



Technical Document – Confidential

GSM PROTOCOL STACK

MULTILAYER TEST SPECIFICATION

SUPPLEMENTARY SERVICES

Document Number:	8403.403.00.104
Version:	0.6
Status:	Draft
Approval Authority:	
Creation Date:	1997-Jul-09
Last changed:	2015-Mar-08 by XGUTTEFE
File Name:	mss.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third-party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, including photocopying and recording, for any purpose without the express written permission of TI.

Change History

Date	Changed by	Approved by	Version	Status	Notes
1997-Jul-09	Stefan Lemke et al		0.1		1
2000-Sep-12	Bernd Sabborrosch		0.2		2
2001-Aug-23	SBK		0.3		3
2001-Dez-20	SBK		0.4		4
2002-Jun-26	HM		0.5		5
2003-May-20	XGUTTEFE		0.6	Draft	

Note s:

1. Initial version
2. Update CC parts of tests, add testing of +CCWV indication
3. Aligned to new behaviour of ATDxxx
4. Adapted to new race condition of OK and RACH
5. Adapted to new CC behaviour in simulation

Table of Contents

1.1	References	6
1.2	Abbreviations	9
1.3	Terms	11
2	Overview	11
3	Parameters	12
4	TEST CASES	70
4.1	Preambles	70
4.1.1	MSS001: Power On	70
4.1.2	MSS002: Power On (Reestablishment allowed)	71
4.2	Call Forwarding Tests	72
4.2.1	MSS100: Call Forwarding, Registration accepted (31.2.1.1.1)	72
4.3	Advice of Charge Tests	74
4.3.1	MSS200: AoC time related charging, MOC, k=1 (31.6.1.1)	74
4.3.2	MSS201: AoC time related charging, MOC, k=2 (31.6.1.1)	76
4.3.3	MSS202: AoC time related charging, MOC, k=3 (31.6.1.1)	79
4.3.4	MSS203: AoC time related charging, MOC, k=4 (31.6.1.1)	81
4.3.5	MSS204: AoC time related charging, MOC, k=5 (31.6.1.1)	83
4.3.6	MSS205: AoC time related charging, MTC, k=1 (31.6.1.2)	85
4.3.7	MSS206: AoC time related charging, MTC, k=2 (31.6.1.2)	86
4.3.8	MSS207: AoC time related charging, MTC, k=3 (31.6.1.2)	87
4.3.9	MSS208: AoC time related charging, MTC, k=4 (31.6.1.2)	88
4.3.10	MSS209: AoC time related charging, MTC, k=5 (31.6.1.2)	89
4.3.11	MSS210: Change in Charging Information during a call (31.6.1.5)	90
4.3.12	MSS211: Different formats of Charging Information, part 1 (31.6.1.6)	93
4.3.13	MSS212: Different formats of Charging Information, part 2 (31.6.1.6)	96
4.3.14	MSS213: Different formats of Charging Information, part 3 (31.6.1.6)	99
4.3.15	MSS214: AoC on a Call Hold Call (31.6.1.7)	102
4.3.16	MSS215: AoC on a Multiparty Call (31.6.1.8)	104
4.3.17	MSS216: Removal of SIM during an active Call (31.6.2.1)	106
4.3.18	MSS217: Interruption of power supply during an active call (31.6.2.2)	107
4.3.19	MSS218: MS going out of coverage during an active AoCC Call (31.6.2.3)	108
4.3.20	MSS219: ACMmax operation / Mobile originating (31.6.2.4)	110
4.3.21	MSS220: ACMmax operation / Mobile Terminating (31.6.2.5)	112
4.3.22	MSS221: Test of several AT-Commands for AoCC	116
4.3.23	MSS300: Call Barring, Deactivation accepted	119
Appendices		120
A.	Acronyms	120
B.	Glossary	120

List of Figures and Tables

List of References

[ISO 9000:2000]

International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 References

- [1] GSM 2.81, Line Identification Supplementary Services - Stage 1
ETS 300 514, ETSI, September 1994
- [2] GSM 2.82, Call Forwarding Supplementary Services - Stage 1
ETS 300 515, ETSI, September 1994
- [3] GSM 2.83, Call Waiting and Call Hold Supplementary Services - Stage 1
ETS 300 516, ETSI, September 1994
- [4] GSM 2.84, Multi Party Supplementary Services - Stage 1
ETS 300 517, ETSI, September 1994
- [5] GSM 2.85, Closed User Group Supplementary Services - Stage 1
ETS 300 518, ETSI, September 1994
- [6] GSM 2.86, Advice of Charge Supplementary Services - Stage 1
ETS 300 519, ETSI, September 1994
- [7] GSM 2.88, Call Barring Supplementary Services - Stage 1
ETS 300 520, ETSI, September 1994
- [8] GSM 3.14, Support of Dual Tone Multi Frequency Signalling via the GSM System
ETS 300 532, ETSI, April 1994
- [9] GSM 3.40, Technical Realization of the Short Message Service Point-to-Point
ETS 300 536, ETSI, January 1996
- [10] GSM 3.41, Technical Realization of Short Message Service Cell Broadcast
ETS 300 537, ETSI, June 1995
- [11] GSM 3.81, Line Identification Supplementary Services - Stage 2
ETS 300 542, ETSI, February 1995
- [12] GSM 3.82, Call Forwarding Supplementary Services - Stage 2
ETS 300 543, ETSI, February 1995
- [13] GSM 3.83, Call Waiting and Call Hold Supplementary Services - Stage 2
ETS 300 544, ETSI, November 1994
- [14] GSM 3.84, Multi Party Supplementary Services - Stage 2
ETS 300 545, ETSI, November 1994
- [15] GSM 3.85, Closed User Group Supplementary Services - Stage 2
ETS 300 546, ETSI, January 1996
- [16] GSM 3.86, Advice of Charge Supplementary Services - Stage 2
ETS 300 547, ETSI, March 1995
- [17] GSM 3.88, Call Barring Supplementary Services - Stage 2
ETS 300 548, ETSI, November 1994
- [18] GSM 4.01, MS-BSS Interface General Aspects and Principles
ETS 300 550, ETSI, September 1994
- [18a] GSM 4.03, MS-BSS Interface Channel Structures and Access Capabilities
ETS 300 552, ETSI, September 1994
- [19] GSM 4.05, Data Link Layer General Aspects
ETS 300 554, ETSI, September 1994
- [20] GSM 4.06, MS-BSS Interface Data Link Layer Specification
ETS 300 555, ETSI, September 1994
- [21] GSM 4.07, Mobile Radio Interface Signalling Layer 3 General Aspects
ETS 300 556, ETSI, February 1995

- [22] GSM 4.08, Mobile Radio Interface Layer 3 Specification
ETS 300 557, ETSI, January 1996
- [23] GSM 4.10, Mobile Radio Interface Layer 3 Supplementary Services Specification
General Aspects
ETS 300 558, ETSI, February 1995
- [24] GSM 4.11, Point-to-Point Short Message Service Support on Mobile Radio Interface
ETS 300 559, ETSI, October 1995
- [25] GSM 4.12, Short Message Service Cell Broadcast Support on Mobile Radio Interface
ETS 300 560, ETSI, January 1996
- [26] GSM 4.80, Mobile Radio Interface Supplementary Services Specification Formats and Coding
ETS 300 564, ETSI, February 1995
- [27] GSM 4.81, Line Identification Supplementary Services - Stage 3
ETS 300 565, ETSI, February 1995
- [28] GSM 4.82, Call Forwarding Supplementary Services - Stage 3
ETS 300 566, ETSI, February 1995
- [29] GSM 4.83, Call Waiting and Call Hold Supplementary Services - Stage 3
ETS 300 567, ETSI, February 1995
- [30] GSM 4.84, Multi Party Supplementary Services - Stage 3
ETS 300 568, ETSI, February 1995
- [31] GSM 4.85, Closed User Group Supplementary Services - Stage 3
ETS 300 569, ETSI, February 1995
- [32] GSM 4.86, Advice of Charge Supplementary Services - Stage 3
ETS 300 570, ETSI, February 1995
- [33] GSM 4.88, Call Barring Supplementary Services - Stage 3
ETS 300 571, ETSI, February 1995
- [34] GSM 5.01, Physical Layer on the Radio Path General Description
ETS 300 573, ETSI, October 1995
- [35] GSM 5.02, Multiplexing and Multiple Access on the Radio Path
ETS 300 574, ETSI, January 1996
- [36] GSM 5.08, Radio Sub-system Link Control
ETS 300 578, ETSI, January 1996
- [37] GSM 5.10, Radio Sub-system Synchronisation
ETS 300 579, ETSI, October 1995
- [38] Service Access Point MMREG
6147.100.96.100; Condat GmbH
- [39] Service Access Point MNCC
6147.101.96.100; Condat GmbH
- [40] Service Access Point MNSS
6147.102.96.100; Condat GmbH
- [41] Service Access Point MNSMS
6147.103.96.100; Condat GmbH
- [42] Service Access Point MMCC
6147.104.97.100; Condat GmbH
- [43] Service Access Point MMSS
6147.105.97.100; Condat GmbH
- [44] Service Access Point MMSMS
6147.106.97.100; Condat GmbH

[45]	Service Access Point RR 6147.107.97.100; Condat GmbH
[46]	Service Access Point SIM 6147.108.97.100; Condat GmbH
[47]	Service Access Point MPH 6147.109.96.100; Condat GmbH
[48]	Service Access Point DL 6147.110.96.100; Condat GmbH
[49]	Service Access Point MDL 6147.111.96.100; Condat GmbH
[50]	Service Access Point PH 6147.112.97.100; Condat GmbH
[51]	Service Access Point MMI 6147.113.96.100; Condat GmbH
[52]	Message Sequence Charts CC 6147.200.97.100; Condat GmbH
[53]	Message Sequence Charts SS 6147.201.97.100; Condat GmbH
[54]	Message Sequence Charts SMS 6147.202.97.100; Condat GmbH
[55]	Message Sequence Charts MM 6147.203.97.100; Condat GmbH
[56]	Message Sequence Charts RR 6147.204.96.100; Condat GmbH
[57]	Message Sequence Charts DL 6147.205.96.100; Condat GmbH
[58]	Users Guide 6147.300.96.100; Condat GmbH
[59]	Test Specification CC 6147.400.97.100; Condat GmbH
[60]	Test Specification SS 6147.401.97.100; Condat GmbH
[61]	Test Specification SMS 6147.402.97.100; Condat GmbH
[62]	Test Specification MM 6147.403.97.100; Condat GmbH
[63]	Test Specification RR 6147.404.97.100; Condat GmbH
[64]	Test Specification DL 6147.405.97.100; Condat GmbH
[65]	Test Specification CCD 6147.406.97.100; Condat GmbH
[66]	SDL Specification CC 6147.500.97.100; Condat GmbH
[67]	SDL Specification SS 6147.501.97.100; Condat GmbH

[68]	SDL Specification SMS 6147.502.97.100; Condat GmbH
[69]	SDL Specification MM 6147.503.97.100; Condat GmbH
[70]	SDL Specification RR 6147.504.97.100; Condat GmbH
[71]	SDL Specification DL 6147.505.97.100; Condat GmbH
[72]	Message Specification CC 6147.600.97.100; Condat GmbH
[73]	Message Specification SS 6147.601.97.100; Condat GmbH
[74]	Message Specification SMS 6147.602.97.100; Condat GmbH
[75]	Message Specification MM 6147.603.97.100; Condat GmbH
[76]	Message Specification RR 6147.604.97.100; Condat GmbH
[77]	Message Specification DL 6147.605.97.100; Condat GmbH
[78]	Technical Documentation CC 6147.700.97.100; Condat GmbH
[79]	Technical Documentation SS 6147.701.97.100; Condat GmbH
[80]	Technical Documentation SMS 6147.702.97.100; Condat GmbH
[81]	Technical Documentation MM 6147.703.97.100; Condat GmbH
[82]	Technical Documentation RR 6147.704.97.100; Condat GmbH
[83]	Technical Documentation DL 6147.705.97.100; Condat GmbH
[84]	Technical Documentation CCD 6147.706.97.100; Condat GmbH

1.2 Abbreviations

AGCH	Access Grant Channel
BCCH	Broadcast Control Channel
BS	Base Station
BSIC	Base Station Identification Code
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CKSN	Ciphering Key Sequence Number
C/R	Command / Response

C1	Path Loss Criterion
C2	Reselection Criterion
DCCH	Dedicated Control Channel
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
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
GP	Guard Period
GSM	Global System for Mobile Communication
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
Kc	Authentication Key
L	Length Indicator
LAI	Location Area Information
LPD	Link Protocol Discriminator
M	More Data Bit
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
NCC	National Colour Code
NECI	New Establishment Causes included
N(R)	Receive Number
N(S)	Send Number
OTD	Observed Time Difference
P	Poll Bit
PCH	Paging Channel
PDU	Protocol Description Unit
P/F	Poll / Final Bit
PL	Physical Layer
PLMN	Public Land Mobile Network
RACH	Random Access Channel
REJ	Reject Frame
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point
SAPI	Service Access Point Identifier
SDCCH	Slow Dedicated Control Channel
SIM	Subscriber Identity Module
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SS	Supplementary Services
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TDMA	Time Division Multiple Access
TMSI	Temporary Mobile Subscriber Identity

UA	Unnumbered Acknowledgement Frame
UI	Unnumbered Information Frame
VPLMN	Visiting Public Land Mobile Network
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable

1.3 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).

2 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 base of the Protocol Stack rests on the physical layer.

The Data Link Layer (DL) is used to handle an acknowledged connection between mobile and base station. The LAPDm protocol is used.

Radio Resource (RR) manages the resources of the air-interface. That means configuration of physical layer, cell selection and cell reselection, data transfer, RR-Connection handling.

Mobility Management (MM) handles registration aspects for the mobile station. It detects changes of location areas and updates a mobile station in the new location area.

Call Control (CC) provides the call functionality. This includes call establishment, call maintenance procedures like Hold, Retrieve or Modify, and call disconnection.

Supplementary Services (SS) handles all call independent supplementary services like call forwarding or call barring.

Short Message Services (SMS) is used for sending and receiving point-to-point short messages. Additionally the reception of cell broadcast short messages is included.

The man machine interface (MMI) is the interface to the user. Normally it is connected with a keypad as input device and a display as output device.

Between the several entities data interfaces are defined. These data interfaces are called Service Access Points (SAPs), indicating that an upper layer uses the services of a lower layer.

The GSM specification do not set out any implementation of the Protocol Stack. The following diagrams show the implementation described in all these documents for the mobile station. All entities except the Man Machine Interface and Physical Layer are implemented as part of the Protocol Stack.

Error! Objects cannot be created from editing field codes.

Figure 1: Mobile-station protocol architecture

This document describes the test as set out in GSM 11.10-1, §31.

3 Parameters

```
#define REST_OCTET 0x2b
#define ARFCN_BCCH 122
#define ARFCN_BCCH_B 48
#define NCC 0x5
#define BCC 0x6
#define BSIC ((NCC<<3)|(BCC))
#define RFN 0
#define SAPI_0 0
/*-----*\
| GSM 11.10
| 10 Generic call set up procedure
| 10.1 Generic call setup-up procedure for mobile terminating speech calls
| 10.1.2 Definition of system information messages
\*-----*/
/*-----*\
| Information Elements
\*-----*/
/*-----*\
| BCCH Frequency list:
| Indicates seven surrounding cells on any ARFCN of the supported
| band, excluding ARFCNs in or immediately adjacent to those
| specified in section 6.2 (GSM 11.10).
| From GSM 11.10, section 6.2 the following ARFCN are given :
| 10, 14, 17, 18, 22, 24, 26, 30, 31, 34, 38, 42, 45, 46, 50,
| 52, 54, 58, 59, 62, 66, 70, 73, 74, 78, 80, 82, 86, 87, 90,
| 94, 98, 101, 102, 106, 108, 110, 114
| The following 7 cells are chosen :
| 121,117, 76, 48, 12,7,1
| Thus BA is : 01100000 00000800 00008000 00000841
\*-----*/
IE_BEGIN(cm_reestablish_request_message_type)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x28,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(cm_reestablish_request_message_type)

IE_BEGIN(ciphering_mode_command_message_type)
    BF(8, 0x35,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(ciphering_mode_command_message_type)
IE_BEGIN(ciphering_mode_complete_message_type)
    BF(8, 0x32,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(ciphering_mode_complete_message_type)
IE_BEGIN(facility_message_type)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x3A,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(facility_message_type)

IE_BEGIN(ss_register_message_type)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x3B,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(ss_register_message_type)

IE_BEGIN(ss_release_complete_message_type)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x2A,ACT_CHECK,ANONYMOUS,SILENT)
```

```
IE_END(ss_release_complete_message_type)

IE_BEGIN( imsi_detach_indication_message_type )
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x01,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( imsi_detach_indication_message_type )

IE_BEGIN(connect_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x07,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(connect_message_type)

IE_BEGIN(connect_acknowledge_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x0F,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(connect_acknowledge_message_type)

IE_BEGIN(alerting_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x01,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(alerting_message_type)

IE_BEGIN(hold_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x18,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(hold_message_type)

IE_BEGIN(hold_ack_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x19,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(hold_ack_message_type)

IE_BEGIN(disconnect_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x25,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(disconnect_message_type)

IE_BEGIN(release_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x2D,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(release_message_type)

IE_BEGIN( tmsi_reallocation_command_message_type )
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x1A,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( tmsi_reallocation_command_message_type )

IE_BEGIN( tmsi_reallocation_complete_message_type )
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x1B,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( tmsi_reallocation_complete_message_type )
```

```
IE_BEGIN( location_updating_request_message_type )
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x08,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( location_updating_request_message_type )

IE_BEGIN( location_updating_accept_message_type )
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x02,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( location_updating_accept_message_type )

IE_BEGIN(assignment_command_message_type)
    BF(8, 0x2E,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(assignment_command_message_type)
IE_BEGIN(cm_service_accept_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x21,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(cm_service_accept_message_type)
IE_BEGIN( channel_release_message_type )
    BF( 8, 0x0D,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( channel_release_message_type )

IE_BEGIN(assignment_complete_message_type)
    BF(8, 0x29,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(assignment_complete_message_type)

IE_BEGIN(cause_16)
    BF(8,      2,ACT_CHECK, length,"two octets")
    BF(1,      1,ACT_CHECK, ext_1,SILENT)
    BF(2,      3,ACT_CHECK, coding_standard, "GSM Standard")
    BF(1,      0,ACT_SHOW, spare,SILENT)
    BF(4,      0,ACT_CHECK, location,"User")
    BF(1,      1,ACT_CHECK, ext_2,SILENT)
    BF(7,      16,ACT_CHECK, cause,"normal clearing")
IE_END(cause_16)

IE_BEGIN(cause_44)
    BF(8,      2,ACT_CHECK, length,"two octets")
    BF(1,      1,ACT_CHECK, ext_1,SILENT)
    BF(2,      3,ACT_CHECK, coding_standard, "GSM Standard")
    BF(1,      0,ACT_SHOW, spare,SILENT)
    BF(4,      0,ACT_CHECK, location,"User")
    BF(1,      1,ACT_CHECK, ext_2,SILENT)
    BF(7,      0x44,ACT_CHECK, cause,"ACM exceeds ACMmax")
IE_END(cause_44)

IE_BEGIN(power_command)
    BF(8,10,ACT_CHECK,power,SILENT)
IE_END(power_command)
IE_BEGIN(description_of_the_first_channel_after_time)
    BF( 5,M5(0,0,0,0,1),ACT_CHECK,      channel_type,"TCH ")
    BF( 3,      3,ACT_CHECK,      time_slot_number,"three")
    BF( 3,      BCC,ACT_CHECK,training_sequence_code,"same as BCCH")
    BF( 1,      0,ACT_CHECK,      hopping,"No")
    BF( 2,      0,ACT_CHECK,      spare,SILENT)
    BF(10, ARFCN_BCCH,ACT_CHECK,      arfcn,"ARFCN of the BCCH")
IE_END(description_of_the_first_channel_after_time)
```

```

IE_BEGIN(authentication_response_message_type)
    BF(1, 0, ACT_CHECK, ANONYMOUS, SILENT)
    BF(1, 0, ACT_SHOW, ANONYMOUS, SILENT)
    BF(6, 0x14, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(authentication_response_message_type)

IE_BEGIN( location Updating_type_attach)
    BF( 1, 0, ACT_CHECK, fo_req, "no follow-on request pending" )
    BF( 1, 0, ACT_CHECK, ANONYMOUS, "spare" )
    BF( 2, M2(1,0), ACT_CHECK, lut, "IMSI attach" )
IE_END( location Updating_type_attach )

IE_BEGIN( location Updating_type_normal)
    BF( 1, 0, ACT_CHECK, fo_req, "no follow-on request pending" )
    BF( 1, 0, ACT_CHECK, ANONYMOUS, "spare" )
    BF( 2, M2(0,0), ACT_CHECK, lut, "normal lup" )
IE_END( location Updating_type_normal)

IE_BEGIN( location Updating_type_periodic)
    BF( 1, 0, ACT_CHECK, fo_req, "no follow-on request pending" )
    BF( 1, 0, ACT_CHECK, ANONYMOUS, "spare" )
    BF( 2, M2(0,1), ACT_CHECK, lut, "periodic lup" )
IE_END( location Updating_type_periodic)

IE_BEGIN(call_proceeding_message_type)
    BF(1, 0, ACT_CHECK, ANONYMOUS, SILENT)
    BF(1, 0, ACT_SHOW, ANONYMOUS, SILENT)
    BF(6, 0x02, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(call_proceeding_message_type)

IE_BEGIN(bcch_frequency_list)
    BF(32, 0x01100000, ACT_CHECK, ANONYMOUS, "bit 128 thru 97")
    BF(32, 0x00000800, ACT_CHECK, ANONYMOUS, "bit 96 thru 65")
    BF(32, 0x00008000, ACT_CHECK, ANONYMOUS, "bit 64 thru 33")
    BF(32, 0x00000841, ACT_CHECK, ANONYMOUS, "bit 32 thru 1")
IE_END(bcch_frequency_list)

IE_BEGIN(cell_channel_description)
    BF(32, 0x00000000, ACT_CHECK, ANONYMOUS, "Includes the ")
    BF(32, 0x00000000, ACT_CHECK, ANONYMOUS, "hopping sequence ")
    BF(32, 0x00000000, ACT_CHECK, ANONYMOUS, "ARFCNs, if hopping ")
    BF(32, 0x00000000, ACT_CHECK, ANONYMOUS, "is used. ")
IE_END(cell_channel_description)

IE_BEGIN(cell_identity)
    BF(16, 0x0001, ACT_CHECK, ANONYMOUS, "CI VALUE 0001 hex (not relevant)")
IE_END(cell_identity)

IE_BEGIN(cell_identity_B)
    BF(16, 0x0002, ACT_CHECK, ANONYMOUS, "CI VALUE 0002 hex (not relevant)")
IE_END(cell_identity_B)

IE_BEGIN(cell_options)
    BF(1, 0, ACT_CHECK, ANONYMOUS, "spare ")
    BF(1, 0, ACT_CHECK, pwr, "power control not set")
    BF(2, 2, ACT_CHECK, dtx, "MS must not use DTX ")
    BF(4, 1, ACT_CHECK, radio_link_time_out, "8 ")
IE_END(cell_options)

IE_BEGIN(cell_selection_parameter)
    BF(3, 0, ACT_CHECK, cell_reselect_hysteresis, "0 dB")
    BF(5, 0, ACT_CHECK, ms_txpwr_max_cch, "Max. output power of MS")
    BF(1, 0, ACT_CHECK, acs, "no additional cell params")
    BF(1, 0, ACT_CHECK, neci, "New est. cause not supp.")
    BF(6, -90+111, ACT_CHECK, rxlev_access_min, "-90 dBm")
IE_END(cell_selection_parameter)

