GSM General Packet Radio Services

Test Specification
RR-GPRS

Confidential

Author: Condat AG
Alt-Moabit 91a
D-10559 Berlin
Germany

Date: October 25, 2001
Document No.: 8441.xxx.99.001
File: RRG.DOC

Table of Contents

0 Document Control 5
0.1 Document History 5
0.2 References 6
0.3 Abbreviations 8
0.4 Terms 11

1 Overview 12
1.1 GRR (RLC/MAC) – Radio Link Control/Medium Access Control 12
1.2 LLC – Logical Link Control 13
1.3 GMM – GPRS Mobility Management 13
1.4 SM – Session Management 13
1.5 SNDCP - Subnetwork Dependant Convergence Protocol 13
1.6 GACI – GPRS AT Command Interpreter 13
1.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver 13
1.8 TOM – Tunnelling of Messages 13

2 Parameters 14
2.1 Declarations 14
2.2 Fields 20
2.3 Constants 31
2.4 Arrays 34
2.5 Structures 41
2.6 Bitbuffer 77

3 TEST CASES 79
3.1 Setup 79
3.1.1 RRG000: Setup the routing and PCO view for the RR test 79
3.2 Cell Selection (100) 80
3.2.1 RRG101: Activation And Cell Selection 80
3.2.2 RRG102: GPRS Indication - Going to IDLE-Mode 84
3.2.3 RRG103: Cell Selection – SI13 on EBCCH 87
3.2.4 RRG104: Cell Selection – GPRS possible but not wanted 92
3.2.5 RRG105: Cell Selection – limited service 96
3.2.6 RRG106: GPRS Indication - Going to IDLE-Mode PBCCH 100
3.3 Packet Idle Mode Procedures (200) 106
3.3.1 RRG201: Listening to Neighbour Cell 106
3.3.2 RRG202: Updating System Information 109
3.3.3 RRG203: Reselection no GPRS 111
3.3.4 RRG204: Cell reselection (start) 112
3.3.5 RRG205: Cell reselection (completion) - the new cell supports GPRS 115
3.3.6 RRG206: Cell reselection (completion) - the new cell doesn't support GPRS 119
3.3.7 RRG210: Extended Measurement initialisation 121
3.3.8 RRG212: Extended Measurement type 1 completion 123
3.3.9 RRG213: Extended Measurement type 1 completion (short list) 125
3.3.10 RRG214: Extended Measurement type 2 completion (short result) 127
3.3.11 RRG215: Extended Measurement type 2 completion (full result) 130
3.3.12 RRG216: Extended Measurement type 2 completion (short list) 133
3.3.13 RRG217: Extended Measurement type 3 completion 135
3.3.14 RRG220: Extended Measurement - premature end (early) 136
3.3.15 RRG221: Extended Measurement - premature end (middle) 137
3.3.16 RRG222: Extended Measurement - premature end (late) 139
3.3.17 RRG225: Extended Measurement interruption by cell reselection (preparation) 141
3.3.18 RRG226: Extended Measurement interruption by cell reselection (early) 142
3.3.19 RRG227: Extended Measurement interruption by cell reselection (middle) 144
3.3.20 RRG228: Extended Measurement interruption by cell reselection (late) 146

3.3.21 RRG230: Extended Measurement request while in cell reselection 149
3.3.22 RRG248: Ncell Synchronization requests from GRR in PIM_PBCCH, empty 150
3.3.23 RRG249: Ncell Synchronization requests from GRR in PIM_PBCCH, full 151
3.3.24 RRG250: Ncell Synchronization requests from GRR in PIM_PBCCH 152
3.3.25 RRG251: RRGRR_CR_REQ (CR_NEW) 153
3.3.26 RRG252: RRGRR_CR_REQ (CR_NEW_NOT_SYNCED) 154
3.3.27 RRG253: RRGRR_SYNC_REQ 155
3.3.28 RRG254: RRGRR_CR_REQ (CR_NEW) followed by CR_CS 156
3.3.29 RRG261: Start by MM for full service(HPLMN), successful, with bcch info 157
3.3.30 RRG262: Restart by RR for full service (HPLMN) is successful 162
3.3.31 RRG263: Restart by RR for full service (HPLMN) fails 169
3.3.32 RRG264: Start by MM for full service, successful, with bcch info, with PBCCH 170
3.3.33 RRG265: Restart by RR for full service (HPLMN) is successful, PBCCH 174
3.3.34 RRG266: Restart by RR for full service (HPLMN) fails 181
3.4 Packet Paging (300) 183
3.4.1 RRG301: RRGRR_PACKET_PAGING_IND 183
3.4.2 RRG302: Channel Request after Paging – IA failed 185
3.4.3 RRG304: Channel Request after Paging – IA success 186
3.5 Packet Access (400) 187
3.5.1 RRG401: Immediate assignment for PDCH / success 187
3.5.2 RRG402: Immediate assignment for SDCCH / success 190
3.5.3 RRG405: Two-message assignment 195
3.5.4 RRG406: Immediate Assignment not received 197
3.5.5 RRG407: Wrong immediate assigment received 198
3.5.6 RRG408: Immediate assignment is incomplete 200
3.5.7 RRG409: Immediate assignment – RLC/MAC failure 202
3.5.8 RRG410: Immediate assignment rejection 202
3.5.9 RRG411: Immediate assignment for SDCCH fails 204
3.5.10 RRG412: Packet Access with PBCCH 206
3.5.11 RRG413: new ncell list in PAM, PBCCH 206
3.5.12 RRG414: Left Blank 2 208
3.5.13 RRG415: Two-message assignment fails 209
3.6 Packet Downlink Assignment Procedure (500) 211
3.6.1 RRG501: Downlink assignment 211
3.6.2 RRG502: Two-message downlink assignment 213
3.7 Dedicated Mode Procedures related to GPRS (600) 216
3.7.1 RRG601: DL_ESTABLISH_REQ (RR_Init) 216
3.7.2 RRG602: RRGRR_DATA_IND 217
3.7.3 RRG605: RRGRR_PACKET_PAGING_IND 219
3.7.4 RRG610: Start of PDCH Assignment in dedicated mode 220
3.7.5 RRG620: Receive of PDCH Assignment 223
3.7.6 RRG622: PDCH Assignment: GRR suspends the dedicated mode 225
3.7.7 RRG624: PDCH Assignment: GRR stops monitoring - PBCCH present 226
3.7.8 RRG625: PDCH Assignment: GRR stops monitoring - No PBCCH present 227
3.7.9 RRG626: PDCH Assignment: GRR indicates establishment of TBF 228
3.7.10 RRG628: PDCH Assignment: GRR indicates release of TBF 230
3.7.11 RRG630: PDCH Assignment: GRR may not accept PDCH assignment 231
3.7.12 RRG632: PDCH Assignment: Timeout during TBF establishment 232
3.7.13 RRG634: PDCH Assignment: Successful Reconnection 233
3.7.14 RRG636: PDCH Assignment: Reconnection Failure 234
3.7.15 RRG640: CCO: Receive of Network Commanded Cell Change Order 235
3.7.16 RRG641: CCO: Receive Sys Info on new cell 236
3.7.17 RRG642: CCO: Successful Immediate Assignment 239
3.7.18 RRG643: CCO: Immediate Assignment rejected 241
3.7.19 RRG644: CCO: Failure, start Reconnection 242

3.7.20 RRG645: CCO: Cannot access new cell, start Reconnection 243
3.7.21 RRG646: CCO: Failure, successful Reconnection 244
3.7.22 RRG647: CCO: Failure, reconnection successful after failed access to new cell 245
3.7.23 RRG648: CCO: Failure, reconnection fails after failed access to new cell 245
3.7.24 RRG649: CCO: Failure, unsuccessful Reconnection 247
3.8 Mobile Terminated Calls (700) 248
3.8.1 RRG701: RRGRR_RR_EST_REQ 248
3.8.2 RRG702: MT CS call during GPRS – without suspension request 250
3.8.3 RRG703: Early Classmark Change 252
3.8.4 RRG704: MT CS call during GPRS – with suspension request 253
3.8.5 RRG705: Channel release after 871/2 254
3.8.6 RRG706: RRGRR_RR_EST_IND 255
3.8.7 RRG707: RRGRR_RR_EST_IND 260
3.8.8 RRG708: RRGRR_PACKET_PAGING_IND 261
3.9 Mobile Originated Calls (800) 263
3.9.1 RRG801: RRGRR_RR_EST_IND 263
3.9.2 RRG802: RRGRR_RR_EST_IND – many pagings 264
3.9.3 RRG803: RRGRR_ACTIVATE_REQ – MO CS Call – with suspension request 267
3.9.4 RRG804: RRGRR_RR_EST_RSP – after many pagings 270
3.9.5 RRG805: RRGRR_RR_EST_RSP 271
3.9.6 RRG806: No GPRS Resumption after MO CS Call 272
3.9.7 RRG807: GPRS Resumption after MO CS Call 274

0 Document Control

© Copyright Condat AG, 2001.
All rights reserved.
Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. Condat AG reserves the right to change the specification of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of Condat AG. Condat AG accepts no liability for any loss or damage arising from the use of any information contained in this document.
The software described in this document is furnished under a licence agreement and may be used or copied only in accordance with the terms of the agreement. It is an offence to copy the software in any way except as specifically set out in the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Condat AG.

Condat AG
Alt Moabit 91a
10559 Berlin
Germany

Internet: http://www.condat.de
E-mail: gsm@condat.de

0.1 Document History

Document Id.	Date	Author	Remarks
8441.xxx.99.001	16 May 1999	Joachim Pilz	Initial
8441.xxx.99.001	26-Feb-02	OT	Adaptations for AMR implementation

0.2 References

[1] GSM 05.02 version 8.0.0 Release 1999
Digital cellular telecommunications system (Phase 2+);
Multiplexing and multiple access on the radio path

[2] GSM 04.60 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station (MS) - Base Station System (BSS) interface;
Radio Link Control/ Medium Access Control (RLC/MAC) protocol

[3] GSM 04.08 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface layer 3 specification

[4] GSM 03.64 version 6.1.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Overall description of the GPRS radio interface; Stage 2

[5] GSM 03.60 version 6.3.1 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Service description; Stage 2

[6] GSM 04.07 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface signalling layer 3; General aspects

[7] GSM 04.64 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station - Serving GPRS Support Node (MS-SGSN)
Logical Link Control (LLC) layer specification

[8] GSM 05.08 version 6.4.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem link control

[9] GSM 05.10 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem synchronization

[10] GSM 03.20 TS 100 929: July 1998 (GSM 03.20 version 6.0.1)
Security related network functions, ETSI

[11] Draft GSM 03.22: August 1998 (GSM 03.22 version 6.1.0)
Functions related to Mobile Station (MS) in idle mode and group receive mode, ETSI

[12] GSM 04.65 V6.3.0: Subnetwork Dependant Convergence Protocol
ETSI, March 1999

[13] ITU-T V42bis ITU-T, Recommendation V.42 bis 1990

[14] GSM 09.60 GPRS Tunneling Protocol (GTP) across the Gn and Gp Interface

[15] RFC 1661 IETF STD 51 July 1994
The Point-to-Point Protocol (PPP)

[16] RFC 1662 IETF STD 51 July 1994
PPP in HDLC-like Framing

[17] RFC 1570 January 1994
PPP LCP Extensions

[18] RFC 1989 August 1996
PPP Link Quality Monitoring

[19] RFC 1332 May 1992
The PPP Internet Protocol Control Protocol (IPCP)

[20] RFC 1877 December 1995
PPP IPCP Extensions for Name Server Addresses

[21] RFC 2153 May 1997
PPP Vendor Extensions

[22] RFC 1334 October 1992
PPP Authentication Protocols (for Password Authentication Protocol only)

[23] RFC 1994 August 1996
PPP Challenge Handshake Authentication Protocol (CHAP)

0.3 Abbreviations

AGCH	Access Grant Channel
AT	Attention sequence "AT" to indicate valid commands of the ACI
BCCH	Broadcast Control Channel
BS	Base Station
BSIC	Base Station Identification Code
C/R	Command/Response
C1	Path Loss Criterion
C2	Reselection Criterion
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CHAP	Challenge Handshake Authentication Protocol
CKSN	Ciphering Key Sequence Number
CRC	Cyclic Redundancy Check
DCCH	Dedicated Control Channel
DCOMP	Identifier of the user data compression algorithm used for the N-DPU
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
DTX	Discontinuous Transmission
E	Extension bit
EA	Extension Bit Address Field
EL	Extension Bit Length Field
EMMI	Electrical Man Machine Interface
F	Final Bit
FACCH	Fast Associated Control Channel
FHO	Forced Handover
GACI	GPRS AT Command Interpreter
GMM	GPRS Mobility Management
GP	Guard Period
GRR	GPRS RR
GSM	Global System for Mobile Communication
HDLC	High-level Data Link Control
HISR	High level Interrupt Service Routine
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
ITU	International Telecommunication Union
IWF	Interworking Function
Kc	Ciphering Key
L	Length Indicator
LAI	Location Area Information

LCP	Link Control Protocol
LISR	Low level Interrupt Service Routine
LLC	Logical Link Control
LPD	Link Protocol Discriminator
LQM	Link Quality Monitoring
M	More bit used to indicate the last segment of N-DPU
MAC	Medium Access Control
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MT	Mobile Termination
N(R)	Receive Number
N(S)	Send Number
NC	Network Control
NCC	National Colour Code
NCP	Network Control Protocol
NECI	New Establishment Causes included
N-PDU	Network Protocol Data Unit
NSAPI	Network Layer Service Access Point Identifier
OTD	Observed Time Difference
P	Poll Bit
P/F	Poll/Final Bit
PACCH	Packet Associated Control Channel
PAP	Password Authentication Protocol
PBCCH	Packet BCCH
PCCCH	Packet CCCH
PCOMP	Identifier of the protocol control information compression algorithm used for the N-DPU
PDCH	Packet Data Channel
PDP	Packet Data Protocol e.g. IP or X.25
PDTCH	Packet Data Traffic Channel
PRACH	Packet RACH
PSI	Packet System Information
PCH	Paging Channel
PCO	Point of Control and Observation
PDU	Protocol Data Unit
PL	Physical Layer
PLMN	Public Land Mobile Network
PPC	Packet Physical Convergence
PPP	Point-to-Point Protocol
PTP	Point to Point
QoS	Quality of Service
RACH	Random Access Channel
REJ	Reject Frame
RLC	Radio Link Control
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
RTOS	Real Time Operating System

SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point
SAPI	Service Access Point Identifier
SDCCH	Stand alone Dedicated Control Channel
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SM	Session Management
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SNDCP	Subnetwork Dependant Convergence Protocol
SNSM	SNDCP-SM
SS	Supplementary Services
TAP	Test Application Program
TBF	Temporary Block Flow
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TE	Terminal Equipment - e. g. a PC
TFI	Temporary Flow Identifier
TLLI	Temporary Logical Link Identifier
TMSI	Temporary Mobile Subscriber Identity
TOM	Tunnelling of Messages
TQI	Temporary Queuing Identifier
UA	Unnumbered Acknowledgement Frame
UART	Universal Asynchronous Receiver Transmitter
UI	Unnumbered Information Frame
USF	Uplink State Flag
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable
VPLMN	Visited Public Land Mobile Network

0.4 Terms

Entity:	Program which executes the functions of a layer
Message:	A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.
Primitive:	A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.
Service Access Point:	A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

1 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation.

The protocol stack for GPRS consists of several entities. Each entity has one ore more service access points, over which the entity provides a service for the upper entity.

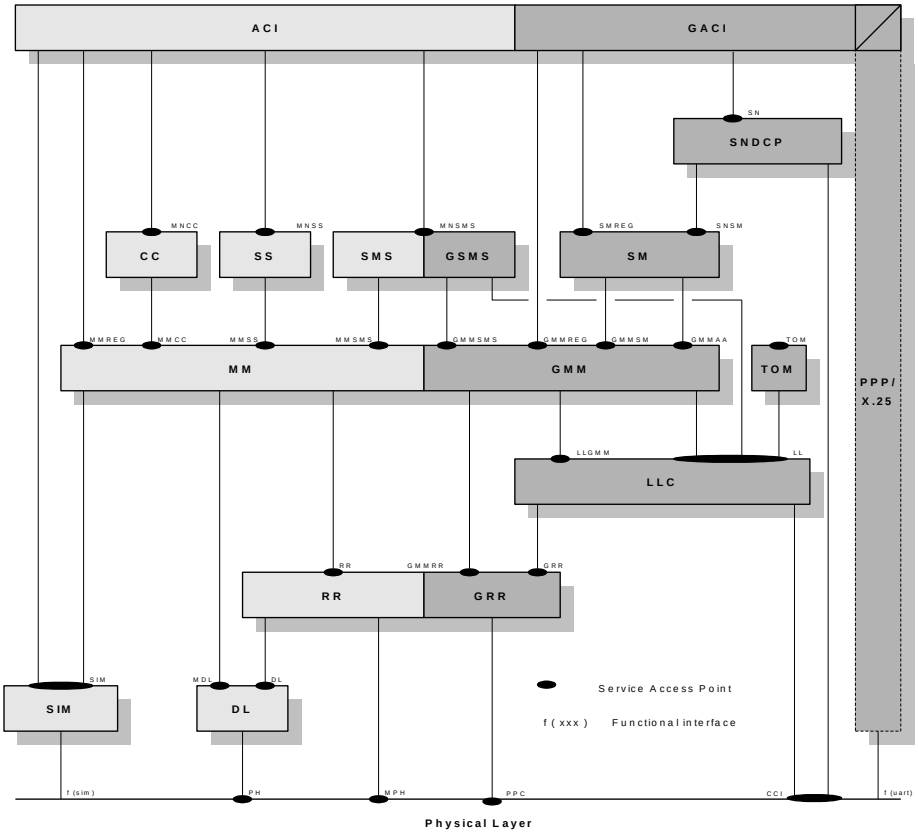


Figure 1-1: Architecture of the GSM/GPRS protocol stack

The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

The entities of the GPRS protocol stack are:

1.1 GRR (RLC/MAC) – Radio Link Control/Medium Access Control

This layer contains two functions: The Radio Link Control function provides a radio-solution-dependent reliable link. The Medium Access Control function controls the access signalling (request and grant) procedures for the radio channel, and the mapping of LLC frames onto the GSM physical channel.

1.2 LLC – Logical Link Control

The LLC entity provides multiple highly reliable logical links for asynchronous data transfer between the MS and the network. It supports variable-length information frames, acknowledged and unacknowledged data transfer, flow and sequence control, error detection and recovery, notification of unrecoverable errors, user identity confidentiality, and ciphering of user and signaling data.

1.3 GMM – GPRS Mobility Management

The GMM entity provides procedures for the mobility of the MS, such as informing the network of its present location, and user identity confidentiality. It manages the GMM context (attach, detach, routing area updating), supports security functions such as authentication of user and MS, controls ciphering of data, and initiates the response to paging messages.

1.4 SM – Session Management

The main function of the session management (SM) is to support PDP context handling of the user terminal. Session Management activates, modifies and deletes the contexts for packet data protocols (PDP). Session Management services are provided at the SMREG-SAP and the SNSM-SAP for anonymous and non-anonymous access. The non-anonymous and anonymous access procedures for PDP context activation and PDP context deactivation are available at the SMREG-SAP. In addition there exists a PDP context modification for non-anonymous PDP contexts.

1.5 SNDCP - Subnetwork Dependant Convergence Protocol

SNDCP carries out all functions related to transfer of Network layer Protocol Data Units (N-PDUs) over GPRS in a transparent way. SNDCP helps to improve channel efficiency by means of compression techniques. The set of protocol entities above SNDCP consists of commonly used network protocols. They all use the same SNDCP entity, which then performs multiplexing of data coming from different sources to be sent using the service provided by the LLC layer.

1.6 GACI – GPRS AT Command Interpreter

The ACI is the GPRS extension of the ACI. It is specified in GSM 07.60. It is responsible for processing of the GPRS related AT Commands to setup, activate and deactivate the PDP context parameter. It also provides functionality for the interworking between GMM/SM/SNDCP and a packet oriented protocol like PPP.

1.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver

The USART is a hardware component that facilitates a connection between the mobile station and terminal equipment (e.g. a PC). This interface uses some of the circuits described in V.24.

The data exchange provided by this unit is serial and asynchronous (synchronous communication is not in the scope of this document). A driver that uses interrupts to manage a circular buffer for the sending and receiving direction is necessary in order to use this component in the GPRS. The driver has to be able to perform flow control.

1.8 TOM – Tunnelling of Messages

The TOM entity is present if and only if HS136 is supported (the feature flag FF_HS136 is enabled).

The main function of TOM is to tunnel non-GSM signalling messages between the MS and the SGSN. The only non-GSM signalling which is currently supported by TOM is for the EGPRS-136 system (according to IS-136-376). Data transfer in both uplink and downlink direction is possible. Two different priorities (high, low) of signalling data transfer are supported. TOM uses the unacknowledged mode of LLC and the acknowledged mode of GRR (RLC/MAC).

2 Parameters

/*

2.1 Declarations

*/

```
DECLARATION (SEND_MODE_2_BURSTS_TC403)
DECLARATION (RACH_2_BURSTS_TC403)
DECLARATION (BA_LIST_651)
DECLARATION (PLMN_ID_123_32)
DECLARATION (NCELLS_EMPTY)
DECLARATION (A_MPH_NCELL_1)
DECLARATION (A_MPH_NCELL_3)
DECLARATION (A_LIST_RES)
DECLARATION (A_LIST_RES_11)
DECLARATION (A_LIST_RES_UNK)
DECLARATION (LIST_RES_11)
DECLARATION (S_MS_ID_IMSI_HPLMN_TMSI)
DECLARATION (LIST_RES_UNK)
DECLARATION (NCELLS_1_UNK)
DECLARATION (A_ARFCN_11)
DECLARATION (A_BSIC_11)
DECLARATION (A_BSIC_UNK)
DECLARATION (RR_CELL_ENV_2149)
DECLARATION (ARFCN_LIST_PBCCH_124_6_11)
DECLARATION (SI4_REST_GPRS_TC706)
DECLARATION (LOC_AREA_IDENT_123_32_2147)
DECLARATION (SERVING_CELL_TC651)
DECLARATION (ARFCN_ARR_1)
DECLARATION (ARFCN_ARR_2)
DECLARATION (S_SI4_REST_EMPTY)
DECLARATION (A_BCCH_CONTENT_20)
DECLARATION (A_IMSI_CONTENT)
DECLARATION (S_BCCH_INFO_20)
DECLARATION (LIST_RES)
DECLARATION (ARFCN_ARR_1_RES)
DECLARATION (ARFCN_ARR_2_RES)
DECLARATION (ARFCN_IDX_ARR_1)
DECLARATION (ARFCN_IDX_ARR_2)
DECLARATION (ARFCN_IDX_VALUE_0)
DECLARATION (ARFCN_IDX_VALUE_1)
DECLARATION (A_MPH_NCELL_1C)
DECLARATION (ARFCN_IDX_VALUE_2)
DECLARATION (ARFCN_IDX_VALUE_3)
DECLARATION (ARFCN_IDX_VALUE_4)
DECLARATION (ARFCN_IDX_VALUE_5)
DECLARATION (ARFCN_IDX_VALUE_6)
DECLARATION (ARFCN_IDX_VALUE_7)
DECLARATION (CELL_DESC_1)
DECLARATION (SI13_PBCCH_DES_BCCH)
DECLARATION (CELL_DESC_2)
DECLARATION (CELL_DESC_BAD_BCCH)
DECLARATION (CLASS_MS)
DECLARATION (CLASS_MS_DUALBAND)
```

DECLARATION	(CIPHERING_2)
DECLARATION	(CIPHERING_3)
DECLARATION	(FRAME_NUMBER_1)
DECLARATION	(HO_PARAM_0)
DECLARATION	(HO_PARAM_1)
DECLARATION	(SI13_INFO_TC701_BCCH)
DECLARATION	(IA_2ND_PART)
DECLARATION	(KCV_EMPTY)
DECLARATION	(KCV_00112233)
DECLARATION	(KCV_12345678)
DECLARATION	(MM_INFO_1)
DECLARATION	(MM_INFO_2)
DECLARATION	(MM_INFO_4)
DECLARATION	(MM_INFO_3)
DECLARATION	(MM_INFO_15)
DECLARATION	(MOBILE_ID_IMSI_HPLMN)
DECLARATION	(MOBILE_ID_NOT_SET)
DECLARATION	(MOBILE_ID_TMSI)
DECLARATION	(MOBILE_ID_IMSI_TEST)
DECLARATION	(NO_CIPHERING)
DECLARATION	(NO_STARTING_TIME)
DECLARATION	(PAGING_NORMAL)
DECLARATION	(ONE_PLMN)
DECLARATION	(TWO_PLMNS)
DECLARATION	(ONE_LAC)
DECLARATION	(TWO_LACS)
DECLARATION	(ONE_RXLEV)
DECLARATION	(TWO_RXLEVS)
DECLARATION	(PLMN_ID_EMPTY)
DECLARATION	(PLMN_ID_123)
DECLARATION	(PLMN_ID_TEST)
DECLARATION	(PLMN_ID_123_V)
DECLARATION	(PLMN_ID_122)
DECLARATION	(PRR_CHANNEL_TYPE_2)
DECLARATION	(PRR_CHANNEL_TYPE_3)
DECLARATION	(CBCH_INACTIVE)
DECLARATION	(PRR_CHANNEL_TYPE_3_CB)
DECLARATION	(PRR_CHANNEL_TYPE_4)
DECLARATION	(PRR_CHANNEL_TYPE_5)
DECLARATION	(PRR_CHANNEL_TYPE_6)
DECLARATION	(PRR_CHANNEL_TYPE_7)
DECLARATION	(PRR_CHANNEL_TYPE_8)
DECLARATION	(PRR_CHANNEL_TYPE_9)
DECLARATION	(PRR_CHANNEL_TYPE_10)
DECLARATION	(PRR_CHANNEL_TYPE_11)
DECLARATION	(PRR_CHANNEL_TYPE_EGSM)
DECLARATION	(ARFCN_LIST_FULL_PBCCH)
DECLARATION	(ARFCN_LIST124_PBCCH2)
DECLARATION	(ARFCN_LIST_EMPTY_PBCCH)
DECLARATION	(MPH_ARFCN_LIST124_PBCCH)
DECLARATION	(PRR_TR_PARA_2)
DECLARATION	(PRR_TR_PARA_2A)
DECLARATION	(PRR_TR_PARA_2B)
DECLARATION	(PRR_TR_PARA_3)
DECLARATION	(PRR_TR_PARA_13)

DECLARATION	(PRR_TR_PARA_HO)
DECLARATION	(PRR_TR_PARA_ASY_HO)
DECLARATION	(RR_CELL_ENV_2147_32)
DECLARATION	(RR_CELL_ENV_2147)
DECLARATION	(RR_CELL_ENV_2147_V)
DECLARATION	(RAI_102)
DECLARATION	(ROUT_AREA_ID_102)
DECLARATION	(RXLEV_ARR_1)
DECLARATION	(RXLEV_ARR_2)
DECLARATION	(SEND_MODE_2_BURSTS)
DECLARATION	(SEND_MODE_2_BURSTS2)
DECLARATION	(SEND_MODE_8_BURSTS)
DECLARATION	(SEND_MODE_NO_BURSTS)
DECLARATION	(STARTING_TIME_1)
DECLARATION	(OP_MODE_EMPTY)
DECLARATION	(OP_MODE_EMPTY_NO_SERV)
DECLARATION	(OP_MODE_NORMAL)
DECLARATION	(OP_MODE_TEST_SIM)
DECLARATION	(OP_MODE_TEST_SIM_NO_SERV)
DECLARATION	(OP_MODE_TEST_SIM_LIM_SERV)
DECLARATION	(OP_MODE_NET_SRCH_MMI_NO_SRV)
DECLARATION	(OP_MODE_NET_SRCH_MMI_LIM_SRV)
DECLARATION	(OP_MODE_NET_SRCH_MMI)
DECLARATION	(CELL_OPT_BCCH_1)
DECLARATION	(CELL_SELECT_1)
DECLARATION	(CELL_SELECT_2)
DECLARATION	(CELL_SELECT_3)
DECLARATION	(CHANNEL_DESC_SDCCH)
DECLARATION	(CHANNEL_DESC_SDCCH2)
DECLARATION	(CHANNEL_DESC_BAD)
DECLARATION	(CHANNEL_DESC_FACCH2)
DECLARATION	(CHANNEL_DESC_HALFRATE)
DECLARATION	(CHANNEL_DESC_FACCH3)
DECLARATION	(CHANNEL_DESC_FACCH4)
DECLARATION	(CHANNEL_NEEDED_1)
DECLARATION	(CIPH_MODE_ON)
DECLARATION	(CIPH_MODE_OFF)
DECLARATION	(CIPH_RESP_NO_IMEI)
DECLARATION	(CIPH_RESP_WITH_IMEI)
DECLARATION	(CKSN_RESERVED)
DECLARATION	(CKSN_NOT_PRES_STRUCT)
DECLARATION	(CTRL_CHAN_DESC_1)
DECLARATION	(FREQ_CHAN_SEQ_1)
DECLARATION	(LOC_AREA_IDENT_122_2147)
DECLARATION	(LOC_AREA_IDENT_122_2147_V)
DECLARATION	(LOC_AREA_IDENT_123_2147)
DECLARATION	(LOC_AREA_IDENT_123_2147_V)
DECLARATION	(LOC_AREA_IDENT_123_2148)
DECLARATION	(MEAS_RESULT_NCELL_0)
DECLARATION	(MEAS_RESULT_NCELL_6_FTA)
DECLARATION	(MEAS_RESULT_NCELL_FTA_B)
DECLARATION	(MEAS_RESULT_NCELL_FTA_C)
DECLARATION	(MEAS_RESULT_NCELL_FTA_D)
DECLARATION	(MEAS_RESULT_NCELL_FTA_E)
DECLARATION	(MOB_CLASS2)

DECLARATION	(MOBILE_ALLOCATION_1)
DECLARATION	(MOBILE_ALLOCATION_2)
DECLARATION	(MOBILE_ALLOCATION_3)
DECLARATION	(MOBILE_ALLOCATION_4)
DECLARATION	(MOBILE_ALLOCATION_EGSM)
DECLARATION	(MOBILE_IDENTITY_TMSI)
DECLARATION	(MOBILE_IDENTITY_IMEISV)
DECLARATION	(MOBILE_IDENTITY_IMSI2)
DECLARATION	(MOBILE_IDENTITY_IMSI_HPLMN)
DECLARATION	(MOBILE_IDENTITY_IMSI_TEST)
DECLARATION	(POW_05_HO)
DECLARATION	(POWER_COMMAND_05)
DECLARATION	(PRIO_NOT_SUPPORTED)
DECLARATION	(RACH_CTRL_1)
DECLARATION	(RACH_CTRL_2)
DECLARATION	(RACH_CTRL_3)
DECLARATION	(REQUEST_REFERENCE_1)
DECLARATION	(REQUEST_REFERENCE_2)
DECLARATION	(REQUEST_REFERENCE_3)
DECLARATION	(REQUEST_REFERENCE_4)
DECLARATION	(REQUEST_REFERENCE_4A)
DECLARATION	(REQUEST_REFERENCE_5)
DECLARATION	(REQUEST_REFERENCE_FAIL)
DECLARATION	(START_TIME_1)
DECLARATION	(SYNCH_IND_1)
DECLARATION	(SYNCH_IND_0)
DECLARATION	(TIMING_ADVANCE_10)
DECLARATION	(TIMING_ADVANCE_27)
DECLARATION	(NCELLS_3)
DECLARATION	(NCELLS_4)
DECLARATION	(NCELLS_1)
DECLARATION	(NCELLS_11)
DECLARATION	(NCELLS_3_SACCH)
DECLARATION	(ARFCN_ARRAY_6)
DECLARATION	(ARFCN_NC)
DECLARATION	(ARFCN_NC1)
DECLARATION	(ARFCN_NC4)
DECLARATION	(RX_LEV_NC1)
DECLARATION	(RX_LEV_NC4)
DECLARATION	(BSIC_NC1)
DECLARATION	(BSIC_NC4)
DECLARATION	(FRAME_OFFSET_NC1)
DECLARATION	(TIME_ALIGNMENT_NC1)
DECLARATION	(TIME_ALIGNMENT_NC4)
DECLARATION	(BA_LIST)
DECLARATION	(BA_RANGE)
DECLARATION	(BCCH_INFO_TC701)
DECLARATION	(BSIC_ARFCN_EMPTY_LIST)
DECLARATION	(BSIC_NC)
DECLARATION	(CHAN_CODING_TC770)
DECLARATION	(CHANNEL_DESC_SDCCH8)
DECLARATION	(CHANNEL_DESC_SDCCH8_2)
DECLARATION	(CHANNEL_REQ_TC770)
DECLARATION	(CR_PAR_01)
DECLARATION	(CR_PAR_02)

DECLARATION	(CR_PAR_03)
DECLARATION	(CR_PAR_04)
DECLARATION	(CR_PAR_05)
DECLARATION	(DED_TLLI_BYTEARRAY_TC800)
DECLARATION	(DED_TLLI_BITBUFFER_TC770)
DECLARATION	(DED_TLLI_BITBUFFER_TC800)
DECLARATION	(FRAME_OFFSET_NC)
DECLARATION	(FRAME_OFFSET_NC4)
DECLARATION	(FREQUENCY_RANGE)
DECLARATION	(G_CHANNEL_DESC)
DECLARATION	(GPRS_INDIC_TC701)
DECLARATION	(GPRS_INDIC_TC703)
DECLARATION	(IA_IND_TC730)
DECLARATION	(IA_REST_ASSIGN_PARAM)
DECLARATION	(IA_REST_ASSIGN_PARAM2)
DECLARATION	(IA_REST_FREQ_PARAM)
DECLARATION	(IA_REST_OCTETS_TC730)
DECLARATION	(IA_REST_OCTETS_TC733)
DECLARATION	(IA_REST_OCTETS_UL_ASSIGN_TMA1)
DECLARATION	(IA_REST_OCT_PARAM_UL_ASSIGN_TMA1)
DECLARATION	(IA_REST_ASSIGN_PARAM_UL_TMA1)
DECLARATION	(PACKET_UL_ASS_TMA1)
DECLARATION	(IA_REST_OCTETS_UL_ASSIGN_TMA2)
DECLARATION	(IA_REST_OCT_PARAM_UL_ASSIGN_TMA2)
DECLARATION	(IA_REST_ASSIGN_PARAM_UL_TMA2)
DECLARATION	(IA_REST_PARAM)
DECLARATION	(IA_REST_PARAM2)
DECLARATION	(IA_REST_OCTETS_DL_ASSIGN)
DECLARATION	(IA_REST_OCTETS_DL_ASSIGN_COPY)
DECLARATION	(IA_REST_OCT_PARAM_DL_ASSIGN)
DECLARATION	(IA_REST_OCT_PARAM_DL_ASSIGN_COPY)
DECLARATION	(IA_REST_OCTETS_EMPTY)
DECLARATION	(IA_REST_ASSIGN_PARAM_DL)
DECLARATION	(IA_REST_ASSIGN_PARAM_DL_COPY)
DECLARATION	(PACKET_DL_ASS)
DECLARATION	(PACKET_DL_ASS_SHOW)
DECLARATION	(IMSI_EMPTY)
DECLARATION	(IMSI_TMSI_EMPTY)
DECLARATION	(IMSI_TMSI_EMPTY_PTMSI_NULL)
DECLARATION	(IMSIELEMENTE)
DECLARATION	(MEASUREMENT_RESULTS_TC970)
DECLARATION	(MEASURES_TC770)
DECLARATION	(MIMSI_1233347114912)
DECLARATION	(MOB_CLASS3_900)
DECLARATION	(MOB_CLASS3_900_SHOW)
DECLARATION	(MOB_IDENT_IMSI)
DECLARATION	(MS_ID_IMSI_HPLMN_TMSI_TC702)
DECLARATION	(MS_ID_IMSI_HPLMN_TMSI_TC704)
DECLARATION	(MS_ID_IMSI_HPLMN_TMSI_TC970)
DECLARATION	(NC_MODE_TC791)
DECLARATION	(NC_MODE_TC640)
DECLARATION	(NON_GPRS_ACT_REQ)
DECLARATION	(NON_GPRS_EST_REQ_TC860)
DECLARATION	(P0_BTS_PRMODE_01)
DECLARATION	(PACKET_CHANNEL_DESC)

DECLARATION	(PRR_CHANNEL_TYPE_5A)
DECLARATION	(RACH_2_BURSTS_TC871)
DECLARATION	(RACH_2_BURSTS_TC721)
DECLARATION	(RACH_2_BURSTS_TC803)
DECLARATION	(RACH_2_BURSTS_TC730)
DECLARATION	(REQUESTED_CHANNEL_TC970)
DECLARATION	(RESUMPTION_NO)
DECLARATION	(RESUMPTION_YES)
DECLARATION	(RX_LEV_NC)
DECLARATION	(SEND_MODE_2_BURSTS_TC721)
DECLARATION	(SEND_MODE_2_BURSTS_TC803)
DECLARATION	(SEND_MODE_2_BURSTS_TC730)
DECLARATION	(SEND_MODE_2_BURSTS_TC871)
DECLARATION	(SERVING_CELL_TC105)
DECLARATION	(SERVING_CELL_TC205)
DECLARATION	(SERVING_CELL_TC702)
DECLARATION	(SERVING_CELL_TC706)
DECLARATION	(SI_STATES_ONLY_SI13)
DECLARATION	(SI_STATES_TC702)
DECLARATION	(SI1_REST_GPRS_TC701)
DECLARATION	(SERVING_CELL_TC702_32)
DECLARATION	(SI13_IND_TC702)
DECLARATION	(SI13_INFO_TC701)
DECLARATION	(SI13_PBCCH_DES)
DECLARATION	(SI13_REST_GPRS_TC701)
DECLARATION	(SI13_REST_GPRS_TC701_BCCH)
DECLARATION	(SI3_REST_NOGPRS_TC206)
DECLARATION	(SI3_REST_GPRS_TC701)
DECLARATION	(SI3_REST_GPRS_TC703)
DECLARATION	(SI4_LSA_ID)
DECLARATION	(SI4_LSA_PARA)
DECLARATION	(SI4_OPT_SEL)
DECLARATION	(SI4_REST_NOGPRS_TC206)
DECLARATION	(SI4_REST_GPRS_TC701)
DECLARATION	(SI4_REST_GPRS_TC703)
DECLARATION	(SMS_SM_VAL_1)
DECLARATION	(TIMING_ADVANCE_30)
DECLARATION	(TIME_ALIGNMENT_NC)
DECLARATION	(TFI_TC800)
DECLARATION	(CHANNEL_DESC_AFTER)
DECLARATION	(RR_PACKET_DOWNLINK_TC980)
DECLARATION	(PACKET_TIMING_TC980)
DECLARATION	(POWER_CONTROL_TC980)
DECLARATION	(TAGGED_GAMMA_ARR)
DECLARATION	(TAGGED_GAMMA_EMPTY)
DECLARATION	(TAGGED_GAMMA_TC980)
DECLARATION	(NCELLS_2_CR_CANDIDATE)
DECLARATION	(RX_LEV_NC_CR)
DECLARATION	(ES_IND_ENABLE)
DECLARATION	(BCCH_BITMAP)
DECLARATION	(BA_LIST_205)
DECLARATION	(BA_LIST_702)
DECLARATION	(MPH_NCELL_2)
DECLARATION	(ARFCN_LIST_20)
DECLARATION	(XMEAS_RES_EMPTY)

```
DECLARATION             (XMEAS_RES_1A)
DECLARATION             (XMEAS_RES_1B)
DECLARATION             (XMEAS_RES_1C)
DECLARATION             (XMEAS_RES_2A)
DECLARATION             (XMEAS_RES_2B)
DECLARATION             (XMEAS_RES_2C)
DECLARATION             (XMEAS_RES_3)
DECLARATION             (XMEAS_RES_VALUE_1_0)
DECLARATION             (XMEAS_RES_VALUE_1_1)
DECLARATION             (XMEAS_RES_VALUE_1_2)
DECLARATION             (XMEAS_RES_VALUE_1_3)
DECLARATION             (XMEAS_RES_VALUE_1_4)
DECLARATION             (XMEAS_RES_VALUE_1_5)
DECLARATION             (XMEAS_RES_VALUE_1_6)
DECLARATION             (XMEAS_RES_VALUE_1_7)
DECLARATION             (XMEAS_RES_VALUE_1_8)
DECLARATION             (XMEAS_RES_VALUE_1_9)
DECLARATION             (XMEAS_RES_VALUE_1_10)
DECLARATION             (XMEAS_RES_VALUE_2_0)
DECLARATION             (XMEAS_RES_VALUE_2_1)
DECLARATION             (XMEAS_RES_VALUE_3_0)
DECLARATION             (XMEAS_RES_VALUE_3_1)
DECLARATION             (XMEAS_RES_VALUE_3_2)
DECLARATION             (XMEAS_RES_VALUE_3_3)
DECLARATION             (XMEAS_RES_VALUE_3_4)
DECLARATION             (XMEAS_RES_VALUE_3_5)
DECLARATION             (XMEAS_RES_VALUE_3_6)
DECLARATION             (XMEAS_RES_VALUE_3_7)
DECLARATION             (BA_LIST_ADDED_1)
DECLARATION             (BA_LIST_ADDED_2)
DECLARATION             (BA_LIST_REMOVED_1)
DECLARATION             (ADDED_FREQ_1)
DECLARATION             (ADDED_FREQ_2)
DECLARATION             (ADDED_FREQ_3)
DECLARATION             (ADDED_FREQ_4)
DECLARATION             (ADDED_FREQ_5)
DECLARATION             (ADDED_FREQ_6)
DECLARATION             (ADDED_FREQ_7)
DECLARATION             (ADDED_FREQ_8)
DECLARATION             (ADDED_FREQ_9)
DECLARATION             (ADDED_FREQ_10)
DECLARATION             (ADDED_FREQ_11)
DECLARATION             (REMOVED_FREQ_1)

/*
```

2.2 Fields

```
*/

/*
```

0x80, 0x00	length in bits
0x00, 0x00	offset in bits
0x00, ...	content (16 byte)

```
*/
```

```
FIELD (NCELL_DESC_2)            0x80, 0x00,
                                0x00, 0x00,
                                0x00, 0x00, 0x00, 0x04,
                                0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00,
                                0x80, 0x00, 0x00, 0x04
ENDFIELD (NCELL_DESC_2, 20)
/* todo: use really different ncell descriptions or always the same handle but not identical copies NCELL_DESC_1 and
NCELL_DESC_2 */
FIELD (NCELL_DESC_1)            0x80, 0x00,
                                0x00, 0x00,
                                0x08, 0x00, 0x00, 0x04,
                                0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00,
                                0x80, 0x00, 0x00, 0x04
ENDFIELD (NCELL_DESC_1, 20)
FIELD (NCELL_DESC_3)            0x80, 0x00,
                                0x00, 0x00,
                                0x00, 0x00, 0x00, 0x06,
                                0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00,
                                0x90, 0x00, 0x00, 0x04
ENDFIELD (NCELL_DESC_3, 20)

/*
```

31	length of list
43,0	channel 0x43
0xFF, 0xFF	end of list

```
*/
/*
```

0x80, 0x00	length in bits
0x00, 0x00	offset in bits
0x00, 0x04, 0x00, 0x00	115
0x00, 0x02, 0x00, 0x01	82 65
0x00, 0x08, 0x00, 0x00	52
0x81, 0x00, 0x00, 0x00	32 25

```
*/

FIELD (CELL_CHAN_DESC_1)        0x80, 0x00,
                                0x00, 0x00,
                                0x00, 0x04, 0x00, 0x00,
                                0x00, 0x02, 0x00, 0x01,
                                0x00, 0x08, 0x00, 0x00,
                                0x81, 0x00, 0x00, 0x00
ENDFIELD (CELL_CHAN_DESC_1, 20)
/*
```

0x80, 0x00	length in bits
0x00, 0x00	offset in bits
0x00, 0x04, 0x00, 0x00	115
0x00, 0x02, 0x00, 0x02	82 65
0x00, 0x09, 0x00, 0x00	52
0x81, 0x00, 0x00, 0x00	32 25

```
*/
```

```
FIELD (CELL_CHAN_DESC_2)        0x80, 0x00,
                                0x00, 0x00,
                                0x00, 0x04, 0x00, 0x00,
                                0x00, 0x02, 0x00, 0x02,
                                0x00, 0x09, 0x00, 0x00,
                                0x81, 0x00, 0x00, 0x00
ENDFIELD (CELL_CHAN_DESC_2, 20)

/*
```

0, 0x37, ...	time in TDMA frames between random bursts

```
*/
FIELD (DELTA_TWO_BURSTS)        0, 0x37,0,0,0,0,0,0
ENDFIELD (DELTA_TWO_BURSTS, 8)
FIELD (DELTA_EIGHT_BURSTS)      0, 0x56,0x56,0x56,0x56,0x56,0x56,0x56
ENDFIELD (DELTA_EIGHT_BURSTS, 8)

/*
```

	Frequency Channel Sequence (1,25,50,65,111,112,113,114)
0x01	Lowest ARFCN = 1
0x0	+15 = 16 not used
9	+9 = 25
0x0	+15 = 40 not used
a	+10 = 50
0xf	+15 = 65
0	+15 = 80 not used
0x0	+15 = 95 not used
0	+15 = 110 not used
0x1	+1 = 111
1	+1 = 112
0x1	+1 = 113
1	+1 = 114
0x0	not used
0	not used
0x0	not used
0	not used

```
*/
FIELD (FREQ_CHAN_SEQ_1_INC_SKIP)        0x0,0x9,0x0,0xa,0xf,0x0,0x0,0x0,0x0,0x1,0x1,0x1,0x1,0x0,0x0,0x0,0x0,0x0
ENDFIELD (FREQ_CHAN_SEQ_1_INC_SKIP,16)

/*
```

0x01, ...	digits
0x0F	end of digits

```
*/
FIELD (IMSI_1233247114912)      0x01, 0x02, 0x03, 0x03,
                                0x02, 0x04, 0x07, 0x01,
                                0x01, 0x04, 0x09, 0x01,
                                0x02,
                                0x0F
ENDFIELD (IMSI_1233247114912, 14)
FIELD (IMSI_0010147114912)      0x00, 0x00, 0x01, 0x00,
                                0x01, 0x04, 0x07, 0x01,
                                0x01, 0x04, 0x09, 0x01,
```

```
                                    0x02,
                                    0x0F
ENDFIELD (IMSI_0010147114912, 14)
/*
| 0x00, ... | Kc values |
*/
FIELD (KC_00112233)                 0x00, 0x00, 0x01, 0x01,
                                    0x02, 0x02, 0x03, 0x03
ENDFIELD (KC_00112233, 8)
/*
| 0x01, ... | Kc values |
*/
FIELD (KC_12345678)                 0x01, 0x02, 0x03, 0x04,
                                    0x05, 0x06, 0x07, 0x08
ENDFIELD (KC_12345678, 8)
/*
| 0x01, ... | IMEI digits (in messages) |
*/
FIELD (M_IMEI)                      1, 3, 5, 7, 9, 0, 2, 4,
                                    6, 8, 1, 1, 2, 2, 7, 8
ENDFIELD (M_IMEI, 16)
/*
| 0x00, ... | IMSI digits (in messages) |
*/
FIELD (M_IMSI_08158912)             0,8,1,5,8,9,1,2
ENDFIELD (M_IMSI_08158912, 8)
/*
| 0x01, ... | IMSI digits (in messages) |
*/
FIELD (M_IMSI_1233247114912)        0x01, 0x02, 0x03, 0x03,
                                    0x02, 0x04, 0x07, 0x01,
                                    0x01, 0x04, 0x09, 0x01,
                                    0x02
ENDFIELD (M_IMSI_1233247114912, 13)
FIELD (M_IMSI_0010147114912)        0x00, 0x00, 0x01, 0x00,
                                    0x01, 0x04, 0x07, 0x01,
                                    0x01, 0x04, 0x09, 0x01,
                                    0x02
ENDFIELD (M_IMSI_0010147114912, 13)
/*
| 0x01, ... | mobile country code 122 |
*/
FIELD (MCC_122)                     0x01, 0x02, 0x02
ENDFIELD (MCC_122, 3)
FIELD (MCC_001)                     0x00, 0x00, 0x01
ENDFIELD (MCC_001, 3)
/*
| 0x01, ... | Mobile country code 123 |
*/
FIELD (MCC_123)                     0x01, 0x02, 0x03
ENDFIELD (MCC_123, 3)
/*
```

3, 3	mobile network code 33

```
*/
FIELD (MNC_33)                  0x03, 0x03, 0x0F
ENDFIELD (MNC_33, 3)
FIELD (MNC_32)                  0x03, 0x02, 0x0F
ENDFIELD (MNC_32, 3)
FIELD (MNC_325)                 0x03, 0x02, 0x05
ENDFIELD (MNC_325, 3)
FIELD (MNC_01)                  0x00, 0x01, 0x0F
ENDFIELD (MNC_01, 3)
/*
```

0x16	mobile allocation

```
*/
FIELD (MOB_ALLOC_1)             0x16
ENDFIELD (MOB_ALLOC_1, 1)
FIELD (MOB_ALLOC_EGSM)          0x1F
ENDFIELD (MOB_ALLOC_EGSM, 1)
/*
```

0x14	mobile allocation

```
*/
FIELD (MOB_ALLOC_2)             0x14
ENDFIELD (MOB_ALLOC_2, 1)
/*
```

0x3F	mobile allocation

```
*/
FIELD (MOB_ALLOC_3)             0x3F
ENDFIELD (MOB_ALLOC_3, 1)
/*
```

	mobile allocation

```
*/
FIELD (MOB_ALLOC_4)             0x0F, 0xFF
ENDFIELD (MOB_ALLOC_4, 2)


/*
```

0x20, 0	channel 32
0x34, 0	channel 52
0x52, 0	channel 82
0xFF, 0xFF	end of hopping list
0x00, ...	rest of empty list

```
*/
FIELD (PRR_HOPPING_1)           0x20, 0x00,
                                0x34, 0x00,
                                0x52, 0x00,
                                0xFF, 0xFF,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

ENDFIELD (PRR_HOPPING_1, 128)

/*

0x34, 0	channel 52
0x52, 0	channel 82
0xFF, 0xFF	end of hopping list
0x00, ...	rest of empty list

*/

FIELD (PRR_HOPPING_2) 0x34, 0x00,
0x52, 0x00,
0xFF, 0xFF,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

/*

0x19, 0	channel 25
0x20, 0	channel 32
0x34, 0	channel 52
0x41, 0	channel 65
0x52, 0	channel 82
0x73, 0	channel 115
0xFF, 0xFF	end of hopping list
0x00, ...	rest of empty list

*/

```
FIELD (PRR_HOPPING_3)           0x19, 0,
                                0x20, 0,
                                0x34, 0,
                                0x41, 0,
                                0x52, 0,
                                0x73, 0,
                                0xFF, 0xFF,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
```

```
                                0x04, 0,
                                0x19, 0,
                                0x20, 0,
                                0x21, 0,
                                0x34, 0,
                                0x41, 0,
                                0x52, 0,
                                0x61, 0,
                                0x73, 0,
                                0xFF, 0xFF,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (PRR_HOPPING_4, 130)

/*
```

0x01, 0	channel 1
0x02, 0	channel 2
0x03, 0	channel 3
0x04, 0	channel 4
0x19, 0	channel 25
0x20, 0	channel 32
0xFF, 0xFF	end of hopping list
0x00, ...	rest of empty list

```
*/
FIELD (PRR_HOPPING_5)           0x01, 0,
                                0x02, 0,
                                0x03, 0,
                                0x04, 0,
                                0x19, 0,
                                0x20, 0,
                                0xFF, 0xFF,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

```
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00
ENDFIELD (PRR_HOPPING_5, 130)

/*
```

0x01, 0	channel 1
...	thru
0x20, 0	channel 32
0x5D, 0	channel 93
...	thru
0x7C, 0	channel 124
0xFF, 0xFF	end of hopping list

```
*/
FIELD (PRR_HOPPING_6)
        0x01, 0x00, 0x02, 0x00, 0x03, 0x00, 0x04, 0x00,
        0x05, 0x00, 0x06, 0x00, 0x07, 0x00, 0x08, 0x00,
        0x09, 0x00, 0x0A, 0x00, 0x0B, 0x00, 0x0C, 0x00,
        0x0D, 0x00, 0x0E, 0x00, 0x0F, 0x00, 0x10, 0x00,
        0x11, 0x00, 0x12, 0x00, 0x13, 0x00, 0x14, 0x00,
        0x15, 0x00, 0x16, 0x00, 0x17, 0x00, 0x18, 0x00,
        0x19, 0x00, 0x1A, 0x00, 0x1B, 0x00, 0x1C, 0x00,
        0x1D, 0x00, 0x1E, 0x00, 0x1F, 0x00, 0x20, 0x00,
        0x5D, 0x00, 0x5E, 0x00, 0x5F, 0x00, 0x60, 0x00,
        0x61, 0x00, 0x62, 0x00, 0x63, 0x00, 0x64, 0x00,
        0x65, 0x00, 0x66, 0x00, 0x67, 0x00, 0x68, 0x00,
        0x69, 0x00, 0x6A, 0x00, 0x6B, 0x00, 0x6C, 0x00,
        0x6D, 0x00, 0x6E, 0x00, 0x6F, 0x00, 0x70, 0x00,
        0x71, 0x00, 0x72, 0x00, 0x73, 0x00, 0x74, 0x00,
        0x75, 0x00, 0x76, 0x00, 0x77, 0x00, 0x78, 0x00,
        0x79, 0x00, 0x7A, 0x00, 0x7B, 0x00, 0x7C, 0x00,
        0xFF, 0xFF
ENDFIELD (PRR_HOPPING_6, 130)
/*
```

	channel 1,25,50,65,111,112,113,114

```
*/
FIELD (PRR_HOPPING_7)
                                1, 0, 25, 0, 50, 0, 65, 0, 111, 0, 112, 0, 113, 0, 114, 0,
                                0xFF, 0xFF,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (PRR_HOPPING_7, 130)
```

```
/*start by mse */

FIELD (PRR_HOPPING_8)           0xFF, 0xFF,
                                0x00, 0x00,
                                0x00, 0x00,
                                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 
```

0b00011011	Msg_type : SI3
0x00 0x01	Cell Identity
0b0001	Location Area Identification: Mobile Country Code
0b0001	Location Area Identification: Mobile Network Code
0x00 0x01	Location Area Identification: Location Area Code
0b0	Control Channel Description: spare
0b1	Control Channel Description: Attach / detach allowed
0b000	Control Channel Description: BS Access Grant Blocks reserved:
0b001	Control Channel Description: Common Control Channel Configuration: CCD_CCCH_1_COMB
0b00000	Control Channel Description: spare
0b011	Control Channel Description: BS Paging Multiframes
0x00	Control Channel Description: T 3212 time-out value
0b0	Cell Options: spare
0b0	Cell Options: Power Control: No
0b10	Cell Options: Discontinuous Transmission (BCCH): The MS shall not use uplink DTX
0b0001	Cell Options: Radio Link Timeout
0b110	Cell Selection Parameters: Cell Reselect Hysteresis 12dB
0b10011	Cell Selection Parameters: MS-TXPWR-MAX-CCH
0b0	Cell Selection Parameters: ACS; Use SI4
0b0	Cell Selection Parameters: NECI: No
0b000000	Cell Selection Parameters: RX Level Access Minimum
0b00	RACH Control Parameters: max_retrans: MAX_RETRANS_1
0b0010	RACH Control Parameters: tx_integer: SPREAD_TRANS_5
0b0	RACH Control Parameters: cell_bar_access: The cell is not barred
0b0	RACH Control Parameters: re: REESTAB_YES
0x00 0x00 - 15 byte	RACH Control Parameters: ac
0b0	SI 3 Rest Octets: No selection Parameter : L = 0
0b0	SI 3 Rest Octets: No Power Offset: L = 0
0b1	SI 3 Rest Octets: No SI 2ter Indicator: L = 1
0b0	SI 3 Rest Octets: No Early Classmark: L = 0
0b1	SI 3 Rest Octets: No Scheduling: L = 1
0b1	SI 3 Rest Octets: GPRS Indicator: H = 1
0b001	SI 3 Rest Octets: GPRS Indicator: RA Color = 1
0b0	SI 3 Rest Octets: GPRS Indicator:SI13 position
	Paddingbits

*/

```
FIELD (SI3_data)
                                        0xB8, 0x00,
                                        0x00, 0x00,
                                        0x1b, 0x00, 0x00, 0x11, 0x00, 0x00, 0x41, 0x03,
                                        0x00, 0x21, 0xd3, 0x00, 0x08, 0x00, 0x00, 0x2c,
                                        0xab, 0x2b, 0x2b
ENDFIELD (SI3_data, 23)

FIELD (ARFCN_43_20_124)
                                        0x43, 0,
                                        0x20, 0,
                                        124, 0
ENDFIELD (ARFCN_43_20_124, 6)
FIELD (RXLEV_22_21_20)
                                        0x22,0x21,0x20
ENDFIELD (RXLEV_22_21_20, 3)
```

/*

2.3 Constants

*/

```
BYTE    A5_1_SUPPORTED                  0 /* different coding to other A5_x values */
BYTE    STD_DUAL_EXT_0A                 0x0b
BYTE    ONE_PLMN_AVAILABLE              1
BYTE    TWO_PLMN_AVAILABLE              2
BYTE    BCC_3                           3
BYTE    ALGO_0                          0
BYTE    ATC_1                           1
BYTE    BA_0                            0
BYTE    BA_1                            1
BYTE    BCC_5                           5
BYTE    BCC_6                           6
BYTE    EOTD_ AVAIL                     1
BYTE    EOTD_NOT_AVAIL                  0
BYTE    BCCH_ARFCN_CELL_DESC_HI         0
BYTE    BCCH_ARFCN_CELL_DESC_HI_2       2
BYTE    BCCH_ARFCN_CELL_DESC_LO         0x20
BYTE    BCCH_ARFCN_CELL_DESC_LO_2       0x13
BYTE    BCCH_ARFCN_CELL_DESC_HI_BAD     3
BYTE    BCCH_ARFCN_CELL_DESC_LO_BAD     0x20
BYTE    BS_AG_BLKS_RES_5                0x05
BYTE    BS_PA_MFRMS_2                   0x02
BYTE    BS_PA_MFRMS_7                   0x07
BYTE    BSIC_0                          0x00
BYTE    BSIC_1                          1
BYTE    BSIC_3                          0x03
BYTE    BSIC_5                          0x05
BYTE    BSIC_0_NCC2                     0x10
BYTE    BSIC_3_NCC2                     0x13
BYTE    BSIC_5_NCC2                     0x15
BYTE    BSIC_6_NCC2                     0x16
BYTE    BSIC_0_NCC4                     0x20
BYTE    BSIC_3_NCC4                     0x23
BYTE    BSIC_3_NCC3                     0x1B
BYTE    BSIC_CELL_DESC_NCC              0x0
BYTE    BSIC_CELL_DESC_BCC              0x5
BYTE    BSIC_CELL_DESC_BCC_2            0x3
BYTE    CHANNELS_3                      3
BYTE    CKSN_7                          7
BYTE    C_SEND_NOT_CONT                 0
BYTE    CONST_0                         0
BYTE    CONST_1                         1
BYTE    DLT_23                          23
BYTE    DLT_42                          42
BYTE    DLT_LIMITED                     10
BYTE    FREQ_CHAN_SEQ_1_LOW_ARFCN       1
BYTE    HANDOV_REF_1                    0xA5
```

BYTE	HSN_4	0x04
BYTE	HSN_14	0x14
BYTE	IDX_0	0
BYTE	IDX_1	1
BYTE	IDX_2	2
BYTE	IDX_3	3
BYTE	IDX_4	4
BYTE	IDX_5	5
BYTE	IDX_6	6
BYTE	IDX_7	7
BYTE	LIMITED_NO	0
BYTE	LIMITED_YES	1
BYTE	MAIO_2	0x02
BYTE	MAIO_12	0x12
BYTE	MAIO_10	0x10
BYTE	MS_TXPWR_MAX_CCH_02	0x02
BYTE	RAC_9	9
SHORT	ACC_CTRL_CLASS_0000	0x0000
SHORT	ACC_CTRL_CLASS_0008	0x0008
SHORT	ACC_0005	0x0040
SHORT	ARFCN_3	0x0003
SHORT	ARFCN_X18	0x0018
SHORT	ARFCN_24	24
SHORT	ARFCN_X20	0x0020
SHORT	ARFCN_32	32
SHORT	ARFCN_X43	0x0043
SHORT	ARFCN_67	67
SHORT	ARFCN_X73	0x0073
SHORT	ARFCN_115	115
SHORT	ARFCN_124	124
SHORT	ARFCN_527	527
SHORT	ARFCN_531	531
SHORT	ARFCN_800	0x0320
SHORT	CELL_IDENT_3748	0x3748
SHORT	CELL_IDENT_3749	0x3749
SHORT	FN_OFFSET_1_SEC	1000
SHORT	LAC_2147	0x2147
SHORT	LAC_2148	0x2148
BYTE	NCI_1	1
BYTE	NCI_0	0
BYTE	NCC_0	0
BYTE	CALL_REF_1	0x01
BYTE	NCC_PERMITTED_1	0x01
BYTE	NCC_PERMITTED_2_4	0x14
BYTE	NCC_PERMITTED_FF	255
BYTE	NO_BURSTS	0
BYTE	PARAM_0	0
BYTE	PG_0	0
BYTE	POW_05	5
BYTE	POW_05_HO_ATC	1
BYTE	POW_05_HO_POW	5
BYTE	RA_1	0x91
BYTE	RA_2	0x10
BYTE	RA_3	0x0B
BYTE	RLT_64	0x0F

BYTE	ROT_0	0
BYTE	ROT_1	1
BYTE	RR_CAUSE_0A	0x0A
BYTE	NUM_CHAN_4	4
BYTE	NUM_CHAN_8	8
BYTE	RR_CAUSE_0	0x0
BYTE	RX_LEV_10	0x10
LONG	TIME_ALIGN_1	1947
BYTE	RX_LEV_20	0x20
BYTE	RX_LEV_35	0x2D
BYTE	RX_LEV_ACCESS_MIN_16	0x16
BYTE	RX_LEV_ACCESS_MIN_22	0x22
BYTE	RX_LEV_ACCESS_MIN_24	0x24
BYTE	RXLEV_17	17
BYTE	RXLEV_24	24
BYTE	RXLEV_31	31
BYTE	RXLEV_34	34
BYTE	RXLEV_41	41
BYTE	RXLEV_50	50
BYTE	RX_QUAL_1	1
BYTE	SS_SCREEN_3	0x03
BYTE	SS_SCREEN_1	0x01
BYTE	T1	4
BYTE	T2	6
BYTE	T3	8
BYTE	T3212_36_MIN	0x06
BYTE	T3212_0_MIN	0x00
BYTE	TAV_1	1
BYTE	TI_0	0x00
BYTE	TIME_ADV_27	27
BYTE	TIME_ADV_30	30
BYTE	TIME_ADV_10	10
BYTE	TIME_HPLMN_EMPTY	0x00
BYTE	TIME_HPLMN_VALID	0x05
BYTE	TN_0	0
BYTE	TN_3	3
BYTE	TN_4	4
BYTE	TSC_2	2
BYTE	TWO_BURSTS	2
BYTE	EIGHT_BURSTS	8
BYTE	FOUR_NCELLS	4
BYTE	SIX_NCELLS	6
BYTE	NUM_CHAN_4	4
BYTE	NUM_CHAN_8	8
BYTE	NO_NCELLS	0
BYTE	V_START_NOT_PRES	0
BYTE	V_START_PRES	1
SHORT	TMSI_142	0x0142
BYTE	DUMMY_01	0x01
BYTE	NC_MODE_DEF	NC_0
BYTE	CHANNEL_ANY	0x00
BYTE	FLAG_ONE	0x01
BYTE	FLAG_ZERO	0x00
BYTE	TN_2	0x02
BYTE	WAIT_INDICATION_0	0x00

BYTE	Ch_Req_Data_DEF	0x78	/* from TEST\GRR.doc */
BYTE	Ch_Req_Data_2nd	0x80	
SHORT	ARFCN_30	0x0014	
SHORT	DUMMY_SHORT	0x1234	
LONG	DUMMY_LONG	0x12345678	
LONG	fn_780	780	/* from TEST\GRR.doc */
LONG	fn_790	790	/* from TEST\GRR.doc */
LONG	fn_890	1024	
LONG	TMSI_0X142	0x00000142	
LONG	TMSI_0X00	0x00000000	
LONG	NOT_PRESENT_32BIT	0xFFFFFFFF	
BYTE	ESTCS_PAGING	0x80	
BYTE	CYCLE_ROUND_UP_1	30	
BYTE	SYNC_CELL	0	
SHORT	ARFCN_X30	0x0030	
SHORT	ARFCN_X40	0x0040	

