SuccessfulNewBSC page

The SuccessfulNewBSC models the behavior of the new BSC during a successful handover. As usually the thick arrows in the middle indicates the successful flow. The page shows the BSC's part of message transfers 4–7, 13–14, 16, and 19–20 from figure 4.5. The SuccessfulNewBSC page is shown on figure 5.17.

5.2.5 OldBTS

The OldBTS page models the old BTS' involvement of the intra-MSC handover. In the top of the page is Channel Management modelled in a substitution transition (Channel Management). This part is the messages exchanged exclusively between the BSC and the BTS. The messages exchanged with the MS are not processed by the BTS and though relayed directly to or from the MS. This is modelled by the two transitions relay RLM and recHND_FAI relay it. The OldBTS page is shown on figure 5.18.

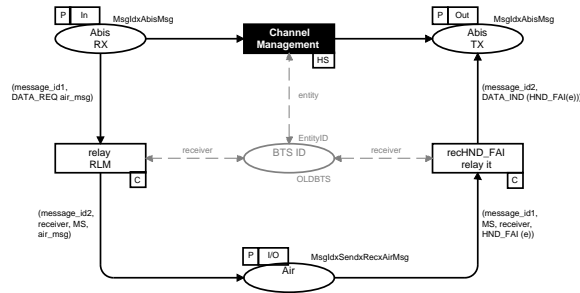


Figure 5.18: The OldBTS page

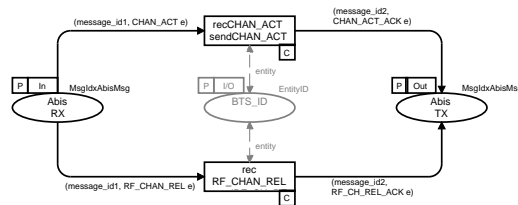


Figure 5.19: The ChannelManagement page

ChannelManagement

The ChannelManagement page is the subpage of OldBTS and NewBTS modelling the internal communication between the BSC and the BTS. The page could have been split into two in order to keep functionality from the old BTS separated from the new BTS, but because of the simplicity of the page, we decided to keep this part together. The topmost transition models message 5–6 from figure 4.5 and the bottom-most transition models message 22 and 24 from the same figure. Figure 5.19 shows the ChannelManagement page.

5.2.6 NewBTS

The NewBTS page models the new BTS' behavior of the handover. In the middle, indicated by the thick arrows, is the successful case, modelling the message exchange from figure 4.5 in which the new BTS is involved (5–6, 11–13, and 15–19). In the top of the page, between the message exchange places for the Abis-interface, is the substitution transition ChannelManagement referring to the subpage with the same name. The only failure condition the new BTS can be involved with, is timeout of its timer (T3105) — or more precisely T3105 timing out Ny1 times. This is modelled by the Ny1xT3105 Timeout transition. The T3105 TimedOut place is for investigating the state of the new BTS after simulation. Figure 5.20 shows the NewBTS page.

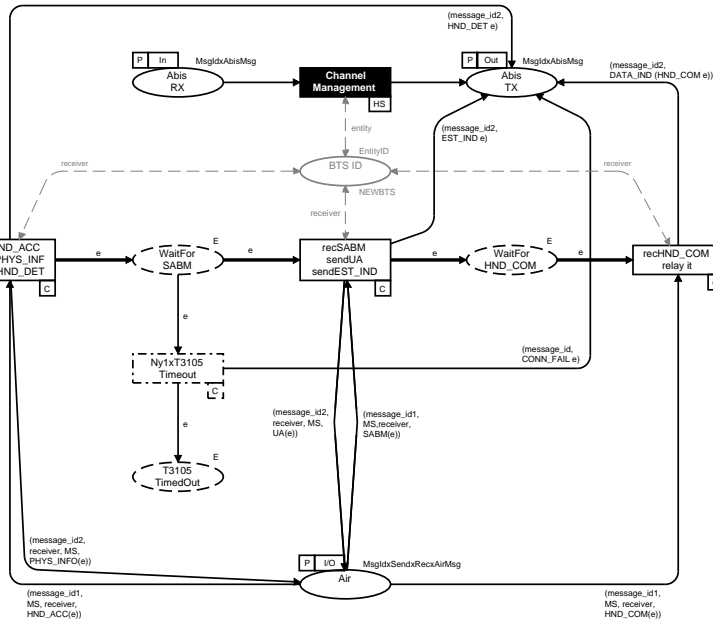


Figure 5.20: The NewBTS page

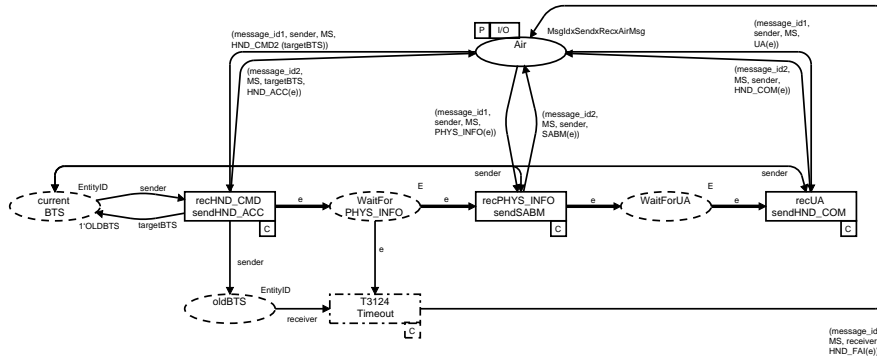


Figure 5.21: The MS page

5.2.7 MS

The last page in the model is the MS page, modelling the MS' involvement in the handover. The middle part is as usual the successful case indicated by the thick arrows. Only one failure condition is possible in the MS — timeout of T3124. The outcome of this timeout is the MS trying to fall back to the old BTS. We have not modelled the actual fall back, because it is only relevant to us, that the situation occurred. The MS page is shown on figure 5.21.

5.3 Summary

Throughout this chapter, we have looked at our CPN model. First we introduced our modelling conventions and in the rest of the chapter, we discussed the details of the individual pages, including the decision of how we have modelled the various parts of the problem domain.

Chapter 6

Validation of the Model

In this chapter we present the methods we have used in order to assure that our model behaves as described in chapter 4. The primary tool for the validation of the model has been generation of message sequence charts. The chapter starts with a discussion of the model structure and its indications of validity of the model. Next we look at the message sequence charts generated by model, illustrating some of the possible executions of the handover.

6.1 Model structure

We have constructed our model to resemble the problem domain as much as possible. We have modelled the entities, i.e. MSC, BSC, BTS and MS, as separate pages in the model, and their communication is done by tokens modelling messages. This makes the validation of their behaviour simpler, because it allows us to look at the instances of the entities separately. The roles of the individual transitions and their correspondance to the problem domain are described in chapter 5.

In addition to the description in chapter 5 we have had a review with some engineers from the problem domain in an early phase of the modelling. The outcome of this meeting was that our model was understandable to these problem domain experts. These engineers are not familiar with the formalism of CPN, but we were still able to discuss problem domain specifics based on the model. This was the first step towards validity of the model.

6.2 Simulation scenarios

The next step in the process of establishing validity of our model, is generation of message sequence charts, which capture the scenarios mentioned in chapter 4. This result of this step serves both as a description of the dynamics of the model, and insurance that our model capture the major scenarios of a handover.

We start by describing the way we generate the message sequence charts, and then walk through the scenarios: Successful handover, Fall back to the old BSS, and Release Call.

6.2.1 Generation of Message sequence charts

The communication between the entities is done in two steps. In the first step a transition places a token, representing a message, on the message exchange place. In the second step another transition removes this token from the place. This two step communication allows messages to overtake each other. We are building the message sequence charts in the same way, to illustrate these possible scenarios, where messages are able to overtake each other.

We generate the message sequence chart by using a library extending the functionality of the message sequence chart library available from [2]. The extended library allows us to generate the message transfer in the two steps as discussed above.

6.2.2 The scenarios

We now look at the contents of the scenarios. They are generated during simulation, where the transitions are chosen manually, in order to generate the scenario we want to illustrate.

Successful handover

In this section we look at one of the possible executions during a successful handover. This execution is not the only one resulting in a successful handover, but only minor differences can be found between scenarios resulting in a successful handover. The generated message sequence chart is illustrated on figure 6.1.

The message exchange can be decomposed into 4 phases: Initiation, channel allocation, handover execution, and deallocation of resources.

- **Initiation** The initiation phase only includes the message *HND_RQD*. The message informs the MSC that a handover is required. It includes a list of possible cells to hand over to.
- **Channel allocation** Channel allocation consist of 4 messages: *HND_REQ*, *CHAN_ACT*, *CHAN_ACT_ACK*, and *HND_REQ_ACK*. In this phase we allocate resources for a call in the new BSS. The new BSS answers the MSC with the *HND_REQ_ACK*, which includes the *HND_CMD* for the MS.
- **Handover Execution** This is by far the largest step, involving 13 messages, starting with *HND_CMD* and ending with *HND_CMP*. It takes

care of informing the MS of the new channel. After this the MS access the specified channel, finishing with the *HND_CMP* message.

- **Deallocation** This phase handles the deallocation of the resources in the old BSS when they are not needed any more, i.e. when the MS has accessed the new channel. It starts with the message *CLR_CMD* and includes the rest of the messages.