```

```

IE_BEGIN(cell_selection_parameter_neci1)
    BF(3, 0,ACT_CHECK,cell_reselect_hysteresis,"0 dB")
    BF(5, 0,ACT_CHECK,ms_txpwr_max_cch      ,"Max. output power of MS")
    BF(1, 0,ACT_CHECK,acs                    ,"no additional cell params")
    BF(1, 1,ACT_CHECK,neci                  ,"New est. cause supp.")
    BF(6,-90+111,ACT_CHECK,rxlev_access_min  ,"-90 dBm")
IE_END(cell_selection_parameter_neci1)

IE_BEGIN(control_channel_description)
    BF(1,0,ACT_CHECK,ANONYMOUS              ,"spare")
    BF(1,0,ACT_CHECK,att                     ,"MS shall not apply (not relevant) ")
    BF(3,0,ACT_CHECK,bs_ag_blks_res,"0 blocks reserved (not relevant) ")
    BF(3,1,ACT_CHECK,ccch_conf              ,"Combined CCCH/SDCCH (not relevant)")
    BF(5,0,ACT_CHECK,ANONYMOUS              ,"spare")
    BF(3,3,ACT_CHECK,bs_pa_mfrms            ,"5 multiframes (not relevant) ")
    BF(8,0,ACT_CHECK,t3212                  ,"Infinite")
IE_END(control_channel_description)

IE_BEGIN(control_channel_description_att)
    BF(1,0,ACT_CHECK,ANONYMOUS              ,"spare")
    BF(1,1,ACT_CHECK,att                     ,"MS shall apply")
    BF(3,0,ACT_CHECK,bs_ag_blks_res,"0 blocks reserved (not relevant) ")
    BF(3,1,ACT_CHECK,ccch_conf              ,"Combined CCCH/SDCCH (not relevant)")
    BF(5,0,ACT_CHECK,ANONYMOUS              ,"spare")
    BF(3,3,ACT_CHECK,bs_pa_mfrms            ,"5 multiframes (not relevant) ")
    BF(8,0,ACT_CHECK,t3212                  ,"Infinite")
IE_END(control_channel_description_att)

IE_BEGIN(control_channel_description_att_per)
    BF(1,0,ACT_CHECK,ANONYMOUS              ,"spare")
    BF(1,1,ACT_CHECK,att                     ,"MS shall apply")
    BF(3,0,ACT_CHECK,bs_ag_blks_res,"0 blocks reserved (not relevant) ")
    BF(3,1,ACT_CHECK,ccch_conf              ,"Combined CCCH/SDCCH (not relevant)")
    BF(5,0,ACT_CHECK,ANONYMOUS              ,"spare")
    BF(3,3,ACT_CHECK,bs_pa_mfrms            ,"5 multiframes (not relevant) ")
    BF(8,1,ACT_CHECK,t3212                  ,"6 minutes")
IE_END(control_channel_description_att_per)

IE_BEGIN(l2_pseudo_length_12)
    BF(6,12,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 1,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(l2_pseudo_length_12)
IE_BEGIN(l2_pseudo_length_18)
    BF(6,18,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 1,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(l2_pseudo_length_18)
IE_BEGIN(l2_pseudo_length_21)
    BF(6,21,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 1,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(l2_pseudo_length_21)
IE_BEGIN(l2_pseudo_length_22)
    BF(6,22,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1, 1,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(l2_pseudo_length_22)
#define MCC 0x262 /* 262 decimal (not relevant) */

```



```

#define MNC 1 /* 01 decimal (not relevant) */
#define LAC 0x0001 /* 0001 hex (not relevant) */
#define LAC_B 0x0002 /* 0002 hex (not relevant) */
IE_BEGIN(location_area_identification)
    BF( 4,6 ,ACT_CHECK,mcc_dig_2,"digit 2 of mobile country code")
    BF( 4,2 ,ACT_CHECK,mcc_dig_1,"digit 1 of mobile country code")
    BF( 4, 0xF,ACT_CHECK,ANONYMOUS ,"end of MCC ")
    BF( 4,2 ,ACT_CHECK,mcc_dig_3,"digit 3 of mobile country code")
    BF( 4,1 ,ACT_CHECK,mnc_dig_2,"digit 2 of mobile network code")
    BF( 4,0 ,ACT_CHECK,mnc_dig_1,"digit 1 of mobile network code")
    BF(16, LAC,ACT_CHECK,lac ,"Location area code ")
IE_END(location_area_identification)
IE_BEGIN(location_area_identification_B)
    BF( 4,6 ,ACT_CHECK,mcc_dig_2,"digit 2 of mobile country code")
    BF( 4,2 ,ACT_CHECK,mcc_dig_1,"digit 1 of mobile country code")
    BF( 4, 0xF,ACT_CHECK,ANONYMOUS ,"end of MCC ")
    BF( 4,2 ,ACT_CHECK,mcc_dig_3,"digit 3 of mobile country code")
    BF( 4,1 ,ACT_CHECK,mnc_dig_2,"digit 2 of mobile network code")
    BF( 4,0 ,ACT_CHECK,mnc_dig_1,"digit 1 of mobile network code")
    BF(16, LAC_B,ACT_CHECK,lac ,"Location area code ")
IE_END(location_area_identification_B)
IE_BEGIN(mobile_identity_tmsi)
    BF(8, 5, ACT_CHECK,length, "five octets to come")
    BF(4, M4(1,1,1,1), ACT_CHECK,ANONYMOUS,"bits 5-8 of octet 3 ='1111'")
    BF( 1, 0, ACT_CHECK,odd_even, "as applicable for TMSI")
    BF( 3, M3(1,0,0), ACT_CHECK,type, "TMSI")
    BF( 8, 0x12, ACT_CHECK, ANONYMOUS, SILENT)
    BF( 8, 0x34, ACT_CHECK, ANONYMOUS, SILENT)
    BF( 8, 0x56, ACT_CHECK, ANONYMOUS, SILENT)
    BF( 8, 0x78, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(mobile_identity_tmsi)
IE_BEGIN(tmsi)
    BF( 8, 0x12, ACT_CHECK, ANONYMOUS, SILENT)
    BF( 8, 0x34, ACT_CHECK, ANONYMOUS, SILENT)
    BF( 8, 0x56, ACT_CHECK, ANONYMOUS, SILENT)
    BF( 8, 0x78, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(tmsi)
IE_BEGIN(repeat_indicator_circular)
    BF(8,0xD1,ACT_CHECK, ri, SILENT)
IE_END(repeat_indicator_circular)
IE_BEGIN(cc_capabilities)
    BF(8, 1, ACT_CHECK, length, SILENT)
    BF(6, 0, ACT_CHECK, spare, SILENT)
    BF(1, 1, ACT_CHECK, pcp, SILENT)
    BF(1, 1, ACT_CHECK, dtmf, SILENT)
IE_END(cc_capabilities)
IE_BEGIN(signal_call_waiting)
    BF(8,M8(0,0,0,0,0,1,1,1),ACT_CHECK,signal_value,"(Any non-res. value)")
IE_END(signal_call_waiting)
IE_BEGIN(bearer_capability)
    BF(8, 3,ACT_CHECK, length, SILENT)
    BF(1, 0,ACT_CHECK, ext3, SILENT)
    BF(2, 1,ACT_CHECK, radio_channel_requirement,SILENT)
    BF(1, 0,ACT_CHECK, coding_standard, SILENT)
    BF(1, 0,ACT_CHECK, transfer_mode, SILENT)
    BF(3, 0,ACT_CHECK, info_transfer_capability, SILENT)
    BF(8,0x02,ACT_CHECK, octet_3a_1, SILENT)

```

```

        BF(8,0x80,ACT_CHECK, octet_3a_2,          SILENT)
    IE_END(bearer_capability)

    IE_BEGIN( 12_pseudo_length_19 )                /* Ref.: [1], §10.5.2.19 */
        BF( 6, 19,          ACT_CHECK,            ANONYMOUS,      SILENT )
        BF( 1, 0,          ACT_CHECK,            ANONYMOUS,      SILENT )
        BF( 1, 1,          ACT_CHECK,            ANONYMOUS,      SILENT )
    IE_END( 12_pseudo_length_19 )

    IE_BEGIN(tmsi_2)
        BF( 8, 0x12,        ACT_CHECK,            ANONYMOUS,      SILENT)
        BF( 8, 0x79,        ACT_CHECK,            ANONYMOUS,      SILENT)
        BF( 8, 0x13,        ACT_CHECK,            ANONYMOUS,      SILENT)
        BF( 8, 0x78,        ACT_CHECK,            ANONYMOUS,      SILENT)
    IE_END(tmsi_2)

    IE_BEGIN(rr_cause)
        BF(8,0,ACT_CHECK,rr_cause,"normal event")
    IE_END(rr_cause)
    IE_BEGIN(mode_of_the_first_channel)
        BF(8,1,ACT_CHECK,mode,"Speech full rate")
    IE_END(mode_of_the_first_channel)
    IE_BEGIN(iei_08)
        BF(8,0x08,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_08)
    IE_BEGIN(iei_13)
        BF(8,0x13,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_13)
    IE_BEGIN(iei_1C)
        BF(8,0x1C,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_1C)

    IE_BEGIN(iei_7F)
        BF(8,0x7F,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_7F)

    IE_BEGIN(iei_15)
        BF(8,0x15,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_15)
    IE_BEGIN(iei_17)
        BF(8,0x17,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_17)
    IE_BEGIN(iei_04)
        BF(8,0x04,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_04)
    IE_BEGIN(iei_34)
        BF(8,0x34,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_34)
    IE_BEGIN(iei_5E)
        BF(8,0x5E,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_5E)
    IE_BEGIN(iei_63)
        BF(8,0x63,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(iei_63)
    IE_BEGIN(transaction_identifier_source)
        BF(4,M4(0,0,0,0),ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(transaction_identifier_source)
    IE_BEGIN(transaction_identifier_source_1)
        BF(4,M4(0,0,0,1),ACT_CHECK,ANONYMOUS,SILENT)