/*

2.4 Arrays

*/

```
BEGINARRAY (A_IMSI_CONTENT, 15)
                0x01, 0x02, 0x03, 0x03, 0x02, 0x04, 0x07, 0x01,
                0x01, 0x04, 0x09, 0x01, 0x02, 0x00, 0x00
ENDARRAY
BEGIN_SHORT_ARRAY (MPH_NCELL_2, 33)
                32, 3, 99,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY (A_MPH_NCELL_1, 33)
                0x20, 0x7C, 0x63, 0x03, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY (A_MPH_NCELL_1C, 33)
                0x7C, 0x20, 0x63, 0x03, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY (A_MPH_NCELL_3,33)
                0x03, 0x1D, 0x20, 0x62, 0x63, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
```

```
                0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY(BA_LIST_651, 32)
                3, 32, 99, 124, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY_PART(ARFCN_LIST124_PBCCH2,1)
                124
ENDARRAY
BEGIN_SHORT_ARRAY_PART(ARFCN_LIST_FULL_PBCCH,12)
                124,1,2,3,4,5,6,7,8,9,10,11
ENDARRAY
BEGIN_SHORT_ARRAY_PART(ARFCN_LIST_PBCCH_124_6_11,3)
                124,6,11
ENDARRAY

BEGIN_SHORT_ARRAY_PART(ARFCN_LIST_EMPTY_PBCCH,1)
                0xFFFF
ENDARRAY

BEGIN_SHORT_ARRAY_PART(MPH_ARFCN_LIST124_PBCCH,1)
                124
ENDARRAY

BEGIN_SHORT_ARRAY(BA_LIST_205, 32)
                3, 29, 32, 98, 99,  0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY(BA_LIST_702, 32)
                3, 32, 99, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                0xFFFF, 0xFFFF
ENDARRAY
BEGIN_SHORT_ARRAY(ARFCN_LIST_20, 1)
                ARFCN_X20
ENDARRAY

BEGINARRAY (BCCH_BITMAP, 16)
                0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
                0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_ARR_1, 80)
                32, 800, 67, 24, 124, 115, 531, 527,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
```

```
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_ARR_1_RES, 80)
                32, 124, 67, 527, 531, 24, 115, 800,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_ARR_2, 80)
                32, 800, 67, 24, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_ARR_2_RES, 80)
                67, 32, 24, 800, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_NC, 6)
                0x03,
                0x20,
                0, 0, 0, 0
ENDARRAY
BEGIN_SHORT_ARRAY_PART (ARFCN_NC1, 1)
                124
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_NC4, 6)
                0x03,
                0x20,
                0x30,
                0x40,
                0, 0
ENDARRAY
```

```
BEGINARRAY_PART (BSIC_NC4, 4)
                6,
                5,
                5,
                5
ENDARRAY
BEGINARRAY (RX_LEV_NC4, 6)
                0x20,
                0x25,
                0x29,
                0x2A,
                0, 0
ENDARRAY
BEGIN_LONG_ARRAY (TIME_ALIGNMENT_NC4, 6)
                0x02,0x04,
                0x02,0x04,0,0
ENDARRAY
BEGIN_LONG_ARRAY (FRAME_OFFSET_NC4, 6)
                0x02,0x04,
                0x01,0x03,0,0
ENDARRAY

BEGINARRAY_PART (RX_LEV_NC1, 1)
                0
ENDARRAY

BEGINARRAY_PART (BSIC_NC1, 1)
                3
ENDARRAY
BEGIN_LONG_ARRAY_PART (TIME_ALIGNMENT_NC1, 1)
                0
ENDARRAY
BEGIN_LONG_ARRAY_PART (FRAME_OFFSET_NC1, 1)
                0
ENDARRAY

BEGINARRAY (RX_LEV_NC_CR, 6)
                0x35,
                0x2A,
                0, 0, 0, 0
ENDARRAY
BEGINARRAY (RX_LEV_NC, 6)
                0x20,
                0x2A,
                0, 0, 0, 0
ENDARRAY
BEGINARRAY (BSIC_NC, 6)
                6,
                5,
                0, 0, 0, 0
ENDARRAY
BEGIN_LONG_ARRAY (TIME_ALIGNMENT_NC, 6)
                0x02,0x04,
                0,0,0,0
ENDARRAY
```

```
BEGIN_LONG_ARRAY (FRAME_OFFSET_NC, 6)
                0x02,0x04,
                0,0,0,0
ENDARRAY

BEGINARRAY (NCELLS_3_SACCH, 24)
                0x20, 0,
                6, 0,
                0x2A,
                1,
                5,
                0,
                0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
ENDARRAY
BEGINARRAY (IMSIELEMENTE,15)
                1,2,3,3,2,4,7,1,1,4,9,1,2,0,0
ENDARRAY
BEGINARRAY (IMSI_EMPTY,15)
                0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
ENDARRAY
BEGINARRAY (RACH_2_BURSTS_TC730, 8)
                0x7B,0x79,
                0,0,0,0,0,0
ENDARRAY
BEGINARRAY (RACH_2_BURSTS_TC403, 8)
                0x9B,0x91,
                0,0,0,0,0,0
ENDARRAY

BEGINARRAY (RACH_2_BURSTS_TC721, 8)
                0x9B,0x91,
                0,0,0,0,0,0
ENDARRAY
BEGINARRAY (RACH_2_BURSTS_TC803, 8)
                0xFB,0xF1,
                0,0,0,0,0,0
ENDARRAY
BEGINARRAY (RACH_2_BURSTS_TC871, 8)
                0x9B,0x91,
                0,0,0,0,0,0
ENDARRAY
BEGINARRAY (MIMSI_1233347114912, 13)
                0x01, 0x02, 0x03, 0x03, 0x03, 0x04, 0x07, 0x01,
                0x01, 0x04, 0x09, 0x01, 0x02
ENDARRAY
BEGINARRAY (RXLEV_ARR_1, 80)
                50, 41, 34, 34, 31, 24, 24, 17,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
```

```
                0, 0, 0, 0, 0, 0, 0, 0
ENDARRAY
BEGINARRAY (RXLEV_ARR_2, 80)
                50, 41, 34, 34, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0
ENDARRAY
BEGIN_SHORT_ARRAY (ARFCN_ARRAY_6, 6)        /* Array containing 6 randomly choosen ARFCN's */
                0x1A,
                0x33,
                0x43,
                0x58,
                0x64,
                0x7B
ENDARRAY
BEGINARRAY (IA_IND_TC730, 27)
                0xB0, 0x00,
                0x10, 0x00, /*offset*/
                0x00, 0x00, 0x00, 0x06, 0x3f, 0x10, 0x0b, 0x55,
                0x1e, 0xc8, 0x41, 0x86, 0xc0, 0x45, 0x9f, 0x08,
                0x41, 0x80, 0x0a, 0xc0, 0x00, 0x9f, 0x08
ENDARRAY
BEGIN_STRUCT_ARRAY_PART (BA_LIST_ADDED_1, 6)
                ADDED_FREQ_2,
                ADDED_FREQ_3,
                ADDED_FREQ_4,
                ADDED_FREQ_5,
                ADDED_FREQ_6,
                ADDED_FREQ_1
ENDARRAY
BEGIN_STRUCT_ARRAY_PART (BA_LIST_ADDED_2, 6)
                ADDED_FREQ_7,
                ADDED_FREQ_8,
                ADDED_FREQ_9,
                ADDED_FREQ_10,
                ADDED_FREQ_11,
                ADDED_FREQ_6
ENDARRAY
BEGIN_STRUCT_ARRAY_PART (BA_LIST_REMOVED_1, 1)
                REMOVED_FREQ_1
ENDARRAY

BEGIN_STRUCT_ARRAY (ARFCN_IDX_ARR_1, 8)
                ARFCN_IDX_VALUE_0,
                ARFCN_IDX_VALUE_1,
                ARFCN_IDX_VALUE_2,
                ARFCN_IDX_VALUE_3,
                ARFCN_IDX_VALUE_4,
```

```
                ARFCN_IDX_VALUE_5,
                ARFCN_IDX_VALUE_6,
                ARFCN_IDX_VALUE_7
ENDARRAY
BEGIN_STRUCT_ARRAY (ARFCN_IDX_ARR_2, 4)
                ARFCN_IDX_VALUE_0,
                ARFCN_IDX_VALUE_1,
                ARFCN_IDX_VALUE_2,
                ARFCN_IDX_VALUE_3
ENDARRAY
BEGINARRAY (DED_TLLI_BYTEARRAY_TC800, 9)
                0x20, 0x00,
                0x00, 0x00,
                0x11, 0x0D, 0xAA, 0xFF, 0x11
ENDARRAY
/* without the parameter the empty array cannot be compiled */
BEGIN_STRUCT_ARRAY (XMEAS_RES_EMPTY, 0)
                XMEAS_RES_VALUE_1_0
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_1A, 6)
                XMEAS_RES_VALUE_1_0,
                XMEAS_RES_VALUE_1_1,
                XMEAS_RES_VALUE_1_2,
                XMEAS_RES_VALUE_1_3,
                XMEAS_RES_VALUE_1_4,
                XMEAS_RES_VALUE_1_5
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_1B, 4)
                XMEAS_RES_VALUE_1_6,
                XMEAS_RES_VALUE_1_7,
                XMEAS_RES_VALUE_1_8,
                XMEAS_RES_VALUE_1_9
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_1C, 1)
                XMEAS_RES_VALUE_1_10
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_2A, 3)
                XMEAS_RES_VALUE_1_1,
                XMEAS_RES_VALUE_2_0,
                XMEAS_RES_VALUE_2_1
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_2B, 6)
                XMEAS_RES_VALUE_1_0,
                XMEAS_RES_VALUE_1_1,
                XMEAS_RES_VALUE_1_2,
                XMEAS_RES_VALUE_2_0,
                XMEAS_RES_VALUE_1_5,
                XMEAS_RES_VALUE_2_1
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_2C, 2)
                XMEAS_RES_VALUE_1_7,
                XMEAS_RES_VALUE_1_8
ENDARRAY
BEGIN_STRUCT_ARRAY (XMEAS_RES_3, 8)
                XMEAS_RES_VALUE_3_0,
```

```
                    XMEAS_RES_VALUE_3_1,
                    XMEAS_RES_VALUE_3_2,
                    XMEAS_RES_VALUE_3_3,
                    XMEAS_RES_VALUE_3_4,
                    XMEAS_RES_VALUE_3_5,
                    XMEAS_RES_VALUE_3_6,
                    XMEAS_RES_VALUE_3_7
ENDARRAY

/*
32, 124, 67, 527, 531, 24, 115, 800,
50, 41, 34, 34, 31, 24, 24, 17,
*/

/*
```

2.5 Structures

```
 */
BEGIN_PSTRUCT ("sync_result", LIST_RES)
                SET_COMP("arfcn",     124)
                SET_COMP("sb_flag", SB_FOUND)
                SET_COMP("bsic",      3)
ENDSTRUCT
BEGIN_PSTRUCT ("sync_result", LIST_RES_11)
                SET_COMP("arfcn",     11)
                SET_COMP("sb_flag", SB_FOUND)
                SET_COMP("bsic",      4)
ENDSTRUCT
BEGIN_PSTRUCT ("sync_result", LIST_RES_UNK)
                SET_COMP("arfcn",     11)
                SET_COMP("sb_flag", 2)
                SKIP_COMP("bsic")
ENDSTRUCT

BEGIN_STRUCT_ARRAY_PART(A_LIST_RES,1)
                    LIST_RES
ENDARRAY
BEGIN_STRUCT_ARRAY_PART(A_LIST_RES_UNK,1)
                    LIST_RES_UNK
ENDARRAY

BEGIN_STRUCT_ARRAY_PART(A_LIST_RES_11,1)
                    LIST_RES_11
ENDARRAY

BEGIN_PSTRUCT ("classmark", CLASS_MS)
                SET_COMP ("pclass",             CLASS_2)
                SKIP_COMP ("pclass2")
ENDSTRUCT
BEGIN_PSTRUCT ("classmark", CLASS_MS_DUALBAND)
                SET_COMP ("pclass",             CLASS_4)
                SET_COMP ("pclass2",            CLASS_1)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("bcch_info",                 S_BCCH_INFO_20)
              SET_COMP ("v_bcch",           V_BCCH_PRES)
              SET_COMP ("bcch",             A_BCCH_CONTENT_20)
ENDSTRUCT
BEGINARRAY (A_BCCH_CONTENT_20, 16)
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00
ENDARRAY
BEGIN_PSTRUCT ("frame_no", FRAME_NUMBER_1)
              SET_COMP ("t1",               T1)
              SET_COMP ("t2",               T2)
              SET_COMP ("t3",               T3)
ENDSTRUCT
BEGIN_PSTRUCT ("kcv", KCV_00112233)
              SET_COMP ("v_kc",             V_KC_PRES)
              SET_COMP ("kc",               KC_00112233)
ENDSTRUCT
BEGIN_PSTRUCT ("kcv", KCV_12345678)
              SET_COMP ("v_kc",             V_KC_PRES)
              SET_COMP ("kc",               KC_12345678)
ENDSTRUCT
BEGIN_PSTRUCT ("kcv", KCV_EMPTY)
              SET_COMP ("v_kc",             V_KC_NOT_PRES)
              SKIP_COMP ("kc")
ENDSTRUCT
BEGIN_PSTRUCT ("mm_info", MM_INFO_1)
              SET_COMP ("valid",            MM_INFO_PRES)
              SET_COMP ("la",               LA_NOT_IN_FRBD_LST_INCL)
              SET_COMP ("att",              ATT_ALLOW)
              SET_COMP ("re",               REESTAB_YES)
              SET_COMP ("band",             BND_DMY_VAL)
              SET_COMP ("ncc",              NCC_0)
              SET_COMP ("bcc",              BCC_6)
              SET_COMP ("t3212",            T3212_36_MIN)
ENDSTRUCT
BEGIN_PSTRUCT ("mm_info", MM_INFO_2)
              SET_COMP ("valid",            MM_INFO_PRES)
              SET_COMP ("la",               LA_NOT_IN_FRBD_LST_INCL)
              SET_COMP ("att",              ATT_ALLOW)
              SET_COMP ("re",               REESTAB_YES)
              SET_COMP ("band",             BND_DMY_VAL)
              SET_COMP ("ncc",              NCC_0)
              SET_COMP ("bcc",              BCC_5)
              SET_COMP ("t3212",            T3212_36_MIN)
ENDSTRUCT
BEGIN_PSTRUCT ("mm_info", MM_INFO_4)
              SET_COMP ("valid",            MM_INFO_PRES)
              SET_COMP ("la",               LA_NOT_IN_FRBD_LST_INCL)
              SET_COMP ("att",              ATT_ALLOW)
              SET_COMP ("re",               REESTAB_YES)
              SET_COMP ("band",             BND_DMY_VAL)
              SET_COMP ("ncc",              NCC_0)
              SET_COMP ("bcc",              BCC_3)
              SET_COMP ("t3212",            T3212_36_MIN)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("mm_info", MM_INFO_15)
                SET_COMP ("valid",              MM_INFO_PRES)
                SET_COMP ("la",                 LA_NOT_IN_FRBD_LST_INCL)
                SET_COMP ("att",                ATT_NOT_ALLOW)
                SET_COMP ("re",                 REESTAB_NO)
                SET_COMP ("band",               BND_DMY_VAL)
                SET_COMP ("ncc",                NCC_0)
                SET_COMP ("bcc",                BCC_5)
                SET_COMP ("t3212",              T3212_0_MIN)
ENDSTRUCT
BEGIN_PSTRUCT ("mm_info", MM_INFO_3)
                SET_COMP ("valid",              MM_INFO_PRES)
                SET_COMP ("la",                 LA_NOT_IN_FRBD_LST_INCL)
                SET_COMP ("att",                ATT_ALLOW)
                SET_COMP ("re",                 REESTAB_YES)
                SET_COMP ("band",               BND_DMY_VAL)
                SET_COMP ("ncc",                NCC_0)
                SET_COMP ("bcc",                BCC_6)
                SET_COMP ("t3212",              T3212_36_MIN)
ENDSTRUCT
BEGIN_PSTRUCT ("imsi_struct", MOBILE_ID_NOT_SET)
                SET_COMP ("v_mid",              V_MID_PRES)
                SET_COMP ("id_type",            NOT_PRESENT_8BIT)
                SKIP_COMP ("id")
                SKIP_COMP ("tmsi_dig")
ENDSTRUCT
BEGIN_PSTRUCT ("imsi_struct", MOBILE_ID_IMSI_HPLMN)
                SET_COMP ("v_mid",              V_MID_PRES)
                SET_COMP ("id_type",            TYPE_IMSI)
                SET_COMP ("id",                 IMSI_1233247114912)
                SKIP_COMP ("tmsi_dig")
ENDSTRUCT
BEGIN_PSTRUCT ("imsi_struct", MOBILE_ID_IMSI_TEST)
                SET_COMP ("v_mid",              V_MID_PRES)
                SET_COMP ("id_type",            TYPE_IMSI)
                SET_COMP ("id",                 IMSI_0010147114912)
                SKIP_COMP ("tmsi_dig")
ENDSTRUCT
BEGIN_PSTRUCT ("tmsi_struct", MOBILE_ID_TMSI)
                SET_COMP ("v_mid",              V_MID_PRES)
                SET_COMP ("id_type",            TYPE_TMSI)
                SKIP_COMP ("id")
                SET_COMP ("tmsi_dig",           TMSI_142)
ENDSTRUCT
BEGIN_PSTRUCT ("ciph", NO_CIPHERING)
                SET_COMP ("stat",               STAT_CIPH_OFF)
                SKIP_COMP ("algo")
                SKIP_COMP ("kc")
ENDSTRUCT
BEGIN_PSTRUCT ("ciph", CIPHERING_2)
                SET_COMP ("stat",               STAT_CIPH_ON)
                SET_COMP ("algo",               ALGO_0)
                SET_COMP ("kc",                 KC_12345678)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("ciph", CIPHERING_3)
                SET_COMP ("stat",               STAT_CIPH_ON)
                SET_COMP ("algo",               ALGO_0)
                SET_COMP ("kc",                 KC_00112233)
ENDSTRUCT
BEGIN_PSTRUCT ("start", NO_STARTING_TIME)
                SET_COMP ("v_start",            V_START_NOT_PRES)
                SKIP_COMP ("t1")
                SKIP_COMP ("t2")
                SKIP_COMP ("t3")
ENDSTRUCT
BEGIN_PSTRUCT ("start", STARTING_TIME_1)
                SET_COMP ("v_start",            V_START_PRES)
                SET_COMP ("t1",                 T1)
                SET_COMP ("t2",                 T2)
                SET_COMP ("t3",                 T3)
ENDSTRUCT
BEGIN_PSTRUCT ("plmn", PLMN_ID_EMPTY)
                SET_COMP ("v_plmn",             V_PLMN_NOT_PRES)
                SKIP_COMP ("mcc")
                SKIP_COMP ("mnc")
ENDSTRUCT
BEGIN_PSTRUCT ("plmn", PLMN_ID_123)
                SET_COMP ("v_plmn",             V_PLMN_PRES)
                SET_COMP ("mcc",                MCC_123)
                SET_COMP ("mnc",                MNC_325)
ENDSTRUCT
BEGIN_PSTRUCT ("plmn", PLMN_ID_123_32)
                SET_COMP ("v_plmn",             V_PLMN_PRES)
                SET_COMP ("mcc",                MCC_123)
                SET_COMP ("mnc",                MNC_32)
ENDSTRUCT
BEGIN_PSTRUCT ("plmn", PLMN_ID_123_V)
                SET_COMP ("v_plmn",             V_PLMN_PRES)
                SET_COMP ("mcc",                MCC_123)
                SET_COMP ("mnc",                MNC_33)
ENDSTRUCT
BEGIN_PSTRUCT ("plmn", PLMN_ID_122)
                SET_COMP ("v_plmn",             V_PLMN_PRES)
                SET_COMP ("mcc",                MCC_122)
                SET_COMP ("mnc",                MNC_325)
ENDSTRUCT
BEGIN_PSTRUCT ("plmn", PLMN_ID_TEST)
                SET_COMP ("v_plmn",             V_PLMN_PRES)
                SET_COMP ("mcc",                MCC_001)
                SET_COMP ("mnc",                MNC_01)
ENDSTRUCT
BEGIN_STRUCT_ARRAY_PART (ONE_PLMN, 1)
                PLMN_ID_123_V
ENDARRAY
BEGIN_STRUCT_ARRAY_PART (TWO_PLMNS, 2)
                PLMN_ID_123_V,
                PLMN_ID_123_32
ENDARRAY
BEGIN_SHORT_ARRAY_PART (ONE_LAC, 1)
```

```
            LAC_2147
ENDARRAY
BEGIN_SHORT_ARRAY_PART (TWO_LACS, 2)
            LAC_2147,  LAC_2147
ENDARRAY
BEGINARRAY_PART (ONE_RXLEV,1)
            0x21
ENDARRAY
BEGINARRAY_PART (TWO_RXLEVS,1)
            0x21, 0x22
ENDARRAY
BEGIN_PSTRUCT ("emlpp_prio", PRIO_NOT_SUPPORTED)
            SET_COMP ("v_emlpp",        FALSE)
            SET_COMP ("emlpp",          CALL_PRIO_NO)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_EMPTY)
            SET_COMP ("v_op",           V_OP_PRES)
            SET_COMP ("ts",             TS_NO_AVAIL)
            SET_COMP ("m",              M_AUTO)
            SET_COMP ("sim_ins",        SIM_NO_INSRT)
            SET_COMP ("func",           FUNC_LIM_SERV_ST_SRCH)
            SET_COMP ("service",        LIMITED_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_EMPTY_NO_SERV)
            SET_COMP ("v_op",           V_OP_PRES)
            SET_COMP ("ts",             TS_NO_AVAIL)
            SET_COMP ("m",              M_AUTO)
            SET_COMP ("sim_ins",        SIM_NO_INSRT)
            SET_COMP ("func",           FUNC_LIM_SERV_ST_SRCH)
            SET_COMP ("service",        NO_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_NORMAL)
            SET_COMP ("v_op",           V_OP_PRES)
            SET_COMP ("ts",             TS_NO_AVAIL)
            SET_COMP ("m",              M_AUTO)
            SET_COMP ("sim_ins",        SIM_INSRT)
            SET_COMP ("func",           FUNC_PLMN_SRCH)
            SET_COMP ("service",        FULL_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_NET_SRCH_MMI_NO_SRV)
            SET_COMP ("v_op",           V_OP_PRES)
            SET_COMP ("ts",             TS_AVAIL)
            SET_COMP ("m",              M_AUTO)
            SET_COMP ("sim_ins",        SIM_INSRT)
            SET_COMP ("func",           FUNC_NET_SRCH_BY_MMI)
            SET_COMP ("service",        NO_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_NET_SRCH_MMI_LIM_SRV)
            SET_COMP ("v_op",           V_OP_PRES)
            SET_COMP ("ts",             TS_AVAIL)
            SET_COMP ("m",              M_AUTO)
            SET_COMP ("sim_ins",        SIM_INSRT)
            SET_COMP ("func",           FUNC_NET_SRCH_BY_MMI)
            SET_COMP ("service",        LIMITED_SERVICE)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("op", OP_MODE_NET_SRCH_MMI)
                SET_COMP ("v_op",               V_OP_PRES)
                SET_COMP ("ts",                 TS_AVAIL)
                SET_COMP ("m",                  M_AUTO)
                SET_COMP ("sim_ins",            SIM_INSRT)
                SET_COMP ("func",               FUNC_NET_SRCH_BY_MMI)
                SET_COMP ("service",            FULL_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_TEST_SIM)
                SET_COMP ("v_op",               V_OP_PRES)
                SET_COMP ("ts",                 TS_AVAIL)
                SET_COMP ("m",                  M_AUTO)
                SET_COMP ("sim_ins",            SIM_INSRT)
                SET_COMP ("func",               FUNC_PLMN_SRCH)
                SET_COMP ("service",            FULL_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_TEST_SIM_NO_SERV)
                SET_COMP ("v_op",               V_OP_PRES)
                SET_COMP ("ts",                 TS_AVAIL)
                SET_COMP ("m",                  M_AUTO)
                SET_COMP ("sim_ins",            SIM_INSRT)
                SET_COMP ("func",               FUNC_PLMN_SRCH)
                SET_COMP ("service",            NO_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("op", OP_MODE_TEST_SIM_LIM_SERV)
                SET_COMP ("v_op",               V_OP_PRES)
                SET_COMP ("ts",                 TS_AVAIL)
                SET_COMP ("m",                  M_AUTO)
                SET_COMP ("sim_ins",            SIM_INSRT)
                SET_COMP ("func",               FUNC_PLMN_SRCH)
                SET_COMP ("service",            LIMITED_SERVICE)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_2)
                SET_COMP ("ch",                 CH_SDCCH_4_1)
                SET_COMP ("tn",                 TN_3)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("h",                  HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",               MAIO_12)
                SET_COMP ("hsn",                HSN_14)
                SET_COMP ("ma",                 PRR_HOPPING_1)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_3)
                SET_COMP ("ch",                 CH_SDCCH_4_1)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("h",                  HOP_NO)
                SET_COMP ("arfcn",              ARFCN_X73)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
                SKIP_COMP ("ma")
ENDSTRUCT
BEGIN_PSTRUCT ("cbch", PRR_CHANNEL_TYPE_3_CB)
                SET_COMP ("stat",               TRUE)
                SET_COMP ("ch",                 CH_SDCCH_4_1)
```

```
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("h",                  HOP_NO)
                SET_COMP ("arfcn",              ARFCN_X73)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
                SKIP_COMP ("ma")
ENDSTRUCT
BEGIN_PSTRUCT ("cbch", CBCH_INACTIVE)
                SET_COMP ("stat",               FALSE)
                SHOW_COMP ("ch")
                SHOW_COMP ("tn")
                SHOW_COMP ("tsc")
                SHOW_COMP ("h")
                SHOW_COMP ("arfcn")
                SHOW_COMP ("maio")
                SHOW_COMP ("hsn")
                SHOW_COMP ("ma")
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_4)
                SET_COMP ("ch",                 CH_TCH_F)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("h",                  HOP_NO)
                SET_COMP ("arfcn",              ARFCN_X73)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
                SKIP_COMP ("ma")
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_5)
                SKIP_COMP ("ch")
                SKIP_COMP ("tn")
                SKIP_COMP ("tsc")
                SKIP_COMP ("h")
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",               MAIO_10)
                SKIP_COMP ("hsn")
                SET_COMP ("ma",                 PRR_HOPPING_2)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_5A)
                SKIP_COMP ("ch")
                SKIP_COMP ("tn")
                SKIP_COMP ("tsc")
                SKIP_COMP ("h")
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",               0x00)
                SKIP_COMP ("hsn")
                SET_COMP ("ma",                 PRR_HOPPING_8)
ENDSTRUCT
BEGIN_PSTRUCT ("ncells",  NCELLS_3)
        SET_COMP ("no_of_ncells",               2)
        SET_COMP ("arfcn",                      ARFCN_NC)
        SET_COMP ("rx_lev",                     RX_LEV_NC)
        SET_COMP ("bsic",                       BSIC_NC)
        SET_COMP ("time_alignmt",               TIME_ALIGNMENT_NC)
```

```
        SET_COMP ("frame_offset",          FRAME_OFFSET_NC)
ENDSTRUCT
BEGIN_PSTRUCT ("ncells",  NCELLS_4)
        SET_COMP ("no_of_ncells",                4)
        SET_COMP ("arfcn",                       ARFCN_NC4)
        SET_COMP ("rx_lev",                      RX_LEV_NC4)
        SET_COMP ("bsic",                        BSIC_NC4)
        SET_COMP ("time_alignmt",                TIME_ALIGNMENT_NC4)
        SET_COMP ("frame_offset",          FRAME_OFFSET_NC)
ENDSTRUCT

BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_6)
                SET_COMP ("ch",                  CH_TCH_F)
                SET_COMP ("tn",                  TN_4)
                SET_COMP ("tsc",                 TSC_2)
                SET_COMP ("h",                   HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",                MAIO_12)
                SET_COMP ("hsn",                 HSN_14)
                SET_COMP ("ma",                  PRR_HOPPING_2)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_7)
                SET_COMP ("ch",                  CH_TCH_F)
                SET_COMP ("tn",                  TN_4)
                SET_COMP ("tsc",                 TSC_2)
                SET_COMP ("h",                   HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",                MAIO_2)
                SET_COMP ("hsn",                 HSN_4)
                SET_COMP ("ma",                  PRR_HOPPING_3)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_8)
                SET_COMP ("ch",                  CH_TCH_F)
                SET_COMP ("tn",                  TN_4)
                SET_COMP ("tsc",                 TSC_2)
                SET_COMP ("h",                   HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",                MAIO_12)
                SET_COMP ("hsn",                 HSN_14)
                SET_COMP ("ma",                  PRR_HOPPING_4)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_9)
                SET_COMP ("ch",                  CH_TCH_F)
                SET_COMP ("tn",                  TN_4)
                SET_COMP ("tsc",                 TSC_2)
                SET_COMP ("h",                   HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",                MAIO_2)
                SET_COMP ("hsn",                 HSN_4)
                SET_COMP ("ma",                  PRR_HOPPING_5)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_10)
                SET_COMP ("ch",                  CH_TCH_F)
                SET_COMP ("tn",                  TN_4)
                SET_COMP ("tsc",                 TSC_2)
```

```
                SET_COMP ("h",              HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",           MAIO_2)
                SET_COMP ("hsn",            HSN_4)
                SET_COMP ("ma",             PRR_HOPPING_6)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_11)
                SET_COMP ("ch",             CH_TCH_F)
                SET_COMP ("tn",             TN_4)
                SET_COMP ("tsc",            TSC_2)
                SET_COMP ("h",              HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",           MAIO_2)
                SET_COMP ("hsn",            HSN_4)
                SET_COMP ("ma",             PRR_HOPPING_7)
ENDSTRUCT
BEGIN_PSTRUCT ("ch_type", PRR_CHANNEL_TYPE_EGSM)
                SET_COMP ("ch",             CH_TCH_F)
                SET_COMP ("tn",             TN_4)
                SET_COMP ("tsc",            TSC_2)
                SET_COMP ("h",              HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",           MAIO_12)
                SET_COMP ("hsn",            HSN_14)
                SET_COMP ("ma",             PRR_HOPPING_EGSM)
ENDSTRUCT
BEGIN_PSTRUCT ("ho_param", HO_PARAM_1)
                SET_COMP ("ho_ref",                 HANDOV_REF_1)
                SET_COMP ("ho_pow",                 POW_05)
                SET_COMP ("ho_acc_type",            ATC_1)
                SET_COMP ("ho_nci",                 NCI_1)
ENDSTRUCT
BEGIN_PSTRUCT ("ho_param", HO_PARAM_0)
                SET_COMP ("ho_ref",                 HANDOV_REF_1)
                SET_COMP ("ho_pow",                 POW_05)
                SET_COMP ("ho_acc_type",            ATC_1)
                SET_COMP ("ho_nci",                 NCI_0)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_2)
                SET_COMP ("power",          MS_TXPWR_MAX_CCH_02)
                SET_COMP ("dtx",            DTX_USED)
                SET_COMP ("rlt",            RLT_64)
                SET_COMP ("tav",            TIME_ADV_27)
                SET_COMP ("pwrc",           POW_CTRL_YES)
                SET_COMP ("mode",           CHM_SIG_ONLY)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_2A)
                SET_COMP ("power",          MS_TXPWR_MAX_CCH_02)
                SET_COMP ("dtx",            DTX_USED)
                SET_COMP ("rlt",            RLT_64)
                SET_COMP ("tav",            TIME_ADV_30)
                SET_COMP ("pwrc",           POW_CTRL_YES)
                SET_COMP ("mode",           CHM_SIG_ONLY)
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("es_ind_tag", ES_IND_ENABLE)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_2B)
                SET_COMP ("power",          MS_TXPWR_MAX_CCH_02)
                SET_COMP ("dtx",            DTX_USED)
                SET_COMP ("rlt",            RLT_64)
                SET_COMP ("tav",            TIME_ADV_10)
                SET_COMP ("pwrc",           POW_CTRL_YES)
                SET_COMP ("mode",           CHM_SIG_ONLY)
ENDSTRUCT
BEGIN_PSTRUCT ("rai", RAI_102)
                SET_COMP ("plmn",           PLMN_ID_123)
                SET_COMP ("lac",            LAC_2147)
                SET_COMP ("rac",            RAC_9)
ENDSTRUCT
BEGIN_MSTRUCT ("rout_area_id", ROUT_AREA_ID_102)
                SET_COMP ("mcc",            MCC_123)
                SET_COMP ("mnc",            MNC_325)
                SET_COMP ("lac",            LAC_2147)
                SET_COMP ("rac",            RAC_9)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_3)
                SET_COMP ("power",          POW_05)
                SET_COMP ("dtx",            DTX_USED)
                SET_COMP ("rlt",            RLT_64)
                SKIP_COMP ("tav")
                SET_COMP ("pwrc",           POW_CTRL_YES)
                SET_COMP ("mode",           CHM_SIG_ONLY)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_13)
                SET_COMP ("power",          POW_05)
                SET_COMP ("dtx",            DTX_USED)
                SET_COMP ("rlt",            RLT_64)
                SKIP_COMP ("tav")
                SET_COMP ("pwrc",           POW_CTRL_YES)
                SET_COMP ("mode",           CHM_SPEECH)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_HO)
                SKIP_COMP ("power")
                SKIP_COMP ("dtx")
                SKIP_COMP ("rlt")
                SET_COMP ("tav",            TAV_1)
                SKIP_COMP ("pwrc")
                SET_COMP ("mode",           CHM_SPEECH)
ENDSTRUCT
BEGIN_PSTRUCT ("tr_para", PRR_TR_PARA_ASY_HO)
                SKIP_COMP ("power")
                SKIP_COMP ("dtx")
                SKIP_COMP ("rlt")
                SKIP_COMP ("tav")
                SKIP_COMP ("pwrc")
                SET_COMP ("mode",           CHM_SPEECH)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS)
                SET_COMP ("c",              C_SEND_NOT_CONT)
```

```
                SET_COMP ("no",                 TWO_BURSTS)
                SET_COMP ("delta",              DELTA_TWO_BURSTS)
                SET_COMP ("rach",               RACH_2_BURSTS)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_8_BURSTS)
                SET_COMP ("c",                  C_SEND_NOT_CONT)
                SET_COMP ("no",                 EIGHT_BURSTS)
                SET_COMP ("delta",              DELTA_EIGHT_BURSTS)
                SET_COMP ("rach",               RACH_8_BURSTS)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS2)
                SET_COMP ("c",                  C_SEND_NOT_CONT)
                SET_COMP ("no",                 TWO_BURSTS)
                SET_COMP ("delta",              DELTA_TWO_BURSTS)
                SET_COMP ("rach",               RACH_2_BURSTS2)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_NO_BURSTS)
                SET_COMP ("c",                  C_SEND_NOT_CONT)
                SET_COMP ("no",                 NO_BURSTS)
                SKIP_COMP ("delta")
                SKIP_COMP ("rach")
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_0)
        SET_COMP ("arfcn",                      ARFCN_32)
        SET_COMP ("idx",                        IDX_5)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_1)
        SET_COMP ("arfcn",                      ARFCN_800)
        SET_COMP ("idx",                        IDX_7)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_2)
        SET_COMP ("arfcn",                      ARFCN_67)
        SET_COMP ("idx",                        IDX_0)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_3)
        SET_COMP ("arfcn",                      ARFCN_24)
        SET_COMP ("idx",                        IDX_1)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_4)
        SET_COMP ("arfcn",                      ARFCN_124)
        SET_COMP ("idx",                        IDX_2)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_5)
        SET_COMP ("arfcn",                      ARFCN_115)
        SET_COMP ("idx",                        IDX_3)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_6)
        SET_COMP ("arfcn",                      ARFCN_531)
        SET_COMP ("idx",                        IDX_4)
ENDSTRUCT
BEGIN_PSTRUCT ("arfcn_idx",                     ARFCN_IDX_VALUE_7)
        SET_COMP ("arfcn",                      ARFCN_527)
        SET_COMP ("idx",                        IDX_6)
ENDSTRUCT
```

```
BEGIN_SHORT_ARRAY_PART(A_ARFCN_11,1)
        11
ENDARRAY
BEGINARRAY_PART(A_BSIC_11,1)
        4
ENDARRAY
BEGINARRAY_PART(A_BSIC_UNK,1)
        MPH_BSIC_UNKNOWN
ENDARRAY
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_0)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_0)
        SET_COMP ("rxlev",                  RXLEV_50)
        SET_COMP ("bsic",                   BSIC_0_NCC2)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_1)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_4)
        SET_COMP ("rxlev",                  RXLEV_41)
        SET_COMP ("bsic",                   BSIC_5_NCC2)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_2)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_2)
        SET_COMP ("rxlev",                  RXLEV_34)
        SET_COMP ("bsic",                   BSIC_3_NCC4)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_3)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_7)
        SET_COMP ("rxlev",                  RXLEV_34)
        SET_COMP ("bsic",                   BSIC_3_NCC3)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_4)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_6)
        SET_COMP ("rxlev",                  RXLEV_31)
        SET_COMP ("bsic",                   RRGRR_INVALID_BSIC)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_5)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_3)
        SET_COMP ("rxlev",                  RXLEV_24)
        SET_COMP ("bsic",                   BSIC_6_NCC2)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_6)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_2)
        SET_COMP ("rxlev",                  RXLEV_50)
        SET_COMP ("bsic",                   BSIC_3_NCC4)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_7)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_0)
        SET_COMP ("rxlev",                  RXLEV_41)
        SET_COMP ("bsic",                   BSIC_0_NCC2)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_8)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_3)
        SET_COMP ("rxlev",                  RXLEV_34)
        SET_COMP ("bsic",                   BSIC_6_NCC2)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_9)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_1)
        SET_COMP ("rxlev",                  RXLEV_34)
        SET_COMP ("bsic",                   RRGRR_INVALID_BSIC)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_1_10)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_2)
        SET_COMP ("rxlev",                  RXLEV_50)
        SET_COMP ("bsic",                   BSIC_3_NCC3)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_2_0)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_6)
        SET_COMP ("rxlev",                  RXLEV_31)
        SET_COMP ("bsic",                   BSIC_3_NCC2)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_2_1)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_5)
        SET_COMP ("rxlev",                  RXLEV_24)
        SET_COMP ("bsic",                   BSIC_0_NCC4)
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_0)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_0)
        SET_COMP ("rxlev",                  RXLEV_50)
        SKIP_COMP ("bsic")
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_1)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_4)
        SET_COMP ("rxlev",                  RXLEV_41)
        SKIP_COMP ("bsic")
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_2)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_2)
        SET_COMP ("rxlev",                  RXLEV_34)
        SKIP_COMP ("bsic")
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_3)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_7)
        SET_COMP ("rxlev",                  RXLEV_34)
        SKIP_COMP ("bsic")
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_4)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_6)
        SET_COMP ("rxlev",                  RXLEV_31)
        SKIP_COMP ("bsic")
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_5)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_3)
        SET_COMP ("rxlev",                  RXLEV_24)
        SKIP_COMP ("bsic")
ENDSTRUCT
BEGIN_PSTRUCT ("xmeas_res",                 XMEAS_RES_VALUE_3_6)
        SET_COMP ("arfcn_idx",              ARFCN_IDX_VALUE_5)
        SET_COMP ("rxlev",                  RXLEV_24)
        SKIP_COMP ("bsic")
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("xmeas_res",                     XMEAS_RES_VALUE_3_7)
        SET_COMP ("arfcn_idx",                  ARFCN_IDX_VALUE_1)
        SET_COMP ("rxlev",                      RXLEV_17)
        SKIP_COMP ("bsic")
ENDSTRUCT

BEGIN_PSTRUCT ("ncells",  NCELLS_1)
        SET_COMP ("no_of_ncells",               1)
        SET_COMP ("arfcn",                      ARFCN_NC1)
        SET_COMP ("rx_lev",                     RX_LEV_NC1)
        SET_COMP ("bsic",                       BSIC_NC1)
        SET_COMP ("time_alignmt",               TIME_ALIGNMENT_NC1)
        SET_COMP ("frame_offset",       FRAME_OFFSET_NC1)
ENDSTRUCT
BEGIN_PSTRUCT ("ncells",  NCELLS_11)
        SET_COMP ("no_of_ncells",               1)
        SET_COMP ("arfcn",                      A_ARFCN_11)
        SKIP_COMP ("rx_lev")
        SET_COMP ("bsic",                       A_BSIC_11)
        SKIP_COMP ("time_alignmt")
        SKIP_COMP ("frame_offset")
ENDSTRUCT
BEGIN_PSTRUCT ("ncells",  NCELLS_1_UNK)
        SET_COMP ("no_of_ncells",               1)
        SET_COMP ("arfcn",                      A_ARFCN_11)
        SKIP_COMP ("rx_lev")
        SET_COMP ("bsic",                       A_BSIC_UNK)
        SKIP_COMP ("time_alignmt")
        SKIP_COMP ("frame_offset")
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",   ADDED_FREQ_1)
        SET_COMP ("arfcn",              ARFCN_3)
        SET_COMP ("bsic",               5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_02)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",   ADDED_FREQ_2)
        SET_COMP ("arfcn",              ARFCN_X20)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_03)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",   ADDED_FREQ_3)
        SET_COMP ("arfcn",              ARFCN_X30)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_01)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",   ADDED_FREQ_4)
        SET_COMP ("arfcn",              ARFCN_X40)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_04)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_5)
        SET_COMP ("arfcn",              ARFCN_X43)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_01)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_6)
        SET_COMP ("arfcn",              0xFFFF)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_01)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_7)
        SET_COMP ("arfcn",              ARFCN_3)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_05)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_8)
        SET_COMP ("arfcn",              ARFCN_X20)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_05)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_9)
        SET_COMP ("arfcn",              ARFCN_X30)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_05)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_10)
        SET_COMP ("arfcn",              ARFCN_X40)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_05)
ENDSTRUCT
BEGIN_PSTRUCT ("add_freq_list",    ADDED_FREQ_11)
        SET_COMP ("arfcn",              ARFCN_X43)
        SET_COMP ("bsic",               BSIC_5)
        SET_COMP ("v_cr_par",           1)
        SET_COMP ("cr_par",             CR_PAR_05)
ENDSTRUCT
BEGIN_PSTRUCT ("rm_freq_list",     REMOVED_FREQ_1)
        SET_COMP ("arfcn",              0xFFFF)
        SET_COMP ("bsic",               BSIC_5)
ENDSTRUCT

BEGIN_PSTRUCT ("ncells",  NCELLS_EMPTY)
        SET_COMP ("no_of_ncells",               0)
        SKIP_COMP ("arfcn")
        SKIP_COMP ("rx_lev")
        SKIP_COMP ("bsic")
        SKIP_COMP ("time_alignmt")
        SKIP_COMP ("frame_offset")
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("page_mode", PAGING_NORMAL)
        SET_COMP ("pm",             PAGING_NORM)
ENDSTRUCT
BEGIN_MSTRUCT ("start_time", START_TIME_1)
        SET_COMP ("t1",                 T1)
        SET_COMP ("t3",                 T3)
        SET_COMP ("t2",                 T2)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_desc", CELL_DESC_1)
        SET_COMP ("bcch_arfcn_hi",      BCCH_ARFCN_CELL_DESC_HI)
        SET_COMP ("ncc",                BSIC_CELL_DESC_NCC)
        SET_COMP ("bcc",                BSIC_CELL_DESC_BCC)
        SET_COMP ("bcch_arfcn_lo",      BCCH_ARFCN_CELL_DESC_LO)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_desc", CELL_DESC_2)
        SET_COMP ("bcch_arfcn_hi",      BCCH_ARFCN_CELL_DESC_HI_2)
        SET_COMP ("ncc",                BSIC_CELL_DESC_NCC)
        SET_COMP ("bcc",                BSIC_CELL_DESC_BCC_2)
        SET_COMP ("bcch_arfcn_lo",      BCCH_ARFCN_CELL_DESC_LO_2)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_desc", CELL_DESC_BAD_BCCH)
        SET_COMP ("bcch_arfcn_hi",      BCCH_ARFCN_CELL_DESC_HI_BAD)
        SET_COMP ("ncc",                BSIC_CELL_DESC_NCC)
        SET_COMP ("bcc",                BSIC_CELL_DESC_BCC)
        SET_COMP ("bcch_arfcn_lo",      BCCH_ARFCN_CELL_DESC_LO_BAD)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_opt_bcch", CELL_OPT_BCCH_1)
        SET_COMP ("pow_ctrl",           POW_CTRL_YES)
        SET_COMP ("dtx_b",              B_DTX_SHALL_USE)
        SET_COMP ("rlt",                RLT_64)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_select", CELL_SELECT_1)
        SET_COMP ("cell_resel_hyst",    CELL_HYST_4_DB)
        SET_COMP ("ms_txpwr_max_cch",   MS_TXPWR_MAX_CCH_02)
        SET_COMP ("acs",                ACS_USE_SI4)
        SET_COMP ("neci",               NECI_YES)
        SET_COMP ("rxlev_access_min",   RX_LEV_ACCESS_MIN_16)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_select", CELL_SELECT_2)
        SET_COMP ("cell_resel_hyst",    CELL_HYST_4_DB)
        SET_COMP ("ms_txpwr_max_cch",   MS_TXPWR_MAX_CCH_02)
        SET_COMP ("acs",                ACS_USE_SI4)
        SET_COMP ("neci",               NECI_YES)
        SET_COMP ("rxlev_access_min",   RX_LEV_ACCESS_MIN_22)
ENDSTRUCT
BEGIN_MSTRUCT ("cell_select", CELL_SELECT_3)
        SET_COMP ("cell_resel_hyst",    CELL_HYST_4_DB)
        SET_COMP ("ms_txpwr_max_cch",   MS_TXPWR_MAX_CCH_02)
        SET_COMP ("acs",                ACS_USE_SI4)
        SET_COMP ("neci",               NECI_YES)
        SET_COMP ("rxlev_access_min",   RX_LEV_ACCESS_MIN_24)
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_SDCCH)
        SET_COMP ("chan_type",          SDCCH_4_S1)
```

```
                SET_COMP ("tn",                 TN_3)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",               MAIO_12)
                SET_COMP ("hsn",                HSN_14)
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_SDCCH8)    /* for TC740 according to ETSI 11.10 */
                SET_COMP ("chan_type",          SDCCH_8_S3)
                SET_COMP ("tn",                 TN_3)       /* Chosen arbitrarily */
                SET_COMP ("tsc",                TSC_2)      /* dito */
                SET_COMP ("hop",                HOP_NO)
                SET_COMP ("arfcn",              ARFCN_30)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_SDCCH8_2)
                SET_COMP ("chan_type",          SDCCH_8_S7) /* same ch., different TDMA */
                SET_COMP ("tn",                 TN_4)       /* equal to last one */
                SET_COMP ("tsc",                TSC_2)      /* dito */
                SET_COMP ("hop",                HOP_NO)     /* dito */
                SET_COMP ("arfcn",              ARFCN_30)   /* dito */
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_SDCCH2)
                SET_COMP ("chan_type",          SDCCH_4_S1)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_NO)
                SET_COMP ("arfcn",              ARFCN_X73)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_BAD)
                SET_COMP ("chan_type",          SDCCH_4_S1)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_NO)
                SET_COMP ("arfcn",              ARFCN_800)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_FACCH2)
                SET_COMP ("chan_type",          TCH_F)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_NO)
                SET_COMP ("arfcn",              ARFCN_X73)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_HALFRATE)
                SET_COMP ("chan_type",          TCH_H_S1)
                SET_COMP ("tn",                 TN_4)
```

```
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_NO)
                SET_COMP ("arfcn",              ARFCN_X73)
                SKIP_COMP ("maio")
                SKIP_COMP ("hsn")
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_FACCH3)
                SET_COMP ("chan_type",          TCH_F)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",               MAIO_12)
                SET_COMP ("hsn",                HSN_14)
ENDSTRUCT
BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_FACCH4)
                SET_COMP ("chan_type",          TCH_F)
                SET_COMP ("tn",                 TN_4)
                SET_COMP ("tsc",                TSC_2)
                SET_COMP ("hop",                HOP_YES)
                SKIP_COMP ("arfcn")
                SET_COMP ("maio",               MAIO_2)
                SET_COMP ("hsn",                HSN_4)
ENDSTRUCT
BEGIN_MSTRUCT ("chan_needed", CHANNEL_NEEDED_1)
                SET_COMP ("cn2",                CN_ANY_CHAN)
                SET_COMP ("cn1",                CN_ANY_CHAN)
ENDSTRUCT
BEGIN_MSTRUCT ("ciph_key_num", CKSN_RESERVED)
                SET_COMP ("key_seq",            CKSN_7)
ENDSTRUCT
BEGIN_MSTRUCT ("ciph_key_num", CKSN_NOT_PRES_STRUCT)
                SET_COMP ("key_seq",            NOT_PRESENT_8BIT)
ENDSTRUCT
BEGIN_MSTRUCT ("ciph_mode_set", CIPH_MODE_OFF)
                SET_COMP ("algo_ident",         ALGO_0)
                SET_COMP ("sc",                 CIPH_OFF)
ENDSTRUCT
BEGIN_MSTRUCT ("ciph_mode_set", CIPH_MODE_ON)
                SET_COMP ("algo_ident",         ALGO_0)
                SET_COMP ("sc",                 CIPH_ON)
ENDSTRUCT
BEGIN_MSTRUCT ("ciph_res", CIPH_RESP_NO_IMEI)
                SET_COMP ("cr",                 INC_IMEISV_NO)
ENDSTRUCT
BEGIN_MSTRUCT ("ciph_res", CIPH_RESP_WITH_IMEI)
                SET_COMP ("cr",                 INC_IMEISV_YES)
ENDSTRUCT
BEGIN_MSTRUCT ("ctrl_chan_desc", CTRL_CHAN_DESC_1)
                SET_COMP ("att",                CCD_ATT_YES)
                SET_COMP ("bs_ag_blks_res",     BS_AG_BLKS_RES_5)
                SET_COMP ("ccch_conf",          CCD_CCCH_1_NOT_COMB)
                SET_COMP ("bs_pa_mfrms",        BS_PA_MFRMS_2)
                SET_COMP ("t3212",              T3212_36_MIN)
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("freq_chan_seq",FREQ_CHAN_SEQ_1)
                SET_COMP ("low_arfcn",              FREQ_CHAN_SEQ_1_LOW_ARFCN)
                SET_COMP ("inc_skip",               FREQ_CHAN_SEQ_1_INC_SKIP)
ENDSTRUCT
BEGIN_MSTRUCT ("loc_area_ident", LOC_AREA_IDENT_122_2147)
                SET_COMP ("mcc",                    MCC_122)
                SET_COMP ("mnc",                    MNC_33)
                SET_COMP ("lac",                    LAC_2147)
ENDSTRUCT
BEGIN_MSTRUCT ("loc_area_ident", LOC_AREA_IDENT_122_2147_V)
                SET_COMP ("mcc",                    MCC_122)
                SET_COMP ("mnc",                    MNC_325)
                SET_COMP ("lac",                    LAC_2147)
ENDSTRUCT
BEGIN_MSTRUCT ("loc_area_ident", LOC_AREA_IDENT_123_2147)
                SET_COMP ("mcc",                    MCC_123)
                SET_COMP ("mnc",                    MNC_325)
                SET_COMP ("lac",                    LAC_2147)
ENDSTRUCT
BEGIN_MSTRUCT ("loc_area_ident", LOC_AREA_IDENT_123_32_2147)
                SET_COMP ("mcc",                    MCC_123)
                SET_COMP ("mnc",                    MNC_32)
                SET_COMP ("lac",                    LAC_2147)
ENDSTRUCT
BEGIN_MSTRUCT ("loc_area_ident", LOC_AREA_IDENT_123_2147_V)
                SET_COMP ("mcc",                    MCC_123)
                SET_COMP ("mnc",                    MNC_33)
                SET_COMP ("lac",                    LAC_2147)
ENDSTRUCT
BEGIN_MSTRUCT ("loc_area_ident", LOC_AREA_IDENT_123_2148)
                SET_COMP ("mcc",                    MCC_123)
                SET_COMP ("mnc",                    MNC_325)
                SET_COMP ("lac",                    LAC_2148)
ENDSTRUCT
BEGIN_PSTRUCT ("ncells", NCELLS_2_CR_CANDIDATE)
                SET_COMP ("no_of_ncells",           2)
                SET_COMP ("arfcn",                  ARFCN_NC)
                SET_COMP ("rx_lev",                 RX_LEV_NC_CR)
                SET_COMP ("bsic",                   BSIC_NC)
                SET_COMP ("time_alignmt",           TIME_ALIGNMENT_NC)
                SET_COMP ("frame_offset",           FRAME_OFFSET_NC)
ENDSTRUCT
/* not needed
*/
BEGIN_MSTRUCT ("meas_result", MEAS_RESULT_NCELL_0)
                SET_COMP ("ba_used",                BA_0)
                SET_COMP ("dtx_used",               DTX_NOT_USED)
                SET_COMP ("rxlev_full",             RX_LEV_20)
                SET_COMP ("meas_valid",             MEAS_VALID_YES)
                SET_COMP ("rxlev_sub",              RX_LEV_20)
                SET_COMP ("rxqual_full",            RX_QUAL_1)
                SET_COMP ("rxqual_sub",             RX_QUAL_1)
                SET_COMP ("num_ncell",              NO_NCELLS)
                SKIP_COMP ("ncell")
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("meas_result", MEAS_RESULT_NCELL_6_FTA)
                SET_COMP ("ba_used",            BA_1)
                SET_COMP ("dtx_used",           DTX_NOT_USED)
                SET_COMP ("rxlev_full",         RX_LEV_20)
                SET_COMP ("meas_valid",         MEAS_VALID_YES)
                SET_COMP ("rxlev_sub",          RX_LEV_20)
                SET_COMP ("rxqual_full",        RX_QUAL_1)
                SET_COMP ("rxqual_sub",         RX_QUAL_1)
                SET_COMP ("num_ncell",          SIX_NCELLS)
                SKIP_COMP ("ncell")
ENDSTRUCT
BEGIN_MSTRUCT ("meas_result", MEAS_RESULT_NCELL_FTA_B)
                SET_COMP ("ba_used",            BA_1)
                SET_COMP ("dtx_used",           DTX_NOT_USED)
                SET_COMP ("rxlev_full",         RX_LEV_20)
                SET_COMP ("meas_valid",         MEAS_VALID_YES)
                SET_COMP ("rxlev_sub",          RX_LEV_20)
                SET_COMP ("rxqual_full",        RX_QUAL_1)
                SET_COMP ("rxqual_sub",         RX_QUAL_1)
                SET_COMP ("num_ncell",          SIX_NCELLS)
                SKIP_COMP ("ncell")
ENDSTRUCT
BEGIN_MSTRUCT ("meas_result", MEAS_RESULT_NCELL_FTA_C)
                SET_COMP ("ba_used",            BA_1)
                SET_COMP ("dtx_used",           DTX_NOT_USED)
                SET_COMP ("rxlev_full",         RX_LEV_20)
                SET_COMP ("meas_valid",         MEAS_VALID_YES)
                SET_COMP ("rxlev_sub",          RX_LEV_20)
                SET_COMP ("rxqual_full",        RX_QUAL_1)
                SET_COMP ("rxqual_sub",         RX_QUAL_1)
                SET_COMP ("num_ncell",          SIX_NCELLS)
                SKIP_COMP ("ncell")
ENDSTRUCT
BEGIN_MSTRUCT ("meas_result", MEAS_RESULT_NCELL_FTA_D)
                SET_COMP ("ba_used",            BA_1)
                SET_COMP ("dtx_used",           DTX_NOT_USED)
                SET_COMP ("rxlev_full",         RX_LEV_20)
                SET_COMP ("meas_valid",         MEAS_VALID_YES)
                SET_COMP ("rxlev_sub",          RX_LEV_20)
                SET_COMP ("rxqual_full",        RX_QUAL_1)
                SET_COMP ("rxqual_sub",         RX_QUAL_1)
                SET_COMP ("num_ncell",          SIX_NCELLS)
                SKIP_COMP ("ncell")
ENDSTRUCT
BEGIN_MSTRUCT ("meas_result", MEAS_RESULT_NCELL_FTA_E)
                SET_COMP ("ba_used",            BA_1)
                SET_COMP ("dtx_used",           DTX_NOT_USED)
                SET_COMP ("rxlev_full",         RX_LEV_20)
                SET_COMP ("meas_valid",         MEAS_VALID_YES)
                SET_COMP ("rxlev_sub",          RX_LEV_20)
                SET_COMP ("rxqual_full",        RX_QUAL_1)
                SET_COMP ("rxqual_sub",         RX_QUAL_1)
                SET_COMP ("num_ncell",          FOUR_NCELLS)
                SKIP_COMP ("ncell")
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("mob_alloc", MOBILE_ALLOCATION_1)
        SET_COMP ("mac",            MOB_ALLOC_1)
ENDSTRUCT
BEGIN_MSTRUCT ("mob_alloc", MOBILE_ALLOCATION_2)
        SET_COMP ("mac",            MOB_ALLOC_2)
ENDSTRUCT
BEGIN_MSTRUCT ("mob_alloc", MOBILE_ALLOCATION_3)
        SET_COMP ("mac",            MOB_ALLOC_3)
ENDSTRUCT
BEGIN_MSTRUCT ("mob_alloc", MOBILE_ALLOCATION_4)
        SET_COMP ("mac",            MOB_ALLOC_4)
ENDSTRUCT
BEGIN_MSTRUCT ("mob_alloc", MOBILE_ALLOCATION_EGSM)
        SET_COMP ("mac",            MOB_ALLOC_EGSM)
ENDSTRUCT
BEGIN_MSTRUCT ("mob_class_2", MOB_CLASS2)
        SET_COMP ("rev_lev",        PHASE_2)
        SET_COMP ("es_ind",         SUPPORTED)
        SET_COMP ("a5_1",           A5_1_SUPPORTED)
        SET_COMP ("rf_pow_cap",     RF_CLASS_4)
        SET_COMP ("ps",             NOT_SUPPORTED)
        SET_COMP ("ss_screen",      SS_SCREEN_1)
        SET_COMP ("mt_pp_sms",      SUPPORTED)
        SET_COMP ("vbs",            NOT_SUPPORTED)
        SET_COMP ("vgcs",           NOT_SUPPORTED)
        SET_COMP ("egsm",           SUPPORTED)
        SET_COMP ("class3",         SUPPORTED)
        SET_COMP ("lcsva",          NOT_SUPPORTED)
        SET_COMP ("ucs2_treat",     SUPPORTED)
        SET_COMP ("solsa",          NOT_SUPPORTED)
        SET_COMP ("cmsp",           SUPPORTED)
        SET_COMP ("a5_3",           NOT_SUPPORTED)
        SET_COMP ("a5_2",           SUPPORTED)
ENDSTRUCT
BEGIN_MSTRUCT ("measurement", SMS_SM_VAL_1)
        SET_COMP ("sms_val",        SM_4)
        SET_COMP ("sm_val",         SM_5)
ENDSTRUCT
BEGIN_MSTRUCT ("mob_class_3", MOB_CLASS3_900)
        SET_COMP ("mb_value",       MB_DUAL_EXT)
        SET_COMP ("a5_7",           NOT_SUPPORTED)
        SET_COMP ("a5_6",           NOT_SUPPORTED)
        SET_COMP ("a5_5",           NOT_SUPPORTED)
        SET_COMP ("a5_4",           NOT_SUPPORTED)
        SET_COMP ("radio_cap_2",    POWER_CLASS_1)
        SET_COMP ("radio_cap_1",    POWER_CLASS_4)
        SKIP_COMP ("rgsm_class")
        SKIP_COMP ("ms_class")
        SET_COMP ("ucs2_treat",     SUPPORTED)
        SET_COMP ("ext_meas",       NOT_SUPPORTED)
        SKIP_COMP ("measurement")
        SKIP_COMP ("pos_method")
        SKIP_COMP ("edge_ms_class")
        SKIP_COMP ("egde_struct")
        SKIP_COMP ("gsm400_struct")
```

```
                SKIP_COMP ("gsm850_cap")
                SKIP_COMP ("pcs1900_cap")
                SET_COMP ("umts_fdd",           NOT_SUPPORTED)
                SET_COMP ("umts_tdd",           NOT_SUPPORTED)
                SET_COMP ("cdma2000",           NOT_SUPPORTED)
                SKIP_COMP ("dtm_ms")
                SKIP_COMP ("single_band")
ENDSTRUCT
BEGIN_MSTRUCT ("mob_class_3", MOB_CLASS3_900_SHOW)
                SHOW_COMP ("mb_value"                   )
                SHOW_COMP ("a5_7"                       )
                SHOW_COMP ("a5_6"                       )
                SHOW_COMP ("a5_5"                       )
                SHOW_COMP ("a5_4"                       )
                SHOW_COMP ("radio_cap_2"                )
                SHOW_COMP ("radio_cap_1"                )
                SHOW_COMP ("rgsm_class"                 )
                SHOW_COMP ("ms_class"                   )
                SHOW_COMP ("ucs2_treat"                 )
                SHOW_COMP ("ext_meas"                   )
                SHOW_COMP ("measurement"                )
                SHOW_COMP ("pos_method"                 )
                SHOW_COMP ("edge_ms_class"              )
                SHOW_COMP ("egde_struct"                )
                SHOW_COMP ("gsm400_struct"              )
                SHOW_COMP ("gsm850_cap"                 )
                SHOW_COMP ("pcs1900_cap"                )
                SHOW_COMP ("umts_fdd"                   )
                SHOW_COMP ("umts_tdd"                   )
                SHOW_COMP ("cdma2000"                   )
                SHOW_COMP ("dtm_ms"                     )
                SHOW_COMP ("single_band"                )
ENDSTRUCT
BEGIN_MSTRUCT ("mob_ident", MOBILE_IDENTITY_IMSI2)
                SET_COMP ("ident_type",         TYPE_IMSI)
                SET_COMP ("odd_even",           EVEN)
                SET_COMP ("ident_dig",          M_IMSI_08158912)
                SKIP_COMP ("tmsi_1")
ENDSTRUCT
BEGIN_MSTRUCT ("mob_ident", MOBILE_IDENTITY_IMSI_HPLMN)
                SET_COMP ("ident_type",         TYPE_IMSI)
                SET_COMP ("odd_even",           ODD)
                SET_COMP ("ident_dig",          M_IMSI_1233247114912)
                SKIP_COMP ("tmsi_1")
ENDSTRUCT
BEGIN_MSTRUCT ("mob_ident", MOBILE_IDENTITY_IMSI_TEST)
                SET_COMP ("ident_type",         TYPE_IMSI)
                SET_COMP ("odd_even",           ODD)
                SET_COMP ("ident_dig",          M_IMSI_0010147114912)
                SKIP_COMP ("tmsi_1")
ENDSTRUCT
BEGIN_MSTRUCT ("mob_ident", MOBILE_IDENTITY_IMEISV)
                SET_COMP ("ident_type",         ID_TYPE_IMEISV)
                SET_COMP ("odd_even",           EVEN)
                SET_COMP ("ident_dig",          M_IMEI)
```

```
                SKIP_COMP ("tmsi_1")
ENDSTRUCT
BEGIN_MSTRUCT ("mob_ident", MOBILE_IDENTITY_TMSI)
                SET_COMP ("ident_type",         TYPE_TMSI)
                SET_COMP ("odd_even",           EVEN)
                SKIP_COMP ("ident_dig")
                SET_COMP ("tmsi_1",             TMSI_1)
ENDSTRUCT
BEGIN_MSTRUCT ("pow_cmd", POWER_COMMAND_05)
                SET_COMP ("pow",                POW_05)
ENDSTRUCT
BEGIN_MSTRUCT ("pow_cmd_access", POW_05_HO)
                SET_COMP ("atc",            POW_05_HO_ATC)
                SET_COMP ("pow",            POW_05_HO_POW)
ENDSTRUCT
BEGIN_MSTRUCT ("rach_ctrl", RACH_CTRL_1)
                SET_COMP ("max_retrans",        MAX_RETRANS_1)
                SET_COMP ("tx_integer",         SPREAD_TRANS_3)
                SET_COMP ("cell_bar_access",    BARRED_NO)
                SET_COMP ("re",                 REESTAB_YES)
                SET_COMP ("ac",                 ACC_0005)
ENDSTRUCT
BEGIN_MSTRUCT ("rach_ctrl", RACH_CTRL_2)
                SET_COMP ("max_retrans",        MAX_RETRANS_1)
                SET_COMP ("tx_integer",         SPREAD_TRANS_3)
                SET_COMP ("cell_bar_access",    BARRED_YES)
                SET_COMP ("re",                 REESTAB_YES)
                SET_COMP ("ac",                 ACC_0005)
ENDSTRUCT
BEGIN_MSTRUCT ("rach_ctrl", RACH_CTRL_3)
                SET_COMP ("max_retrans",        MAX_RETRANS_1)
                SET_COMP ("tx_integer",         SPREAD_TRANS_3)
                SET_COMP ("cell_bar_access",    BARRED_NO)
                SET_COMP ("re",                 REESTAB_YES)
                SET_COMP ("ac",                 ACC_0005)
ENDSTRUCT

BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_1)
                SET_COMP ("ra",                 RA_1)
                SET_COMP ("t1",                 T1)
                SET_COMP ("t3",                 T3)
                SET_COMP ("t2",                 T2)
ENDSTRUCT
BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_2)
                SET_COMP ("ra",                 RA_2)
                SET_COMP ("t1",                 T1)
                SET_COMP ("t3",                 T3)
                SET_COMP ("t2",                 T2)
ENDSTRUCT
BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_3)
                SET_COMP ("ra",                 RA_3)  /* 0x0b */
                SET_COMP ("t1",                 T1)
                SET_COMP ("t3",                 T3)
                SET_COMP ("t2",                 T2)
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_4)   /*for TC 730 by MSE*/
                SET_COMP ("ra",                  0x7B)
                SET_COMP ("t1",                  T1)
                SET_COMP ("t3",                  T3)
                SET_COMP ("t2",                  T2)
ENDSTRUCT
BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_4A)  /*for TC 871 by MSE*/
                SET_COMP ("ra",                  0x9B)
                SET_COMP ("t1",                  T1)
                SET_COMP ("t3",                  T3)
                SET_COMP ("t2",                  T2)
ENDSTRUCT
BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_5)   /*for TC 803 */
                SET_COMP ("ra",                  0xFB)
                SET_COMP ("t1",                  T1)
                SET_COMP ("t3",                  T3)
                SET_COMP ("t2",                  T2)
ENDSTRUCT

BEGIN_MSTRUCT ("req_ref", REQUEST_REFERENCE_FAIL)
                SET_COMP ("ra",                  0xFF)
                SET_COMP ("t1",                  T1)
                SET_COMP ("t3",                  T3)
                SET_COMP ("t2",                  T2)
ENDSTRUCT
BEGIN_PSTRUCT ("rr_cell_env", RR_CELL_ENV_2147)
                SET_COMP ("plmn",                PLMN_ID_123)
                SET_COMP ("lac",                 LAC_2147)
                SET_COMP ("cid",                 CELL_IDENT_3748)
ENDSTRUCT
BEGIN_PSTRUCT ("rr_cell_env", RR_CELL_ENV_2147_32)
                SET_COMP ("plmn",                PLMN_ID_123_32)
                SET_COMP ("lac",                 LAC_2147)
                SET_COMP ("cid",                 CELL_IDENT_3748)
ENDSTRUCT
BEGIN_PSTRUCT ("rr_cell_env", RR_CELL_ENV_2147_V)
                SET_COMP ("plmn",                PLMN_ID_123_V)
                SET_COMP ("lac",                 LAC_2147)
                SET_COMP ("cid",                 CELL_IDENT_3748)
ENDSTRUCT
BEGIN_PSTRUCT ("rr_cell_env", RR_CELL_ENV_2149)
                SET_COMP ("plmn",                PLMN_ID_123)
                SET_COMP ("lac",                 LAC_2147)
                SET_COMP ("cid",                 CELL_IDENT_3749)
ENDSTRUCT
BEGIN_MSTRUCT ("synch_ind", SYNCH_IND_1)
                SET_COMP ("nci",                 NCI_1)
                SET_COMP ("rot",                 ROT_1)
                SET_COMP ("si",                  SYI_PRE_SYNCH)
ENDSTRUCT
BEGIN_MSTRUCT ("synch_ind", SYNCH_IND_0)
                SET_COMP ("nci",                 NCI_0)
                SET_COMP ("rot",                 ROT_0)
                SET_COMP ("si",                  SYI_NON_SYNCH)
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("time_advance", TIMING_ADVANCE_27)
        SET_COMP ("ta",                 TIME_ADV_27)
ENDSTRUCT
BEGIN_MSTRUCT ("time_advance", TIMING_ADVANCE_10)
        SET_COMP ("ta",                 TIME_ADV_10)
ENDSTRUCT

BEGIN_MSTRUCT ("opt_sel_par", SI4_OPT_SEL)
        SET_COMP ("cell_bar_qual",          CBQ_YES)
        SET_COMP ("cell_resel_offs",        0x00)
        SET_COMP ("temp_offs",              0x00)
        SET_COMP ("penalty_time",           0x00)
ENDSTRUCT
BEGIN_MSTRUCT ("lsa_param", SI4_LSA_PARA)
        SET_COMP ("prio_thr",               0x00)
        SET_COMP ("lsa_offs",               0x00)
        SET_COMP ("solsa_exc_acc",          SOLSA_EXCLUSIVE_YES)
        SKIP_COMP ("opt_mcc"                )
        SET_COMP ("mnc",                    MNC_33)
ENDSTRUCT
BEGIN_MSTRUCT ("lsa_id_info", SI4_LSA_ID)
        SET_COMP ("lsa_id",                 0xAA)
        SKIP_COMP ("lsa_id_add"             )
ENDSTRUCT
BEGIN_MSTRUCT ("gprs_indic", GPRS_INDIC_TC701)
        SET_COMP ("ra_color",               0x02)
        SET_COMP ("si13_pos",               SI13_ON_BCCH_NORM)
ENDSTRUCT
BEGIN_MSTRUCT ("gprs_indic", GPRS_INDIC_TC703)
        SET_COMP ("ra_color",               0x02)
        SET_COMP ("si13_pos",               SI13_ON_BCCH_EXT)
ENDSTRUCT
BEGIN_MSTRUCT ("si3_rest_oct", SI3_REST_NOGPRS_TC206)
        SKIP_COMP ("opt_sel_par"            )
        SET_COMP ("pow_offs",               POW_OFFS_0_DB)      /* according to ETSI 11.10 */
        SKIP_COMP ("si2ter_ind"             )                   /* no si2ter exists */
        SET_COMP ("es_ind_tag",             ES_IND_ENABLE)      /* Early Classmark enabled */
        SKIP_COMP ("if_and_where"           )
        SKIP_COMP ("gprs_indic"             )
ENDSTRUCT
BEGIN_MSTRUCT ("si3_rest_oct", SI3_REST_GPRS_TC701)
        SKIP_COMP ("opt_sel_par"            )
        SET_COMP ("pow_offs",               POW_OFFS_0_DB)      /* according to ETSI 11.10 */
        SKIP_COMP ("si2ter_ind"             )                   /* no si2ter exists */
        SET_COMP ("es_ind_tag",             ES_IND_ENABLE)      /* Early Classmark enabled */
        SKIP_COMP ("if_and_where"           )
        SET_COMP ("gprs_indic",             GPRS_INDIC_TC701)
ENDSTRUCT
BEGIN_MSTRUCT ("si3_rest_oct", SI3_REST_GPRS_TC703)
        SKIP_COMP ("opt_sel_par"            )
        SET_COMP ("pow_offs",               POW_OFFS_0_DB)
        SKIP_COMP ("si2ter_ind"             )
        SKIP_COMP ("es_ind_tag"             )
        SKIP_COMP ("if_and_where"           )
```

```
        SET_COMP ("gprs_indic",                 GPRS_INDIC_TC703)
ENDSTRUCT
BEGIN_MSTRUCT ("si4_rest_oct", SI4_REST_NOGPRS_TC206)
        SET_COMP ("opt_sel_par",                SI4_OPT_SEL)
        SET_COMP ("pow_offs",                   POW_OFFS_0_DB)
        SKIP_COMP ("gprs_indic"                 )
        SET_COMP ("lsa_param",                  SI4_LSA_PARA)
        SET_COMP ("cell_ident",                 CELL_IDENT_3748)
        SET_COMP ("lsa_id_info",                SI4_LSA_ID)
ENDSTRUCT
BEGIN_MSTRUCT ("si4_rest_oct", SI4_REST_GPRS_TC701)
        SET_COMP ("opt_sel_par",                SI4_OPT_SEL)
        SET_COMP ("pow_offs",                   POW_OFFS_0_DB)
        SET_COMP ("gprs_indic",                 GPRS_INDIC_TC701)
        SET_COMP ("lsa_param",                  SI4_LSA_PARA)
        SET_COMP ("cell_ident",                 CELL_IDENT_3748)
        SET_COMP ("lsa_id_info",                SI4_LSA_ID)
ENDSTRUCT
BEGIN_MSTRUCT ("si4_rest_oct", SI4_REST_GPRS_TC706)
        SET_COMP ("opt_sel_par",                SI4_OPT_SEL)
        SET_COMP ("pow_offs",                   POW_OFFS_0_DB)
        SET_COMP ("gprs_indic",                 GPRS_INDIC_TC701)
        SET_COMP ("lsa_param",                  SI4_LSA_PARA)
        SET_COMP ("cell_ident",                 CELL_IDENT_3749)
        SET_COMP ("lsa_id_info",                SI4_LSA_ID)
ENDSTRUCT
BEGIN_MSTRUCT ("si4_rest_oct", SI4_REST_GPRS_TC703)
        SET_COMP ("opt_sel_par",                SI4_OPT_SEL)
        SET_COMP ("pow_offs",                   POW_OFFS_0_DB)
        SET_COMP ("gprs_indic",                 GPRS_INDIC_TC703)
        SET_COMP ("lsa_param",                  SI4_LSA_PARA)
        SET_COMP ("cell_ident",                 CELL_IDENT_3748)
        SET_COMP ("lsa_id_info",                SI4_LSA_ID)
ENDSTRUCT
BEGIN_MSTRUCT ("si1_rest_oct", SI1_REST_GPRS_TC701)
        SET_COMP ("nch_position",               NCH_1BLK_NR0)
ENDSTRUCT
BEGIN_MSTRUCT ("pbcch_des", SI13_PBCCH_DES)     /*values according to ETSI 11.10 */
        SET_COMP ("pb",                         0x06)
        SET_COMP ("tsc",                        0x02)           /* range: 0..7 */
        SET_COMP ("tn",                         0x03)           /* range: 0..7 */
        SET_COMP ("flag",                       0x00)
        SET_COMP ("flag2",                      0x01)
        SET_COMP ("arfcn",                      0x20)
        SKIP_COMP ("maio"                       )
ENDSTRUCT
BEGIN_MSTRUCT ("si13_info", SI13_INFO_TC701)
        SET_COMP ("bcch_cm",                    0x00)
        SET_COMP ("si_cf",                      SI_CF_UNSPEC)
        SET_COMP ("flag",                       0)
        SKIP_COMP ("si13_cm"                    )
        SKIP_COMP ("gprs_ma")
        SET_COMP ("flag1",                      1)
        SKIP_COMP ("rac"                        )
        SKIP_COMP ("spgc"                       )
```

```
        SKIP_COMP ("prio_acc_thr"                   )
        SKIP_COMP ("nco"                            )
        SKIP_COMP ("gprs_opt"                       )
        SKIP_COMP ("alpha"                          )
        SKIP_COMP ("t_avg_w"                        )
        SKIP_COMP ("t_avg_t"                        )
        SKIP_COMP ("pc_meas_chan"                   )
        SKIP_COMP ("n_avg_i"                        )
        SET_COMP ("psi1_period",                    7)
        SET_COMP ("pbcch_des",                      SI13_PBCCH_DES)
ENDSTRUCT
BEGIN_MSTRUCT ("si13_info", SI13_INFO_TC701_BCCH)
        SET_COMP ("bcch_cm",                        0x00)
        SET_COMP ("si_cf",                          SI_CF_UNSPEC)
        SET_COMP ("flag",                           0)
        SKIP_COMP ("si13_cm"                        )
        SKIP_COMP ("gprs_ma")
        SET_COMP ("flag1",                          0)
        SKIP_COMP ("rac"                            )
        SKIP_COMP ("spgc"                           )
        SKIP_COMP ("prio_acc_thr"                   )
        SKIP_COMP ("nco"                            )
        SKIP_COMP ("gprs_opt"                       )
        SKIP_COMP ("alpha"                          )
        SKIP_COMP ("t_avg_w"                        )
        SKIP_COMP ("t_avg_t"                        )
        SKIP_COMP ("pc_meas_chan"                   )
        SKIP_COMP ("n_avg_i"                        )
        SKIP_COMP ("psi1_period")
        SKIP_COMP ("pbcch_des")
ENDSTRUCT
BEGIN_MSTRUCT ("si13_rest_oct", SI13_REST_GPRS_TC701)
        SET_COMP ("si13_info",                      SI13_INFO_TC701)
ENDSTRUCT
BEGIN_MSTRUCT ("si13_rest_oct", SI13_REST_GPRS_TC701_BCCH)
        SET_COMP ("si13_info",                      SI13_INFO_TC701_BCCH)
ENDSTRUCT
BEGIN_PSTRUCT ("bcch_info", BCCH_INFO_TC701)
        SET_COMP ("v_bcch",                         V_BCCH_PRES)
        SET_COMP ("bcch",                           BCCH_BITMAP)
ENDSTRUCT
BEGIN_PSTRUCT ("si_states", SI_STATES_TC702)
        SET_COMP ("si1_state",                      SI1_RECEIVED)
        SET_COMP ("si3_state",                      SI3_RECEIVED)
        SET_COMP ("si13_state",                     SI13_RECEIVED)
ENDSTRUCT
BEGIN_PSTRUCT ("si_states", SI_STATES_ONLY_SI13)
        SET_COMP ("si1_state",                      SI1_NOT_RECEIVED)
        SET_COMP ("si3_state",                      SI3_NOT_RECEIVED)
        SET_COMP ("si13_state",                     SI13_RECEIVED)
ENDSTRUCT
BEGIN_PSTRUCT ("serving_cell_info", SERVING_CELL_TC105)
        SET_COMP ("gprs_rxlev_access_min",          0x16)
        SET_COMP ("gprs_ms_txpwr_max_cch",          0x02)
        SET_COMP("bcch_arfcn",                      ARFCN_X43)
```

```
        SET_COMP("bcch_bsic",                   BSIC_5)
        SET_COMP ("ac_class",                   0x40)
        SET_COMP ("pwr_offset",                 0)
        SET_COMP ("plmn_forb",                  TRUE)
        SET_COMP ("la_forb",                    FALSE)
        SET_COMP ("rr_cell_env",                RR_CELL_ENV_2147)
        SET_COMP("limited",                     LIMITED_YES)
        SET_COMP("bs_pa_mfrms",                 BS_PA_MFRMS_2)
ENDSTRUCT
BEGIN_PSTRUCT ("serving_cell_info", SERVING_CELL_TC205)
        SET_COMP ("gprs_rxlev_access_min",      0x16)
        SET_COMP ("gprs_ms_txpwr_max_cch",      0x02)
        SET_COMP("bcch_arfcn",                  ARFCN_X20)
        SET_COMP("bcch_bsic",                   BSIC_5)
        SET_COMP ("ac_class",                   0x40)
        SET_COMP ("pwr_offset",                 0)
        SET_COMP ("plmn_forb",                  FALSE)
        SET_COMP ("la_forb",                    FALSE)
        SET_COMP ("rr_cell_env",                RR_CELL_ENV_2149)
        SET_COMP("limited",                     LIMITED_NO)
        SET_COMP("bs_pa_mfrms",                 BS_PA_MFRMS_2)
ENDSTRUCT
BEGIN_PSTRUCT ("serving_cell_info", SERVING_CELL_TC702)
        SET_COMP ("gprs_rxlev_access_min",      0x16)
        SET_COMP ("gprs_ms_txpwr_max_cch",      0x02)
        SET_COMP("bcch_arfcn",                  ARFCN_X43)
        SET_COMP("bcch_bsic",                   BSIC_5)
        SET_COMP ("ac_class",                   0x40)
        SET_COMP ("pwr_offset",                 0)
        SET_COMP ("plmn_forb",                  FALSE)
        SET_COMP ("la_forb",                    FALSE)
        SET_COMP ("rr_cell_env",                RR_CELL_ENV_2147)
        SET_COMP("limited",                     LIMITED_NO)
        SET_COMP("bs_pa_mfrms",                 BS_PA_MFRMS_2)
ENDSTRUCT
BEGIN_PSTRUCT ("serving_cell_info", SERVING_CELL_TC702_32)
        SET_COMP ("gprs_rxlev_access_min",      0x16)
        SET_COMP ("gprs_ms_txpwr_max_cch",      0x02)
        SET_COMP("bcch_arfcn",                  ARFCN_X43)
        SET_COMP("bcch_bsic",                   BSIC_5)
        SET_COMP ("ac_class",                   0x40)
        SET_COMP ("pwr_offset",                 0)
        SET_COMP ("plmn_forb",                  FALSE)
        SET_COMP ("la_forb",                    FALSE)
        SET_COMP ("rr_cell_env",                RR_CELL_ENV_2147_32)
        SET_COMP("limited",                     LIMITED_NO)
        SET_COMP("bs_pa_mfrms",                 BS_PA_MFRMS_2)
ENDSTRUCT
BEGIN_PSTRUCT ("serving_cell_info", SERVING_CELL_TC651)
        SET_COMP ("gprs_rxlev_access_min",      0x16)
        SET_COMP ("gprs_ms_txpwr_max_cch",      0x02)
        SET_COMP("bcch_arfcn",                  ARFCN_X20)
        SET_COMP("bcch_bsic",                   BSIC_5)
        SET_COMP ("ac_class",                   0x40)
        SET_COMP ("pwr_offset",                 0)
```

```
        SET_COMP ("plmn_forb",                  FALSE)
        SET_COMP ("la_forb",                    FALSE)
        SET_COMP ("rr_cell_env",                RR_CELL_ENV_2147_V)
        SET_COMP("limited",                     LIMITED_NO)
        SET_COMP("bs_pa_mfrms",                 BS_PA_MFRMS_2)
ENDSTRUCT
BEGIN_PSTRUCT ("serving_cell_info", SERVING_CELL_TC706)
        SET_COMP ("gprs_rxlev_access_min",      0x16)
        SET_COMP ("gprs_ms_txpwr_max_cch",      0x02)
        SET_COMP("bcch_arfcn",                  ARFCN_124)
        SET_COMP("bcch_bsic",                   BSIC_3)
        SET_COMP ("ac_class",                   0x40)
        SET_COMP ("pwr_offset",                 0)
        SET_COMP ("plmn_forb",                  FALSE)
        SET_COMP ("la_forb",                    FALSE)
        SET_COMP ("rr_cell_env",                RR_CELL_ENV_2149)
        SET_COMP("limited",                     LIMITED_NO)
        SET_COMP("bs_pa_mfrms",                 BS_PA_MFRMS_2)
ENDSTRUCT
BEGIN_PSTRUCT ("mid", MS_ID_IMSI_HPLMN_TMSI_TC702)
        SET_COMP ("len_imsi",                   0x0D)
        SET_COMP ("imsi",                       IMSIELEMENTE)
        SET_COMP ("v_tmsi",                     1)
        SET_COMP ("tmsi",                       TMSI_0X142)
        SET_COMP ("v_ptmsi",                    0)
        SET_COMP ("ptmsi",                      0x00000000)
        SET_COMP ("v_ptmsi2",                   0)
        SET_COMP ("ptmsi2",                     0x00000000)
ENDSTRUCT
BEGIN_PSTRUCT ("mid",                   S_MS_ID_IMSI_HPLMN_TMSI)
        SET_COMP ("len_imsi",           0x0D)
        SET_COMP ("imsi",               A_IMSI_CONTENT)
        SET_COMP ("v_tmsi",             1)
        SET_COMP ("tmsi",               0x142)
        SET_COMP ("v_ptmsi",            0)
        SKIP_COMP ("ptmsi")
        SET_COMP ("v_ptmsi2",           0)
        SET_COMP ("ptmsi2",             0x00000000)
ENDSTRUCT
BEGIN_PSTRUCT ("mid", MS_ID_IMSI_HPLMN_TMSI_TC704)
        SET_COMP ("len_imsi",                   0x0D)
        SET_COMP ("imsi",                       IMSIELEMENTE)
        SET_COMP ("v_tmsi",                     1)
        SET_COMP ("tmsi",                       TMSI_0X142)
        SET_COMP ("v_ptmsi",                    0)
        SET_COMP ("ptmsi",                      0x00000000)
        SET_COMP ("v_ptmsi2",                   0)
        SET_COMP ("ptmsi2",                     0x00000000)
ENDSTRUCT
BEGIN_PSTRUCT ("mid", IMSI_TMSI_EMPTY)
        SET_COMP ("len_imsi",                   0x0)
        SET_COMP ("imsi",                       IMSI_EMPTY)
        SET_COMP ("v_tmsi",                     0)
        SET_COMP ("tmsi",                       0x00000000)
        SET_COMP ("v_ptmsi",                    0)
```

```
        SET_COMP ("ptmsi",                  0x00000000)
        SET_COMP ("v_ptmsi2",               0)
        SET_COMP ("ptmsi2",                 0x00000000)
ENDSTRUCT
BEGIN_PSTRUCT ("mid", IMSI_TMSI_EMPTY_PTMSI_NULL)
        SET_COMP ("len_imsi",               0x0)
        SET_COMP ("imsi",                   IMSI_EMPTY)
        SET_COMP ("v_tmsi",                 0)
        SET_COMP ("tmsi",                   0x00000000)
        SET_COMP ("v_ptmsi",                0x00)
        SET_COMP ("ptmsi",                  0x00000000)
        SET_COMP ("v_ptmsi2",               0)
        SET_COMP ("ptmsi2",                 0x00000000)
ENDSTRUCT
BEGIN_PSTRUCT ("mid", MS_ID_IMSI_HPLMN_TMSI_TC970)
        SET_COMP ("len_imsi",               0x0D)
        SET_COMP ("imsi",                   IMSIELEMENTE)
        SET_COMP ("v_tmsi",                 1)
        SET_COMP ("tmsi",                   TMSI_0X142)
        SET_COMP ("v_ptmsi",                1)
        SET_COMP ("ptmsi",                  DUMMY_LONG)
        SET_COMP ("v_ptmsi2",               0)
        SET_COMP ("ptmsi2",                 0x00000000)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS_TC730)
        SET_COMP ("c",                      C_SEND_NOT_CONT)
        SET_COMP ("no",                     TWO_BURSTS)
        SET_COMP ("delta",                  DELTA_TWO_BURSTS)
        SET_COMP ("rach",                   RACH_2_BURSTS_TC730)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS_TC403)
        SET_COMP ("c",                      C_SEND_NOT_CONT)
        SET_COMP ("no",                     TWO_BURSTS)
        SET_COMP ("delta",                  DELTA_TWO_BURSTS)
        SET_COMP ("rach",                   RACH_2_BURSTS_TC403)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS_TC721)
        SET_COMP ("c",                      C_SEND_NOT_CONT)
        SET_COMP ("no",                     TWO_BURSTS)
        SET_COMP ("delta",                  DELTA_TWO_BURSTS)
        SET_COMP ("rach",                   RACH_2_BURSTS_TC721)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS_TC803)
        SET_COMP ("c",                      C_SEND_NOT_CONT)
        SET_COMP ("no",                     TWO_BURSTS)
        SET_COMP ("delta",                  DELTA_TWO_BURSTS)
        SET_COMP ("rach",                   RACH_2_BURSTS_TC803)
ENDSTRUCT
BEGIN_PSTRUCT ("send_mode", SEND_MODE_2_BURSTS_TC871)
        SET_COMP ("c",                      C_SEND_NOT_CONT)
        SET_COMP ("no",                     TWO_BURSTS)
        SET_COMP ("delta",                  DELTA_TWO_BURSTS)
        SET_COMP ("rach",                   RACH_2_BURSTS_TC871)
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("pck_chan_desc", PACKET_CHANNEL_DESC)
        SET_COMP ("pck_chan_type",          PCK_CH_1)
        SET_COMP ("tn",                     0x07) /* range: 0..7, Chosen arbitrarily */
        SET_COMP ("tsc",                    0x06)                   /* dito */
        SET_COMP ("hop",                    HOP_NO)
        SET_COMP ("indir",                  0x00)                   /*hop=0*/
        SET_COMP ("arfcn",                  ARFCN_30)               /*hop=0 && indir=0*/
        SKIP_COMP ("maio"                   )                       /*hop=0 && indir=1 || hop=1*/
        SKIP_COMP ("ma_num"                 )                       /*hop=0 && indir=1*/
        SKIP_COMP ("flag")
        SKIP_COMP ("ch_mark1")              /*hop=0 && indir=1*/
        /*SKIP_COMP ("hsn",                 HSN_14)*/               /*hop=1*/
        SKIP_COMP("hsn")
ENDSTRUCT

/*
Immediate Assignment Components
*/

BEGIN_MSTRUCT ("time_advance", TIMING_ADVANCE_30)
        SET_COMP ("ta",                     0x1E)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_DL_ASSIGN)
        SET_COMP ("ia_rest_oct_par",        IA_REST_OCT_PARAM_DL_ASSIGN)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_DL_ASSIGN_COPY)
        SET_COMP ("ia_rest_oct_par",        IA_REST_OCT_PARAM_DL_ASSIGN_COPY)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct_par", IA_REST_OCT_PARAM_DL_ASSIGN)
        SET_COMP ("flag",                   0x01)
        SKIP_COMP ("ia_freq_par"            )                       /*flag=0*/
        SET_COMP ("ia_assign_par",          IA_REST_ASSIGN_PARAM_DL)    /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct_par", IA_REST_OCT_PARAM_DL_ASSIGN_COPY)
        SET_COMP ("flag",                   0x01)
        SKIP_COMP ("ia_freq_par"            )                       /*flag=0*/
        SET_COMP ("ia_assign_par",          IA_REST_ASSIGN_PARAM_DL_COPY) /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_assign_par", IA_REST_ASSIGN_PARAM_DL)
        SET_COMP ("flag",                   0x00)
        SET_COMP ("flag1",                  0x01)                   /*flag=0*/
        SKIP_COMP ("pck_upl_ass_ia"         )                       /*flag=0 && flag1=0*/
        SET_COMP ("pck_downl_ass_ia",       PACKET_DL_ASS)          /*flag=0 && flag1#0*/
        SKIP_COMP ("ia_2nd_part"            )                       /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_assign_par", IA_REST_ASSIGN_PARAM_DL_COPY)
        SET_COMP ("flag",                   0x00)
        SET_COMP ("flag1",                  0x01)                   /*flag=0*/
        SKIP_COMP ("pck_upl_ass_ia"         )                       /*flag=0 && flag1=0*/
        SET_COMP ("pck_downl_ass_ia",       PACKET_DL_ASS_SHOW)     /*flag=0 && flag1#0*/
        SKIP_COMP ("ia_2nd_part"            )                       /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("pck_downl_ass_ia", PACKET_DL_ASS)
        SET_COMP ("ded_tlli",               DED_TLLI_BITBUFFER_TC800)
        SET_COMP ("tfi_ass_rlc",            TFI_TC800)
```

```
        SKIP_COMP ("ta_idx"                     )
        SKIP_COMP ("tbf_start_time"             )
        SKIP_COMP ("p0_bts_prmode"              )
ENDSTRUCT
BEGIN_MSTRUCT ("pck_downl_ass_ia", PACKET_DL_ASS_SHOW)
        SET_COMP ("ded_tlli",                   DED_TLLI_BITBUFFER_TC800)
        SKIP_COMP ("tfi_ass_rlc"                )
        SKIP_COMP ("ta_idx"                     )
        SKIP_COMP ("tbf_start_time"             )
        SET_COMP ("p0_bts_prmode",              P0_BTS_PRMODE_01)
ENDSTRUCT

BEGIN_MSTRUCT ("pck_dl_ass", RR_PACKET_DOWNLINK_TC980)
        SET_COMP ("mac_mode",                   MAC_DYN_ALLOC)
        SET_COMP ("rlc_mode",                   RLC_UNACK_MODE)
        SET_COMP ("ts_all",                     0x02)
        SET_COMP ("pck_ta",                     PACKET_TIMING_TC980)
        SET_COMP ("pwr_ctrl",                   POWER_CONTROL_TC980)
        SET_COMP ("tfi",                        0x03)
        SKIP_COMP ("pck_meas_par"               )                       /* no measurement */
ENDSTRUCT

BEGIN_MSTRUCT ("tfi_ass_rlc", TFI_TC800)
        SET_COMP ("tfi",                        0x02)                   /* Any value not used before */
        SET_COMP ("rlc_mode",                   RLC_UNACK_MODE) /* RLC unacknowledged mode */
        SET_COMP ("alpha",                      0x05)                   /* 0.5 */
        SET_COMP ("gamma",                      0x09)  /* For GSM 900, +9 dBm, For DCS 1800, +6 dBm */
        SET_COMP ("poll",                       POLL_NO_ACTION)
        SET_COMP ("ta_valid",                   TA_VALID_YES)
ENDSTRUCT
BEGIN_MSTRUCT ("p0_bts_prmode", P0_BTS_PRMODE_01)
        SET_COMP ("p0",                         0x02)
        SET_COMP ("pwr_ctrl_mode",              0x01)
        SET_COMP ("pr_mode",                    0x01)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_UL_ASSIGN_TMA1)
        SET_COMP ("ia_rest_oct_par",            IA_REST_OCT_PARAM_UL_ASSIGN_TMA1)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct_par", IA_REST_OCT_PARAM_UL_ASSIGN_TMA1)
        SET_COMP ("flag",                       0x01)
        SKIP_COMP ("ia_freq_par"                )                       /*flag=0*/
        SET_COMP ("ia_assign_par",              IA_REST_ASSIGN_PARAM_UL_TMA1)  /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_assign_par", IA_REST_ASSIGN_PARAM_UL_TMA1)
        SET_COMP ("flag",                       0x00)
        SET_COMP ("flag1",                      0x00)                   /*flag=0*/
        SET_COMP ("pck_upl_ass_ia",             PACKET_UL_ASS_TMA1)     /*flag=0 && flag1=0*/
        SKIP_COMP ("pck_downl_ass_ia"           )                       /*flag=0 && flag1#0*/
        SKIP_COMP ("ia_2nd_part"                )                       /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("pck_upl_ass_ia", PACKET_UL_ASS_TMA1)
        SKIP_COMP ("tfi_ass_alloc"              )
        SKIP_COMP ("alpha"                      )
        SET_COMP ("gamma",                      0x0A)
        SKIP_COMP ("ta_idx"                     )
```

```
        SKIP_COMP ("tbf_start_time"                 )
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_UL_ASSIGN_TMA2)
        SET_COMP ("ia_rest_oct_par",                IA_REST_OCT_PARAM_UL_ASSIGN_TMA2)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct_par", IA_REST_OCT_PARAM_UL_ASSIGN_TMA2)
        SET_COMP ("flag",                           0x01)
        SKIP_COMP ("ia_freq_par"                    )                   /*flag=0*/
        SET_COMP ("ia_assign_par",                  IA_REST_ASSIGN_PARAM_UL_TMA2)  /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_assign_par", IA_REST_ASSIGN_PARAM_UL_TMA2)
        SET_COMP ("flag",                           0x01)
        SKIP_COMP ("flag1"                          )                   /*flag=0*/
        SKIP_COMP ("pck_upl_ass_ia"                 )                   /*flag=0 && flag1=0*/
        SKIP_COMP ("pck_downl_ass_ia"               )                   /*flag=0 && flag1#0*/
        SKIP_COMP ("ia_2nd_part"                    )                   /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_assign_par", IA_REST_ASSIGN_PARAM)
        SET_COMP ("flag",                           0x00)
        SKIP_COMP ("flag1"                          )                   /*flag=0*/
        SKIP_COMP ("pck_upl_ass_ia"                 )                   /*flag=0 && flag1=0*/
        SKIP_COMP ("pck_downl_ass_ia"               )                   /*flag=0 && flag1#0*/
        SKIP_COMP ("ia_2nd_part"                    )                   /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_2nd_part", IA_2ND_PART)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_assign_par", IA_REST_ASSIGN_PARAM2)
        SET_COMP ("flag",                           0x01)
        SET_COMP ("flag1",                          0x00)               /*flag=0*/
        SKIP_COMP ("pck_upl_ass_ia"                 )                   /*flag=0 && flag1=0*/
        SKIP_COMP ("pck_downl_ass_ia"               )                   /*flag=0 && flag1#0*/
        SET_COMP ("ia_2nd_part",                    IA_2ND_PART)        /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_freq_par", IA_REST_FREQ_PARAM)
        SET_COMP ("fp_len",                         0x02)
        SET_COMP ("maio",                           0x10)
        SET_COMP ("mac",                            MOB_ALLOC_2)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct_par", IA_REST_PARAM)
        SET_COMP ("flag",                           0x00)
        SET_COMP ("ia_freq_par",                    IA_REST_FREQ_PARAM) /*flag=0*/
        SKIP_COMP ("ia_assign_par")                                     /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct_par", IA_REST_PARAM2)
        SET_COMP ("flag",                           0x00)
        SET_COMP ("ia_freq_par",                    IA_REST_FREQ_PARAM) /*flag=0*/
        SKIP_COMP ("ia_assign_par")                                     /*flag=1*/
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_TC730)
        SET_COMP ("ia_rest_oct_par",                IA_REST_PARAM)
ENDSTRUCT
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_TC733)
        SET_COMP ("ia_rest_oct_par",                IA_REST_PARAM2)
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("ia_rest_oct", IA_REST_OCTETS_EMPTY)
        SKIP_COMP ("ia_rest_oct_par")
ENDSTRUCT
/*
IA until here
*/

BEGIN_PSTRUCT ("p_chan_req_des", REQUESTED_CHANNEL_TC970)
        SET_COMP ("mo_mt",                      0x01)                   /* Flag */
        SET_COMP ("prio",                       CRD_PRIO_4)
        SET_COMP ("rlc_mode_req",               RLC_ACK_MODE)
        SET_COMP ("llc_type",                   LLC_IS_ACK)
        SET_COMP ("req_bwd",                    DUMMY_SHORT)
        SET_COMP ("rlc_octets",                 0xFADE)
ENDSTRUCT
BEGIN_PSTRUCT ("gprs_meas_results", MEASUREMENT_RESULTS_TC970)
        SET_COMP ("c_value",                    0x01)                   /*0..63*/
        SET_COMP ("rxqual",                     0x01)                   /*0..7 */
        SET_COMP ("sign_var",                   0x01)
ENDSTRUCT
BEGIN_MSTRUCT ("chan_coding", CHAN_CODING_TC770)
        SET_COMP ("mac_mode",                   MAC_FIX_ALLOC)
        SET_COMP ("cod_scheme",                 COD_S_3)
ENDSTRUCT
BEGIN_MSTRUCT ("chan_req_desc", CHANNEL_REQ_TC770)
        SET_COMP ("or_ty",                      0x01)
        SET_COMP ("crd_prio",                   CRD_PRIO_4)
        SET_COMP ("rlc_mode",                   RLC_ACK_MODE)
        SET_COMP ("llc_fr_type",                LLC_IS_ACK)
        SET_COMP ("rbw",                        DUMMY_SHORT)
        SET_COMP ("rlc_c_oct",                  0xFADE)
ENDSTRUCT
BEGIN_MSTRUCT ("gprs_meas_res", MEASURES_TC770)
        SET_COMP ("c_val",                      0x01)                   /*0..63*/
        SET_COMP ("rxqual",                     0x01)                   /*0..7 */
        SET_COMP ("sign_var",                   0x01)
ENDSTRUCT

BEGIN_MSTRUCT ("ba_range", BA_RANGE)
        SET_COMP ("num_range",                  0x01)
/*      SKIP_COMP ("freq_range",                FREQUENCY_RANGE)*/
        SKIP_COMP ("freq_range")
ENDSTRUCT
BEGIN_MSTRUCT ("group_chan_desc", G_CHANNEL_DESC)
        SET_COMP ("chan_type",                  TCH_F)
        SET_COMP ("tn",                         TN_3)
        SET_COMP ("tsc",                        TSC_2)
        SET_COMP ("hop",                        HOP_NO)
        SET_COMP ("arfcn",                      ARFCN_X20)              /* hop=0 */
        SKIP_COMP ("maio"                       )                       /* hop=1 */
        SKIP_COMP ("hsn"                        )                       /* hop=1 */
        SKIP_COMP ("mac"                        )                       /* hop=1 */
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("gprs_resum", RESUMPTION_YES)
        SET_COMP ("res_ack",                    RES_ACK_YES)
ENDSTRUCT
BEGIN_MSTRUCT ("gprs_resum", RESUMPTION_NO)
        SET_COMP ("res_ack",                    RES_ACK_NO)
ENDSTRUCT
BEGIN_MSTRUCT ("ba_list_pref", BA_LIST)
        SKIP_COMP ("freq_range")
        SET_COMP ("arfcn",                      ARFCN_LIST_20)
ENDSTRUCT
BEGIN_MSTRUCT ("freq_range", FREQUENCY_RANGE)
        SET_COMP ("freq_lower",                 ARFCN_3)
        SET_COMP ("freq_higher",                ARFCN_800)
ENDSTRUCT

BEGIN_MSTRUCT ("chan_desc", CHANNEL_DESC_AFTER)
        SET_COMP ("chan_type",                  TCH_F)
        SET_COMP ("tn",                         TN_2)
        SET_COMP ("tsc",                        TSC_2)
        SET_COMP ("hop",                        HOP_NO)
        SET_COMP ("arfcn",                      ARFCN_X43)
        SKIP_COMP ("maio"                       )
        SKIP_COMP ("hsn"                        )
ENDSTRUCT
BEGIN_MSTRUCT ("pck_ta", PACKET_TIMING_TC980)
        SET_COMP ("ta",                         0x30)
        SKIP_COMP ("ta_idx_nm"                  )
ENDSTRUCT
BEGIN_MSTRUCT ("pwr_ctrl", POWER_CONTROL_TC980)
        SET_COMP ("alpha",                      0x05)
        SET_COMP ("tagged_gamma",               TAGGED_GAMMA_ARR)
ENDSTRUCT
BEGIN_MSTRUCT ("tagged_gamma", TAGGED_GAMMA_EMPTY)
        SKIP_COMP ("gamma")
ENDSTRUCT
BEGIN_MSTRUCT ("tagged_gamma", TAGGED_GAMMA_TC980)
        SET_COMP ("gamma",                      0x01)
ENDSTRUCT
BEGIN_STRUCT_ARRAY(TAGGED_GAMMA_ARR, 8)
        TAGGED_GAMMA_EMPTY,
        TAGGED_GAMMA_EMPTY,
        TAGGED_GAMMA_EMPTY,
        TAGGED_GAMMA_EMPTY,
        TAGGED_GAMMA_TC980,
        TAGGED_GAMMA_EMPTY,
        TAGGED_GAMMA_EMPTY,
        TAGGED_GAMMA_EMPTY
ENDARRAY
/* PDCH Assignment */

BEGIN_MSTRUCT ("nc_mode", NC_MODE_TC791)
        SET_COMP("ncm", NC_MODE_DEF             )
ENDSTRUCT
```

```
BEGIN_MSTRUCT ("nc_mode", NC_MODE_TC640)
        SET_COMP("ncm", NC_MODE_DEF             )
ENDSTRUCT
BEGIN_PSTRUCT ("non_gprs", NON_GPRS_EST_REQ_TC860)
        SET_COMP ("v_non_gprs",                 0x00)
        SKIP_COMP ("att"                        )
        SET_COMP ("v_T3212",                    0x00)
        SKIP_COMP ("T3212"                      )
        SKIP_COMP ("neci"                       )
        SKIP_COMP ("pwrc"                       )
        SKIP_COMP ("dtx"                        )
        SKIP_COMP ("radio_link_timeout"         )
        SKIP_COMP ("bs_ag_blks_res"             )
        SKIP_COMP ("ccch_conf"                  )
        SKIP_COMP ("bs_pa_mfrms"                )
        SKIP_COMP ("max_retrans"                )
        SKIP_COMP ("tx_integer"                 )
        SKIP_COMP ("ec"                         )
        SKIP_COMP ("gprs_ms_txpwr_max_cch"      )
        SET_COMP ("v_ext_ie",                   0x00)
        SKIP_COMP ("ext_ie"                     )
ENDSTRUCT

BEGIN_PSTRUCT ("non_gprs",                      NON_GPRS_ACT_REQ)
        SET_COMP ("v_non_gprs",                 0x01)
        SET_COMP ("att",                        0x01)
        SET_COMP ("v_T3212",                    0x00)
        SKIP_COMP ("T3212"                      )
        SKIP_COMP ("neci"                       )
        SKIP_COMP ("pwrc"                       )
        SKIP_COMP ("dtx"                        )
        SKIP_COMP ("radio_link_timeout"         )
        SKIP_COMP ("bs_ag_blks_res"             )
        SKIP_COMP ("ccch_conf"                  )
        SKIP_COMP ("bs_pa_mfrms"                )
        SKIP_COMP ("max_retrans"                )
        SKIP_COMP ("tx_integer"                 )
        SKIP_COMP ("ec"                         )
        SKIP_COMP ("gprs_ms_txpwr_max_cch"      )
        SET_COMP ("v_ext_ie",                   0x00)
        SKIP_COMP ("ext_ie"                     )
ENDSTRUCT

BEGIN_PSTRUCT ("cr_par", CR_PAR_01)
        SET_COMP("cell_bar_access_2", CBA_2_NORMAL)
        SET_COMP("exc_acc", EXC_ACC_NORMAL )
        SET_COMP("same_ra_as_serving_cell", SAME_RA_AS_SCELL)
        SET_COMP("gprs_rxlev_access_min", 0x16)
        SET_COMP("gprs_ms_txpwr_max_cch", 2)
        SET_COMP("gprs_temporary_offset", 0)
        SET_COMP("gprs_penalty_time", 0)
        SET_COMP("gprs_reselect_offset", 23)
        SET_COMP("priority_class", 0)
        SET_COMP("hcs_thr", 0)
        SET_COMP("si13_location", 0)
```

```
        SET_COMP("pbcch_location", 2)
        SET_COMP("psi1_repeat_period", 2)
ENDSTRUCT
BEGIN_PSTRUCT ("cr_par", CR_PAR_02)
        SET_COMP("cell_bar_access_2", CBA_2_NORMAL)
        SET_COMP("exc_acc", EXC_ACC_NORMAL )
        SET_COMP("same_ra_as_serving_cell", SAME_RA_AS_SCELL)
        SET_COMP("gprs_rxlev_access_min", 0x16)
        SET_COMP("gprs_ms_txpwr_max_cch", 2)
        SET_COMP("gprs_temporary_offset", 0)
        SET_COMP("gprs_penalty_time", 0)
        SET_COMP("gprs_reselect_offset", 23)
        SET_COMP("priority_class", 2)
        SET_COMP("hcs_thr", 0)
        SET_COMP("si13_location", 0)
        SET_COMP("pbcch_location", 2)
        SET_COMP("psi1_repeat_period", 2)
ENDSTRUCT
BEGIN_PSTRUCT ("cr_par", CR_PAR_03)
        SET_COMP("cell_bar_access_2", CBA_2_NORMAL)
        SET_COMP("exc_acc", EXC_ACC_NORMAL )
        SET_COMP("same_ra_as_serving_cell", SAME_RA_AS_SCELL)
        SET_COMP("gprs_rxlev_access_min", 0x16)
        SET_COMP("gprs_ms_txpwr_max_cch", 2)
        SET_COMP("gprs_temporary_offset", 0)
        SET_COMP("gprs_penalty_time", 0)
        SET_COMP("gprs_reselect_offset", 23)
        SET_COMP("priority_class", 2)
        SET_COMP("hcs_thr", 0)
        SET_COMP("si13_location", 0)
        SET_COMP("pbcch_location", 2)
        SET_COMP("psi1_repeat_period", 2)
ENDSTRUCT
BEGIN_PSTRUCT ("cr_par", CR_PAR_04)
        SET_COMP("cell_bar_access_2", CBA_2_NORMAL)
        SET_COMP("exc_acc", EXC_ACC_NORMAL )
        SET_COMP("same_ra_as_serving_cell", SAME_RA_AS_SCELL)
        SET_COMP("gprs_rxlev_access_min", 0x16)
        SET_COMP("gprs_ms_txpwr_max_cch", 2)
        SET_COMP("gprs_temporary_offset", 0)
        SET_COMP("gprs_penalty_time", 0)
        SET_COMP("gprs_reselect_offset", 23)
        SET_COMP("priority_class", 2)
        SET_COMP("hcs_thr", 0)
        SET_COMP("si13_location", 0)
        SET_COMP("pbcch_location", 2)
        SET_COMP("psi1_repeat_period", 2)
ENDSTRUCT
BEGIN_PSTRUCT ("cr_par", CR_PAR_05)
        SET_COMP("cell_bar_access_2", CBA_2_NORMAL)
        SET_COMP("exc_acc", EXC_ACC_NORMAL )
        SET_COMP("same_ra_as_serving_cell", SAME_RA_AS_SCELL)
        SET_COMP("gprs_rxlev_access_min", 0x16)
        SET_COMP("gprs_ms_txpwr_max_cch", 2)
        SET_COMP("gprs_temporary_offset", 0)
```

```
        SET_COMP("gprs_penalty_time", 0)
        SET_COMP("gprs_reselect_offset", 23)
        SET_COMP("priority_class", 2)
        SET_COMP("hcs_thr", 28)
        SET_COMP("si13_location", 0)
        SET_COMP("pbcch_location", 2)
        SET_COMP("psi1_repeat_period", 2)
ENDSTRUCT
BEGIN_MSTRUCT("mob_id", MOB_IDENT_IMSI)
        SET_COMP ("ident_type",          ID_TYPE_IMSI)
        SET_COMP ("odd_even",            ODD)
        SET_COMP ("ident_dig",           MIMSI_1233347114912)
        SKIP_COMP ("tmsi"                )
        SKIP_COMP ("dmy"                 )
ENDSTRUCT

/*
```