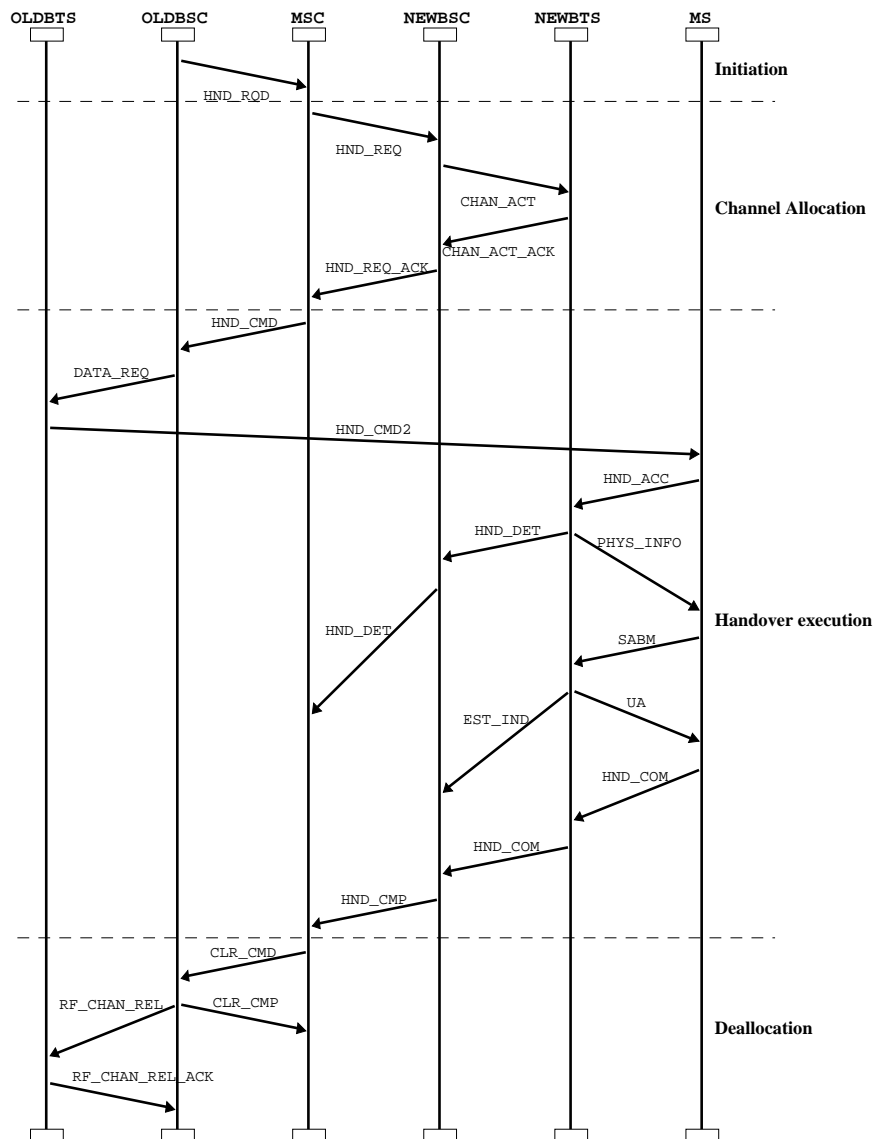


Figure 6.1: Scenario with a successful handover

Fall back to old BSS

This scenario captures the situation where the MS is unable to access the new channel and therefore falls back to the old channel. This could happen due to several different reasons, e.g. the MS is moved towards the old BTS and away from the new BTS, resulting in a lack of radio path to the new BTS. The scenario is illustrated on figure 6.2.

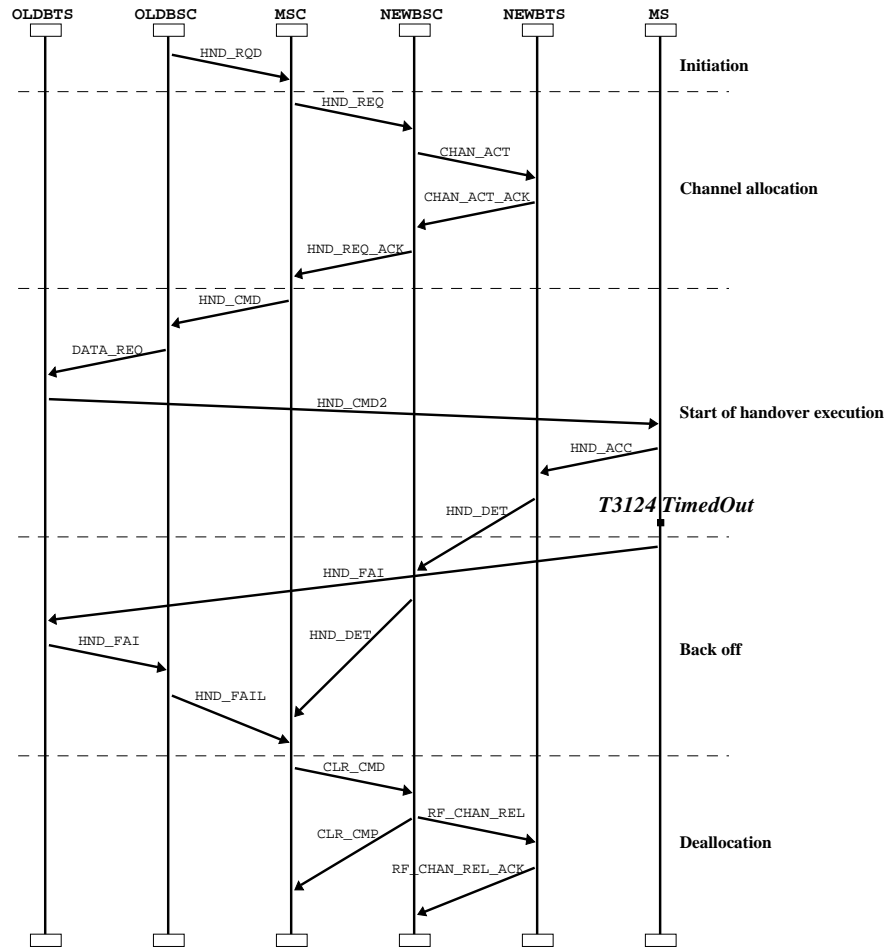


Figure 6.2: Scenario where the MS fall back to the old BSS.

The scenario can be divided into the following phases: initiation, channel allocation, start of handover execution, Back off, and deallocation.

- **Initiation** The initiation of the handover is sending of *HND_RQD* from the old BSS to the MSC.
- **Channel allocation** Channel allocation consist of 4 messages: *HND_REQ*, *CHAN_ACT*, *CHAN_ACT_ACK*, and *HND_REQ_ACK*. In this phase

we allocate resources for a call in the new BSS. The new BSS answers the MSC with the *HND_REQ_ACK*, which includes the *HND_CMD* for the MS.

- **Start of handover execution** The execution of the handover stops when the MS realises that it can not access the new channel, represented by the timeout of T3124. This phase includes the messages *HND_CMD*, *DATA_REQ*, *HND_CMD2*, and *HND_ACC*. The phase is completed by a timeout of T3124, represented in the message sequence chart by the processmark *T3124TimedOut*.
- **Back off** When failing to access the new channel the MS sends a *HND_FAI* messages on the old channel, initiating the back off phase. The message *HND_FAI* (OLDBTS to OLDBSC) and *HND_FAIL* informs the MSC that handover has failed and the call continue on the old channel.
- **Deallocation** When receiving the *HND_FAIL* message from the old BSS, the MSC releases the allocated resources in the new BSS, and the call continues as if nothing has happened.

In the message sequence chart we also have two messages not mentioned in the above phases: The two *HND_DET* on the Abis and A interfaces. These messages are sent because the new BSS has not yet been informed about the unsuccessful handover. This could cause the MSC to switch the speech to the new BSS, until receiving the *HND_FAIL* from the old BSS.

Release Call

In this scenario we capture the situation where none of the BTSs are able to offer a radio path to the MS. The situation is of course not appreciated, but none the less the situation can occur in a real system. The scenario is illustrated on figure 6.3.

Again we divide the scenario into phases:

- **Initiation** The initiation phase just includes the sending of *HND_RQD*.
- **Start of handover execution** This phase includes the following messages: *HND_REQ*, *CHAN_ACT*, and *CHAN_ACT_ACK*. This allocates resources in the new BSS for MS to enable handover.
- **Loss of radio path** The event **Radio path lost** models that the old BSS loses connection to the MS, and this initiates the request for release of resources with *CLR_REQ*. This happens concurrently with the *start of handover execution* phase.

- **Release resources in old BSS** This phase is executed as a response to the *CLR_REQ*. It includes: *CLR_CMD*, *CLR_CMP*, *RF_CHAN_REL*, and *RF_CHAN_REL_ACK*. Notice that the old BSC acknowledges the *CLR_CMD* in the same step as the sending of *RF_CHAN_REL*.
- **Release resources in new BSS** When the resources in the old BSS are released, we start the release of the resources in the new BSS. The phase includes: *CLR_CMD*, *CLR_CMP*, *RF_CHAN_REL*, and in the end *RF_CHAN_REL_ACK*. This phase finishes the scenario and the call is released.