```

```
IE_END(transaction_identifier_source_1)
IE_BEGIN(page_mode)
    BF(2,0,ACT_CHECK,spare,"two spare bits ")
    BF(2,0,ACT_CHECK,pm,"Normal Paging")
IE_END(page_mode)
```

```
IE_BEGIN(facility_aocc_storage1)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e1_value_lsb, "e1 = 10.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x26,ACT_CHECK,e2_value_msb, "e2=55.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e4_value_lsb, "e4=10.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e7_value_lsb, "e7=10.0")
IE_END(facility_aocc_storage1)
```

```
IE_BEGIN(facility_aocc_acmmmax1)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e1_value_lsb, "e1 = 1.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x2C,ACT_CHECK,e2_value_msb, "e2=30.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_acmmmax1)
```

```
IE_BEGIN(facility_aocc_zero_cai)
    BF(8, 43, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA1, ACT_CHECK, component_type_tag, "INVOKE")
    BF(8, 41, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
    BF(8, 2, ACT_CHECK, operation_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, operation_code_length, SILENT)
    BF(8, 125, ACT_CHECK, operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30, ACT_CHECK, sequence_identifier, SILENT)
    BF(8, 33, ACT_CHECK, ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80, ACT_CHECK, ss_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, ss_code_length, SILENT)
    BF(8, 0x72, ACT_CHECK, ss_code, "AoC Charging")
    BF(8, 0xA1, ACT_CHECK, charging_info_id, SILENT)
    BF(8, 28, ACT_CHECK, charging_info_length, SILENT)
    BF(8, 0x81, ACT_CHECK, e1_tag, SILENT)
    BF(8, 2, ACT_CHECK, e1_length, SILENT)
    BF(8, 0, ACT_CHECK, e1_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e1_value_lsb, "e1 = 0.0")
    BF(8, 0x82, ACT_CHECK, e2_tag, SILENT)
    BF(8, 2, ACT_CHECK, e2_length, SILENT)
    BF(8, 0, ACT_CHECK, e2_value_lsb, SILENT)
    BF(8, 0, ACT_CHECK, e2_value_msb, "e2=0.0")
    BF(8, 0x83, ACT_CHECK, e3_tag, SILENT)
    BF(8, 2, ACT_CHECK, e3_length, SILENT)
    BF(8, 0, ACT_CHECK, e3_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e3_value_lsb, "e3=0")
    BF(8, 0x84, ACT_CHECK, e4_tag, SILENT)
    BF(8, 2, ACT_CHECK, e4_length, SILENT)
    BF(8, 0, ACT_CHECK, e4_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e4_value_lsb, "e4=0.0")
    BF(8, 0x85, ACT_CHECK, e5_tag, SILENT)
    BF(8, 2, ACT_CHECK, e5_length, SILENT)
    BF(8, 0, ACT_CHECK, e5_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e5_value_lsb, "e5=0")
    BF(8, 0x86, ACT_CHECK, e6_tag, SILENT)
    BF(8, 2, ACT_CHECK, e6_length, SILENT)
    BF(8, 0, ACT_CHECK, e6_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e6_value_lsb, "e6=0")
    BF(8, 0x87, ACT_CHECK, e7_tag, SILENT)
    BF(8, 2, ACT_CHECK, e7_length, SILENT)
    BF(8, 0, ACT_CHECK, e7_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_zero_cai)
```

```
IE_BEGIN(facility_aocc_k1)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 60 ,ACT_CHECK,e1_value_lsb, "e1 = 6.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 140 ,ACT_CHECK,e2_value_msb, "e2=14.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 250 ,ACT_CHECK,e4_value_lsb, "e4=25.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0x58,ACT_CHECK,e7_value_lsb, "e7=60.0")
IE_END(facility_aocc_k1)
```

```
IE_BEGIN(facility_aocc_k2)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_lsb, "e1 = 0.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0 ,ACT_CHECK,e2_value_msb, "e2=0.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 3 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0xE8,ACT_CHECK,e4_value_lsb, "e4=100.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_k2)
```



```
IE_BEGIN(facility_aocc_k3)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 9 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 0xC4,ACT_CHECK,e1_value_lsb, "e1 = 250.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 160 ,ACT_CHECK,e2_value_msb, "e2=16.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 200 ,ACT_CHECK,e3_value_lsb, "e3=2")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0x13,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0x88,ACT_CHECK,e4_value_lsb, "e4=500.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0x58,ACT_CHECK,e7_value_lsb, "e7=60.0")
IE_END(facility_aocc_k3)
```

```
IE_BEGIN(facility_aocc_k4)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e1_value_lsb, "e1 = 1.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 10 ,ACT_CHECK,e2_value_msb, "e2=1.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e5_value_lsb, "e5=10.0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e6_value_lsb, "e6=10.0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e7_value_lsb, "e7=1.0")
IE_END(facility_aocc_k4)
```

```
IE_BEGIN(facility_aocc_k5)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 125 ,ACT_CHECK,e1_value_lsb, "e1 = 12.5")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x2C,ACT_CHECK,e2_value_msb, "e2=30.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 250 ,ACT_CHECK,e4_value_lsb, "e4=25.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e5_value_lsb, "e5=10.0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e6_value_lsb, "e6=10.0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0x2C,ACT_CHECK,e7_value_lsb, "e7=30.0")
IE_END(facility_aocc_k5)
```

```
IE_BEGIN(facility_aocc_hold1)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 70 ,ACT_CHECK,e1_value_lsb, "e1 = 7.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x90,ACT_CHECK,e2_value_msb, "e2=40.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_hold1)
```

```
IE_BEGIN(facility_aocc_hold2)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 130 ,ACT_CHECK,e1_value_lsb, "e1 = 13.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x90,ACT_CHECK,e2_value_msb, "e2=40.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_hold2)
```

```
IE_BEGIN(facility_aocc_mpty1)
  BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
  BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
  BF(8, 41 ,ACT_CHECK,component_length, SILENT)
  BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
  BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
  BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
  BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
  BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
  BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
  BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
  BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
  BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
  BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
  BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
  BF(8, 190 ,ACT_CHECK,e1_value_lsb, "e1 = 19.0")
  BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
  BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
  BF(8, 0x90,ACT_CHECK,e2_value_msb, "e2=40.0")
  BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
  BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
  BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
  BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
  BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_mpty1)
```

```
IE_BEGIN(facility_aocc_mpty2)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 0x22,ACT_CHECK,e1_value_lsb, "e1 = 29.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x90,ACT_CHECK,e2_value_msb, "e2=40.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_mpty2)
```

```
IE_BEGIN(facility_aocc_cnf)
    BF(8, 5, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA2, ACT_CHECK, component_type_tag, "Return Result")
    BF(8, 3, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
IE_END(facility_aocc_cnf)

IE_BEGIN(facility_aocc_k1_mtc)
    BF(8, 43, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA1, ACT_CHECK, component_type_tag, "INVOKE")
    BF(8, 41, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
    BF(8, 2, ACT_CHECK, operation_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, operation_code_length, SILENT)
    BF(8, 125, ACT_CHECK, operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30, ACT_CHECK, sequence_identifier, SILENT)
    BF(8, 33, ACT_CHECK, ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80, ACT_CHECK, ss_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, ss_code_length, SILENT)
    BF(8, 0x72, ACT_CHECK, ss_code, "AoC Charging")
    BF(8, 0xA1, ACT_CHECK, charging_info_id, SILENT)
    BF(8, 28, ACT_CHECK, charging_info_length, SILENT)
    BF(8, 0x81, ACT_CHECK, e1_tag, SILENT)
    BF(8, 2, ACT_CHECK, e1_length, SILENT)
    BF(8, 0, ACT_CHECK, e1_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e1_value_lsb, "e1 = 0.0")
    BF(8, 0x82, ACT_CHECK, e2_tag, SILENT)
    BF(8, 2, ACT_CHECK, e2_length, SILENT)
    BF(8, 0, ACT_CHECK, e2_value_lsb, SILENT)
    BF(8, 0, ACT_CHECK, e2_value_msb, "e2=0.0")
    BF(8, 0x83, ACT_CHECK, e3_tag, SILENT)
    BF(8, 2, ACT_CHECK, e3_length, SILENT)
    BF(8, 0, ACT_CHECK, e3_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e3_value_lsb, "e3=0")
    BF(8, 0x84, ACT_CHECK, e4_tag, SILENT)
    BF(8, 2, ACT_CHECK, e4_length, SILENT)
    BF(8, 0, ACT_CHECK, e4_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e4_value_lsb, "e4=0.0")
    BF(8, 0x85, ACT_CHECK, e5_tag, SILENT)
    BF(8, 2, ACT_CHECK, e5_length, SILENT)
    BF(8, 0, ACT_CHECK, e5_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e5_value_lsb, "e5=0.0")
    BF(8, 0x86, ACT_CHECK, e6_tag, SILENT)
    BF(8, 2, ACT_CHECK, e6_length, SILENT)
    BF(8, 0, ACT_CHECK, e6_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e6_value_lsb, "e6=0.0")
    BF(8, 0x87, ACT_CHECK, e7_tag, SILENT)
    BF(8, 2, ACT_CHECK, e7_length, SILENT)
    BF(8, 0, ACT_CHECK, e7_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_k1_mtc)
```



```
IE_BEGIN(facility_aocc_k2_mtc)
  BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
  BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
  BF(8, 41 ,ACT_CHECK,component_length, SILENT)
  BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
  BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
  BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
  BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
  BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
  BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
  BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
  BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
  BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
  BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
  BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e1_value_lsb, "e1 = 0.0")
  BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
  BF(8, 0 ,ACT_CHECK,e2_value_msb, "e2=0.0")
  BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
  BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
  BF(8, 3 ,ACT_CHECK,e4_value_msb, SILENT)
  BF(8, 0xE8,ACT_CHECK,e4_value_lsb, "e4=100.0")
  BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0.0")
  BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0.0")
  BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_k2_mtc)
```

```
IE_BEGIN(facility_aocc_k3_mtc)
    BF(8, 43, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA1, ACT_CHECK, component_type_tag, "INVOKE")
    BF(8, 41, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
    BF(8, 2, ACT_CHECK, operation_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, operation_code_length, SILENT)
    BF(8, 125, ACT_CHECK, operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30, ACT_CHECK, sequence_identifier, SILENT)
    BF(8, 33, ACT_CHECK, ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80, ACT_CHECK, ss_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, ss_code_length, SILENT)
    BF(8, 0x72, ACT_CHECK, ss_code, "AoC Charging")
    BF(8, 0xA1, ACT_CHECK, charging_info_id, SILENT)
    BF(8, 28, ACT_CHECK, charging_info_length, SILENT)
    BF(8, 0x81, ACT_CHECK, e1_tag, SILENT)
    BF(8, 2, ACT_CHECK, e1_length, SILENT)
    BF(8, 0, ACT_CHECK, e1_value_msb, SILENT)
    BF(8, 60, ACT_CHECK, e1_value_lsb, "e1 = 6.0")
    BF(8, 0x82, ACT_CHECK, e2_tag, SILENT)
    BF(8, 2, ACT_CHECK, e2_length, SILENT)
    BF(8, 0, ACT_CHECK, e2_value_lsb, SILENT)
    BF(8, 140, ACT_CHECK, e2_value_msb, "e2=14.0")
    BF(8, 0x83, ACT_CHECK, e3_tag, SILENT)
    BF(8, 2, ACT_CHECK, e3_length, SILENT)
    BF(8, 0, ACT_CHECK, e3_value_msb, SILENT)
    BF(8, 100, ACT_CHECK, e3_value_lsb, "e3=1.0")
    BF(8, 0x84, ACT_CHECK, e4_tag, SILENT)
    BF(8, 2, ACT_CHECK, e4_length, SILENT)
    BF(8, 0, ACT_CHECK, e4_value_msb, SILENT)
    BF(8, 250, ACT_CHECK, e4_value_lsb, "e4=25.0")
    BF(8, 0x85, ACT_CHECK, e5_tag, SILENT)
    BF(8, 2, ACT_CHECK, e5_length, SILENT)
    BF(8, 0, ACT_CHECK, e5_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e5_value_lsb, "e5=0.0")
    BF(8, 0x86, ACT_CHECK, e6_tag, SILENT)
    BF(8, 2, ACT_CHECK, e6_length, SILENT)
    BF(8, 0, ACT_CHECK, e6_value_msb, SILENT)
    BF(8, 0, ACT_CHECK, e6_value_lsb, "e6=0.0")
    BF(8, 0x87, ACT_CHECK, e7_tag, SILENT)
    BF(8, 2, ACT_CHECK, e7_length, SILENT)
    BF(8, 2, ACT_CHECK, e7_value_msb, SILENT)
    BF(8, 0x58, ACT_CHECK, e7_value_lsb, "e7=60.0")
IE_END(facility_aocc_k3_mtc)
```

```

IE_BEGIN(facility_aocc_k4_mtc)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e1_value_lsb, "e1 = 1.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 10 ,ACT_CHECK,e2_value_msb, "e2=1.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1.0")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0.0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0.0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 10 ,ACT_CHECK,e7_value_lsb, "e7=1.0")
IE_END(facility_aocc_k4_mtc)

```

```
IE_BEGIN(facility_aocc_k5_mtc)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 125 ,ACT_CHECK,e1_value_lsb, "e1 = 12.5")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x2C,ACT_CHECK,e2_value_msb, "e2=30.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1.0")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 250 ,ACT_CHECK,e4_value_lsb, "e4=25.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0.0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0.0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0x2C,ACT_CHECK,e7_value_lsb, "e7=30.0")
IE_END(facility_aocc_k5_mtc)
```

```
IE_BEGIN(facility_aocc_set_1)
  BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
  BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
  BF(8, 41 ,ACT_CHECK,component_length, SILENT)
  BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
  BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
  BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
  BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
  BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
  BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
  BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
  BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
  BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
  BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
  BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e1_value_lsb, "e1 = 10.0")
  BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
  BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
  BF(8, 0x18,ACT_CHECK,e2_value_msb, "e2=28.0")
  BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
  BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e4_value_lsb, "e4=10.0")
  BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
  BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
  BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
  BF(8, 2 ,ACT_CHECK,e7_value_msb, SILENT)
  BF(8, 0x58,ACT_CHECK,e7_value_lsb, "e7=60.0")
IE_END(facility_aocc_set_1)
```

```
IE_BEGIN(facility_aocc_set_2)
  BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
  BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
  BF(8, 41 ,ACT_CHECK,component_length, SILENT)
  BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
  BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
  BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
  BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
  BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
  BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
  BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
  BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
  BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
  BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
  BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
  BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e1_value_lsb, "e1 = 10.0")
  BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e2_value_lsb, SILENT)
  BF(8, 140 ,ACT_CHECK,e2_value_msb, "e2=14.0")
  BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
  BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
  BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
  BF(8, 50 ,ACT_CHECK,e4_value_lsb, "e4=5.0")
  BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
  BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
  BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
  BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
  BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
  BF(8, 2 ,ACT_CHECK,e7_value_msb, SILENT)
  BF(8, 0x58,ACT_CHECK,e7_value_lsb, "e7=60.0")
IE_END(facility_aocc_set_2)
```

```

IE_BEGIN(facility_aocc_diff_1)
    BF(8, 43 ,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 41 ,ACT_CHECK,component_length, SILENT)
    BF(8, 2 ,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0 ,ACT_CHECK,invoke_id, SILENT)
    BF(8, 2 ,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 125 ,ACT_CHECK,operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30,ACT_CHECK,sequence_identifier, SILENT)
    BF(8, 33 ,ACT_CHECK,ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 1 ,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x72,ACT_CHECK,ss_code, "AoC Charging")
    BF(8, 0xA1,ACT_CHECK,charging_info_id, SILENT)
    BF(8, 28 ,ACT_CHECK,charging_info_length, SILENT)
    BF(8, 0x81,ACT_CHECK,e1_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e1_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e1_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e1_value_lsb, "e1 = 10.0")
    BF(8, 0x82,ACT_CHECK,e2_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e2_length, SILENT)
    BF(8, 1 ,ACT_CHECK,e2_value_lsb, SILENT)
    BF(8, 0x90,ACT_CHECK,e2_value_msb, "e2=40.0")
    BF(8, 0x83,ACT_CHECK,e3_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e3_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e3_value_msb, SILENT)
    BF(8, 100 ,ACT_CHECK,e3_value_lsb, "e3=1")
    BF(8, 0x84,ACT_CHECK,e4_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e4_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e4_value_lsb, "e4=0.0")
    BF(8, 0x85,ACT_CHECK,e5_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e5_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e5_value_lsb, "e5=0")
    BF(8, 0x86,ACT_CHECK,e6_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e6_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e6_value_lsb, "e6=0")
    BF(8, 0x87,ACT_CHECK,e7_tag, SILENT)
    BF(8, 2 ,ACT_CHECK,e7_length, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_msb, SILENT)
    BF(8, 0 ,ACT_CHECK,e7_value_lsb, "e7=0.0")
IE_END(facility_aocc_diff_1)

```

```
IE_BEGIN(facility_aocc_diff_2)
    BF(8, 27, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA1, ACT_CHECK, component_type_tag, "INVOKE")
    BF(8, 25, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
    BF(8, 2, ACT_CHECK, operation_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, operation_code_length, SILENT)
    BF(8, 125, ACT_CHECK, operation_code, "ForwardChargeAdviceArg")
    BF(8, 0x30, ACT_CHECK, sequence_identifier, SILENT)
    BF(8, 17, ACT_CHECK, ForwardChargeAdviceArgLength, SILENT)
    BF(8, 0x80, ACT_CHECK, ss_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, ss_code_length, SILENT)
    BF(8, 0x72, ACT_CHECK, ss_code, "AoC Charging")
    BF(8, 0xA1, ACT_CHECK, charging_info_id, SILENT)
    BF(8, 12, ACT_CHECK, charging_info_length, SILENT)
    BF(8, 0x81, ACT_CHECK, e1_tag, SILENT)
    BF(8, 2, ACT_CHECK, e1_length, SILENT)
    BF(8, 0, ACT_CHECK, e1_value_msb, SILENT)
    BF(8, 100, ACT_CHECK, e1_value_lsb, "e1 = 10.0")
    BF(8, 0x82, ACT_CHECK, e2_tag, SILENT)
    BF(8, 2, ACT_CHECK, e2_length, SILENT)
    BF(8, 1, ACT_CHECK, e2_value_lsb, SILENT)
    BF(8, 0x90, ACT_CHECK, e2_value_msb, "e2=40.0")
    BF(8, 0x83, ACT_CHECK, e3_tag, SILENT)
    BF(8, 2, ACT_CHECK, e3_length, SILENT)
    BF(8, 0, ACT_CHECK, e3_value_msb, SILENT)
    BF(8, 100, ACT_CHECK, e3_value_lsb, "e3=1")
IE_END(facility_aocc_diff_2)
```



```
IE_BEGIN(facility_ie_mpty_invoke)
    BF(8, 8, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA1, ACT_CHECK, component_type_tag, "INVOKE")
    BF(8, 6, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
    BF(8, 2, ACT_CHECK, operation_code_tag, SILENT)
    BF(8, 1, ACT_CHECK, operation_code_length, SILENT)
    BF(8, 124, ACT_CHECK, operation_code, "Build Multi-Party")
IE_END(facility_ie_mpty_invoke)

IE_BEGIN(facility_ie_mpty_return_result)
    BF(8, 5, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA2, ACT_CHECK, component_type_tag, "Return Result")
    BF(8, 3, ACT_CHECK, component_length, SILENT)
    BF(8, 2, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 1, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0, ACT_CHECK, invoke_id, SILENT)
IE_END(facility_ie_mpty_return_result)

IE_BEGIN(facility_100A_req)
    BF(8, 0x1A, ACT_CHECK, length_of_fac_ie_content, SILENT)
    BF(8, 0xA1, ACT_CHECK, component_type_tag, "INVOKE")
    BF(8, 0x18, ACT_CHECK, component_length, SILENT)
    BF(8, 0x02, ACT_CHECK, invoke_id_tag, SILENT)
    BF(8, 0x01, ACT_CHECK, invoke_id_length, SILENT)
    BF(8, 0x00, ACT_CHECK, invoke_id, SILENT)
    BF(8, 0x02, ACT_CHECK, operation_code_tag, SILENT)
    BF(8, 0x01, ACT_CHECK, operation_code_length, SILENT)
    BF(8, 0x0A, ACT_CHECK, operation_code, "RegisterSS")
    BF(8, 0x30, ACT_CHECK, register_SS_Arg, SILENT)
    BF(8, 0x10, ACT_CHECK, register_SS_Arg_length, SILENT)
    BF(8, 0x04, ACT_CHECK, ss_code_tag, SILENT)
    BF(8, 0x01, ACT_CHECK, ss_code_length, SILENT)
    BF(8, 0x2A, ACT_CHECK, ss_code, "CFNRy")
    BF(8, 0x83, ACT_CHECK, basic_service_code_id, SILENT)
    BF(8, 0x01, ACT_CHECK, teleservice_length, SILENT)
    BF(8, 0x11, ACT_CHECK, teleservice_code, "AllSpeechTransmission")
    BF(8, 0x84, ACT_CHECK, forwardedToNumber_id, SILENT)
    BF(8, 0x05, ACT_CHECK, forwardedToNumber_length, SILENT)
    BF(8, 0x81, ACT_CHECK, address_type, "unknown Number")
    BF(8, 0x00, ACT_CHECK, number_0, SILENT)
    BF(8, 0x34, ACT_CHECK, number_1, SILENT)
    BF(8, 0x21, ACT_CHECK, number_2, SILENT)
    BF(8, 0x43, ACT_CHECK, number_3, SILENT)
    BF(8, 0x85, ACT_CHECK, no_reply_tag, SILENT)
    BF(8, 0x01, ACT_CHECK, no_reply_length, SILENT)
    BF(8, 0x05, ACT_CHECK, no_reply_value, SILENT)
IE_END(facility_100A_req)
```

```
IE_BEGIN(facility_100A_cnf)
    BF(8, 0x22,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA2,ACT_CHECK,component_type_tag,"Return Result")
    BF(8, 0x20,ACT_CHECK,component_length, SILENT)
    BF(8, 0x02,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0x00,ACT_CHECK,invoke_id, SILENT)
    BF(8, 0x30,ACT_CHECK,sequence_tag, SILENT)
    BF(8, 0x1B,ACT_CHECK,sequence_length, SILENT)
    BF(8, 0x02,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 0x0A,ACT_CHECK,operation_code, "RegisterSS")
    BF(8, 0xA0,ACT_CHECK,forwarding_info_tag, SILENT)
    BF(8, 0x16,ACT_CHECK,forwarding_info_length, SILENT)
    BF(8, 0x04,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x2A,ACT_CHECK,ss_code, "CFNRy")
    BF(8, 0x30,ACT_CHECK,forwarding_feature_list_seq, SILENT)
    BF(8, 0x11,ACT_CHECK,forwarding_feature_list_length, SILENT)
    BF(8, 0x30,ACT_CHECK,forwarding_feature_seq, SILENT)
    BF(8, 0x0F,ACT_CHECK,forwarding_feature_length, SILENT)
    BF(8, 0x83,ACT_CHECK,basic_service_code_id, SILENT)
    BF(8, 0x01,ACT_CHECK,teleservice_length, SILENT)
    BF(8, 0x11,ACT_CHECK,teleservice_code, "AllSpeechTransmission")
    BF(8, 0x84,ACT_CHECK,ss_status_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,ss_status_length, SILENT)
    BF(8, 0x07,ACT_CHECK,ss_status_code, "Prov., Registered, Active")
    BF(8, 0x85,ACT_CHECK,forwardedToNumber_id, SILENT)
    BF(8, 0x04,ACT_CHECK,forwardedToNumber_length, SILENT)
    BF(8, 0x91,ACT_CHECK,address_type, "International Number")
    BF(8, 0x34,ACT_CHECK,number_1, SILENT)
    BF(8, 0x21,ACT_CHECK,number_2, SILENT)
    BF(8, 0x43,ACT_CHECK,number_3, SILENT)
    BF(8, 0x87,ACT_CHECK,no_reply_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,no_reply_length, SILENT)
    BF(8, 0x05,ACT_CHECK,no_reply_value, SILENT)
IE_END(facility_100A_cnf)
```

```

IE_BEGIN(facility_100B_req)
    BF(8, 0x17,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA1,ACT_CHECK,component_type_tag,"INVOKE")
    BF(8, 0x15,ACT_CHECK,component_length, SILENT)
    BF(8, 0x02,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0x01,ACT_CHECK,invoke_id, SILENT)
    BF(8, 0x02,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 0x0A,ACT_CHECK,operation_code, "RegisterSS")
    BF(8, 0x30,ACT_CHECK,register_SS_Arg, SILENT)
    BF(8, 0x0D,ACT_CHECK,register_SS_Arg_length, SILENT)
    BF(8, 0x04,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x21,ACT_CHECK,ss_code, "CFU")
    BF(8, 0x83,ACT_CHECK,basic_service_code_id, SILENT)
    BF(8, 0x01,ACT_CHECK,teleservice_length, SILENT)
    BF(8, 0x60,ACT_CHECK,teleservice_code, "AllFaxServices")
    BF(8, 0x84,ACT_CHECK,forwardedToNumber_id, SILENT)
    BF(8, 0x05,ACT_CHECK,forwardedToNumber_length, SILENT)
    BF(8, 0x81,ACT_CHECK,address_type, "unknown number Number")
    BF(8, 0x00,ACT_CHECK,number_0, SILENT)
    BF(8, 0x34,ACT_CHECK,number_1, SILENT)
    BF(8, 0x21,ACT_CHECK,number_2, SILENT)
    BF(8, 0x43,ACT_CHECK,number_3, SILENT)
IE_END(facility_100B_req)

IE_BEGIN(facility_100B_cnf)
    BF(8, 0x29,ACT_CHECK,length_of_fac_ie_content,SILENT)
    BF(8, 0xA2,ACT_CHECK,component_type_tag,"Return Result")
    BF(8, 0x80,ACT_CHECK,component_length, SILENT)
    BF(8, 0x02,ACT_CHECK,invoke_id_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,invoke_id_length, SILENT)
    BF(8, 0x01,ACT_CHECK,invoke_id, SILENT)
    BF(8, 0x30,ACT_CHECK,sequence_tag, SILENT)
    BF(8, 0x80,ACT_CHECK,sequence_length, SILENT)
    BF(8, 0x02,ACT_CHECK,operation_code_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,operation_code_length, SILENT)
    BF(8, 0x0A,ACT_CHECK,operation_code, "RegisterSS")
    BF(8, 0xA0,ACT_CHECK,forwarding_info_tag, SILENT)
    BF(8, 0x80,ACT_CHECK,forwarding_info_length, SILENT)
    BF(8, 0x04,ACT_CHECK,ss_code_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,ss_code_length, SILENT)
    BF(8, 0x21,ACT_CHECK,ss_code, "CFU")
    BF(8, 0x30,ACT_CHECK,forwarding_feature_list_seq, SILENT)
    BF(8, 0x80,ACT_CHECK,forwarding_feature_list_length, SILENT)
    BF(8, 0x30,ACT_CHECK,forwarding_feature_seq, SILENT)
    BF(8, 0x80,ACT_CHECK,forwarding_feature_length, SILENT)
    BF(8, 0x83,ACT_CHECK,basic_service_code_id, SILENT)
    BF(8, 0x01,ACT_CHECK,teleservice_length, SILENT)
    BF(8, 0x60,ACT_CHECK,teleservice_code, "AllFaxServices")
    BF(8, 0x84,ACT_CHECK,ss_status_tag, SILENT)
    BF(8, 0x01,ACT_CHECK,ss_status_length, SILENT)
    BF(8, 0x07,ACT_CHECK,ss_status_code, "Prov., Registered, Active")
    BF(8, 0x85,ACT_CHECK,forwardedToNumber_id, SILENT)
    BF(8, 0x04,ACT_CHECK,forwardedToNumber_length, SILENT)
    BF(8, 0x91,ACT_CHECK,address_type, "International Number")
    BF(8, 0x34,ACT_CHECK,number_1, SILENT)
    BF(8, 0x21,ACT_CHECK,number_2, SILENT)
    BF(8, 0x43,ACT_CHECK,number_3, SILENT)