2.6 Bitbuffer

```
*/

SET_BITBUF ("ded_tlli", DED_TLLI_BITBUFFER_TC770, 32)
        0x12, 0x34, 0x56, 0x78
ENDBITBUF

SET_OBITBUF ("ded_tlli", DED_TLLI_BITBUFFER_TC800, 32)
        0x12, 0x34, 0x56, 0x78
ENDBITBUF
```

3 TEST CASES

3.1 Setup

3.1.1 RRG000: Setup the routing and PCO view for the RR test

Description:

Routings for the RR tests are set. Parameter Setup. Dual Band Extended (std=6) is set for all Tests due to GPRS

Preamble:

none

```
     MM/GRR                              RR                                  PL/DL
       |                                  |                                    |
COMMAND (TAP RESET)
COMMAND (PL RESET)
COMMAND (RR RESET)
COMMAND (GRR RESET)
COMMAND (MM RESET)
COMMAND (DL RESET)
       |                                  |                                    |
COMMAND (TAP REDIRECT CLEAR)
COMMAND (RR REDIRECT CLEAR)
COMMAND (PL REDIRECT CLEAR)
COMMAND (MM REDIRECT CLEAR)
COMMAND (GRR REDIRECT CLEAR)
       |                                  |                                    |
COMMAND (DL REDIRECT RR NULL)
COMMAND (PL REDIRECT RR NULL)
COMMAND (GRR REDIRECT RR NULL)
COMMAND (MM REDIRECT RR NULL)
       |                                   |                                   |
COMMAND (RR REDIRECT PL TAP)
COMMAND (RR REDIRECT DL TAP)
COMMAND (RR REDIRECT MM TAP)
COMMAND (RR REDIRECT GRR TAP)
       |                                  |                                    |
COMMAND (TAP REDIRECT TAP RR)
COMMAND (RR CONFIG NO_SYS_TIME)
COMMAND (PL CONFIG STD=6)
       |                                  |                                    |
```

Parametrization

Primitive	Parameter	Value

History:

01-Sep-00	MSE	RRG0700 adapted for new GPRS stack

3.2 Cell Selection (100)

3.2.1 RRG101: Activation And Cell Selection

Description:

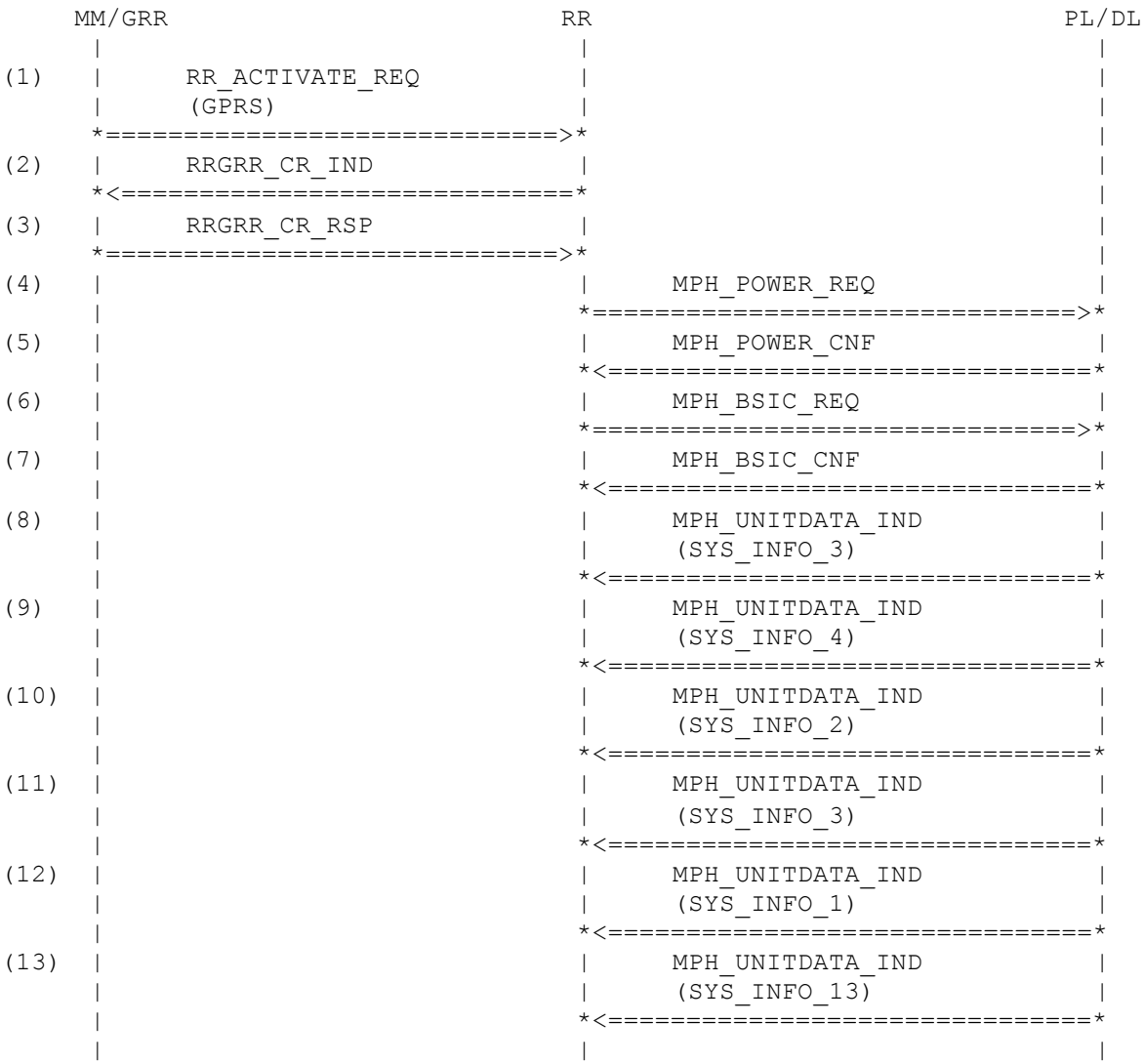
MM activates RR and indicates if it wants GPRS or not. RR starts cell selection. During reading the SysInfos RR gets information if the serving cell supports GPRS.

Preamble:

RRG000

Variants:

<A>..<B>



Parametrization

Primitive	Parameter	Value

(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	BCCH_INFO_TC701
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(3) RRGRR_CR_RSP		
(4) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A
(5) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(8) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(9) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR

	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(10) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_2
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(11) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(12) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	

(13) MPH_UNITDATA_IND			
		arfcn	ARFCN_X43
		fn	fn_780
		sdu	
		{	
		component	RR
		direction	DOWNLINK
		pd	D_SYS_INFO_13
		ti	TI_0
<A>		si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
<B>		si13_rest_oct	SI13_REST_GPRS_TC701
		}	

History:

21-July-00	MSE	Initial
06-December-00	MSE	+cell_type; +timing_info
25-June-01	MSE	undo last change

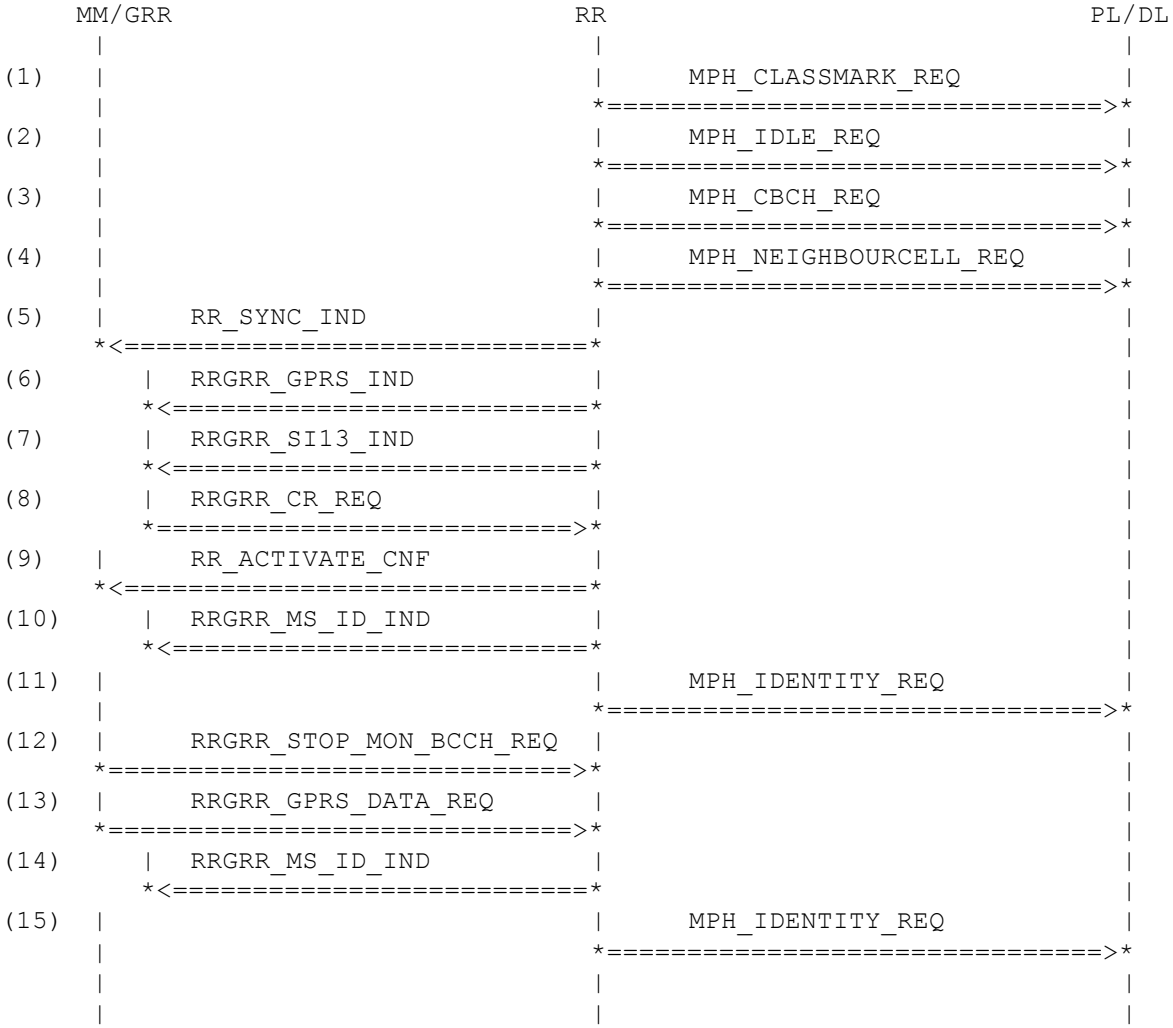
3.2.2 RRG102: GPRS Indication - Going to IDLE-Mode

Description:

RR has all necessary information to go to IDLE-Mode. After reaching IDLE, RR informs MM and GRR. RR sends RRGRR_GPRS_IND to GRR. This indication is used by RR to inform the GRR if the GPRS service is supported in the serving cell. RR also forwards the received SI13 Message to GRR.

Preamble:

RRG101A



Parametrization

Primitive	Parameter	Value
(1) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(2) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5

	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(3) MPH_CBCH_REQ		
	cbch	CBCH_INACTIVE
(4) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	MPH_NCELL_2
	sync_only	NORMAL_BA
(5) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(6) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702
(7) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(1) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X43
	bsic	BSIC_5
(8) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3748
	plmn	PLMN_ID_123
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES

(9) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(10) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(11) RRGRR_STOP_MON_BCCH_REQ		
(12) RRGRR_GPRS_DATA_REQ		
	old_ptmsi	DUMMY_LONG
	new_ptmsi	NOT_PRESENT_32BIT
	tlli	DUMMY_LONG
	p_chan_req_des	REQUESTED_CHANNEL_TC970
	gprs_meas_results	MEASUREMENT_RESULTS_TC970
	mac_req	MAC_FIX_ALLOC
	cs_req	COD_S_3
	rai	RAI_102
(13) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(14) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC970

History:

22-Sep-00	MSE	Initial
08-Jan-01	MSE	_cell_info_ind integrated in _si13_ind
12-Feb-01	MSE	+gprs_indic in RR_ACTIVATE_CNF
16-Jul-01	MSE	IDLE_REQ: +reorg_only

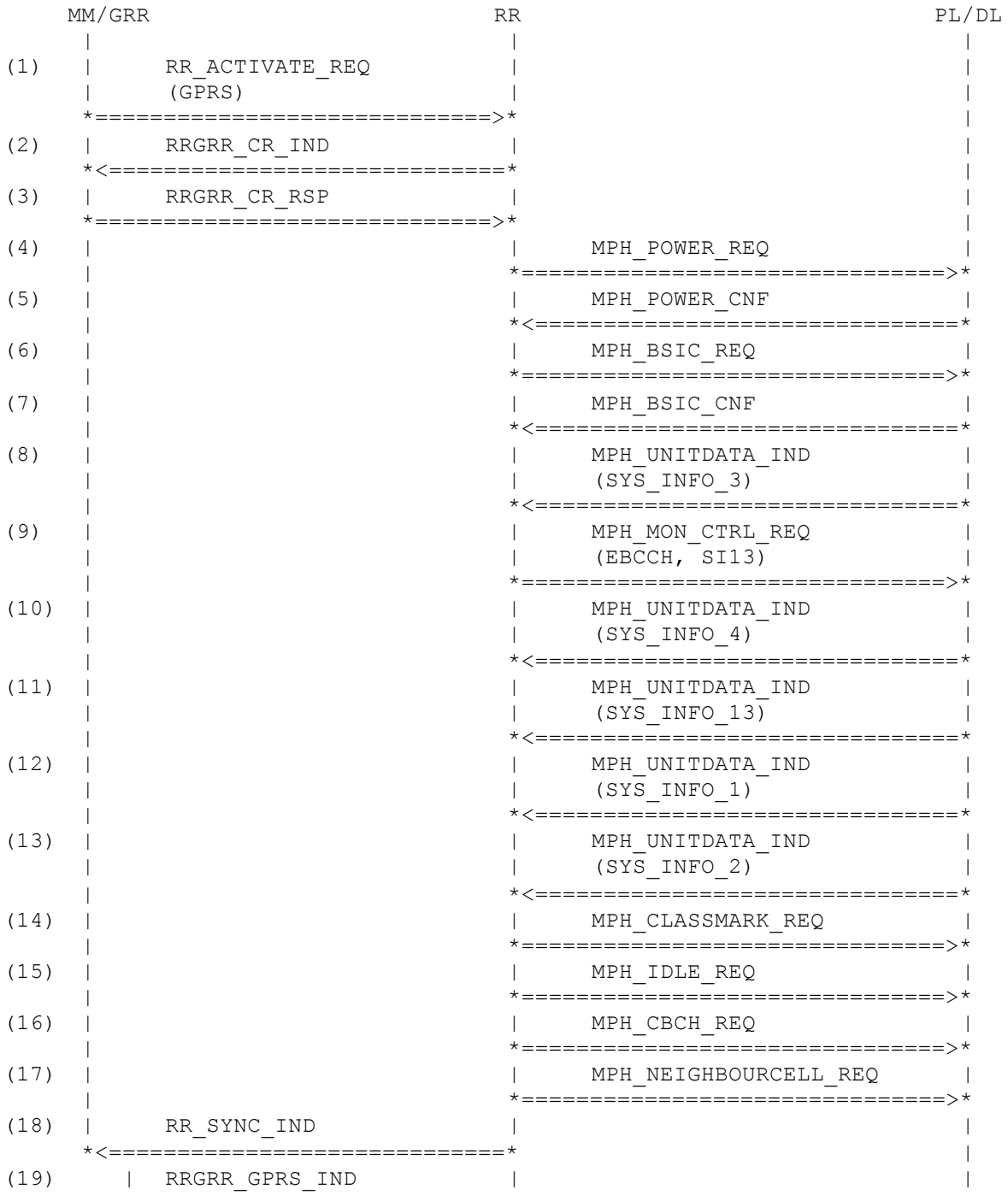
3.2.3 RRG103: Cell Selection – SI13 on EBCCH

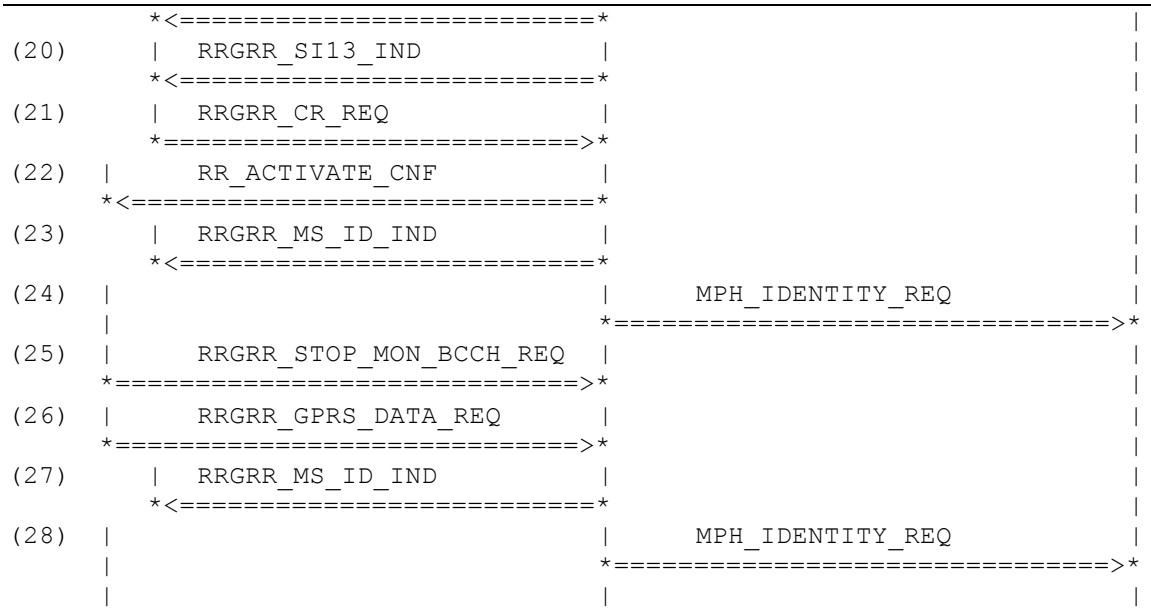
Description:

TC is equal to 701+702 except that SI13 is sent on EBCCH: After Activation RR is informed that SI13 is sent on EBCCH and requests monitoring of this channel.

Preamble:

RRG000





Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	BCCH_INFO_TC701
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(3) RRGRR_CR_RSP		
(4) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A
(5) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR

(8) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC703
	}	
(9) MPH_MON_CTRL_REQ		
	action	START_MON_EBCCH
	si_to_read	UPDATE_SI13
(10) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC703
	}	
(11) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(12) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR

	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(13) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_2
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(14) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(15) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(16) MPH_CBCH_REQ		
	cbch	CBCH_INACTIVE
(17) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	MPH_NCELL_2
	sync_only	NORMAL_BA
(18) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED

(19) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702
(20) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(21) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X43
	bsic	BSIC_5
(22) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3748
	plmn	PLMN_ID_123
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES
(23) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(24) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(25) RRGRR_STOP_MON_BCCH_REQ		
(26) RRGRR_GPRS_DATA_REQ		
	old_ptmsi	DUMMY_LONG
	new_ptmsi	NOT_PRESENT_32BIT
	tlli	DUMMY_LONG
	p_chan_req_des	REQUESTED_CHANNEL_TC970
	gprs_meas_results	MEASUREMENT_RESULTS_TC970
	mac_req	MAC_FIX_ALLOC
	cs_req	COD_S_3
	rai	RAI_102
(27) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(28) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC970

Confidential

History:

21-July-00	MSE	Initial
06-Dec-00	MSE	+cell_type; +timing_info
12-Feb-01	MSE	mon bcch after si4 removed
16-Jul-01	MSE	obsolete params of MPH_BSIC_REQ removed

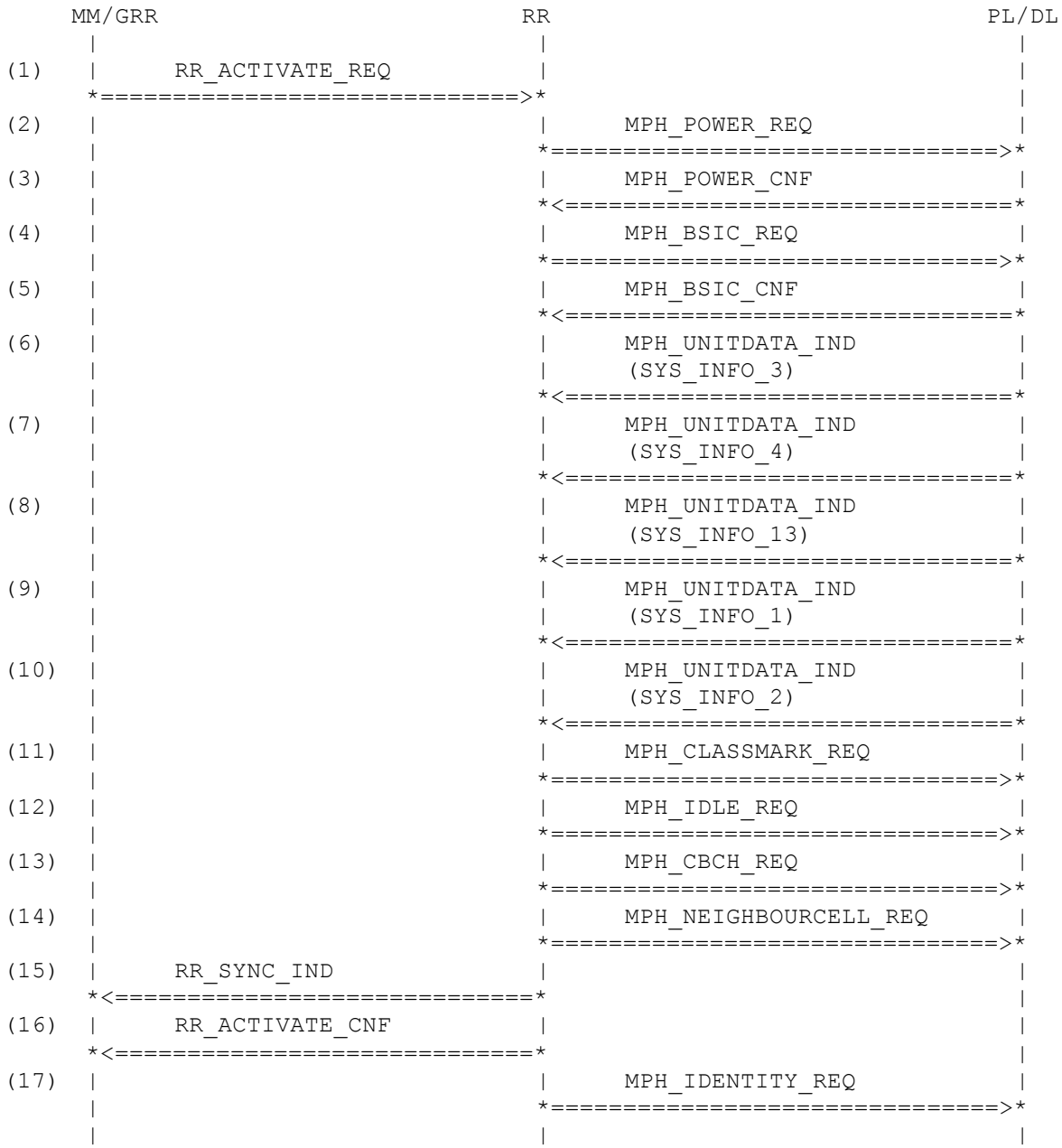
3.2.4 RRG104: Cell Selection – GPRS possible but not wanted

Description:

RR is activated and starts cell selection. RR sends RRGRR_GPRS_IND to GRR. This indication is used by RR to inform the GRR if the GPRS service is supported in the serving cell.
Although the Serving Cell supports GPRS, RR sends a NO-GPRS primitive to GRR because MM did not request GPRS Service.