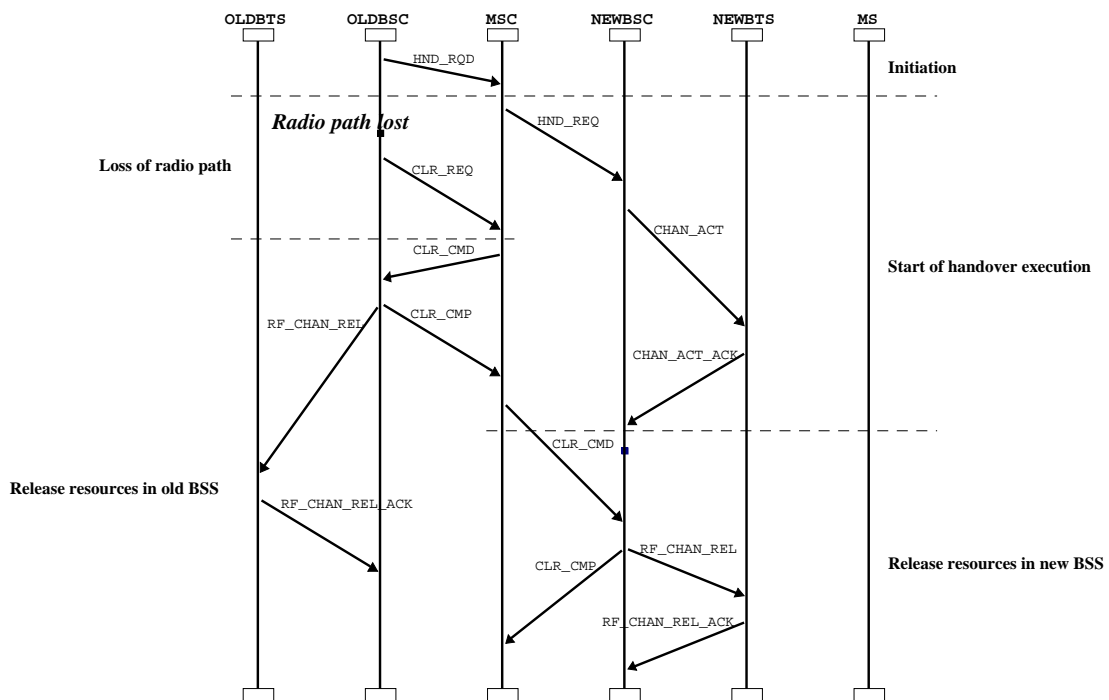


Figure 6.3: Scenario where we have to release the call

The release of the call is due to the loss of radio path before we have sent the *HND_CMD*, and therefore we have not yet informed the MS of the coming change of radio path. None of the BTSs have a connection to the MS and therefore a release of the call is necessary. Notice that some of the phases after the initiation phases happen concurrently on figure 6.3.

6.3 Summary

In this chapter we have validated our model to a certain level. The validity of our model is based on both structure and simulation. We have a close correspondance

between the model and the problem domain, based on the description in chapter 5 and review with problem domain experts. We have validated that our model is able to simulate essential scenarios of the handover. This concludes the validation process with curtain trust in the model.

Chapter 7

Verification

When looking at distributed protocols such as the GSM handover, there are various properties to verify. We have focussed on progress and outcome in our verification of the intra-MSD handover. We want to ensure that there is a progress in all possible executions. Furthermore we want the involved entities to agree on the outcome of the handover. We use the occurrence graph tool of Design/CPN to reason about these aspects.

In the following we start with a discussion on how to locate absence of progress and disagreement on the outcome of the handover. Next we look at the progress analysis based on arguments about the size of the occurrence graph and the strongly connected components graph. We end the analysis with an inspection of the terminal nodes of the occurrence graph to locate possible situations, where the entities do not agree on the outcome of the handover.

7.1 Discussion of progress and outcome

The handover can end in different situations. Whether the handover completes with success or failure is not important in our work. Our concerns are, whether the progress of the protocol is maintained and whether the entities in the network agree on the outcome.

The absence of progress in a protocol can happen in two different ways. On one hand we could have that the protocol came into a situation where execution is still possible, but the protocol would not be able complete. Another case where no progress would be possible is if two entities expects information from each other to be able to continue execution. We will cover these two cases separately. The first kind of progress problems would be located in cycles of the state space. Cycles in the state space imply that the model can diverge between a number of states, the result is that the protocol is allowed execute without progress infinitely. The second case indicates that the protocol stops as the two entities are waiting for the other. In this case we have to look at the terminal nodes of the state space.

The marking of the terminal nodes will reveal whether the protocol finished. If the protocol did not finish in this situation, we could have an absence of progress.

The last part of the progress analysis is closely related to the outcome analysis. In the outcome analysis we also inspect the terminal nodes of the state space to see if the model of the protocol ends in a situation where the entities of the network agree on the outcome. Therefore the second part of the progress analysis will be done in association with the outcome analysis.

7.2 Analysis of progress and outcome

We follow the order from the discussion section, starting with the progress and end with the analysis of outcome. The analysis of outcome includes the last part of the progress analysis, as the analysis methods applied are related.

7.2.1 Progress of handover

In section 7.1 we came to the conclusion that situations with progress problems could be found in the cycles of the state space of the model. We can conclude absence of this kind of progress problems by looking at the statistics section of the occurrence graph report:

```
Statistics
-----
Occurrence Graph
Nodes: 3617
Arcs: 11084
Secs: 8
Status: Full

Scc Graph
Nodes: 3617
Arcs: 11084
Secs: 1
-----
```

First of all the `Status` field tells us that the state space can be completely calculated. If this was not the case we would not be able to argue about absence of cycles in the state space. Uncalculated parts of a partially calculated state space could still contain the undesired properties. The absence of the first kind of progress problems is verified by looking at the size of the occurrence graph and the strongly connected components (Scc) graph. The equality in the size of the two graphs eliminates the existence of nontrivial Scc's, i.e. Scc's with more than

one node are not present. The absence of nontrivial Scc's eliminates cycles in the state space.

7.2.2 Outcome of handover

In this section we partition the set of terminal nodes based on the entity state of the MSC. By the phrase *entity state of the MSC* we mean that a token is on one of the places representing the state of the MSC. The same is the case when we talk about the state of the other entities. We start by listing the dead markings with respect to the partitioning of the entity state of the MSC. Next we argue that the states where the MSC ends up in a consistent state, does not contain progress problems and that a consistent protocol state has been reached. This reasoning will be based on the marking of the terminal nodes. The marking of all the terminal nodes partitioned as described above can be found appendix E. Finally we look at the terminal nodes where the MSC does not end up in a consistent end state.

Results from occurrence graph report

We have partitioned the set of terminal nodes on the entity state of the MSC. The possible combinations of contents on the three places *Handover succeeded*, *Failed Back* and *Call released* give three categories by the same names and a fourth, representing that non of these states have been reached by the MSC. Note that these categories are disjoint, i.e. we do not have that both *Handover succeeded* and *Failed Back* contain a token a the same time. This has been verified by a simple function on the state space, calculating the intersections of the categories. The categories and the terminal nodes belonging to the categories are shown in table 7.1.

Category	States
Handover Succeeded	3532, 3444, 3383, 3203
Failed Back	88, 3137, 3136, 2743, 2742, 2265
Call Released	520, 3617, 3616, 3611, 3610, 3609, 3608, 3603, 3602, 3601, 3586, 3585, 3572, 3571, 3570, 357, 3569, 3554, 3553, 3476, 3470, 3465, 3464, 3459, 3458, 3457, 3253, 3235, 3229, 3224, 3223, 2851
No end state	949, 3531, 3443, 3305, 2903, 1337

Table 7.1: Categorization of terminal nodes

The first three categories should not contain any progress problems, but as shown in appendix E the model might end in states where the interfaces are not empty. Messages being left on the interfaces are due to the modelling of the communication between the entities. We model the communication by multisets, as opposed to for example lists (queues). This allows messages to overtake each other, and the result is that some messages are not received before they are outdated, and therefore left unhandled on the interfaces.

Handover Succeeded The markings of the terminal nodes in the *Handover succeeded* case as shown in figure 7.1 illustrate that the entities agree on the outcome of the handover. This conclusion is derived by looking at contents of `OldBSC'ResourcesReleased` and `SuccessfulNewBSC'HND_CMPsent`. Both of these entity state places are end states of the old BSC and new BSC respectively. The resources has been released in the old BSS, and the new BSC has sent the `HND_CMP` message to the MSC, completing the handover for the new BSC. We conclude that no progress problems are present when the handover succeeds, none of the entities expect information from other parties as well.

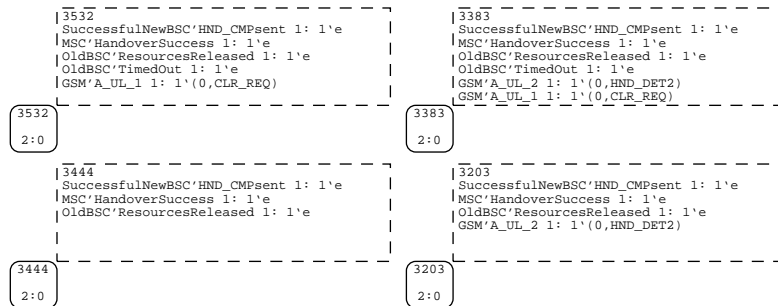


Figure 7.1: The marking of the terminal nodes when MSC is in state *Handover Succeeded*.

Failed Back The markings of the terminal nodes in the *Failed Back* case are shown in figure 7.2. Five of the terminal nodes are similar to each other with respect to the outcome; these are all except node 88. In these states the entities agree on the outcome of the handover. This can be seen by looking at `OldBSC'HandoverFailed` and `NewBSC'ResourcesReleased`. When falling back to the old BSS the handover fails, and resources are released in the new BSS, as they are not needed any more.