```

```

BF(8, 0x00,ACT_CHECK,end_of_content_tag, SILENT)
BF(8, 0x00,ACT_CHECK,length_indicator, SILENT)
BF(8, 0x00,ACT_CHECK,end_of_content_tag, SILENT)
BF(8, 0x00,ACT_CHECK,length_indicator, SILENT)
BF(8, 0x00,ACT_CHECK,end_of_content_tag, SILENT)
BF(8, 0x00,ACT_CHECK,length_indicator, SILENT)
BF(8, 0x00,ACT_CHECK,end_of_content_tag, SILENT)
BF(8, 0x00,ACT_CHECK,length_indicator, SILENT)
BF(8, 0x00,ACT_CHECK,end_of_content_tag, SILENT)
BF(8, 0x00,ACT_CHECK,length_indicator, SILENT)
IE_END(facility_100B_cnf)

IE_BEGIN(facility_deactBAOC_cnf)
BF(8, 0x19,ACT_CHECK,length_of_fac_ie_content,SILENT)
BF(8, 0xA2,ACT_CHECK,component_type_tag,"Return Result")
BF(8, 0x17,ACT_CHECK,component_length, SILENT)
BF(8, 2,ACT_CHECK,invoke_id_tag, SILENT)
BF(8, 1,ACT_CHECK,invoke_id_length, SILENT)
BF(8, 1,ACT_CHECK,invoke_id, SILENT)
BF(8, 0x30,ACT_CHECK,seq_id, SILENT)
BF(8, 0x12,ACT_CHECK,seq_length, SILENT)
BF(8, 0x02,ACT_CHECK,opc_id, SILENT)
BF(8, 1,ACT_CHECK,opc_length, SILENT)
BF(8, 0x0d,ACT_CHECK,opc_deact, SILENT)
BF(8, 0xA1,ACT_CHECK,cbinf_id, SILENT)
BF(8, 0x0d,ACT_CHECK,cbinf_length, SILENT)
BF(8, 0x04,ACT_CHECK,sscd_id, SILENT)
BF(8, 1,ACT_CHECK,sscd_length, SILENT)
BF(8, 0x92,ACT_CHECK,sscd_baoc, SILENT)
BF(8, 0x30,ACT_CHECK,cbfeatlst_id, SILENT)
BF(8, 0x08,ACT_CHECK,cbfeatlst_length, SILENT)
BF(8, 0x30,ACT_CHECK,cbfeat_id, SILENT)
BF(8, 0x06,ACT_CHECK,cbfeat_length, SILENT)
BF(8, 0x83,ACT_CHECK,tele_id, SILENT)
BF(8, 1,ACT_CHECK,tele_length, SILENT)
BF(8, 0,ACT_CHECK,tele_tlphny, SILENT)
BF(8, 0x84,ACT_CHECK,ssst_id, SILENT)
BF(8, 1,ACT_CHECK,ssst_length, SILENT)
BF(8, 1,ACT_CHECK,ssst_p, SILENT)
IE_END(facility_deactBAOC_cnf)

IE_BEGIN(facility_deactBAOC_req)
BF(8, 0x10,ACT_CHECK,length_of_fac_ie_content,SILENT)
BF(8, 0xA1,ACT_CHECK,component_type_tag,"Invoke Cmp")
BF(8, 0x0E,ACT_CHECK,component_length, SILENT)
BF(8, 2,ACT_CHECK,invoke_id_tag, SILENT)
BF(8, 1,ACT_CHECK,invoke_id_length, SILENT)
BF(8, 0,ACT_CHECK,invoke_id, SILENT)
BF(8, 2,ACT_CHECK,opc_id, SILENT)
BF(8, 1,ACT_CHECK,opc_length, SILENT)
BF(8, 0x0D,ACT_CHECK,opc_deact, SILENT)
BF(8, 0x30,ACT_CHECK,opprm_id, SILENT)
BF(8, 6,ACT_CHECK,opprm_length, SILENT)
BF(8, 0x04,ACT_CHECK,sscd_id, SILENT)
BF(8, 1,ACT_CHECK,sscd_length, SILENT)
BF(8, 0x92,ACT_CHECK,sscd_baoc, SILENT)
BF(8, 0x83,ACT_CHECK,tele_id, SILENT)
BF(8, 1,ACT_CHECK,tele_length, SILENT)
BF(8, 0,ACT_CHECK,tele_tlphny, SILENT)
IE_END(facility_deactBAOC_req)

```

```

IE_BEGIN(ss_version)
    BF(8, 1, ACT_CHECK, length, SILENT)
    BF(8, 0, ACT_SHOW, version, SILENT)
IE_END(ss_version)

IE_BEGIN(release_complete_message_type)
    BF(1, 0, ACT_CHECK, ANONYMOUS, SILENT)
    BF(1, 0, ACT_SHOW, ANONYMOUS, SILENT)
    BF(6, 0x2A, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(release_complete_message_type)

IE_BEGIN(called_party_bcd_number)
    BF( 8,7,ACT_CHECK,length, "length of IE")
    BF( 1,1,ACT_CHECK,ext, "Extension Bit")
    BF( 3,0,ACT_CHECK,type_of_number, "Unknown")
    BF( 4,1,ACT_CHECK,numbering_plan, "ISDN/telephony")
    BF( 4,3,ACT_CHECK,digit_2, "Digit 2")
    BF( 4,0,ACT_CHECK,digit_1, "Digit 1")
    BF( 4,3,ACT_CHECK,digit_4, "Digit 4")
    BF( 4,0,ACT_CHECK,digit_3, "Digit 3")
    BF( 4,0,ACT_CHECK,digit_6, "Digit 6")
    BF( 4,9,ACT_CHECK,digit_5, "Digit 5")
    BF( 4,4,ACT_CHECK,digit_8, "Digit 8")
    BF( 4,9,ACT_CHECK,digit_7, "Digit 7")
    BF( 4,1,ACT_CHECK,digit_10, "Digit 10")
    BF( 4,1,ACT_CHECK,digit_9, "Digit 9")
    BF( 4,0xF,ACT_CHECK,digit_12, "Digit 12")
    BF( 4,7,ACT_CHECK,digit_11, "Digit 11")
IE_END(called_party_bcd_number)

IE_BEGIN(bearer_capability_data)
    BF(8,7,ACT_CHECK, length, SILENT)
    BF(1,1,ACT_CHECK, ext3, SILENT)
    BF(2,1,ACT_CHECK, radio_channel_requirement, SILENT)
    BF(1,0,ACT_CHECK, coding_standard, SILENT)
    BF(1,0,ACT_CHECK, transfer_mode, SILENT)
    BF(3,2,ACT_CHECK, info_transfer_capability, SILENT)
    BF(1,1,ACT_CHECK, ext4, SILENT)
    BF(1,1,ACT_CHECK, compression, SILENT)
    BF(2,0,ACT_CHECK, structure, SILENT)
    BF(1,1,ACT_CHECK, duplex_mode, SILENT)
    BF(1,0,ACT_CHECK, configuration, SILENT)
    BF(1,0,ACT_CHECK, NIRR, SILENT)
    BF(1,0,ACT_CHECK, establishment, SILENT)
    BF(1,1,ACT_CHECK, ext5, SILENT)
    BF(2,1,ACT_CHECK, access_identifier, SILENT)
    BF(2,0,ACT_CHECK, rate_adaption, SILENT)
    BF(3,1,ACT_CHECK, access_protocol, SILENT)
    BF(1,0,ACT_CHECK, ext6, SILENT)
    BF(2,1,ACT_CHECK, layer_1, SILENT)
    BF(4,0,ACT_CHECK, default_layer_1, SILENT)
    BF(1,1,ACT_CHECK, sync_async, SILENT)
    BF(1,0,ACT_CHECK, ext6a, SILENT)
    BF(1,0,ACT_CHECK, stop_bits, SILENT)
    BF(1,0,ACT_CHECK, negotiation, SILENT)
    BF(1,1,ACT_CHECK, data_bits, SILENT)
    BF(4,3,ACT_CHECK, user_rate, SILENT)
    BF(1,0,ACT_CHECK, ext6b, SILENT)
    BF(2,3,ACT_CHECK, intermediate_rate, SILENT)

```

```

    BF(1,0,ACT_CHECK,    NIC_TX,                SILENT)
    BF(1,0,ACT_CHECK,    NIC_RX,                SILENT)
    BF(3,3,ACT_CHECK,    parity,                SILENT)
    BF(1,1,ACT_CHECK,    ext6c,                SILENT)
    BF(2,1,ACT_CHECK,    connection_element,    SILENT)
    BF(5,3,ACT_CHECK,    modem_type,           SILENT)
IE_END(bearer_capability_data)

```

```

IE_BEGIN(bearer_capability_ts_61)
    BF(8,7,ACT_CHECK,    length,                SILENT)
    BF(1,1,ACT_CHECK,    ext3,                SILENT)
    BF(2,0,ACT_CHECK,    radio_channel_requirement, SILENT)
    BF(1,0,ACT_CHECK,    coding_standard,        SILENT)
    BF(1,0,ACT_CHECK,    transfer_mode,          SILENT)
    BF(3,3,ACT_CHECK,    info_transfer_capability, SILENT)
    BF(1,1,ACT_CHECK,    ext4,                SILENT)
    BF(1,0,ACT_CHECK,    compression,           SILENT)
    BF(2,3,ACT_CHECK,    structure,             SILENT)
    BF(1,1,ACT_CHECK,    duplex_mode,           SILENT)
    BF(1,0,ACT_CHECK,    configuration,          SILENT)
    BF(1,0,ACT_CHECK,    NIRR,                 SILENT)
    BF(1,0,ACT_CHECK,    establishment,          SILENT)
    BF(1,1,ACT_CHECK,    ext5,                SILENT)
    BF(2,0,ACT_CHECK,    access_identifier,      SILENT)
    BF(2,0,ACT_CHECK,    rate_adaption,          SILENT)
    BF(3,1,ACT_CHECK,    access_protocol,        SILENT)
    BF(1,0,ACT_CHECK,    ext6,                SILENT)
    BF(2,1,ACT_CHECK,    layer_1,              SILENT)
    BF(4,0,ACT_CHECK,    default_layer_1,        SILENT)
    BF(1,0,ACT_CHECK,    sync_async,            SILENT)
    BF(1,0,ACT_CHECK,    ext6a,                SILENT)
    BF(1,0,ACT_CHECK,    stop_bits,             SILENT)
    BF(1,0,ACT_CHECK,    negotiation,           SILENT)
    BF(1,1,ACT_CHECK,    data_bits,             SILENT)
    BF(4,3,ACT_CHECK,    user_rate,             SILENT)
    BF(1,0,ACT_CHECK,    ext6b,                SILENT)
    BF(2,2,ACT_CHECK,    intermediate_rate,      SILENT)
    BF(1,0,ACT_CHECK,    NIC_TX,                SILENT)
    BF(1,0,ACT_CHECK,    NIC_RX,                SILENT)
    BF(3,3,ACT_CHECK,    parity,                SILENT)
    BF(1,1,ACT_CHECK,    ext6c,                SILENT)
    BF(2,0,ACT_CHECK,    connection_element,    SILENT)
    BF(5,0,ACT_CHECK,    modem_type,           SILENT)
IE_END(bearer_capability_ts_61)

```

```

IE_BEGIN(bearer_capability_bs_61)
    BF(8,7,ACT_CHECK,    length,                SILENT)
    BF(1,1,ACT_CHECK,    ext3,                SILENT)
    BF(2,0,ACT_CHECK,    radio_channel_requirement, SILENT)
    BF(1,0,ACT_CHECK,    coding_standard,        SILENT)
    BF(1,0,ACT_CHECK,    transfer_mode,          SILENT)
    BF(3,2,ACT_CHECK,    info_transfer_capability, SILENT)
    BF(1,1,ACT_CHECK,    ext4,                SILENT)
    BF(1,0,ACT_CHECK,    compression,           SILENT)
    BF(2,3,ACT_CHECK,    structure,             SILENT)
    BF(1,1,ACT_CHECK,    duplex_mode,           SILENT)
    BF(1,0,ACT_CHECK,    configuration,          SILENT)
    BF(1,0,ACT_CHECK,    NIRR,                 SILENT)
    BF(1,0,ACT_CHECK,    establishment,          SILENT)

```

```

BF(1,1,ACT_CHECK,    ext5,                                SILENT)
BF(2,0,ACT_CHECK,    access_identifier,                    SILENT)
BF(2,0,ACT_CHECK,    rate_adaption,                        SILENT)
BF(3,1,ACT_CHECK,    access_protocol,                      SILENT)
BF(1,0,ACT_CHECK,    ext6,                                SILENT)
BF(2,1,ACT_CHECK,    layer_1,                              SILENT)
BF(4,0,ACT_CHECK,    default_layer_1,                     SILENT)
BF(1,1,ACT_CHECK,    sync_async,                          SILENT)
BF(1,0,ACT_CHECK,    ext6a,                                SILENT)
BF(1,0,ACT_CHECK,    stop_bits,                            SILENT)
BF(1,0,ACT_CHECK,    negotiation,                         SILENT)
BF(1,1,ACT_CHECK,    data_bits,                            SILENT)
BF(4,3,ACT_CHECK,    user_rate,                            SILENT)
BF(1,0,ACT_CHECK,    ext6b,                                SILENT)
BF(2,2,ACT_CHECK,    intermediate_rate,                   SILENT)
BF(1,0,ACT_CHECK,    NIC_TX,                               SILENT)
BF(1,0,ACT_CHECK,    NIC_RX,                               SILENT)
BF(3,3,ACT_CHECK,    parity,                               SILENT)
BF(1,1,ACT_CHECK,    ext6c,                                SILENT)
BF(2,0,ACT_CHECK,    connection_element,                  SILENT)
BF(5,3,ACT_CHECK,    modem_type,                           SILENT)
IE_END(bearer_capability_bs_61)

IE_BEGIN(cause_88)
BF(8,                2,ACT_CHECK, length,"two octets")
BF(1,                1,ACT_CHECK, ext_1,SILENT)
BF(2,                3,ACT_CHECK, coding_standard, "GSM Standard")
BF(1,                0,ACT_SHOW,  spare,SILENT)
BF(4,                0,ACT_CHECK, location,"User")
BF(1,                1,ACT_CHECK, ext_2,SILENT)
BF(7,                88,ACT_CHECK, cause,"incompatible destination")
IE_END(cause_88)

IE_BEGIN(request_reference)
BF(3,                M3(1,0,0),ACT_NOP,random_access_info,"As in CHAN REQ")
BF(5,M5(1,1,1,1,1),ACT_NOP,  random_reference,SILENT)
BF(5,                0,ACT_NOP,    t1_,SILENT)
BF(6,                0,ACT_NOP,    t3_,SILENT)
BF(5,                0,ACT_NOP,    t2_,SILENT)
IE_END(request_reference)
IE_BEGIN(ciphering_key_sequence_number)
BF(1,                0,ACT_CHECK,    spare,SILENT)
BF(3,M3(0,1,1),ACT_CHECK,key_sequence,"from SIM card (3)")
IE_END(ciphering_key_sequence_number)
IE_BEGIN(authentication_parameter_rand)
BF(32,0x80000000,ACT_CHECK,rand_127_096,SILENT)
BF(32,0x00000012,ACT_CHECK,rand_095_064,SILENT)
BF(32,0x34000000,ACT_CHECK,rand_063_032,SILENT)
BF(32,0x0000000F,ACT_CHECK,rand_031_000,SILENT)
IE_END(authentication_parameter_rand)
IE_BEGIN(authentication_parameter_sres)
BF(32,0x0000000F,ACT_NOP,sres_031_000,SILENT)
IE_END(authentication_parameter_sres)
IE_BEGIN(ciphering_key_sequence_number_2)
BF(1,                0,ACT_CHECK, spare,SILENT)
BF(3,M3(0,1,0),ACT_CHECK,key_sequence,"sent BS->MS")
IE_END(ciphering_key_sequence_number_2)
IE_BEGIN(ciphering_mode_setting)
BF(3,M3(0,0,0),ACT_CHECK,algorithm_identifier,"A5/1")
BF(1,                1,ACT_CHECK,    start_ciphering,"Start ciphering")