Preamble:

RRG000



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	BCCH_INFO_TC701
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_NO
(2) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A
(3) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(7) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK

	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(8) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(9) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(10) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_2
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(11) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(12) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5

	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_NOT_USED
(13) MPH_CBCH_REQ		
	cbch	CBCH_INACTIVE
(14) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	MPH_NCELL_2
	sync_only	NORMAL_BA
(15) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(16) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3748
	plmn	PLMN_ID_123
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_NO
(17) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC704

History:

11-Oct-00	MSE	Initial
06-Dec-00	MSE	mph_bsic_req: +cell_type; +timing_info
16-Jul-01	MSE	undo last change
16-Jul-01	MSE	IDLE_REQ: +reorg_only
16-Jul-01	MSE	MPH_NEIGHBOURCELL_REQ: +sync_only

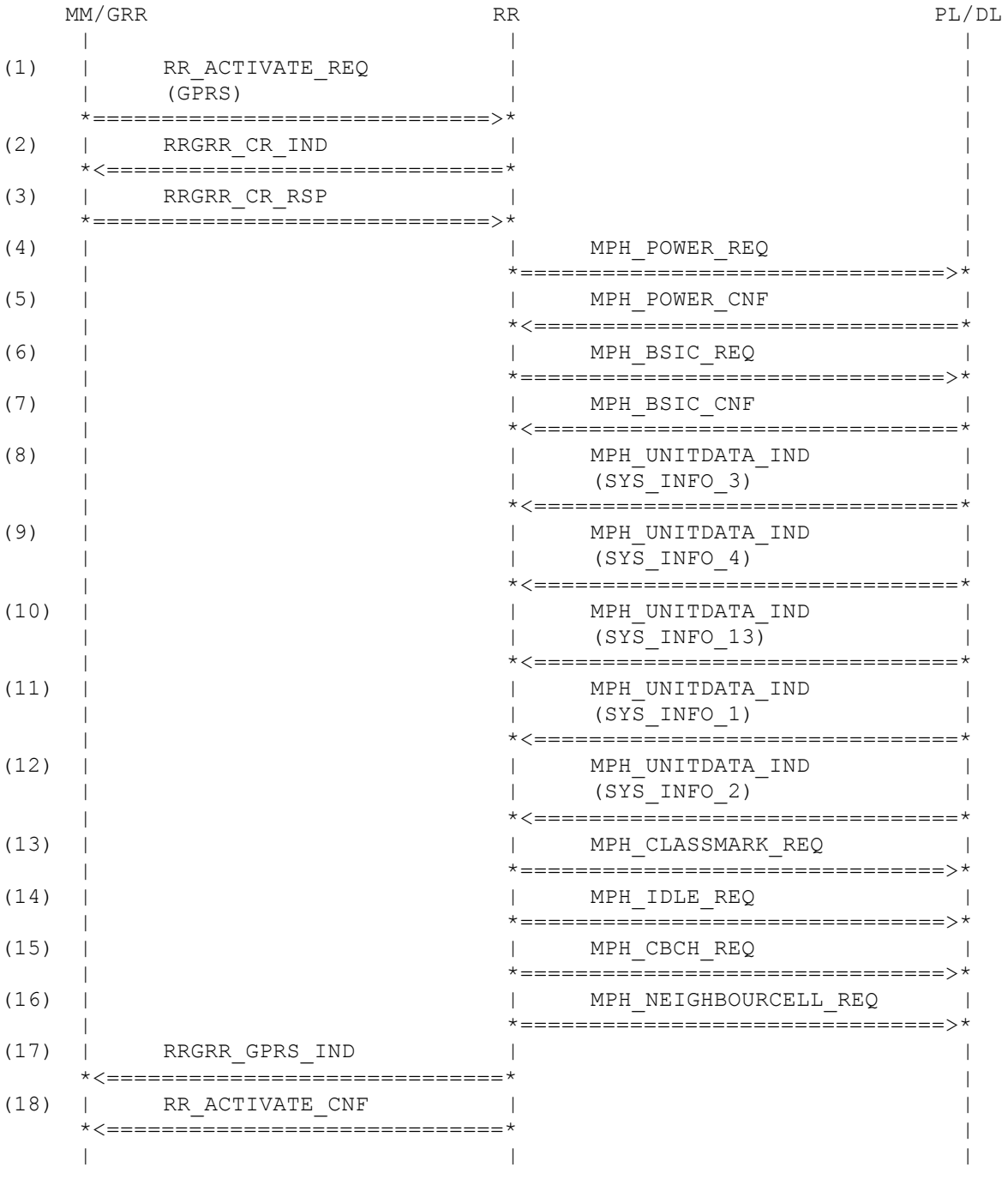
3.2.5 RRG105: Cell Selection – limited service

Description:

RR is activated for Limited Service and with GPRS acitivated (which should actually not happen, but this is easier to test then doing a full service scan first and then selecting a limited service cell) and starts cell selection. RR sends a RRGRR_GPRS_IND to GRR.

Preamble:

RRG000



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_EMPTY
	op	OP_MODE_EMPTY
	cksn	CKSN_NOT_PRES
	kcv	KCV_EMPTY
	accc	ACC_CTRL_CLASS_0000
	imsi_struct	MOBILE_ID_NOT_SET
	tmsi_struct	MOBILE_ID_NOT_SET
	thplmn	TIME_HPLMN_EMPTY
	bcch_info	BCCH_INFO_TC701
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(3) RRGRR_CR_RSP		
(4) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A
(5) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(8) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(9) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43

	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(10) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(11) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(12) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_2
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(13) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND

(14) MPH_IDLE_REQ

mod	MODE_CELL_SELECTION
arfcn	ARFCN_X43
ext_bcch	BSIC_5
comb_ccch	CCD_CCCH_1_NOT_COMB
tn	TN_0
dlt	DLT_LIMITED
pg	PG_0
bs_ag_blocks_res	BS_AG_BLKS_RES_5
bs_pa_mfrms	BS_PA_MFRMS_7
power	MS_TXPWR_MAX_CCH_02
ncc_permitted	NCC_PERMITTED_FF
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	MPH_GPRS_PROCS_NOT_USED

(15) MPH_CBCH_REQ

cbch	CBCH_INACTIVE

(16) MPH_NEIGHBOURCELL_REQ

multi_band	MULTI_BAND_0
arfcn	NOT_USED
sync_only	NORMAL_BA

(17) RRGRR_GPRS_IND

cause	GPRS_SUPPORTED
serving_cell_info	SERVING_CELL_TC105

(18) RR_ACTIVATE_CNF

op	OP_MODE_EMPTY
mm_info	MM_INFO_2
cid	CELL_IDENT_3748
plmn	PLMN_ID_123
lac	LAC_2147
power	RF_CLASS_4
gprs_indication	GPRS_YES

History:

16-Oct-00	MSE	Initial
06-Dec-00	MSE	+cell_type; +timing_info
08-Jan-01	MSE	_cell_info_ind integrated in _si13_ind
16-Jul-01	MSE	obsolete params in MPH_BSIC_REQ removed
16-Jul-01	MSE	IDLE_REQ: +reorg_only
16-Jul-01	MSE	MPH_NEIGHBOURCELL_REQ: +sync_only
22-Mar-02	LG	changed serving_cell_info handle

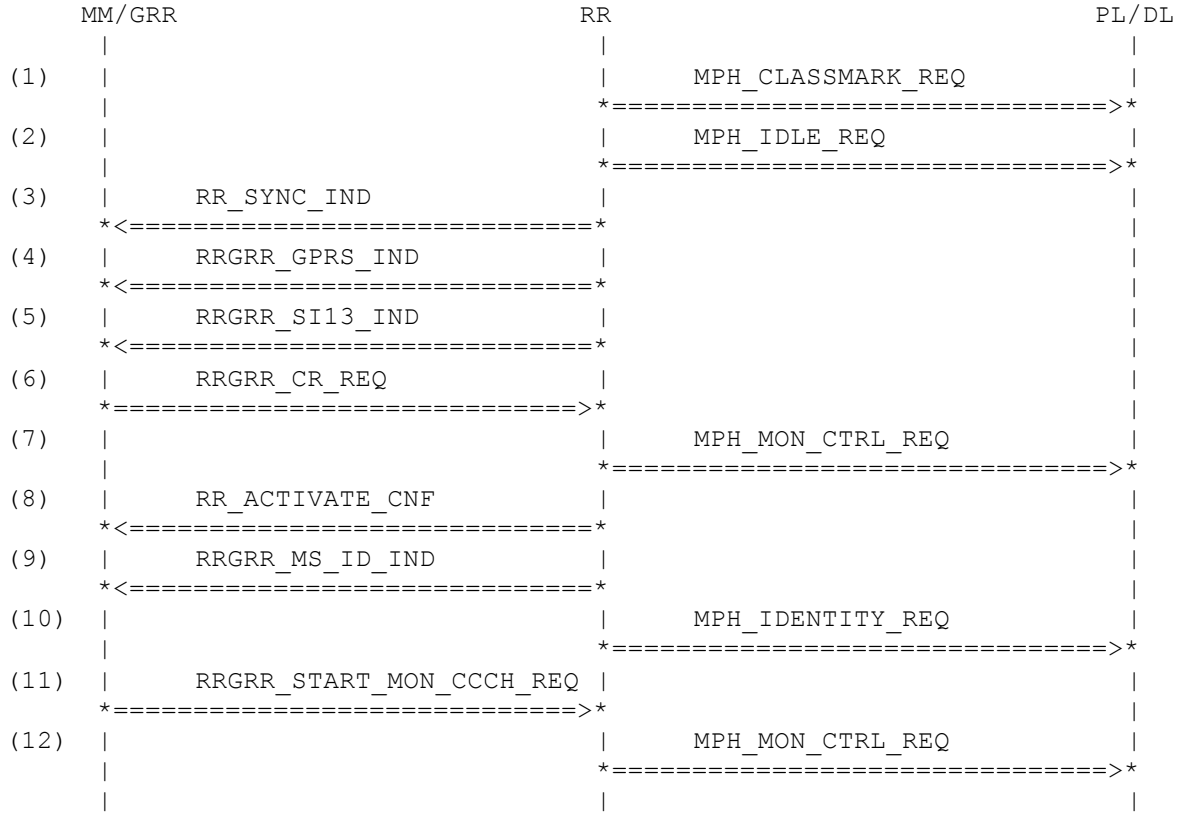
3.2.6 RRG106: GPRS Indication - Going to IDLE-Mode PBCCH

Description:

RR has all necessary information to go to IDLE-Mode. After reaching IDLE, RR informs MM and GRR. RR sends RRGRR_GPRS_IND to GRR. This indication is used by RR to inform the GRR if the GPRS service is supported in the serving cell. RR also forwards the received SI13 Message to GRR.

Preamble:

RRG101B



Parametrization

Primitive	Parameter	Value
(1) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(2) MPH_IDLE_REQ		
	mod	MODE_CONFIG_PL
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02

	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(3) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(4) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702
(5) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(6) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X43
	bsic	BSIC_5
(7) MPH_MON_CTRL_REQ		
	action	ENTER_PIM_PBCCH
	si_to_read	NOT_USED
(8) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3748
	plmn	PLMN_ID_123
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES
(9) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(10) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(11) RRGRR_START_MON_CCCH_REQ		
	pag_mode	NOT_USED
	split_pg	NOT_USED
(12) MPH_MON_CTRL_REQ		
	action	START_MON_CCCH
	si_to_read	NOT_USED

History:

22-Sep-00	MSE	Initial
08-Jan-01	MSE	_cell_info_ind integrated in _si13_ind
12-Feb-01	MSE	+gprs_indic in RR_ACTIVATE_CNF
16-Jul-01	MSE	IDLE_REQ: +reorg_only

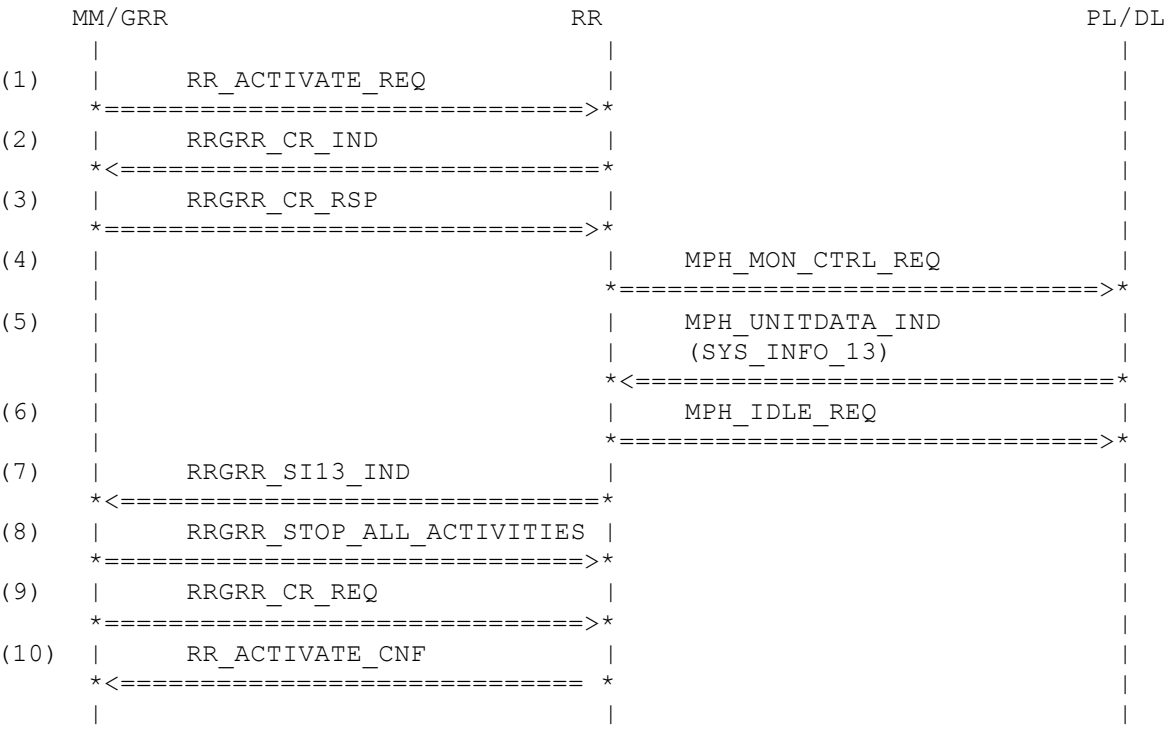
3.2.7 RRG903: GPRS Activation

Description:

RR is GSM only activated. GPRS Attach is initiated, and RR should read SI13 and indicate to GRR.

Preamble:

RRG104



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	BCCH_INFO_TC701
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES

(2) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(3) RRGRR_CR_RSP		
(4) MPH_MON_CTRL_REQ		
	action	START_MON_NBCCH
	si_to_read	6
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(6) MPH_IDLE_REQ		
	mod	MODE_CONFIG_PL
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	NOT_USED
	gprs_support	MPH_GPRS_PROCS_USED
(7) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(8) RRGRR_STOP_ALL_ACTIVITIES		
(9) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X43
	bsic	BSIC_5
(10) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM

mm_info	MM_INFO_2
cid	CELL_IDENT_3748
plmn	PLMN_ID_123
lac	LAC_2147
power	RF_CLASS_4
gprs_indication	GPRS_YES

History:

02-Apr-03	RB	Initial

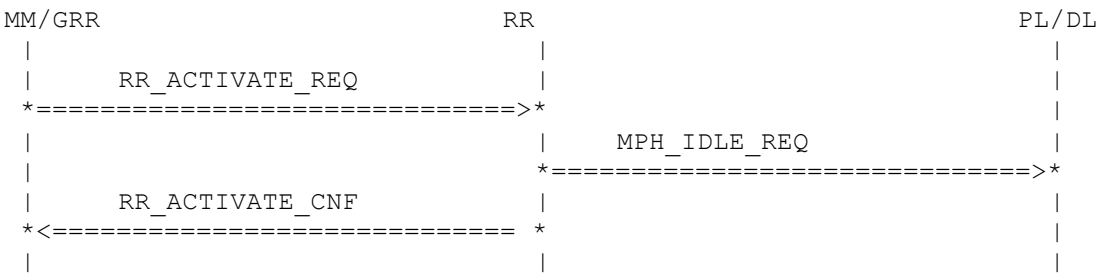
3.2.8 RRG906: GPRS Detach

Description:

MS camped on a cell with GPRS on CCCH. MS is GPRS attached, and GPRS Detach is initiated.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	BCCH_INFO_TC701
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_NO
(2) MPH_IDLE_REQ		
	mod	MODE_SYS_INFO_CHANGE
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1

reorg_only	NORMAL_PGM
eotd_avail	NOT_USED
gprs_support	GPRS_NO

(3) RR_ACTIVATE_CNF

op	OP_MODE_TEST_SIM
mm_info	MM_INFO_2
cid	CELL_IDENT_3748
plmn	PLMN_ID_123
lac	LAC_2147
power	RF_CLASS_4
gprs_indication	MPH_GPRS_PROCS_NOT_USED

(1) History:

02-Apr-03	RB	Initial

3.3 Packet Idle Mode Procedures (200)

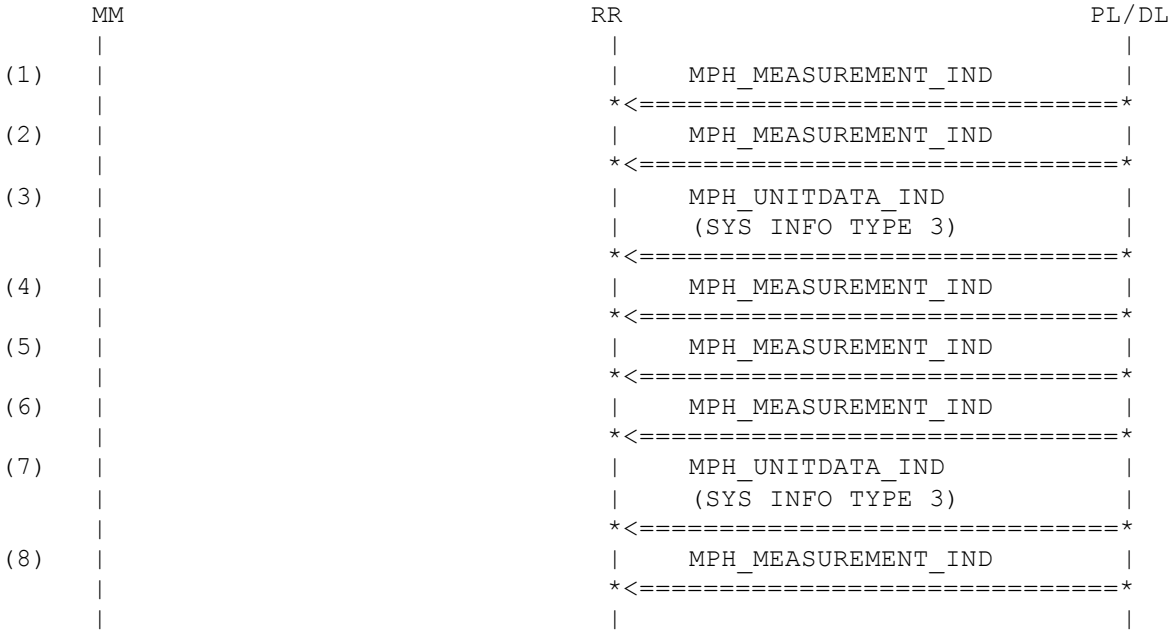
3.3.1 RRG201: Listening to Neighbour Cell

Description:

Layer 1 sends a measurement report with a neighbour cell. RR starts listening to the neighbour cell. First the channel 3 is read. After reading of system information type 3 message on report is waited and then channel 18 is requested.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1

	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(3) MPH_UNITDATA_IND		
	arfcn	ARFCN_3
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(4) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(5) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(6) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1

rx_qual_sub	RX_QUAL_1
dtx	DTX_NOT_USED
otd	TIME_ADV_27
valid	TRUE
fn_offset	FN_OFFSET_1_SEC
ncells	NCELLS_3
gprs_sync	NORMAL_MEAS_REP

(7) MPH_UNITDATA_IND

arfcn	ARFCN_X18
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_SYS_INFO_3
ti	TI_0
cell_ident	CELL_IDENT_3748
loc_area_ident	LOC_AREA_IDENT_123_2147
ctrl_chan_desc	CTRL_CHAN_DESC_1
cell_opt_bcch	CELL_OPT_BCCH_1
cell_select	CELL_SELECT_1
rach_ctrl	RACH_CTRL_1
si3_rest_oct	SI3_REST_GPRS_TC701
}	

(8) MPH_MEASUREMENT_IND

arfcn	ARFCN_X43
rx_lev_full	RX_LEV_20
rx_lev_sub	RX_LEV_20
rx_qual_full	RX_QUAL_1
rx_qual_sub	RX_QUAL_1
dtx	DTX_NOT_USED
otd	TIME_ADV_27
valid	TRUE
fn_offset	FN_OFFSET_1_SEC
ncells	NCELLS_3
gprs_sync	NORMAL_MEAS_REP

History:

04-July-97	DL	Initial
04-Dec-2000	MSE	132 adapted
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

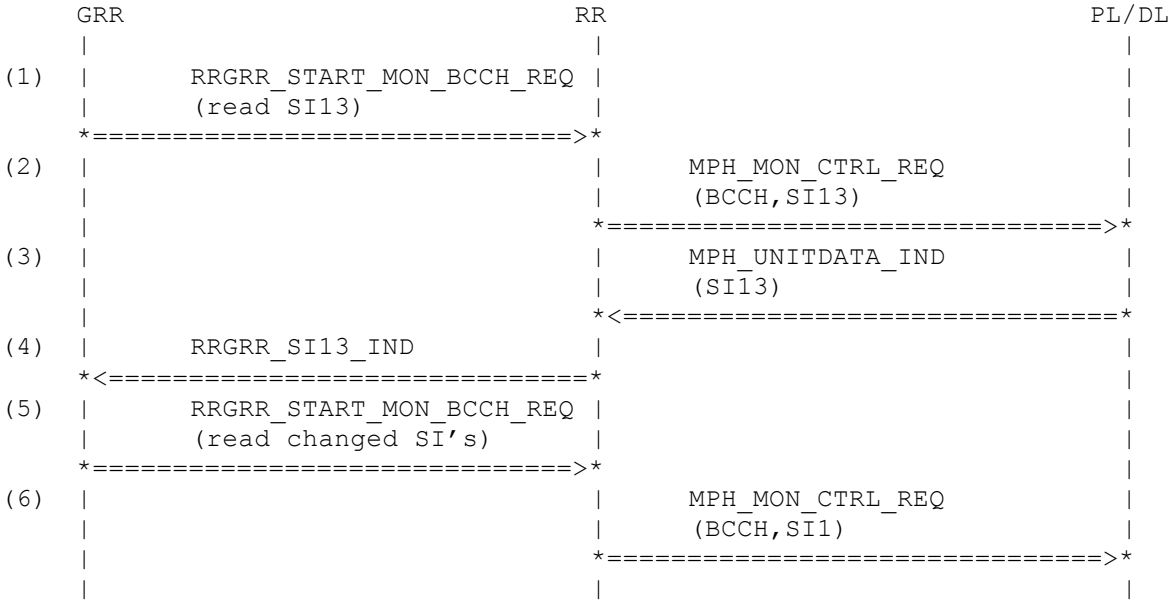
3.3.2 RRG202: Updating System Information

Description:

The MS has to read the system information's every 30 seconds. If GPRS is supported (and SI13 is sent) only the SI13 has to be read every 30 seconds. The SI13 will be forwarded to GRR and according the CHANGE_MARK GRR will request reading of the other SI's.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_START_MON_BCCH_REQ		
	si_to_read	UPDATE_SI13
(2) MPH_MON_CTRL_REQ		
	action	START_MON_NBCCH
	si_to_read	UPDATE_SI13
(3) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(4) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702

	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(5) RRGRR_START_MON_BCCH_REQ		
	si_to_read	UPDATE_SI1
(6) MPH_MON_CTRL_REQ		
	action	START_MON_NBCCH
	si_to_read	UPDATE_SI1

History:

21-July-00	MSE	Initial

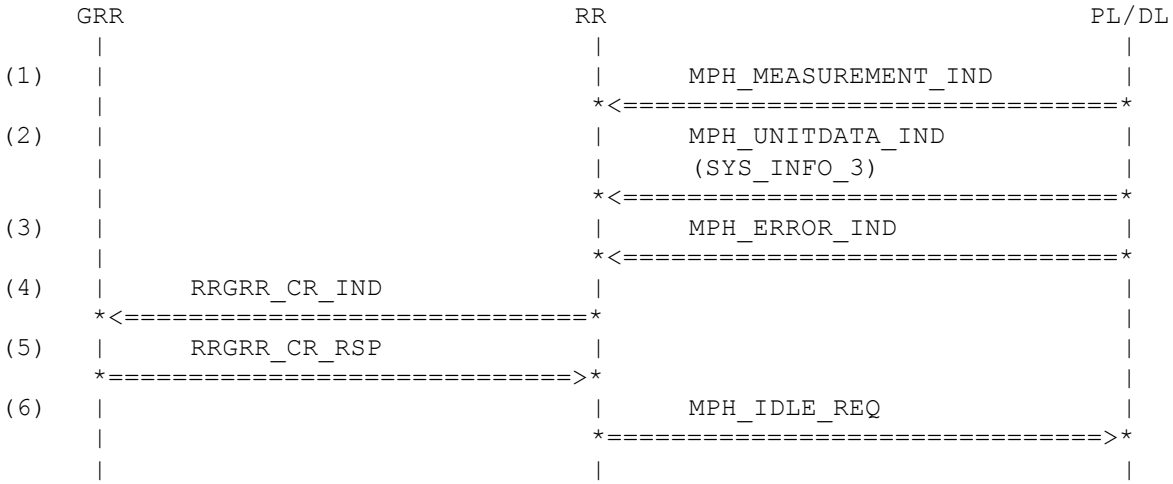
3.3.3 RRG203: Reselection no GPRS

Description:

Because of low RX level measurements a cell reselection is necessary. But the NC does not support GPRS. unfinished

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1

	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(3) MPH_ERROR_IND		
	cs	CS_DOWN_LINK_FAIL
	arfcn	ARFCN_X43
(4) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(5) RRGRR_CR_RSP		
(6) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X20
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NOT_PRESENT_8BIT
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT

History:

05-Mar-01	MSE	Initial
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

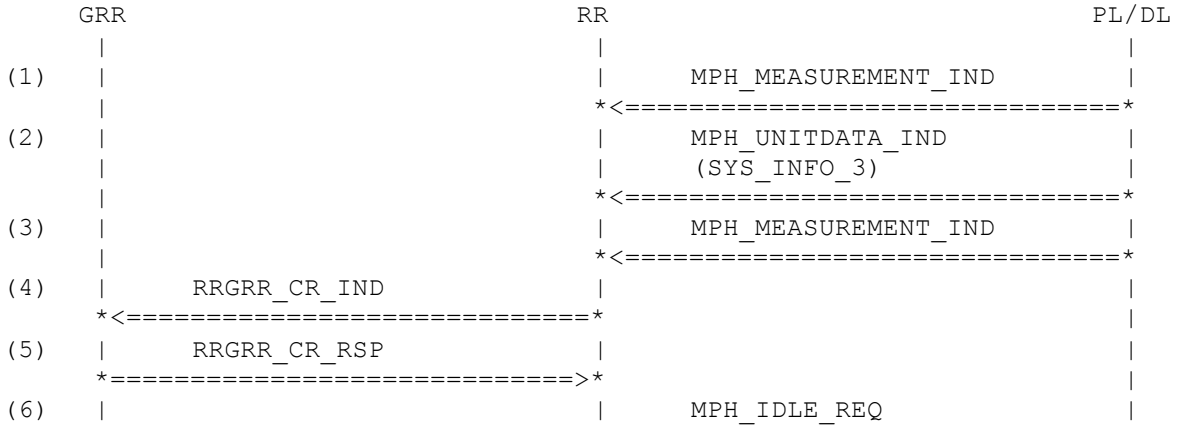
3.3.4 RRG204: Cell reselection (start)

Description:

Because of much better rxlev values of a neighbour cell a cell reselection is started..

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(3) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(4) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(5) RRGRR_CR_RSP		
(6) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION

arfcn	ARFCN_X20
ext_bcch	BSIC_5
comb_ccch	CCD_CCCH_1_NOT_COMB
tn	TN_0
dlt	DLT_23
pg	PG_0
bs_ag_blocks_res	BS_AG_BLKS_RES_5
bs_pa_mfrms	BS_PA_MFRMS_2
power	MS_TXPWR_MAX_CCH_02
ncc_permitted	NOT_PRESENT_8BIT
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	NOT_PRESENT_8BIT

History:

19-Mar-01	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

3.3.5 RRG205: Cell reselection (completion) - the new cell supports GPRS

Description:

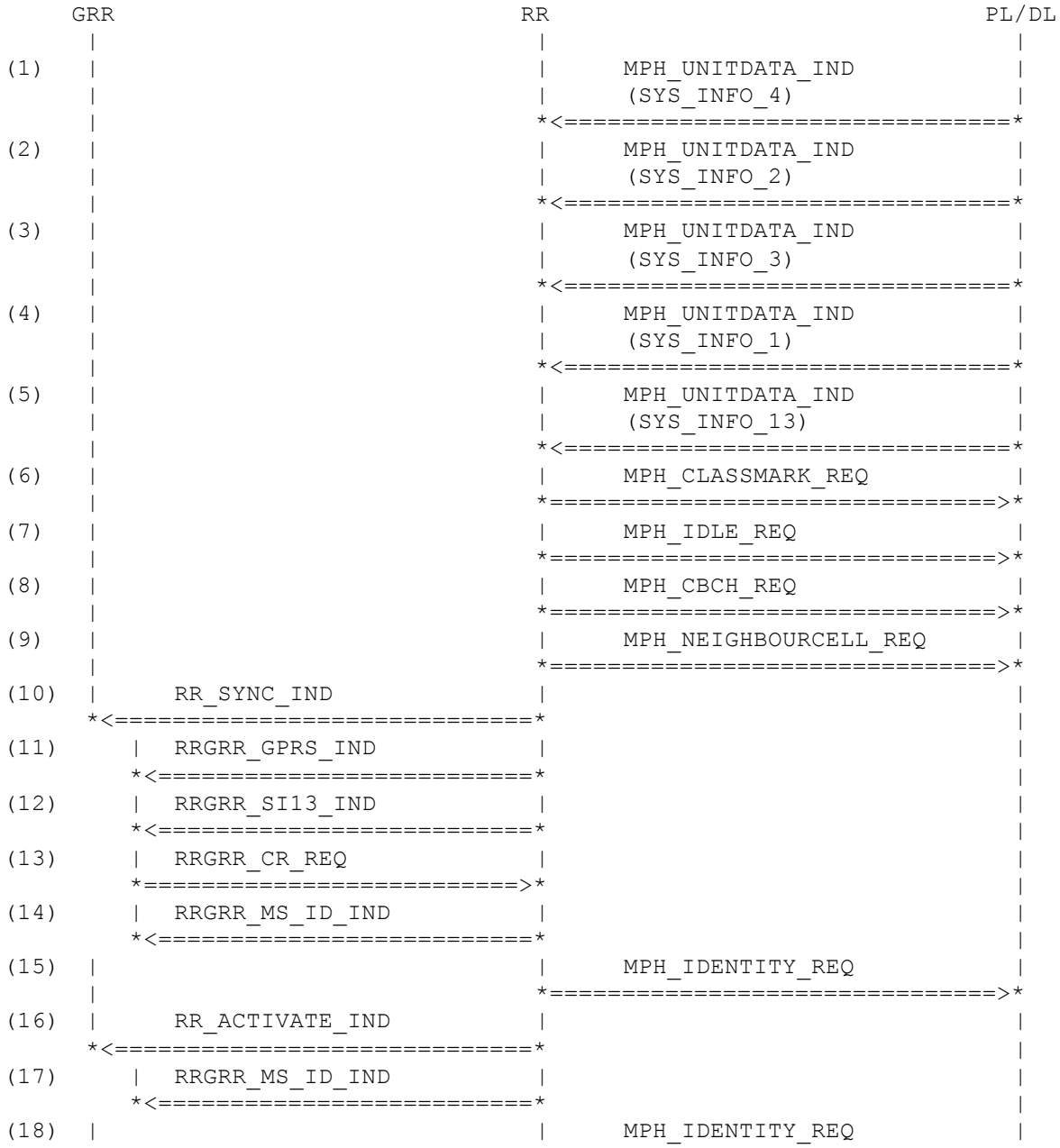
The cell reselection is completed. The new cell supports GPRS.

Variants:

<A>..<B>

Preamble:

<A>RRG204
<B>RRG230



```
        |                                  *================================>*
        |                                  |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_3
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(3) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	

(4) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_2
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(6) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(7) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X20
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(8) MPH_CBCH_REQ		
	cbch	NOT_USED
(9) MPH_NEIGHBOURCELL_REQ		
	multi_band	NOT_USED
	arfcn	A_MPH_NCELL_3
	sync_only	NOT_USED
(10) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED

	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(11) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC205
(12) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC205
	arfcn	BA_LIST_205
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(13) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X20
	bsic	BSIC_5
(14) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(15) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC970
(16) RR_ACTIVATE_IND		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3749
	plmn	PLMN_ID_123
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES
(17) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(18) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC970

History:

13-Feb-03	LG	Initial

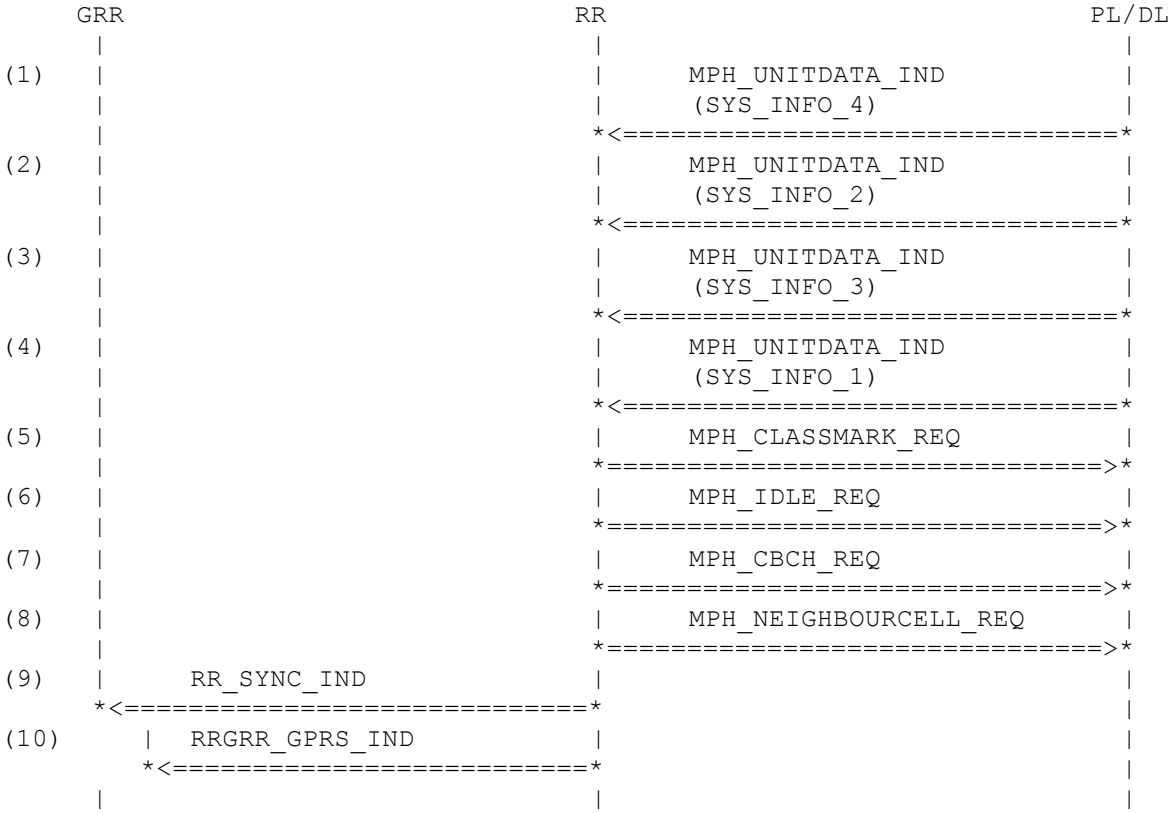
3.3.6 RRG206: Cell reselection (completion) - the new cell doesn't support GPRS

Description:

The cell reselection is completed. The new cell supports GPRS.

Preamble:

RRG204



Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED

	si4_rest_oct	SI4_REST_NOGPRS_TC206
	}	
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_3
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(3) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_NOGPRS_TC206
	}	
(4) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_2
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(5) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(6) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X20
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB

	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_NOT_USED
(7) MPH_CBCH_REQ		
	cbch	NOT_USED
(8) MPH_NEIGHBOURCELL_REQ		
	multi_band	NOT_USED
	arfcn	A_MPH_NCELL_3
	sync_only	NOT_USED
(9) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(10) RRGRR_GPRS_IND		
	cause	GPRS_NOT_SUPPORTED
	serving_cell_info	SERVING_CELL_TC205

History:

13-Feb-03	LG	Initial

3.3.7 RRG210: Extended Measurement initialisation

Description:

GRR requests the start of the extended measurement procedure of different types.

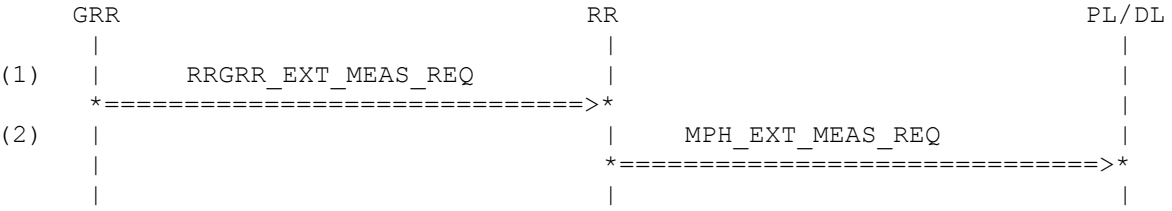
Variant A: report type 1 more than 6 carriers
B: report type 2 more than 6 carriers
C: report type 3 (more than 6 carriers)
D: report type 1 less than 6 carriers
E: report type 2 less than 6 carriers
F: as variant D, but different preamble

Variants:

<A>....<F>

Preamble:

<A>RRG102
<B>RRG102
<C>RRG102
<D>RRG102
<E>RRG102
<F>RRG225



Parametrization

Primitive	Parameter	Value
(1) RRGRR_EXT_MEAS_REQ		
<A>	arfcn_idx	ARFCN_IDX_ARR_1
<B>	arfcn_idx	ARFCN_IDX_ARR_1
<C>	arfcn_idx	ARFCN_IDX_ARR_1
<D>	arfcn_idx	ARFCN_IDX_ARR_2
<E>	arfcn_idx	ARFCN_IDX_ARR_2
<F>	arfcn_idx	ARFCN_IDX_ARR_2
	call_ref	CALL_REF_1
<A>	report_type	REP_TYPE_1
<B>	report_type	REP_TYPE_2
<C>	report_type	REP_TYPE_3
<D>	report_type	REP_TYPE_1
<E>	report_type	REP_TYPE_2
<F>	report_type	REP_TYPE_1
	ncc_permitted	NCC_PERMITTED_2_4
(2) MPH_EXT_MEAS_REQ		
	pch_interrupt	NO_PCH_INTERRUPT
	freq_bands	NOT_USED
<A>	num_of_chan	NUM_CHAN_8
<B>	num_of_chan	NUM_CHAN_8
<C>	num_of_chan	NUM_CHAN_8
<D>	num_of_chan	NUM_CHAN_4

<E>	num_of_chan	NUM_CHAN_4
<F>	num_of_chan	NUM_CHAN_4
<A>	arfcn	ARFCN_ARR_1
<B>	arfcn	ARFCN_ARR_1
<C>	arfcn	ARFCN_ARR_1
<D>	arfcn	ARFCN_ARR_2
<E>	arfcn	ARFCN_ARR_2
<F>	arfcn	ARFCN_ARR_2

History:

12-Apr-02	LG	Initial
11-Feb-03	LG	added variant F

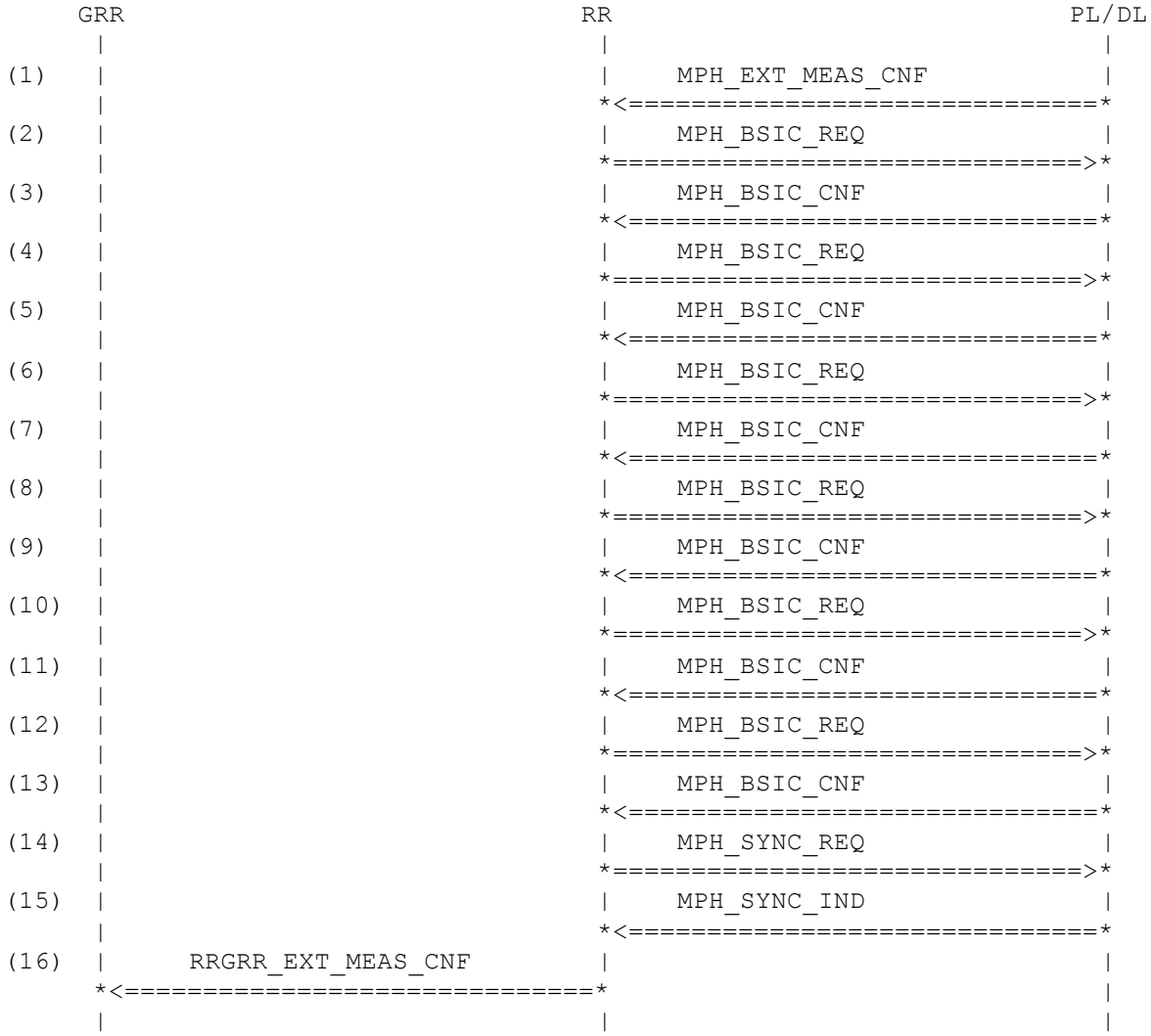
3.3.8 RRG212: Extended Measurement type 1 completion

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values. Because of the report_type of 1 the synchronisation and BSIC reading must be tried for the 6 strongest carriers before sending the result to GRR.

Preamble:

RRG210A



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_8
	arfcn	ARFCN_ARR_1_RES
	rx_lev	RXLEV_ARR_1

(2) MPH_BSIC_REQ		
	arfcn	ARFCN_32
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0_NCC2
	cs	CS_NO_ERROR
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_124
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_124
	bsic	BSIC_5_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC4
	cs	CS_NO_ERROR
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_527
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_527
	bsic	BSIC_3_NCC3
	cs	CS_NO_ERROR
(10) MPH_BSIC_REQ		
	arfcn	ARFCN_531
(11) MPH_BSIC_CNF		
	arfcn	ARFCN_531
	bsic	0
	cs	CS_NO_BCCH_AVAIL
(12) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(13) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_6_NCC2
	cs	CS_NO_ERROR
(14) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(15) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(16) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_1A
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_OK

History:

15-Apr-02	LG	Initial

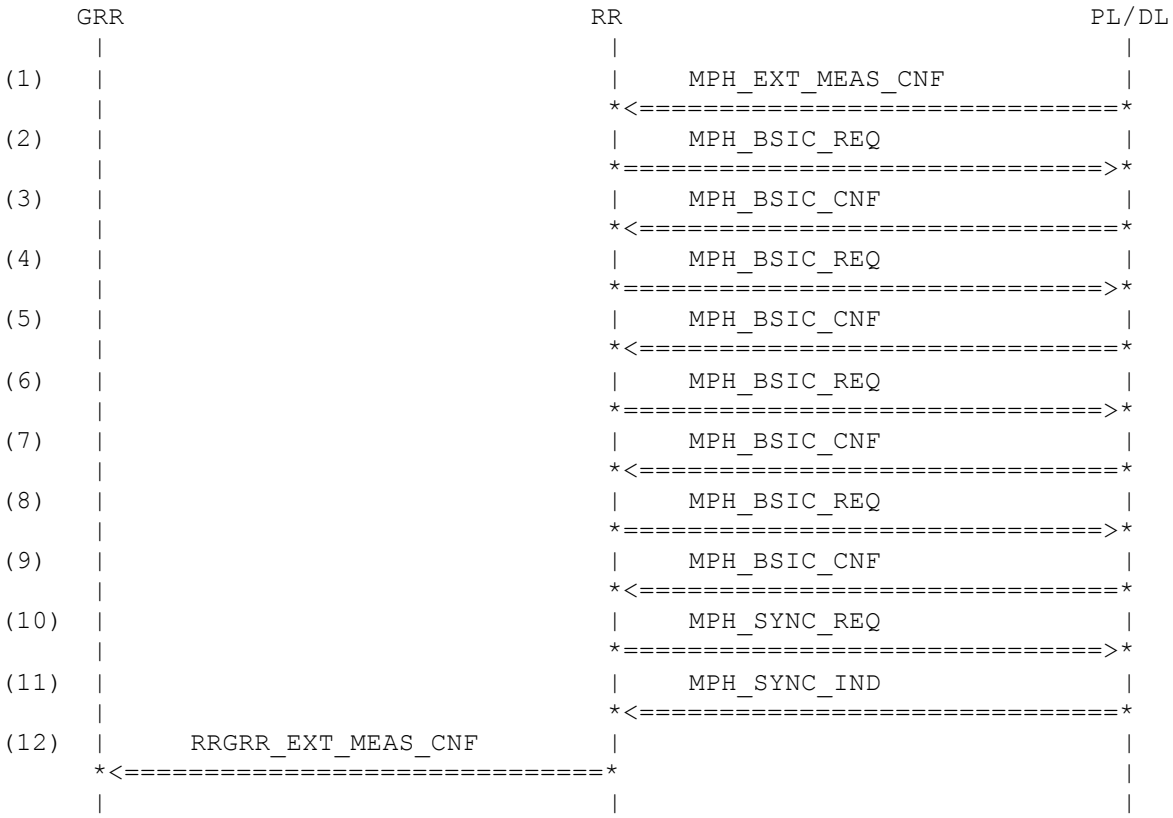
3.3.9 RRG213: Extended Measurement type 1 completion (short list)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values. Because of the report_type of 1 the synchronisation and BSIC reading must be tried for all carriers before sending the result to GRR.

Preamble:

RRG210D



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC4
	cs	CS_NO_ERROR
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_32

(5) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_6_NCC2
	cs	CS_NO_ERROR
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_800
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_800
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(10) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(11) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(12) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_1B
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_OK

History:

15-Apr-02	LG	Initial

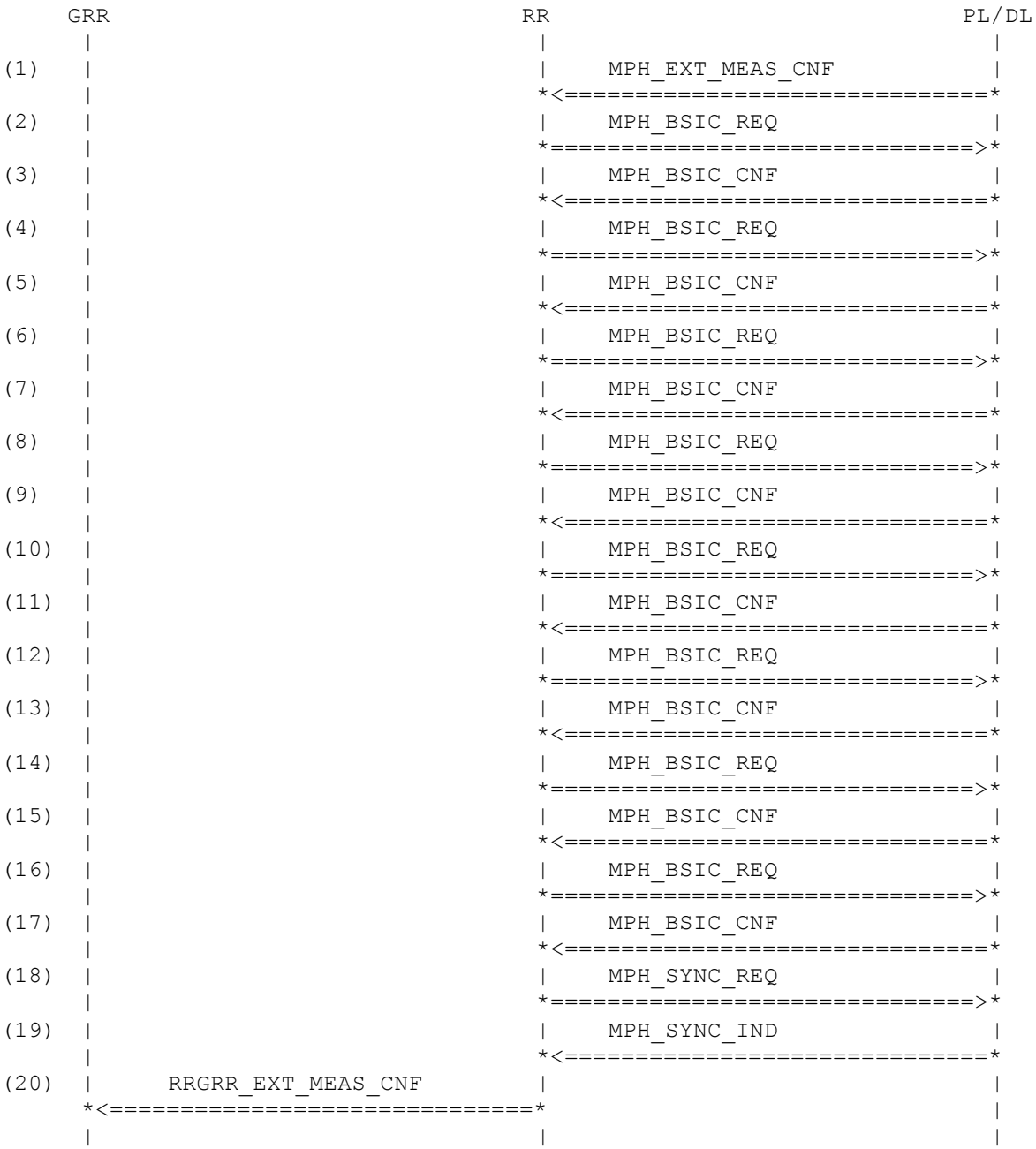
3.3.10 RRG214: Extended Measurement type 2 completion (short result)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values. Because of the report_type of 2 the synchronisation and BSIC reading must be tried to search for the 6 strongest carriers with successful BSIC reading of required NCC. All carriers must be read due to failing synchronisation for some of them before sending the result to GRR.

Preamble:

RRG210B



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_8
	arfcn	ARFCN_ARR_1_RES
	rx_lev	RXLEV_ARR_1
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_32
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_124
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_124
	bsic	BSIC_5_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_527
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_527
	bsic	BSIC_3_NCC3
	cs	CS_NO_ERROR
(10) MPH_BSIC_REQ		
	arfcn	ARFCN_531
(11) MPH_BSIC_CNF		
	arfcn	ARFCN_531
	bsic	BSIC_3_NCC2
	cs	CS_NO_ERROR
(12) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(13) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(14) MPH_BSIC_REQ		
	arfcn	ARFCN_115
(15) MPH_BSIC_CNF		
	arfcn	ARFCN_115

	bsic	BSIC_0_NCC4
	cs	CS_NO_ERROR
(16) MPH_BSIC_REQ		
	arfcn	ARFCN_800
(17) MPH_BSIC_CNF		
	arfcn	ARFCN_800
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(18) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(19) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(20) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_2A
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_OK

History:

15-Apr-02	LG	Initial

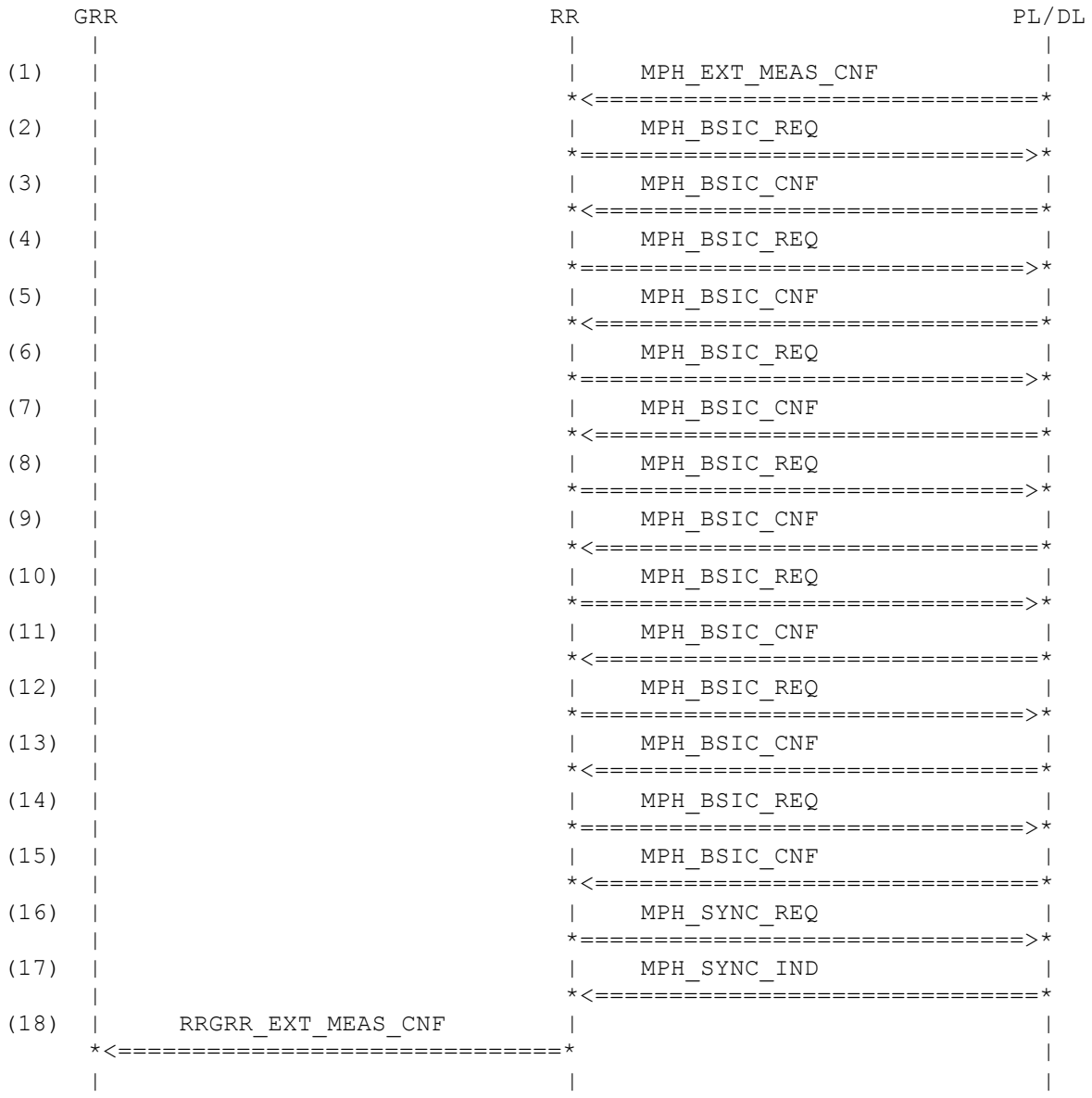
3.3.11 RRG215: Extended Measurement type 2 completion (full result)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values. Because of the report_type of 2 the synchronisation and BSIC reading must be tried to search for the 6 strongest carriers with successful BSIC reading of required NCC. Not all carriers must be read due to successfull reading 6 of them before sending the result to GRR.

Preamble:

RRG210B



Parametrization

Primitive	Parameter	Value

(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_8
	arfcn	ARFCN_ARR_1_RES
	rx_lev	RXLEV_ARR_1
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_32
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0_NCC2
	cs	CS_NO_ERROR
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_124
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_124
	bsic	BSIC_5_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC4
	cs	CS_NO_ERROR
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_527
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_527
	bsic	BSIC_3_NCC3
	cs	CS_NO_ERROR
(10) MPH_BSIC_REQ		
	arfcn	ARFCN_531
(11) MPH_BSIC_CNF		
	arfcn	ARFCN_531
	bsic	BSIC_3_NCC2
	cs	CS_NO_ERROR
(12) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(13) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_6_NCC2
	cs	CS_NO_ERROR
(14) MPH_BSIC_REQ		
	arfcn	ARFCN_115
(15) MPH_BSIC_CNF		
	arfcn	ARFCN_115
	bsic	BSIC_0_NCC4
	cs	CS_NO_ERROR

(16) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(17) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(18) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_2B
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_OK

History:

15-Apr-02	LG	Initial

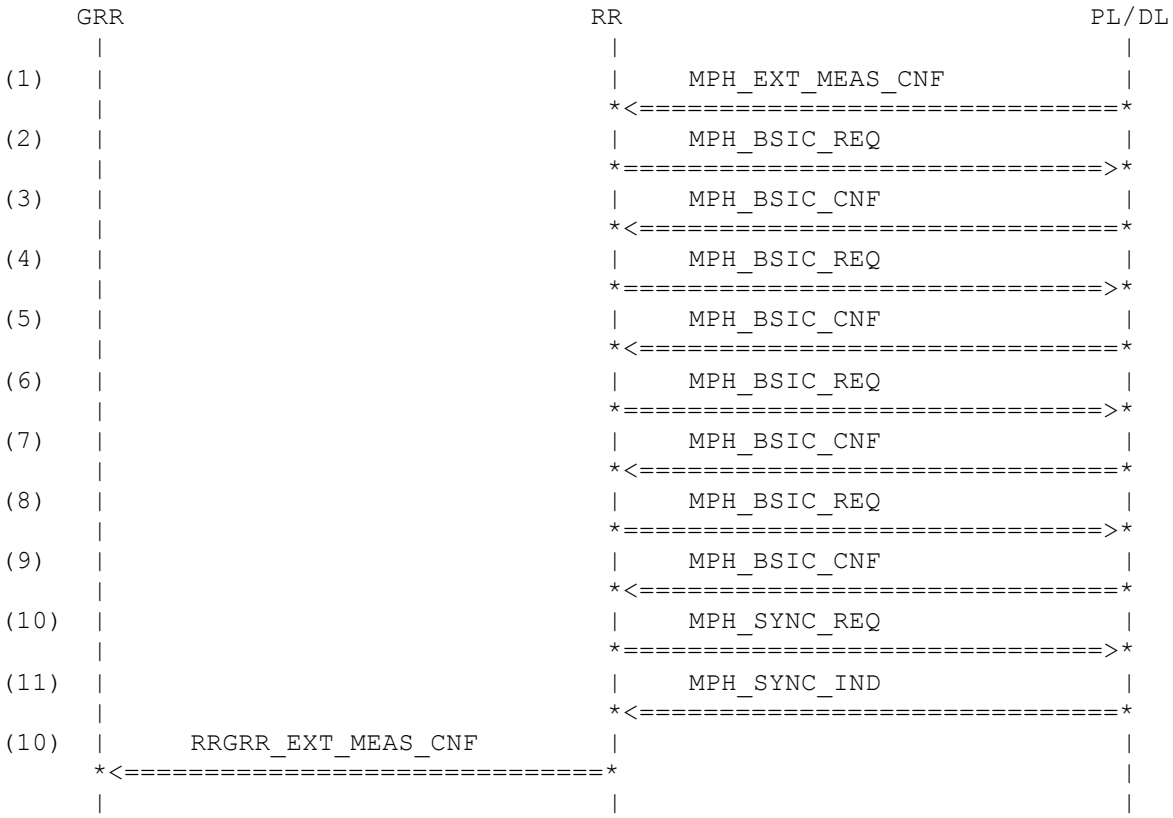
3.3.12 RRG216: Extended Measurement type 2 completion (short list)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values. Because of the report_type of 2 the synchronisation and BSIC reading must be tried for all carriers before sending the result to GRR.

Preamble:

RRG210E



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC3
	cs	CS_NO_ERROR
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_32

(5) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_6_NCC2
	cs	CS_NO_ERROR
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_800
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_800
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(10) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(11) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(12) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_2C
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_OK

History:

15-Apr-02	LG	Initial

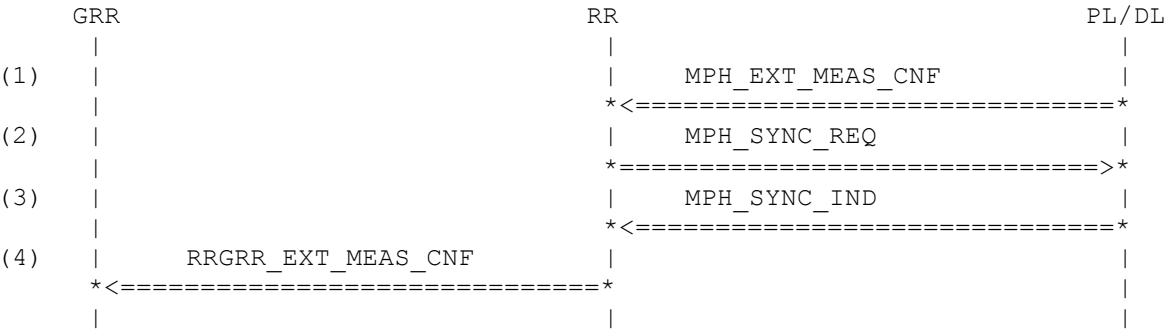
3.3.13 RRG217: Extended Measurement type 3 completion

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values. Because of the report_type of 3 no synchronisation and BSIC reading is needed.

Preamble:

RRG210C



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_8
	arfcn	ARFCN_ARR_1_RES
	rx_lev	RXLEV_ARR_1
(2) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(3) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(4) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_3
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_OK

History:

15-Apr-02	LG	Initial

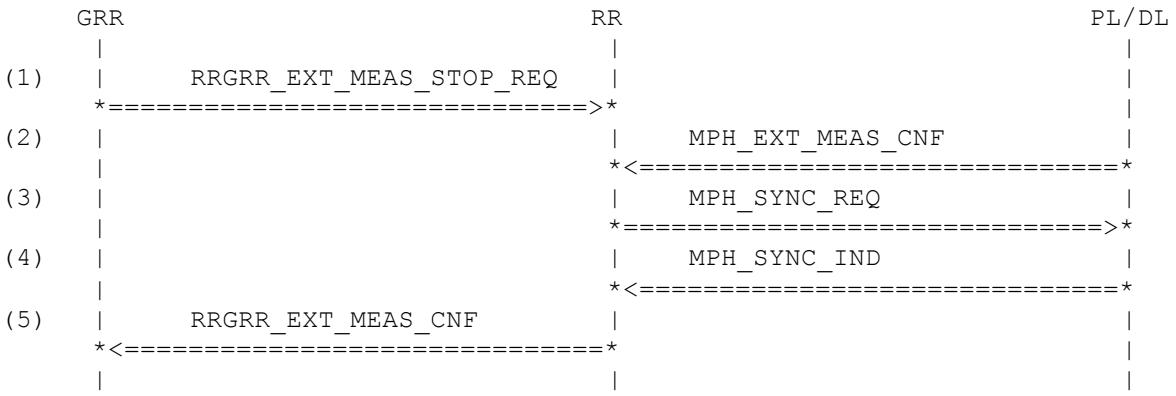
3.3.14 RRG220: Extended Measurement - premature end (early)

Description:

The extended measurement procedure has been started. ALR has not yet responded with the sorted measurement values. GRR stops the extended measurement procedure. After getting the measurements there is performed no synchronisation. After stopping the extended measuerement procedure in ALR the end of extended measurement procedure is indicated to GRR.

Preamble:

RRG210D



Parametrization

Primitive	Parameter	Value
(1) RRGRR_EXT_MEAS_STOP_REQ		
(2) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(3) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(4) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(5) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_EMPTY
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_RESET

History:

10-Feb-03	LG	Initial

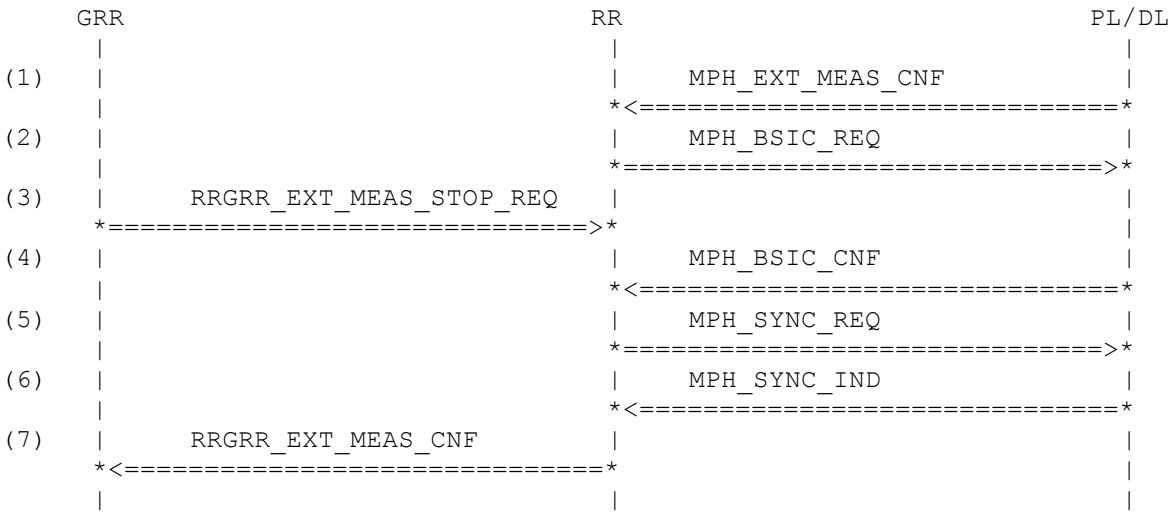
3.3.15 RRG221: Extended Measurement - premature end (middle)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values and the BSIC synchronization is in progress. GRR stops the extended measurement procedure. No more synchronisation is performed. After stopping the extended measurement procedure in ALR the end of extended measurement procedure is indicated to GRR.

Preamble:

RRG210D



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(3) RRGRR_EXT_MEAS_STOP_REQ		
(4) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC3
	cs	CS_NO_ERROR
(5) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(6) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(7) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_1C

			call_ref	CALL_REF_1
			xmeas_cause	EXT_MEAS_RESET

History:

10-Feb-03	LG	Initial	

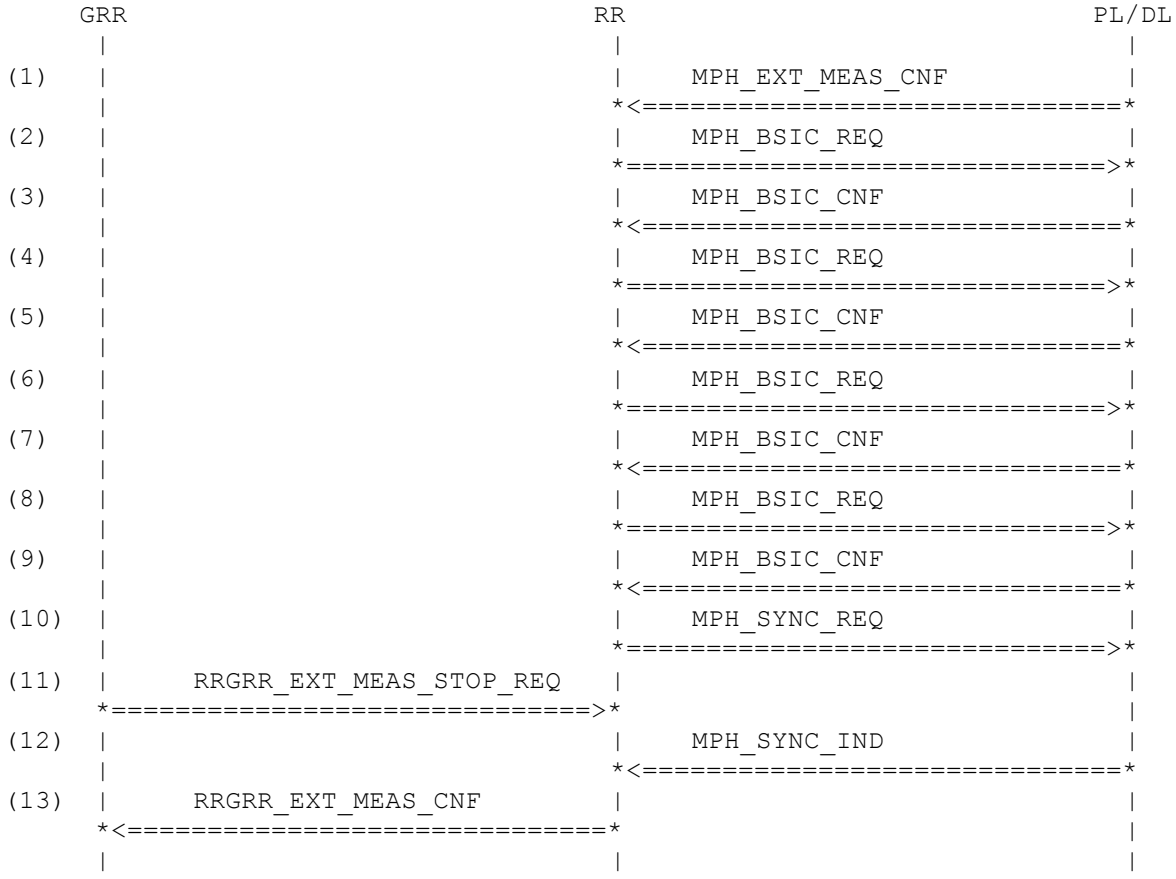
3.3.16 RRG222: Extended Measurement - premature end (late)

Description:

The extended measurement procedure has been started. When the procedure is almost completed (RR is stopping the procedure in ALR), GRR stops the extended measurement procedure. The end of extended measurement procedure is indicated to GRR.

Preamble:

RRG210D



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC4
	cs	CS_NO_ERROR

(4) MPH_BSIC_REQ		
	arfcn	ARFCN_32
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_6_NCC2
	cs	CS_NO_ERROR
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_800
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_800
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(10) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(11) RRGRR_EXT_MEAS_STOP_REQ		
(12) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(13) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_1B
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_RESET

History:

10-Feb-03	LG	Initial

3.3.17 RRG225: Extended Measurement interruption by cell reselection (preparation)

Description:

After this test case the extended measurement procedure will be started (see RRG210). ALR sends a measurement indication, which will later trigger a cell reselection.

Preamble:

RRG102

Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	

History:

11-Feb-03	LG	Initial

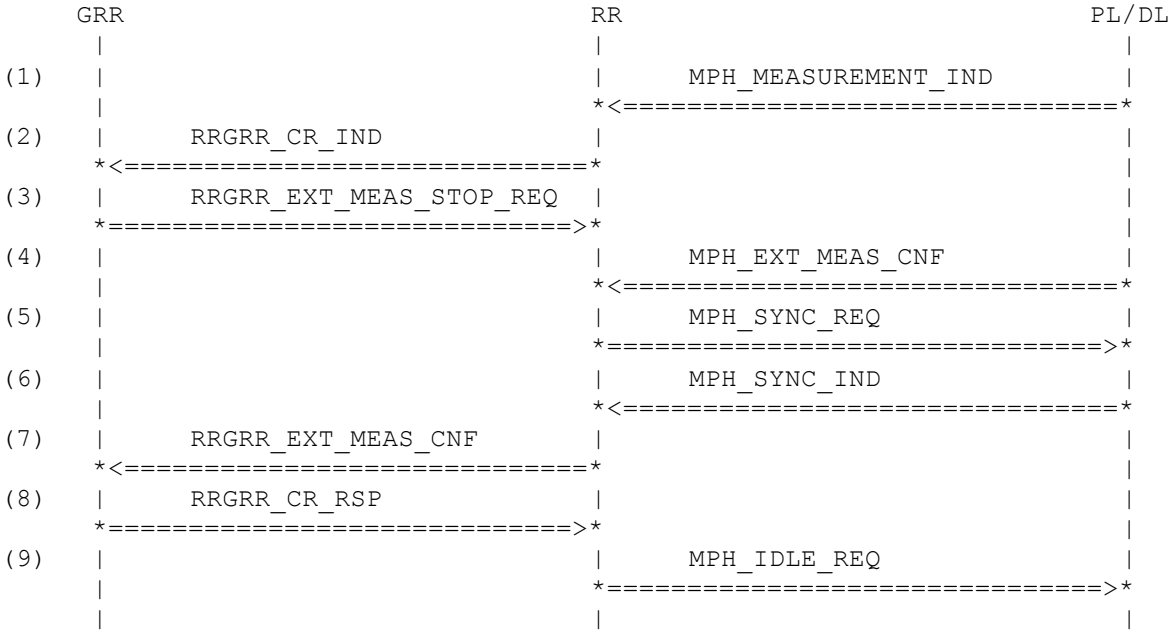
3.3.18 RRG226: Extended Measurement interruption by cell reselection (early)

Description:

The extended measurement procedure has been started. ALR has not yet responded with the sorted measurement values, but sends a measurement indication, which triggers a cell reselection. The extended measurement procedure is stopped and the cell reselection is started.

Preamble:

RRG210F



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(2) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(3) RRGRR_EXT_MEAS_STOP_REQ		
(4) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4

	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(5) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(6) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(7) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_EMPTY
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_RESET
(8) RRGRR_CR_RSP		
(9) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X20
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NOT_PRESENT_8BIT
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT

History:

11-Feb-03	LG	Initial

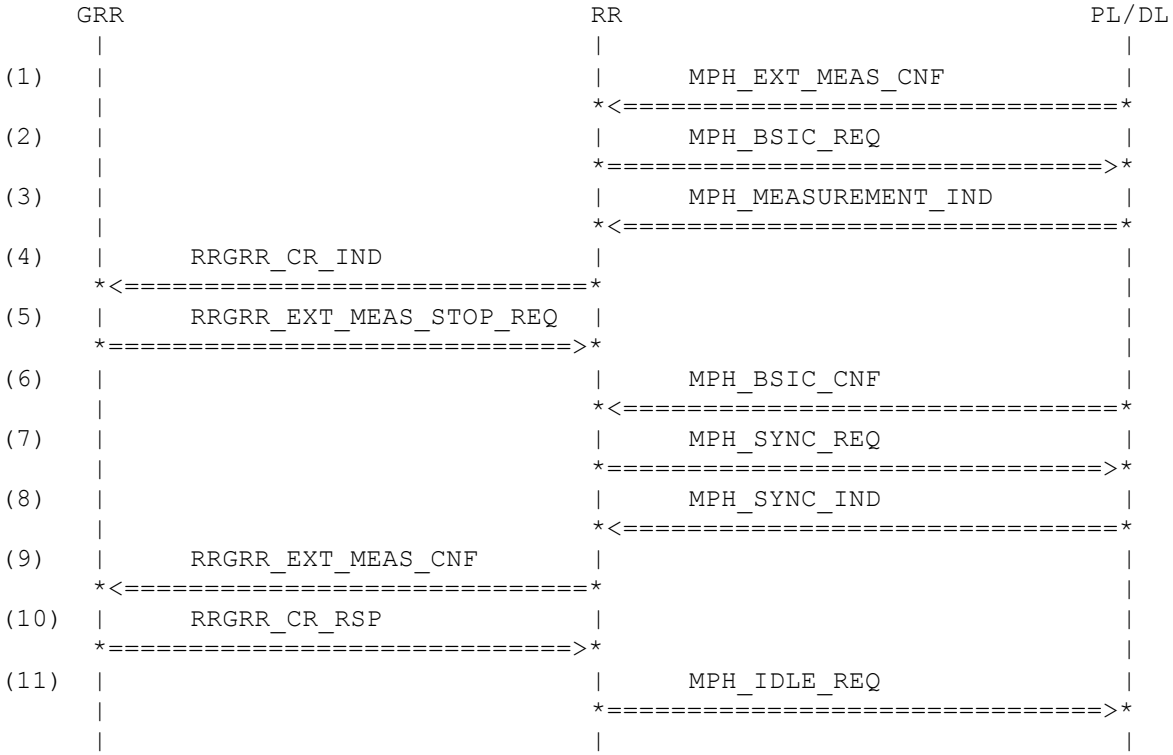
3.3.19 RRG227: Extended Measurement interruption by cell reselection (middle)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values, and the BSIC synchronization is in progress. ALR sends a measurement indication, which triggers a cell reselection. The extended measurement procedure is stopped and the cell reselection is started.

Preamble:

RRG210F



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(3) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27

	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(4) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(5) RRGRR_EXT_MEAS_STOP_REQ		
(6) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC3
	cs	CS_NO_ERROR
(7) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(8) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(9) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_1C
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_RESET
(10) RRGRR_CR_RSP		
(11) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X20
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NOT_PRESENT_8BIT
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT

History:

12-Feb-03	LG	Initial

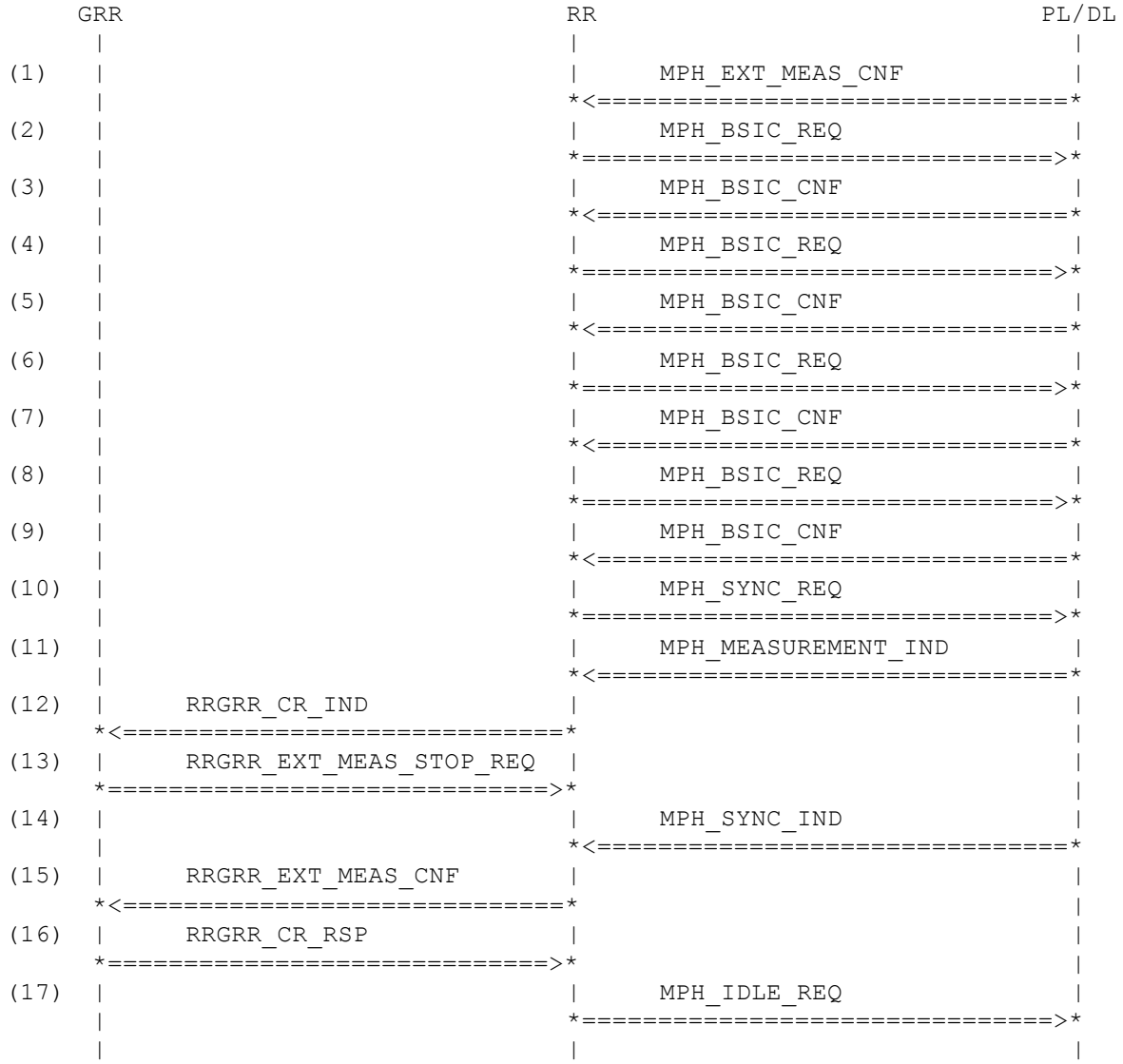
3.3.20 RRG228: Extended Measurement interruption by cell reselection (late)

Description:

The extended measurement procedure has been started. ALR responds with the sorted measurement values and the BSIC synchronization is performed. When RR erequest to stop the procedure, ALR sends a measurement indication, which triggers a cell reselection. The extended measurement procedure is stopped and the cell reselection is started.

Preamble:

RRG210F



Parametrization

Primitive	Parameter	Value
(1) MPH_EXT_MEAS_CNF		
	num_of_chan	NUM_CHAN_4
	arfcn	ARFCN_ARR_2_RES
	rx_lev	RXLEV_ARR_2

(2) MPH_BSIC_REQ		
	arfcn	ARFCN_67
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_67
	bsic	BSIC_3_NCC4
	cs	CS_NO_ERROR
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_32
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_32
	bsic	BSIC_0_NCC2
	cs	CS_NO_ERROR
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_24
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_24
	bsic	BSIC_6_NCC2
	cs	CS_NO_ERROR
(8) MPH_BSIC_REQ		
	arfcn	ARFCN_800
(9) MPH_BSIC_CNF		
	arfcn	ARFCN_800
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL
(10) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(12) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_10
	rx_lev_sub	RX_LEV_10
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_2_CR_CANDIDATE
	gprs_sync	NORMAL_MEAS_REP
(13) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(14) RRGRR_EXT_MEAS_STOP_REQ		
(14) MPH_SYNC_IND		
	cs	CS_STOP_PLMN_SEARCH
	arfcn	NOT_USED
(15) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_1B
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_RESET

(16) RRGRR_CR_RSP

(17) MPH_IDLE_REQ

mod	MODE_CELL_RESELECTION
arfcn	ARFCN_X20
ext_bcch	BSIC_5
comb_ccch	CCD_CCCH_1_NOT_COMB
tn	TN_0
dlt	DLT_23
pg	PG_0
bs_ag_blocks_res	BS_AG_BLKS_RES_5
bs_pa_mfrms	BS_PA_MFRMS_2
power	MS_TXPWR_MAX_CCH_02
ncc_permitted	NOT_PRESENT_8BIT
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	NOT_PRESENT_8BIT

History:

12-Feb-03	LG	Initial

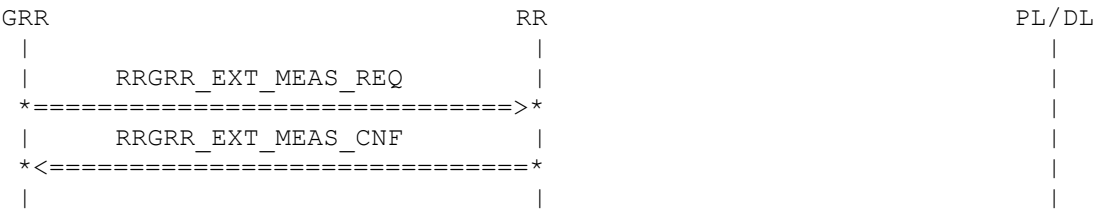
3.3.21 RRG230: Extended Measurement request while in cell reselection

Description:

Cell reselection is in progress. GRR requires the extended measurement procedure. The request is rejected.

Preamble:

RRG204



Parametrization

Primitive	Parameter	Value
(1) RRGRR_EXT_MEAS_REQ		
	arfcn_idx	ARFCN_IDX_ARR_2
	call_ref	CALL_REF_1
	report_type	REP_TYPE_1
	ncc_permitted	NCC_PERMITTED_2_4
(2) RRGRR_EXT_MEAS_CNF		
	xmeas_res	XMEAS_RES_EMPTY
	call_ref	CALL_REF_1
	xmeas_cause	EXT_MEAS_RESET

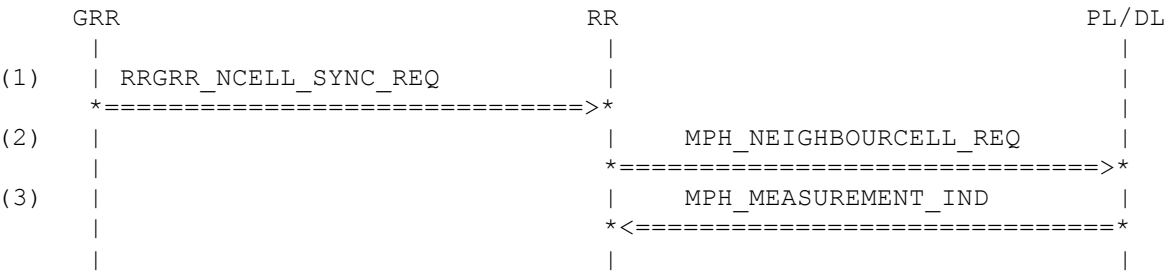
History:

17-Feb-03	LG	Initial

3.3.22 RRG248: Ncell Synchronization requests from GRR in PIM_PBCCH, empty

Description:

Preamble:
RRG106



Parametrization

Primitive	Parameter	Value
(1) RRGRR_NCELL_SYNC_REQ		
	arfcn	ARFCN_LIST_EMPTY_PBCCH
(2) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	ARFCN_LIST_EMPTY_PBCCH
	sync_only	SYNC_LIST
(3) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_EMPTY
	gprs_sync	SYNC_RESULTS

History:

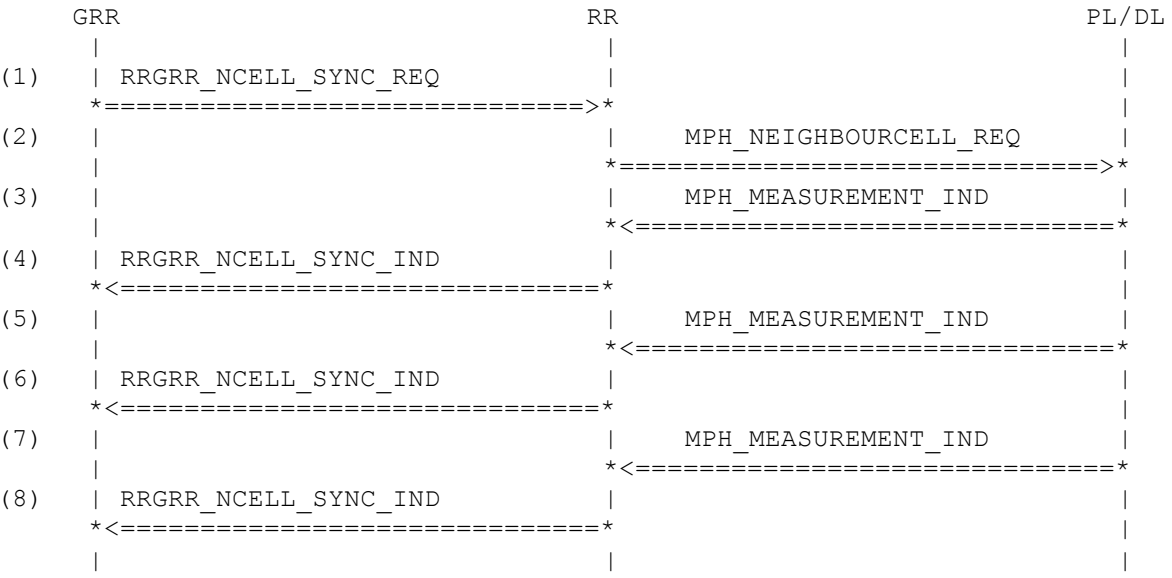
02-Sep-02	MPA	Initial

3.3.23 RRG249: Ncell Synchronization requests from GRR in PIM_PBCCH, full

Description:

Preamble:

RRG106



Parametrization

Primitive	Parameter	Value
(1) RRGRR_NCELL_SYNC_REQ		
	arfcn	ARFCN_LIST_FULL_PBCCH
(2) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	ARFCN_LIST_FULL_PBCCH
	sync_only	SYNC_LIST
(3) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_1
	gprs_sync	SYNC_RESULTS
(4) RRGRR_NCELL_SYNC_IND		
	sync_result	A_LIST_RES
(5) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43

	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_11
	gprs_sync	SYNC_RESULTS
(6) RRGRR_NCELL_SYNC_IND		
	sync_result	A_LIST_RES_11
(7) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_1_UNK
	gprs_sync	SYNC_RESULTS
(8) RRGRR_NCELL_SYNC_IND		
	sync_result	A_LIST_RES_UNK

History:

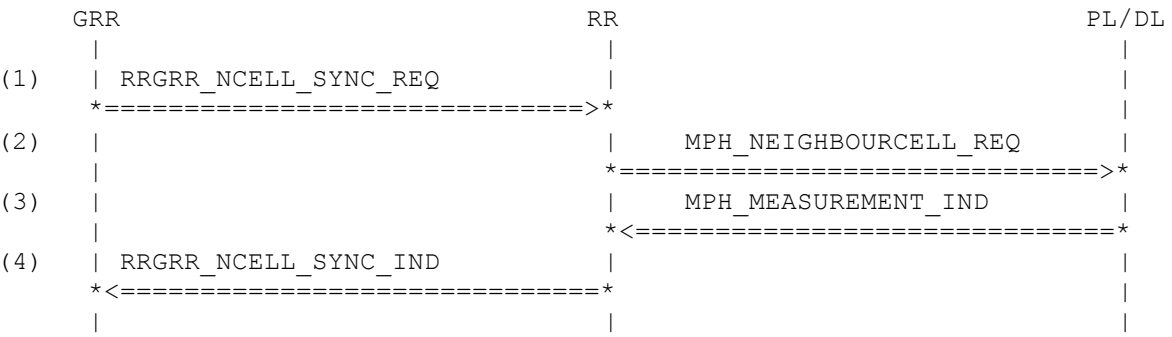
02-Sep-02	MPA	Initial

3.3.24 RRG250: Ncell Synchronization requests from GRR in PIM_PBCCH

Description:

Preamble:

RRG106



Parametrization

Primitive	Parameter	Value

(4) RRGRR_NCELL_SYNC_REQ		
	arfcn	ARFCN_LIST124_PBCCH2
(5) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	MPH_ARFCN_LIST124_PBCCH
	sync_only	SYNC_LIST
(6) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_1
	gprs_sync	SYNC_RESULTS
(7) RRGRR_NCELL_SYNC_IND		
	sync_result	A_LIST_RES

History:

02-Sep-02	MPA	Initial

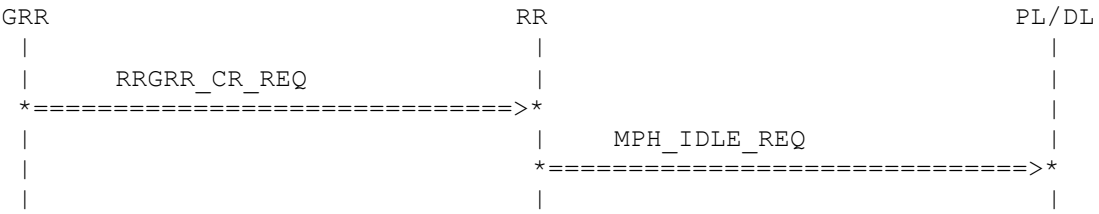
3.3.25 RRG251: RRGRR_CR_REQ (CR_NEW)

Description:

This primitive requests that RR shall reselect a new cell.

Preamble:

RRG250



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CR_REQ		
	cr_type	CR_NEW
	arfcn	ARFCN_124
	bsic	BSIC_3
(2) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_124
	ext_bcch	BSIC_3
	comb_ccch	NOT_USED
	tn	NOT_USED

dlt	NOT_USED
pg	NOT_USED
bs_ag_blocks_res	NOT_USED
bs_pa_mfrms	NOT_USED
power	NOT_USED
ncc_permitted	NOT_PRESENT_8BIT
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	NOT_PRESENT_8BIT

History:

21-July-00	MSE	Initial
31-Oct-00	MSE	individual params instead of sdu
02-Sep-02	MPA	changed

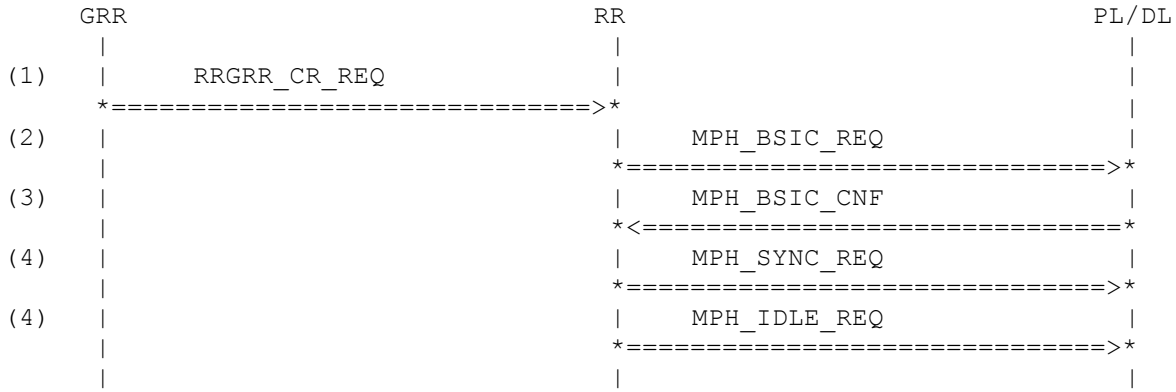
3.3.26 RRG252: RRGRR_CR_REQ (CR_NEW_NOT_SYNCED)

Description:

This primitive requests that RR shall reselect a new cell. XXX The feature tested by this case is not implemented. The test will fail therefore XXX.

Preamble:

RRG250



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CR_REQ		
	cr_type	CR_NEW_NOT_SYNCED
	arfcn	ARFCN_3
	bsic	BSIC_1
(2) MPH_BSIC_REQ		
	arfcn	ARFCN_3
(3) MPH_BSIC_CNF		
	arfcn	ARFCN_3
	bsic	BSIC_1
	cs	CS_NO_ERROR
(4) MPH_SYNC_REQ		
	cs	CS_STOP_BCCH_READING

(5) MPH_IDLE_REQ

mod	MODE_CELL_RESELECTION
arfcn	ARFCN_3
ext_bcch	BSIC_1
comb_ccch	NOT_USED
tn	NOT_USED
dlt	NOT_USED
pg	NOT_USED
bs_ag_blocks_res	NOT_USED
bs_pa_mfrms	NOT_USED
power	NOT_USED
ncc_permitted	NOT_PRESENT_8BIT
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	NOT_PRESENT_8BIT

History:

02-Sep-02	MPA	initial

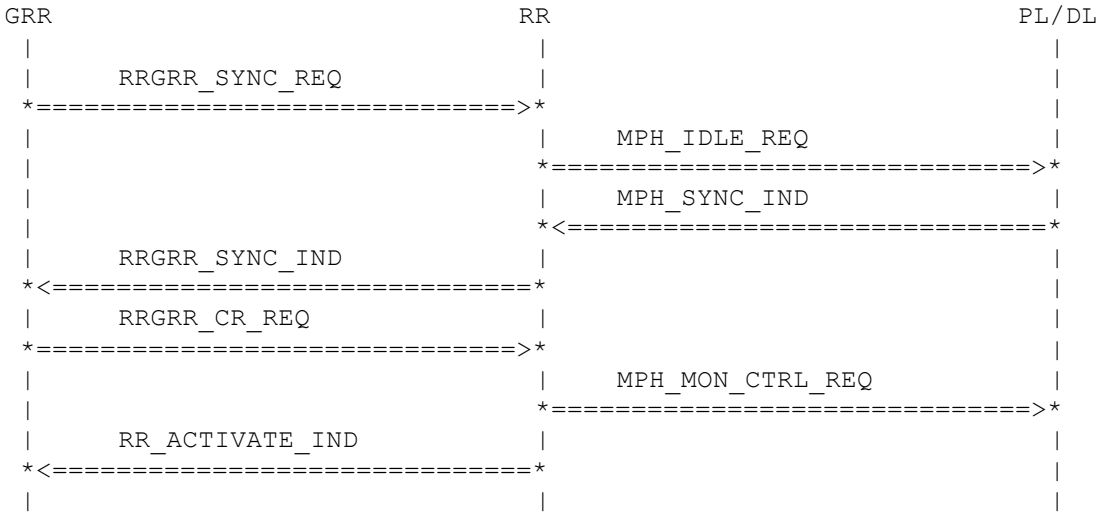
3.3.27 RRG253: RRGRR_SYNC_REQ

Description:

This primitive requests that RR shall reselect a new cell. GRR knows that the new cell has a PBCCH so we only need to sycn to it. XXX There is still a problem: the CID and LAI have to be passed from GRR in the CR_REQ compl. XXX not impl

Preamble:

RRG250



Parametrization

Primitive	Parameter	Value
(1) RRGRR_SYNC_REQ		
	arfcn	ARFCN_67
	bsic	BSIC_0
(2) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION_SYNC_ONLY

	arfcn	ARFCN_67
	ext_bcch	BSIC_0
	comb_ccch	NOT_USED
	tn	NOT_USED
	dlt	NOT_USED
	pg	NOT_USED
	bs_ag_blocks_res	NOT_USED
	bs_pa_mfrms	NOT_USED
	power	NOT_USED
	ncc_permitted	NOT_PRESENT_8BIT
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT
(3) MPH_SYNC_IND		
	cs	CS_SCELL_IS_SYNCED
	arfcn	ARFCN_67
(4) RRGRR_SYNC_IND		
	sync_res	SYNC_OK
(5) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_67
	bsic	BSIC_0
(6) MPH_MON_CTRL_REQ		
	action	ENTER_PIM_PBCCH
	si_to_read	NOT_USED
(7) RR_ACTIVATE_IND		
	op	NOT_USED
	mm_info	NOT_USED
	cid	NOT_USED
	plmn	NOT_USED
	lac	NOT_USED
	power	RF_CLASS_4
	gprs_indication	NOT_USED

History:

02-Sep-02	MPA	initial

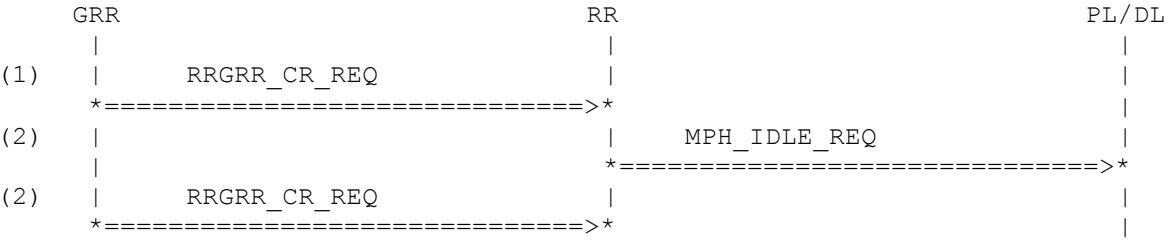
3.3.28 RRG254: RRGRR_CR_REQ (CR_NEW) followed by CR_CS

Description:

This primitive requests that RR shall reselect a new cell.

Preamble:

RRG250



