

ETSI TS 129 168 V8.1.0 (2009-04)

Technical Specification

**Universal Mobile Telecommunications System (UMTS);
LTE;
Cell Broadcast Centre interfaces with the Evolved Packet Core;
Stage 3
(3GPP TS 29.168 version 8.1.0 Release 8)**



Reference

RTS/TSGC-0429168v810

Keywords

LTE, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2009.
All rights reserved.

DECTTM, **PLUGTESTS**TM, **UMTS**TM, **TIPHON**TM, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

LTETM is a Trade Mark of ETSI currently being registered

for the benefit of its Members and of the 3GPP Organizational Partners.

GSM[®] and the GSM logo are Trade Marks registered and owned by the GSM Association.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

Contents

Intellectual Property Rights	2
Foreword.....	2
Foreword.....	5
1 Scope	6
2 References	6
3 Definitions, symbols and abbreviations	6
3.1 Definitions	6
3.2 Symbols.....	7
3.3 Abbreviations	7
4 SBc description	7
4.1 Transport	7
4.1.1 General.....	7
4.1.2 Network layer	8
4.1.3 Transport layer.....	8
4.1.4 Services expected from signalling transport	8
4.2 SBc-AP functions	8
4.2.1 Function of SBc-AP.....	8
4.3 SBc-AP procedure	8
4.3.1 General.....	8
4.3.2 List of SBc-AP elementary procedure.....	8
4.3.3 Warning Message Transmission procedure	9
4.3.3.1 General	9
4.3.3.2 Successful Operation	9
4.3.3.3 Unsuccessful Operation.....	9
4.3.4 Message functional definition and content.....	10
4.3.4.1 Message contents	10
4.3.4.1.1 Presence.....	10
4.3.4.1.2 Criticality.....	10
4.3.4.1.3 Range.....	10
4.3.4.1.4 Assigned Criticality	10
4.3.4.2 Warning Message Transmission Messages	10
4.3.4.2.1 WRITE-REPLACE WARNING REQUEST	10
4.3.4.2.2 WRITE-REPLACE WARNING RESPONSE	11
4.3.4.3 Information element definition.....	11
4.3.4.3.1 Message Type.....	11
4.3.4.3.2 Cause	12
4.3.4.3.3 Criticality Diagnostics	12
4.4 Message and information element abstract syntax	13
4.4.1 General.....	13
4.4.2 Usage of protocol extension mechanism for non-standard use	14
4.4.3 Elementary procedure definitions	14
4.4.4 PDU definitions	16
4.4.5 Information element definitions.....	17
4.4.6 Common definitions	20
4.4.7 Constant definitions	21
4.4.8 Container Definitions.....	22
4.4.9 Message transfer syntax	24
4.5 Handling of unknown, unforeseen or erroneous protocol data.....	24
4.5.1 General	24
4.5.2 Transfer Syntax Error.....	24
4.5.3 Abstract Syntax Error.....	24
4.5.3.1 General.....	24
4.5.3.2 Criticality information	25

4.5.3.3	Presence information	25
4.5.3.4	Not comprehended IE/IE group	26
4.5.3.4.1	Procedure code	26
4.5.3.4.2	Type of Message	26
4.5.3.4.3	IEs other than the Procedure Code and Type of Message	26
4.5.3.5	Missing IE or IE group	27
4.5.3.6	IEs or IE groups received in wrong order or with too many occurrences or erroneously present	28
4.5.4	Logical Error	29
4.5.5	Exceptions	29
Annex A (informative): Change history:.....		30
History		31

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
 - 1 presented to TSG for information;
 - 2 presented to TSG for approval;
 - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document specifies the procedures and the SBc Application Part (SBc-AP) messages used on the SBc-AP interface between the Mobility Management Entity (MME) and the Cell Broadcast Centre (CBC).

The present document supports the following functions.

- Warning Message Transmission function in the EPS.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] IETF RFC 2460 (December 1998): "Internet Protocol, Version 6 (IPv6) Specification".
- [3] IETF RFC 791 (September 1981): "Internet Protocol".
- [4] IETF RFC 4960 (September 2007): "Stream Control Transmission Protocol".
- [5] IETF RFC 3309: "Stream Control Transmission Protocol (SCTP) Checksum Change".
- [6] IETF RFC 2960: "Stream Control Transmission Protocol".
- [7] 3GPP TS 36.413: "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)".
- [8] ITU-T Recommendation X.680 (07/2002): "Information Technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".
- [9] ITU-T Recommendation X.681 (07/2002): "Information Technology - Abstract Syntax Notation One (ASN.1): Information object specification".
- [10] ITU-T Recommendation X.691 (07/2002): "Information Technology - ASN.1 encoding rules - Specification of Packed Encoding Rules (PER)".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

Elementary Procedure: SBc-AP consists of Elementary Procedures (EPs). An Elementary Procedure is a unit of interaction between MME and CBC. These Elementary Procedures are defined separately and are intended to be used to build up complete sequences in a flexible manner. If the independence between some EPs is restricted, it is described

under the relevant EP description. Unless otherwise stated by the restrictions, the EPs may be invoked independently of each other as stand alone procedures, which can be active in parallel. Examples on using several SBc-APs together with each other and EPs from other interfaces can be found in reference [FFS].

An EP consists of an initiating message and possibly a response message. Two kinds of EPs are used:

- **Class 1:** Elementary Procedures with response (success and/or failure).
- **Class 2:** Elementary Procedures without response.

For Class 1 EPs, the types of responses can be as follows:

Successful:

- A signalling message explicitly indicates that the elementary procedure successfully completed with the receipt of the response.

Unsuccessful:

- A signalling message explicitly indicates that the EP failed.
- On time supervision expiry (i.e. absence of expected response).

Successful and Unsuccessful:

- One signalling message reports both successful and unsuccessful outcome for the different included requests. The response message used is the one defined for successful outcome.

Class 2 EPs are considered always successful.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

<none>

Editor's note: To be completed or section removed.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

EPC	Evolved Packet Core
EPS	Evolved Packet System
ETWS	Earthquake and Tsunami Warning System

4 SBc description

4.1 Transport

4.1.1 General

This subclause specifies the standards for signalling transport to be used across SBc-AP interface. SBc-AP interface is a logical interface between the MME and the CBC. All the SBc-AP messages described in the present document require an SCTP association between the MME and the CBC.

4.1.2 Network layer

The MME and the CBC shall support IPv6 (see IETF RFC 2460 [2]) and/or IPv4 (see IETF RFC 791 [3]).

The IP layer of SBc-AP only supports point-to-point transmission for delivering SBc-AP messages.

4.1.3 Transport layer

SCTP (see IETF RFC 4960 [4]) shall be supported as the transport layer of SBc-AP messages.

Semi-permanent SCTP associations shall be established between MME and CBC, i.e. the SCTP associations shall remain up under normal circumstances.

Local multi-homing should be supported. Remote multi-homing shall be supported.

Multiple local SCTP endpoints may be supported. Multiple remote SCTP endpoints shall be supported. When multiple local or remote SCTP endpoints are configured, several simultaneous SCTP associations shall be supported between MME and CBC.

Checksum calculation for SCTP shall be supported as specified in RFC 3309 [5] instead of the method specified in RFC 2960 [6].

The CBC shall establish the SCTP association.

The registered port number for SBc-AP is XXXX.