```

```

IE_END(ciphering_mode_setting)
IE_BEGIN(cipher_response)
    BF(3,0,ACT_CHECK, spare,SILENT)
    BF(1,0,ACT_CHECK,cipher_response,"IMEISV shall not be included")
IE_END(cipher_response)
IE_BEGIN(ms_classmark)
    BF(8, 3,ACT_CHECK, length,SILENT)
    BF(1, 0,ACT_CHECK, spare,SILENT)
    BF(2, M2(0,1),ACT_CHECK, revision_level,"phase 2 MS")
    BF(1, 1,ACT_CHECK, es_ind,"Contr. Early Classmark Send.")
    BF(1, 0,ACT_CHECK, a5_1,"encryption algorithm A5/1 available")
    BF(3,M3(0,0,1),ACT_CHECK, rf_power_capability,"class 2")
    BF(1, 0,ACT_CHECK, spare2,SILENT)
    BF(1, 0,ACT_CHECK, ps_capability,SILENT)
    BF(2, M2(1,1),ACT_CHECK, ss_screening_indicator,SILENT)
    BF(1, 1,ACT_CHECK, sm_capability,"point to point SMS")
    BF(1, 0,ACT_CHECK, vbs,"no VBS cap. or no notific. wanted")
    BF(1, 0,ACT_CHECK, vgcs,"no VGCS cap. or no notific. wanted")
    BF(1, 0,ACT_CHECK, frequency_capability,"no extention band G1")
    BF(1, 1,ACT_CHECK, classmark_3," no add. MS cap. information")
    BF(4, 0,ACT_CHECK, spare3,SILENT)
    BF(1, 1,ACT_CHECK, cmsp,"CCBS supported")
    BF(1, 0,ACT_CHECK, a5_3,"A5/3 not available")
    BF(1, 0,ACT_CHECK, a5_2,"A5/2 not available")
IE_END(ms_classmark)

IE_BEGIN( mobile_station_classmark_1)
    BF( 1, 0, ACT_CHECK, ANONYMOUS, "spare")
    BF( 2, M2(0,1) , ACT_CHECK, rev_lev, "Used by phase 2 MSs")
    BF( 1, 0, ACT_CHECK, es_ind, SILENT)
    BF( 1, 0, ACT_CHECK, a5_1, "algorithm A5/1 available")
    BF( 3, M3(0,0,1) , ACT_CHECK, rf_pow_cap, "Class 2" )
IE_END( mobile_station_classmark_1)

IE_BEGIN(mobile_allocation)
    BF(8,0,ACT_CHECK,length,"length 0 due to hopping disabled")
IE_END(mobile_allocation)
IE_BEGIN(ia_rest_octets) /* maximum length (11), no hop, no start time */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 0 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 1 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 2 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 3 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 4 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 5 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 6 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 7 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 8 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 9 */
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 10 */
IE_END(ia_rest_octets)

IE_BEGIN(channel_description)
    BF( 5,M5(0,0,1,0,1),ACT_CHECK, channel_type,"SDCCH/SACCH 4(1) ")
    BF( 3, 0,ACT_CHECK, time_slot_number,"zero")
    BF( 3, BCC,ACT_CHECK, training_sequence_code,"same as BCCH")
    BF( 1, 0,ACT_CHECK, hopping,"No")
    BF( 2, 0,ACT_CHECK, spare,SILENT)
    BF(10, ARFCN_BCCH,ACT_CHECK, arfcn,"ARFCN of the BCCH")

```



```

IE_END(channel_description)

IE_BEGIN(channel_description_tch)
    BF( 5,M5(0,0,0,0,1),ACT_CHECK,channel_type,"TCH Full Rate")
    BF( 3,0,ACT_CHECK,time_slot_number,"zero ")
    BF( 3,BCC,ACT_CHECK,tsc,"same as BCCH ")
    BF( 1,0,ACT_CHECK,hopping,"No ")
    BF( 2,0,ACT_CHECK,spare,SILENT )
    BF(10,ARFCN_BCCH,ACT_CHECK,arfcn,"ARFCN of the BCCH ")
IE_END(channel_description_tch)

IE_BEGIN(mobility_management_protocol_discriminator)
    BF(4, 5,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(mobility_management_protocol_discriminator)

IE_BEGIN(identity_type)
    BF(4,1,ACT_CHECK,type_of_identity,"IMSI")
IE_END(identity_type)

IE_BEGIN(cm_service_type_moc)
    BF(4,M4(0,0,0,1),ACT_CHECK,service_type,"mobile originated call")
IE_END(cm_service_type_moc)

IE_BEGIN(cm_service_type_ss)
    BF(4,M4(1,0,0,0),ACT_CHECK,service_type,"supplementary services")
IE_END(cm_service_type_ss)

IE_BEGIN( cm_service_type_ec )
    BF( 4,M4( 0,0,1,0 ),ACT_CHECK,service,"Emergency Call establishment")
IE_END( cm_service_type_ec )

IE_BEGIN(authentication_request_message_type)
    BF(1,0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x12,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(authentication_request_message_type)

IE_BEGIN(cm_service_request_message_type)
    BF(1,0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x24,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(cm_service_request_message_type)

IE_BEGIN(identity_request_message_type)
    BF(1,0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x18,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(identity_request_message_type)

IE_BEGIN(identity_response_message_type)
    BF(1,0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,0,ACT_SHOW, ANONYMOUS,SILENT)
    BF(6, 0x19,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(identity_response_message_type)

IE_BEGIN(identity_response_0_message_type)
    BF(1,0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,0,ACT_CHECK, ANONYMOUS,SILENT)
    BF(6, 0x19,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(identity_response_0_message_type)

```

```

IE_BEGIN(identity_response_1_message_type)
    BF(1, 0, ACT_CHECK, ANONYMOUS, SILENT)
    BF(1, 1, ACT_CHECK, ANONYMOUS, SILENT)
    BF(6, 0x19, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(identity_response_1_message_type)

IE_BEGIN(rach_reest)
    BF(3, M3(1,1,0), ACT_CHECK, establishment_cause, "LUP")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_reest)
IE_BEGIN(rach_detach_111)
    BF(3, M3(1,1,1), ACT_CHECK, establishment_cause, "LUP")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_detach_111)
IE_BEGIN(rach_reest_neci_0)
    BF(3, M3(1,1,0), ACT_CHECK, establishment_cause, "Reest, NECI=0, TCH/F")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_reest_neci_0)
IE_BEGIN(rach_111)
    BF(3, M3(1,1,1), ACT_CHECK, establishment_cause, "SS")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_111)
IE_BEGIN(rach_lup_000)
    BF(3, M3(0,0,0), ACT_CHECK, establishment_cause, "LUP")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_lup_000)
IE_BEGIN(rach_detach_0001)
    BF(4, M4(0,0,0,1), ACT_CHECK, establishment_cause, "LUP")
    BF(4, M4(1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_detach_0001)
IE_BEGIN(rach_lup_0000)
    BF(4, M4(0,0,0,0), ACT_CHECK, establishment_cause, "LUP")
    BF(4, M4(1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_lup_0000)
IE_BEGIN(rach_moc)
    BF(3, M3(1,1,1), ACT_CHECK, establishment_cause, "MOC & TCH/F")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_moc)
IE_BEGIN(rach)
    BF(3, M3(1,0,0), ACT_CHECK, establishment_cause, "paging ind. any chan'")
    BF(5, M5(1,1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach)
IE_BEGIN(rach_0001)
    BF(4, M4(0,0,0,1), ACT_CHECK, establishment_cause, "paging ind. sdcch'")
    BF(4, M4(1,1,1,1), ACT_NOP, random_reference, "ignore Random Ref.")
IE_END(rach_0001)
IE_BEGIN(spare_half_octet)
    BF(4, 0, ACT_CHECK, ANONYMOUS, SILENT)
IE_END(spare_half_octet)

IE_BEGIN(timing_advance)
    BF(2, 0, ACT_CHECK, spare, SILENT)
    BF(6, 0, ACT_CHECK, timing_advance, "0" )
IE_END(timing_advance)
IE_BEGIN(pl_rest_octets)
/* pag. req. type1 : 22 - 11 (L2 pseud. len) = 11 bytes */
    BF(8, REST_OCTET, ACT_CHECK, ANONYMOUS, SILENT) /* 0 */
    BF(8, REST_OCTET, ACT_CHECK, ANONYMOUS, SILENT) /* 1 */
    BF(8, REST_OCTET, ACT_CHECK, ANONYMOUS, SILENT) /* 2 */
    BF(8, REST_OCTET, ACT_CHECK, ANONYMOUS, SILENT) /* 3 */

```

```

        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 4 */
        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 5 */
        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 6 */
        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 7 */
        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 8 */
        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 9 */
        BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,SILENT) /* 10 */
    IE_END(pl_rest_octets)

    IE_BEGIN(mobile_identity) /* has 8 octets */
        BF(8,          7,ACT_CHECK, length,"seven octets to come")
        BF(4,          2,ACT_CHECK, digit_1,SILENT)
        BF(1,          1,ACT_CHECK, odd_even,SILENT)
        BF(3,  M3(0,0,1),ACT_CHECK, type,"IMSI")
        BF(4,          2,ACT_CHECK, digit_3,SILENT)
        BF(4,          6,ACT_CHECK, digit_2,SILENT)
        BF(4,          1,ACT_CHECK, digit_4,SILENT)
        BF(4,          0,ACT_CHECK, digit_3,SILENT)
        BF(4,          7,ACT_CHECK, digit_6,SILENT)
        BF(4,          4,ACT_CHECK, digit_5,SILENT)
        BF(4,          1,ACT_CHECK, digit_8,SILENT)
        BF(4,          1,ACT_CHECK, digit_7,SILENT)
        BF(4,          9,ACT_CHECK, digit_10,SILENT)
        BF(4,          4,ACT_CHECK, digit_9,SILENT)
        BF(4,          2,ACT_CHECK, digit_12,SILENT)
        BF(4,          1,ACT_CHECK, digit_11,SILENT)
    IE_END(mobile_identity)

    IE_BEGIN(channels_needed_for_mobiles_1_and_2)
        BF(2,0,ACT_CHECK,second_channel,"spare, any channel")
        BF(2,0,ACT_CHECK, first_channel,"spare, any channel")
    IE_END(channels_needed_for_mobiles_1_and_2)

    IE_BEGIN(channels_needed_any_channel)
        BF(2,0,ACT_CHECK,second_channel,"spare, any channel")
        BF(2,0,ACT_CHECK, first_channel,"spare, any channel")
    IE_END(channels_needed_any_channel)

    IE_BEGIN(channels_needed_sdcch)
        BF(2,1,ACT_CHECK,second_channel,"sdcch")
        BF(2,1,ACT_CHECK, first_channel,"sdcch")
    IE_END(channels_needed_sdcch)

    IE_BEGIN(channels_needed_tch_full)
        BF(2,2,ACT_CHECK,second_channel,"tch full")
        BF(2,2,ACT_CHECK, first_channel,"tch full")
    IE_END(channels_needed_tch_full)

    IE_BEGIN(channels_needed_tch_full_or_half)
        BF(2,3,ACT_CHECK,second_channel,"tch full or half")
        BF(2,3,ACT_CHECK, first_channel,"tch full or half")
    IE_END(channels_needed_tch_full_or_half)

    IE_BEGIN(transaction_identifier_dest)
        BF(4,M4(1,0,0,0),ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(transaction_identifier_dest)

    IE_BEGIN(transaction_identifier_dest_1)
        BF(4,M4(1,0,0,1),ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(transaction_identifier_dest_1)

```

```

IE_BEGIN(call_control_protocol_discriminator)
    BF(4, 3,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(call_control_protocol_discriminator)

IE_BEGIN(supplementary_services_protocol_discriminator)
    BF(4, M4(1,0,1,1),ACT_CHECK,ANONYMOUS,SILENT)
IE_END(supplementary_services_protocol_discriminator)

IE_BEGIN(l2_pseudo_length_11) /* pag req type 1 with TMSI (one mobile)*/
    BF(8, 0,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(l2_pseudo_length_11)

IE_BEGIN(ncc_permitted)
    BF(8,0xFF,ACT_CHECK,ncc_permit,"e.g. all NCCs permitted")
IE_END(ncc_permitted)
IE_BEGIN(rach_control_parameter)
    BF( 2,0,ACT_CHECK,max_retrans           ,"Any Value  ")
    BF( 4,0,ACT_CHECK,tx_integer            ,"Any Value  ")
    BF( 1,0,ACT_CHECK,cell_bar_access       ,"Not barred ")
    BF( 1,1,ACT_CHECK,call_re_establishment ,"Not Allowed")
    BF( 5,0,ACT_CHECK,access_control_class_15_11 ,"None Barred")
    BF( 1,0,ACT_CHECK,emergency_call        ,"Allowed   ")
    BF(10,0,ACT_CHECK,access_control_class_09_00 ,"None Barred")
IE_END(rach_control_parameter)

IE_BEGIN(rach_control_parameter_reest)
    BF( 2,0,ACT_CHECK,max_retrans           ,"Any Value  ")
    BF( 4,0,ACT_CHECK,tx_integer            ,"Any Value  ")
    BF( 1,0,ACT_CHECK,cell_bar_access       ,"Not barred ")
    BF( 1,0,ACT_CHECK,call_re_establishment ,"Allowed")
    BF( 5,0,ACT_CHECK,access_control_class_15_11 ,"None Barred")
    BF( 1,0,ACT_CHECK,emergency_call        ,"Allowed   ")
    BF(10,0,ACT_CHECK,access_control_class_09_00 ,"None Barred")
IE_END(rach_control_parameter_reest)

IE_BEGIN(rr_management_protocol_discriminator)
    BF(4, 6,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(rr_management_protocol_discriminator)

IE_BEGIN(skip_indicator)
    BF(4, 0,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(skip_indicator)

IE_BEGIN(si_1_rest_octets)
    BF(8,REST_OCTET,ACT_CHECK,ANONYMOUS,"Spare Octets")
IE_END(si_1_rest_octets)

IE_BEGIN(si_3_rest_octets) /* optionally contains cell (re)select params */
    BF(1, 0,ACT_CHECK,p1 ,"C2 parameters not present")
    BF(7,REST_OCTET & 0x7F,ACT_CHECK,ANONYMOUS,SILENT )
    BF(8,REST_OCTET ,ACT_CHECK,ANONYMOUS,SILENT )
    BF(8,REST_OCTET ,ACT_CHECK,ANONYMOUS,SILENT )
    BF(8,REST_OCTET ,ACT_CHECK,ANONYMOUS,SILENT )
IE_END(si_3_rest_octets)
IE_BEGIN(si_4_rest_octets) /* optionally contains cell (re)select params */
    BF(1, 0,ACT_CHECK,p1 ,"C2 parameters not present")
    BF(7,REST_OCTET & 0x7F,ACT_CHECK,ANONYMOUS,SILENT )
    BF(8,REST_OCTET ,ACT_CHECK,ANONYMOUS,SILENT )
    BF(8,REST_OCTET ,ACT_CHECK,ANONYMOUS,SILENT )

```

```

        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
        BF(8,REST_OCTET      ,ACT_CHECK,ANONYMOUS,SILENT      )
    IE_END(si_4_rest_octets)
    IE_BEGIN(system_information_type_1_message_type)
        BF(8, 0x19,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(system_information_type_1_message_type)
    IE_BEGIN(system_information_type_2_message_type)
        BF(8, 0x1A,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(system_information_type_2_message_type)
    IE_BEGIN(system_information_type_3_message_type)
        BF(8, 0x1B,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(system_information_type_3_message_type)
    IE_BEGIN(system_information_type_4_message_type)
        BF(8, 0x1C,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(system_information_type_4_message_type)
    IE_BEGIN(system_information_type_5_message_type)
        BF(8, 0x1D,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(system_information_type_5_message_type)
    IE_BEGIN(system_information_type_6_message_type)
        BF(8, 0x1E,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(system_information_type_6_message_type)

    IE_BEGIN(paging_request_type_1_message_type)
        BF(8, 0x21,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(paging_request_type_1_message_type)
    IE_BEGIN(immediate_assignment_message_type)
        BF(8, 0x3F,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(immediate_assignment_message_type)

    IE_BEGIN( immediate_assignment_reject_message_type )
        BF( 8, 0x3A,ACT_CHECK,ANONYMOUS,SILENT )
    IE_END( immediate_assignment_reject_message_type )

    IE_BEGIN(paging_response_message_type)
        BF(8, 0x27,ACT_CHECK,ANONYMOUS,SILENT)
    IE_END(paging_response_message_type)

    IE_BEGIN( request_reference_2 )
        BF( 3, M3(1,0,0),      ACT_CHECK, random_access_info,SILENT )
        BF( 5, M5(1,1,1,1,1 ), ACT_CHECK, ref, SILENT)
        BF( 5, 0,              ACT_CHECK, t1, SILENT )
        BF( 6, 0,              ACT_CHECK, t3, SILENT )
        BF( 5, 0,              ACT_CHECK, t2, SILENT )
    IE_END( request_reference_2 )

/*-----*\
| Messages
\*-----*/
MSG3_BEGIN(system_information_type_1)
    IE(l2_pseudo_length_21)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_1_message_type)
    IE(cell_channel_description)

```

```
    IE(rach_control_parameter)
    IE(si_1_rest_octets)
MSG3_END(system_information_type_1)
MSG3_BEGIN(system_information_type_2)
    IE(l2_pseudo_length_22)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_2_message_type)
    IE(bcch_frequency_list)
    IE(ncc_permitted)
    IE(rach_control_parameter)
MSG3_END(system_information_type_2)
MSG3_BEGIN(system_information_type_3)
    IE(l2_pseudo_length_18)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_3_message_type)
    IE(cell_identity)
    IE(location_area_identification)
    IE(control_channel_description)
    IE(cell_options)
    IE(cell_selection_parameter)
    IE(rach_control_parameter)
    IE(si_3_rest_octets)
MSG3_END(system_information_type_3)
MSG3_BEGIN(system_information_type_4)
    IE(l2_pseudo_length_12)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_4_message_type)
    IE(location_area_identification)
    IE(cell_selection_parameter)
    IE(rach_control_parameter)
    IE(si_4_rest_octets)
MSG3_END(system_information_type_4)
MSG3_BEGIN(system_information_type_1_reest)
    IE(l2_pseudo_length_21)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_1_message_type)
    IE(cell_channel_description)
    IE(rach_control_parameter_reest)
    IE(si_1_rest_octets)
MSG3_END(system_information_type_1_reest)
MSG3_BEGIN(system_information_type_2_reest)
    IE(l2_pseudo_length_22)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_2_message_type)
    IE(bcch_frequency_list)
    IE(ncc_permitted)
    IE(rach_control_parameter_reest)
MSG3_END(system_information_type_2_reest)
MSG3_BEGIN(system_information_type_3_reest)
    IE(l2_pseudo_length_18)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_3_message_type)
    IE(cell_identity)
    IE(location_area_identification)
```

```

    IE(control_channel_description)
    IE(cell_options)
    IE(cell_selection_parameter)
    IE(rach_control_parameter_reest)
    IE(si_3_rest_octets)
MSG3_END(system_information_type_3_reest)
MSG3_BEGIN(system_information_type_4_reest)
    IE(l2_pseudo_length_12)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_4_message_type)
    IE(location_area_identification)
    IE(cell_selection_parameter)
    IE(rach_control_parameter_reest)
    IE(si_4_rest_octets)
MSG3_END(system_information_type_4_reest)
MSG3_BEGIN(system_information_type_5)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_5_message_type)
    IE(bcch_frequency_list)
MSG3_END(system_information_type_5)
MSG3_BEGIN(system_information_type_6)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(system_information_type_6_message_type)
    IE(cell_identity)
    IE(location_area_identification)
    IE(cell_options)
    IE(ncc_permitted)
MSG3_END(system_information_type_6)
/*-----*/
| GSM 11.10
| 10 Generic call set up procedure
| 10.1 Generic call setup-up procedure for mobile terminating speech calls
| 10.1.4 Specific message contents
/*-----*/
/*-----*/
| Information Elements
/*-----*/
IE_BEGIN( emergency_setup_message_type )
    BF( 1,      0,ACT_CHECK, ANONYMOUS, SILENT)
    BF( 1,      0,ACT_SHOW,  ANONYMOUS, SILENT)
    BF( 6, 0x0E,ACT_CHECK,ANONYMOUS,SILENT )
IE_END( emergency_setup_message_type )

IE_BEGIN(setup_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW,  ANONYMOUS,SILENT)
    BF(6, 0x05,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(setup_message_type)

IE_BEGIN(call_confirmed_message_type)
    BF(1,      0,ACT_CHECK,ANONYMOUS,SILENT)
    BF(1,      0,ACT_SHOW,  ANONYMOUS,SILENT)
    BF(6, 0x08,ACT_CHECK,ANONYMOUS,SILENT)
IE_END(call_confirmed_message_type)

MSG3_BEGIN(ciphering_mode_command)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)