```
(3)   |                                  |     MPH_POWER_REQ                |
      |                                  *===============================>*
      |                                  |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) RRGRR_CR_REQ		
	cr_type	CR_NEW
	arfcn	ARFCN_124
	bsic	BSIC_3
(2) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_124
	ext_bcch	BSIC_3
	comb_ccch	NOT_USED
	tn	NOT_USED
	dlt	NOT_USED
	pg	NOT_USED
	bs_ag_blocks_res	NOT_USED
	bs_pa_mfrms	NOT_USED
	power	NOT_USED
	ncc_permitted	NOT_PRESENT_8BIT
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT
(3) RRGRR_CR_REQ		
	cr_type	CR_CS
	arfcn	NOT_PRESENT_16BIT
	bsic	RRGRR_INVALID_BSIC
(4) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A

History:

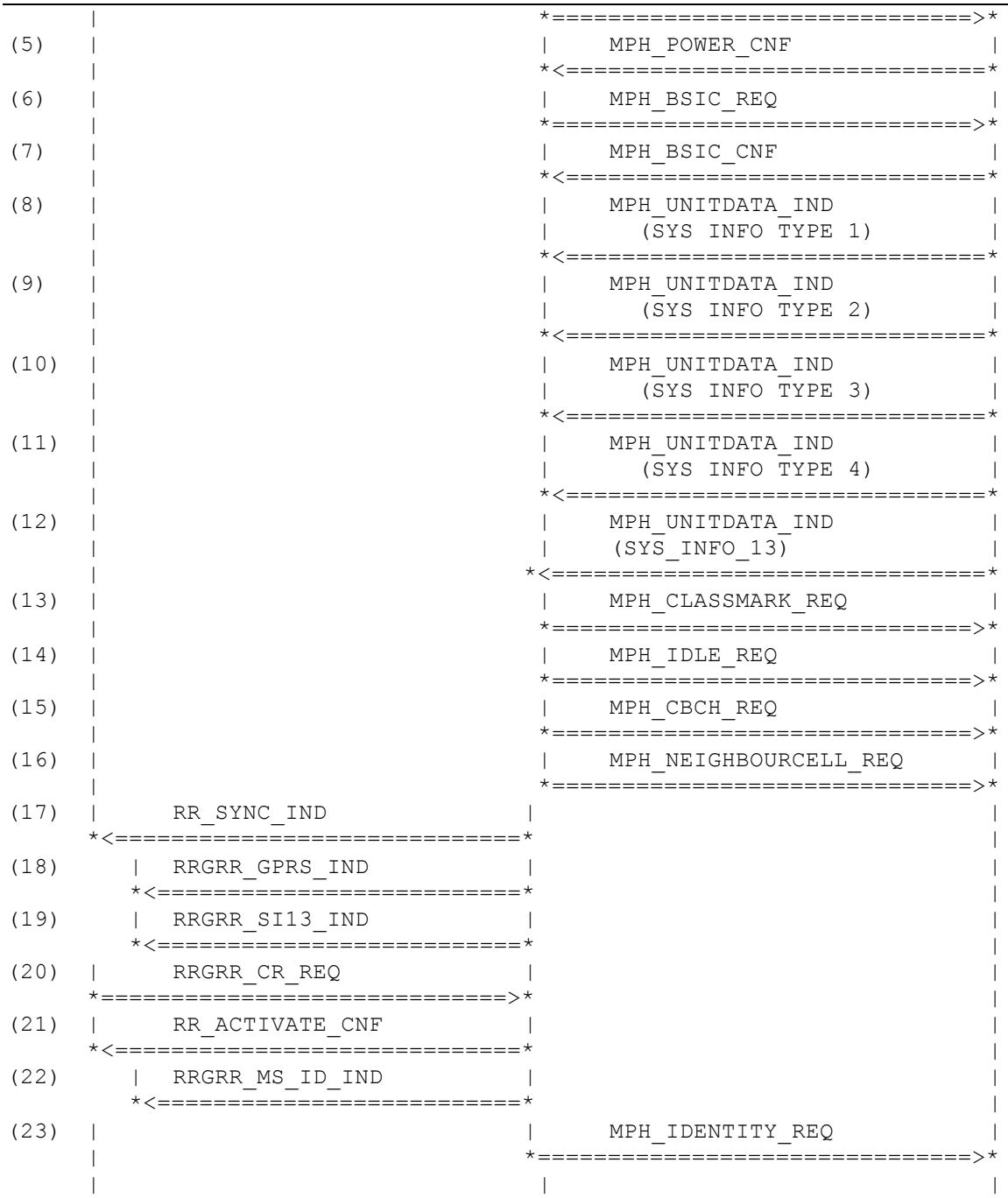
02-Sep-02	MPA	initial

3.3.29 RRG261: Start by MM for full service(HPLMN), successful, with bcch info

Description: MM starts an attempt for full service for a visiting PLMN in the HPLMN country. The request contains BCCH information from the SIM card. The attempt is successful. The channel ARFCN_20 shall be used. It is member of the BCCH information. It is expected that the registration timer in RR for searching the HPLMN is started with 2 Minutes.

Preamble: RRG000

```
      MM                                RR                                PL/DL
      |                                  |                                  |
COMMAND (RR CONFIG TIMER_SET=<TREG, 400>)
(1)   |      RR_ACTIVATE_REQ            |                                  |
      *===============================>*                                  |
(2)   |      RRGRR_CR_IND              |                                  |
      *<==============================*                                   |
(3)   |      RRGRR_CR_RSP              |                                  |
      *===============================>*                                  |
(4)   |                                  |     MPH_POWER_REQ                |
```



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123_V
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN

	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	S_BCCH_INFO_20
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(3) RRGRR_CR_RSP		
(4) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	NOT_USED
(5) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_X20
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_X20
	bsic	BSIC_5
	cs	CS_NO_ERROR
(8) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	}	
(9) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_1
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(10) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	

	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147_V
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(11) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147_V
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(12) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(13) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(14) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X20
	ext_bcch	NOT_USED
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02

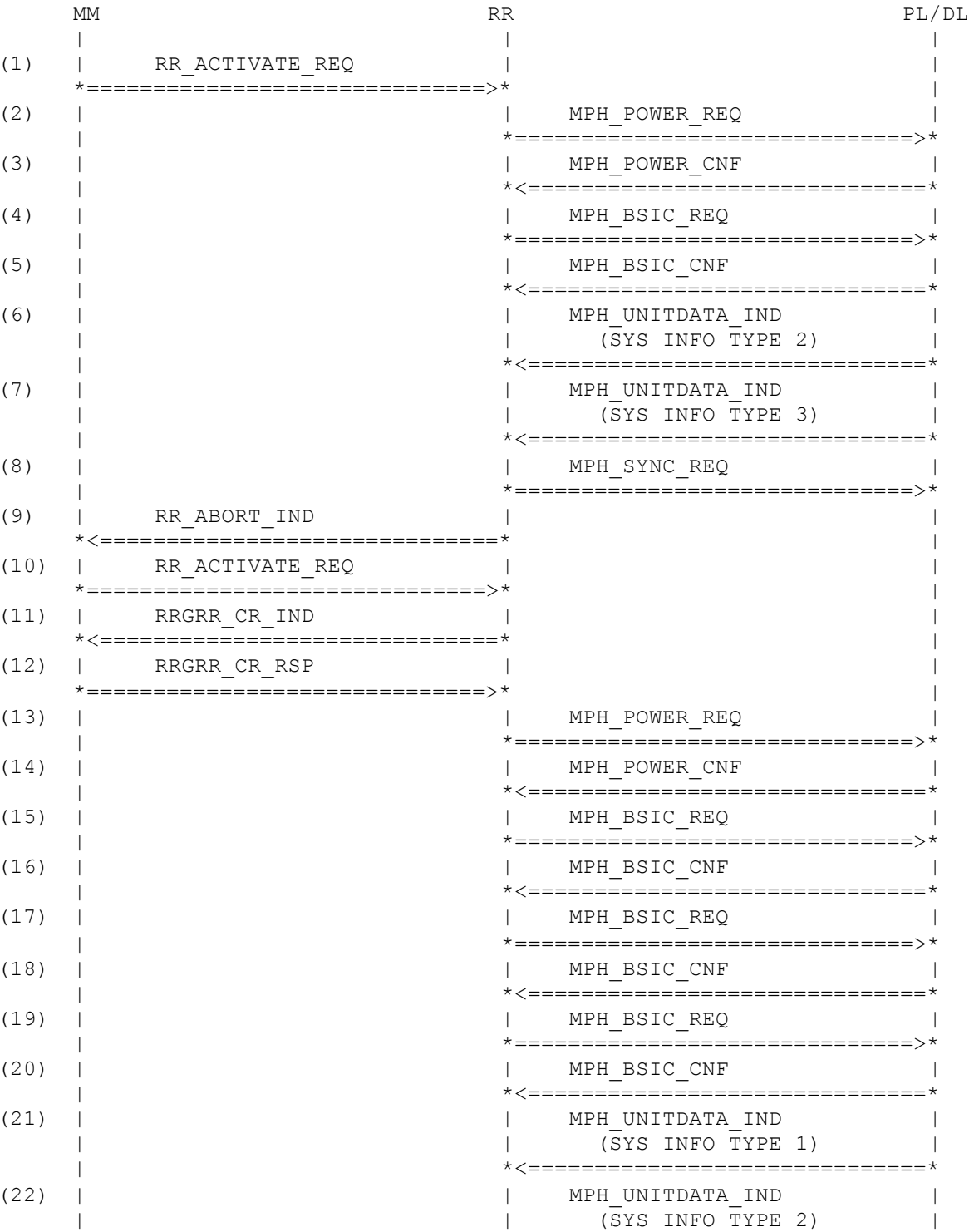
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NOT_USED
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(15) MPH_CBCH_REQ		
	cbch	NOT_USED
(16) MPH_NEIGHBOURCELL_REQ		
	multi_band	NOT_USED
	arfcn	A_MPH_NCELL_1C
	sync_only	NOT_USED
(17) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(18) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC651
(19) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC651
	arfcn	BA_LIST_651
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(20) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X20
	bsic	BSIC_5
(21) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3748
	plmn	PLMN_ID_123_V
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES
(22) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(23) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702

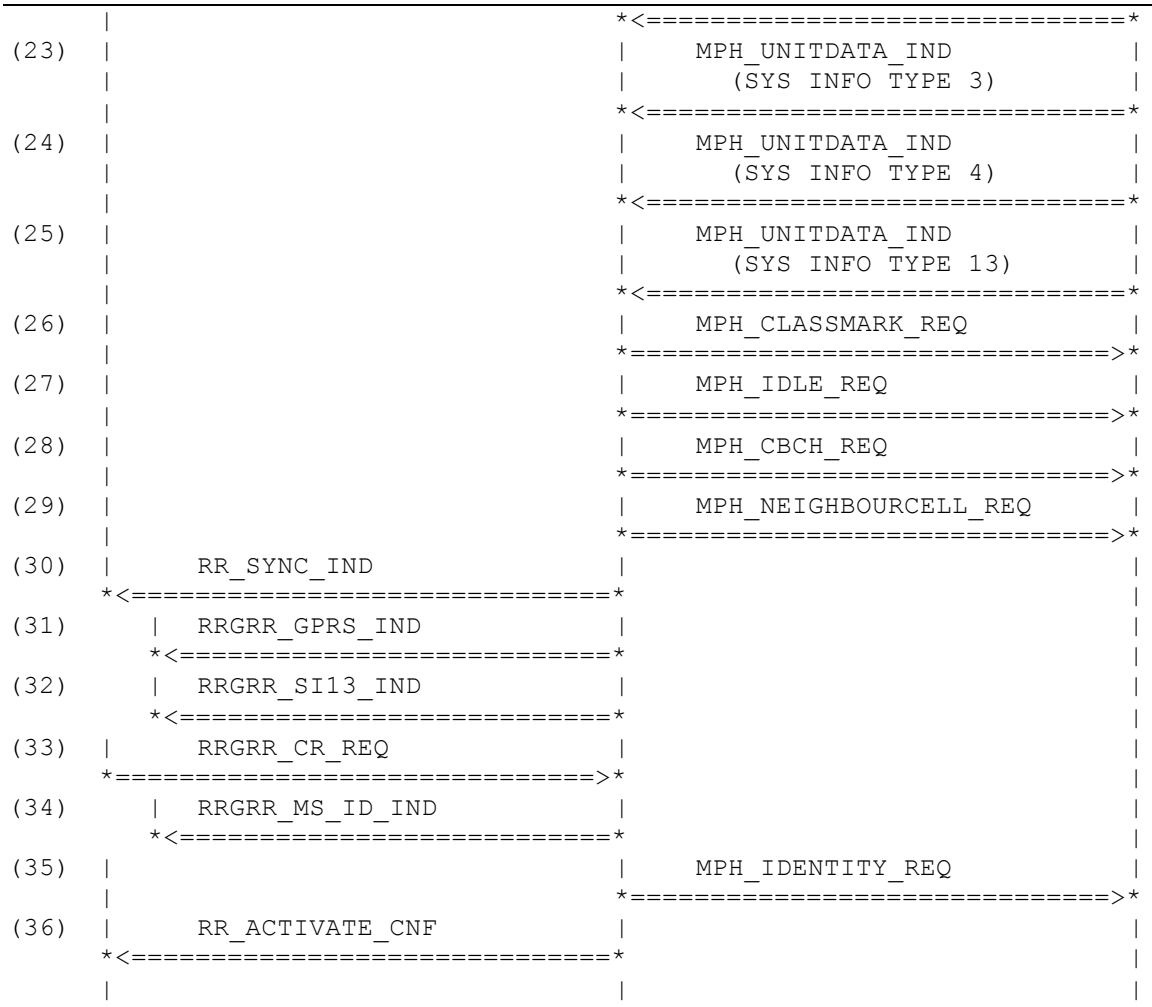
History:	22.10.02	MPA	Initial

3.3.30 RRG262: Restart by RR for full service (HPLMN) is successful

Description: MM has requested a full service search for the VPLMN in the HPLMN country and RR is in full service. After timeout of the registration timer RR starts the full service search for HPLMN. The search is successful and MM is informed.

Preamble: RRG261





Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	NOT_USED
	op	OP_MODE_NET_SRCH_MMI
	cksn	NOT_USED
	kcv	NOT_USED
	accc	NOT_USED
	imsi_struct	NOT_USED
	tmsi_struct	NOT_USED
	thplmn	NOT_USED
	bcch_info	NOT_USED
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) MPH_POWER_REQ		
	pch_interrupt	NO_PCH_INTERRUPT
	freq_bands	NOT_USED
(3) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3

	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_1
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(7) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_32_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(8) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(9) RR_ABORT_IND		
	op	OP_MODE_NET_SRCH_MMI
	cause	RRCS_ABORT_CEL_SEL_FAIL
	plmn_avail	TWO_PLMN_AVAILABLE
	plmn	TWO_PLMNS
	lac_list	TWO_LACS
	rxlevel	TWO_RXLEVS
	power	RF_CLASS_4
(10) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123_32
	op	OP_MODE_TEST_SIM

	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	NOT_USED
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(11) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(12) RRGRR_CR_RSP		
(13) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	NOT_USED
(14) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(15) MPH_BSIC_REQ		
	arfcn	ARFCN_X20
(16) MPH_BSIC_CNF		
	arfcn	ARFCN_X20
	bsic	BSIC_5
	cs	CS_NO_BCCH_AVAIL
(17) MPH_BSIC_REQ		
	arfcn	ARFCN_124
(18) MPH_BSIC_CNF		
	arfcn	ARFCN_124
	bsic	BSIC_5
	cs	CS_NO_BCCH_AVAIL
(19) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(20) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(21) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1

	rach_ctrl	RACH_CTRL_1
	}	
(22) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_1
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(23) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_32_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(24) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_32_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(25) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	

	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(26) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(27) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	NOT_USED
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NOT_USED
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(28) MPH_CBCH_REQ		
	cbch	NOT_USED
(29) MPH_NEIGHBOURCELL_REQ		
	multi_band	NOT_USED
	arfcn	A_MPH_NCELL_1
	sync_only	NOT_USED
(30) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(31) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702_32
(32) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702_32
	arfcn	BA_LIST_651
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	

(33)	RRGRR_CR_REQ		
		cr_type	CR_COMPLETE
		arfcn	ARFCN_X43
		bsic	BSIC_5
(34)	RRGRR_MS_ID_IND		
		tmsi	TMSI_0X142
(35)	MPH_IDENTITY_REQ		
		mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(36)	RR_ACTIVATE_CNF		
		op	OP_MODE_TEST_SIM
		mm_info	MM_INFO_2
		cid	CELL_IDENT_3748
		plmn	PLMN_ID_123_32
		lac	LAC_2147
		power	RF_CLASS_4
		gprs_indication	GPRS_YES

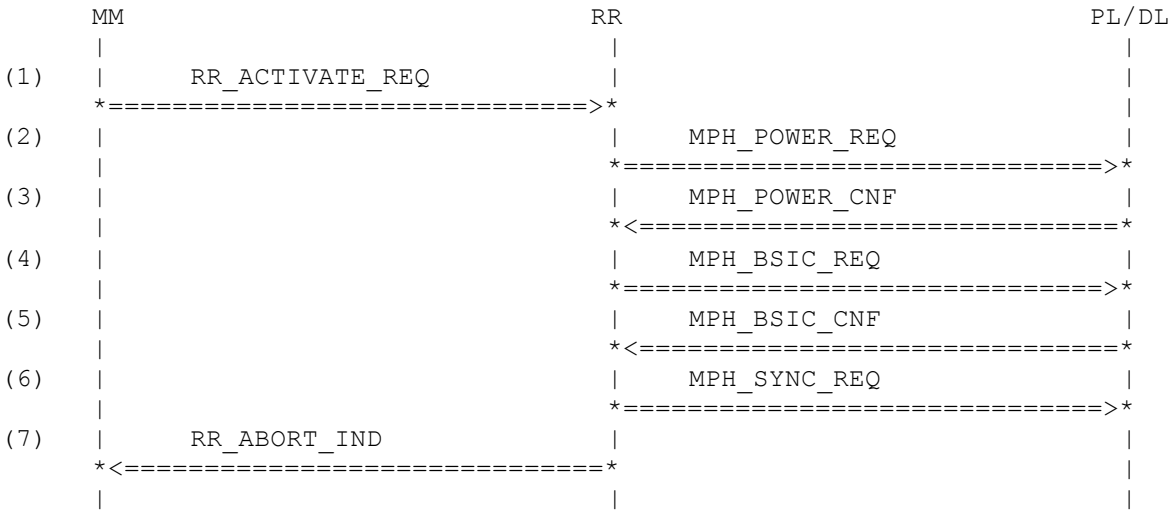
History:

22.10.02	MPA	Initial

3.3.31 RRG263: Restart by RR for full service (HPLMN) fails

Description: MM has requested a full service search for the VPLMN in the HPLMN country and RR is in full service. After timeout of the registration timer RR starts the full service search for HPLMN. The search fails and the registration timer in RR is set to the HPLMN period from the SIM card.

Preamble: RRG261



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	NOT_USED
	op	OP_MODE_NET_SRCH_MMI
	cksn	NOT_USED
	kcv	NOT_USED
	accc	NOT_USED
	imsi_struct	NOT_USED
	tmsi_struct	NOT_USED
	thplmn	NOT_USED
	bcch_info	NOT_USED
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) MPH_POWER_REQ		
	pch_interrupt	NO_PCH_INTERRUPT
	freq_bands	NOT_USED
(3) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL

(6) MPH_SYNC_REQ

cs	CS_STOP_PLMN_SEARCH

(7) RR_ABORT_IND

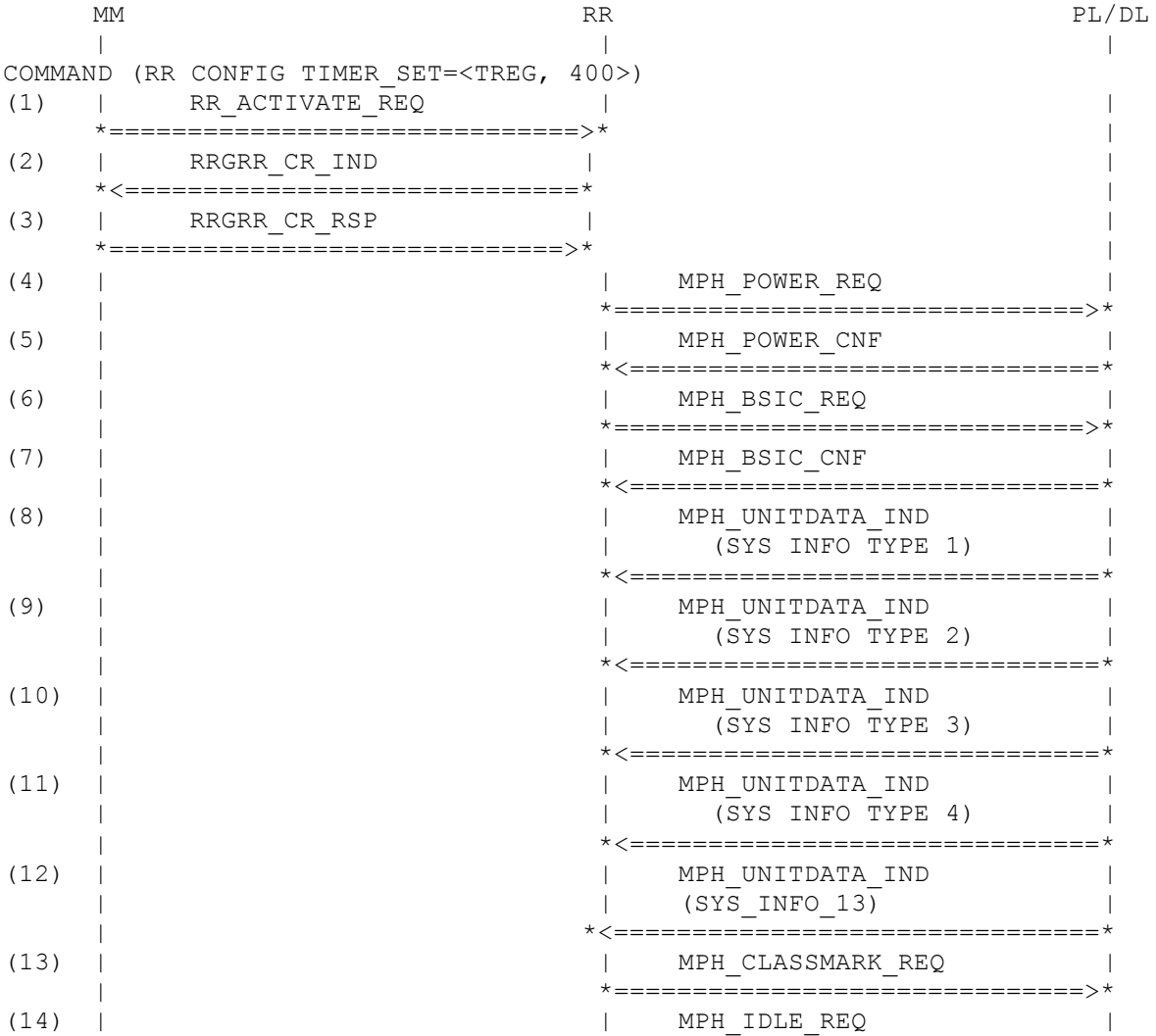
op	OP_MODE_NET_SRCH_MMI
cause	RRCS_ABORT_CEL_SEL_FAIL
plmn_avail	ONE_PLMN_AVAILABLE
plmn	ONE_PLMN
lac_list	ONE_LAC
rxlevel	ONE_RXLEV
power	RF_CLASS_4

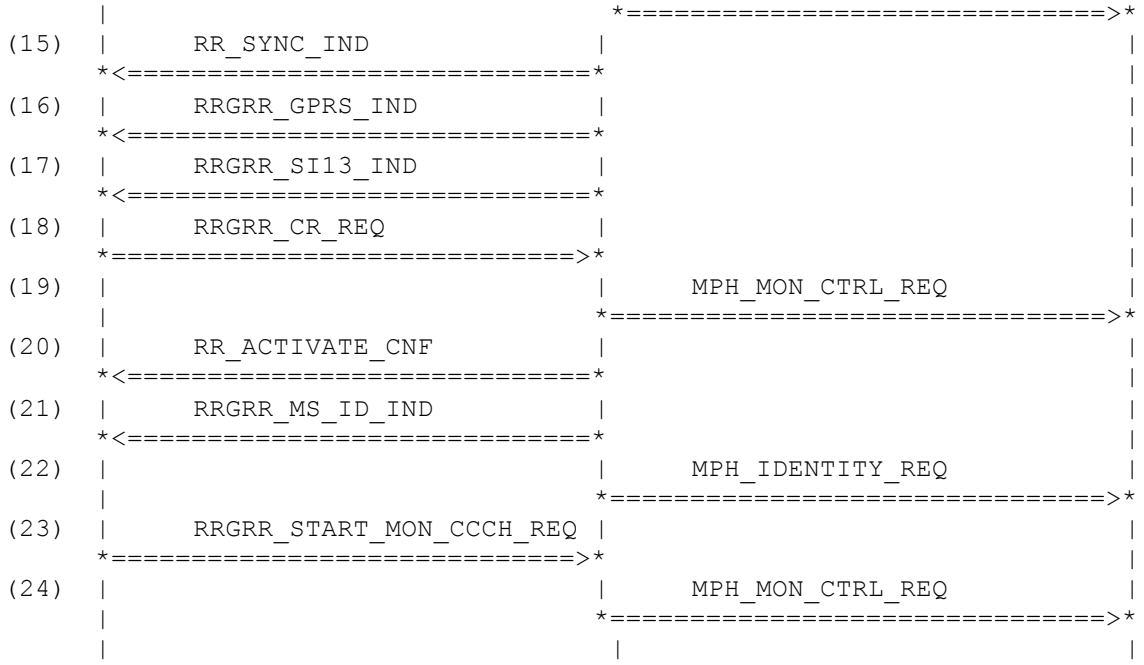
History:	22.10.02	MPA	Initial

3.3.32 RRG264: Start by MM for full service, successful, with bcch info, with PBCCH

Description: MM starts an attempt for full service for a visiting PLMN in the HPLMN country. The request contains BCCH information from the SIM card. The attempt is successful. The channel ARFCN_20 shall be used. It is member of the BCCH information. It is expected that the registration timer in RR for searching the HPLMN is started with 2 Minutes.

Preamble: RRG000





Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123_V
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	S_BCCH_INFO_20
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(3) RRGRR_CR_RSP		
(4) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	NOT_USED
(5) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(6) MPH_BSIC_REQ		
	arfcn	ARFCN_X20
(7) MPH_BSIC_CNF		
	arfcn	ARFCN_X20

	bsic	BSIC_5
	cs	CS_NO_ERROR
(8) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	}	
(9) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_1
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(10) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147_V
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(11) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0

	loc_area_ident	LOC_AREA_IDENT_123_2147_V
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(12) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(13) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(14) MPH_IDLE_REQ		
	mod	MODE_CONFIG_PL
	arfcn	ARFCN_X20
	ext_bcch	NOT_USED
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NOT_USED
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(15) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(16) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC651
(17) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC651
	arfcn	BA_LIST_651
	sdu	
	{	
	component	RR
	direction	DOWNLINK

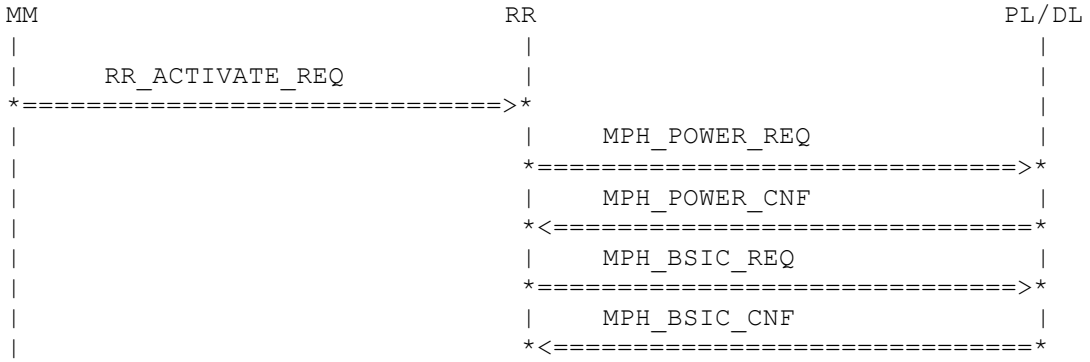
		pd	D_SYS_INFO_13
		ti	TI_0
		si13_rest_oct	SI13_REST_GPRS_TC701
		}	
(18)	RRGRR_CR_REQ		
		cr_type	CR_COMPLETE
		arfcn	ARFCN_X20
		bsic	BSIC_5
(19)	MPH_MON_CTRL_REQ		
		action	ENTER_PIM_PBCCH
		si_to_read	NOT_USED
(20)	RR_ACTIVATE_CNF		
		op	OP_MODE_TEST_SIM
		mm_info	MM_INFO_2
		cid	CELL_IDENT_3748
		plmn	PLMN_ID_123_V
		lac	LAC_2147
		power	RF_CLASS_4
		gprs_indication	GPRS_YES
(21)	RRGRR_MS_ID_IND		
		tmsi	TMSI_0X142
(22)	MPH_IDENTITY_REQ		
		mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(23)	RRGRR_START_MON_CCCH_REQ		
		pag_mode	NOT_USED
		split_pg	NOT_USED
(24)	MPH_MON_CTRL_REQ		
		action	START_MON_CCCH
		si_to_read	NOT_USED

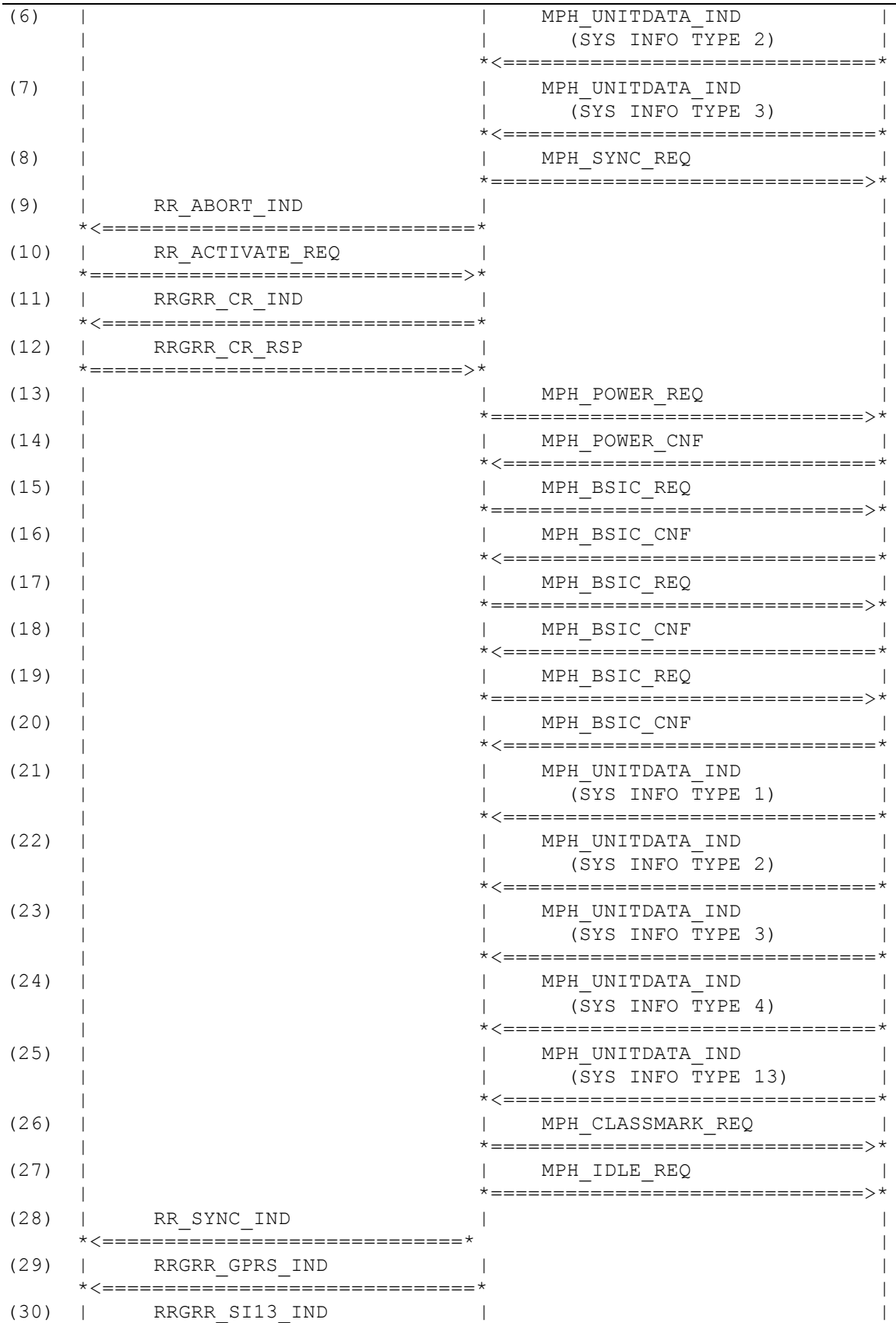
History:	22.10.02	MPA	Initial

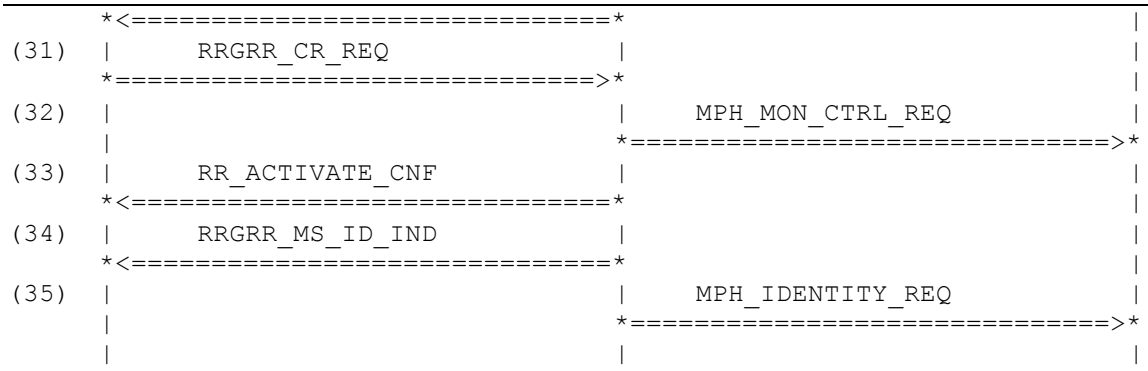
3.3.33 RRG265: Restart by RR for full service (HPLMN) is successful, PBCCH

Description: MM has requested a full service search for the VPLMN in the HPLMN country and RR is in full service. After timeout of the registration timer RR starts the full service search for HPLMN. The search is successful and MM is informed.