The state numbered 88 is a bit more interesting. The old BSC has lost radio path meanwhile the MSC continues the call on the old BSS, which is a problem. Both `LossOfRadioPath'RadioPathLost` and `MSC'Fallback` are end states of the old BSC and the MSC, respectively. The result is that normal operation would continue when reaching these states. This would allow the MSC to process the `CLR_REQ` message from the old BSC, and the outcome would be that resources are released. Based on this discussion we conclude that no progress problems are present when the call is falling back to the old channel, although one of the cases would inevitably end in a release of the call, where all entities agree on this being the result.

Call Released The category where the MSC releases the call is by far the largest, and for that reason we did not include a figure illustrating the markings; the results are included in appendix E. In all the terminal nodes of this category we end up in states where the MSC and the two BSC's agree on the outcome.

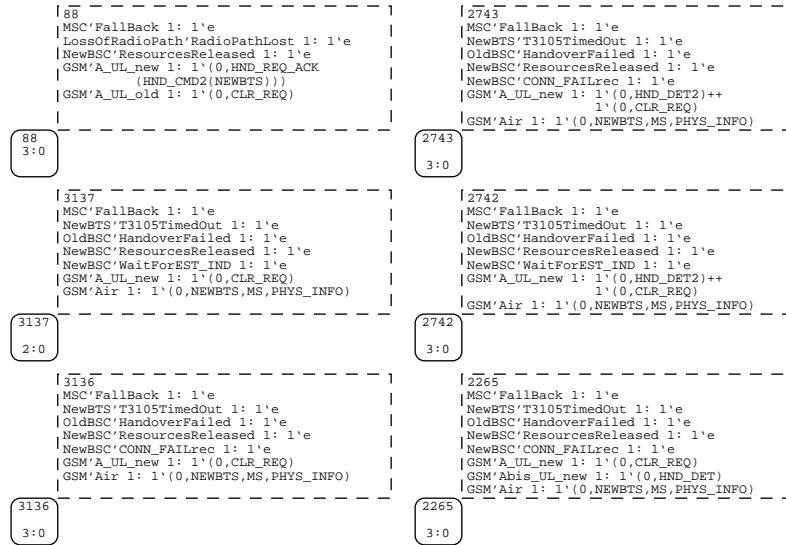


Figure 7.2: The marking of the terminal nodes when MSC is in state *FailedBack*.

We see that the places `OldBSC'ResourcesReleased` and `NewBSC'ResourcesReleased` are present in all the states, meaning that resources has been released in both BSS's and the call is implicitly Released. The different nodes are variants over the possible contents on the interfaces when ending with a call release.

No end state The category representing that the MSC does not end in a consistent state is the obvious place to look for progress problems. The markings are illustrated on figure 7.3, and all states indicate progress problems. The category can be further partitioned into two groups, the first includes node 949 and 1337, and the second the last four. The partitioning is based on the location of the problem possibly leading to an absence of progress.

In the first group we see that the problem is present on the old A-interface. The old BSC is in an end state, `OldBSC'HandoverFailed`, and we assume it will continue normal operation. The result would be that it receives the unhandled `CLR_CMD` and releases the resources, ending up in a release of the call.

In the second group the problem is present on the new A-interface. The new BSC has ended in a state, where the handover ended in success (`Successful-NewBSC'HND_CMP_sent`). This concludes the handover for the new BSC, and we assume that it continues normal operation. Normal operation would result in the reception of the `CLR_CMD`, and end up in a situation where the call is released.

7.3 Summary

In this chapter we have analysed the intra-MSC handover for progress problems and outcome agreement of the entities. The absence of progress problems have

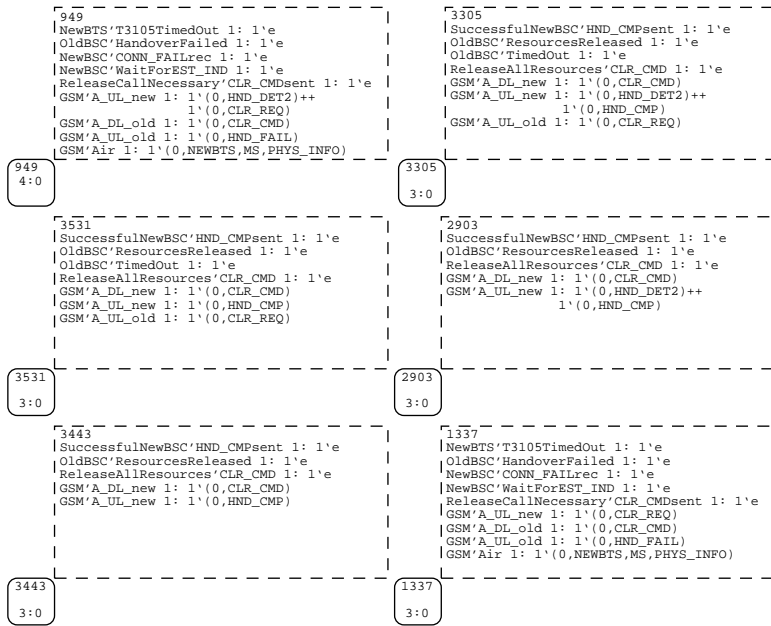


Figure 7.3: The marking of the terminal nodes when MSC does not reach an end state.

been verified by looking at the statistics section of the occurrence graph tool and the terminal nodes of the state space. The analysis of consistent outcome of the handover was also based on inspection of the terminal nodes. Some of these nodes indicate progress problems when the protocol is about to finish. We argued that these would be caught by normal operation, that takes over when handover finishes from the entities point of view. The results are valid under the limitations described in chapter 4.

Chapter 8

Future Work

Having completed this investigation of the intra-MSC handover within GSM networks, gives raise to new challenges. The obvious next step would be to look at handovers between different network standards. This is a great challenge, but also interesting. As stated in chapter 1, this were our goal before we found out how big the GSM part was. When dealing with short range wireless networks, time becomes a challenge. If the range of the network is 30 meters, the handover decision and execution must be very fast. If the person using the phone starts running, 30 meters just gives the network about 1-2 seconds the perform the handover. This, and lots of other interesting problems arise, when performing handover between different types of networks.

Chapter 9

Conclusion

In this thesis we have investigated the intra-MSD handover of GSM networks. Our work can be split into four phases: Studying and interpretation the GSM recommendations, modelling the handover, validating the model, and finally analysing the model.

The studying and interpreting phase has been the most time consuming of them all. We have searched through thousands of pages with specifications and located the parts relevant to our work. We have clarified ambiguous parts in order to have a simple, understandable, abstract, specification of the intra-MSD handover. We have also read books giving us a better overview of specific areas, but the major challenge has been putting all the pieces together to give one consistent specification of the handover.

Through the modelling we have achieved great knowledge of the GSM domain. The model has been developed in an iteratively manner, which is typical for projects, where the initial understanding of the problem domain is limited. These iterations has produced a rather abstract model, containing only the essential parts necessary for simulation of a handover. Our model gives an abstract view of the intra-MSD handover, encapsulating the relevant problems from the real systems. The simulation capabilities of Design/CPN has given us a thorough understanding of the dynamics in the handover. It is our believe that this understanding could not have been achieved by just looking at the SDLs of the recommendations.

The model has been validated through reviews by GSM experts, with whom we were able to discuss domain specific details based on our presented model. It gave us a strong feeling of validity to the model and our understanding, which lay behind. We have also generated message sequence charts for some of the major scenarios of the handover, and inspected them to find out, that they behave in a correct manner. This ensured that our model is able simulate some central scenarios, which increases our trust of validity. Beside the documented scenarios, we have simulated lots of interactive executions, in order to convince ourselves.

Finally we have used formal analysis compared with domain specific knowl-

edge to be able to argue that the intra-MSD handover is a progressing protocol and when completed, all devices share the same understanding of the outcome. This is achieved using the occurrence graph tool of Design/CPN. The results are based both on the standard occurrence graph report and inspection of terminal nodes of the state space. Our results are only valid with respect to our limitations.

Appendix A

Introduction to SDL

In order to discuss the SDLs (Specification and Description Language), we present a short introduction to the syntax and semantics. Only the parts of SDLs used in our work are presented. An SDL describes the flow of a system in a graphical way, having states, functions, selections, internal page references, and message exchanges. Between the graphical symbols are lines; they are usually arrows, but it is not required. The basic SDL symbols are shown on figure A.1.

An SDL has an initial state, typically placed in the top of the diagram. Functions are used to represent internal processing without communication. Selections are case-blocks having two or more ways out. Internal page references are used, when the flow continues in another part of the diagram. Messages being sent to an external process, e.g. a user or another device, is expressed by the send message symbol and reception of messages by the receive message symbol.

SDLs are specified in [12].

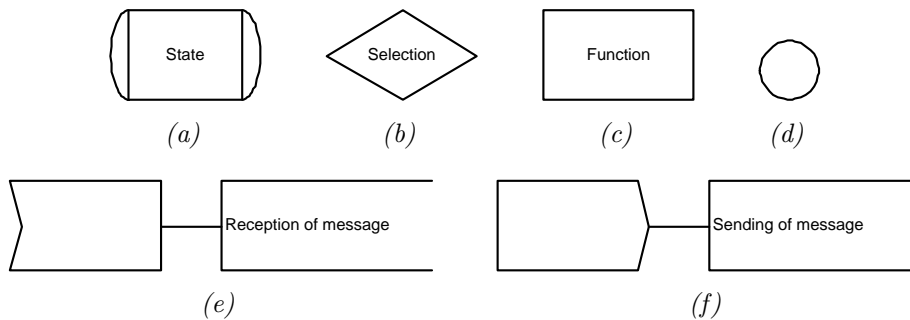


Figure A.1: The SDL symbols used in our work. Their syntactical meaning is: (a) State, (b) Selection, (c) Function, (d) Internal page reference, (e) Reception of message, and (f) Sending a message.