```

```
    IE(ciphering_mode_command_message_type)
    IE(cipher_response)
    IE(ciphering_mode_setting)
MSG3_END(ciphering_mode_command)

MSG3_BEGIN(ciphering_mode_complete)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(ciphering_mode_complete_message_type)
MSG3_END(ciphering_mode_complete)

MSG3_BEGIN(cm_reestablish_request)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(cm_reestablish_request_message_type)
    IE(spare_half_octet)
    IE(ciphering_key_sequence_number)
    IE(ms_classmark)
    IE(mobile_identity)
MSG3_END(cm_reestablish_request)

MSG3_BEGIN(setup) /* contains 'signal' but no 'Bearer Cap' */
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
MSG3_END(setup)

MSG3_BEGIN(release_complete_88)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(release_complete_message_type)
    IE(iei_08)
    IE(cause_88)
MSG3_END(release_complete_88)

MSG3_BEGIN(setup_data)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
    IE(iei_04)
    IE(bearer_capability_data)
    IE(iei_5E)
    IE(called_party_bcd_number)
MSG3_END(setup_data)

MSG3_BEGIN(setup_ts_61)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
    IE(repeat_indicator_circular)
    IE(iei_04)
    IE(bearer_capability)
    IE(iei_04)
    IE(bearer_capability_ts_61)
    IE(iei_5E)
    IE(called_party_bcd_number)
MSG3_END(setup_ts_61)

MSG3_BEGIN(setup_bs_61)
    IE(transaction_identifier_source)
```



```
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
    IE(repeat_indicator_circular)
    IE(iei_04)
    IE(bearer_capability)
    IE(iei_04)
    IE(bearer_capability_bs_61)
    IE(iei_5E)
    IE(called_party_bcd_number)
MSG3_END(setup_bs_61)

MSG3_BEGIN(setup_no_bc)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
MSG3_END(setup_no_bc)

MSG3_BEGIN(call_confirmed)          /* contains bearer capability */
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(call_confirmed_message_type)
    IE(iei_04)
    IE(bearer_capability)
MSG3_END(call_confirmed)

MSG3_BEGIN(disconnect)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(disconnect_message_type)
    IE(cause_16)
MSG3_END(disconnect)

MSG3_BEGIN(disconnect_moc)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(disconnect_message_type)
    IE(cause_16)
MSG3_END(disconnect_moc)

MSG3_BEGIN(disconnect_1)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
    IE(disconnect_message_type)
    IE(cause_16)
MSG3_END(disconnect_1)

MSG3_BEGIN(disconnect_acm)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(disconnect_message_type)
    IE(cause_44)
MSG3_END(disconnect_acm)

MSG3_BEGIN(disconnect_acm_mtc)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(disconnect_message_type)
    IE(cause_44)
MSG3_END(disconnect_acm_mtc)
```

```
MSG3_BEGIN(cc_hold)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(hold_message_type)
MSG3_END(cc_hold)

MSG3_BEGIN(cc_hold_ack)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(hold_ack_message_type)
MSG3_END(cc_hold_ack)

MSG3_BEGIN(disconnect_mtc)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(disconnect_message_type)
    IE(cause_16)
MSG3_END(disconnect_mtc)

MSG3_BEGIN(release_complete)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(release_complete_message_type)
MSG3_END(release_complete)

MSG3_BEGIN(release_complete_moc)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(release_complete_message_type)
MSG3_END(release_complete_moc)

MSG3_BEGIN(release_complete_1)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
    IE(release_complete_message_type)
MSG3_END(release_complete_1)

MSG3_BEGIN(release_mtc)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(release_message_type)
MSG3_END(release_mtc)
```

```
MSG3_BEGIN(release_complete_mtc)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(release_complete_message_type)
MSG3_END(release_complete_mtc)

MSG3_BEGIN(release)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(release_message_type)
MSG3_END(release)

MSG3_BEGIN(release_moc)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(release_message_type)
MSG3_END(release_moc)

MSG3_BEGIN(release_1)
    IE(transaction_identifier_source_1)
    IE(call_control_protocol_discriminator)
    IE(release_message_type)
MSG3_END(release_1)

MSG3_BEGIN(facility_msg_aocc_acmmmax1)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_acmmmax1)
MSG3_END(facility_msg_aocc_acmmmax1)

MSG3_BEGIN(facility_msg_aocc_zero_cai)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_zero_cai)
MSG3_END(facility_msg_aocc_zero_cai)

MSG3_BEGIN(facility_msg_aocc_k1)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_k1_mtc)
MSG3_END(facility_msg_aocc_k1)

MSG3_BEGIN(facility_msg_aocc_k2)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_k2_mtc)
MSG3_END(facility_msg_aocc_k2)

MSG3_BEGIN(facility_msg_aocc_k3)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_k3_mtc)
MSG3_END(facility_msg_aocc_k3)

MSG3_BEGIN(facility_msg_aocc_k4)
```

```
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_k4_mtc)
MSG3_END(facility_msg_aocc_k4)

MSG3_BEGIN(facility_msg_aocc_k5)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_k5_mtc)
MSG3_END(facility_msg_aocc_k5)

MSG3_BEGIN(facility_msg_aocc_set_1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_set_1)
MSG3_END(facility_msg_aocc_set_1)

MSG3_BEGIN(facility_msg_aocc_set_2)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_set_2)
MSG3_END(facility_msg_aocc_set_2)

MSG3_BEGIN(facility_msg_aocc_diff_1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_diff_1)
MSG3_END(facility_msg_aocc_diff_1)

MSG3_BEGIN(facility_msg_aocc_diff_2)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_diff_2)
MSG3_END(facility_msg_aocc_diff_2)

MSG3_BEGIN(facility_mpty_invoke)
    IE(transaction_identifier_source_1)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_ie_mpty_invoke)
MSG3_END(facility_mpty_invoke)

MSG3_BEGIN(facility_mpty_return_result)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_ie_mpty_return_result)
MSG3_END(facility_mpty_return_result)

MSG3_BEGIN(paging_request_type_1)
    IE(l2_pseudo_length_11)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(paging_request_type_1_message_type)
```

```
    IE(channels_needed_for_mobiles_1_and_2)
    IE(page_mode)
    IE(mobile_identity)
    IE(p1_rest_octets)
MSG3_END(paging_request_type_1)

MSG3_BEGIN(paging_request_type_1_any_channel)
    IE(l2_pseudo_length_11)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(paging_request_type_1_message_type)
    IE(channels_needed_any_channel)
    IE(page_mode)
    IE(mobile_identity)
    IE(p1_rest_octets)
MSG3_END(paging_request_type_1_any_channel)

MSG3_BEGIN(paging_request_type_1_sdcch)
    IE(l2_pseudo_length_11)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(paging_request_type_1_message_type)
    IE(channels_needed_sdcch)
    IE(page_mode)
    IE(mobile_identity)
    IE(p1_rest_octets)
MSG3_END(paging_request_type_1_sdcch)

MSG3_BEGIN(paging_request_type_1_tch_full)
    IE(l2_pseudo_length_11)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(paging_request_type_1_message_type)
    IE(channels_needed_tch_full)
    IE(page_mode)
    IE(mobile_identity)
    IE(p1_rest_octets)
MSG3_END(paging_request_type_1_tch_full)

MSG3_BEGIN(paging_request_type_1_tch_full_or_half)
    IE(l2_pseudo_length_11)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(paging_request_type_1_message_type)
    IE(channels_needed_tch_full_or_half)
    IE(page_mode)
    IE(mobile_identity)
    IE(p1_rest_octets)
MSG3_END(paging_request_type_1_tch_full_or_half)

MSG3_BEGIN(channel_request_ss)
    IE(rach_111)
MSG3_END(channel_request_ss)

MSG3_BEGIN(channel_request_moc)
    IE(rach_moc)
MSG3_END(channel_request_moc)

MSG3_BEGIN(channel_request)
    IE(rach)
```

```
MSG3_END(channel_request)

MSG3_BEGIN(channel_request_reest_neci_0)
    IE(rach_reest_neci_0)
MSG3_END(channel_request_reest_neci_0)

MSG3_BEGIN(immediate_assignment)
    IE(l2_pseudo_length_21)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(immediate_assignment_message_type)
    IE(spare_half_octet)
    IE(page_mode)
    IE(channel_description)
    IE(request_reference)
    IE(timing_advance)
    IE(mobile_allocation)
    IE(ia_rest_octets)
MSG3_END(immediate_assignment)
MSG3_BEGIN(paging_response)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(paging_response_message_type)
    IE(spare_half_octet)
    IE(ciphering_key_sequence_number)
    IE(ms_classmark)
    IE(mobile_identity)
MSG3_END(paging_response)

MSG3_BEGIN(immediate_assignment_tch)
    IE(l2_pseudo_length_21)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(immediate_assignment_message_type)
    IE(spare_half_octet)
    IE(page_mode)
    IE(channel_description_tch)
    IE(request_reference)
    IE(timing_advance)
    IE(mobile_allocation)
    IE(ia_rest_octets)
MSG3_END(immediate_assignment_tch)

MSG3_BEGIN(cm_service_request)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(cm_service_request_message_type)
    IE(ciphering_key_sequence_number)
    IE(cm_service_type_moc)
    IE(ms_classmark)
    IE(mobile_identity)
MSG3_END(cm_service_request)

MSG3_BEGIN( cm_service_request_ec )
    IE( skip_indicator )
    IE( mobility_management_protocol_discriminator )
    IE( cm_service_request_message_type )
    IE( ciphering_key_sequence_number )
    IE( cm_service_type_ec )
    IE( ms_classmark )
```

```
    IE( mobile_identity )
MSG3_END( cm_service_request_ec )

MSG3_BEGIN(cm_service_request_ss)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(cm_service_request_message_type)
    IE(ciphering_key_sequence_number)
    IE(cm_service_type_ss)
    IE(ms_classmark)
    IE(mobile_identity)
MSG3_END(cm_service_request_ss)

MSG3_BEGIN(identity_request)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(identity_request_message_type)
    IE(spare_half_octet)
    IE(identity_type)
MSG3_END(identity_request)
```

```
MSG3_BEGIN(identity_response)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(identity_response_message_type)
    IE(mobile_identity)
MSG3_END(identity_response)

MSG3_BEGIN(identity_response_0)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(identity_response_0_message_type)
    IE(mobile_identity)
MSG3_END(identity_response_0)

MSG3_BEGIN(identity_response_1)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(identity_response_1_message_type)
    IE(mobile_identity)
MSG3_END(identity_response_1)

MSG3_BEGIN( emergency_setup )          /* Ref.: [1], §9.3.8 */
    IE( transaction_identifier_source )
    IE( call_control_protocol_discriminator )
    IE( emergency_setup_message_type )
    IE( iei_04 )
    IE( bearer_capability )
MSG3_END( emergency_setup )

MSG3_BEGIN(setup_moc)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
    IE(iei_04)
    IE(bearer_capability)
    IE(iei_5E)
    IE(called_party_bcd_number)
    IE(iei_15)
    IE(cc_capabilities)
MSG3_END(setup_moc)
MSG3_BEGIN(setup_moc_1)
    IE(transaction_identifier_source_1)
    IE(call_control_protocol_discriminator)
    IE(setup_message_type)
    IE(iei_04)
    IE(bearer_capability)
    IE(iei_5E)
    IE(called_party_bcd_number)
MSG3_END(setup_moc_1)
MSG3_BEGIN(authentication_request)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(authentication_request_message_type)
    IE(spare_half_octet)
    IE(ciphering_key_sequence_number_2)
    IE(authentication_parameter_rand)
MSG3_END(authentication_request)
MSG3_BEGIN(authentication_response)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(authentication_response_message_type)
```



```
    IE(authentication_parameter_sres)
MSG3_END(authentication_response)
MSG3_BEGIN(call_proceeding)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(call_proceeding_message_type)
MSG3_END(call_proceeding)
```

```
MSG3_BEGIN(call_proceeding_1)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
    IE(call_proceeding_message_type)
MSG3_END(call_proceeding_1)
MSG3_BEGIN(assignment_command)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(assignment_command_message_type)
    IE(description_of_the_first_channel_after_time)
    IE(power_command)
    IE(iei_63)
    IE(mode_of_the_first_channel)
MSG3_END(assignment_command)
MSG3_BEGIN(assignment_complete)
    IE(skip_indicator)
    IE(rr_management_protocol_discriminator)
    IE(assignment_complete_message_type)
    IE(rr_cause)
MSG3_END(assignment_complete)

MSG3_BEGIN(connect_acknowledge)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(connect_acknowledge_message_type)
MSG3_END(connect_acknowledge)

MSG3_BEGIN(connect_acknowledge_1)
    IE(transaction_identifier_source_1)
    IE(call_control_protocol_discriminator)
    IE(connect_acknowledge_message_type)
MSG3_END(connect_acknowledge_1)

MSG3_BEGIN(alerting)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(alerting_message_type)
MSG3_END(alerting)

MSG3_BEGIN(alerting_1)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
    IE(alerting_message_type)
MSG3_END(alerting_1)

MSG3_BEGIN(connect_aocc_storage1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_storage1)
MSG3_END(connect_aocc_storage1)

MSG3_BEGIN(connect_aocc_acmmmax1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_acmmmax1)
MSG3_END(connect_aocc_acmmmax1)
```

```
MSG3_BEGIN(connect_aocc_zero_cai)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_zero_cai)
MSG3_END(connect_aocc_zero_cai)

MSG3_BEGIN(connect_aocc_k1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_k1)
MSG3_END(connect_aocc_k1)

MSG3_BEGIN(connect_aocc_k2)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_k2)
MSG3_END(connect_aocc_k2)

MSG3_BEGIN(connect_aocc_k3)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_k3)
MSG3_END(connect_aocc_k3)

MSG3_BEGIN(connect_aocc_k4)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_k4)
MSG3_END(connect_aocc_k4)

MSG3_BEGIN(connect_aocc_k5)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_k5)
MSG3_END(connect_aocc_k5)

MSG3_BEGIN(connect_aocc_hold1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_hold1)
MSG3_END(connect_aocc_hold1)

MSG3_BEGIN(connect_aocc_hold2)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
```

```
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_hold2)
MSG3_END(connect_aocc_hold2)

MSG3_BEGIN(connect_aocc_mpty1)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_mpty1)
MSG3_END(connect_aocc_mpty1)

MSG3_BEGIN(connect_aocc_mpty2)
    IE(transaction_identifier_dest_1)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
    IE(iei_1C)
    IE(facility_aocc_mpty2)
MSG3_END(connect_aocc_mpty2)

MSG3_BEGIN(connect)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(connect_message_type)
MSG3_END(connect)

MSG3_BEGIN(facility_aocc_return_result)
    IE(transaction_identifier_source)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_cnf)
MSG3_END(facility_aocc_return_result)

MSG3_BEGIN(facility_aocc_return_result_1)
    IE(transaction_identifier_source_1)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_cnf)
MSG3_END(facility_aocc_return_result_1)

MSG3_BEGIN(facility_aocc_return_result_mtc)
    IE(transaction_identifier_dest)
    IE(call_control_protocol_discriminator)
    IE(facility_message_type)
    IE(facility_aocc_cnf)
MSG3_END(facility_aocc_return_result_mtc)

MSG3_BEGIN( channel_release ) /* Ref.: [1], §9.1.7; [2], §26.6.14 */
    IE( skip_indicator )
    IE( rr_management_protocol_discriminator )
    IE( channel_release_message_type )
    IE( rr_cause )
MSG3_END( channel_release )

MSG3_BEGIN(cm_service_accept)
    IE(skip_indicator)
    IE(mobility_management_protocol_discriminator)
    IE(cm_service_accept_message_type)
```

```
MSG3_END(cm_service_accept)

MSG3_BEGIN(ss_register_100A)
    IE(transaction_identifier_source)
    IE(supplementary_services_protocol_discriminator)
    IE(ss_register_message_type)
    IE(iei_1C)
    IE(facility_100A_req)
    IE(iei_7F)
    IE(ss_version)
MSG3_END(ss_register_100A)

MSG3_BEGIN(ss_release_complete_100A)
    IE(transaction_identifier_dest)
    IE(supplementary_services_protocol_discriminator)
    IE(ss_release_complete_message_type)
    IE(iei_1C)
    IE(facility_100A_cnf)
MSG3_END(ss_release_complete_100A)

MSG3_BEGIN(ss_register_100B)
    IE(transaction_identifier_source)
    IE(supplementary_services_protocol_discriminator)
    IE(ss_register_message_type)
    IE(iei_1C)
    IE(facility_100B_req)
    IE(iei_7F)
    IE(ss_version)
MSG3_END(ss_register_100B)

MSG3_BEGIN(ss_release_complete_100B)
    IE(transaction_identifier_dest)
    IE(supplementary_services_protocol_discriminator)
    IE(ss_release_complete_message_type)
    IE(iei_1C)
    IE(facility_100B_cnf)
MSG3_END(ss_release_complete_100B)

MSG3_BEGIN(ss_register_deactBAOC)
    IE(transaction_identifier_source)
    IE(supplementary_services_protocol_discriminator)
    IE(ss_register_message_type)
    IE(iei_1C)
    IE(facility_deactBAOC_req)
    IE(iei_7F)
    IE(ss_version)
MSG3_END(ss_register_deactBAOC)

MSG3_BEGIN(ss_release_complete_deactBAOC)
    IE(transaction_identifier_dest)
    IE(supplementary_services_protocol_discriminator)
    IE(ss_release_complete_message_type)
    IE(iei_1C)
    IE(facility_deactBAOC_cnf)
MSG3_END(ss_release_complete_deactBAOC)
```

4 TEST CASES

4.1 Preambles

4.1.1 MSS001: Power On

Description: This test describes the initialization of environment and activation of the Mobile Station.

Preamble: None

Script:

```
ISS_INIT (4);

BS_SET_SYS_INFO ( 0 , system_information_type_1 );
BS_SET_SYS_INFO ( 0 , system_information_type_2 );
BS_SET_SYS_INFO ( 0 , system_information_type_3 );
BS_SET_SYS_INFO ( 0 , system_information_type_4 );
BS_SET_SYS_INFO_SACCH ( 0 , system_information_type_5 );
BS_SET_SYS_INFO_SACCH ( 0 , system_information_type_6 );

BS_SET_SCH ( 0 , BSIC , RFN );
BS_SET_ARFCN ( 0 , ARFCN_BCCH );
BS_SET_POWER ( 0 , -50 );
BS_ON_OFF ( 0 , TRUE );

COMMAND ("PL CONFIG STD=5"); /* use Dualband Vers.*/

AT_SEND ("AT+CFUN=1;+COPS=0\r\n", SILENT);
AT_RECEIVE("OK", SILENT);

ISS_DELAY (20000);
```

History: 17.12.97 LE Initial

4.1.2 MSS002: Power On (Reestablishment allowed)

Description: This test describes the initialization of environment and activation of the Mobile Station.

Preamble: None

Script:

```
ISS_INIT (4);

BS_SET_SYS_INFO ( 0 , system_information_type_1_reest );
BS_SET_SYS_INFO ( 0 , system_information_type_2_reest );
BS_SET_SYS_INFO ( 0 , system_information_type_3_reest );
BS_SET_SYS_INFO ( 0 , system_information_type_4_reest );
BS_SET_SYS_INFO_SACCH ( 0 , system_information_type_5 );
BS_SET_SYS_INFO_SACCH ( 0 , system_information_type_6 );

BS_SET_SCH ( 0 , BSIC , RFN );
BS_SET_ARFCN ( 0 , ARFCN_BCCH );
BS_SET_POWER ( 0 , -50 );
BS_ON_OFF ( 0 , TRUE );

COMMAND ("PL CONFIG STD=5"); /* use Dualband Vers.*/

AT_SEND ("AT+CFUN=1;+COPS=0\r\n", SILENT);
AT_RECEIVE("OK", SILENT);

ISS_DELAY (20000);
```

History: 17.12.97 LE Initial

4.2 Call Forwarding Tests

4.2.1 MSS100: Call Forwarding, Registration accepted (31.2.1.1.1)

Description: A registration of a call forwarding service is started. It is accepted by the network.