Editor's note: The port number for SBc-AP will be assigned by IANA once this specification is issued.

The registered payload protocol identifier for SBc-AP is XX.

Editor's note: The payload protocol identifier for SBc-AP will be assigned by IANA once this specification is issued.

4.1.4 Services expected from signalling transport

The signalling connection shall provide in-sequence delivery of SBc-AP messages. SBc-AP shall be notified if the signalling connection breaks.

4.2. SBc-AP functions

4.2.1 Function of SBc-AP

SBc-AP has the following function:

- Warning Message Transmission function:
This functionality provides the means to start and overwrite the broadcasting of warning message.

4.3 SBc-AP procedure

4.3.1 General

This sub-clause describes the parameters and detailed behaviors of different procedures.

4.3.2 List of SBc-AP elementary procedure

SBc-AP has the following EP defined as a class 1 procedure:

Table 4.3.2-1: Meaning of abbreviations used in SBc-AP messages

Elementary Procedure	Initiating Message	Successful Outcome	Unsuccessful Outcome
		Response message	Response message
Write-Replace Warning procedure	WRITE-REPLACE WARNING REQUEST	WRITE-REPLACE WARNING RESPONSE	

4.3.3 Warning Message Transmission procedure

4.3.3.1 General

The purpose of Write-Replace Warning procedure is to start, overwrite the broadcasting of warning message.

4.3.3.2 Successful Operation

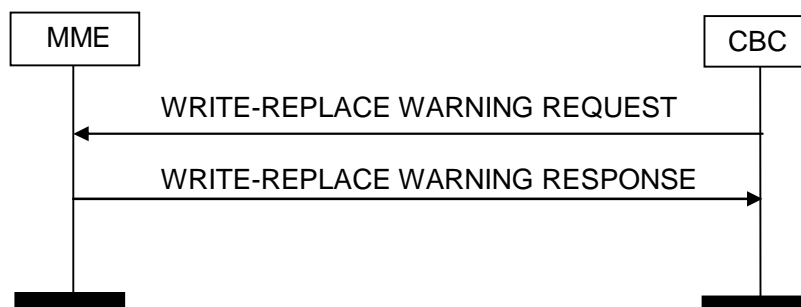


Figure 4.3.3.2-1: Write-Replace Warning procedure. Successful operation.

The CBC initiates the procedure by sending a WRITE-REPLACE WARNING REQUEST message to the MME.

Upon reception of WRITE-REPLACE WARNING REQUEST message, the MME shall forward the message towards the eNBs belonged to the tracking area as indicated in *List of TAIs* IE.

If none of *List of TAIs* IE or *Warning Area List* IE is present in WRITE-REPLACE WARNING REQUEST message, the MME shall forward the message towards all connected eNBs.

The MME returns a WRITE-REPLACE WARNING RESPONSE to the CBC immediately after the reception of the WRITE-REPLACE WARNING REQUEST message without waiting responses from eNBs.

MME sets the cause IE to "Message accepted" in the WRITE-REPLACE WARNING RESPONSE message.

4.3.3.3 Unsuccessful Operation

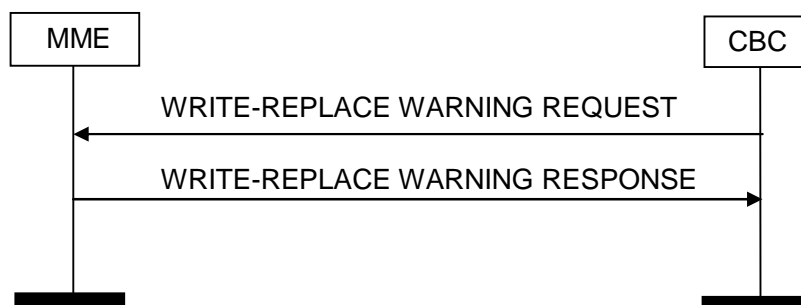


Figure 4.3.3.3-1: Write-Replace Warning procedure. Unsuccessful operation.

The CBC initiates the procedure by sending a WRITE-REPLACE WARNING REQUEST message to the MME.

If MME cannot process the received WRITE-REPLACE WARNING REQUEST message, the MME shall return a WRITE-REPLACE WARNING RESPONSE message towards the CBC and the MME shall not forward the message towards the eNBs belonged to the tracking area as indicated in *List of TAIs* IE.

MME indicates a reason of failure in the cause IE.

4.3.4 Message functional definition and content

4.3.4.1 Message contents

4.3.4.1.1 Presence

All information elements in the message descriptions below are marked mandatory, optional or conditional according to table 4.3.4.1.1-1.

Table 4.3.4.1.1-1: Meaning of abbreviations used in SBc-AP messages

Abbreviation	Meaning
M	IEs marked as Mandatory (M) shall always be included in the message.
O	IEs marked as Optional (O) may or may not be included in the message.
C	IEs marked as Conditional (C) shall be included in a message only if the condition is satisfied. Otherwise the IE shall not be included.

4.3.4.1.2 Criticality

Each Information Element or Group of Information Elements may have criticality information applied to it. Following cases are possible:

Table 4.3.4.1.2-1: Meaning of content within "Criticality" column

Abbreviation	Meaning
–	No criticality information is applied explicitly.
YES	Criticality information is applied. This is usable only for non-repeatable IEs
GLOBAL	The IE and all its repetitions together have one common criticality information. This is usable only for repeatable IEs.
EACH	Each repetition of the IE has its own criticality information. It is not allowed to assign different criticality values to the repetitions. This is usable only for repeatable IEs.

4.3.4.1.3 Range

The Range column indicates the allowed number of copies of repetitive IEs/IE groups.

4.3.4.1.4 Assigned Criticality

This column provides the actual criticality information as defined in subclause 4.5.3.2, if applicable.

4.3.4.2 Warning Message Transmission Messages

4.3.4.2.1 WRITE-REPLACE WARNING REQUEST

This message is sent by the CBC to request start and overwrite of an ETWS message broadcast.

Direction: CBC → MME

Table 4.3.4.2.1-1: WRITE-REPLACE WARNING REQUEST message contents

IE/Group Name	Presence	Range	IE type and reference	Semantics description	Criticality	Assigned Criticality
Message Type	M		4.3.4.3.1		YES	ignore
Message Identifier	M		[7]		YES	reject
Serial Number	M		[7]		YES	reject
List of TAIs	O				YES	reject
>TAl List Item		1 to <maxnoofTAl>				
>>TAl	M					
Warning Area List	O		[7]		YES	ignore
Repetition Period	M		[7]		YES	reject
Number of Broadcast Requested	M		[7]		YES	reject
Warning Type	O		[7]		YES	ignore
Warning Security Information	O		[7]		YES	ignore
Data Coding Scheme	O		[7]		YES	ignore
Warning Message Contents	O		[7]		YES	ignore

Table 4.3.4.2.1-2: RANGE explanation

Range bound	Explanation
maxnoofTAl	Maximum no. of TAl subject for warning message broadcast. Value is 65535.

4.3.4.2.2 WRITE-REPLACE WARNING RESPONSE

This message is sent by the MME to acknowledge the CBC on the start or overwrite request of an ETWS message.