Appendix B

SDLs from GSM 03.09

We have included the first four sheets of the SDLs from [8] in this appendix for easy reference. They describe the behavior of the MSC during an intra-MSC handover. The rest of the 26 sheets are concerned with inter-MSC handover and are therefore excluded from this work.

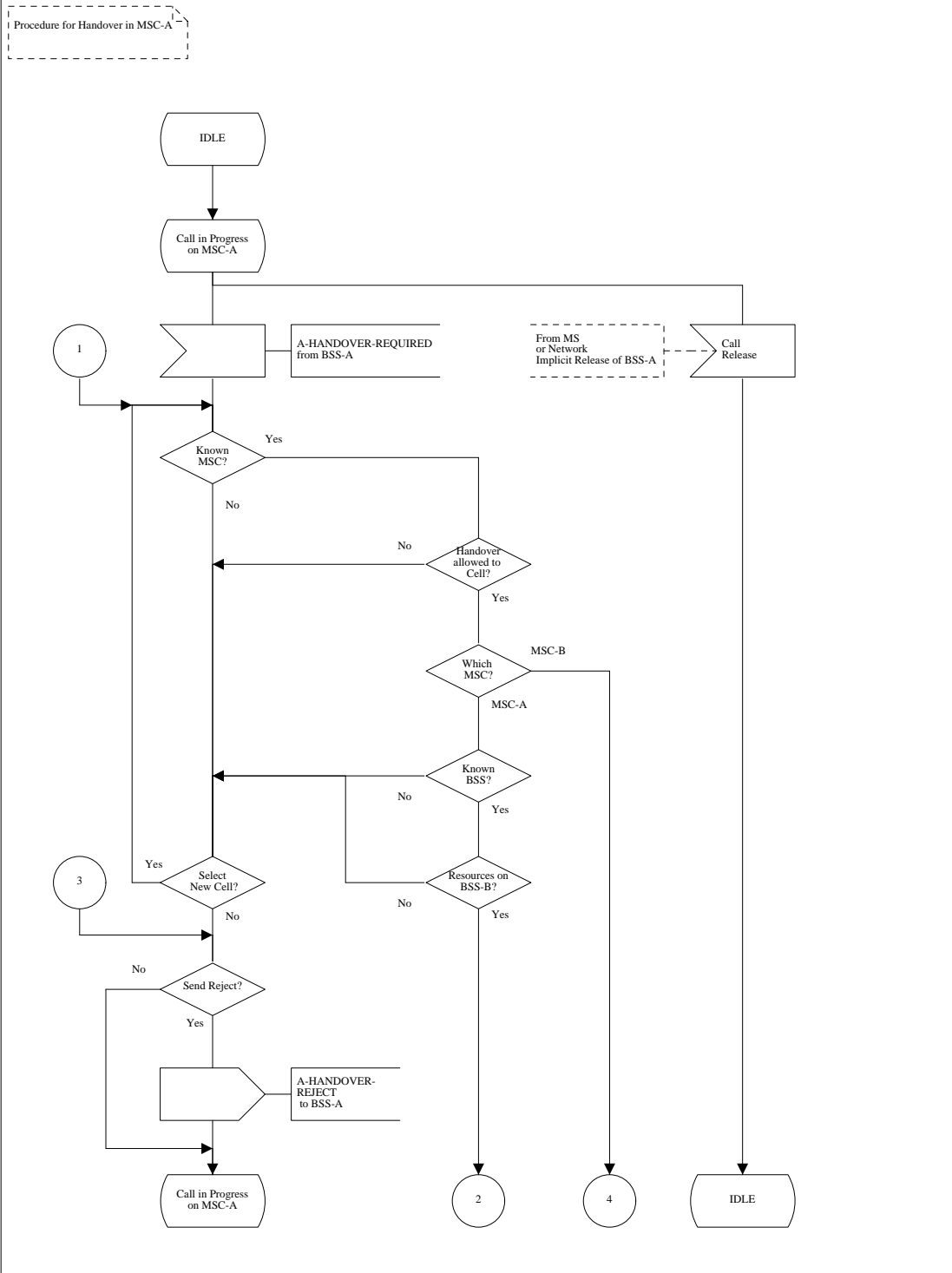


Figure B.1: SDL from GSM 03.09 ([8]), sheet 1.

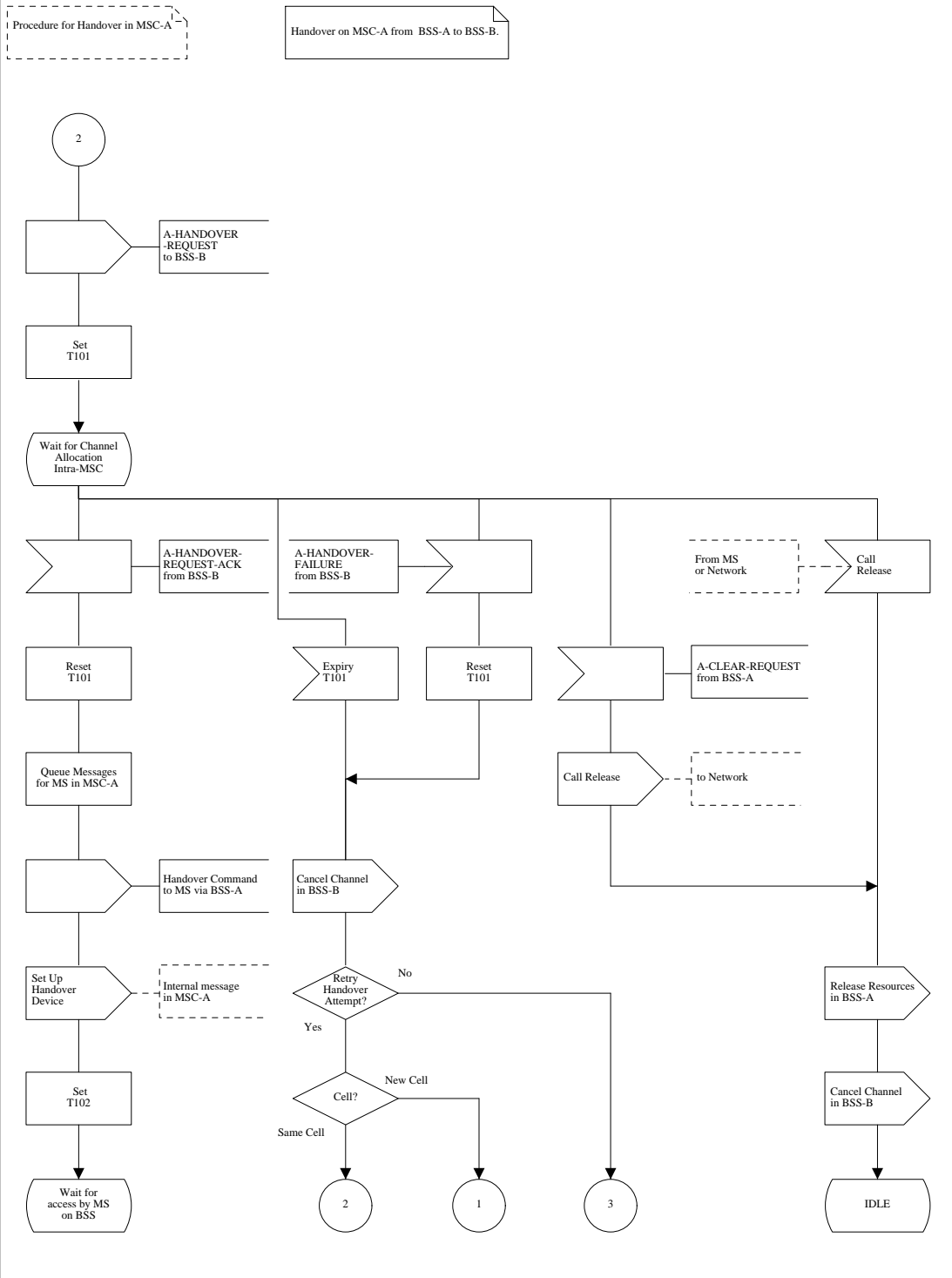


Figure B.2: SDL from GSM 03.09 ([8]), sheet 2.

Procedure for Handover in MSC-A

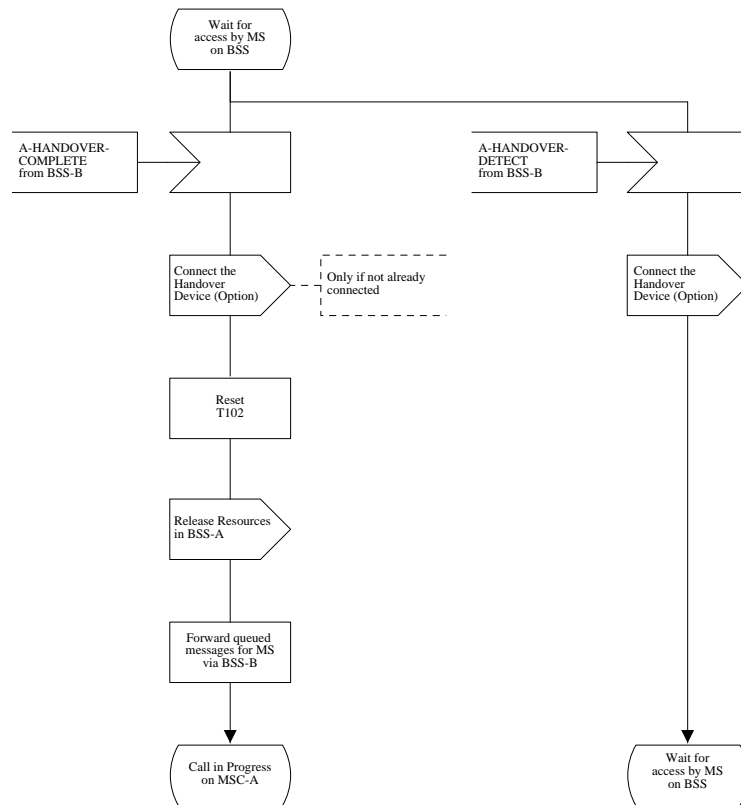


Figure B.3: SDL from GSM 03.09 ([8]), sheet 3.