Preamble: RRG264




(6) MPH_UNITDATA_IND (SYS INFO TYPE 2)
(7) MPH_UNITDATA_IND (SYS INFO TYPE 3)
(8) MPH_SYNC_REQ
(9) RR_ABORT_IND
(10) RR_ACTIVATE_REQ
(11) RRGRR_CR_IND
(12) RRGRR_CR_RSP
(13) MPH_POWER_REQ
(14) MPH_POWER_CNF
(15) MPH_BSIC_REQ
(16) MPH_BSIC_CNF
(17) MPH_BSIC_REQ
(18) MPH_BSIC_CNF
(19) MPH_BSIC_REQ
(20) MPH_BSIC_CNF
(21) MPH_UNITDATA_IND (SYS INFO TYPE 1)
(22) MPH_UNITDATA_IND (SYS INFO TYPE 2)
(23) MPH_UNITDATA_IND (SYS INFO TYPE 3)
(24) MPH_UNITDATA_IND (SYS INFO TYPE 4)
(25) MPH_UNITDATA_IND (SYS INFO TYPE 13)
(26) MPH_CLASSMARK_REQ
(27) MPH_IDLE_REQ
(28) RR_SYNC_IND
(29) RRGRR_GPRS_IND
(30) RRGRR_SI13_IND



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	NOT_USED
	op	OP_MODE_NET_SRCH_MMI
	cksn	NOT_USED
	kcv	NOT_USED
	accc	NOT_USED
	imsi_struct	NOT_USED
	tmsi_struct	NOT_USED
	thplmn	NOT_USED
	bcch_info	NOT_USED
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) MPH_POWER_REQ		
	pch_interrupt	NO_PCH_INTERRUPT
	freq_bands	NOT_USED
(3) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_1
	ncc_permit	NCC_PERMITTED_1

	rach_ctrl	RACH_CTRL_1
	}	
(7) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_32_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(8) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(9) RR_ABORT_IND		
	op	OP_MODE_NET_SRCH_MMI
	cause	RRCS_ABORT_CEL_SEL_FAIL
	plmn_avail	TWO_PLMN_AVAILABLE
	plmn	TWO_PLMNS
	lac_list	TWO_LACS
	rxlevel	TWO_RXLEVS
	power	RF_CLASS_4
(10) RR_ACTIVATE_REQ		
	plmn	PLMN_ID_123_32
	op	OP_MODE_TEST_SIM
	cksn	CKSN_NOT_PRES
	kcv	KCV_12345678
	accc	ACC_CTRL_CLASS_0008
	imsi_struct	MOBILE_ID_IMSI_HPLMN
	tmsi_struct	MOBILE_ID_TMSI
	thplmn	TIME_HPLMN_VALID
	bcch_info	NOT_USED
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(11) RRGRR_CR_IND		
	cr_type	CR_NORMAL
(12) RRGRR_CR_RSP		
(13) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	NOT_USED
(14) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20

(15) MPH_BSIC_REQ		
	arfcn	ARFCN_X20
(16) MPH_BSIC_CNF		
	arfcn	ARFCN_X20
	bsic	BSIC_5
	cs	CS_NO_BCCH_AVAIL
(17) MPH_BSIC_REQ		
	arfcn	ARFCN_124
(18) MPH_BSIC_CNF		
	arfcn	ARFCN_124
	bsic	BSIC_5
	cs	CS_NO_BCCH_AVAIL
(19) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(20) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(21) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	}	
(22) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_1
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(23) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3

	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_32_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(24) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_32_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(25) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(26) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(27) MPH_IDLE_REQ		
	mod	MODE_CONFIG_PL
	arfcn	ARFCN_X43
	ext_bcch	NOT_USED
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NOT_USED
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED

(28) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(29) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702_32
(30) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702_32
	arfcn	BA_LIST_651
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(31) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X43
	bsic	BSIC_5
(32) MPH_MON_CTRL_REQ		
	action	ENTER_PIM_PBCCH
	si_to_read	NOT_USED
(33) RR_ACTIVATE_CNF		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_2
	cid	CELL_IDENT_3748
	plmn	PLMN_ID_123_32
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES
(34) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(35) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702

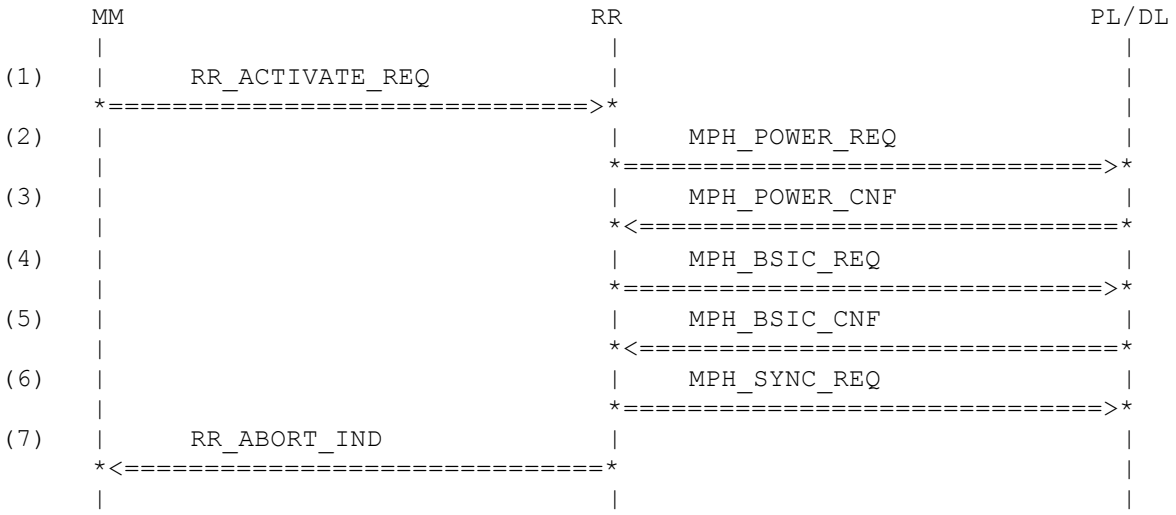
History:

22.10.02	MPA	Initial

3.3.34 RRG266: Restart by RR for full service (HPLMN) fails

Description: MM has requested a full service search for the VPLMN in the HPLMN country and RR is in full service. After timeout of the registration timer RR starts the full service search for HPLMN. The search fails and the registration timer in RR is set to the HPLMN period from the SIM card.

Preamble: RRG264



Parametrization

Primitive	Parameter	Value
(1) RR_ACTIVATE_REQ		
	plmn	NOT_USED
	op	OP_MODE_NET_SRCH_MMI
	cksn	NOT_USED
	kcv	NOT_USED
	accc	NOT_USED
	imsi_struct	NOT_USED
	tmsi_struct	NOT_USED
	thplmn	NOT_USED
	bcch_info	NOT_USED
	cell_test	CELL_TEST_DISABLE
	gprs_indication	GPRS_YES
(2) MPH_POWER_REQ		
	pch_interrupt	NO_PCH_INTERRUPT
	freq_bands	NOT_USED
(3) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(4) MPH_BSIC_REQ		
	arfcn	ARFCN_X43
(5) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_0
	cs	CS_NO_BCCH_AVAIL

(6) MPH_SYNC_REQ		
	cs	CS_STOP_PLMN_SEARCH
(7) RR_ABORT_IND		
	op	OP_MODE_NET_SRCH_MMI
	cause	RRCS_ABORT_CEL_SEL_FAIL
	plmn_avail	ONE_PLMN_AVAILABLE
	plmn	ONE_PLMN
	lac_list	ONE_LAC
	rxlevel	ONE_RXLEV
	power	RF_CLASS_4

History:	22.10.02	MPA	Initial

3.4 Packet Paging (300)

3.4.1 RRG301: RRGRR_PACKET_PAGING_IND

Description:

The MS is in the idle mode camping on a cell. The MS is paged by a packet paging message. This message may contain a paging request type 1,2 or 3. Further the message indicates whether the paging type was IMSI or P-TMSI (GPRS TMSI).
Pagings will only be accepted if there was at least one successful measurement.
The immediate assignment procedure is started.

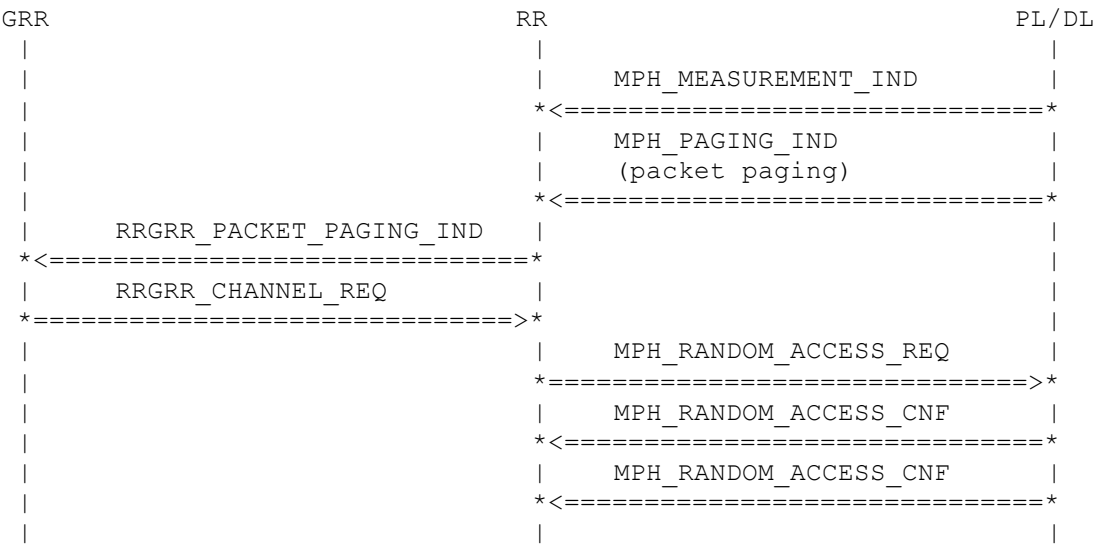
A Paged by PTMSI
B Paged by TMSI
C Paged by IMSI

Preamble:

RRG102

Variants:

<A>..<C>



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_35
	rx_lev_sub	RX_LEV_35
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC

	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_PAGING_IND		
<A>	identity_type	ID_PTMSI
<B>	identity_type	ID_TMSI
<C>	identity_type	ID_IMSI
	channel_needed	CN_PACKET
(3) RRGRR_PACKET_PAGING_IND		
<A>	pg_type	RRGRR_PTMSI
<B>	pg_type	NOT_USED
<C>	pg_type	RRGRR_IMSI
(4) RRGRR_CHANNEL_REQ		
	req_data	ESTCS_PAGING
(5) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC721
(6) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(7) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1

History:

21-July-00	MSE	Initial
01-Nov-00	MSE	Measurement added

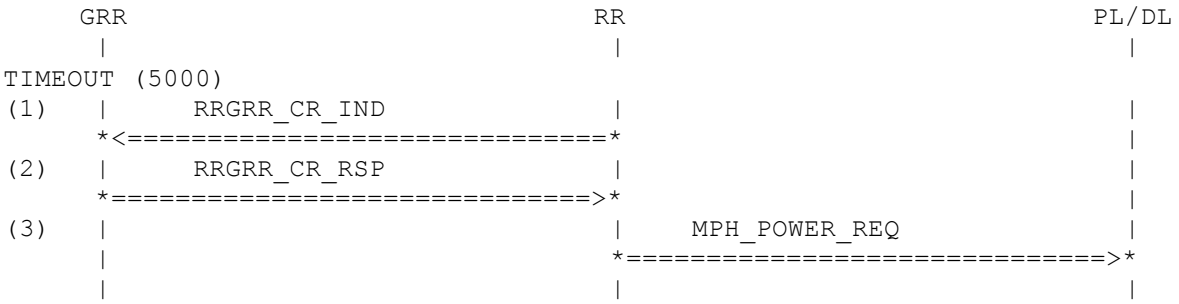
3.4.2 RRG302: Channel Request after Paging – IA failed

Description:

The MS requests a channel assignment after being paged.
The immediate assignment can not be finished because there is no channel assignment from network. After timeout a RRGRR_CR_IND is sent to GRR and a cell selection is performed (because an cell reselection fails).

Preamble:

RRG301A



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(2) RRGRR_CR_RSP		
(3) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A

History:

21-Nov-2000	MSE	Initial

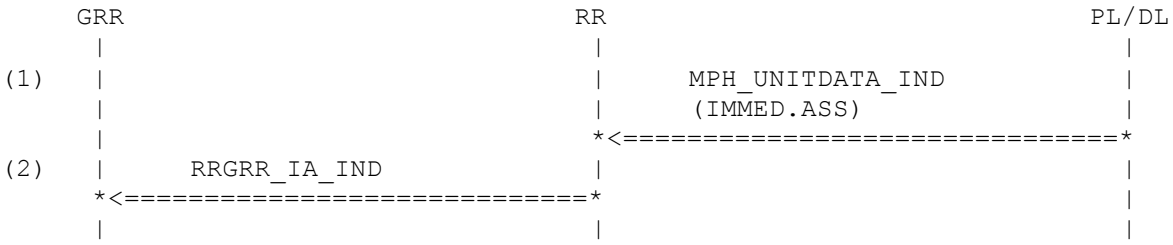
3.4.3 RRG304: Channel Request after Paging – IA success

Description:

The MS has requested a channel after being paged.
The BS assigns a channel for a TBF.
NOTE: RR will send an MPH_IDLE_REQ after the RRGRR_IA_IND

Preamble:

RRG301A



Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4A
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	NOT_USED
	}	
(2) RRGRR_IA_IND		
	fn	fn_780
	r_bit	CHAN_REQ_SENT_MORE
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0

d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4A
time_advance	TIMING_ADVANCE_30
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	NOT_USED
}	

History:

10-Jan-2001	MSE	Initial

3.5 Packet Access (400)

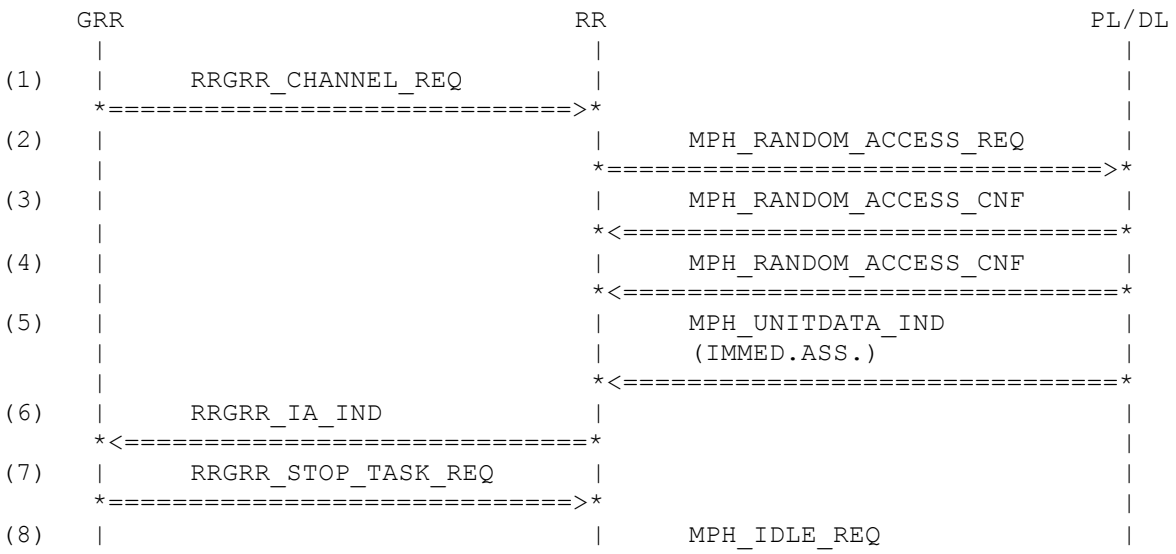
3.5.1 RRG401: Immediate assignment for PDCH / success

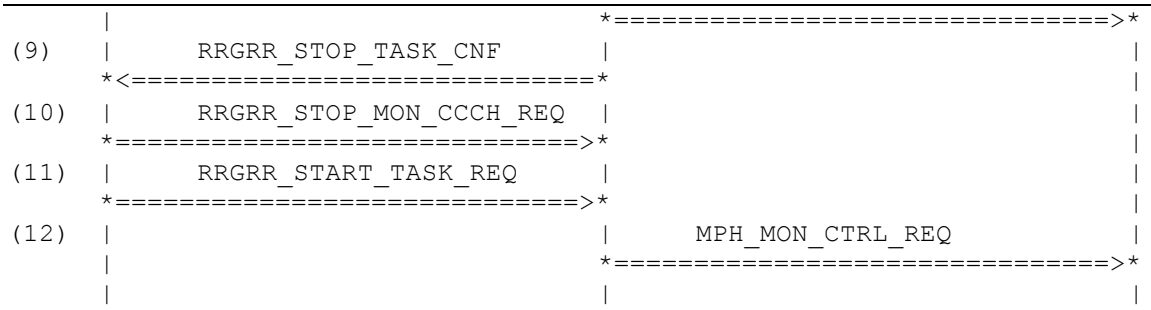
Description:

RR is in the idle mode. GRR requests the establishment of an PDCH using the CCCH.
(RR 1): RR receives the channel request from GRR. RR starts the access procedure on the CCCH. RR does not need to check if there is a call active(GMM/GRR know this) or if the access is allowed(also checked by GRR). The wait indication(T3122) needs not to be checked for packet access and T3142 is checked by GRR.
(RR 2): T3146 is started after sending all random bursts. T3146 uses T3126(the GSM equivalent) as they are the same.
(RR 3): On receipt of an Immediate Assignment message RR checks if the Message is either a TBF or dedicated mode assignment (If the Immediate Assignment message was part of a TMA(two message assignment) is by from RR, as PL handles the TMA and sends only the relevant part to RR). RR checks if the IA is valid by comparing the random access reference. For a valid TBF assignment RR informs GRR via the RRGRR_IA_IND (there is also only 1 RRGRR_IA_IND for a two message assignment with all the relevant parts merged). For a dedicated mode assignment RR sends a RRGRR_IA_IND if GRR requested a channel else the normal RR procedures are used. T3146 is stopped on receipt of a valid IA.

Preamble:

RRG102





Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	
	IA_REST_OCTETS_UL_ASSIGN_TMA1	
	}	
(6) RRGRR_IA_IND		
	fn	fn_780
	r_bit	CHAN_REQ_SENT_MORE
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN

	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	
	IA_REST_OCTETS_UL_ASSIGN_TMA1	
	}	
(7) RRGRR_STOP_TASK_REQ		
	v_stop_ccch	STOP_CCCH
(8) MPH_IDLE_REQ		
	mod	MODE_PACKET_TRANSFER
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT
(9) RRGRR_STOP_TASK_CNF		
(10) RRGRR_STOP_MON_CCCH_REQ		
	is_pbcch_present	FALSE
(11) RRGRR_START_TASK_REQ		
	state	TASK_STATE_PTM
(12) MPH_MON_CTRL_REQ		
	action	ENTER_PTM_BCCH
	si_to_read	NOT_USED

History:

21-July-00	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only
14-Apr-03	MPA	add START_TASK

3.5.2 RRG402: Immediate assignment for SDCCH / success

Description:

RR is in the idle mode. GRR requests the establishment of an PDCH using the CCCH. The network assigns an SDCCH.

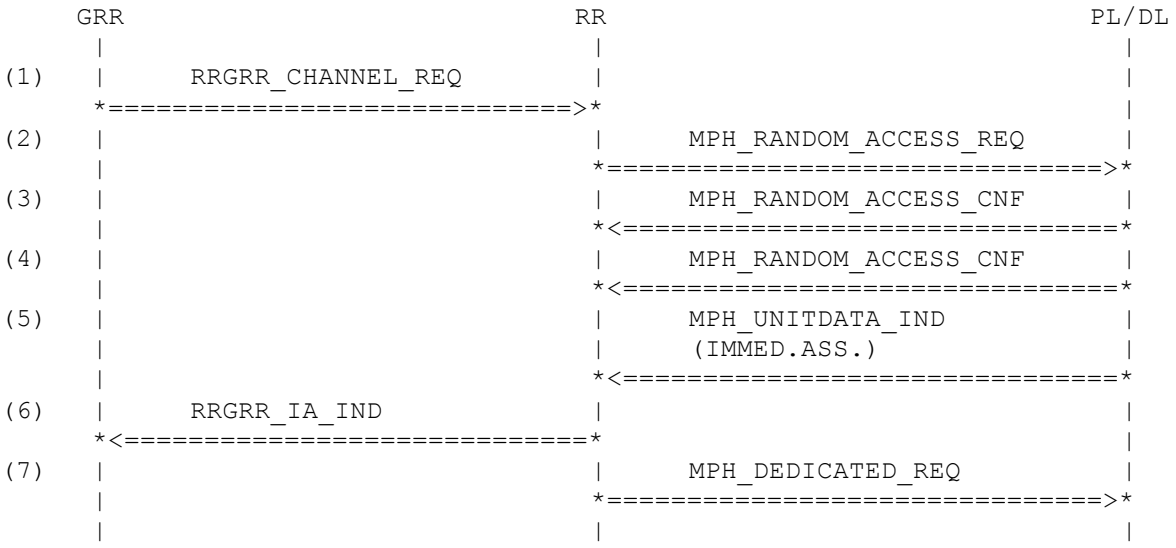
(RR 1): RR receives the channel request from GRR. RR starts the access procedure on the CCCH. RR doesnot need to check if there is a call active(GMM/GRR know this) or if the access is allowed(also checked by GRR). The wait indication(T3122) needs not to be checked for packet access and T3142 is checked by GRR.

(RR 2): T3146 is started after sending all random bursts. T3146 uses T3126 (the GSM equivalent) as they are the same.

(RR 3): On receipt of an Immediate Assignment message RR checks if the Message is either a TBF or dedicated mode assignment (If the Immediate Assignment message was part of a TMA(two message assignment) is by from RR, as PL handles the TMA and sends only the relevant part to RR). RR checks if the IA is valid by comparing the random access reference.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1

(5) MPH_UNITDATA_IND

arfcn	ARFCN_X43
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_0
dl	DL_0
d_t	D_T_DED
page_mode	PAGING_NORMAL
chan_desc	CHANNEL_DESC_SDCCH
pck_chan_desc	NOT_USED
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	IA_REST_OCTETS_TC730
}	

(6) RRGRR_IA_IND

fn	fn_780
r_bit	CHAN_REQ_SENT_MORE
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_0
dl	DL_0
d_t	D_T_DED
page_mode	PAGING_NORMAL
chan_desc	CHANNEL_DESC_SDCCH
pck_chan_desc	NOT_USED
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	IA_REST_OCTETS_TC730
}	

(7) MPH_DEDICATED_REQ

mod	MODE_IMM_ASSIGN
start	STARTING_TIME_1
ch_type	PRR_CHANNEL_TYPE_2
ch_type2	DISPLAY_ONLY
arfcn	ARFCN_X43
bsic	BSIC_0
ho_param	NOT_USED
tr_para	PRR_TR_PARA_2
ciph	NO_CIPHERING
amr_conf	NOT_USED

History:

	21-July-00	MSE	Initial

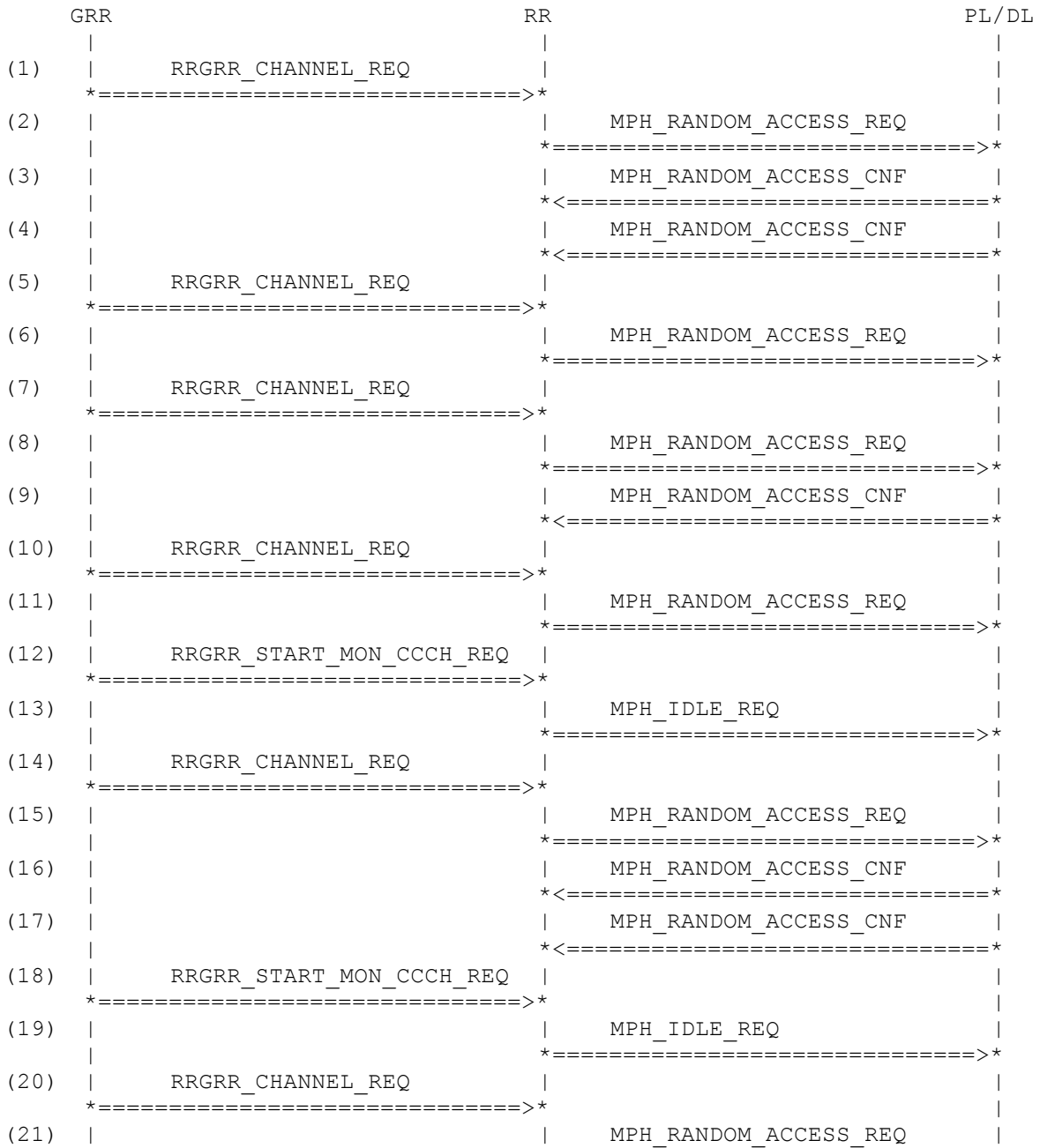
3.5.3 RRG403: Restart of packet access procedure

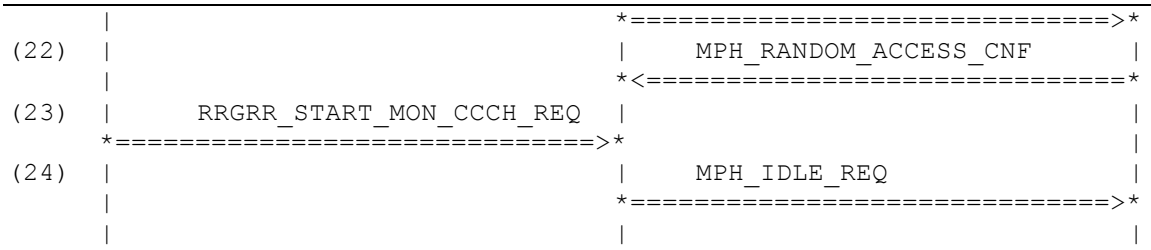
Description:

GRR establishes a TBF. After sending the RACH bursts, RR waits for an Immediate Assignment message but does not receive it.

Preamble:

RRG102





Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_2nd
(6) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC403
(7) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(8) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(9) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(10) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_2nd
(11) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC403
(12) RRGRR_START_MON_CCCH_REQ		
	pag_mode	PAG_MODE_REORG
	split_pg	NOT_USED
(13) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02

	ncc_permitted	NCC_PERMITTED_1
	reorg_only	REORG_ONLY
	eotd_avail	NOT_USED
	gprs_support	MPH_GPRS_PROCS_USED
(14) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(15) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(16) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(17) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(18) RRGRR_START_MON_CCCH_REQ		
	pag_mode	PAG_MODE_DEFAULT
	split_pg	NOT_USED
(19) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	NOT_USED
	gprs_support	MPH_GPRS_PROCS_USED
(20) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_2nd
(21) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC403
(22) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(23) RRGRR_START_MON_CCCH_REQ		
	pag_mode	PAG_MODE_REORG
	split_pg	NOT_USED
(24) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2

	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	REORG_ONLY
	eotd_avail	NOT_USED
	gprs_support	MPH_GPRS_PROCS_USED

History:

26-Mar-02	MPA	Initial

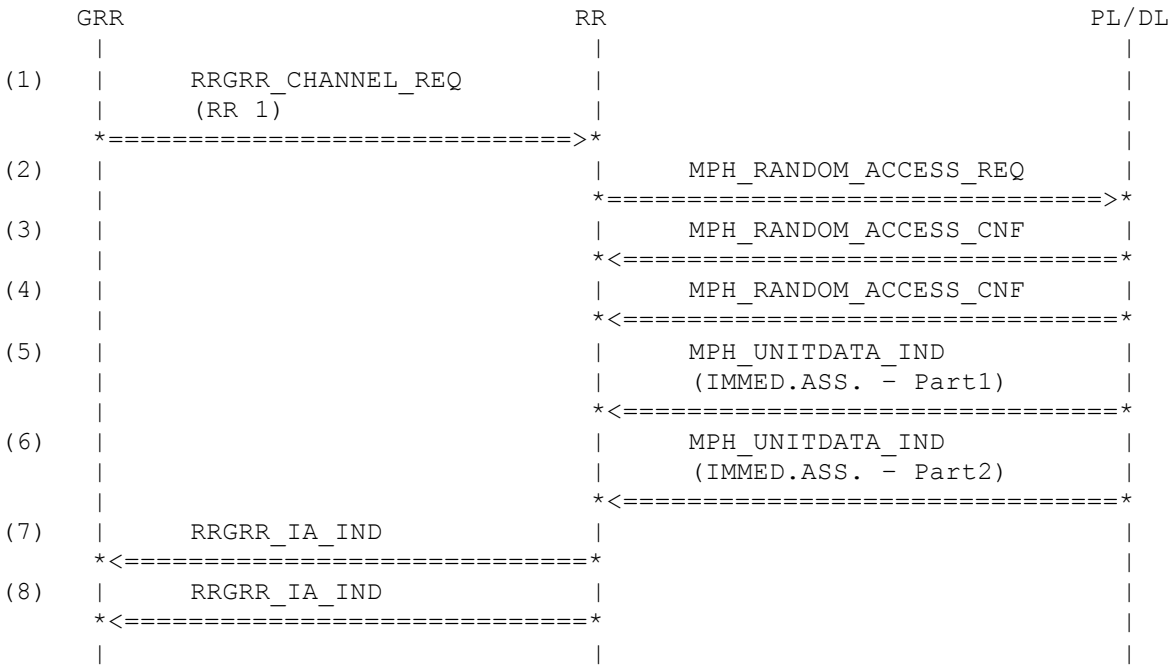
3.5.4 RRG405: Two-message assignment

Description:

RR is in the idle mode. GRR requests the establishment of a PDCH. The network assigns it by two IA messages, which is noticeable in the flag TMA.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1

(5) MPH_UNITDATA_IND

arfcn	ARFCN_X43
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_1
dl	DL_0
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	
IA_REST_OCTETS_UL_ASSIGN_TMA1	
}	

(6) MPH_UNITDATA_IND

arfcn	ARFCN_X43
fn	fn_790
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_0
dl	DL_0
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	
IA_REST_OCTETS_UL_ASSIGN_TMA2	
}	

(7) RRGRR_IA_IND

fn	FLAG_ZERO
r_bit	FLAG_ZERO
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_1

dl	DL_0
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	
IA_REST_OCTETS_UL_ASSIGN_TMA1	
}	

(8) RRGRR_IA_IND

fn	fn_790
r_bit	CHAN_REQ_SENT_MORE
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_0
dl	DL_0
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	
IA_REST_OCTETS_UL_ASSIGN_TMA2	
}	

History:

12-Oct-00	MSE	Initial

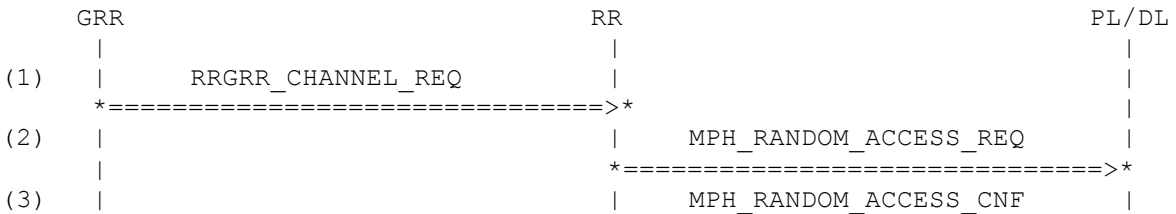
3.5.5 RRG406: Immediate Assignment not received

Description:

GRR establishes a TBF. After sending the RACH bursts, RR waits for an Immediate Assignment message but does not receive it.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF	frame_no	FRAME_NUMBER_1
(5) RRGRR_CR_IND	cr_type	CR_ABNORMAL
(6) RRGRR_CR_RSP		
(7) MPH_POWER_REQ	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A

History:

21-July-00	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only

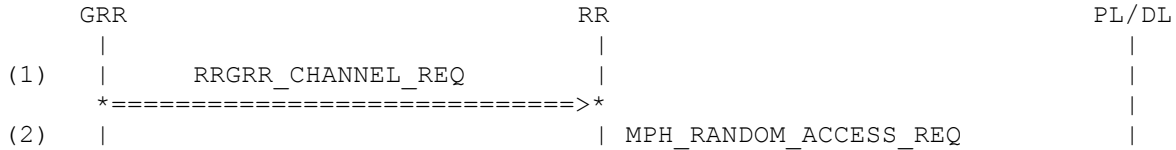
3.5.6 RRG407: Wrong immediate assigment received

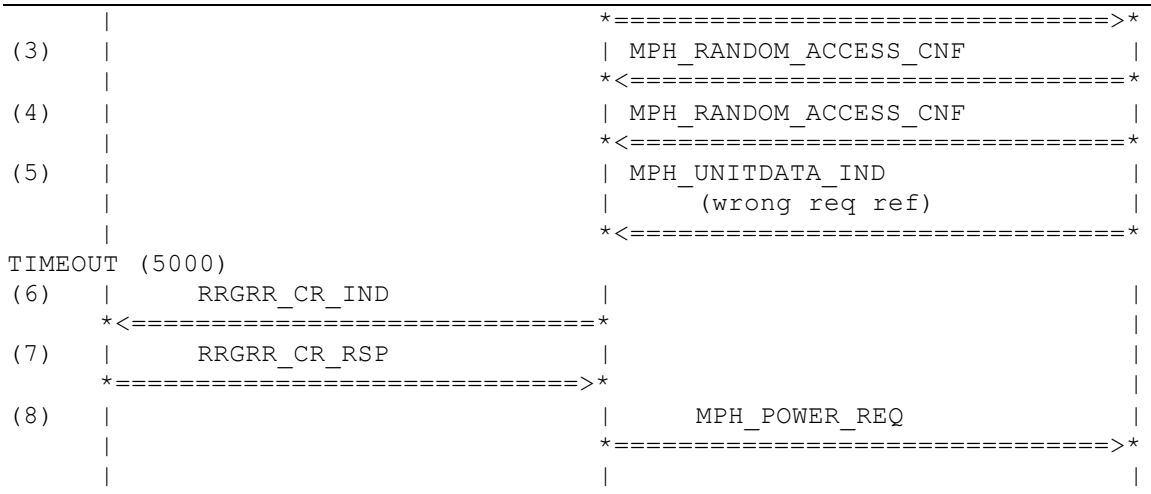
Description:

RR is in the idle mode. GRR requests the establishment of a PDCH.
The Immediate Assignment procedure fails because only immediate assignment messages are received which have a wrong request reference. Such messages are ignored. After expiry of T3126 GRR is informed, that the immediate assignment procedure has failed. GRR requests a cell reselection. Because there are no suitable neighbour cells a cell selection is started.

Preamble:

RRG102





Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_3
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_TC730
	}	
(6) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(7) RRGRR_CR_RSP		

Confidential

(8) MPH_POWER_REQ

pch_interrupt	PCH_INTERRUPT
freq_bands	STD_DUAL_EXT_0A

History:

11-Oct-00	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only

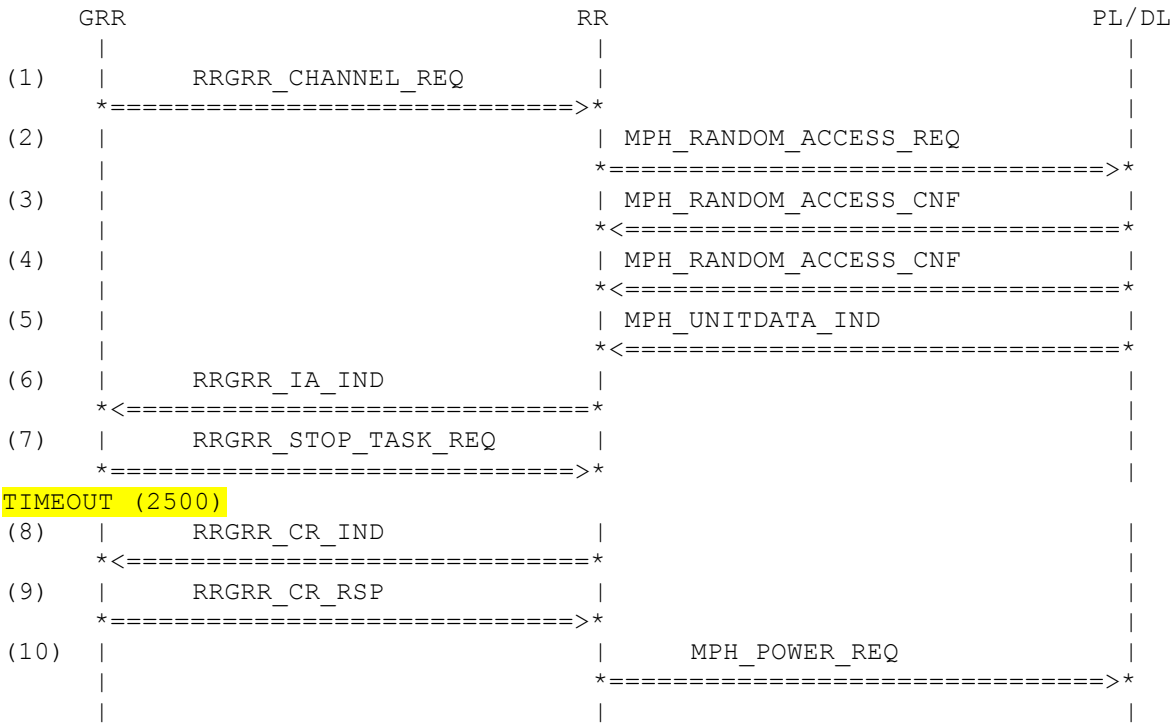
3.5.7 RRG408: Immediate assignment is incomplete

Description:

RR is in idle mode. GRR request the establishment od an PDCH. An IA message is passed to GRR which rejects the message as incomplete.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK

	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	NOT_USED
	}	
(6) RRGRR_IA_IND		
	fn	fn_780
	r_bit	CHAN_REQ_SENT_MORE
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	NOT_USED
	}	
(7) RRGRR_STOP_TASK_REQ		
	v_stop_ccch	INVALID_MSG
(9) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(10) RRGRR_CR_RSP		
(11) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A

History:

11-Oct-00	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only

3.5.8 RRG409: Immediate assignment – RLC/MAC failure

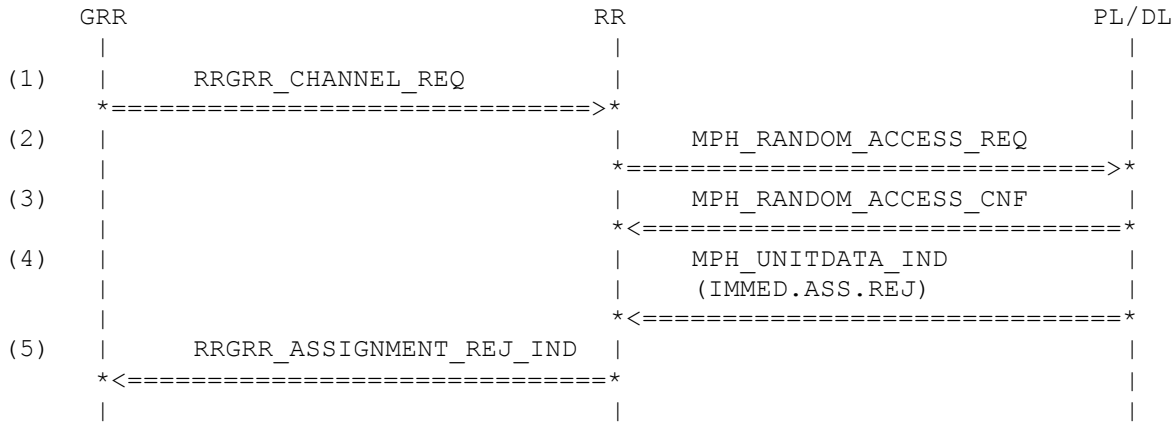
3.5.9 RRG410: Immediate assignment rejection

Description:

This primitive indicates to the GRR that the ASSIGNMENT REJECT was received. incomplete!!!!

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN_REJ
	ti	TI_0
	page_mode	PAGING_NORMAL
	req_ref	REQUEST_REFERENCE_1
	t3122	WAIT_INDICATION_0
	req_ref_2	REQUEST_REFERENCE_2
	t3122_2	WAIT_INDICATION_0
	req_ref_3	REQUEST_REFERENCE_3
	t3122_3	WAIT_INDICATION_0
	req_ref_4	REQUEST_REFERENCE_4

	t3122_4 }	WAIT_INDICATION_0
(5) RRGRR_ASSIGNMENT_REJ_IND		
	wait_ind r_bit	WAIT_INDICATION_0 CHAN_REQ_SENT_ONCE

History:

21-July-00	MSE	Initial

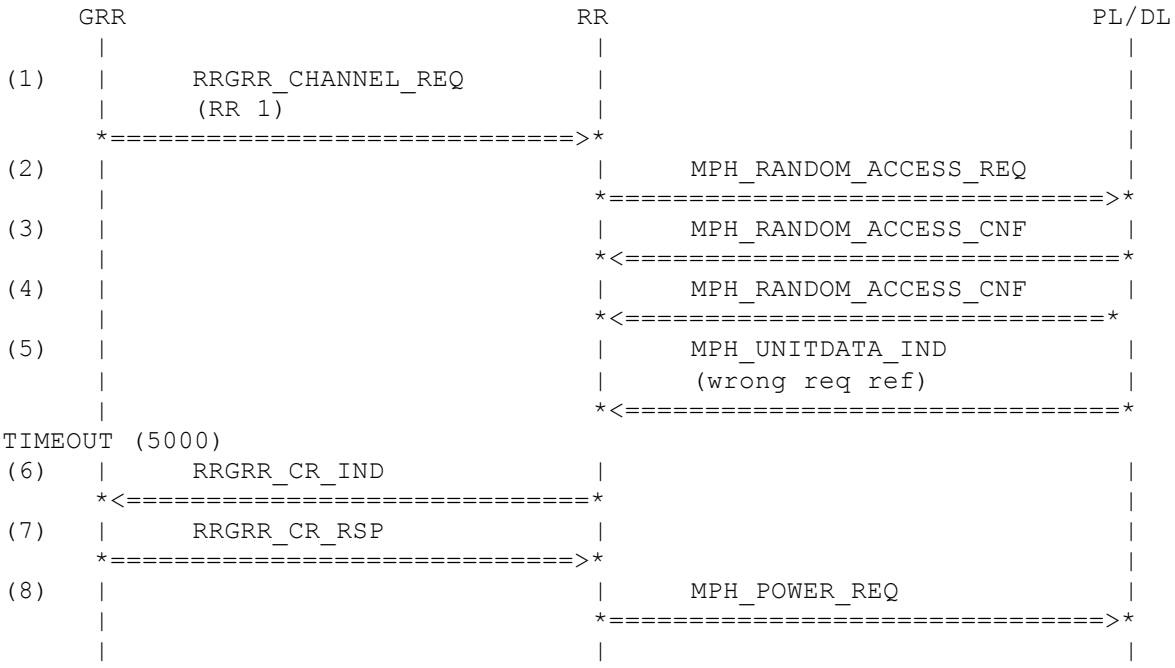
3.5.10 RRG411: Immediate assignment for SDCCH fails

Description:

This primitive indicates to the GRR that RR has stopped the channel access procedure due to expiry of T3146. It is only a trigger.
The Immediate Assignment Message is invalid becouse of a wrong request reference.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN

	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_DED
	page_mode	PAGING_NORMAL
	chan_desc	CHANNEL_DESC_SDCCH
	pck_chan_desc	NOT_USED
	req_ref	REQUEST_REFERENCE_FAIL
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_TC730
	}	
(6) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(7) RRGRR_CR_RSP		
(8) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A

History:

17-Oct-00	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only

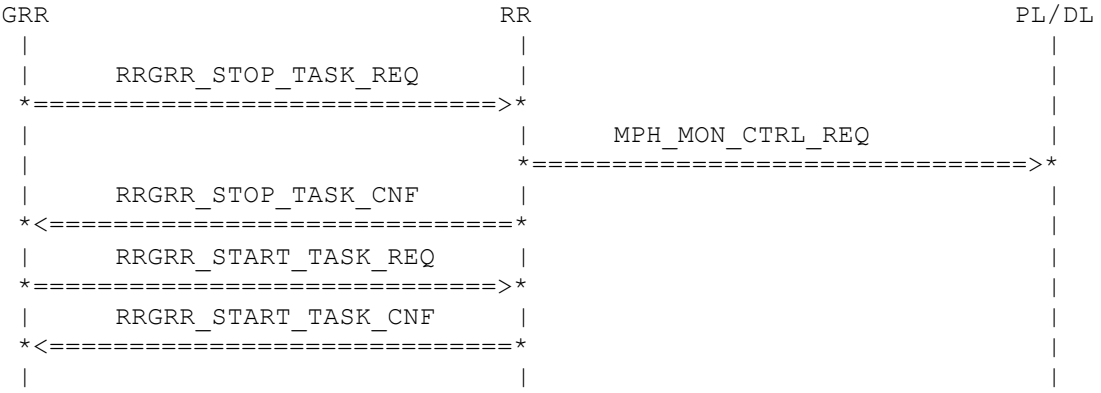
3.5.11 RRG412: Packet Access with PBCCH

Description:

Packet access with PBCCH.

Preamble:

RRG249



Parametrization

Primitive	Parameter	Value
(1) RRGRR_STOP_TASK_REQ		
	v_pim_pbcch	LEAVE_PIM_PBCCH
(2) MPH_MON_CTRL_REQ		
	action	LEAVING_PIM_PBCCH
	si_to_read	NOT_USED
(3) RRGRR_STOP_TASK_CNF		
(4) RRGRR_START_TASK_REQ		
	state	TASK_STATE_PAM
(5) RRGRR_START_TASK_CNF		

History:

25-Nov-02	MPA	Initial

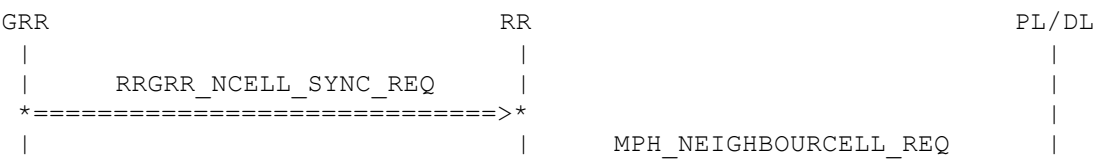
3.5.12 RRG413: new ncell list in PAM, PBCCH

Description:

.

Preamble:

RRG412



```
        |                                   *==============================>*
        |                                   |                                |
```

Parametrization

Primitive	Parameter	Value
(1) RRGRR_NCELL_SYNC_REQ		
	arfcn	ARFCN_LIST_PBCCH_124_6_11
(2) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	ARFCN_LIST_PBCCH_124_6_11
	sync_only	SYNC_LIST

History:

25-Nov-02	MPA	Initial

3.5.13 RRG414: Left Blank 2

Description:

Dummy

Preamble:

None

Parametrization

Primitive	Parameter	Value

History:

16-Jul-01	MSE	Initial

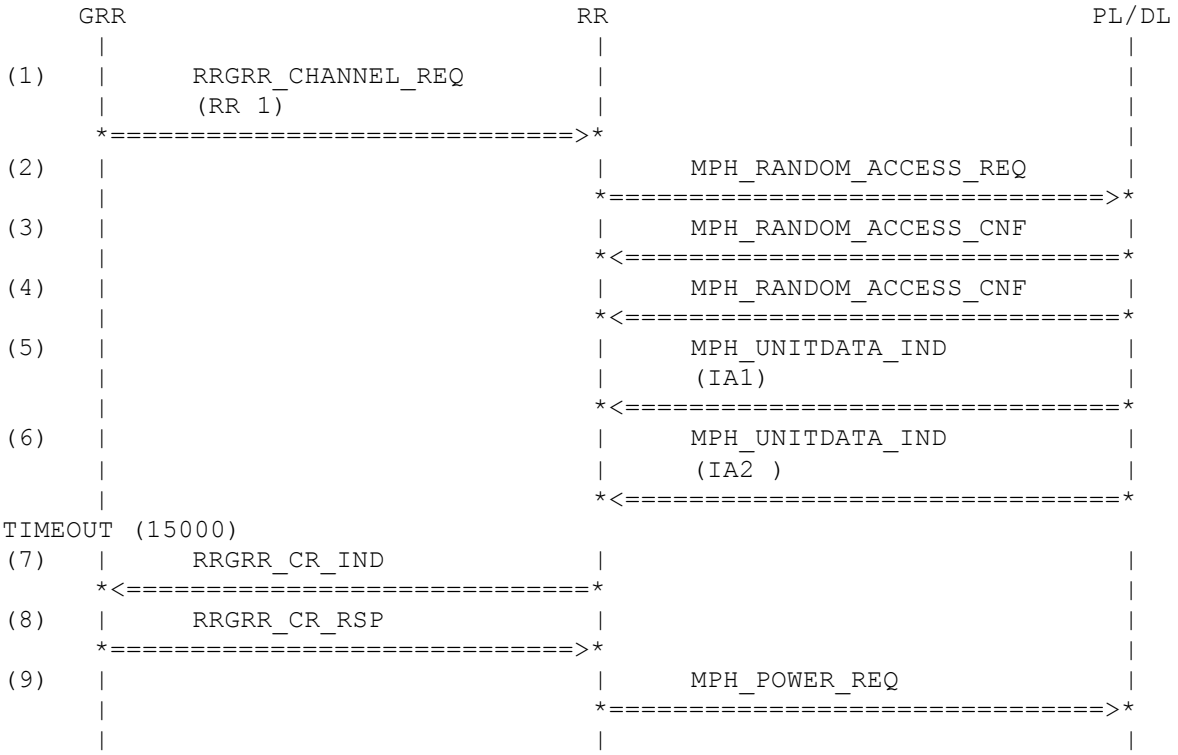
3.5.14 RRG415: Two-message assignment fails

Description:

This primitive indicates to the GRR that RR has stopped the channel access procedure due to expiry of T3146. It is only a trigger.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR

	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_1
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_TC730
	}	
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_890
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_FAIL
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_TC733
	}	
(7) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(8) RRGRR_CR_RSP		
(9) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A

History:

11-Oct-00	MSE	Initial
16-Jul-01	MSE	IDLE_REQ: +reorg_only

3.6 Packet Downlink Assignment Procedure (500)

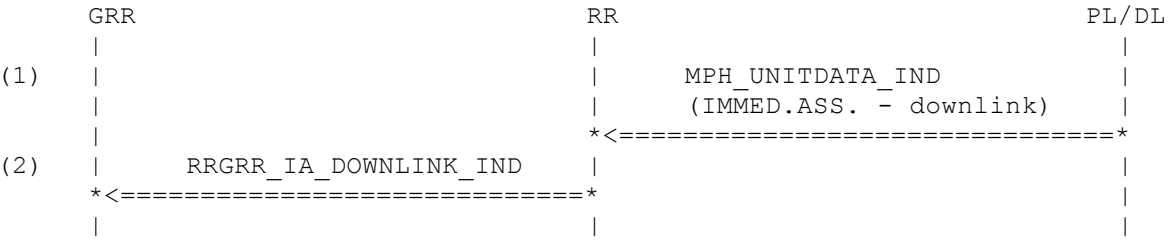
3.6.1 RRG501: Downlink assignment

Description:

An IA for downlink assignment is received and forwarded to GRR.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_1
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_DL_ASSIGN
	}	
(2) RRGRR_IA_DOWNLINK_IND		
	fn	fn_780
	r_bit	NOT_PRESENT_8BIT
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0

dl	DL_1
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	IA_REST_OCTETS_DL_ASSIGN
}	

History:

12-Oct-00	MSE	Initial

3.6.2 RRG502: Two-message downlink assignment

Description:

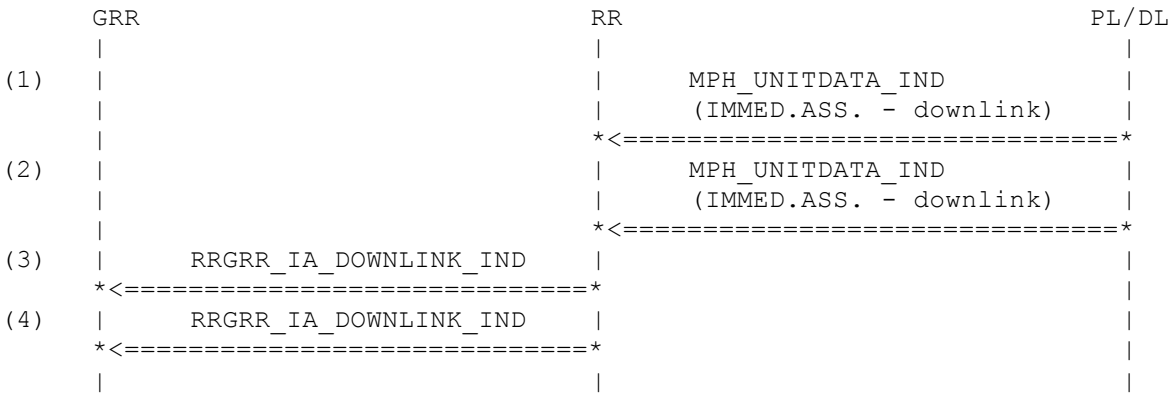
A two message IA for downlink assignment is received and forwarded after receiption of second message to GRR.

Preamble:

RRG102

Variants:

<A>....<B>



Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_1
	dl	DL_1
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_DL_ASSIGN
	}	
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_790
	sdu	
	{	

component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_1
dl	DL_1
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	NOT_USED
start_time	NOT_USED
ia_rest_oct	
IA_REST_OCTETS_DL_ASSIGN_COPY	
}	

(3) RRGRR_IA_DOWNLINK_IND

fn	fn_780
r_bit	NOT_PRESENT_8BIT
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_1
dl	DL_1
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4
time_advance	TIMING_ADVANCE_27
mob_alloc	MOBILE_ALLOCATION_1
start_time	START_TIME_1
ia_rest_oct	IA_REST_OCTETS_DL_ASSIGN
}	

(4) RRGRR_IA_DOWNLINK_IND

fn	fn_790
r_bit	NOT_PRESENT_8BIT
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN
ti	TI_0
tma	TMA_1
dl	DL_1
d_t	D_T_TBF
page_mode	PAGING_NORMAL
chan_desc	NOT_USED
pck_chan_desc	PACKET_CHANNEL_DESC
req_ref	REQUEST_REFERENCE_4

time_advance	TIMING_ADVANCE_27
mob_alloc	NOT_USED
start_time	NOT_USED
ia_rest_oct	IA_REST_OCTETS_DL_ASSIGN_COPY
}	

History:

12-Oct-00	MSE	Initial

3.7 Dedicated Mode Procedures related to GPRS (600)

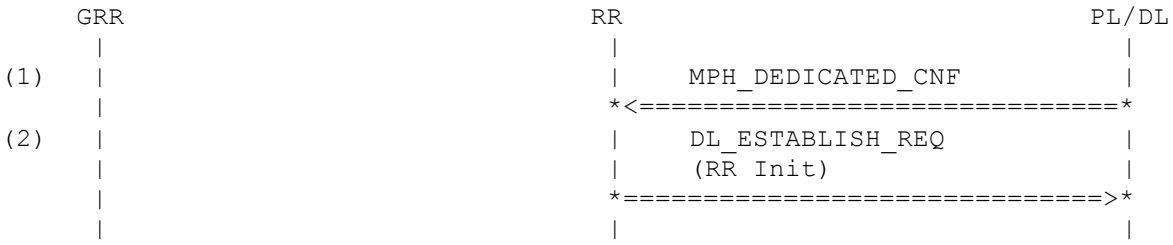
3.7.1 RRG601: DL_ESTABLISH_REQ (RR_Init)

Description:

The DL_ESTABLISH_REQ is used to initiate a mobile originated multiple frame acknowledged operation. It contains an SDU with an RR Initialisation Request to request the establishment of dedicated mode.

Preamble:

RRG402



Parametrization

Primitive	Parameter	Value
(1) MPH_DEDICATED_CNF		
	dedi_res	DEDI_RES_OK
(2) DL_ESTABLISH_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	D_RR_INIT_REQ
	ti	TI_0
	ciph_key_num	CKSN_RESERVED
	chan_coding	CHAN_CODING_TC770
	mob_class_2	MOB_CLASS2
	ded_tlli	DED_TLLI_BITBUFFER_TC770
	chan_req_desc	CHANNEL_REQ_TC770
	gprs_meas_res	MEASURES_TC770
	}	

History:

21-July-00	MSE	Initial

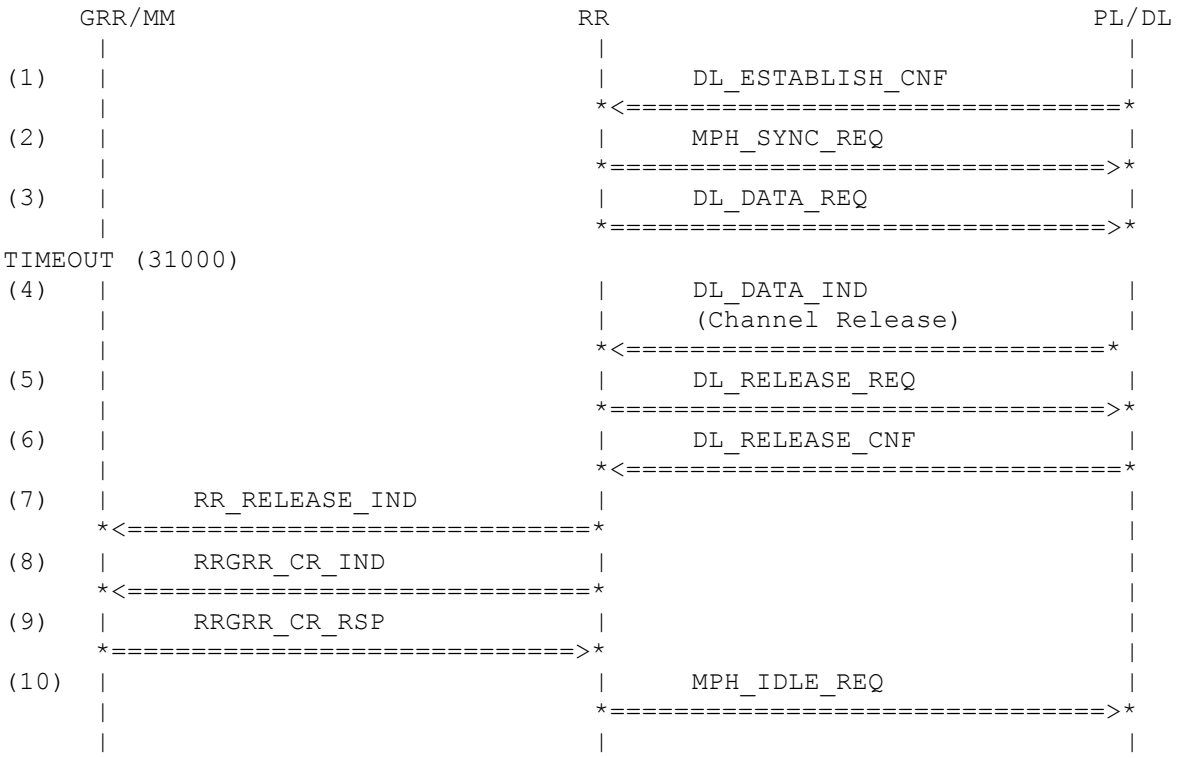
3.7.2 RRG602: RRGRR_DATA_IND

Description:

This primitive indicates to the GRR that a CTRL Message on DCCH for RR was received.

Preamble:

RRG601



Parametrization

Primitive	Parameter	Value
(1) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(2) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(3) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_CLASS_CHNG
	ti	TI_0
	mob_class_2	MOB_CLASS2

	mob_class_3	MOB_CLASS3_900
	}	
(4) DL_DATA_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_CHAN_REL
	ti	TI_0
	rr_cause	RR_CAUSE_0
	ba_range	NOT_USED
	group_chan_desc	G_CHANNEL_DESC
	group_ckn	NOT_USED
	gprs_resum	RESUMPTION_YES
	ba_list_pref	BA_LIST
	}	
(5) DL_RELEASE_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(6) DL_RELEASE_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(7) RR_RELEASE_IND		
	cause	NOT_USED
	sapi	SAPI_0
	gprs_resumption	NOT_USED
(8) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(9) RRGRR_CR_RSP		
(10) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NOT_PRESENT_8BIT
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT

History:

21-July-00	MSE	Initial
13-Feb-01	MSE	+MPH_SYNC_REQ
16-Jul-01	MSE	IDLE_REQ: +reorg_only
19-Feb-02	LG	+RRGRR_CR_RSP

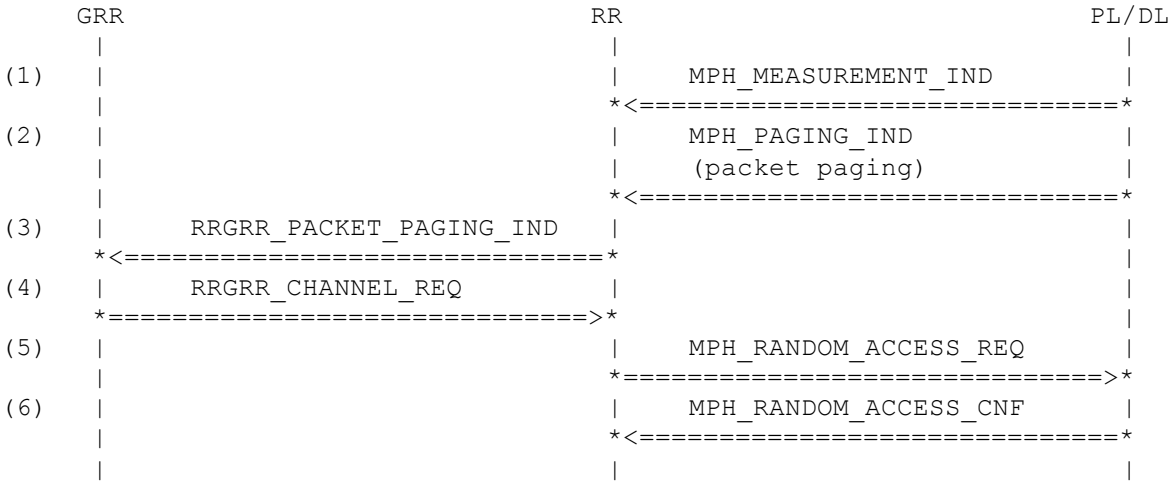
3.7.3 RRG605: RRGRR_PACKET_PAGING_IND

Description:

RR receives an MPH_PAGING_IND and forwards it to GRR. GRR requests the initiation of immediate assignment procedure. RR requests a channel.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_35
	rx_lev_sub	RX_LEV_35
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_PAGING_IND		
	identity_type	ID_PTMSI
	channel_needed	CN_PACKET
(3) RRGRR_PACKET_PAGING_IND		
	pg_type	RRGRR_PTMSI
(4) RRGRR_CHANNEL_REQ		
	req_data	ESTCS_PAGING
(5) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC721
(6) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1

History:

	08-Aug-01	LG	Initial

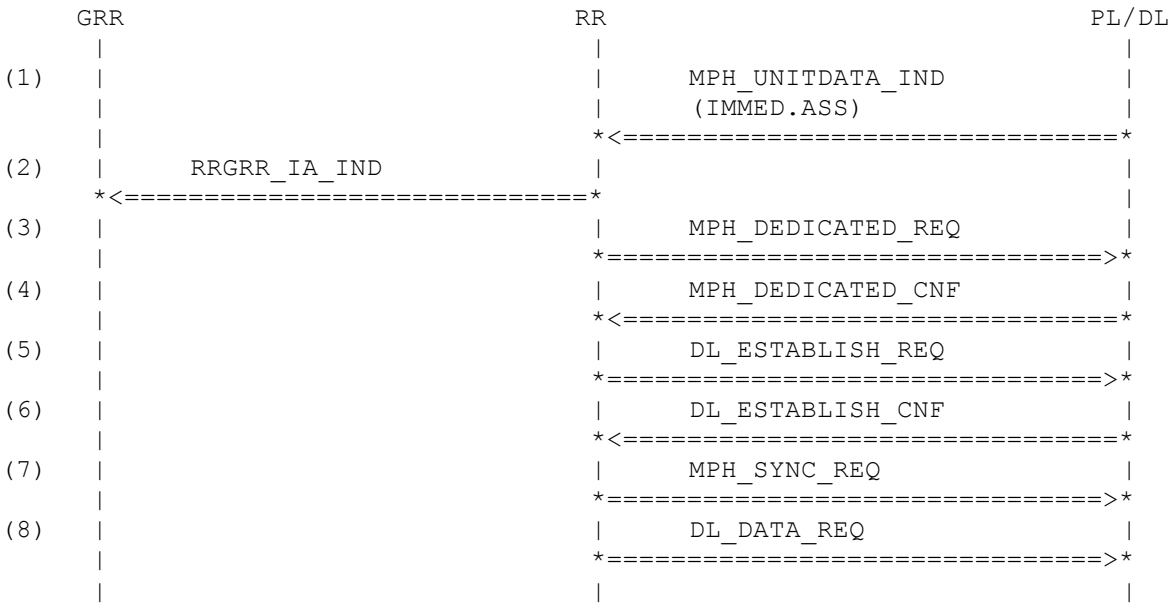
3.7.4 RRG610: Start of PDCH Assignment in dedicated mode

Description:

The MS has requested a channel after being paged.
The net assigns a dedicated channel. After establishment of data link the MS performs early classmark sending.