Direction: CBC → MME

Table 4.3.4.2.2-1: WRITE-REPLACE WARNING RESPONSE message contents

IE/Group Name	Presence	Range	IE type and reference	Semantics description	Criticality	Assigned Criticality
Message Type	M		4.3.4.3.1		YES	ignore
Message Identifier	M		[7]		YES	reject
Serial Number	M		[7]		YES	reject
Cause	M		4.3.4.3.2		YES	ignore
Criticality Diagnostics	O		4.3.4.3.3		YES	ignore

4.3.4.3 Information element definition

4.3.4.3.1 Message Type

The *Message Type* IE uniquely identifies the message being sent. It is mandatory for all messages

Table 4.3.4.3.1-1: Message Type information element

IE/Group Name	Presence	Range	IE type and reference	Semantics description
Message Type				Assumed max no of messages is 256.
>Procedure Code	M		(Write-Replace Warning, ...)	
>Type of Message	M		CHOICE (Initiating Message, Successful Outcome, Unsuccessful Outcome, ...)	

4.3.4.3.2 Cause

The purpose of the *Cause* IE is to indicate the reason for a particular event for the SBC-AP protocol.

Table 4.3.4.3.2-1: Cause information element

IE/Group Name	Presence	Range	IE Type and Reference	Semantics Description
Cause	M		INTEGER (Message accepted, Parameter not recognised, Parameter value invalid, Valid message not identified, Tracking area not valid, Unrecognised message, Missing mandatory element, MME capacity exceeded, MME memory exceeded, Warning broadcast not supported, Warning broadcast not operational, Message reference already used, Unspecified error, Transfer syntax error, Semantic error, Message not compatible with receiver state, Abstract syntax error reject, Abstract syntax error ignore and notify, Abstract syntax error falsely constructed message, ...)	

4.3.4.3.3 Criticality Diagnostics

The *Criticality Diagnostics* IE is sent by the MME when parts of a received message have not been comprehended or were missing, or if the message contained logical errors. When applicable, it contains information about which IEs were not comprehended or were missing.

Table 4.3.4.3.3-1: Criticality Diagnostics information element

IE/Group Name	Presence	Range	IE type and reference	Semantics description
Procedure Code	O		INTEGER (0..255)	Procedure Code is to be used if Criticality Diagnostics is part of Error Indication procedure, and not within the response message of the same procedure that caused the error
Triggering Message	O		ENUMERATED(initiating message, successful outcome, unsuccessful outcome, outcome)	The Triggering Message is used only if the Criticality Diagnostics is part of Error Indication procedure.
Procedure Criticality	O		ENUMERATED(reject, ignore, notify)	This Procedure Criticality is used for reporting the Criticality of the Triggering message (Procedure).
Information Element Criticality Diagnostics		0 to <maxnoof errors>		
>IE Criticality	M		ENUMERATED(reject, ignore, notify)	The IE Criticality is used for reporting the criticality of the triggering IE. The value 'ignore' shall not be used.
>IE ID	M		INTEGER (0..65535)	The IE ID of the not understood or missing IE
>Type of Error	M		ENUMERATED(not understood, missing, ...)	

Table 4.3.4.3.3-2: RANGE explanation

Range bound	Explanation
maxnooferrors	Maximum no. of IE errors allowed to be reported with a single message. The value for maxnooferrors is 256.

4.4 Message and information element abstract syntax

4.4.1 General

SBC-AP ASN.1 definition conforms with [8] and [9].

The ASN.1 definition specifies the structure and content of SBC-AP messages. SBC-AP messages can contain any IEs specified in the object set definitions for that message without the order or number of occurrence being restricted by ASN.1. However, for this version of the standard, a sending entity shall construct a SBC-AP message according to the PDU definitions module and with the following additional rules (Note that in the following IE means an IE in the object set with an explicit id. If one IE needed to appear more than once in one object set, then the different occurrences have different IE ids):

- IEs shall be ordered (in an IE container) in the order they appear in object set definitions.
- Object set definitions specify how many times IEs may appear. An IE shall appear exactly once if the presence field in an object has value "mandatory". An IE may appear at most once if the presence field in an object has value "optional" or "conditional". If in a tabular format there is multiplicity specified for an IE (i.e. an IE list)

then in the corresponding ASN.1 definition the list definition is separated into two parts. The first part defines an IE container list where the list elements reside. The second part defines list elements. The IE container list appears as an IE of its own. For this version of the standard an IE container list may contain only one kind of list elements.

If a SBC-AP message that is not constructed as defined above is received, this shall be considered as Abstract Syntax Error, and the message shall be handled as defined for Abstract Syntax error in subclause 4.5.3.6.

4.4.2 Usage of protocol extension mechanism for non-standard use

The protocol extension mechanism for non-standard use may be used:

- for special operator- (and/or vendor) specific features considered not to be part of the basic functionality, i.e. the functionality required for a complete and high-quality specification in order to guarantee multivendor interoperability.
- by vendors for research purposes, e.g. to implement and evaluate new algorithms/features before such features are proposed for standardisation.

The extension mechanism shall not be used for basic functionality. Such functionality shall be standardised.

4.4.3 Elementary procedure definitions

Editor's note: The ASN.1 Object Identifiers (OID) for SBC-AP will be assigned by ITU-T via ETSI once this specification is issued. The following OID is expected to be assigned.
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0) eps-Access (21) modules (3) SBC-AP (X) version1 (1) SBC-AP-PDU-Contents (1)

Spec Manager's note:

Value **3** assigned to sbc-AP (X).