Procedure for Handover in MSC-A

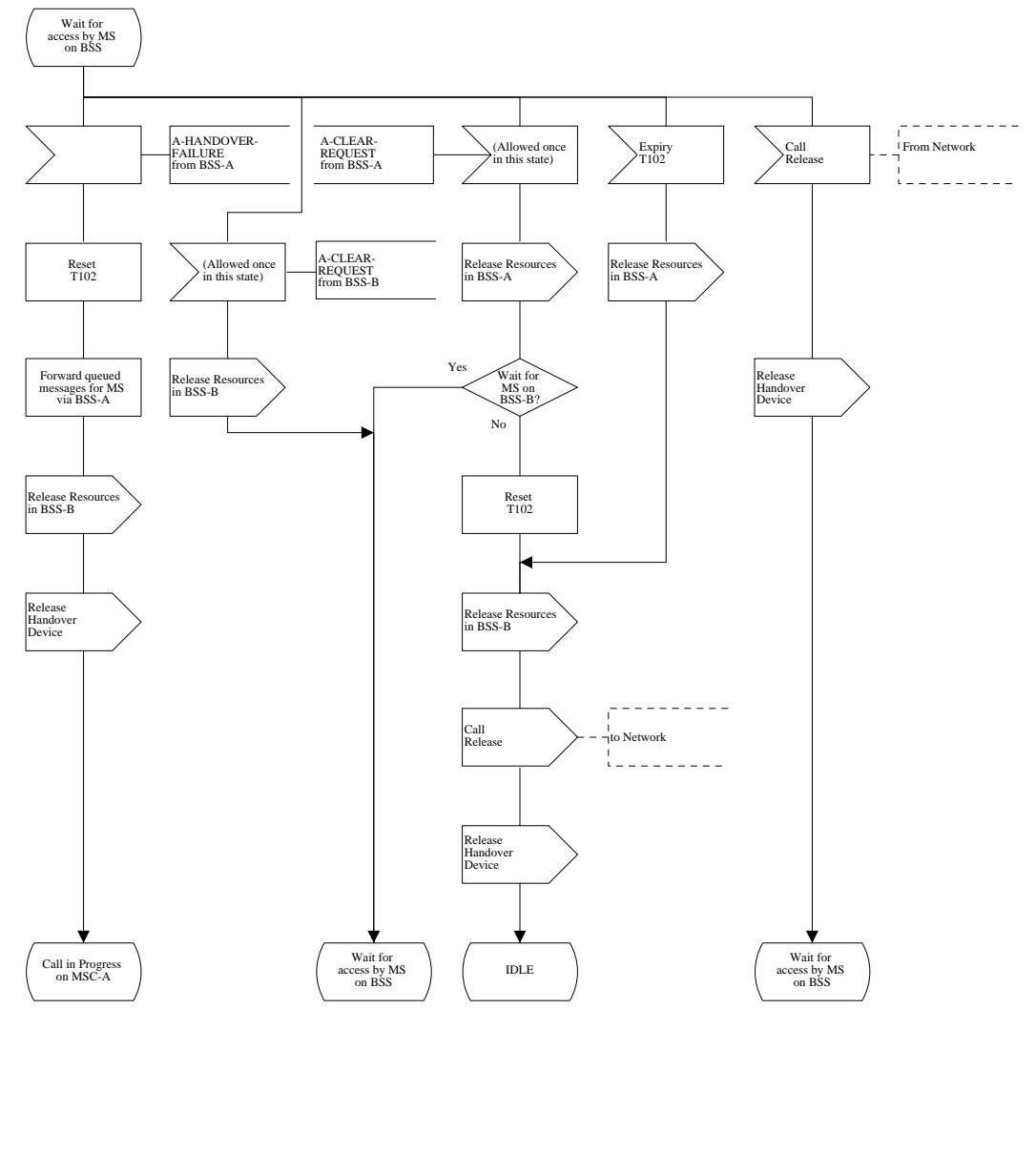
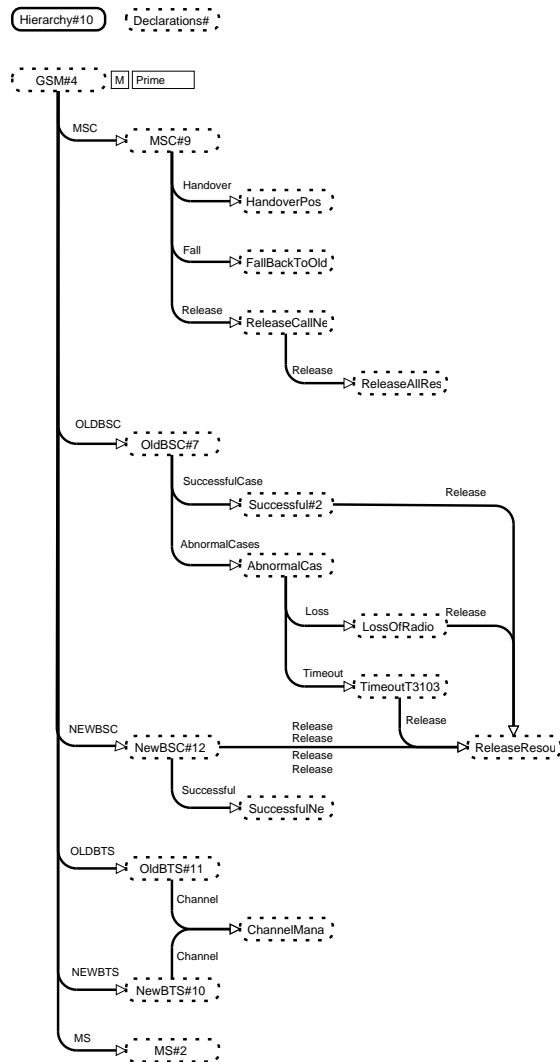


Figure B.4: SDL from GSM 03.09 ([8]), sheet 4.

Appendix C

CPN Hierarchy



Appendix D

Occurrence Graph Report

Statistics

Occurrence Graph

Nodes: 3617
Arcs: 11084
Secs: 8
Status: Full

Scc Graph

Nodes: 3617
Arcs: 11084
Secs: 1

Boundedness Properties

Best Integers Bounds	Upper	Lower
FallBackToOldBSS'CLR_CMDsent 1	1	0
GSM'A_DL_old 1	2	0
GSM'A_DL_new 1	2	0
GSM'A_UL_old 1	2	0
GSM'A_DL_new 1	3	0
GSM'Air 1	2	0
GSM'Abis_DL_old 1	2	0
GSM'Abis_DL_new 1	1	0
GSM'Abis_UL_old 1	2	0
GSM'Abis_UL_new 1	4	0
HandoverPossible'HND_CMPrec 1	1	0

LossOfRadioPath'RadioPathLost	1	0
MS'Air	2	0
MS'WaitForPHYS_INFO	1	0
MS'WaitForUA	1	0
MS'currentBTS	1	1
MS'oldBTS	1	0
MSC'CallReleased	1	0
MSC'FallBack	1	0
MSC'HandoverSuccess	1	0
MSC'WaitChanAlloc	1	0
MSC'WaitMSAccess	1	0
NewBSC'BSC_ID	1	1
NewBSC'CONN_FAILrec	1	0
NewBSC'ResourcesReleased	1	0
NewBSC'WaitForEST_IND	1	0
NewBSC'WaitForHND_COM	1	0
NewBSC'WaitForHND_DET	1	0
NewBTS'Air	2	0
NewBTS'BTS_ID	1	1
NewBTS'T3105TimedOut	1	0
NewBTS'WaitForHND_COM	1	0
NewBTS'WaitForSABM	1	0
OldBSC'BSC_ID	1	1
OldBSC'HandoverFailed	1	0
OldBSC'ResourcesReleased	1	0
OldBSC'TimedOut	1	0
OldBSC'WaitForCLR_CMD	1	0
OldBSC'WaitForHND_CMD	1	0
OldBTS'Air	2	0
OldBTS'BTS_ID	1	1
ReleaseAllResources'CLR_CMD	1	0
ReleaseCallNecessary'CLR_CMDsent	1	0

```

1 0
ReleaseResourcesBSC'WaitForREL_ACK 1 1
1 0
ReleaseResourcesBSC'WaitForREL_ACK 2 2
1 0
ReleaseResourcesBSC'WaitForREL_ACK 3 3
1 0
ReleaseResourcesBSC'WaitForREL_ACK 4 4
1 0
ReleaseResourcesBSC'WaitForREL_ACK 5 5
1 0
ReleaseResourcesBSC'WaitForREL_ACK 6 6
1 0
ReleaseResourcesBSC'WaitForREL_ACK 7 7
1 0
Successful'handoverTo 1 1
1 0
SuccessfulNewBSC'HND_CMPsent 1 1
1 0
SuccessfulNewBSC'WaitForCHAN_ACT_ACK 1 1
1 0
Timeout'WaitForCLR_CMD2 1 1
1 0