The first CCFC command has the following parameters:

2 CFNRy
3 SS Register
431234 Forwarded To Number
145 International Number
1 Voice Services
5 Reply time

The second CCFC command has the following parameters:

0 CFU
3 SS Register
431234 Forwarded To Number
145 International Number
4 Fax Services
0 No Reply time

Preamble: MSS001

Script:

```
AT_SEND ("ATD**61*00431234*11*5#\r\n",SILENT) ;

BS_RACH_AWAIT(0,channel_request_ss,SILENT) ;

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0) ;
BS_STORE_RACH_PARAMS (0, 0) ;
BS_MSG3_SEND (0,immediate_assignment,SILENT) ;

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0) ;
BS_MSG3_AWAIT(0,cm_service_request_ss,SILENT) ;
BS_MSG3_SEND (0,cm_service_accept,SILENT) ;

BS_MSG3_AWAIT(0,ss_register_100A,SILENT) ;
BS_MSG3_SEND (0,ss_release_complete_100A,SILENT) ;

AT_RECEIVE ("OK",SILENT) ;

BS_MSG3_SEND (0,channel_release,SILENT) ;
```



```
ISS_DELAY (5000);

AT_SEND ("ATD**21*00431234*13#\r\n",SILENT);

BS_RACH_AWAIT(0,channel_request_ss,SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request_ss,SILENT);
BS_MSG3_SEND (0,cm_service_accept,SILENT);

BS_MSG3_AWAIT(0,ss_register_100B,SILENT);
BS_MSG3_SEND (0,ss_release_complete_100B,SILENT);

AT_RECEIVE ("OK",SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);
```

History: 17.02.98 LE Initial

4.3 Advice of Charge Tests

4.3.1 MSS200: AoC time related charging, MOC, k=1 (31.6.1.1)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second, stores the correct value in the ACM field of the SIM and that the MS ignores non-zero e5, e6 parameters.
In case k=1 the parameters are set to e1=6, e2=14, e3=1, e4=25, e5=0, e6=0, e7=60. It is expected that after 90 seconds CCM has the value 43 and ACM is increased by 43.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n","call meter warning event");
AT_RECEIVE("OK","OK");

/*
 * Set ACMmax value to 000030, PIN2 is 7890
 */
AT_SEND ("AT+CMM=\\"000030\\","\\"7890\\\"\r\n","set ACMmax to 0x30");
AT_RECEIVE("OK","OK");

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n","read stored ACMmax");
AT_RECEIVE("+CMM: \\"000030\\\"", "+CMM: \\"000030\\\"");
AT_RECEIVE("OK","OK");

/*
 * Check CCM, it should be set to zero
 * +CAOC is used as the default is equal to +CAOC=0
 */
AT_SEND ("AT+CAOC\r\n","read CCM, +CAOC");
AT_RECEIVE("+CAOC: \\"000000\\\"", "+CAOC: \\"000000\\\"");
AT_RECEIVE("OK","OK");

AT_SEND ("ATD03039094117;\r\n","dail voice call");

BS_RACH_AWAIT(0,channel_request_moc,"CHANNEL REQUEST");
/*
 * the following OK should be commented out if problems occur
 * as there is a race condition between the OK and the RACH
 * access; the infrastructure simulation does not support co-regions
 * like MSCs of Z.120 do, unfortunately
 */
AT_RECEIVE("OK","OK");

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,"IMMEDIATE ASSIGNMENT");

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,"CM SERVICE REQUEST");
```

```
BS_MSG3_SEND (0,cm_service_accept, "CM SERVICE ACCEPT");

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_k1,SILENT);

BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (69000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

/* timer value should be corrected */
/* +CCWV is used as trigger which is not accurate enough */
ISS_DELAY (16000+1500)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);
/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE("+CAOC: \"00002B\"",SILENT);
AT_RECEIVE("OK",SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"00002B\"",SILENT);
AT_RECEIVE("OK",SILENT);
/*
 * Check Call Timer %CTV
 */
AT_SEND ("AT%CTV\r\n",SILENT);

History:          17.02.98          LE          Initial
                  23-Aug-01        SBK          Adapt to immediate OK after voice
call
```

4.3.2 MSS201: AoC time related charging, MOC, k=2 (31.6.1.1)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second, stores the correct value in the ACM field of the SIM and that the MS ignores non-zero e5, e6 parameters.

In case k=2 the parameters are set to e1=0, e2=0, e3=1, e4=100, e5=0, e6=0, e7=0. It is expected that after 90 seconds CCM has the value 100 and ACM is increased by 100.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"000065\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: \\"000065\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", SILENT);
BS_RACH_AWAIT(0, channel_request_moc, "CHANNEL REQUEST");
/*
 * the following OK should be commented out if problems occur
 * as there is a race condition between the OK and the RACH
 * access; the infrastructure simulation does not support co-regions
 * like MSCs of Z.120 do, unfortunately
 */
AT_RECEIVE ("OK", "OK");

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT);
BS_MSG3_SEND (0, call_proceeding, SILENT);
```

```
BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_k2,SILENT);

BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE (" +CAOC:  \"000064\"",SILENT);
AT_RECEIVE ("OK",SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM:  \"000064\"",SILENT);
AT_RECEIVE ("OK",SILENT);
```

```
/*
 * Check CCM with PUCT
 */
AT_SEND ("AT%CAOC\r\n", SILENT);
AT_RECEIVE ("%CAOC: \r\n", "\"5.50\"", SILENT);
AT_RECEIVE ("OK", SILENT);
/*
 * Check ACM with PUCT
 */
AT_SEND ("AT%CACM\r\n", SILENT);
AT_RECEIVE ("%CACM: \r\n", "\"5.50\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History:	17.02.98	LE	Initial
	23-Aug-01	SBK	Adapt to immediate OK after voice
call			

4.3.3 MSS202: AoC time related charging, MOC, k=3 (31.6.1.1)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second, stores the correct value in the ACM field of the SIM and that the MS ignores non-zero e5, e6 parameters.

In case k=3 the parameters are set to e1=250, e2=16, e3=2, e4=500, e5=0, e6=0, e7=60. It is expected that after 90 seconds CCM has the value 2000 and ACM is increased by 2000.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"000834\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: \\"000834\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", SILENT);
BS_RACH_AWAIT(0, channel_request_moc, "CHANNEL REQUEST");
/*
 * the following OK should be commented out if problems occur
 * as there is a race condition between the OK and the RACH
 * access; the infrastructure simulation does not support co-regions
 * like MSCs of Z.120 do, unfortunately
 */
AT_RECEIVE ("OK", "OK");

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT);
BS_MSG3_SEND (0, call_proceeding, SILENT);
```

```
BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_k3,SILENT);

BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (57000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

ISS_DELAY (28000)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);
/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE("+CAOC: \"0007D0\"",SILENT);
AT_RECEIVE("OK",SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"0007D0\"",SILENT);
AT_RECEIVE("OK",SILENT);
```

History:	17.02.98	LE	Initial
	23-Aug-01	SBK	Adapt to immediate OK after voice
call			

4.3.4 MSS203: AoC time related charging, MOC, k=4 (31.6.1.1)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second, stores the correct value in the ACM field of the SIM and that the MS ignores non-zero e5, e6 parameters.

In case k=4 the parameters are set to e1=1, e2=1, e3=1, e4=0, e5=10, e6=10, e7=1. It is expected that after 90 seconds CCM has the value 90 and ACM is increased by 90.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"00005A\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: \\"00005A\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", SILENT);
BS_RACH_AWAIT(0, channel_request_moc, "CHANNEL REQUEST");
/*
 * the following OK should be commented out if problems occur
 * as there is a race condition between the OK and the RACH
 * access; the infrastructure simulation does not support co-regions
 * like MSCs of Z.120 do, unfortunately
 */
AT_RECEIVE ("OK", "OK");

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT);
BS_MSG3_SEND (0, call_proceeding, SILENT);
```

```
BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_k4,SILENT);

BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (55000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

/* timer value should be corrected */
/* +CCWV is used as trigger which is not accurate enough */
ISS_DELAY (30000+1500)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE("+CAOC: \"00005A\"",SILENT);
AT_RECEIVE("OK",SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"00005A\"",SILENT);
AT_RECEIVE("OK",SILENT);

History:          17.02.98          LE          Initial
                  23-Aug-01        SBK          Adapt to immediate OK after voice
call
```

4.3.5 MSS204: AoC time related charging, MOC, k=5 (31.6.1.1)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second, stores the correct value in the ACM field of the SIM and that the MS ignores non-zero e5, e6 parameters.

In case k=5 the parameters are set to e1=12.5, e2=30, e3=1, e4=25, e5=10, e6=10, e7=30. It is expected that after 90 seconds CCM has the value 50 and ACM is increased by 50.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"000064\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: \\"000064\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", SILENT);
BS_RACH_AWAIT(0, channel_request_moc, "CHANNEL REQUEST");
/*
 * the following OK should be commented out if problems occur
 * as there is a race condition between the OK and the RACH
 * access; the infrastructure simulation does not support co-regions
 * like MSCs of Z.120 do, unfortunately
 */
AT_RECEIVE ("OK", "OK");

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT);
BS_MSG3_SEND (0, call_proceeding, SILENT);
```

```
BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_k5,SILENT);

BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (85000) /* plus 5 seconds for message exchange */

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE ("CAOC: \"000032\"",SILENT);
AT_RECEIVE ("OK",SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE ("CACM: \"000032\"",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History:	17.02.98	LE	Initial
	23-Aug-01	SBK	Adapt to immediate OK after voice
call			

4.3.6 MSS205: AoC time related charging, MTC, k=1 (31.6.1.2)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Facility message sent after the connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second and stores the correct value in the ACM field of the SIM. In case k=1 the parameters are set to e1=0, e2=0, e3=0, e4=0, e5=0, e6=0, e7=0. It is expected that after 90 seconds CCM has the value 0 and ACM is increased by 0.

Preamble: MSS001

Script:

```
BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0, paging_request_type_1, SILENT);
BS_RACH_AWAIT(0, channel_request, SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, paging_response, SILENT);

BS_MSG3_SEND (0, setup, SILENT);
BS_MSG3_AWAIT(0, call_confirmed, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_AWAIT(0, alerting, SILENT);

AT_SEND ("ATA\r\n", "Hook off");

BS_MSG3_AWAIT(0, connect, SILENT);
BS_MSG3_SEND (0, facility_msg_aocc_k1, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result_mtc, SILENT);
BS_MSG3_SEND (0, connect_acknowledge, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, disconnect_mtc, SILENT);
BS_MSG3_AWAIT(0, release_mtc, SILENT);
BS_MSG3_SEND (0, release_complete_mtc, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000000\"", SILENT);
AT_RECEIVE ("OK", SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000000\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History: 17.02.98 LE Initial

4.3.7 MSS206: AoC time related charging, MTC, k=2 (31.6.1.2)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Facility message sent after the connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second and stores the correct value in the ACM field of the SIM. In case k=2 the parameters are set to e1=0, e2=0, e3=1, e4=100, e5=0, e6=0, e7=0. It is expected that after 90 seconds CCM has the value 100 and ACM is increased by 100.

Preamble: MSS001

Script:

```
BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0, paging_request_type_1, SILENT);
BS_RACH_AWAIT (0, channel_request, SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT (0, paging_response, SILENT);

BS_MSG3_SEND (0, setup, SILENT);
BS_MSG3_AWAIT (0, call_confirmed, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT (0, assignment_complete, SILENT);

BS_MSG3_AWAIT (0, alerting, SILENT);

AT_SEND ("ATA\r\n", "Hook Off");

BS_MSG3_AWAIT (0, connect, SILENT);
BS_MSG3_SEND (0, facility_msg_aocc_k2, SILENT);
BS_MSG3_AWAIT (0, facility_aocc_return_result_mtc, SILENT);
BS_MSG3_SEND (0, connect_acknowledge, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, disconnect_mtc, SILENT);
BS_MSG3_AWAIT (0, release_mtc, SILENT);
BS_MSG3_SEND (0, release_complete_mtc, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE (" +CAOC: \"000064\"", SILENT);
AT_RECEIVE ("OK", SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE (" +CACM: \"000064\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History: 17.02.98 LE Initial

4.3.8 MSS207: AoC time related charging, MTC, k=3 (31.6.1.2)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Facility message sent after the connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second and stores the correct value in the ACM field of the SIM. In case k=3 the parameters are set to e1=6, e2=14, e3=1, e4=25, e5=0, e6=0, e7=60. It is expected that after 90 seconds CCM has the value 43 and ACM is increased by 43.

Preamble: MSS001

Script:

```
BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0, paging_request_type_1, SILENT);
BS_RACH_AWAIT(0, channel_request, SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, paging_response, SILENT);

BS_MSG3_SEND (0, setup, SILENT);
BS_MSG3_AWAIT(0, call_confirmed, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_AWAIT(0, alerting, SILENT);

AT_SEND ("ATA\r\n", "Hook Off");

BS_MSG3_AWAIT(0, connect, SILENT);
BS_MSG3_SEND (0, facility_msg_aocc_k3, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result_mtc, SILENT);
BS_MSG3_SEND (0, connect_acknowledge, SILENT);

ISS_DELAY (92000)

BS_MSG3_SEND (0, disconnect_mtc, SILENT);
BS_MSG3_AWAIT(0, release_mtc, SILENT);
BS_MSG3_SEND (0, release_complete_mtc, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"00002B\"", SILENT);
AT_RECEIVE ("OK", SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"00002B\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History: 17.02.98 LE Initial

4.3.9 MSS208: AoC time related charging, MTC, k=4 (31.6.1.2)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Facility message sent after the connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second and stores the correct value in the ACM field of the SIM. In case k=4 the parameters are set to e1=1, e2=1, e3=1, e4=0, e5=0, e6=0, e7=1. It is expected that after 90 seconds CCM has the value 90 and ACM is increased by 90.

Preamble: MSS001

Script:

```
BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0,paging_request_type_1,SILENT);
BS_RACH_AWAIT(0,channel_request,SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,paging_response,SILENT);

BS_MSG3_SEND (0,setup,SILENT);
BS_MSG3_AWAIT(0,call_confirmed,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_AWAIT(0,alerting,SILENT);

AT_SEND ("ATA\r\n","Hook Off");

BS_MSG3_AWAIT(0,connect,SILENT);
BS_MSG3_SEND (0,facility_msg_aocc_k4, SILENT)
BS_MSG3_AWAIT(0,facility_aocc_return_result_mtc,SILENT);
BS_MSG3_SEND (0,connect_acknowledge,SILENT);

ISS_DELAY (90400) /* timer is not accurate under WIN */

BS_MSG3_SEND (0,disconnect_mtc,SILENT);
BS_MSG3_AWAIT(0,release_mtc,SILENT);
BS_MSG3_SEND (0,release_complete_mtc, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE ("CAOC: \"00005A\"",SILENT);
AT_RECEIVE ("OK",SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE ("CACM: \"00005A\"",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.10 MSS209: AoC time related charging, MTC, k=5 (31.6.1.2)

Description: To verify that when the MS receives the AoCC parameters in a facility IE which is contained in the Facility message sent after the connect message and when a TCH has already been assigned, the MS returns a facility message containing the acknowledgement within one second and stores the correct value in the ACM field of the SIM. In case k=5 the parameters are set to e1=12.5, e2=30, e3=1, e4=25, e5=0, e6=0, e7=30. It is expected that after 90 seconds CCM has the value 50 and ACM is increased by 50.

Preamble: MSS001

Script:

```
BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0,paging_request_type_1,SILENT);
BS_RACH_AWAIT(0,channel_request,SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,paging_response,SILENT);

BS_MSG3_SEND (0,setup,SILENT);
BS_MSG3_AWAIT(0,call_confirmed,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_AWAIT(0,alerting,SILENT);

AT_SEND ("ATA\r\n","Hook Off");

BS_MSG3_AWAIT(0,connect,SILENT);
BS_MSG3_SEND (0,facility_msg_aocc_k5, SILENT)
BS_MSG3_AWAIT(0,facility_aocc_return_result_mtc,SILENT);
BS_MSG3_SEND (0,connect_acknowledge,SILENT);

ISS_DELAY (85000) /* plus 5 seconds for messages */

BS_MSG3_SEND (0,disconnect_mtc,SILENT);
BS_MSG3_AWAIT(0,release_mtc,SILENT);
BS_MSG3_SEND (0,release_complete_mtc, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE (" +CAOC: \"000032\" ",SILENT);
AT_RECEIVE ("OK",SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM: \"000032\" ",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.11 MSS210: Change in Charging Information during a call (31.6.1.5)

Description: To verify that when the MS receives new AoCC parameters mid-way through a call in a Facility IE which is contained within a Facility message the MS returns a facility message containing the acknowledgement within one second and that the MS correctly indicates the total charge considering both sets of charging information. The first set of parameters is set to e1=10, e2=28, e3=1, e4=10, e5=0, e6=0, e7=60. After 80 seconds the second set of parameters is send. The parameters are set to e1=10, e2=14, e3=1, e4=5, e5=0, e6=0, e7=60. It is expected that after 180 seconds CCM has the value 65 and ACM is increased by 65.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"00004A\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: \\"00004A\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);
```

```
BS_MSG3_SEND (0,facility_msg_aocc_set_1,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (80000)

BS_MSG3_SEND (0,facility_msg_aocc_set_2,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (75000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

ISS_DELAY (20000)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);

BS_MSG3_SEND (0,channel_release,SILENT);
```

```
ISS_DELAY (5000);

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000041\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000041\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History: 17.02.98 LE Initial

4.3.12 MSS211: Different formats of Charging Information, part 1 (31.6.1.6)

Description: To verify that when the MS receives a facility IE in which certain e-parameters are set to zero the total charge accumulated is the same as that when the same e-parameters are completely omitted from the facility IE. Different types of channel release are verified. In part 1 a normal disconnect, release, release complete and channel release sequence is processed.

The parameters are set for k=0 to e1=10, e2=40, e3=1, e4=10, e5=0, e6=0, e7=0. After 90 seconds the connection is released. The second and third connection has the parameters set to e1=10, e2=40 and e3=1 and the rest omitted. It is expected that CCM has the value 20 after each connection and ACM is increased by 20 for each connection.

Preamble: MSS001

Script:

```
AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);

BS_MSG3_SEND (0,facility_msg_aocc_diff_1,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (10000)
```

```
/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);

BS_MSG3_SEND (0, facility_msg_aocc_diff_2, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, disconnect, SILENT);
BS_MSG3_AWAIT(0, release, SILENT);
BS_MSG3_SEND (0, release_complete, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (10000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

```
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE (" +CACM: \"000028\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);

BS_MSG3_SEND (0, facility_msg_aocc_diff_2, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, disconnect, SILENT);
BS_MSG3_AWAIT(0, release, SILENT);
BS_MSG3_SEND (0, release_complete, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (10000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE (" +CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE (" +CACM: \"00003C\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History: 17.02.98 LE Initial

4.3.13 MSS212: Different formats of Charging Information, part 2 (31.6.1.6)

Description: To verify that when the MS receives a facility IE in which certain e-parameters are set to zero the total charge accumulated is the same as that when the same e-parameters are completely omitted from the facility IE. Different types of channel release are verified. In part 2 the connection is disconnected by release complete and channel release.

The parameters are set for k=0 to e1=10, e2=40, e3=1, e4=10, e5=0, e6=0, e7=0. After 90 seconds the connection is released. The second and third connection has the parameters set to e1=10, e2=40 and e3=1 and the rest omitted. It is expected that CCM has the value 20 after each connection and ACM is increased by 20 for each connection.