"SBC-AP-PDU-Contents (**1**)" in note above is rendered as "sbc-AP-PDU-Contents (**0**)" below. Value **1** is assumed to be correct in accordance with §4.4.4 where it is rendered as " sbc-AP-PDU-Contents (**1**)".

```
-- *****
--
-- Elementary Procedure definitions
--
-- *****

SBC-AP-PDU-Descriptions {
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0)
eps-Access (21) modules (3) sbc-AP (3) version1 (1) sbc-AP-PDU-Descriptions (0)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- *****
--
-- IE parameter types from other modules.
--
-- *****

IMPORTS
    Criticality,
    ProcedureCode
FROM SBC-AP-CommonDataTypes

    Write-Replace-Warning-Request,
    Write-Replace-Warning-Response
FROM SBC-AP-PDU-Contents

    id-Write-Replace-Warning
FROM SBC-AP-Constants;

-- *****
--
-- Interface Elementary Procedure Class
--
```

```

-- *****
SBC-AP-ELEMENTARY-PROCEDURE ::= CLASS {
    &InitiatingMessage          ,
    &SuccessfulOutcome          OPTIONAL,
    &UnsuccessfulOutcome        OPTIONAL,
    &procedureCode              ProcedureCode UNIQUE,
    &criticality                Criticality   DEFAULT ignore
}
WITH SYNTAX {
    INITIATING MESSAGE          &InitiatingMessage
    [SUCCESSFUL OUTCOME        &SuccessfulOutcome]
    [UNSUCCESSFUL OUTCOME      &UnsuccessfulOutcome]
    PROCEDURE CODE              &procedureCode
    [CRITICALITY                &criticality]
}

-- *****
--
-- Interface PDU Definition
--
-- *****

SBC-AP-PDU ::= CHOICE {
    initiatingMessage  InitiatingMessage,
    successfulOutcome  SuccessfulOutcome,
    unsuccessfulOutcome UnsuccessfulOutcome,
    ...
}

InitiatingMessage ::= SEQUENCE {
    procedureCode  SBC-AP-ELEMENTARY-PROCEDURE.&procedureCode  ({SBC-AP-ELEMENTARY-PROCEDURES}),
    criticality    SBC-AP-ELEMENTARY-PROCEDURE.&criticality      ({SBC-AP-ELEMENTARY-
PROCEDURES}@procedureCode)},
    value         SBC-AP-ELEMENTARY-PROCEDURE.&InitiatingMessage ({SBC-AP-ELEMENTARY-
PROCEDURES}@procedureCode)}
}

SuccessfulOutcome ::= SEQUENCE {
    procedureCode  SBC-AP-ELEMENTARY-PROCEDURE.&procedureCode  ({SBC-AP-ELEMENTARY-PROCEDURES}),
    criticality    SBC-AP-ELEMENTARY-PROCEDURE.&criticality      ({SBC-AP-ELEMENTARY-
PROCEDURES}@procedureCode)},
    value         SBC-AP-ELEMENTARY-PROCEDURE.&SuccessfulOutcome ({SBC-AP-ELEMENTARY-
PROCEDURES}@procedureCode)}
}

UnsuccessfulOutcome ::= SEQUENCE {
    procedureCode  SBC-AP-ELEMENTARY-PROCEDURE.&procedureCode  ({SBC-AP-ELEMENTARY-PROCEDURES}),
    criticality    SBC-AP-ELEMENTARY-PROCEDURE.&criticality      ({SBC-AP-ELEMENTARY-
PROCEDURES}@procedureCode)},
    value         SBC-AP-ELEMENTARY-PROCEDURE.&UnsuccessfulOutcome  ({SBC-AP-ELEMENTARY-
PROCEDURES}@procedureCode)}
}

-- *****
--
-- Interface Elementary Procedure List
--
-- *****

SBC-AP-ELEMENTARY-PROCEDURES SBC-AP-ELEMENTARY-PROCEDURE ::= {
    SBC-AP-ELEMENTARY-PROCEDURES-CLASS-1 |
    SBC-AP-ELEMENTARY-PROCEDURES-CLASS-2 ,
    ...
}

SBC-AP-ELEMENTARY-PROCEDURES-CLASS-1 SBC-AP-ELEMENTARY-PROCEDURE ::= {
    write-Replace-Warning ,
    ...
}

SBC-AP-ELEMENTARY-PROCEDURES-CLASS-2 SBC-AP-ELEMENTARY-PROCEDURE ::= {
    ...
}

write-Replace-Warning SBC-AP-ELEMENTARY-PROCEDURE ::= {
    INITIATING MESSAGE  Write-Replace-Warning-Request
    SUCCESSFUL OUTCOME  Write-Replace-Warning-Response
}

```



```

    PROCEDURE CODE      id-Write-Replace-Warning
    CRITICALITY         reject
}
END

```

4.4.4 PDU definitions

```

-- *****
--
-- PDU definitions for SBC-AP.
--
-- *****

SBC-AP-PDU-Contents {
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0)
eps-Access (21) modules (3) sbc-AP (3) version1 (1) sbc-AP-PDU-Contents (1)}
  Spec Manager's note:
    "sbc-AP-PDU-Contents" has value (1) above; contrast with value (0) in clause 4.4.3.

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- *****
--
-- IE parameter types from other modules.
--
-- *****

IMPORTS

    Cause,
    Criticality-Diagnostics,
    Data-Coding-Scheme,
    Message-Identifier,
    Serial-Number,
    List-of-TAIs,
    Warning-Area-List,
    Repetition-Period,
    Number-of-Broadcasts-Requested,
    Warning-Type,
    Warning-Security-Information,
    Warning-Message-Content
FROM SBC-AP-IEs

    ProtocolExtensionContainer{},
    ProtocolIE-Container{},
    SBC-AP-PROTOCOL-EXTENSION,
    SBC-AP-PROTOCOL-IES
FROM SBC-AP-Containers

    id-Criticality-Diagnostics,
    id-Cause,
    id-Data-Coding-Scheme,
    id-List-of-TAIs,
    id-Message-Identifier,
    id-Serial-Number,
    id-Number-of-Broadcasts-Requested,
    id-Radio-Resource-Loading-List,
    id-Recovery-Indication,
    id-Repetition-Period,
    id-Warning-Area-List,
    id-Warning-Message-Content,
    id-Warning-Security-Information,
    id-Warning-Type
FROM SBC-AP-Constants;

-- *****
--
-- Write-Replace-Warning-Request
--
-- *****

```

```

Write-Replace-Warning-Request ::= SEQUENCE {
    protocolIEs          ProtocolIE-Container      { {Write-Replace-Warning-Request-IEs} },
    protocolExtensions   ProtocolExtensionContainer { {Write-Replace-Warning-Request-Extensions} }
} OPTIONAL,
...
}

Write-Replace-Warning-Request-IEs SBC-AP-PROTOCOL-IES ::= {
    { ID id-Message-Identifier CRITICALITY reject TYPE Message-Identifier PRESENCE mandatory } |
    { ID id-Serial-Number CRITICALITY reject TYPE Serial-Number PRESENCE mandatory } |
    { ID id-List-of-TAIs CRITICALITY reject TYPE List-of-TAIs PRESENCE optional } |
    { ID id-Warning-Area-List CRITICALITY ignore TYPE Warning-Area-List PRESENCE optional } |
    { ID id-Repetition-Period CRITICALITY reject TYPE Repetition-Period PRESENCE mandatory } |
    { ID id-Number-of-Broadcasts-Requested CRITICALITY reject TYPE Number-of-Broadcasts-Requested PRESENCE mandatory } |
    { ID id-Warning-Type CRITICALITY ignore TYPE Warning-Type PRESENCE optional } |
    { ID id-Warning-Security-Information CRITICALITY ignore TYPE Warning-Security-
Information PRESENCE optional } |
    { ID id-Data-Coding-Scheme CRITICALITY ignore TYPE Data-Coding-Scheme PRESENCE optional } |
    { ID id-Warning-Message-Content CRITICALITY ignore TYPE Warning-Message-Content PRESENCE optional } ,
    ...
}

Write-Replace-Warning-Request-Extensions SBC-AP-PROTOCOL-EXTENSION ::= {
    ...
}

-- *****
--
-- Write-Replace-Warning-Response
--
-- *****

Write-Replace-Warning-Response ::= SEQUENCE {
    protocolIEs          ProtocolIE-Container      { {Write-Replace-Warning-Response-IEs} },
    protocolExtensions   ProtocolExtensionContainer { {Write-Replace-Warning-Response-Extensions} }
} OPTIONAL,
...
}

Write-Replace-Warning-Response-IEs SBC-AP-PROTOCOL-IES ::= {
    { ID id-Message-Identifier CRITICALITY reject TYPE Message-Identifier PRESENCE mandatory } |
    { ID id-Serial-Number CRITICALITY reject TYPE Serial-Number PRESENCE mandatory } |
    { ID id-Cause CRITICALITY ignore TYPE Cause PRESENCE mandatory } |
    { ID id-Criticality-Diagnostics CRITICALITY ignore TYPE Criticality-Diagnostics PRESENCE
optional } ,
    ...
}

Write-Replace-Warning-Response-Extensions SBC-AP-PROTOCOL-EXTENSION ::= {
    ...
}

END

```

4.4.5 Information element definitions

```

-- *****
--
-- Information Element Definitions
--
-- *****

SBC-AP-IEs {
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0)
eps-Access (21) modules (3) sbc-AP (3) version1 (1) sbc-AP-IEs (2)}