```

Best Upper Multi-set Bounds

```

FallBackToOldBSS'CLR_CMDsent 1
1'e
GSM'A_DL_old 1 1'(0,HND_CMD(HND_CMD2(NEWBTS)))+
1'(0,CLR_CMD)
GSM'A_DL_new 1 1'(0,HND_REQ(NEWBTS))++ 1'(0,CLR_CMD)
GSM'A_UL_old 1 1'(0,HND_RQD(NEWBTS))++ 1'(0,CLR_CMP)+
1'(0,HND_FAIL)+ 1'(0,CLR_REQ)
GSM'A_DL_new 1 1'(0,HND_REQ_ACK(HND_CMD2(NEWBTS)))+
1'(0,HND_DET2)+ 1'(0,HND_CMP)+
1'(0,CLR_CMP)+ 1'(0,CLR_REQ)
GSM'Air 1 1'(0,OLDBTS,MS,HND_CMD2(NEWBTS))++
1'(0,NEWBTS,MS,PHYS_INFO)+
1'(0,NEWBTS,MS,UA)+
1'(0,MS,OLDBTS,HND_FAI)+
1'(0,MS,NEWBTS,HND_ACC)+
1'(0,MS,NEWBTS,SABM)+
1'(0,MS,NEWBTS,HND_COM)
GSM'Abis_DL_old 1 1'(0,DATA_REQ(HND_CMD2(NEWBTS)))+

```

```

1'(0,RF_CHAN_REL)
GSM'Abis_DL_new 1 1'(0,CHAN_ACT)++ 1'(0,RF_CHAN_REL)
GSM'Abis_UL_old 1 1'(0,DATA_IND(HND_FAI))++
1'(0,RF_CH_REL_ACK)
GSM'Abis_UL_new 1 1'(0,CHAN_ACT_ACK)++ 1'(0,HND_DET)++
1'(0,EST_IND)++
1'(0,DATA_IND(HND_COM))++
1'(0,RF_CH_REL_ACK)++ 1'(0,CONN_FAIL)
HandoverPossible'HND_CMPrec 1
1'e
LossOfRadioPath'RadioPathLost 1
1'e
MS'Air 1 1'(0,OLDBTS,MS,HND_CMD2(NEWBTS))++
1'(0,NEWBTS,MS,PHYS_INFO)++
1'(0,NEWBTS,MS,UA)++
1'(0,MS,OLDBTS,HND_FAI)++
1'(0,MS,NEWBTS,HND_ACC)++
1'(0,MS,NEWBTS,SABM)++
1'(0,MS,NEWBTS,HND_COM)
MS'WaitForPHYS_INFO 1
1'e
MS'WaitForUA 1 1'e
MS'currentBTS 1 1'OLDBTS++ 1'NEWBTS
MS'oldBTS 1 1'OLDBTS
MSC'CallReleased 1 1'e
MSC'FallBack 1 1'e
MSC'HandoverSuccess 1
1'e
MSC'WaitChanAlloc 1 1'e
MSC'WaitMSAccess 1 1'e
NewBSC'BSC_ID 1 1'NEWBSC
NewBSC'CONN_FAILrec 1
1'e
NewBSC'ResourcesReleased 1
1'e
NewBSC'WaitForEST_IND 1
1'e
NewBSC'WaitForHND_COM 1
1'e
NewBSC'WaitForHND_DET 1
1'e
NewBTS'Air 1 1'(0,OLDBTS,MS,HND_CMD2(NEWBTS))++
1'(0,NEWBTS,MS,PHYS_INFO)++

```

```

1'(O,NEWBTS,MS,UA)++
1'(O,MS,OLDBTS,HND_FAI)++
1'(O,MS,NEWBTS,HND_ACC)++
1'(O,MS,NEWBTS,SABM)++
1'(O,MS,NEWBTS,HND_COM)
NewBTS'BTS_ID 1 1'NEWBTS
NewBTS'T3105TimedOut 1
1'e
NewBTS'WaitForHND_COM 1
1'e
NewBTS'WaitForSABM 1
1'e
OldBSC'BSC_ID 1 1'OLDBSC
OldBSC'HandoverFailed 1
1'e
OldBSC'ResourcesReleased 1
1'e
OldBSC'TimedOut 1 1'e
OldBSC'WaitForCLR_CMD 1
1'e
OldBSC'WaitForHND_CMD 1
1'e
OldBTS'Air 1 1'(O,OLDBTS,MS,HND_CMD2(NEWBTS))++
1'(O,NEWBTS,MS,PHYS_INFO)++
1'(O,NEWBTS,MS,UA)++
1'(O,MS,OLDBTS,HND_FAI)++
1'(O,MS,NEWBTS,HND_ACC)++
1'(O,MS,NEWBTS,SABM)++
1'(O,MS,NEWBTS,HND_COM)
OldBTS'BTS_ID 1 1'OLDBTS
ReleaseAllResources'CLR_CMD 1
1'e
ReleaseCallNecessary'CLR_CMDsent 1
1'e
ReleaseResourcesBSC'WaitForREL_ACK 1
1'e
ReleaseResourcesBSC'WaitForREL_ACK 2
1'e
ReleaseResourcesBSC'WaitForREL_ACK 3
1'e
ReleaseResourcesBSC'WaitForREL_ACK 4
1'e
ReleaseResourcesBSC'WaitForREL_ACK 5

```

```

1'e
ReleaseResourcesBSC'WaitForREL_ACK 6
1'e
ReleaseResourcesBSC'WaitForREL_ACK 7
1'e
Successful'handoverTo 1
1'NEWBTS
SuccessfulNewBSC'HND_CMPsent 1
1'e
SuccessfulNewBSC'WaitForCHAN_ACT_ACK 1
1'NEWBTS
Timeout'WaitForCLR_CMD2 1
1'e

Best Lower Multi-set Bounds
FallBackToOldBSS'CLR_CMDsent 1
empty
GSM'A_DL_old 1 empty
GSM'A_DL_new 1 empty
GSM'A_UL_old 1 empty
GSM'A_DL_new 1 empty
GSM'Air 1 empty
GSM'Abis_DL_old 1 empty
GSM'Abis_DL_new 1 empty
GSM'Abis_UL_old 1 empty
GSM'Abis_UL_new 1 empty
HandoverPossible'HND_CMPrec 1
empty
LossOfRadioPath'RadioPathLost 1
empty
MS'Air 1 empty
MS'WaitForPHYS_INFO 1
empty
MS'WaitForUA 1 empty
MS'currentBTS 1 empty
MS'oldBTS 1 empty
MSC'CallReleased 1 empty
MSC'FallBack 1 empty
MSC'HandoverSuccess 1
empty
MSC'WaitChanAlloc 1 empty
MSC'WaitMSAccess 1 empty
NewBSC'BSC_ID 1 1'NEWBSC

```

NewBSC'CONN_FAILrec 1
 empty
 NewBSC'ResourcesReleased 1
 empty
 NewBSC'WaitForEST_IND 1
 empty
 NewBSC'WaitForHND_COM 1
 empty
 NewBSC'WaitForHND_DET 1
 empty
 NewBTS'Air 1 empty
 NewBTS'BTS_ID 1 1'NEWBTS
 NewBTS'T3105TimedOut 1
 empty
 NewBTS'WaitForHND_COM 1
 empty
 NewBTS'WaitForSABM 1
 empty
 OldBSC'BSC_ID 1 1'OLDBSC
 OldBSC'HandoverFailed 1
 empty
 OldBSC'ResourcesReleased 1
 empty
 OldBSC'TimedOut 1 empty
 OldBSC'WaitForCLR_CMD 1
 empty
 OldBSC'WaitForHND_CMD 1
 empty
 OldBTS'Air 1 empty
 OldBTS'BTS_ID 1 1'OLDBTS
 ReleaseAllResources'CLR_CMD 1
 empty
 ReleaseCallNecessary'CLR_CMDsent 1
 empty
 ReleaseResourcesBSC'WaitForREL_ACK 1
 empty
 ReleaseResourcesBSC'WaitForREL_ACK 2
 empty
 ReleaseResourcesBSC'WaitForREL_ACK 3
 empty
 ReleaseResourcesBSC'WaitForREL_ACK 4
 empty
 ReleaseResourcesBSC'WaitForREL_ACK 5

empty
ReleaseResourcesBSC'WaitForREL_ACK 6
empty
ReleaseResourcesBSC'WaitForREL_ACK 7
empty
Successful'handoverTo 1
empty
SuccessfulNewBSC'HND_CMPsent 1
empty
SuccessfulNewBSC'WaitForCHAN_ACT_ACK 1
empty
Timeout'WaitForCLR_CMD2 1
empty

Home Properties

Home Markings: None

Liveness Properties

Dead Markings: 48 [949,88,520,3617,3616,...]

Dead Transitions Instances:

ChannelManagement'recCHAN_ACT 2

Live Transitions Instances: None

Fairness Properties

No infinite occurrence sequences.