Preamble: MSS001

Script:

```
AT_SEND ("ATD03039094117;\r\n","Dial");

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);

BS_MSG3_SEND (0,facility_msg_aocc_diff_1,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (10000)
```



```
/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);

BS_MSG3_SEND (0, facility_msg_aocc_diff_2, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, release_complete, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (10000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000028\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");
```

```
BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);

BS_MSG3_SEND (0,facility_msg_aocc_diff_2,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (10000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE (" +CAOC:  \"000014\" ",SILENT);
AT_RECEIVE ("OK",SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM:  \"00003C\" ",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.14 MSS213: Different formats of Charging Information, part 3 (31.6.1.6)

Description: To verify that when the MS receives a facility IE in which certain e-parameters are set to zero the total charge accumulated is the same as that when the same e-parameters are completely omitted from the facility IE. Different types of channel release are verified. In part 3 the connection is disconnected by a channel release message. The parameters are set for k=0 to e1=10, e2=40, e3=1, e4=10, e5=0, e6=0, e7=0. After 90 seconds the connection is released. The second and third connection has the parameters set to e1=10, e2=40 and e3=1 and the rest omitted. It is expected that CCM has the value 20 after each connection and ACM is increased by 20 for each connection.

Preamble: MSS001

Script:

```
AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);

BS_MSG3_SEND (0, facility_msg_aocc_diff_1, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (10000)
```

```
/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);

BS_MSG3_SEND (0, facility_msg_aocc_diff_2, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (10000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \"000014\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \"000028\"", SILENT);
AT_RECEIVE ("OK", SILENT);
```

```
AT_SEND ("ATD03039094117;\r\n","Dial");

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);

BS_MSG3_SEND (0,facility_msg_aocc_diff_2,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (10000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE (" +CAOC:  \\"000014\\" ",SILENT);
AT_RECEIVE ("OK",SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM:  \\"00003C\\" ",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.15 MSS214: AoC on a Call Hold Call (31.6.1.7)

Description: To verify that when the MS invokes a call hold call and hence receives a facility IE containing AoCC e-parameters for each chargeable call the MS returns a Facility messages containing the AoCC acknowledgement within one seconds of transmission of each set of e-parameters. The CCM shall record the sum of the charges of each chargeable call and the MS shall store the correct value in the ACM field of the SIM.

Preamble: MSS001

Script:

```
/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"000037\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: \\"000037\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", SILENT);          /* Dial          */

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect_aocc_hold1, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

AT_SEND ("ATD03039094117;\r\n", SILENT);          /* Dial second call      */
```

```
BS_MSG3_AWAIT(0,cc_hold,SILENT);
BS_MSG3_SEND (0,cc_hold_ack, SILENT);

BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc_1, SILENT)
BS_MSG3_SEND (0,call_proceeding_1,SILENT);

BS_MSG3_SEND (0,alerting_1,SILENT);
BS_MSG3_SEND (0,connect_aocc_hold2,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge_1,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result_1,SILENT);

ISS_DELAY (85000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

BS_MSG3_SEND (0,disconnect_1,SILENT);
BS_MSG3_AWAIT(0,release_1,SILENT);
BS_MSG3_SEND (0,release_complete_1, SILENT);

ISS_DELAY (75000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

ISS_DELAY (10000)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000)
/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE("+CAOC: \"000036\"",SILENT);
AT_RECEIVE("OK",SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"000036\"",SILENT);
AT_RECEIVE("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.16 MSS215: AoC on a Multiparty Call (31.6.1.8)

Description: To verify that when the MS invokes a multiparty call and hence receives a facility IE containing AoCC e-parameters for each chargeable call the MS returns a Facility messages containing the AoCC acknowledgement within one seconds of transmission of each set of e-parameters. The CCM shall record the sum of the charges of each chargeable call and the MS shall store the correct value in the ACM field of the SIM.

Preamble: MSS001

Script:

```

/*
 * Enable call meter warning event
 */
AT_SEND ("AT+CCWE=1\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax value
 */
AT_SEND ("AT+CMM=\\"000087\\",\\"7890\\"r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Get ACMmax value
 */
AT_SEND ("AT+CMM?r\n", SILENT);
AT_RECEIVE ("CMM: \\"000087\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM, it should be set to zero
 */
AT_SEND ("AT+CAOCr\n", SILENT);
AT_RECEIVE ("CAOC: \\"000000\\\"", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;r\n", SILENT);          /* Dial          */

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect_aocc_mpty1, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

AT_SEND ("ATD03039094117;r\n", SILENT);          /* Dial second call      */

```



```

BS_MSG3_AWAIT(0,cc_hold,SILENT);
BS_MSG3_SEND (0,cc_hold_ack, SILENT);

BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc_1, SILENT)
BS_MSG3_SEND (0,call_proceeding_1,SILENT);

BS_MSG3_SEND (0,alerting_1,SILENT);
BS_MSG3_SEND (0,connect_aocc_mpty2,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge_1,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result_1,SILENT);

AT_SEND ("AT+CHLD=3\r\n",SILENT);          /* Start Multiparty      */
BS_MSG3_AWAIT(0,facility_mpty_invoke, SILENT)
BS_MSG3_SEND (0,facility_mpty_return_result,SILENT);

ISS_DELAY (85000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

BS_MSG3_SEND (0,disconnect_1,SILENT);
BS_MSG3_AWAIT(0,release_1,SILENT);
BS_MSG3_SEND (0,release_complete_1, SILENT);

ISS_DELAY (75000) /* 5000 msec waiting time for +CCWV */

AT_RECEIVE("+CCWV",SILENT);

ISS_DELAY (10000)

BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n",SILENT);
AT_RECEIVE("+CAOC: \"000086\"",SILENT);
AT_RECEIVE("OK",SILENT);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"000086\"",SILENT);
AT_RECEIVE("OK",SILENT);

```

History: 17.02.98 LE Initial

4.3.17 MSS216: Removal of SIM during an active Call (31.6.2.1)

Description: To verify that when the SIM is removed from the ME during an active AoCC call the ME immediately terminates the call and that the ME has written the total charge up to that point in the call to the ACM field of the SIM.

Preamble: MSS001

Script:

```
AT_SEND ("ATD03039094117;\r\n",SILENT);          /* Dial          */

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_storage1,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

COMMAND ("SIM CONFIG REMOVE");      /* Remove SIM. */

ISS_DELAY (10000)
AT_SEND ("AT+CFUN=0\r\n","Power Off");
ISS_DELAY (10000);
AT_SEND ("AT+CFUN=1\r\n","Power On");
AT_SEND ("AT+COPS=0\r\n","Automatic Registration");
ISS_DELAY (20000);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"00001E\"",SILENT);
AT_RECEIVE("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.18 MSS217: Interruption of power supply during an active call (31.6.2.2)

Description: To verify that when the ME is switched off during an active AoCC call the ME immediately terminates the call and that the ME has written the total charge up to that point in the call to the ACM field of the SIM.

Preamble: MSS001

Script:

```
AT_SEND ("ATD03039094117;\r\n",SILENT);          /* Dial          */

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_storage1,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)

AT_SEND ("AT+CFUN=0\r\n","Power Off");
ISS_DELAY (10000);
AT_SEND ("AT+CFUN=1\r\n","Power On");
AT_SEND ("AT+COPS=0\r\n","Automatic Registration");
ISS_DELAY (20000);

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE("+CACM: \"00001E\"",SILENT);
AT_RECEIVE("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.19 MSS218: MS going out of coverage during an active AoCC Call (31.6.2.3)

Description: To verify that when the ME has detected a radio link failure during an active AoCC call the ME immediately suspends charging and resumes charging after re-establishment and that the ME has written the total charge up to that point in the call to the ACM field of the SIM.

Preamble: MSS002

Script:

```
AT_SEND ("ATD03039094117;\r\n",SILENT);          /* Dial          */

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_storage1,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (55000) /* plus five for messages */

BS_SET_ERROR ( 0, 4);                               /* Radio Link Failure */

SET_TIMEOUT (20000);
BS_RACH_AWAIT(0,channel_request_reest_neci_0,SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_reestablish_request,SILENT);

ISS_DELAY (10000)
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM: \"000014\"",SILENT);
AT_RECEIVE ("OK",SILENT);

BS_MSG3_SEND (0,ciphering_mode_command,SILENT);
BS_MSG3_AWAIT(0,ciphering_mode_complete,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);
```

ISS_DELAY (10000)

```
/*  
 * Check ACM  
 */  
AT_SEND ("AT+CACM?\r\n",SILENT);  
AT_RECEIVE("+CACM: \"00001E\"",SILENT);  
AT_RECEIVE("OK",SILENT);  
  
BS_MSG3_SEND (0,disconnect,SILENT);  
BS_MSG3_AWAIT(0,release,SILENT);  
BS_MSG3_SEND (0,release_complete, SILENT);  
BS_MSG3_SEND (0,channel_release,SILENT);
```

History: 17.02.98 LE Initial

4.3.20 MSS219: ACMmax operation / Mobile originating (31.6.2.4)

Description: The following steps are performed:
Set ACMmax to 2 using PIN2.
Start ordinary mobile originated call.
Disconnect after two units the call due to ACM exceeds ACMmax.
Start ordinary mobile originated call which is rejected by MMI.
Start emergency call with zero-CAI.
Check ACM (which shall be set to 2).

Preamble: MSS001

Script:

```
AT_SEND ("AT+CMM="000002",\r\n",SILENT); /* set ACMmax with PIN2*/
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n",SILENT);          /* Dial          */

BS_RACH_AWAIT(0,channel_request_moc,SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_acmmmax1,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

SET_TIMEOUT (20000)
ISS_DELAY (80000)
BS_MSG3_AWAIT(0,disconnect_acm,SILENT);
BS_MSG3_SEND (0,release,SILENT);
BS_MSG3_AWAIT(0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (10000)

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM: \"000002\"",SILENT);
AT_RECEIVE ("OK", SILENT);
```

```
AT_SEND ("ATD03039094117;\r\n",SILENT);          /* Dial          */
AT_RECEIVE ("ERROR", SILENT);

ISS_DELAY (10000)
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM: \"000002\"",SILENT);
AT_RECEIVE ("OK",SILENT);

AT_SEND ("ATD112;\r\n",SILENT);          /* Dial Emergency Call */

BS_RACH_AWAIT_BEGIN( 0, channel_request_moc, SILENT)
    BF_SET_VAL      ( establishment_cause, M3(1,0,1 ), "Emergency call" )
BS_RACH_AWAIT_END( )
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request_ec,SILENT);
BS_MSG3_SEND (0,cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, emergency_setup, SILENT)
BS_MSG3_SEND (0,call_proceeding,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_SEND (0,alerting,SILENT);
BS_MSG3_SEND (0,connect_aocc_zero_cai,SILENT);
BS_MSG3_AWAIT(0,connect_acknowledge,SILENT);
BS_MSG3_AWAIT(0,facility_aocc_return_result,SILENT);

ISS_DELAY (90000)
BS_MSG3_SEND (0,disconnect,SILENT);
BS_MSG3_AWAIT(0,release,SILENT);
BS_MSG3_SEND (0,release_complete, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);
ISS_DELAY (10000)

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM: \"000002\"",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.21 MSS220: ACMmax operation / Mobile Terminating (31.6.2.5)

Description: The following steps are performed:
Set ACMmax to 2 using PIN2.
Start mobile terminating call.
Disconnect after two units the call due to ACM exceeds ACMmax.
Start mobile terminating call with non-zero CAI
Disconnect call after reception of non-zero CAI.
Start mobile terminating call with zero CAI
Disconnect call after 120 seconds
Check ACM (which shall be set to 2).

Preamble: MSS001

Script:

```
AT_SEND ("AT+CMM=\\"000002\\",\\"7890\\"r\n",SILENT); /* set ACMmax with PIN2 */
AT_RECEIVE ("OK", SILENT);

BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0,paging_request_type_1,SILENT);
BS_RACH_AWAIT(0,channel_request,SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,paging_response,SILENT);

BS_MSG3_SEND (0,setup,SILENT);
BS_MSG3_AWAIT(0,call_confirmed,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_AWAIT(0,alerting,SILENT);

AT_SEND ("ATA\r\n","Hook Off");

BS_MSG3_AWAIT(0,connect,SILENT);
BS_MSG3_SEND (0,connect_acknowledge,SILENT);

BS_MSG3_SEND (0,facility_msg_aocc_acmmx1, SILENT)
BS_MSG3_AWAIT(0,facility_aocc_return_result_mtc,SILENT);

ISS_DELAY (90000)

BS_MSG3_AWAIT(0,disconnect_acm_mtc,SILENT);
BS_MSG3_SEND (0,release_mtc,SILENT);
BS_MSG3_AWAIT(0,release_complete_mtc, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);
```



```
ISS_DELAY (5000)

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE (" +CACM: \"000002\"", SILENT);
AT_RECEIVE ("OK", SILENT);

ISS_DELAY (10000);

BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0, paging_request_type_1, SILENT);
BS_RACH_AWAIT (0, channel_request, SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT (0, paging_response, SILENT);

BS_MSG3_SEND (0, setup, SILENT);
BS_MSG3_AWAIT (0, call_confirmed, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT (0, assignment_complete, SILENT);

BS_MSG3_AWAIT (0, alerting, SILENT);

AT_SEND ("ATA\r\n", "Hook Off");

BS_MSG3_AWAIT (0, connect, SILENT);
BS_MSG3_SEND (0, connect_acknowledge, SILENT);

BS_MSG3_SEND (0, facility_msg_aocc_acmmmax1, SILENT);
BS_MSG3_AWAIT (0, disconnect_acm_mtc, SILENT);
BS_MSG3_AWAIT (0, facility_aocc_return_result_mtc, SILENT);

BS_MSG3_SEND (0, release_mtc, SILENT);
BS_MSG3_AWAIT (0, release_complete_mtc, SILENT);
BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (5000)

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE (" +CACM: \"000002\"", SILENT);
AT_RECEIVE ("OK", SILENT);

ISS_DELAY (10000)

BS_CONFIG_CHANNEL (0, PCH, UNACK, SAPI_0);
BS_MSG3_SEND (0, paging_request_type_1, SILENT);
BS_RACH_AWAIT (0, channel_request, SILENT);

BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
```

```
BS_MSG3_SEND (0,immediate_assignment,SILENT) ;
```

```
BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0,paging_response,SILENT);

BS_MSG3_SEND (0,setup,SILENT);
BS_MSG3_AWAIT(0,call_confirmed,SILENT);

BS_MSG3_SEND (0,assignment_command,SILENT);
BS_MSG3_AWAIT(0,assignment_complete,SILENT);

BS_MSG3_AWAIT(0,alerting,SILENT);

AT_SEND ("ATA\r\n","Hook Off");

BS_MSG3_AWAIT(0,connect,SILENT);
BS_MSG3_SEND (0,connect_acknowledge,SILENT);

BS_MSG3_SEND (0,facility_msg_aocc_zero_cai, SILENT)
BS_MSG3_AWAIT(0,facility_aocc_return_result_mtc,SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0,disconnect_mtc,SILENT);
BS_MSG3_AWAIT(0,release_mtc,SILENT);
BS_MSG3_SEND (0,release_complete_mtc, SILENT);
BS_MSG3_SEND (0,channel_release,SILENT);

ISS_DELAY (5000)

/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n",SILENT);
AT_RECEIVE (" +CACM: \"000002\"",SILENT);
AT_RECEIVE ("OK",SILENT);
```

History: 17.02.98 LE Initial

4.3.22 MSS221: Test of several AT-Commands for AoCC

Description: Test of AT-Commands used for Advice of Charge.

Preamble: MSS001

Script:

```
/*
 * Check Call Timer %CTV
 */
AT_SEND ("AT%CTV\r\n", SILENT);
AT_RECEIVE ("%CTV: 0", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax
 */
AT_SEND ("AT+CMM="000002", "7890"\r\n", SILENT); /* set ACMmax with PIN2 */
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACMmax
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: 000002", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Set ACMmax
 */
AT_SEND ("AT+CMM="000000", "7890"\r\n", SILENT); /* set ACMmax with PIN2 */
AT_RECEIVE ("OK", SILENT);

/*
 * Check ACMmax
 */
AT_SEND ("AT+CMM?\r\n", SILENT);
AT_RECEIVE ("CMM: 000000", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check +CMM=? (no output expected)
 */
AT_SEND ("AT+CMM=?\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check +CACM=? (no output expected)
 */
AT_SEND ("AT+CACM=?\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);
```

```
/*
 * Check +CPUC?
 */
AT_SEND ("AT+CPUC?\r\n", SILENT);
AT_RECEIVE ("CPUC: \r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check +CPUC=? (no output expected)
 */
AT_SEND ("AT+CPUC=?\r\n", SILENT);
AT_RECEIVE ("OK", SILENT);

AT_SEND ("ATD03039094117;\r\n", "Dial");

BS_RACH_AWAIT(0, channel_request_moc, SILENT);
BS_CONFIG_CHANNEL (0, AGCH, UNACK, SAPI_0);
BS_STORE_RACH_PARAMS (0, 0);
BS_MSG3_SEND (0, immediate_assignment, SILENT);

BS_CONFIG_CHANNEL (0, SDCCH, 1, SAPI_0);
BS_MSG3_AWAIT(0, cm_service_request, SILENT);
BS_MSG3_SEND (0, cm_service_accept, SILENT);

BS_MSG3_AWAIT(0, setup_moc, SILENT)
BS_MSG3_SEND (0, call_proceeding, SILENT);

BS_MSG3_SEND (0, assignment_command, SILENT);
BS_MSG3_AWAIT(0, assignment_complete, SILENT);

BS_MSG3_SEND (0, alerting, SILENT);
BS_MSG3_SEND (0, connect_aocc_k1, SILENT);
BS_MSG3_AWAIT(0, connect_acknowledge, SILENT);
BS_MSG3_AWAIT(0, facility_aocc_return_result, SILENT);

ISS_DELAY (90000)

BS_MSG3_SEND (0, disconnect, SILENT);
BS_MSG3_AWAIT(0, release, SILENT);
BS_MSG3_SEND (0, release_complete, SILENT);

BS_MSG3_SEND (0, channel_release, SILENT);

ISS_DELAY (5000)

/*
 * Check CCM
 */
AT_SEND ("AT+CAOC\r\n", SILENT);
AT_RECEIVE ("CAOC: \r\n", SILENT);
AT_RECEIVE ("OK", SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\r\n", SILENT);
AT_RECEIVE ("CACM: \r\n", SILENT);
AT_RECEIVE ("OK", SILENT);
```

```
/*
 * Set +CPUC
 */
AT_SEND ("AT+CPUC=\"FRA\", \"6.02\", \"7890\"\\r\\n\", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Check CCM with Puct
 */
AT_SEND ("AT%CAOC\\r\\n\", SILENT);
AT_RECEIVE ("%CAOC: \"FRA\", \"258.86\"\", SILENT);
AT_RECEIVE ("OK", SILENT);

/*
 * Reset ACM
 */
AT_SEND ("AT+CACM=\"7890\"\\r\\n\", SILENT); /* reset ACM with PIN2 */
AT_RECEIVE ("OK", SILENT);
/*
 * Check ACM
 */
AT_SEND ("AT+CACM?\\r\\n\", SILENT);
AT_RECEIVE (" +CACM: \"000000\"\", SILENT);
AT_RECEIVE ("OK", SILENT);
```

History: 17.02.98 LE Initial

4.3.23 MSS300: Call Barring, Deactivation accepted

Description: A deactivation of a call barring BAOC service is started. It is accepted by the network.

Preamble: MSS001

Script:

```
AT_SEND("ATD#33**10#\n\r",SILENT);

BS_RACH_AWAIT(0,channel_request_ss,SILENT);

BS_CONFIG_CHANNEL(0,AGCH,UNACK,SAPI_0);
BS_STORE_RACH_PARAMS(0,0);
BS_MSG3_SEND(0,immediate_assignment,SILENT);

BS_CONFIG_CHANNEL(0,SDCCH,1,SAPI_0);
BS_MSG3_AWAIT(0,cm_service_request_ss,SILENT);
BS_MSG3_SEND(0,cm_service_accept,SILENT);

BS_MSG3_AWAIT(0,ss_register_deactBAOC,SILENT);
BS_MSG3_SEND(0,ss_release_complete_deactBAOC,SILENT);

AT_RECEIVE("OK",SILENT);

BS_MSG3_SEND(0,channel_release,SILENT);

ISS_DELAY(5000);
```

History: 30.09.99 AK Initial

Appendices

A. Acronyms

DS-WCDMA Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000) Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: <http://www.imt-2000.org/>>