```

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

maxNrOfErrors,
maxNrOfTAIs,
maxnoofTAIs,
maxnoofCellID,
maxnoofEmerAreaIDs,

id-TypeOfError

FROM SBC-AP-Constants

Criticality,
ProcedureCode,
TriggeringMessage,
ProtocolIE-ID

FROM SBC-AP-CommonDataTypes

ProtocolExtensionContainer{ },

SBC-AP-PROTOCOL-EXTENSION

FROM SBC-AP-Containers;

-- A

-- B

-- C

Cause ::= INTEGER {
message-accepted (0),
parameter-not-recognised (1),
parameter-value-invalid (2),
valid-message-not-identified (3),
tracking-area-not-valid (4),
unrecognised-message (5),
missing-mandatory-element (6),
mME-capacity-exceeded (7),
mME-memory-exceeded (8),
warning-broadcast-not-supported (9),
warning-broadcast-not-operational (10),
message-reference-already-used (11),
unspecified-error (12),
transfer-syntax-error (13),
semantic-error (14),
message-not-compatible-with-receiver-state (15),
abstract-syntax-error-reject (16),
abstract-syntax-error-ignore-and-notify (17),
abstract-syntax-error-falsely-constructed-message (18)
} (0..255)

Cell-ID-List ::= SEQUENCE (SIZE(1..maxnoofCellID)) OF EUTRAN-CGI

CellIdentity ::= BIT STRING (SIZE (28))

Criticality-Diagnostics ::= SEQUENCE {
procedureCode ProcedureCode OPTIONAL,
triggeringMessage TriggeringMessage OPTIONAL,
procedureCriticality Criticality OPTIONAL,
iEsCriticalityDiagnostics CriticalityDiagnostics-IE-List OPTIONAL,
iE-Extensions ProtocolExtensionContainer { {CriticalityDiagnostics-ExtIEs} } OPTIONAL,
...
}

CriticalityDiagnostics-ExtIEs SBC-AP-PROTOCOL-EXTENSION ::= {
...
}

CriticalityDiagnostics-IE-List ::= SEQUENCE (SIZE (1..maxNrOfErrors)) OF
SEQUENCE {
iECriticality Criticality,
iE-ID ProtocolIE-ID,

```

        typeOfError          TypeOfError,
        iE-Extensions        ProtocolExtensionContainer {{CriticalityDiagnostics-IE-Item-ExtIEs}}
OPTIONAL,
    ...
}

CriticalityDiagnostics-IE-Item-ExtIEs SBC-AP-PROTOCOL-EXTENSION ::= {
    ...
}

-- D
Data-Coding-Scheme          ::= BIT STRING (SIZE (8))

-- E
Emergency-Area-ID-List     ::= SEQUENCE (SIZE(1..maxnoofEmerAreaIDs)) OF Emergency-Area-ID
Emergency-Area-ID          ::= OCTET STRING (SIZE (3))

EUTRAN-CGI ::= SEQUENCE {
    pLMNidentity            PLMNidentity,
    cell-ID                 CellIdentity,
    iE-Extensions          ProtocolExtensionContainer { {EUTRAN-CGI-ExtIEs} } OPTIONAL,
    ...
}

EUTRAN-CGI-ExtIEs SBC-AP-PROTOCOL-EXTENSION ::= {
    ...
}

-- F
-- G
-- H
-- I
-- J
-- K
-- L

List-of-TAIs               ::= SEQUENCE (SIZE (1..maxNrOfTAIs)) OF
    SEQUENCE {
        tai                 OCTET STRING (SIZE (2))
    }

-- M

Message-Identifier ::= BIT STRING (SIZE (16))

-- N

Number-of-Broadcasts-Requested ::= INTEGER (1..65535)

-- O

-- P
PLMNidentity              ::= TBCD-STRING

-- Q

-- R

Repetition-Period         ::= INTEGER (0..4096)
-- 1 to 4096: Each unit represents a repetition of one second to a maximum of
-- once per 4096 seconds (~1 hour).
-- 0: no repetition

-- S

Serial-Number              ::= BIT STRING (SIZE (16))

-- T

```

```

TAC ::= OCTET STRING

TAI-List ::= SEQUENCE (SIZE(1..maxnoofTAIs)) OF TAI

TAI ::= SEQUENCE {
    plMNidentity          PLMNidentity,
    tAC                   TAC,
    iE-Extensions        ProtocolExtensionContainer { {TAI-ExtIEs} } OPTIONAL
}

TAI-ExtIEs SBC-AP-PROTOCOL-EXTENSION ::= {
    ...
}

TBCD-STRING ::= OCTET STRING (SIZE (3))

TypeOfError ::= ENUMERATED {
    not-understood,
    missing,
    ...
}

-- U

-- V

-- W

Warning-Area-List ::= SEQUENCE {
    cell-ID-List          Cell-ID-List          OPTIONAL,
    tAI-List              TAI-List              OPTIONAL,
    emergency-Area-ID-List Emergency-Area-ID-List OPTIONAL,
    iE-Extensions        ProtocolExtensionContainer { { Warning-Area-List-ExtIEs} } OPTIONAL,
    ...
}

Warning-Area-List-ExtIEs SBC-AP-PROTOCOL-EXTENSION ::= {
    ...
}

Warning-Message-Content ::= OCTET STRING (SIZE (1..9600))
Warning-Security-Information ::= OCTET STRING (SIZE (50))
Warning-Type ::= OCTET STRING (SIZE (2))

-- X

-- Y

END

```

4.4.6 Common definitions

```

-- *****
--
-- Common definitions
--
-- *****

SBC-AP-CommonDataTypes {
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0)
eps-Access (21) modules (3) sbc-AP (3) version1 (1) sbc-AP-CommonDataTypes (3)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

Criticality ::= ENUMERATED { reject, ignore, notify }

Presence ::= ENUMERATED { optional, conditional, mandatory }

ProcedureCode ::= INTEGER (0..255)

ProtocolExtensionID ::= INTEGER (0..65535)

ProtocolIE-ID ::= INTEGER (0..65535)

```

```
TriggeringMessage ::= ENUMERATED {initiating-message, successful-outcome, unsuccessful-
outcome, outcome}
```

```
END
```

4.4.7 Constant definitions

```
-- *****
--
-- Constant definitions
--
-- *****

SBC-AP-Constants {
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0)
eps-Access (21) modules (3) sbc-AP (3) version1 (1) sbc-AP-Constants (4)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- *****
--
-- Elementary Procedures
--
-- *****

id-Write-Replace-Warning          INTEGER ::= 0

-- *****
--
-- IEs
--
-- *****

id-Broadcast-Message-Content      INTEGER ::= 0
id-Cause                          INTEGER ::= 1
id-Criticality-Diagnostics        INTEGER ::= 2
id-Data-Coding-Scheme            INTEGER ::= 3
id-Failure-List                   INTEGER ::= 4
id-Message-Identifier             INTEGER ::= 5
id-Number-of-Broadcasts-Completed-List INTEGER ::= 6
id-Number-of-Broadcasts-Requested INTEGER ::= 7
id-Radio-Resource-Loading-List    INTEGER ::= 8
id-Recovery-Indication            INTEGER ::= 9
id-Repetition-Period              INTEGER ::= 10
id-Serial-Number                  INTEGER ::= 11
id-Service-Areas-List             INTEGER ::= 12
id-TypeOfError                    INTEGER ::= 13
id-List-of-TAIs                   INTEGER ::= 14
id-Warning-Area-List              INTEGER ::= 15
id-Warning-Message-Content        INTEGER ::= 16
id-Warning-Security-Information    INTEGER ::= 17
id-Warning-Type                   INTEGER ::= 18

-- *****
--
-- Extension constants
--
-- *****

-- *****
--
-- Lists
--
-- *****

maxNrOfErrors                     INTEGER ::= 256
maxnoofCellID                     INTEGER ::= 65535
maxNrOfTAIs                       INTEGER ::= 65535
maxnoofEmerAreaIDs                 INTEGER ::= 65535
maxnoofTAIs                        INTEGER ::= 65535

maxProtocolExtensions              INTEGER ::= 65535
maxProtocolIEs                     INTEGER ::= 65535
```