Appendix E

Terminal Nodes of the state space

E.1 HandoverSucceeded

3532

SuccessfulNewBSC'HND_CMPsent 1: 1'e
MSC'HandoverSuccess 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
GSM'A_UL_old 1: 1'(0,CLR_REQ)

3444

SuccessfulNewBSC'HND_CMPsent 1: 1'e
MSC'HandoverSuccess 1: 1'e
OldBSC'ResourcesReleased 1: 1'e

3383

SuccessfulNewBSC'HND_CMPsent 1: 1'e
MSC'HandoverSuccess 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)

3203

SuccessfulNewBSC'HND_CMPsent 1: 1'e
MSC'HandoverSuccess 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
GSM'A_UL_new 1: 1'(0,HND_DET2)

E.2 FailedBack

88

```
MSC'FallBack 1: 1'e
LossOfRadioPath'RadioPathLost 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_REQ_ACK(HND_CMD2(NEWBTS)))
```

3137

```
MSC'FallBack 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'A_UL_new 1: 1'(0,CLR_REQ)
```

3136

```
MSC'FallBack 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'A_UL_new 1: 1'(0,CLR_REQ)
```

2743

```
MSC'FallBack 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)
```

2742

```
MSC'FallBack 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
```

GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

2265

MSC'FallBack 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_new 1: 1'(0,HND_DET)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

E.3 CallReleased

520

MSC'CallReleased 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_DL_old 1: 1'(0,HND_CMD(HND_CMD2(NEWBTS)))

3617

MSC'CallReleased 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,DATA_IND(HND_COM))
GSM'A_UL_old 1: 1'(0,CLR_REQ)

3616

MSC'CallReleased 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,DATA_IND(HND_COM))

3611

MSC'CallReleased 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,EST_IND)++ 1'(0,DATA_IND(HND_COM))
GSM'A_UL_old 1: 1'(0,CLR_REQ)

3610

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3609

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3608

MSC'CallReleased 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,DATA_IND(HND_COM))
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)

3603

MSC'CallReleased 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,EST_IND)++ 1'(0,DATA_IND(HND_COM))

3602

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e

OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3601

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3586

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3585

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3572

MSC'CallReleased 1: 1'e

OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,EST_IND)++ 1'(0,DATA_IND(HND_COM))
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)

3571

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3570

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

357

MSC'CallReleased 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'A_UL_new 1: 1'(0,HND_REQ_ACK(HND_CMD2(NEWBTS)))

3569

MSC'CallReleased 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,DATA_IND(HND_COM))
GSM'A_UL_new 1: 1'(0,HND_DET2)

3554

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3553

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3476

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3470

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_old 1: 1'(0,CLR_REQ)

GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3465

MSC'CallReleased 1: 1'e

OldBSC'TimedOut 1: 1'e

OldBSC'ResourcesReleased 1: 1'e

NewBSC'ResourcesReleased 1: 1'e

GSM'Abis_UL_new 1: 1'(0,HND_DET)++ 1'(0,EST_IND)++ 1'(0,DATA_IND(HND_COM))

GSM'A_UL_old 1: 1'(0,CLR_REQ)

3464

MSC'CallReleased 1: 1'e

NewBTS'T3105TimedOut 1: 1'e

OldBSC'TimedOut 1: 1'e

OldBSC'ResourcesReleased 1: 1'e

NewBSC'ResourcesReleased 1: 1'e

NewBSC'CONN_FAILrec 1: 1'e

GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)

GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))

GSM'Abis_UL_new 1: 1'(0,HND_DET)

GSM'A_UL_old 1: 1'(0,CLR_REQ)

GSM'A_UL_new 1: 1'(0,CLR_REQ)

3459

MSC'CallReleased 1: 1'e

OldBSC'ResourcesReleased 1: 1'e

NewBSC'ResourcesReleased 1: 1'e

GSM'Abis_UL_new 1: 1'(0,EST_IND)++ 1'(0,DATA_IND(HND_COM))

GSM'A_UL_new 1: 1'(0,HND_DET2)

3458

MSC'CallReleased 1: 1'e

NewBTS'T3105TimedOut 1: 1'e

OldBSC'ResourcesReleased 1: 1'e

NewBSC'ResourcesReleased 1: 1'e

NewBSC'CONN_FAILrec 1: 1'e

GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)

GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))

GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3457

MSC'CallReleased 1: 1'e

NewBTS'T3105TimedOut 1: 1'e

OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3253

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'Abis_UL_new 1: 1'(0,HND_DET)
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3235

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3223

MSC'CallReleased 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'Abis_UL_old 1: 1'(0,DATA_IND(HND_FAI))
GSM'Abis_UL_new 1: 1'(0,HND_DET)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

2851

MSC'CallReleased 1: 1'e

MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'Abis_UL_new 1: 1'(0,HND_DET)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

3229

MSC'CallReleased 1: 1'e
MS'WaitForUA 1: 1'e
NewBTS'T3105TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Air 1: 1'(0,MS,NEWBTS,SABM)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3224

MSC'CallReleased 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
NewBSC'ResourcesReleased 1: 1'e
GSM'Abis_UL_new 1: 1'(0,HND_DET)++ 1'(0,EST_IND)++ 1'(0,DATA_IND(HND_COM))

E.4 NoEndState

949

NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
ReleaseCallNecessary'CLR_CMDsent 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'A_UL_old 1: 1'(0,HND_FAIL)
GSM'A_DL_old 1: 1'(0,CLR_CMD)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,CLR_REQ)

3531

SuccessfulNewBSC'HND_CMPsent 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e

ReleaseAllResources'CLR_CMD 1: 1'e
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_CMP)
GSM'A_DL_new 1: 1'(0,CLR_CMD)

3443

SuccessfulNewBSC'HND_CMPsent 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
ReleaseAllResources'CLR_CMD 1: 1'e
GSM'A_UL_new 1: 1'(0,HND_CMP)
GSM'A_DL_new 1: 1'(0,CLR_CMD)

3305

SuccessfulNewBSC'HND_CMPsent 1: 1'e
OldBSC'TimedOut 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
ReleaseAllResources'CLR_CMD 1: 1'e
GSM'A_UL_old 1: 1'(0,CLR_REQ)
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,HND_CMP)
GSM'A_DL_new 1: 1'(0,CLR_CMD)

2903

SuccessfulNewBSC'HND_CMPsent 1: 1'e
OldBSC'ResourcesReleased 1: 1'e
ReleaseAllResources'CLR_CMD 1: 1'e
GSM'A_UL_new 1: 1'(0,HND_DET2)++ 1'(0,HND_CMP)
GSM'A_DL_new 1: 1'(0,CLR_CMD)

1337

NewBTS'T3105TimedOut 1: 1'e
OldBSC'HandoverFailed 1: 1'e
NewBSC'WaitForEST_IND 1: 1'e
NewBSC'CONN_FAILrec 1: 1'e
ReleaseCallNecessary'CLR_CMDsent 1: 1'e
GSM'Air 1: 1'(0,NEWBTS,MS,PHYS_INFO)
GSM'A_UL_old 1: 1'(0,HND_FAIL)
GSM'A_DL_old 1: 1'(0,CLR_CMD)
GSM'A_UL_new 1: 1'(0,CLR_REQ)

Bibliography

- [1] Ericsson Radio Systems AB. *GSM System Survey - Student Text*. EN/LZT 123 3321 R2C. Ericsson Radio Systems AB, 1999.
- [2] CPN Group at Aarhus University. <http://www.daimi.au.dk/designcpn>.
- [3] Gunnar Heine. *GSM Networks: Protocols, Terminology, and Implementation*. Mobile Communications series. Artech House Publishers, 1999.
- [4] ETSI European Telecommunications Standards Institute. Gsm 01.02 (etr 099): "european digital cellular telecommunications system (phase 2); general description of a gsm public land mobile network (plmn)", October 1993.
- [5] ETSI European Telecommunications Standards Institute. Gsm 04.04 (ets 300 553): "european digital cellular telecommunications system (phase 2); layer 1 general requirements", version 4.0.4, September 1994.
- [6] ETSI European Telecommunications Standards Institute. Gsm 04.07 (ets 300 556): "european digital cellular telecommunications system (phase 2); mobile radio interface signalling layer 3 general aspects", version 4.3.1, February 1995.
- [7] ETSI European Telecommunications Standards Institute. Gsm 08.58 (ets 300 596): "digital cellular telecommunications system (phase 2); base station controller - base transceiver station (bsc - bts) interface. layer 3 specification", version 4.9.0, November 1995.
- [8] ETSI European Telecommunications Standards Institute. Gsm 03.09 (ets 300 527): "digital cellular telecommunications system (phase 2); handover procedures", version 4.6.0, September 1996.
- [9] ETSI European Telecommunications Standards Institute. Gsm 08.08 (ets 300 590): "digital cellular telecommunications system (phase 2); mobile switching centre - base station system (msc - bsc) interface. layer 3 specification", version 4.12.1, October 1998.

- [10] ETSI European Telecommunications Standards Institute. Gsm 04.08 (ets 300 557): "digital cellular telecommunications system (phase 2); mobile radio interface layer 3 specification", version 4.23.1, October 1999.
- [11] ETSI European Telecommunications Standards Institute. Gsm 09.02 (ets 300 599): "digital cellular telecommunications system (phase 2); mobile application part (map) specification", version 4.19.1, December 2000.
- [12] ITU-T. Itu-t recommendation z.100, specification and description language (sdl), November 1999.
- [13] Telecom Standardization ITU-T International Telecommunication Union. I.xxx - integrated services digital network.
- [14] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts*. Monographs in Theoretical Computer Science. Springer-Verlag, 1992.
- [15] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 2, Analysis Methods*. Monographs in Theoretical Computer Science. Springer-Verlag, 1994.
- [16] Kurt Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 3, Practical Use*. Monographs in Theoretical Computer Science. Springer-Verlag, 1997.
- [17] Department of Computer Science The Maria group at Helsinki University of Technology and Laboratory for Theoretical Computer Science Engineering. <http://www.tcs.hut.fi/maria/>.
- [18] ITU International Telecommunication Union. X.200 - open systems interconnection reference model.