Preamble:

RRG605



Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_DED
	page_mode	PAGING_NORMAL
	chan_desc	CHANNEL_DESC_SDCCH
	pck_chan_desc	NOT_USED
	req_ref	REQUEST_REFERENCE_4A
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1

	ia_rest_oct	IA_REST_OCTETS_EMPTY
	}	
(2) RRGRR_IA_IND		
	fn	fn_780
	r_bit	CHAN_REQ_SENT_ONCE
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_DED
	page_mode	PAGING_NORMAL
	chan_desc	CHANNEL_DESC_SDCCH
	pck_chan_desc	NOT_USED
	req_ref	REQUEST_REFERENCE_4A
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_EMPTY
	}	
(3) MPH_DEDICATED_REQ		
	mod	MODE_IMM_ASSIGN
	start	STARTING_TIME_1
	ch_type	PRR_CHANNEL_TYPE_2
	ch_type2	DISPLAY_ONLY
	arfcn	ARFCN_X43
	bsic	BSIC_0
	ho_param	NOT_USED
	tr_para	PRR_TR_PARA_2A
	ciph	NO_CIPHERING
	amr_conf	NOT_USED
(4) MPH_DEDICATED_CNF		
	dedi_res	DEDI_RES_OK
(5) DL_ESTABLISH_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	D_RR_INIT_REQ
	ti	TI_0
	ciph_key_num	CKSN_RESERVED
	chan_coding	CHAN_CODING_TC770
	mob_class_2	MOB_CLASS2
	ded_tlli	DED_TLLI_BITBUFFER_TC770
	chan_req_desc	CHANNEL_REQ_TC770
	gprs_meas_res	MEASURES_TC770
	}	

(6) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(7) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(8) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_CLASS_CHNG
	ti	TI_0
	mob_class_2	MOB_CLASS2
	mob_class_3	MOB_CLASS3_900
	}	

History:

27-Jul-2001	LG	Initial
26-Feb-02	OT	Adaptations for AMR implementation

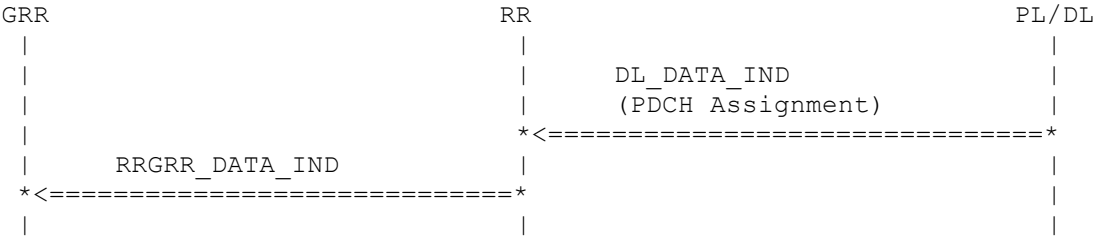
3.7.5 RRG620: Receive of PDCH Assignment

Description:

The MS is in dedicated mode. It receives a PDCH Assignment message and forwards it to GRR.

Preamble:

RRG610



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_PDCH_ASS_CMD
	ti	TI_0
	chan_desc	CHANNEL_DESC_FACCH4
	cell_chan_desc	CELL_CHAN_DESC_1
	freq_list_after	NOT_USED
	mob_alloc_after	MOBILE_ALLOCATION_4
	start_time	NOT_USED
	freq_list_before	NOT_USED
	chan_desc_before	NOT_USED
	freq_chan_seq	NOT_USED
	mob_alloc_before	NOT_USED
	pck_ul_ass	NOT_USED
	pck_dl_ass	RR_PACKET_DOWNLINK_TC980
	}	
(2) RRGRR_DATA_IND		
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_PDCH_ASS_CMD
	ti	TI_0
	chan_desc	CHANNEL_DESC_FACCH4
	cell_chan_desc	CELL_CHAN_DESC_1
	freq_list_after	NOT_USED
	mob_alloc_after	MOBILE_ALLOCATION_4
	start_time	NOT_USED

Confidential

freq_list_before	NOT_USED
chan_desc_before	NOT_USED
freq_chan_seq	NOT_USED
mob_alloc_before	NOT_USED
pck_ul_ass	NOT_USED
pck_dl_ass	RR_PACKET_DOWNLINK_TC980

}

History:

01-Aug-2001	LG	Initial

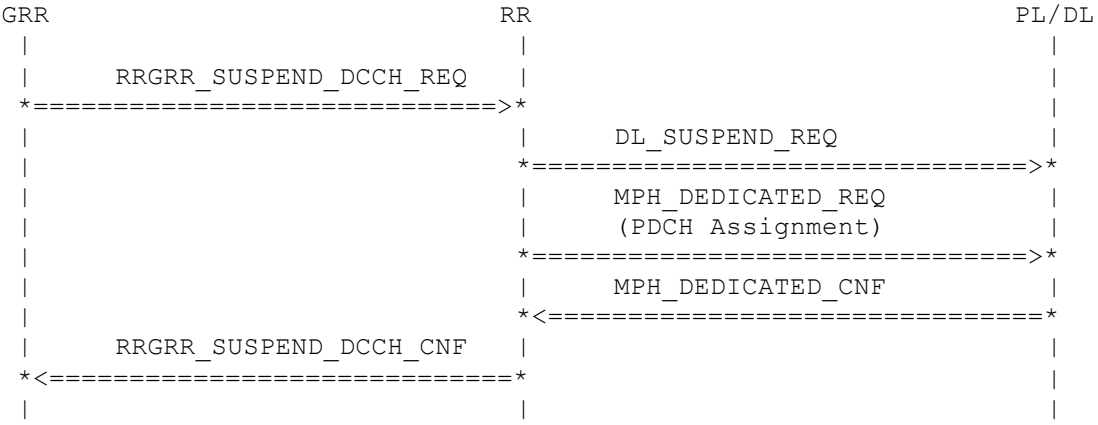
3.7.6 RRG622: PDCH Assignment: GRR suspends the dedicated mode

Description:

The MS is in dedicated mode and had forwarded a PDCH Assignment message to GRR. GRR requests the suspension of dedicated mode. After RR has sent an RRGRR_SUSPEND_DCCH_CNF GRR will try to establish a TBF.

Preamble:

RRG620



Parametrization

Primitive	Parameter	Value
(1) RRGRR_SUSPEND_DCCH_REQ		
(2) DL_SUSPEND_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(3) MPH_DEDICATED_REQ		
	mod	MODE_PDCH_ASSIGN
	start	NO_STARTING_TIME
	ch_type	DISPLAY_ONLY
	ch_type2	DISPLAY_ONLY
	arfcn	DISPLAY_ONLY
	bsic	DISPLAY_ONLY
	ho_param	DISPLAY_ONLY
	tr_para	DISPLAY_ONLY
	ciph	DISPLAY_ONLY
	amr_conf	NOT_USED
(4) MPH_DEDICATED_CNF		
	dedi_res	DEDI_RES_OK
(5) RRGRR_SUSPEND_DCCH_CNF		

History:

08-Aug-2001	LG	Initial
26-Feb-02	OT	Adaptations for AMR implementation

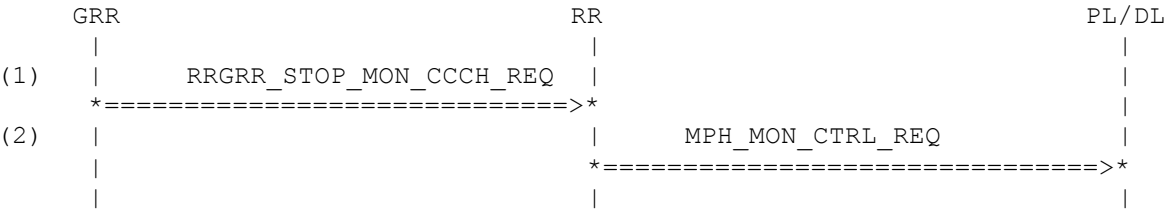
3.7.7 RRG624: PDCH Assignment: GRR stops monitoring - PBCCH present

Description:

The MS is in dedicated mode and had forwarded a PDCH Assignment message to GRR. GRR had requested the suspension of dedicated mode. Before GRR tries to establish a TBF it will stop the monitoring of CCCH.

Preamble:

RRG622



Parametrization

	Primitive	Parameter	Value
(1)	RRGRR_STOP_MON_CCCH_REQ		
		is_pbcch_present	TRUE
(2)	MPH_MON_CTRL_REQ		
		action	STOP_MON_CCCH
		si_to_read	UNSPECIFIED_SI

History:

13-Aug-2001	LG	Initial

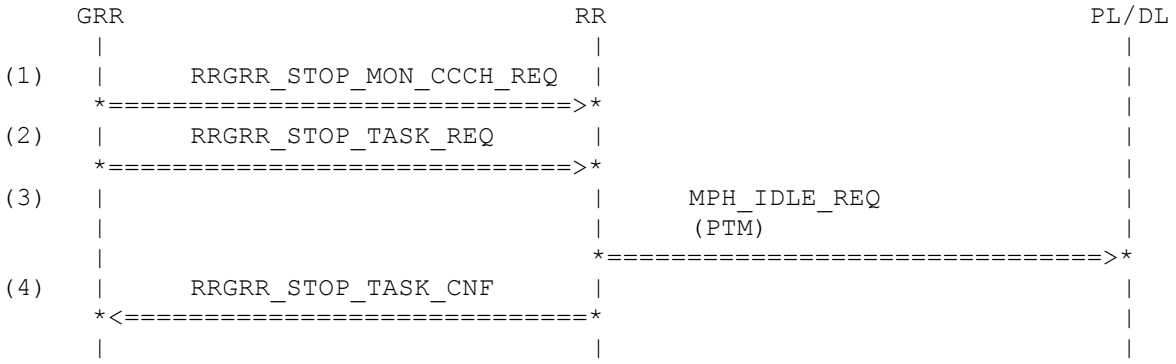
3.7.8 RRG625: PDCH Assignment: GRR stops monitoring - No PBCCH present

Description:

The MS is in dedicated mode and had forwarded a PDCH Assignment message to GRR. GRR had requested the suspension of dedicated mode. Before GRR tries to establish a TBF it will stop the monitoring of CCCH.

Preamble:

RRG622



Parametrization

Primitive	Parameter	Value
(1) RRGRR_STOP_MON_CCCH_REQ		
	is_pbcch_present	FALSE
(2) RRGRR_STOP_TASK_REQ		
	v_stop_ccch	STOP_CCCH
(3) MPH_IDLE_REQ		
	mod	MODE_PACKET_TRANSFER
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT
(4) RRGRR_STOP_TASK_CNF		

History:

13-Aug-2001	LG	Initial

3.7.9 RRG626: PDCH Assignment: GRR indicates establishment of TBF

Description:

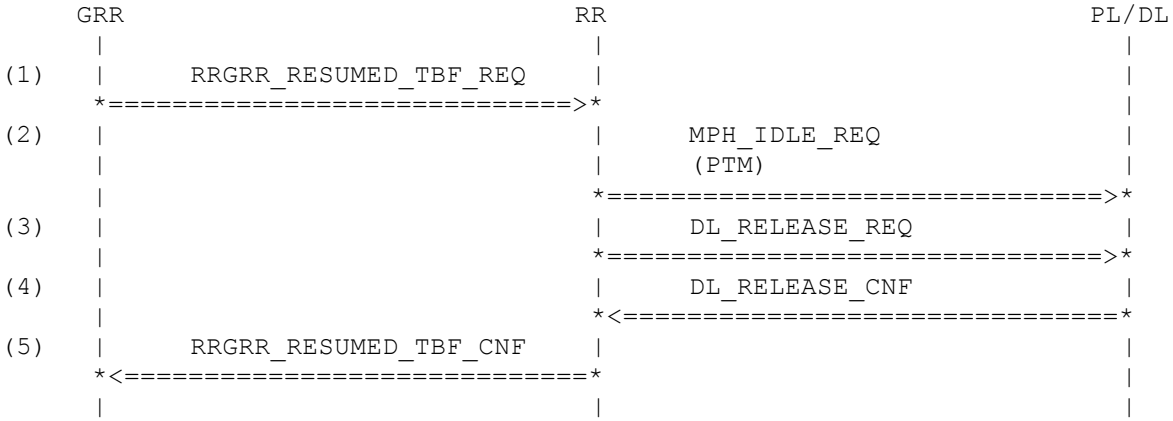
After receive of PDCH Assignment in dedicated mode and suspending the dedicated mode GRR has successfully established a TBF. Now GRR indicates the establishment. Therefore RR discards the old channel configuration.

Variants:

<A>....<B>

Preamble:

<A>RRG624
<B>RRG625



Parametrization

Primitive	Parameter	Value
(1) RRGRR_RESUMED_TBF_REQ		
(2) MPH_IDLE_REQ		
	mod	MODE_PACKET_TRANSFER
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT
(3) DL_RELEASE_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(4) DL_RELEASE_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0

(5) RRGRR_RESUMED_TBF_CNF

History:

	14-Aug-2001	LG	Initial

3.7.10 RRG628: PDCH Assignment: GRR indicates release of TBF

Description:

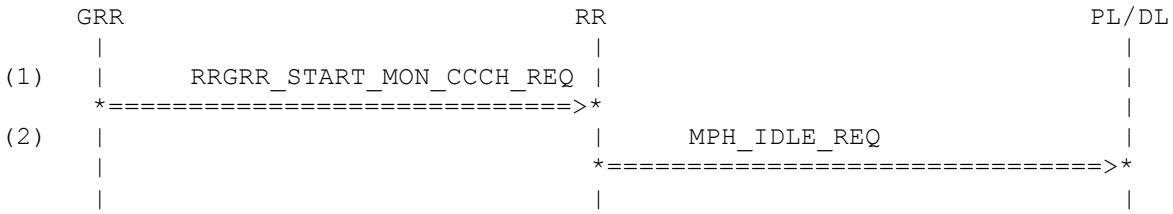
After receive of PDCH Assignment in dedicated mode and establishing a TBF GRR now indicates that GPRS data transfer is finished. The MS returns to packet idle mode.

Variants:

<A>....<B>

Preamble:

<A>RRG626A
<B>RRG626B



Parametrization

Primitive	Parameter	Value
(1) RRGRR_START_MON_CCCH_REQ		
	pag_mode	PAG_MODE_DEFAULT
	split_pg	CYCLE_ROUND_UP_1
(2) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_42
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED

History:

14-Aug-2001	LG	Initial

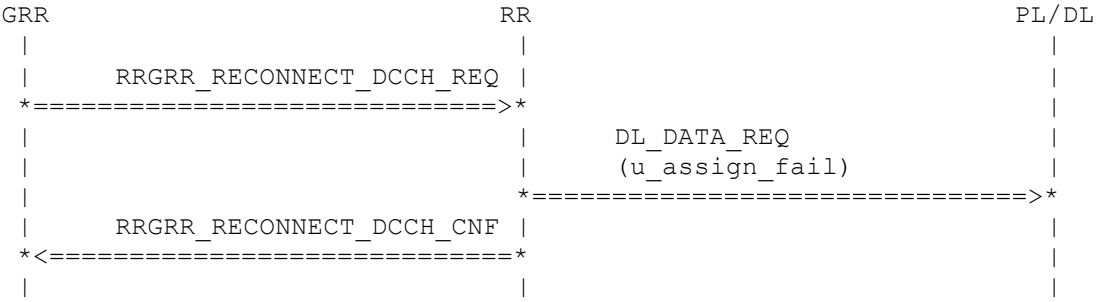
3.7.11 RRG630: PDCH Assignment: GRR may not accept PDCH assignment

Description:

The MS is in dedicated mode and had forwarded a PDCH Assignment message to GRR. GRR may not accept the assignment and therefor reqests the reconnection to old channel. RR sends an assignment failure message to the net and confirms to GRR.

Preamble:

RRG620



Parametrization

	Primitive	Parameter	Value
(1)	RRGRR_RECONNECT_DCCH_REQ		
		reconn_cause	RECONN_FREQ_NOT_IMPL
(2)	DL_DATA_REQ		
		ch_type	CH_TYPE_SDCCH
		sapi	SAPI_0
		sdu	
		{	
		component	RR
		direction	UPLINK
		pd	U_ASSIGN_FAIL
		ti	TI_0
		rr_cause	RRC_FREQ_NOT_IMPL
		}	
(3)	RRGRR_RECONNECT_DCCH_CNF		
		reconn_state	RECONN_OK

History:

14-Aug-2001	LG	Initial

3.7.12 RRG632: PDCH Assignment: Timeout during TBF establishment

Description:

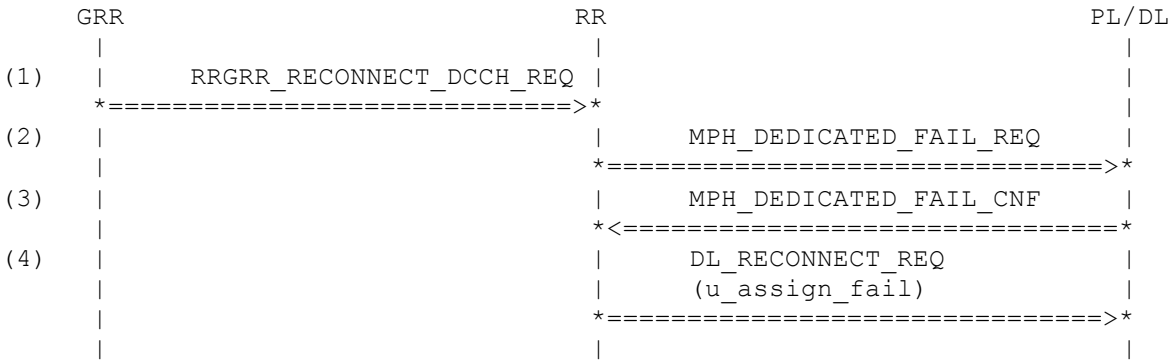
The MS is in dedicated mode and had forwarded a PDCH Assignment message to GRR. GRR has tried to establish a TBF but the timer T3190 has expired. Therefore GRR request to return to old channel.

Variants:

<A>....<B>

Preamble:

<A>RRG624
<B>RRG625



Parametrization

Primitive	Parameter	Value
(1) RRGRR_RECONNECT_DCCH_REQ		
	reconn_cause	RECONN_PROT_UNSPECIFIED
(2) MPH_DEDICATED_FAIL_REQ		
	param	PARAM_0
(3) MPH_DEDICATED_FAIL_CNF		
	param	PARAM_0
(4) DL_RECONNECT_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_ASSIGN_FAIL
	ti	TI_0
	rr_cause	RRC_PROT_UNSPECIFIED
	}	

History:

15-Aug-2001	LG	Initial

3.7.13 RRG634: PDCH Assignment: Successful Reconnection

Description:

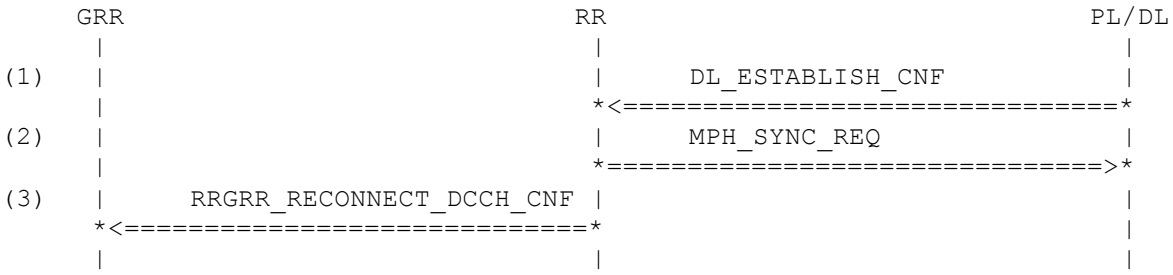
During the PDCH assignment procedure the establishment of TBF has failed. The reconnection to old channel was started. This is completed now successfully.

Variants:

<A>....<B>

Preamble:

<A>RRG632A
<B>RRG632B



Parametrization

Primitive	Parameter	Value
(1) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(2) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(3) RRGRR_RECONNECT_DCCH_CNF		
	reconn_state	RECONN_OK

History:

15-Aug-2001	LG	Initial

3.7.14 RRG636: PDCH Assignment: Reconnection Failure

Description:

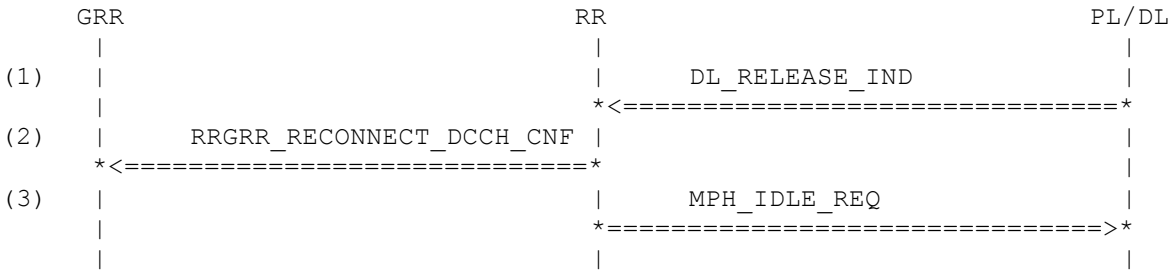
During the PDCH assignment procedure the establishment of TBF has failed. The reconnection to old channel was started. The reconnection fails.

Variants:

<A>....<B>

Preamble:

<A>RRG632A
<B>RRG632B



Parametrization

Primitive	Parameter	Value
(1) DL_RELEASE_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	cs	NOT_PRESENT_8BIT
(2) RRGRR_RECONNECT_DCCH_CNF		
	reconn_state	RECONN_LOW_FAIL
(3) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_FF
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT

History:

15-Aug-2001	LG	Initial

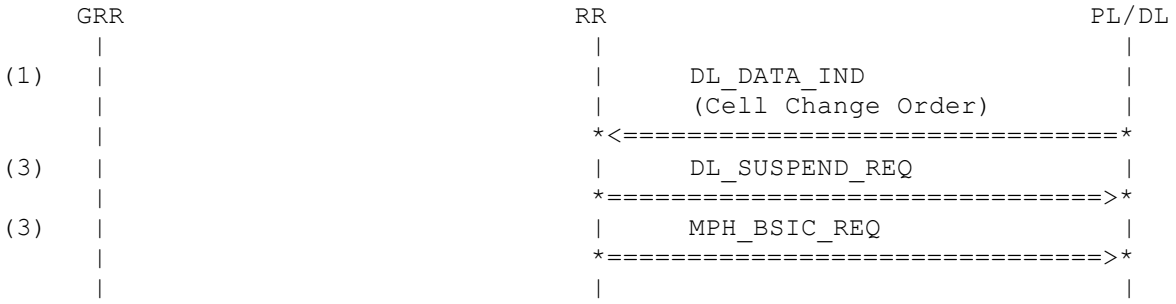
3.7.15 RRG640: CCO: Receive of Network Commanded Cell Change Order

Description:

The MS is in dedicated mode. It receives a Network Commanded Cell Change Order message and forwards it to GRR.

Preamble:

RRG610



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_CHANGE_ORDER
	ti	TI_0
	cell_desc	CELL_DESC_2
	nc_mode	NC_MODE_TC640
	}	
(2) DL_SUSPEND_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(3) MPH_BSIC_REQ		
	arfcn	ARFCN_531

History:

17-Aug-2001	LG	Initial

3.7.16 RRG641: CCO: Receive Sys Info on new cell

Description:

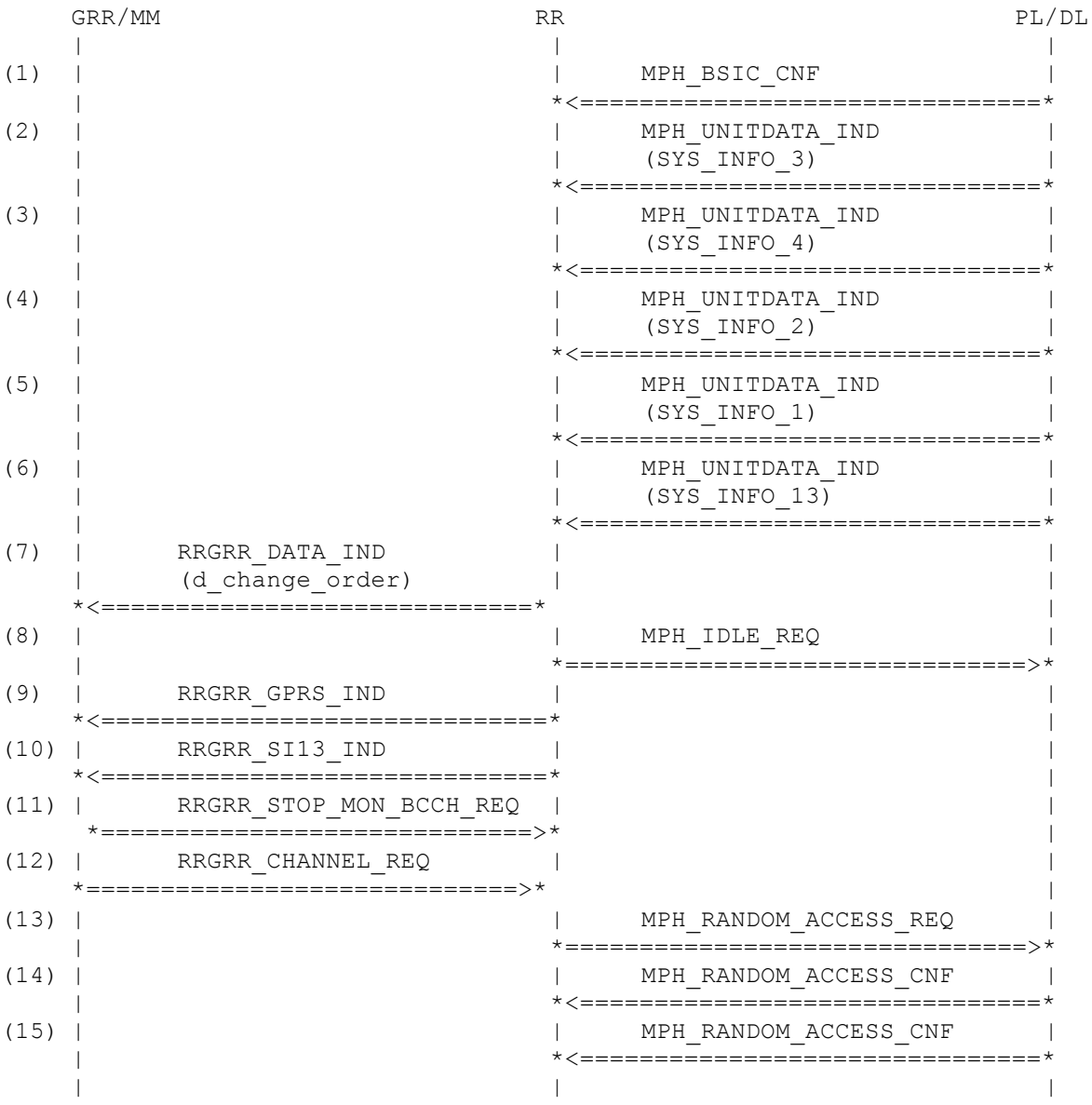
The MS has successfully synchronized to the new cell and reads the System Infromation Messages.
A: The new cell doesn't have a PBCCH
B: The new cell has a PBCCH

Preamble:

RRG640

Variants:

<A>..<B>



Parametrization

Primitive	Parameter	Value

(1) MPH_BSIC_CNF

arfcn	ARFCN_531
bsic	BSIC_3
cs	CS_NO_ERROR

(2) MPH_UNITDATA_IND

arfcn	ARFCN_531
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_SYS_INFO_3
ti	TI_0
cell_ident	CELL_IDENT_3748
loc_area_ident	LOC_AREA_IDENT_123_2147
ctrl_chan_desc	CTRL_CHAN_DESC_1
cell_opt_bcch	CELL_OPT_BCCH_1
cell_select	CELL_SELECT_1
rach_ctrl	RACH_CTRL_1
si3_rest_oct	SI3_REST_GPRS_TC701
}	

(3) MPH_UNITDATA_IND

arfcn	ARFCN_531
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_SYS_INFO_4
ti	TI_0
loc_area_ident	LOC_AREA_IDENT_123_2147
cell_select	CELL_SELECT_1
rach_ctrl	RACH_CTRL_1
chan_desc	NOT_USED
mob_alloc	NOT_USED
si4_rest_oct	SI4_REST_GPRS_TC701
}	

(4) MPH_UNITDATA_IND

arfcn	ARFCN_531
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_SYS_INFO_2
ti	TI_0
neigh_cell_desc	NCELL_DESC_2
ncc_permit	NCC_PERMITTED_1
rach_ctrl	RACH_CTRL_1
}	

(5) MPH_UNITDATA_IND

arfcn	ARFCN_531
fn	fn_780

	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_531
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
<A>	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
<B>	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(7) RRGRR_DATA_IND		
	fn	NOT_USED
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_CHANGE_ORDER
	ti	TI_0
	cell_desc	CELL_DESC_2
	nc_mode	NC_MODE_TC640
	}	
(8) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_531
	ext_bcch	BSIC_3
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(9) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702

(10) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
<A>	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
<B>	si13_rest_oct	SI13_REST_GPRS_TC701
	}	
(11) RRGRR_STOP_MON_BCCH_REQ		
(12) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(13) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(14) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(15) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1

History:

31-Aug-2001	VK	Initial

3.7.17 RRG642: CCO: Successful Immediate Assignment

Description:

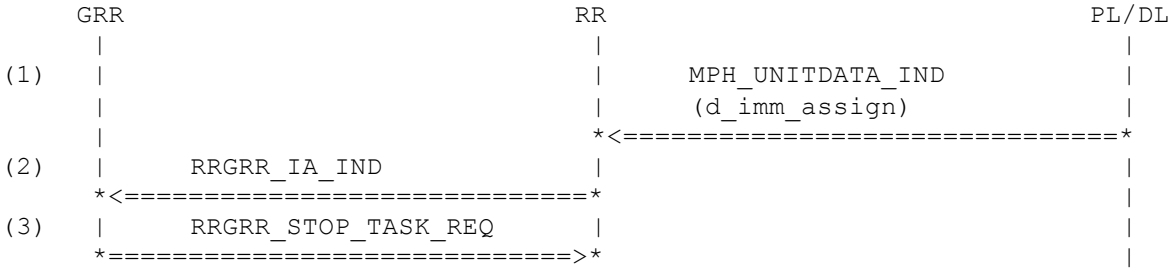
During a Network Controlled Cell Change Order Procedure the Immediate Assignment messages is passed to GRR.
A: The new cell doesn't have a PBCCH
B: The new cell has a PBCCH

Variants:

<A>..<B>

Preamble:

<A>RRG641A
<B>RRG641B



```
(4)   |                              |      MPH_IDLE_REQ                |
      |                              *================================>*
(5)   |      RRGRR_STOP_TASK_CNF     |                                  |
      *<==============================*                                  |
      |                              |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	
	IA_REST_OCTETS_UL_ASSIGN_TMA1	
	}	
(2) RRGRR_IA_IND		
	fn	fn_780
	r_bit	CHAN_REQ_SENT_MORE
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	
	IA_REST_OCTETS_UL_ASSIGN_TMA1	
	}	
(3) RRGRR_STOP_TASK_REQ		
	v_stop_ccch	STOP_CCCH

(4) MPH_IDLE_REQ

mod	MODE_PACKET_TRANSFER
arfcn	ARFCN_X43
ext_bcch	BSIC_5
comb_ccch	CCD_CCCH_1_NOT_COMB
tn	TN_0
dlt	DLT_23
pg	PG_0
bs_ag_blocks_res	BS_AG_BLKS_RES_5
bs_pa_mfrms	BS_PA_MFRMS_2
power	MS_TXPWR_MAX_CCH_02
ncc_permitted	NCC_PERMITTED_1
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	NOT_PRESENT_8BIT

(5) RRGRR_STOP_TASK_CNF

History:

10-Sep-2001	VK	Initial

3.7.18 RRG643: CCO: Immediate Assignment rejected

Description:

During a Network Controlled Cell Change Order Procedure the Ms receives a Immediate Assignment Reject message.

A: The new cell doesn't have a PBCCH

B: The new cell has a PBCCH

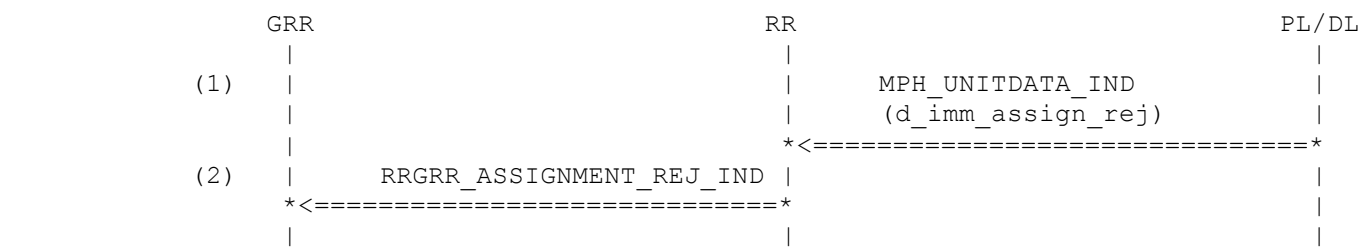
Variants:

<A>..<B>

Preamble:

<A>RRG641A

<B>RRG641B



Parametrization

Primitive	Parameter	Value

(1) MPH_UNITDATA_IND

arfcn	ARFCN_X43
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_IMM_ASSIGN_REJ

	ti	TI_0
	page_mode	PAGING_NORMAL
	req_ref	REQUEST_REFERENCE_1
	t3122	WAIT_INDICATION_0
	req_ref_2	REQUEST_REFERENCE_2
	t3122_2	WAIT_INDICATION_0
	req_ref_3	REQUEST_REFERENCE_3
	t3122_3	WAIT_INDICATION_0
	req_ref_4	REQUEST_REFERENCE_4
	t3122_4	WAIT_INDICATION_0
	}	
(2) RRGRR_ASSIGNMENT_REJ_IND		
	wait_ind	WAIT_INDICATION_0
	r_bit	CHAN_REQ_SENT_MORE

History:

10-Sep-2001	VK	Initial

3.7.19 RRG644: CCO: Failure, start Reconnection

Description:

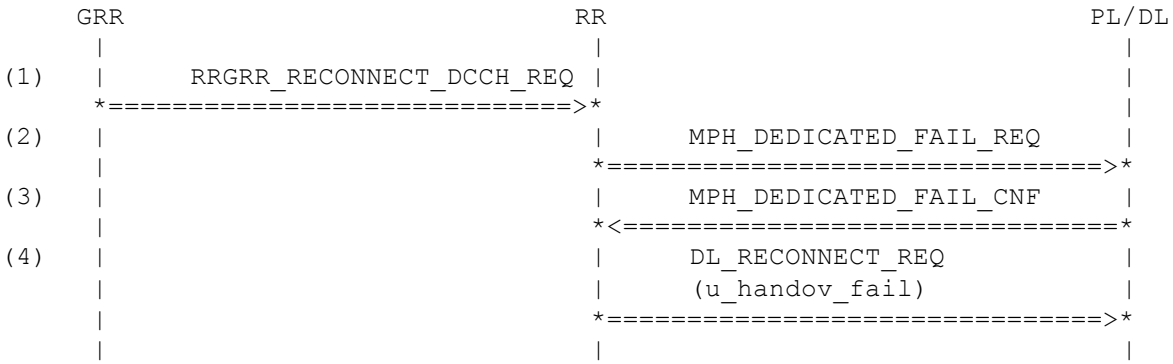
The Network Controlled Cell Change Order Procedure fails. RR starts the Reconnection to the Old Cell.
A: The new cell doesn't have a PBCCH
B: The new cell has a PBCCH

Variants:

<A>..<B>

Preamble:

<A>RRG642A
<B>RRG642B



Parametrization

Primitive	Parameter	Value
(1) RRGRR_RECONNECT_DCCH_REQ		
	reconn_cause	RECONN_FREQ_NOT_IMPL
(2) MPH_DEDICATED_FAIL_REQ		
	param	PARAM_0

(3) MPH_DEDICATED_FAIL_CNF		
	param	PARAM_0
(4) DL_RECONNECT_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_HANDOV_FAIL
	ti	TI_0
	rr_cause	RR_CAUSE_0A
	}	

History:

10-Sep-2001	VK	Initial

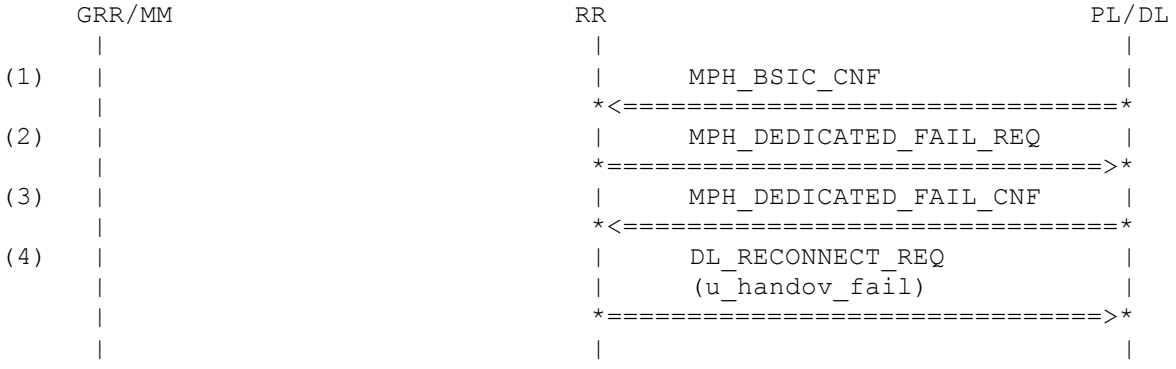
3.7.20 RRG645: CCO: Cannot access new cell, start Reconnection

Description:

The Network Controlled Cell Change Order Procedure fails. The synchronization to the new cell fails. RR starts the Reconnection to the Old Cell.

Preamble:

RRG640



Parametrization

Primitive	Parameter	Value
(1) MPH_BSIC_CNF		
	arfcn	ARFCN_531
	bsic	BSIC_3
	cs	CS_NO_BCCH_AVAIL
(2) MPH_DEDICATED_FAIL_REQ		
	param	PARAM_0
(3) MPH_DEDICATED_FAIL_CNF		
	param	PARAM_0

(4) DL_RECONNECT_REQ

ch_type	CH_TYPE_SDCCH
sapi	SAPI_0
sdu	
{	
component	RR
direction	UPLINK
pd	U_HANDOV_FAIL
ti	TI_0
rr_cause	RRC_PROT_UNSPECIFIED
}	

History:

10-Sep-2001	VK	Initial

3.7.21 RRG646: CCO: Failure, successful Reconnection

Description:

The Network Controlled Cell Change Order Procedure has failed, the Reconnection to the Old Cell is successful.
A: The new cell didn't have a PBCCH
B: The new cell had a PBCCH

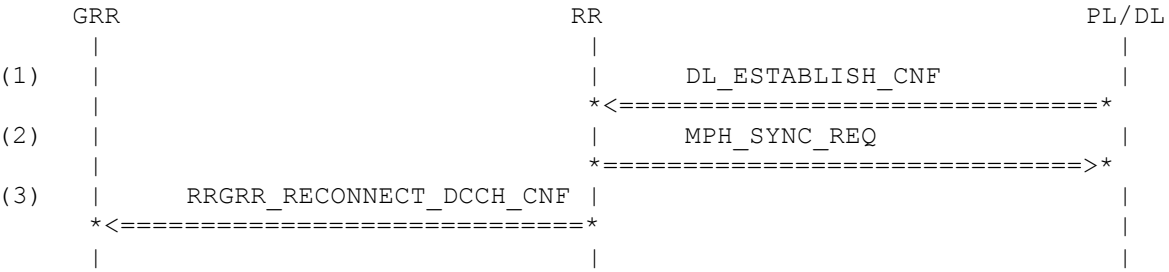
Variants:

<A>..<B>

Preamble:

<A>RRG644A
<B>RRG644B

Preamble:



Parametrization

Primitive	Parameter	Value
(1) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(2) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(3) RRGRR_RECONNECT_DCCH_CNF		
	reconn_state	RECONN_OK

History:

	10-Sep-2001	VK	Initial

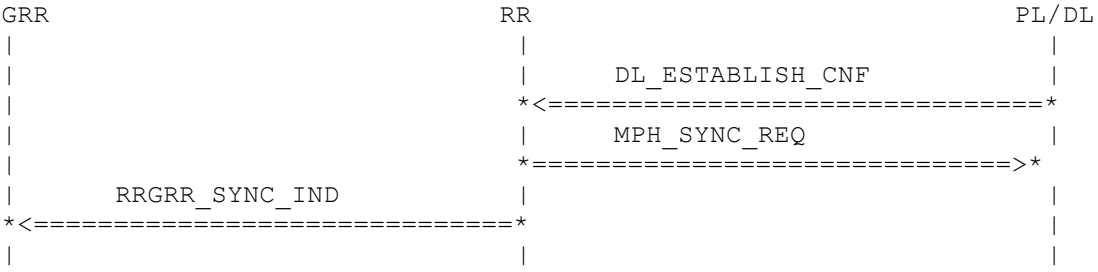
3.7.22 RRG647: CCO: Failure, reconnection successful after failed access to new cell

Description:

The Network Controlled Cell Change Order Procedure has failed, the Reconnection to the Old Cell is successful.

Preamble:

RRG645



Parametrization

Primitive	Parameter	Value
(1) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(4) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(2) RRGRR_SYNC_IND		
	sync_res	SYNC_FAILED

History:

	10-Sep-2001	VK	Initial

3.7.23 RRG648: CCO: Failure, reconnection fails after failed access to new cell

Description:

The Network Controlled Cell Change Order Procedure has failed, the Reconnection to the Old Cell is unsuccessful.

Preamble:

RRG645

Parametrization

Primitive	Parameter	Value

(1) DL_RELEASE_IND

ch_type	CH_TYPE_SDCCH
sapi	SAPI_0
cs	NOT_PRESENT_8BIT

(2) MPH_IDLE_REQ

mod	MODE_CELL_RESELECTION
arfcn	ARFCN_X43
ext_bcch	BSIC_5
comb_ccch	CCD_CCCH_1_NOT_COMB
tn	TN_0
dlt	DLT_23
pg	PG_0
bs_ag_blocks_res	BS_AG_BLKS_RES_5
bs_pa_mfrms	BS_PA_MFRMS_2
power	MS_TXPWR_MAX_CCH_02
ncc_permitted	NCC_PERMITTED_FF
reorg_only	NORMAL_PGM
eotd_avail	EOTD_NOT_AVAIL
gprs_support	NOT_PRESENT_8BIT

History:

10-Sep-2001	VK	Initial

3.7.24 RRG649: CCO: Failure, unsuccessful Reconnection

Description:

The Network Controlled Cell Change Order Procedure has failed, the Reconnection to the Old Cell is unsuccessful.

A: The new cell didn't have a PBCCH

B: The new cell had a PBCCH

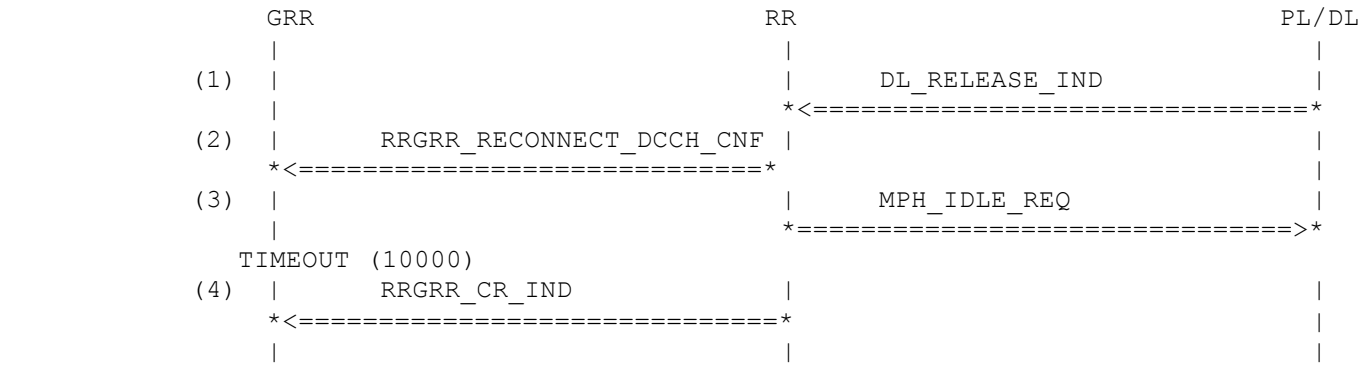
Variants:

<A>..<B>

Preamble:

<A>RRG644A

<B>RRG644B



Parametrization

Primitive	Parameter	Value
(1) DL_RELEASE_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	cs	NOT_PRESENT_8BIT
(2) RRGRR_RECONNECT_DCCH_CNF		
	reconn_state	RECONN_LOW_FAIL
(3) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_FF
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	NOT_PRESENT_8BIT
(4) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL

History:

15-Aug-2001	LG	Initial

3.8 Mobile Terminated Calls (700)

3.8.1 RRG701: RRGRR_RR_EST_REQ

Description:

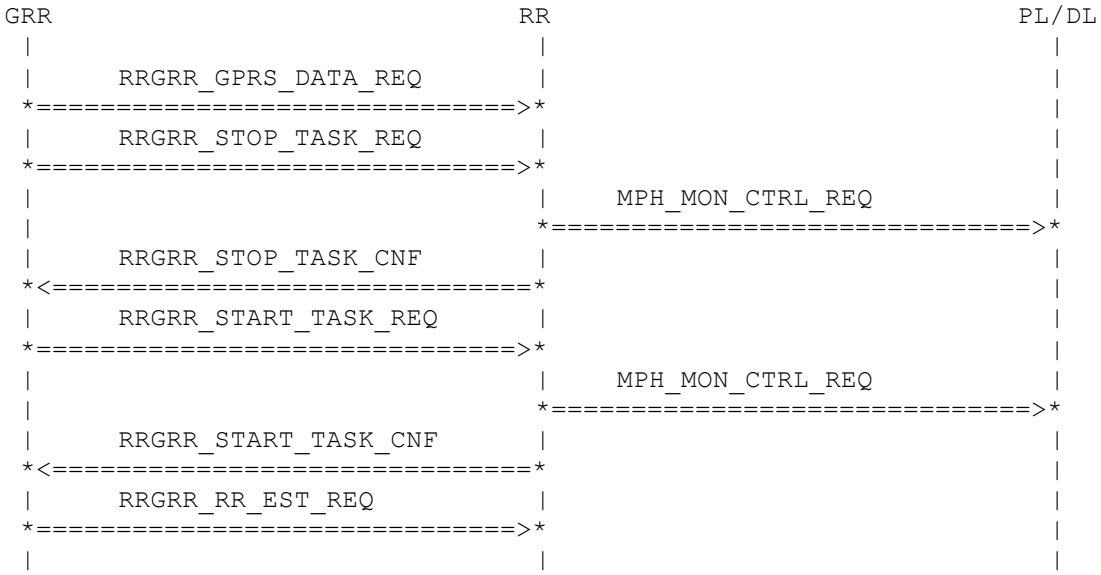
This primitive indicates to the RR a RR connection establishment was received on paging channel (PCCCH or PACCH). RR has to start RR connection as if it has received a connection establishment on CCCH.

Preamble:

RRG106

Variants:

<A>....<B>



Parametrization

Primitive	Parameter	Value
(1) RRGRR_GPRS_DATA_REQ		
	old_ptmsi	DUMMY_LONG
	new_ptmsi	NOT_PRESENT_32BIT
	tlli	DUMMY_LONG
	p_chan_req_des	REQUESTED_CHANNEL_TC970
	gprs_meas_results	MEASUREMENT_RESULTS_TC970
	mac_req	MAC_FIX_ALLOC
	cs_req	COD_S_3
	rai	RAI_102
(2) RRGRR_STOP_TASK_REQ		
	v_pim_pbcch	LEAVE_PIM_PBCCH

(3) MPH_MON_CTRL_REQ			
		action	LEAVING_PIM_PBCCH
		si_to_read	NOT_USED
(4) RRGRR_STOP_TASK_CNF			
(5) RRGRR_START_TASK_REQ			
		state	TASK_STATE_PTM
(6) MPH_MON_CTRL_REQ			
		action	ENTER_PTM_PBCCH
		si_to_read	NOT_USED
(7) RRGRR_START_TASK_CNF			
(8) RRGRR_RR_EST_REQ			
		non_gprs	NON_GPRS_EST_REQ_TC860
		ch_needed	CHANNEL_ANY
		ident_type	ID_TMSI
		emlpp_prio	PRIO_NOT_SUPPORTED
<A>		susp_req	SUSPENSION_REQ_NOT_NEEDED
<B>		susp_req	SUSPENSION_REQ_NEEDED

History:

21-July-00	MSE	Initial
25-Oct-01	VK	add ident_type

3.8.2 RRG702: MT CS call during GPRS – without suspension request

Description:

The base station sends an Immediate Assignment message for the MS. RR informs the PL and tries to establish the layer 2 connection.

A: case history without suspension request needed

B: case history with suspension request needed

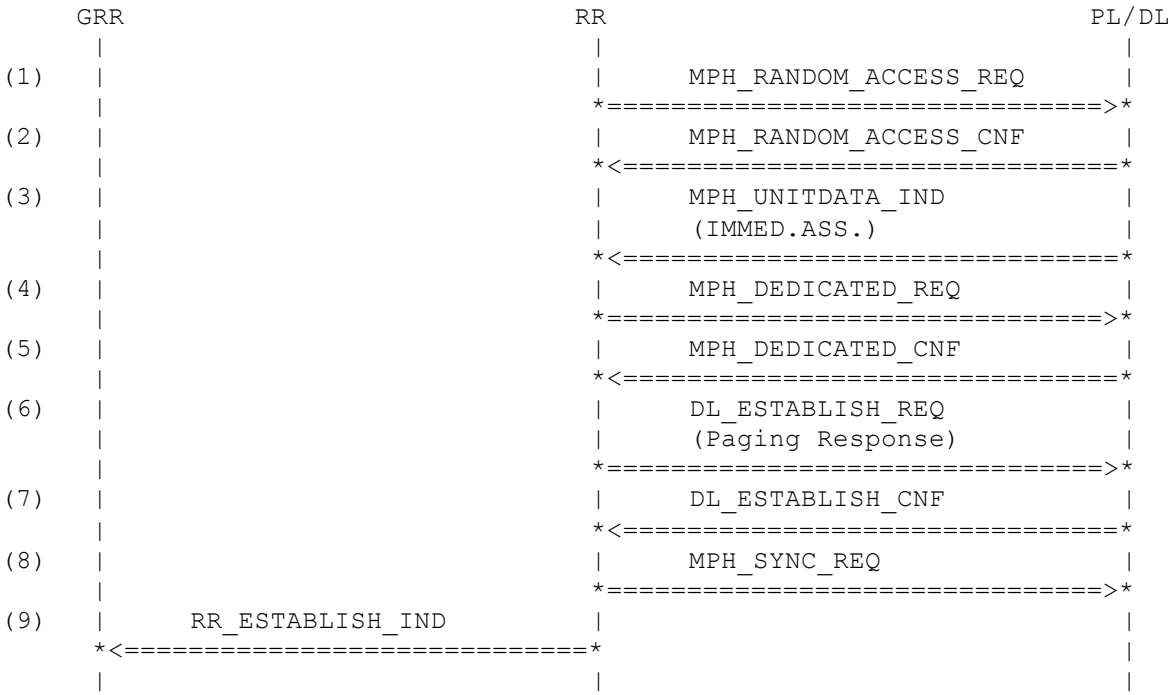
Variants:

<A>....<B>

Preamble:

<A>RRG701A

<B>RRG701B



Parametrization

Primitive	Parameter	Value
(1) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC871
(2) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(3) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR

	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_DED
	page_mode	PAGING_NORMAL
	chan_desc	CHANNEL_DESC_SDCCH
	pck_chan_desc	NOT_USED
	req_ref	REQUEST_REFERENCE_4A
	time_advance	TIMING_ADVANCE_27
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	NOT_USED
	}	
(4) MPH_DEDICATED_REQ		
	mod	MODE_IMM_ASSIGN
	start	STARTING_TIME_1
	ch_type	PRR_CHANNEL_TYPE_2
	ch_type2	NOT_USED
	arfcn	ARFCN_X43
	bsic	NOT_USED
	ho_param	NOT_USED
	tr_para	PRR_TR_PARA_2
	ciph	NO_CIPHERING
	amr_conf	NOT_USED
(5) MPH_DEDICATED_CNF		
	dedi_res	DEDI_RES_OK
(6) DL_ESTABLISH_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_PAG_RES
	ti	TI_0
	ciph_key_num	CKSN_RESERVED
	mob_class_2	MOB_CLASS2
	mob_ident	NOT_USED
	}	
(7) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(8) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(9) RR_ESTABLISH_IND		
	param	NOT_USED

Confidential

History:

09-Nov-00	MSE	Initial
13-Feb-01	MSE	+MPH_SYNC_REQ
26-Feb-02	OT	Adaptations for AMR implementation

3.8.3 RRG703: Early Classmark Change

Description:

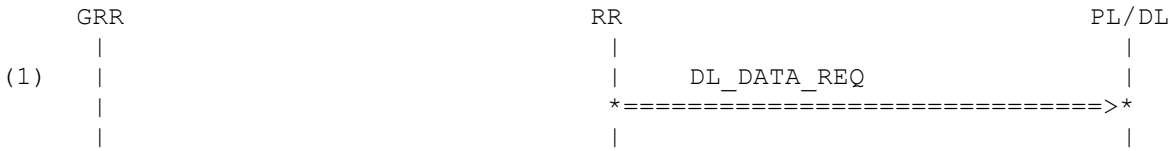
Early classmark sending consists in the mobile station sending as early as possible after access a CLASSMARK CHANGE message to provide the network with additional classmark information.

Variants:

<A>....<B>

Preamble:

<A>RRG702A
<B>RRG702B



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_CLASS_CHNG
	ti	TI_0
	mob_class_2	MOB_CLASS2
	mob_class_3	MOB_CLASS3_900
	}	

History:

13-Feb-01	MSE	Initial	

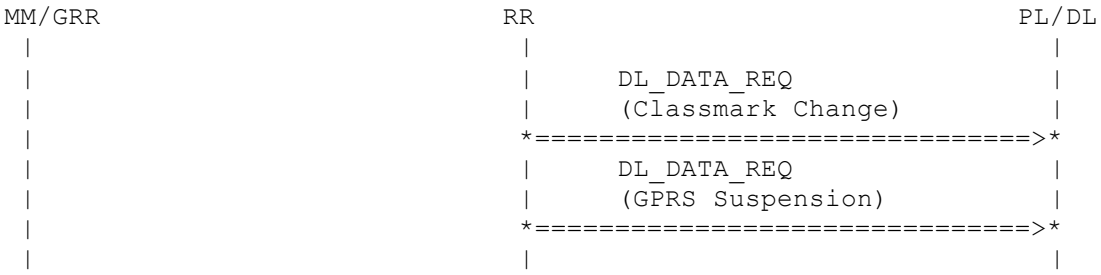
3.8.4 RRG704: MT CS call during GPRS – with suspension request

Description:

GPRS suspension procedure - This can be done as early as possible after access but shall be done after sending a CLASSMARK CHANGE message. The RR sublayer of the mobile station shall indicate a RR GPRS suspend condition to the MM sublayer.

Preamble:

RRG702B



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_CLASS_CHNG
	ti	TI_0
	mob_class_2	MOB_CLASS2
	mob_class_3	MOB_CLASS3_900
	}	
(2) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_GPRS_SUSP_REQ
	ti	TI_0
	ded_tlli	NOT_USED
	rout_area_id	ROUT_AREA_ID_102
	susp_cause	SUSP_C_CALL
	}	

History:

11-Dec-00	MSE	Initial

3.8.5 RRG705: Channel release after 871/2

Description:

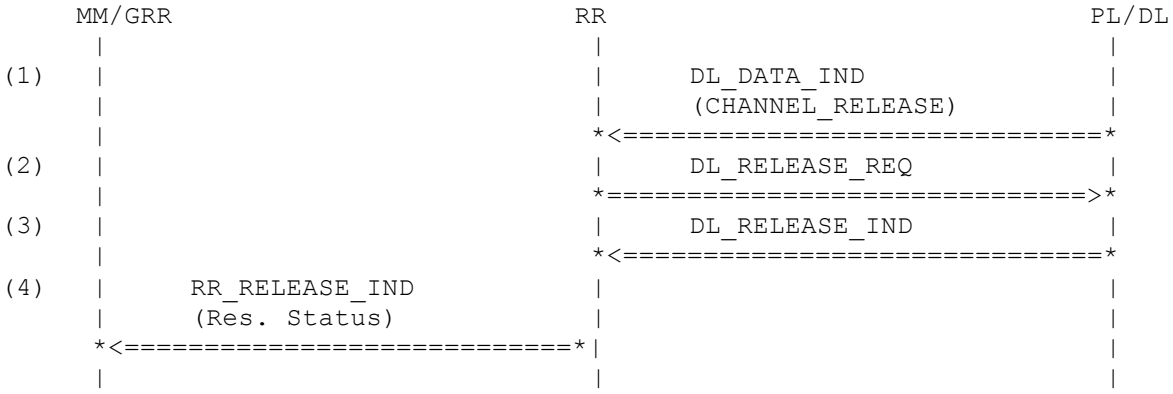
After a connection has been established, it is now time to shut it down.

Variants:

<A>....<B>

Preamble:

<A>RRG703A
<B>RRG704



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_CHAN_REL
	ti	TI_0
	rr_cause	RRC_NORMAL_EVENT
	ba_range	NOT_USED
	group_chan_desc	NOT_USED
	group_ckn	NOT_USED
<A>	gprs_resum	RESUMPTION_NO
<B>	gprs_resum	RESUMPTION_YES
	ba_list_pref	NOT_USED
	}	
(2) DL_RELEASE_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(3) DL_RELEASE_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	cs	NOT_PRESENT_8BIT

(4)	RR_RELEASE_IND		
		cause	RRCS_NORM
		sapi	SAPI_0
	<A>	gprs_resumption	GPRS_RESUMPTION_NOT_ACK
	<B>	gprs_resumption	GPRS_RESUMPTION_ACK

History:

	16-Jan-01	MSE	Initial

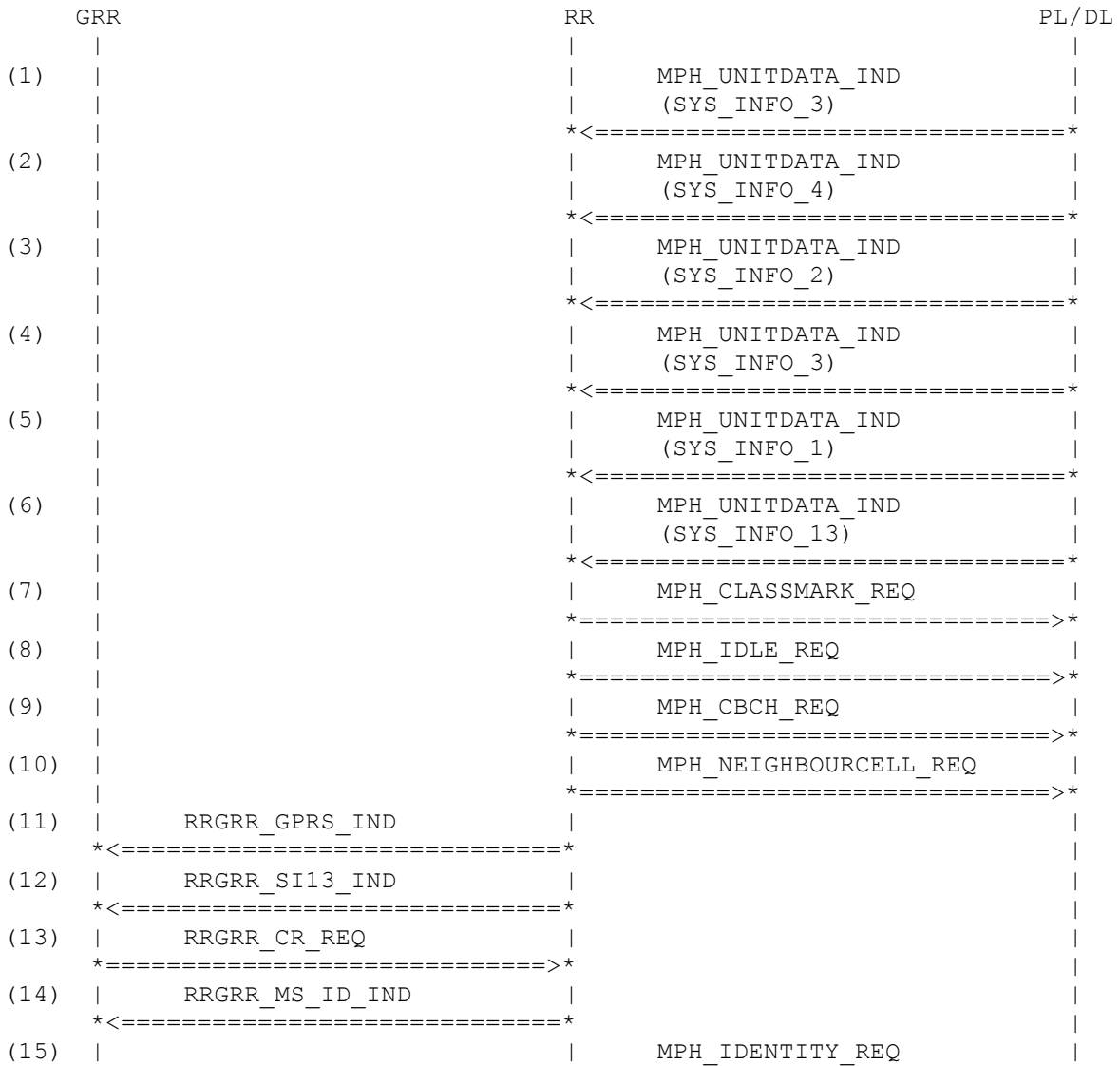
3.8.6 RRG706: RRGRR_RR_EST_IND

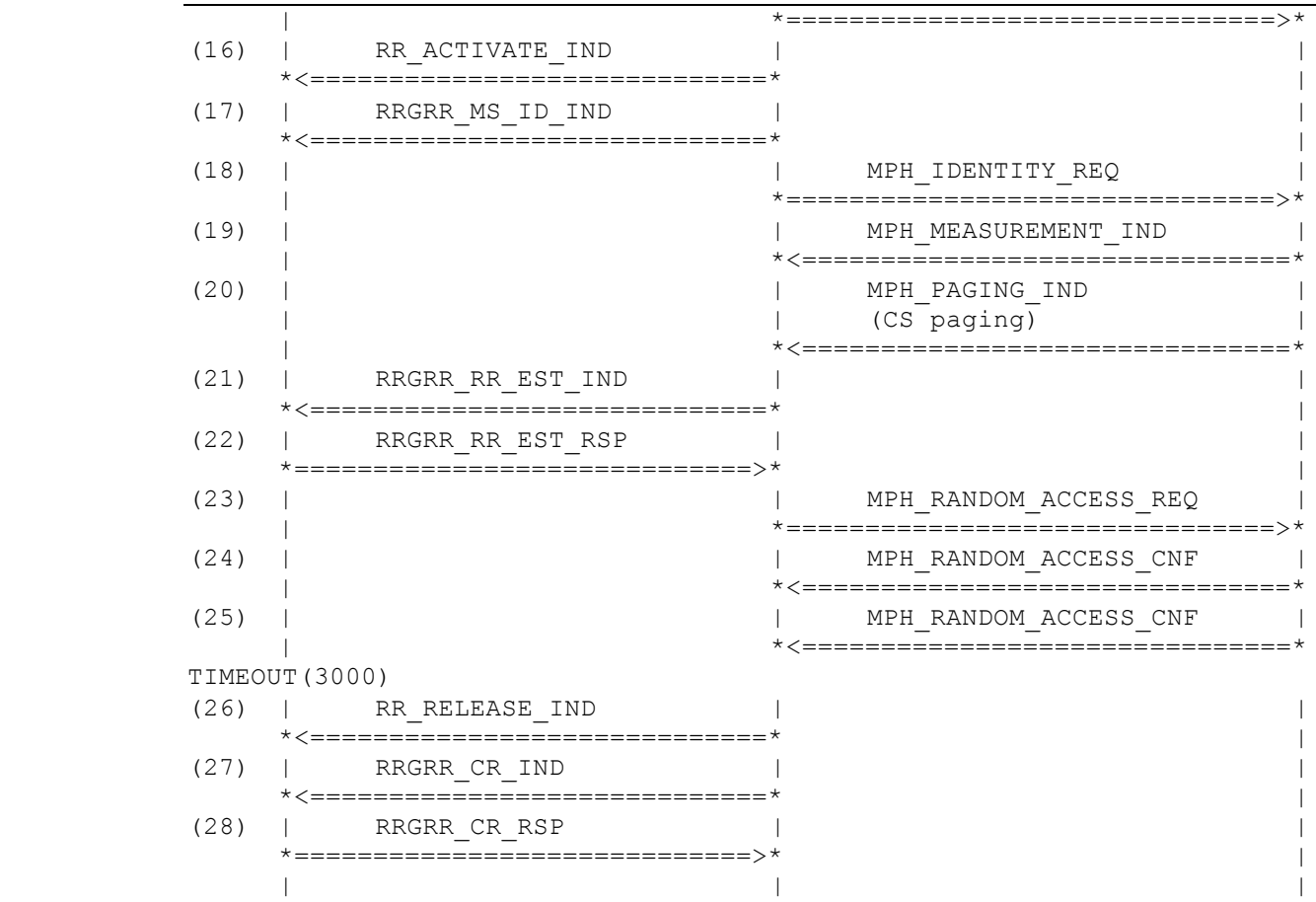
Description:

This primitive indicates to the GRR that an RR connection establishment was received from the network via paging. It is only a trigger.
RR must always ask GRR whether it is allowed to accept paging or not.

Preamble:

RRG251





Parametrization

Primitive	Parameter	Value
(1) MPH_UNITDATA_IND		
	arfcn	ARFCN_124
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(2) MPH_UNITDATA_IND		
	arfcn	ARFCN_124
	fn	fn_780
	sdu	

	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC706
	}	
(3) MPH_UNITDATA_IND		
	arfcn	ARFCN_124
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_2
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(4) MPH_UNITDATA_IND		
	arfcn	ARFCN_124
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3749
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_124
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1

	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_124
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(7) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(8) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_124
	ext_bcch	BSIC_3
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM
	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(9) MPH_CBCH_REQ		
	cbch	CBCH_INACTIVE
(10) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	MPH_NCELL_2
	sync_only	NORMAL_BA
(11) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC706
(12) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC706
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	

(13) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_124
	bsic	BSIC_3
(14) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(15) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(16) RR_ACTIVATE_IND		
	op	OP_MODE_TEST_SIM
	mm_info	MM_INFO_4
	cid	CELL_IDENT_3749
	plmn	PLMN_ID_123
	lac	LAC_2147
	power	RF_CLASS_4
	gprs_indication	GPRS_YES
(17) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(18) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC702
(19) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_124
	rx_lev_full	RX_LEV_35
	rx_lev_sub	RX_LEV_35
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(20) MPH_PAGING_IND		
	identity_type	ID_TMSI
	channel_needed	CN_ANY
(21) RRGRR_RR_EST_IND		
(22) RRGRR_RR_EST_RSP		
	rr_est	RR_EST_ALLOWED
	susp_req	SUSPENSION_REQ_NOT_NEEDED
(23) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC871
(24) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(25) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(26) RR_RELEASE_IND		
	cause	NOT_USED

	sapi	SAPI_0
	gprs_resumption	NOT_USED
(27) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL

History:

21-July-00	MSE	Initial
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

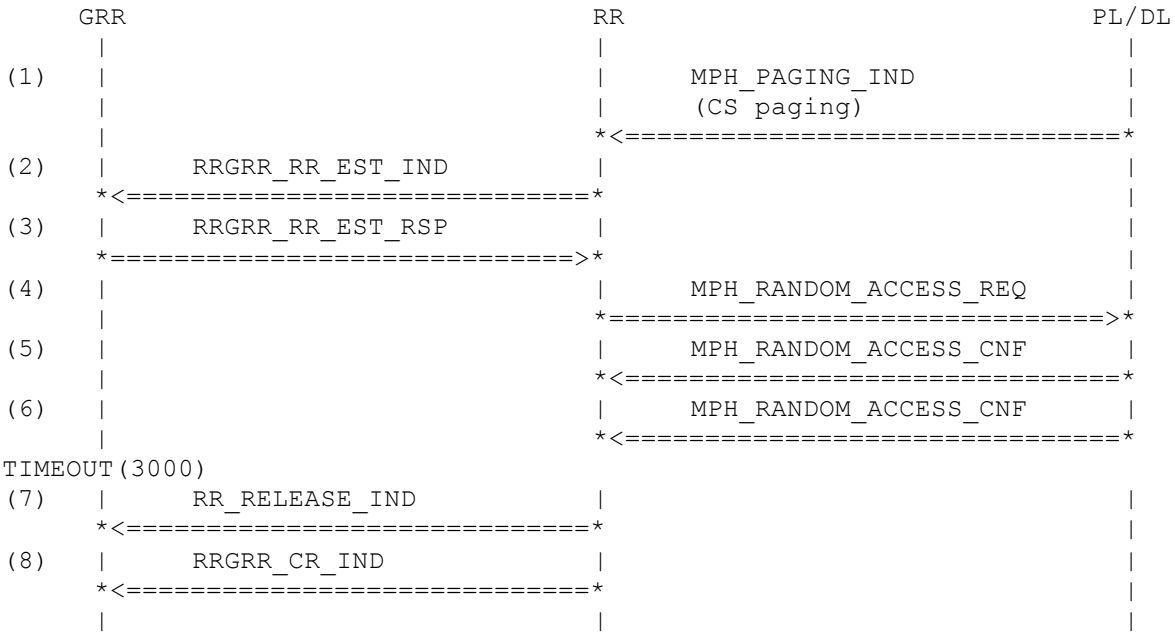
3.8.7 RRG707: RRGRR_RR_EST_IND

Description:

This primitive indicates to the GRR that an RR connection establishment was received from the network via paging. It is only a trigger.
RR must always ask GRR whether it is allowed to accept paging or not.

Preamble:

RRG250



Parametrization

Primitive	Parameter	Value
(1) MPH_PAGING_IND		
	identity_type	ID_TMSI
	channel_needed	CN_ANY
(2) RRGRR_RR_EST_IND		
(3) RRGRR_RR_EST_RSP		
	rr_est	RR_EST_ALLOWED
	susp_req	SUSPENSION_REQ_NOT_NEEDED

(4) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC871
(5) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(6) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(7) RR_RELEASE_IND		
	cause	NOT_USED
	sapi	SAPI_0
	gprs_resumption	NOT_USED
(8) RRGRR_CR_IND		
	cr_type	CR_REQ_CANDIDATE

History:

21-July-00	MSE	Initial
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

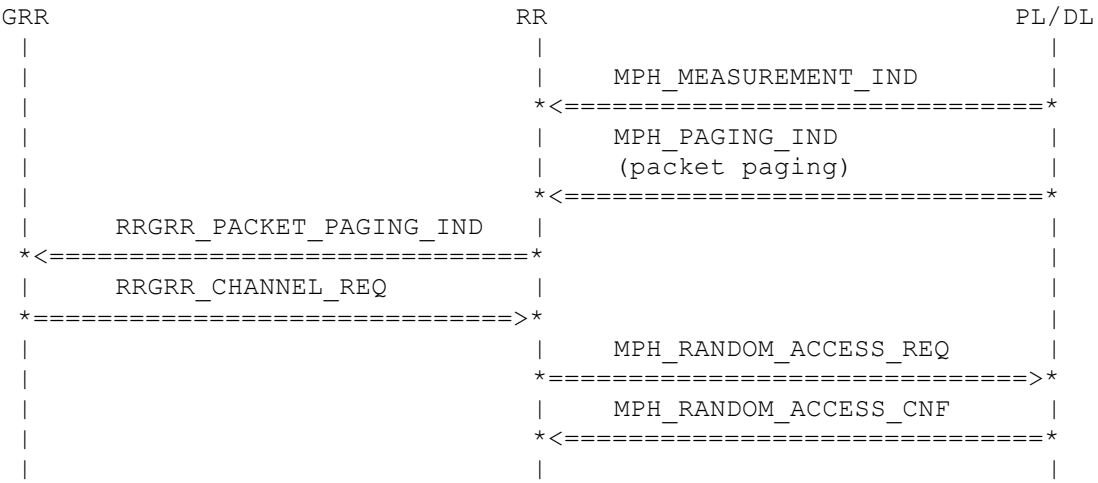
3.8.8 RRG708: RRGRR_PACKET_PAGING_IND

Description:

RR receives an MPH_PAGING_IND and forwards it to GRR. GRR requests the initiation of immediate assignment procedure. RR requests a channel.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(7) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_35
	rx_lev_sub	RX_LEV_35
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED

	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(8) MPH_PAGING_IND		
	identity_type	ID_PTMSI
	channel_needed	CN_PACKET
(9) RRGRR_PACKET_PAGING_IND		
	pg_type	RRGRR_PTMSI
(10) RRGRR_CHANNEL_REQ		
	req_data	ESTCS_PAGING
(11) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC721
(12) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1

History:

08-Aug-01	LG	Initial

3.9 Mobile Originated Calls (800)

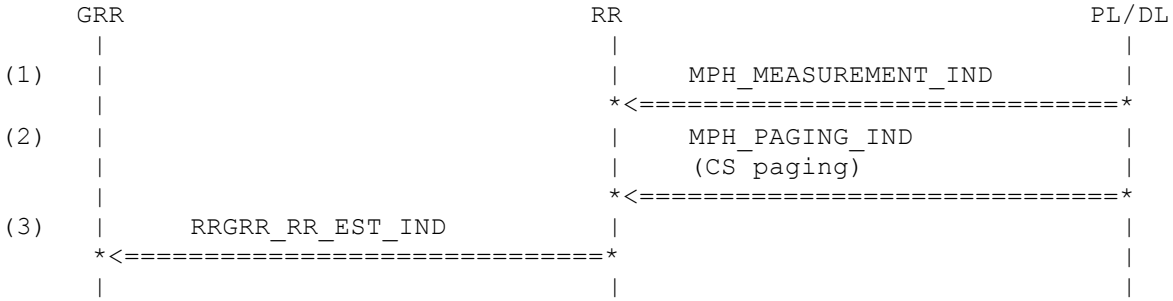
3.9.1 RRG801: RRGRR_RR_EST_IND

Description:

This primitive indicates to the GRR that an RR connection establishment was received from the network via paging. It is only a trigger.
RR must always ask GRR whether it is allowed to accept paging or not.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_35
	rx_lev_sub	RX_LEV_35
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_PAGING_IND		
	identity_type	ID_PTMSI
	channel_needed	CN_ANY
(3) RRGRR_RR_EST_IND		

History:

21-July-00	MSE	Initial
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

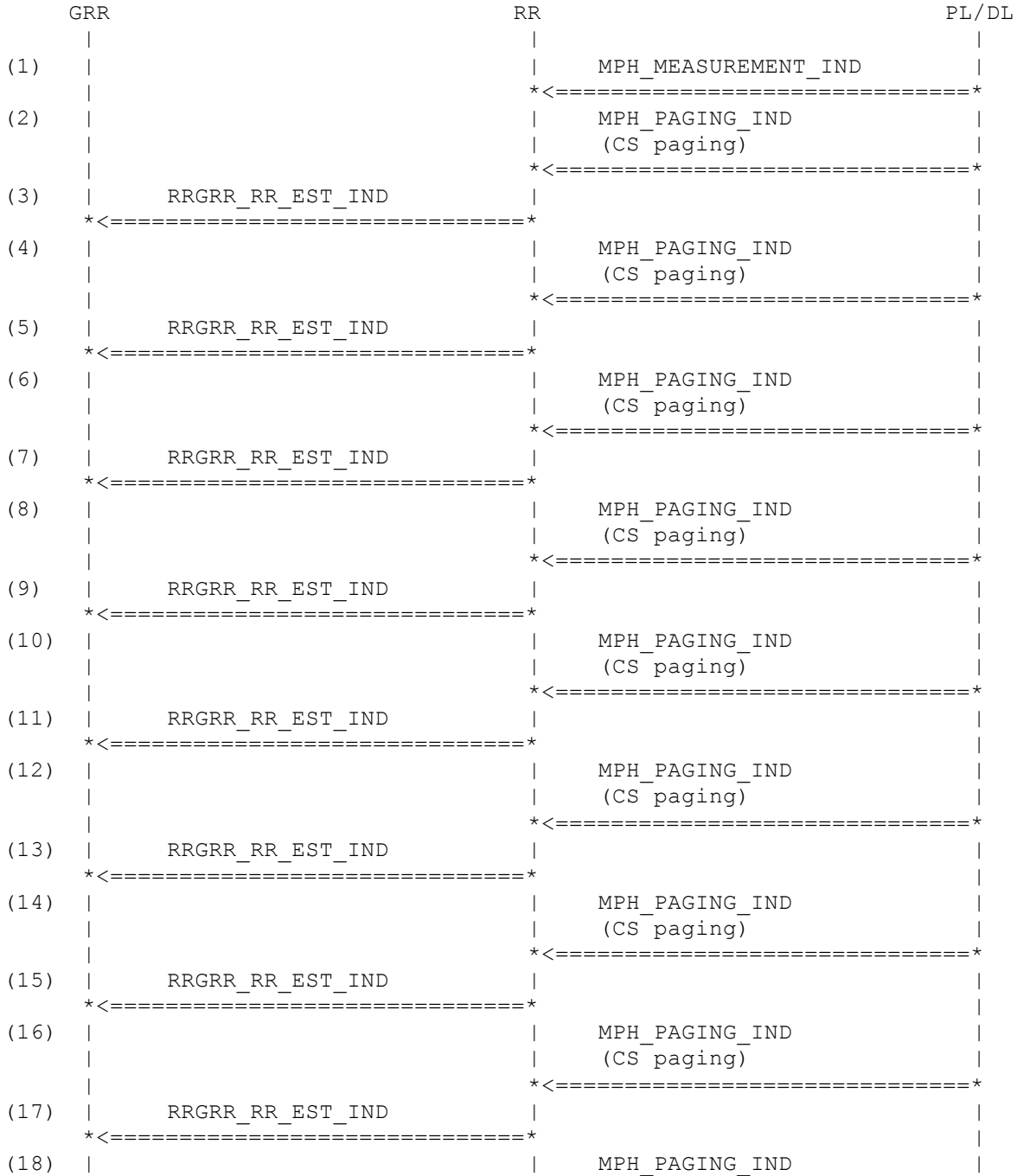
3.9.2 RRG802: RRGRR_RR_EST_IND – many pagings

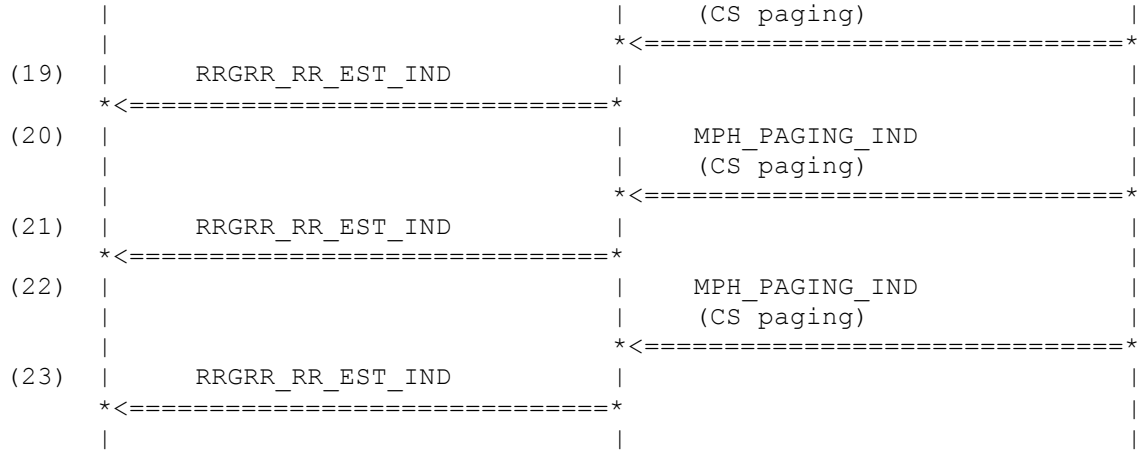
Description:

This primitive indicates to the GRR that an RR connection establishment was received from the network via paging. It is only a trigger.
RR must always ask GRR whether it is allowed to accept paging or not.

Preamble:

RRG102





Parametrization

Primitive	Parameter	Value
(1) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_35
	rx_lev_sub	RX_LEV_35
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_3
	gprs_sync	NORMAL_MEAS_REP
(2) MPH_PAGING_IND		
	identity_type	ID_IMSI
	channel_needed	CN_ANY
(3) RRGRR_RR_EST_IND		
(4) MPH_PAGING_IND		
	identity_type	ID_IMSI
	channel_needed	CN_ANY
(5) RRGRR_RR_EST_IND		
(6) MPH_PAGING_IND		
	identity_type	ID_IMSI
	channel_needed	CN_ANY
(7) RRGRR_RR_EST_IND		
(8) MPH_PAGING_IND		
	identity_type	ID_IMSI
	channel_needed	CN_ANY
(9) RRGRR_RR_EST_IND		
(10) MPH_PAGING_IND		
	identity_type	ID_IMSI
	channel_needed	CN_ANY

(11)	RRGRR_RR_EST_IND		
(12)	MPH_PAGING_IND		
		identity_type	ID_IMSI
		channel_needed	CN_ANY
(13)	RRGRR_RR_EST_IND		
(14)	MPH_PAGING_IND		
		identity_type	ID_IMSI
		channel_needed	CN_ANY
(15)	RRGRR_RR_EST_IND		
(16)	MPH_PAGING_IND		
		identity_type	ID_IMSI
		channel_needed	CN_ANY
(17)	RRGRR_RR_EST_IND		
(18)	MPH_PAGING_IND		
		identity_type	ID_IMSI
		channel_needed	CN_ANY
(19)	RRGRR_RR_EST_IND		
(20)	MPH_PAGING_IND		
		identity_type	ID_IMSI
		channel_needed	CN_ANY
(21)	RRGRR_RR_EST_IND		
(22)	MPH_PAGING_IND		
		identity_type	ID_IMSI
		channel_needed	CN_ANY
(23)	RRGRR_RR_EST_IND		

History:

21-July-00	MSE	Initial
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

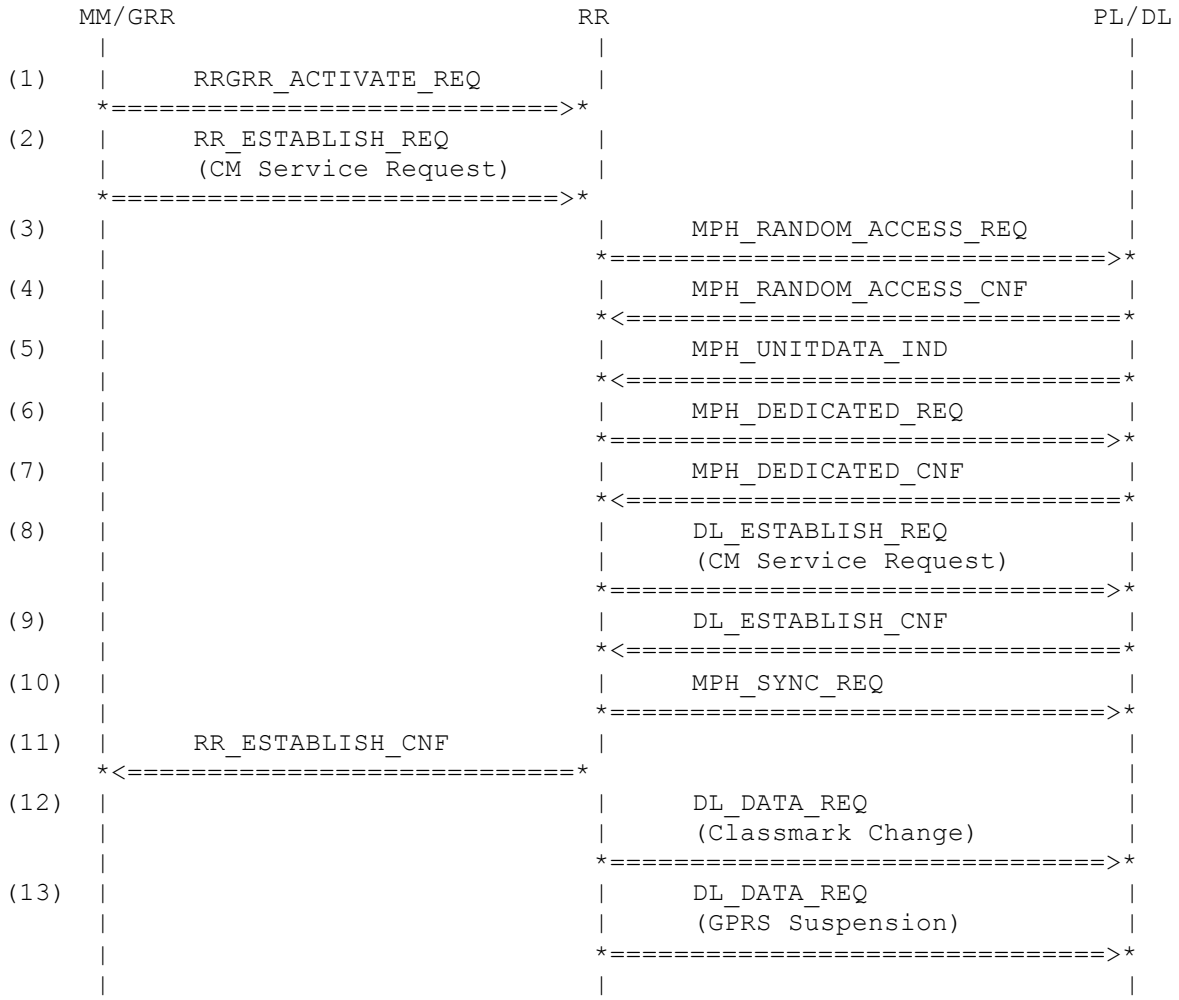
3.9.3 RRG803: RRGRR_ACTIVATE_REQ – MO CS Call – with suspension request

Description:

Mobile originated circuit-switched call
For a class B mobile with an active downlink TBF the GPRS part has to be suspended to make a CS call.
This primitive activates RR. RR acts as if the MS is only GSM service mobile. Monitor CCCH and BCCH.

Preamble:

RRG102



Parametrization

Primitive	Parameter	Value
(1) RRGRR_ACTIVATE_REQ		
	non_gprs	NON_GPRS_EST_REQ_TC860
	susp_req	SUSPENSION_REQ_NEEDED
	rac	RRGRR_RAC_INVALID
(2) RR_ESTABLISH_REQ		
	estcs	
	ESTCS_MOB_ORIG_SPCH_CAL_BY_CC	

	sdu	
	{	
	component	MM
	direction	UPLINK
	pd	U_CM_SERV_REQ
	ti	TI_0
	cm_serv_type	ST_MOC
	ciph_key_num	CKSN_RESERVED
	mob_class_2	MOB_CLASS2
	mob_id	MOB_IDENT_IMSI
	}	
(3) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC803
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_DED
	page_mode	PAGING_NORMAL
	chan_desc	CHANNEL_DESC_SDCCH
	pck_chan_desc	NOT_USED
	req_ref	REQUEST_REFERENCE_5
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_EMPTY
	}	
(6) MPH_DEDICATED_REQ		
	mod	MODE_IMM_ASSIGN
	start	STARTING_TIME_1
	ch_type	PRR_CHANNEL_TYPE_2
	ch_type2	DISPLAY_ONLY
	arfcn	ARFCN_X43
	bsic	BSIC_0
	ho_param	NOT_USED
	tr_para	PRR_TR_PARA_2A
	ciph	NO_CIPHERING
	amr_conf	NOT_USED
(7) MPH_DEDICATED_CNF		
	dedi_res	DEDI_RES_OK
(8) DL_ESTABLISH_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0

	sdu	
	{	
	component	MM
	direction	UPLINK
	pd	U_CM_SERV_REQ
	ti	TI_0
	cm_serv_type	ST_MOC
	ciph_key_num	CKSN_RESERVED
	mob_class_2	MOB_CLASS2
	mob_id	MOB_IDENT_IMSI
	}	
(9) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(10) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(11) RR_ESTABLISH_CNF		
	param	NOT_USED
(12) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_CLASS_CHNG
	ti	TI_0
	mob_class_2	MOB_CLASS2
	mob_class_3	MOB_CLASS3_900
	}	
(13) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_GPRS_SUSP_REQ
	ti	TI_0
	ded_tlli	NOT_USED
	rout_area_id	ROUT_AREA_ID_102
	susp_cause	SUSP_C_CALL
	}	

History:

07-Nov-00	MSE	Initial
13-Feb-01	MSE	+MPH_SYNC_REQ

3.9.4 RRG804: RRGRR_RR_EST_RSP – after many pagings

Description:

This primitive indicates to the RR whether an establishment of RR connection is allowed or not. This primitive is an answer to the RRGRR_RR_EST_IND primitive.

A: connection establishment is allowed; GPRS suspension not needed

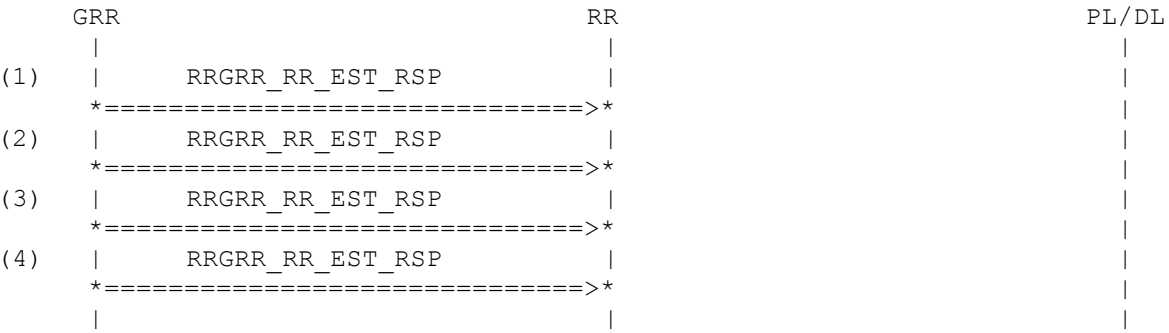
B: connection establishment is allowed; GPRS suspension needed

Preamble:

RRG802

Variants:

<A>....<B>



Parametrization

Primitive	Parameter	Value
(1) RRGRR_RR_EST_RSP		
	rr_est	RR_EST_NOT_ALLOWED
	susp_req	SUSPENSION_REQ_NEEDED
(2) RRGRR_RR_EST_RSP		
	rr_est	RR_EST_NOT_ALLOWED
	susp_req	SUSPENSION_REQ_NEEDED
(3) RRGRR_RR_EST_RSP		
	rr_est	RR_EST_ALLOWED
<A>	susp_req	SUSPENSION_REQ_NOT_NEEDED
<B>	susp_req	SUSPENSION_REQ_NEEDED
(4) RRGRR_RR_EST_RSP		
	rr_est	RR_EST_ALLOWED
<A>	susp_req	SUSPENSION_REQ_NOT_NEEDED
<B>	susp_req	SUSPENSION_REQ_NEEDED

History:

11-Dec-00	MSE	Initial

3.9.5 RRG805: RRGRR_RR_EST_RSP

Description:

This primitive indicates to the RR whether an establishment of RR connection is allowed or not. This primitive is an answer to the RRGRR_RR_EST_IND primitive.

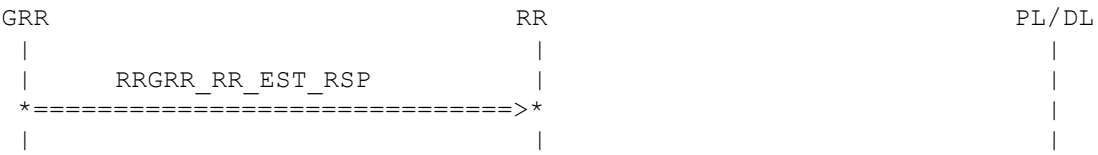
A: connection establishment is allowed; GPRS suspension not needed
B: connection establishment is allowed; GPRS suspension needed
C: connection establishment is not allowed; GPRS suspension unnecessary

Preamble:

RRG801

Variants:

<A>....<C>



Parametrization

Primitive	Parameter	Value
(1) RRGRR_RR_EST_RSP		
<A>	rr_est	RR_EST_ALLOWED
<B>	rr_est	RR_EST_ALLOWED
<C>	rr_est	RR_EST_NOT_ALLOWED
<A>	susp_req	SUSPENSION_REQ_NOT_NEEDED
<B>	susp_req	SUSPENSION_REQ_NEEDED
<C>	susp_req	SUSPENSION_REQ_NOT_NEEDED

History:

21-July-00	MSE	Initial

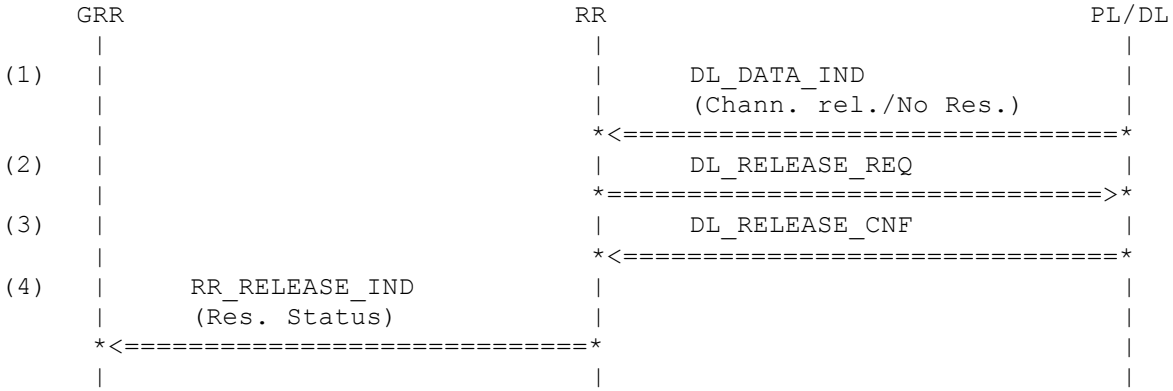
3.9.6 RRG806: No GPRS Resumption after MO CS Call

Description:

unfinished – not fully implemented (16-Oct-00)

Preamble:

RRG803



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_CHAN_REL
	ti	TI_0
	rr_cause	RR_CAUSE_0
	ba_range	NOT_USED
	group_chan_desc	G_CHANNEL_DESC
	group_ckn	NOT_USED
	gprs_resum	RESUMPTION_NO
	ba_list_pref	BA_LIST
	}	
(2) DL_RELEASE_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(3) DL_RELEASE_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(4) RR_RELEASE_IND		
	cause	RRCS_NORM
	sapi	SAPI_0
	gprs_resumption	NOT_USED

History:

07-Nov-00	MSE	Initial
06-Dec-00	MSE	gprs_resumption (RR_RELEASE) added

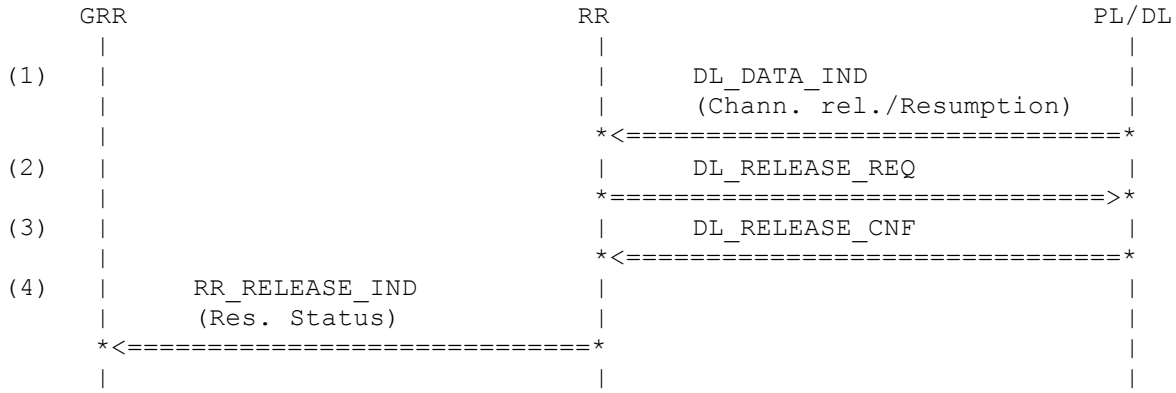
3.9.7 RRG807: GPRS Resumption after MO CS Call

Description:

unfinished – not fully implemented (16-Oct-00)
No Element in RR_RELEASE_IND to provide the resumption status!

Preamble:

RRG803



Parametrization

Primitive	Parameter	Value
(1) DL_DATA_IND		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_CHAN_REL
	ti	TI_0
	rr_cause	RR_CAUSE_0
	ba_range	NOT_USED
	group_chan_desc	G_CHANNEL_DESC
	group_ckn	NOT_USED
	gprs_resum	RESUMPTION_YES
	ba_list_pref	BA_LIST
	}	
(2) DL_RELEASE_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(3) DL_RELEASE_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(4) RR_RELEASE_IND		
	cause	RRCS_NORM
	sapi	SAPI_0
	gprs_resumption	NOT_USED

History:

07-Nov-00	MSE	Initial
06-Dec-00	MSE	gprs_resumption (RR_RELEASE) added

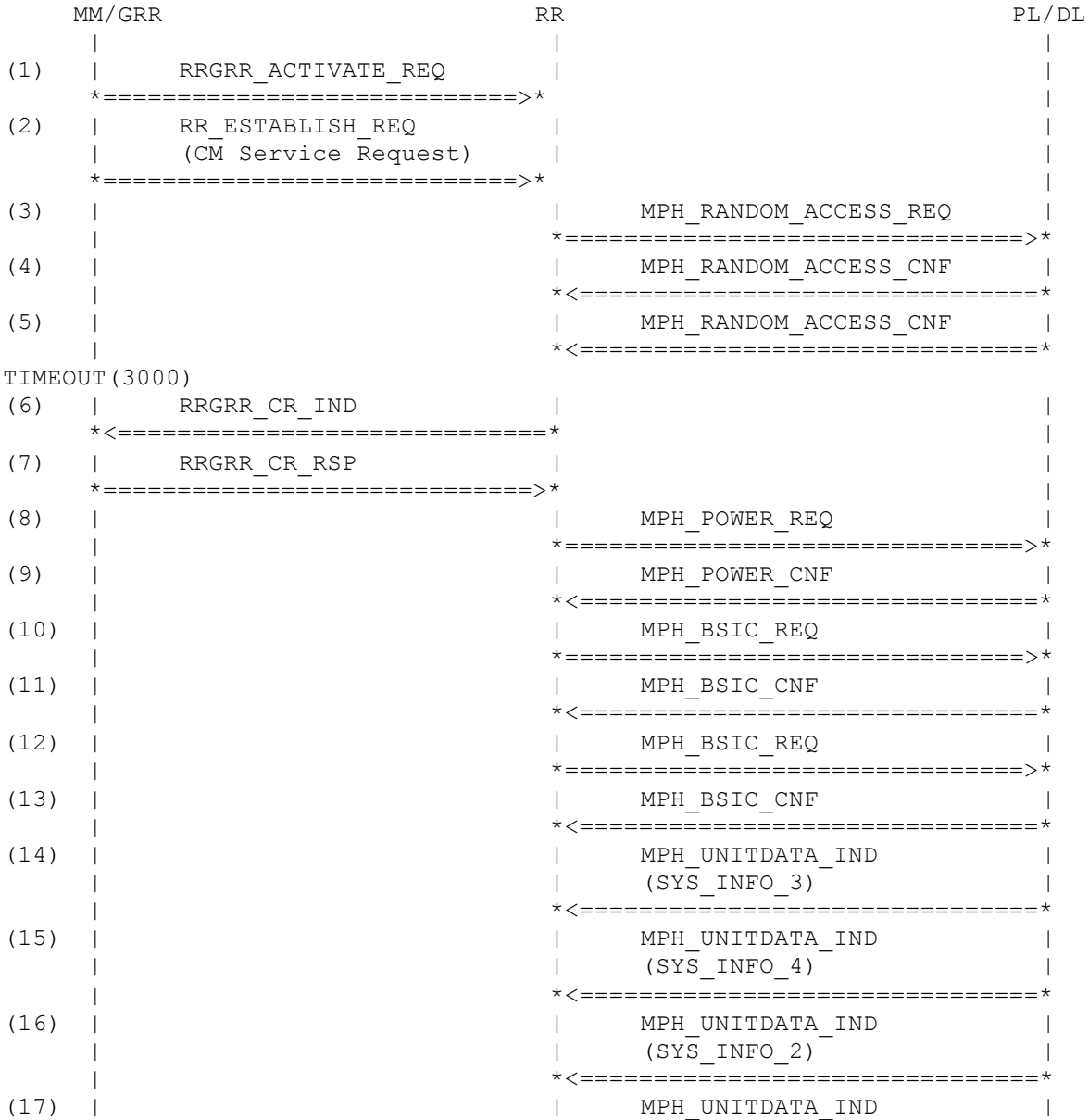
3.9.8 RRG808: RRGRR_ACTIVATE_REQ – MO CS Call – with suspension request

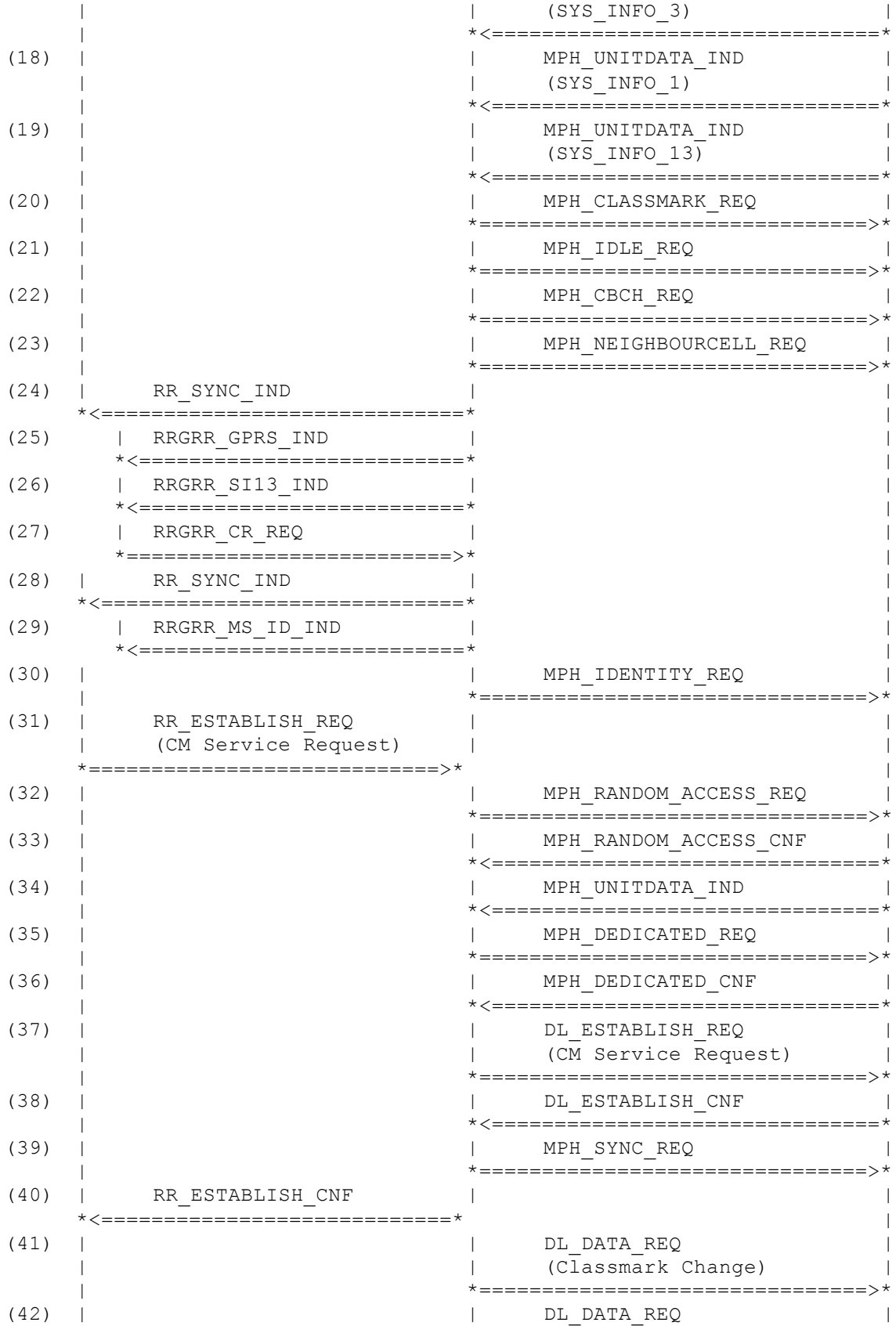
Description:

Mobile originated circuit-switched call
For a class B mobile with an active downlink TBF the GPRS part has to be suspended to make a CS call.
This primitive activates RR. RR acts as if the MS is only GSM service mobile.

Preamble:

RRG102




(SYS_INFO_3)
(18)
MPH_UNITDATA_IND
(SYS_INFO_1)
(19)
MPH_UNITDATA_IND
(SYS_INFO_13)
(20)
MPH_CLASSMARK_REQ
(21)
MPH_IDLE_REQ
(22)
MPH_CBCH_REQ
(23)
MPH_NEIGHBOURCELL_REQ
(24)
RR_SYNC_IND
(25)
RRGRR_GPRS_IND
(26)
RRGRR_SI13_IND
(27)
RRGRR_CR_REQ
(28)
RR_SYNC_IND
(29)
RRGRR_MS_ID_IND
(30)
MPH_IDENTITY_REQ
(31)
RR_ESTABLISH_REQ
(CM Service Request)
(32)
MPH_RANDOM_ACCESS_REQ
(33)
MPH_RANDOM_ACCESS_CNF
(34)
MPH_UNITDATA_IND
(35)
MPH_DEDICATED_REQ
(36)
MPH_DEDICATED_CNF
(37)
DL_ESTABLISH_REQ
(CM Service Request)
(38)
DL_ESTABLISH_CNF
(39)
MPH_SYNC_REQ
(40)
RR_ESTABLISH_CNF
(41)
DL_DATA_REQ
(Classmark Change)
(42)
DL_DATA_REQ

```
      |                                  |      (GPRS Suspension)          |
      |                                  *================================>*
      |                                  |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) RRGRR_ACTIVATE_REQ		
	non_gprs	NON_GPRS_EST_REQ_TC860
	susp_req	SUSPENSION_REQ_NEEDED
	rac	RRGRR_RAC_INVALID
(2) RR_ESTABLISH_REQ		
	estcs	ESTCS_MOB_ORIG_SPCH_CAL_BY_CC
	sdu	
	{	
	component	MM
	direction	UPLINK
	pd	U_CM_SERV_REQ
	ti	TI_0
	cm_serv_type	ST_MOC
	ciph_key_num	CKSN_RESERVED
	mob_class_2	MOB_CLASS2
	mob_id	MOB_IDENT_IMSI
	}	
(3) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC803
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(6) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(7) RRGRR_CR_RSP		
(8) MPH_POWER_REQ		
	pch_interrupt	PCH_INTERRUPT
	freq_bands	STD_DUAL_EXT_0A
(9) MPH_POWER_CNF		
	num_of_chan	CHANNELS_3
	arfcn	ARFCN_43_20_124
	rx_lev	RXLEV_22_21_20
(10) MPH_BSIC_REQ		
	arfcn	ARFCN_X20
(11) MPH_BSIC_CNF		
	arfcn	ARFCN_X20
	bsic	BSIC_5
	cs	CS_NO_BCCH_AVAIL
(12) MPH_BSIC_REQ		
	arfcn	ARFCN_X43

(13) MPH_BSIC_CNF		
	arfcn	ARFCN_X43
	bsic	BSIC_5
	cs	CS_NO_ERROR
(14) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(15) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_4
	ti	TI_0
	loc_area_ident	LOC_AREA_IDENT_123_2147
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	chan_desc	NOT_USED
	mob_alloc	NOT_USED
	si4_rest_oct	SI4_REST_GPRS_TC701
	}	
(16) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_2
	ti	TI_0
	neigh_cell_desc	NCELL_DESC_2
	ncc_permit	NCC_PERMITTED_1
	rach_ctrl	RACH_CTRL_1
	}	
(17) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780

	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_1
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(18) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_1
	ti	TI_0
	cell_chan_desc	CELL_CHAN_DESC_1
	rach_ctrl	RACH_CTRL_1
	si1_rest_oct	SI1_REST_GPRS_TC701
	}	
(19) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(20) MPH_CLASSMARK_REQ		
	classmark	CLASS_MS_DUALBAND
(21) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM

	eotd_avail	EOTD_NOT_AVAIL
	gprs_support	MPH_GPRS_PROCS_USED
(22) MPH_CBCH_REQ		
	cbch	CBCH_INACTIVE
(23) MPH_NEIGHBOURCELL_REQ		
	multi_band	MULTI_BAND_0
	arfcn	MPH_NCELL_2
	sync_only	NORMAL_BA
(24) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	NOT_PRESENT_16BIT
	chm	NOT_USED
(25) RRGRR_GPRS_IND		
	cause	GPRS_SUPPORTED
	serving_cell_info	SERVING_CELL_TC702
(26) RRGRR_SI13_IND		
	si_states	SI_STATES_TC702
	serving_cell_info	SERVING_CELL_TC702
	arfcn	BA_LIST_702
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_13
	ti	TI_0
	si13_rest_oct	SI13_REST_GPRS_TC701_BCCH
	}	
(27) RRGRR_CR_REQ		
	cr_type	CR_COMPLETE
	arfcn	ARFCN_X43
	bsic	BSIC_5
(28) RR_SYNC_IND		
	ciph	NOT_PRESENT_8BIT
	mm_info	NOT_USED
	bcch_info	NOT_USED
	synccs	SYNCCS_IDLE_SELECTION
	chm	NOT_USED
(29) RRGRR_MS_ID_IND		
	tmsi	TMSI_0X142
(30) MPH_IDENTITY_REQ		
	mid	MS_ID_IMSI_HPLMN_TMSI_TC970
(31) RR_ESTABLISH_REQ		
	estcs ESTCS_MOB_ORIG_SPCH_CAL_BY_CC	
	sdu	
	{	
	component	MM
	direction	UPLINK

	pd	U_CM_SERV_REQ
	ti	TI_0
	cm_serv_type	ST_MOC
	ciph_key_num	CKSN_RESERVED
	mob_class_2	MOB_CLASS2
	mob_id	MOB_IDENT_IMSI
	}	
(32) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC803
(33) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(34) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_DED
	page_mode	PAGING_NORMAL
	chan_desc	CHANNEL_DESC_SDCCH
	pck_chan_desc	NOT_USED
	req_ref	REQUEST_REFERENCE_5
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	IA_REST_OCTETS_EMPTY
	}	
(35) MPH_DEDICATED_REQ		
	mod	MODE_IMM_ASSIGN
	start	STARTING_TIME_1
	ch_type	PRR_CHANNEL_TYPE_2
	ch_type2	DISPLAY_ONLY
	arfcn	ARFCN_X43
	bsic	BSIC_0
	ho_param	NOT_USED
	tr_para	PRR_TR_PARA_2A
	ciph	NO_CIPHERING
	amr_conf	NOT_USED
(36) MPH_DEDICATED_CNF		
	dedi_res	DEDI_RES_OK
(37) DL_ESTABLISH_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	MM
	direction	UPLINK

	pd	U_CM_SERV_REQ
	ti	TI_0
	cm_serv_type	ST_MOC
	ciph_key_num	CKSN_RESERVED
	mob_class_2	MOB_CLASS2
	mob_id	MOB_IDENT_IMSI
	}	
(38) DL_ESTABLISH_CNF		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
(39) MPH_SYNC_REQ		
	cs	CS_CLEAN_SYS_INFO
(40) RR_ESTABLISH_CNF		
	param	NOT_USED
(41) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_CLASS_CHNG
	ti	TI_0
	mob_class_2	MOB_CLASS2
	mob_class_3	MOB_CLASS3_900
	}	
(42) DL_DATA_REQ		
	ch_type	CH_TYPE_SDCCH
	sapi	SAPI_0
	sdu	
	{	
	component	RR
	direction	UPLINK
	pd	U_GPRS_SUSP_REQ
	ti	TI_0
	ded_tlli	NOT_USED
	rout_area_id	ROUT_AREA_ID_102
	susp_cause	SUSP_C_CALL
	}	

History:

3.10 25-Jan-03 MPA Initial Packet Transfer Mode (900)

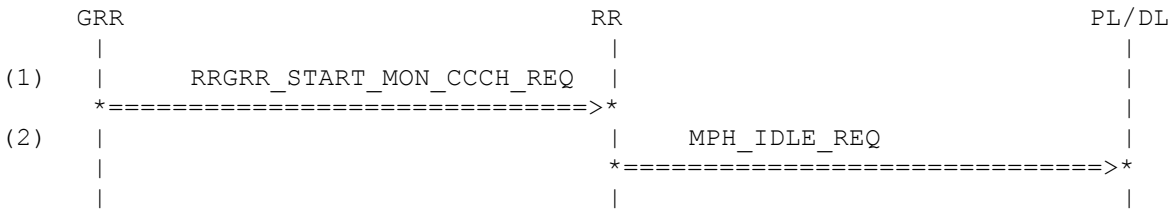
3.10.1 RRG901: Going from PTM to PIM

Description:

GRR leaves PTM, RR enters idle mode on BCCH

Preamble:

RRG401



Parametrization

Primitive	Parameter	Value
(1) RRGRR_START_MON_CCCH_REQ		
	pag_mode	PAG_MODE_REORG
	split_pg	NOT_USED
(2) MPH_IDLE_REQ		
	mod	MODE_CELL_SELECTION
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	REORG_ONLY
	eotd_avail	NOT_USED
	gprs_support	MPH_GPRS_PROCS_USED

History:

	26-Mar-03	MPA	Initial

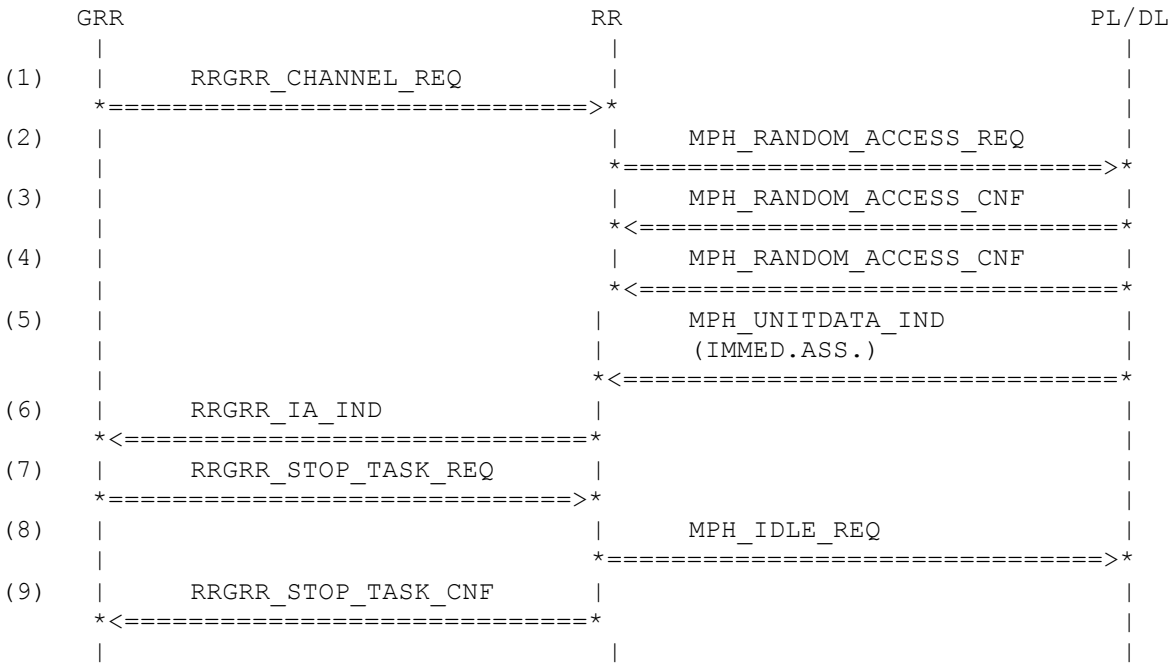
3.10.2 RRG902: Going from PTM to PAM

Description:

GRR is transfer mode and due to an error condition requests the establishment of a new TBF immediatly.

Preamble:

RRG401



Parametrization

Primitive	Parameter	Value
(1) RRGRR_CHANNEL_REQ		
	req_data	Ch_Req_Data_DEF
(2) MPH_RANDOM_ACCESS_REQ		
	send_mode	SEND_MODE_2_BURSTS_TC730
(3) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(4) MPH_RANDOM_ACCESS_CNF		
	frame_no	FRAME_NUMBER_1
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X43
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK

	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	
	IA_REST_OCTETS_UL_ASSIGN_TMA1	
	}	
(6) RRGRR_IA_IND		
	fn	fn_780
	r_bit	CHAN_REQ_SENT_MORE
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_IMM_ASSIGN
	ti	TI_0
	tma	TMA_0
	dl	DL_0
	d_t	D_T_TBF
	page_mode	PAGING_NORMAL
	chan_desc	NOT_USED
	pck_chan_desc	PACKET_CHANNEL_DESC
	req_ref	REQUEST_REFERENCE_4
	time_advance	TIMING_ADVANCE_30
	mob_alloc	MOBILE_ALLOCATION_1
	start_time	START_TIME_1
	ia_rest_oct	
	IA_REST_OCTETS_UL_ASSIGN_TMA1	
	}	
(7) RRGRR_STOP_TASK_REQ		
	v_stop_ccch	STOP_CCCH
(8) MPH_IDLE_REQ		
	mod	MODE_PACKET_TRANSFER
	arfcn	ARFCN_X43
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_1
	reorg_only	NORMAL_PGM

	eotd_avail	NOT_USED
	gprs_support	NOT_PRESENT_8BIT

(9) RRGRR_STOP_TASK_CNF

History:

26-Mar-03	MPA	Initial

3.11 GPRS Cell Reselection Procedures on CCCH

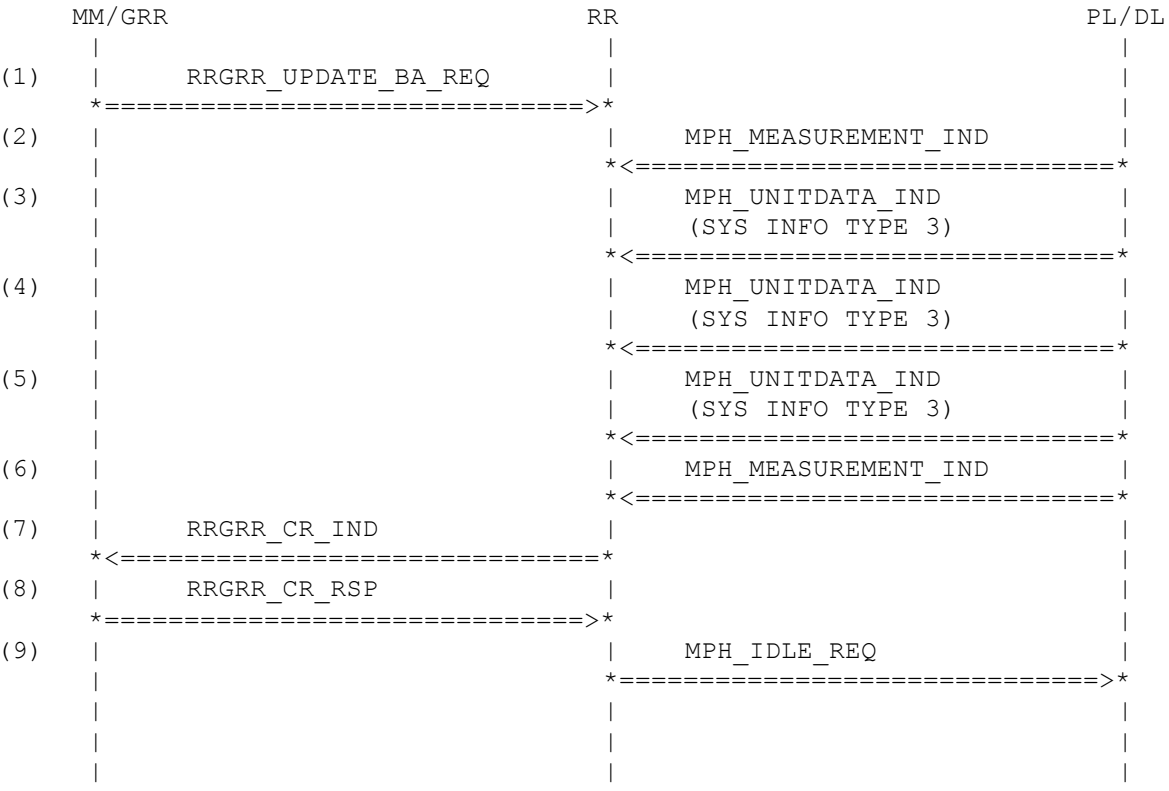
3.11.1 RRG904: Cell Reseceltion to cell with highest C32 value among cells of highest Priority Class

Description:

Cell with Highest C32 value chosen, among cells of highest priority class and which satisfy the condition C31 >= 0

Preamble:

RRG201



Parametrization

Primitive	Parameter	Value
(1) RRGRR_UPDATE_BA_REQ		
	cell_type	SYNC_CELL
	add_freq_list	BA_LIST_ADDED_1
	rm_freq_list	BA_LIST_REMOVED_1

(2) MPH_MEASUREMENT_IND

arfcn	ARFCN_X43
rx_lev_full	RX_LEV_20
rx_lev_sub	RX_LEV_20
rx_qual_full	RX_QUAL_1
rx_qual_sub	RX_QUAL_1
dtx	DTX_NOT_USED
otd	TIME_ADV_27
valid	TRUE
fn_offset	FN_OFFSET_1_SEC
ncells	NCELLS_4
gprs_sync	NORMAL_MEAS_REP

(3) MPH_UNITDATA_IND

arfcn	ARFCN_X30
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_SYS_INFO_3
ti	TI_0
cell_ident	CELL_IDENT_3748
loc_area_ident	LOC_AREA_IDENT_123_2147
ctrl_chan_desc	CTRL_CHAN_DESC_1
cell_opt_bcch	CELL_OPT_BCCH_1
cell_select	CELL_SELECT_1
rach_ctrl	RACH_CTRL_3
si3_rest_oct	SI3_REST_GPRS_TC701
}	

(4) MPH_UNITDATA_IND

arfcn	ARFCN_X40
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK
pd	D_SYS_INFO_3
ti	TI_0
cell_ident	CELL_IDENT_3748
loc_area_ident	LOC_AREA_IDENT_123_2147
ctrl_chan_desc	CTRL_CHAN_DESC_1
cell_opt_bcch	CELL_OPT_BCCH_1
cell_select	CELL_SELECT_1
rach_ctrl	RACH_CTRL_3
si3_rest_oct	SI3_REST_GPRS_TC701
}	

(5) MPH_UNITDATA_IND

arfcn	ARFCN_X20
fn	fn_780
sdu	
{	
component	RR
direction	DOWNLINK

	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_3
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(6) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27
	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_4
	gprs_sync	NORMAL_MEAS_REP
(7) RRGRR_CR_IND		
	cr_type	CR_ABNORMAL
(8) RRGRR_CR_RSP		
(9) MPH_IDLE_REQ		
	mod	MODE_CELL_RESELECTION
	arfcn	ARFCN_X40
	ext_bcch	BSIC_5
	comb_ccch	CCD_CCCH_1_NOT_COMB
	tn	TN_0
	dlt	DLT_23
	pg	PG_0
	bs_ag_blocks_res	BS_AG_BLKS_RES_5
	bs_pa_mfrms	BS_PA_MFRMS_2
	power	MS_TXPWR_MAX_CCH_02
	ncc_permitted	NCC_PERMITTED_FF
	reorg_only	NORMAL_PGM
	eotd_avail	NOT_USED
	gprs_support	NOT_PRESENT_8BIT

History:

04-July-97	DL	Initial
04-Dec-2000	MSE	132 adapted
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync

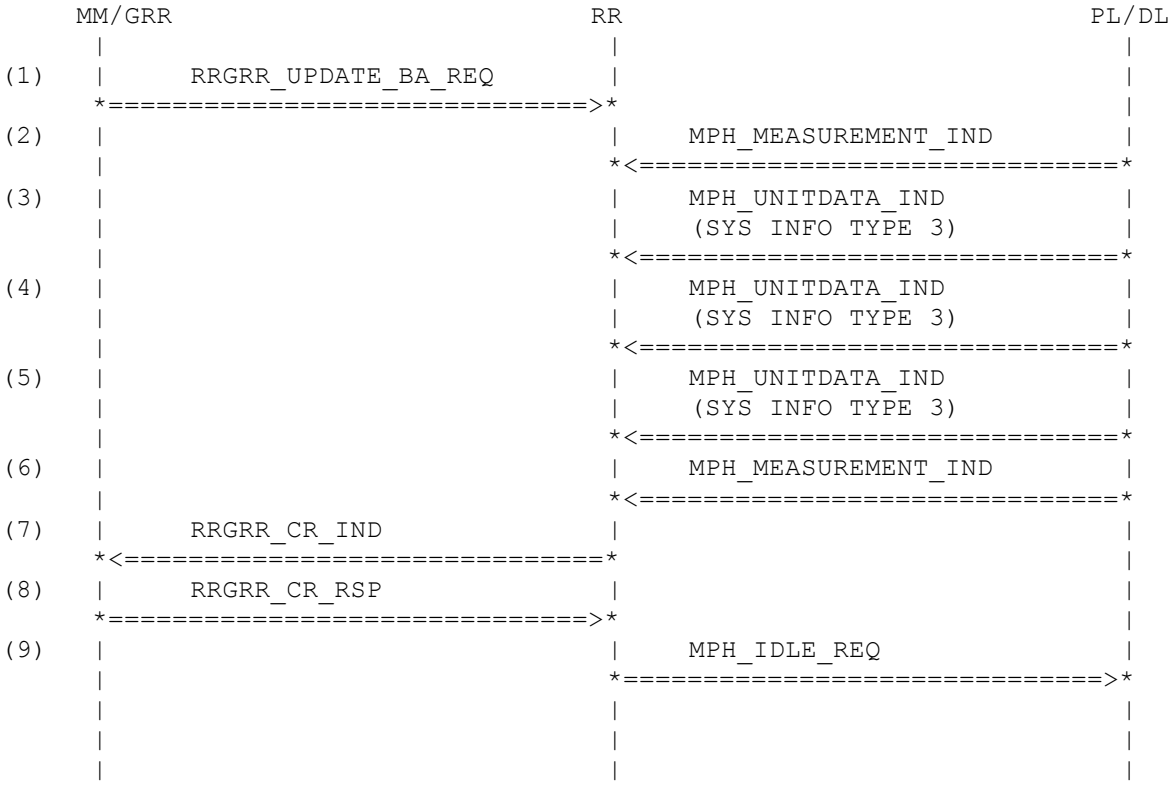
3.11.2 RRG905: Cell Reseceltion to cell with highest C32 value among all cells if no cell with C31>= 0

Description:

Cell with Highest C32 value chosen, among cells of highest priority class and which satisfy the condition C31 >= 0

Preamble:

RRG201



Parametrization

Primitive	Parameter	Value
(2) RRGRR_UPDATE_BA_REQ		
	cell_type	SYNC_CELL
	add_freq_list	BA_LIST_ADDED_2
	rm_freq_list	BA_LIST_REMOVED_1
(3) MPH_MEASUREMENT_IND		
	arfcn	ARFCN_X43
	rx_lev_full	RX_LEV_20
	rx_lev_sub	RX_LEV_20
	rx_qual_full	RX_QUAL_1
	rx_qual_sub	RX_QUAL_1
	dtx	DTX_NOT_USED
	otd	TIME_ADV_27

	valid	TRUE
	fn_offset	FN_OFFSET_1_SEC
	ncells	NCELLS_4
	gprs_sync	NORMAL_MEAS_REP
(4) MPH_UNITDATA_IND		
	arfcn	ARFCN_X30
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_3
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(5) MPH_UNITDATA_IND		
	arfcn	ARFCN_X40
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_3
	si3_rest_oct	SI3_REST_GPRS_TC701
	}	
(6) MPH_UNITDATA_IND		
	arfcn	ARFCN_X20
	fn	fn_780
	sdu	
	{	
	component	RR
	direction	DOWNLINK
	pd	D_SYS_INFO_3
	ti	TI_0
	cell_ident	CELL_IDENT_3748
	loc_area_ident	LOC_AREA_IDENT_123_2147
	ctrl_chan_desc	CTRL_CHAN_DESC_1
	cell_opt_bcch	CELL_OPT_BCCH_1
	cell_select	CELL_SELECT_1
	rach_ctrl	RACH_CTRL_3

	si3_rest_oct	SI3_REST_GPRS_TC701
	}	

(7) MPH_MEASUREMENT_IND

arfcn	ARFCN_X43
rx_lev_full	RX_LEV_20
rx_lev_sub	RX_LEV_20
rx_qual_full	RX_QUAL_1
rx_qual_sub	RX_QUAL_1
dtx	DTX_NOT_USED
otd	TIME_ADV_27
valid	TRUE
fn_offset	FN_OFFSET_1_SEC
ncells	NCELLS_4
gprs_sync	NORMAL_MEAS_REP

(8) RRGRR_CR_IND

cr_type	CR_ABNORMAL

(9) RRGRR_CR_RSP

(10) MPH_IDLE_REQ

mod	MODE_CELL_RESELECTION
arfcn	ARFCN_X40
ext_bcch	BSIC_5
comb_ccch	CCD_CCCH_1_NOT_COMB
tn	TN_0
dlt	DLT_23
pg	PG_0
bs_ag_blocks_res	BS_AG_BLKS_RES_5
bs_pa_mfrms	BS_PA_MFRMS_2
power	MS_TXPWR_MAX_CCH_02
ncc_permitted	NCC_PERMITTED_FF
reorg_only	NORMAL_PGM
eotd_avail	NOT_USED
gprs_support	NOT_PRESENT_8BIT

History:

04-July-97	DL	Initial
04-Dec-2000	MSE	132 adapted
16-Jul-01	MSE	MPH_MEASUREMENT_IND: +gprs_sync