END

4.4.8 Container Definitions

```

-- *****
--
-- Container definitions
--
-- *****

SBC-AP-Containers {
itu-t (0) identified-organization (4) etsi (0) mobileDomain (0)
eps-Access (21) modules (3) sbc-AP (3) version1 (1) sbc-AP-Containers (5)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- *****
--
-- IE parameter types from other modules.
--
-- *****

IMPORTS
    Criticality,
    Presence,
    ProtocolExtensionID,
    ProtocolIE-ID
FROM SBC-AP-CommonDataTypes

    maxProtocolExtensions,
    maxProtocolIEs
FROM SBC-AP-Constants;

-- *****
--
-- Class Definition for Protocol IEs
--
-- *****

SBC-AP-PROTOCOL-IES ::= CLASS {
    &id          ProtocolIE-ID          UNIQUE,
    &criticality Criticality          DEFAULT ignore,
    &value,
    &presence    Presence
}
WITH SYNTAX {
    ID          &id
    CRITICALITY &criticality
    TYPE        &value
    PRESENCE    &presence
}

-- *****
--
-- Class Definition for Protocol Extensions
--
-- *****

SBC-AP-PROTOCOL-EXTENSION ::= CLASS {
    &id          ProtocolExtensionID    UNIQUE,
    &criticality Criticality          DEFAULT ignore,
    &extension,
    &presence    Presence
}
WITH SYNTAX {
    ID          &id
    CRITICALITY &criticality
    EXTENSION   &extension
    PRESENCE    &presence
}

-- *****
--
-- Container for Protocol IEs
--

```

```
-- *****
ProtocolIE-Container {SBC-AP-PROTOCOL-IES : IEsSetParam} ::=
  SEQUENCE (SIZE (0..maxProtocolIEs)) OF
  ProtocolIE-Field {{IEsSetParam}}

ProtocolIE-Field {SBC-AP-PROTOCOL-IES : IEsSetParam} ::= SEQUENCE {
  id          SBC-AP-PROTOCOL-IES.&id          ({IEsSetParam}),
  criticality SBC-AP-PROTOCOL-IES.&criticality  ({IEsSetParam}@id}),
  value       SBC-AP-PROTOCOL-IES.&Value       ({IEsSetParam}@id)}
}

-- *****
--
-- Container Lists for Protocol IE Containers
--
-- *****

ProtocolIE-ContainerList {INTEGER : lowerBound, INTEGER : upperBound, SBC-AP-PROTOCOL-IES :
IEsSetParam} ::=
  SEQUENCE (SIZE (lowerBound..upperBound)) OF
  ProtocolIE-Container {{IEsSetParam}}

-- *****
--
-- Container for Protocol Extensions
--
-- *****

ProtocolExtensionContainer {SBC-AP-PROTOCOL-EXTENSION : ExtensionSetParam} ::=
  SEQUENCE (SIZE (1..maxProtocolExtensions)) OF
  ProtocolExtensionField {{ExtensionSetParam}}

ProtocolExtensionField {SBC-AP-PROTOCOL-EXTENSION : ExtensionSetParam} ::= SEQUENCE {
  id          SBC-AP-PROTOCOL-EXTENSION.&id          ({ExtensionSetParam}),
  criticality SBC-AP-PROTOCOL-EXTENSION.&criticality  ({ExtensionSetParam}@id}),
  extensionValue SBC-AP-PROTOCOL-EXTENSION.&Extension  ({ExtensionSetParam}@id)}
}

END
```


4.4.9 Message transfer syntax

SBC-AP shall use the ASN.1 Basic Packed Encoding Rules (BASIC-PER) Aligned Variant as transfer syntax as specified in ref. [10].

4.5 Handling of unknown, unforeseen or erroneous protocol data

4.5.1 General

Protocol Error cases can be divided into three classes:

- Transfer Syntax Error;
- Abstract Syntax Error;
- Logical Error.

Protocol errors can occur in the following functions within a receiving node:

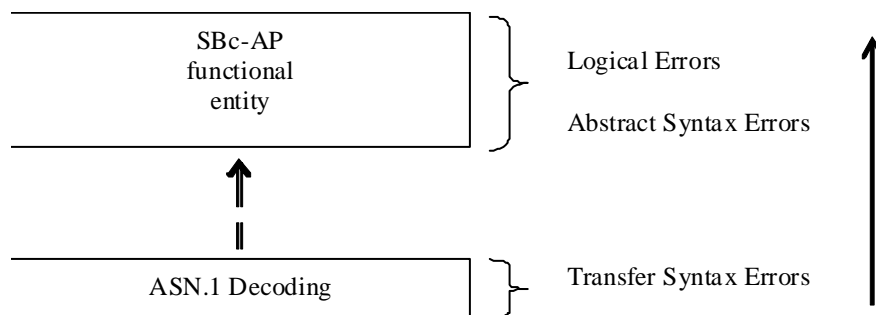


Figure 4.5.1-1: Protocol Errors in SBC-AP

The information stated in subclauses 4.5.2, 4.5.3 and 4.5.4, to be included in the message used when reporting an error, is what at minimum shall be included. Other optional information elements within the message may also be included, if available. This is also valid for the case when the reporting is done with a response message.

4.5.2 Transfer Syntax Error

A Transfer Syntax Error occurs when the receiver is not able to decode the received physical message. Transfer syntax errors are always detected in the process of ASN.1 decoding. If a Transfer Syntax Error occurs, the receiver should initiate Error Indication procedure with appropriate cause value for the Transfer Syntax protocol error.

4.5.3 Abstract Syntax Error

4.5.3.1 General

An Abstract Syntax Error occurs when the receiving functional SBC-AP entity:

1. receives IEs or IE groups that cannot be understood (unknown IE id);
2. receives IEs for which the logical range is violated (e.g.: ASN.1 definition: 0 to 15, the logical range is 0 to 10 (values 11 to 15 are undefined), and 12 will be received; this case will be handled as an abstract syntax error using criticality information sent by the originator of the message);
3. does not receive IEs or IE groups but according to the specified presence of the concerning object, the IEs or IE groups should have been present in the received message;

4. receives IEs or IE groups that are defined to be part of that message in wrong order or with too many occurrences of the same IE or IE group;
5. receives IEs or IE groups but according to the conditional presence of the concerning object and the specified condition, the IEs or IE groups should not have been present in the received message.

Cases 1 and 2 (not comprehended IE/IE group) are handled based on received Criticality information. Case 3 (missing IE/IE group) is handled based on Criticality information and Presence information for the missing IE/IE group specified in the version of the specification used by the receiver. Case 4 (IEs or IE groups in wrong order or with too many occurrences) and Case 5 (erroneously present conditional IEs or IE groups) result in rejecting the procedure.

If an Abstract Syntax Error occurs, the receiver shall read the remaining message and shall then for each detected Abstract Syntax Error act according to the Criticality Information and Presence Information for the IE/IE group due to which Abstract Syntax Error occurred in accordance with subclauses 4.5.3.4 and 4.5.3.5. The handling of cases 4 and 5 is specified in subclause 4.5.3.6.

4.5.3.2 Criticality information

In the SBc-AP messages there is criticality information set for individual IEs and/or IE groups. This criticality information instructs the receiver how to act when receiving an IE or an IE group that is not comprehended i.e. the entire item (IE or IE group) which is not (fully or partially) comprehended shall be treated in accordance with its own criticality information as specified in subclause 4.5.3.4.

In addition, the criticality information is used in case of the missing IE/IE group abstract syntax error (see subclause 4.5.3.5).

The receiving node shall take different actions depending on the value of the Criticality Information. The three possible values of the Criticality Information for an IE/IE group are:

- Reject IE;
- Ignore IE and Notify Sender;
- Ignore IE.

The following rules restrict when a receiving entity may consider an IE, an IE group or an EP not comprehended (not implemented), and when action based on criticality information is applicable:

1. IE or IE group: When one new or modified IE or IE group is implemented for one EP from a standard version, then other new or modified IEs or IE groups specified for that EP in that standard version shall be considered comprehended by the receiving entity (some may still remain unsupported).
2. EP: The comprehension of different EPs within a standard version or between different standard versions is not mandated. Any EP that is not supported may be considered not comprehended, even if another EP from that standard version is comprehended, and action based on criticality shall be applied.

4.5.3.3 Presence information

For many IEs/IE groups which are optional according to the ASN.1 transfer syntax, SBc-AP specifies separately if the presence of these IEs/IE groups is optional or mandatory with respect to MME/CBC application.

The presence field of the indicated classes supports three values:

1. Optional;
2. Conditional;
3. Mandatory.

If an IE/IE group is not included in a received message and the presence of the IE/IE group is mandatory or the presence is conditional and the condition is true according to the version of the specification used by the receiver, an abstract syntax error occurs due to a missing IE/IE group.

4.5.3.4 Not comprehended IE/IE group

4.5.3.4.1 Procedure code

The receiving node shall treat the different types of received criticality information of the *Procedure Code* according to the following:

Reject IE:

- If a message is received with a *Procedure Code* marked with "*Reject IE*" which the receiving node does not comprehend, the receiving node shall reject the procedure using the Error Indication procedure.

Ignore IE and Notify Sender:

- If a message is received with a *Procedure Code* marked with "*Ignore IE and Notify Sender*" which the receiving node does not comprehend, the receiving node shall ignore the procedure and initiate the Error Indication procedure.

Ignore IE:

- If a message is received with a *Procedure Code* marked with "*Ignore IE*" which the receiving node does not comprehend, the receiving node shall ignore the procedure.

When using the Error Indication procedure to reject a procedure or to report an ignored procedure it shall include the *Procedure Code IE*, the *Triggering Message IE*, and the *Procedure Criticality IE* in the *Criticality Diagnostics IE*.

4.5.3.4.2 Type of Message

When the receiving node cannot decode the *Type of Message IE*, the Error Indication procedure shall be initiated with an appropriate cause value.

4.5.3.4.3 IEs other than the Procedure Code and Type of Message

The receiving node shall treat the different types of received criticality information of an IE/IE group other than the *Procedure Code IE* and *Type of Message IE* according to the following:

Reject IE:

- If a message *initiating* a procedure is received containing one or more IEs/IE groups marked with "*Reject IE*" which the receiving node does not comprehend; none of the functional requests of the message shall be executed. The receiving node shall reject the procedure and report the rejection of one or more IEs/IE groups using the message normally used to report unsuccessful outcome of the procedure. In case the information received in the initiating message was insufficient to determine a value for all IEs that are required to be present in the message used to report the unsuccessful outcome of the procedure, the receiving node shall instead terminate the procedure and initiate the Error Indication procedure.
- If a message *initiating* a procedure that does not have a message to report unsuccessful outcome is received containing one or more IEs/IE groups marked with "*Reject IE*" which the receiving node does not comprehend, the receiving node shall terminate the procedure and initiate the Error Indication procedure.
- If a *response* message is received containing one or more IEs marked with "*Reject IE*" which the receiving node does not comprehend, the receiving node shall consider the procedure as unsuccessfully terminated and initiate local error handling.

Ignore IE and Notify Sender:

- If a message *initiating* a procedure is received containing one or more IEs/IE groups marked with "*Ignore IE and Notify Sender*" which the receiving node does not comprehend, the receiving node shall ignore the content of the not comprehended IEs/IE groups, continue with the procedure as if the not comprehended IEs/IE groups were not received (except for the reporting) using the understood IEs/IE groups, and report in the response message of the procedure that one or more IEs/IE groups have been ignored. In case the information received in the initiating message was insufficient to determine a value for all IEs that are required to be present in the response message, the receiving node shall instead terminate the procedure and initiate the Error Indication procedure.
- if a message *initiating* a procedure that does not have a message to report the outcome of the procedure is received containing one or more IEs/IE groups marked with "*Ignore IE and Notify Sender*" which the receiving node does not comprehend, the receiving node shall ignore the content of the not comprehended IEs/IE groups, continue with the procedure as if the not comprehended IEs/IE groups were not received (except for the reporting) using the understood IEs/IE groups, and initiate the Error Indication procedure to report that one or more IEs/IE groups have been ignored.
- If a *response* message is received containing one or more IEs/IE groups marked with "*Ignore IE and Notify Sender*" which the receiving node does not comprehend, the receiving node shall ignore the content of the not comprehended IE/IE groups, continue with the procedure as if the not comprehended IEs/IE groups were not received (except for the reporting) using the understood IEs/IE groups and initiate the Error Indication procedure.

Ignore IE:

- If a message *initiating* a procedure is received containing one or more IEs/IE groups marked with "*Ignore IE*" which the receiving node does not comprehend, the receiving node shall ignore the content of the not comprehended IEs/IE groups and continue with the procedure as if the not comprehended IEs/IE groups were not received using only the understood IEs/IE groups.
- If a *response* message is received containing one or more IEs/IE groups marked with "*Ignore IE*" which the receiving node does not comprehend, the receiving node shall ignore the content of the not comprehended IEs/IE groups and continue with the procedure as if the not comprehended IEs/IE groups were not received using the understood IEs/IE groups.

When reporting not comprehended IEs/IE groups marked with "*Reject IE*" or "*Ignore IE and Notify Sender*" using a response message defined for the procedure, the *Information Element Criticality Diagnostics* IE shall be included in the *Criticality Diagnostics* IE for each reported IE/IE group.

When reporting not comprehended IEs/IE groups marked with "*Reject IE*" or "*Ignore IE and Notify Sender*" using the Error Indication procedure, the *Procedure Code* IE, the *Triggering Message* IE, *Procedure Criticality* IE, and the *Information Element Criticality Diagnostics* IE shall be included in the *Criticality Diagnostics* IE for each reported IE/IE group.

4.5.3.5 Missing IE or IE group

The receiving node shall treat the missing IE/IE group according to the criticality information for the missing IE/IE group in the received message specified in the version of the present document used by the receiver:

Reject IE:

- if a received message *initiating* a procedure is missing one or more IEs/IE groups with specified criticality "*Reject IE*"; none of the functional requests of the message shall be executed. The receiving node shall reject the procedure and report the missing IEs/IE groups using the message normally used to report unsuccessful outcome of the procedure. In case the information received in the initiating message was insufficient to determine a value for all IEs that are required to be present in the message used to report the unsuccessful outcome of the procedure, the receiving node shall instead terminate the procedure and initiate the Error Indication procedure.
- if a received message *initiating* a procedure that does not have a message to report unsuccessful outcome is missing one or more IEs/IE groups with specified criticality "*Reject IE*", the receiving node shall terminate the procedure and initiate the Error Indication procedure.
- if a received *response* message is missing one or more IEs/IE groups with specified criticality "*Reject IE*", the receiving node shall consider the procedure as unsuccessfully terminated and initiate local error handling.

Ignore IE and Notify Sender:

- if a received message *initiating* a procedure is missing one or more IEs/IE groups with specified criticality "*Ignore IE and Notify Sender*", the receiving node shall ignore that those IEs are missing and continue with the procedure based on the other IEs/IE groups present in the message and report in the response message of the procedure that one or more IEs/IE groups were missing. In case the information received in the initiating message was insufficient to determine a value for all IEs that are required to be present in the response message, the receiving node shall instead terminate the procedure and initiate the Error Indication procedure.
- if a received message *initiating* a procedure that does not have a message to report the outcome of the procedure is missing one or more IEs/IE groups with specified criticality "*Ignore IE and Notify Sender*", the receiving node shall ignore that those IEs are missing and continue with the procedure based on the other IEs/IE groups present in the message and initiate the Error Indication procedure to report that one or more IEs/IE groups were missing.
- if a received *response* message is missing one or more IEs/IE groups with specified criticality "*Ignore IE and Notify Sender*", the receiving node shall ignore that those IEs are missing and continue with the procedure based on the other IEs/IE groups present in the message and initiate the Error Indication procedure to report that one or more IEs/IE groups were missing.

Ignore IE:

- if a received message *initiating* a procedure is missing one or more IEs/IE groups with specified criticality "*Ignore IE*", the receiving node shall ignore that those IEs are missing and continue with the procedure based on the other IEs/IE groups present in the message.
- if a received *response* message is missing one or more IEs/IE groups with specified criticality "*Ignore IE*", the receiving node shall ignore that those IEs/IE groups are missing and continue with the procedure based on the other IEs/IE groups present in the message.

When reporting missing IEs/IE groups with specified criticality "*Reject IE*" or "*Ignore IE and Notify Sender*" using a response message defined for the procedure, the *Information Element Criticality Diagnostics* IE shall be included in the *Criticality Diagnostics* IE for each reported IE/IE group.

When reporting missing IEs/IE groups with specified criticality "*Reject IE*" or "*Ignore IE and Notify Sender*" using the Error Indication procedure, the *Procedure Code* IE, the *Triggering Message* IE, *Procedure Criticality* IE, and the *Information Element Criticality Diagnostics* IE shall be included in the *Criticality Diagnostics* IE for each reported IE/IE group.

4.5.3.6 IEs or IE groups received in wrong order or with too many occurrences or erroneously present

If a message with IEs or IE groups in wrong order or with too many occurrences is received or if IEs or IE groups with a conditional presence are present when the condition is not met (i.e. erroneously present), the receiving node shall behave according to the following:

- If a message *initiating* a procedure is received containing IEs or IE groups in wrong order or with too many occurrences or erroneously present, none of the functional requests of the message shall be executed. The receiving node shall reject the procedure and report the cause value "Abstract Syntax Error (Falsely Constructed Message)" using the message normally used to report unsuccessful outcome of the procedure. In case the information received in the initiating message was insufficient to determine a value for all IEs that are required to be present in the message used to report the unsuccessful outcome of the procedure, the receiving node shall instead terminate the procedure and initiate the Error Indication procedure.
- If a message *initiating* a procedure that does not have a message to report unsuccessful outcome is received containing IEs or IE groups in wrong order or with too many occurrences or erroneously present, the receiving node shall terminate the procedure and initiate the Error Indication procedure, and use cause value "Abstract Syntax Error (Falsely Constructed Message)".
- If a *response* message is received containing IEs or IE groups in wrong order or with too many occurrences or erroneously present, the receiving node shall consider the procedure as unsuccessfully terminated and initiate local error handling.

When determining the correct order only the IEs specified in the specification version used by the receiver shall be considered.

4.5.4 Logical Error

Logical error situations occur when a message is comprehended correctly, but the information contained within the message is not valid (i.e. semantic error), or describes a procedure which is not compatible with the state of the receiver. In these conditions, the following behaviour shall be performed (unless otherwise specified) as defined by the class of the elementary procedure, irrespective of the criticality information of the IE's/IE groups containing the erroneous values.

Class 1:

Where the logical error occurs in a request message of a class 1 procedure, and the procedure has a message to report this unsuccessful outcome, this message shall be sent with an appropriate cause value. Typical cause values are:

- Semantic Error;
- Message not compatible with receiver state.

Where the logical error is contained in a request message of a class 1 procedure, and the procedure does not have a message to report this unsuccessful outcome, the procedure shall be terminated and the Error Indication procedure shall be initiated with an appropriate cause value. The *Procedure Code* IE and the *Triggering Message* IE within the *Criticality Diagnostics* IE shall then be included in order to identify the message containing the logical error.

Where the logical error exists in a response message of a class 1 procedure, the procedure shall be considered as unsuccessfully terminated and local error handling shall be initiated.

Class 2:

Where the logical error occurs in a message of a class 2 procedure, the procedure shall be terminated and the Error Indication procedure shall be initiated with an appropriate cause value. The *Procedure Code* IE and the *Triggering Message* IE within the *Criticality Diagnostics* IE shall then be included in order to identify the message containing the logical error.

4.5.5 Exceptions

The error handling for all the cases described hereafter shall take precedence over any other error handling described in the other subclauses of clause 4.5.

- If any type of error (Transfer Syntax Error, Abstract Syntax Error or Logical Error) is detected in the ERROR INDICATION message, it shall not trigger the Error Indication procedure in the receiving Node but local error handling.
- In case a response message or Error Indication message needs to be returned, but the information necessary to determine the receiver of that message is missing, the procedure shall be considered as unsuccessfully terminated and local error handling shall be initiated.
- If an error that terminates a procedure occurs, the returned cause value shall reflect the error that caused the termination of the procedure even if one or more abstract syntax errors with criticality "ignore and notify" have earlier occurred within the same procedure.

Annex A (informative): Change history:

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2008-12	CT#42	CP-080972			V1.0.2 approved in CT#42	1.0.2	8.0.0
2009-03	CT#43	CP-090101	0001	3	General clean-up to make an alignment with RAN specifications	8.0.0	8.1.0
		CP-090022	0002		General clean-up to make an alignment with RAN specifications		

History

Document history		
V8.0.0	January 2009	Publication
V8.1.0	April 2009	Publication