ETSI TS 134 123-3 V3.0.0 (2002-12)

Technical Specification

**Universal Mobile Telecommunications System (UMTS);
User Equipment (UE) conformance specification;
Part 3: Abstract Test Suite (ATS)
(3GPP TS 34.123-3 version 3.0.0 Release 1999)**

Reference
DTS/TSGT-0134123-3v300

Keywords
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, send your comment to:
editor@etsi.org

Copyright Notification

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).

All published ETSI deliverables shall include information which directs the reader to the above source of information.

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 www.etsi.org/key .

Contents

Intellectual Property Rights 2
Foreword 2
Foreword 9
Introduction 9
1 Scope 10
2 References 10
3 Definitions and abbreviations 12
3.1 Definitions 12
3.2 Abbreviations 12
4 Requirements on the TTCN development 12
5 ATS structure 13
5.1 Modularity 13
5.1.1 Module structure 13
5.1.2 Contents of the modules 15
5.1.3 Example of a working platform 15
6 Test method and Testing Architecture 16
6.1 Test method 16
6.2 Testing Architecture 16
6.2.1 Lower tester 17
6.2.2 Configuration and initialisation 17
6.2.3 Upper tester 17
6.2.4 TTCN 18
6.2.5 Model extension 18
6.2.6 Multiplexing of RLC services 18
6.3 NAS test method and architecture 18
6.3.1 Test configuration 18
6.3.2 Routing UL NAS massages in SS 19
6.4 RRC and RAB test method and architecture 20
6.4.1 Test configuration 20
6.4.2 RAB test method 21
6.4.2.1 Sending data on the same TTI 21
6.4.2.2 Sending continuous data on consecutive TTIs 21
6.5 RLC test method and architecture 22
6.5.1 Testing architecture 22
6.5.2 Test method 23
6.5.2.1 Handling SUFIs in TTCN 26
6.6 SMS test method and architecture 27
6.6.1 SMS CS test method and architecture 27
6.6.2 SMS PS test method and architecture 27
6.6.3 SMS Cell broadcasting test method and architecture 27
6.7 MAC test method and architecture 27
6.7.1 Testing architecture 27
6.7.2 Test method 28
6.7.2.1 Abnormal decoding situations 29
6.8 BMC test method and architecture 29
6.8.1 BMC test architecture 29
6.8.2 BMC test method 30
6.9 PDCP test 33
6.9.1 PDCP test architecture 33
6.9.2 PDCP test method 33
6.10 Multi-RAT Handover Test Model 34

6.10.1 Overview 34
6.10.2 ASP function description 35
6.10.2.1 Identities....................35
6.10.2.2 Cell configuration and control....................35
6.10.2.3 L1 (GERAN) configuration and control 35
6.10.2.3.1 Basic physical channel configuration 36
6.10.2.3.2 Multislot configuration for circuit switched channels 36
6.10.2.3.3 Frame in the near future....................36
6.10.2.3.4 L1 header....................36
6.10.2.4 L2 configuration and control....................37
6.10.2.4.1 Don't response to some handover access bursts....................37
6.10.2.4.1 No UA reply to SABM....................37
6.10.2.5 System Information sending....................37
6.10.2.6 Paging 37
7 PCO and ASP definitions....................39
7.1 NAS PCO and ASP definitions 39
7.1.1 NAS PCO Definitions....................39
7.1.2 Primitives used at Dc PCO 39
7.2 Ut PCO and ASP definitions 40
7.2.1 Ut PCO Declarations 40
7.2.2 Primitives used at Ut PCO 40
7.3 RRC PCO and ASP definitions 41
7.3.1 AM/UM/TM PCO and ASP definitions....................41
7.3.1.1 SAP and PCO for data transmission and reception 41
7.3.2 Control PCO and ASP 42
7.3.2.1 SAP and PCO for control primitives transmission and reception 42
7.3.2.2 Control ASP Type Definition....................43
7.3.2.2.1 CPHY_AICH_AckModeSet....................43
7.3.2.2.2 CPHY_Cell_Config....................43
7.3.2.2.3 CPHY_Cell_Config....................44
7.3.2.2.4 CPHY_Ini....................45
7.3.2.2.5 CPHY_Cell_TxPower_Modify 45
7.3.2.2.6 CPHY_Frame_Number 45
7.3.2.2.7 CPHY_Out_of_Sync 46
7.3.2.2.8 CPHY_PRACH_Measurement 46
7.3.2.2.9 CPHY_RL_Modify 47
7.3.2.2.10 CPHY_RL_Release....................48
7.3.2.2.11 CPHY_RL_Setup 49
7.3.2.2.12 CPHY_Sync 52
7.3.2.2.13 CPHY_TrCH_Config....................52
7.3.2.2.14 CPHY_TrCH_Release....................54
7.3.2.2.15 CMAC_BMC_Scheduling....................54
7.3.2.2.16 CMAC_Ciphering_Activate....................55
7.3.2.2.17 CMAC_Config 56
7.3.2.2.18 CMAC_PAGING_Config 59
7.3.2.2.19 CMAC_Restriction....................60
7.3.2.2.20 CMAC_SecurityMode_Config....................61
7.3.2.2.21 CMAC_SequenceNumber....................62
7.3.2.2.22 CMAC_SYSINFO_Config....................62
7.3.2.2.23 CRLC_Ciphering_Activate 63
7.3.2.2.24 CRLC_Config 63
7.3.2.2.25 CRLC_Integrity_Activate 65
7.3.2.2.26 CRLC_Integrity_Failure....................65
7.3.2.2.27 CRLC_Resume....................66
7.3.2.2.28 CRLC_SecurityMode_Config....................66
7.3.2.2.29 CRLC_SequenceNumber 67
7.3.2.2.30 CRLC_Status....................67
7.3.2.2.31 CRLC_Suspend....................67
7.3.2.2.32 CBMC_Config 68
7.3.2.2.33 RLC_TR_DATA 68
7.3.2.2.34 RLC_AM_DATA....................69

7.3.2.2.35 RLC_UM_DATA 70
7.3.3 TTCN Primitives 71
7.3.3.1 UTRAN TTCN Primitives 71
7.3.4 GERAN PCO and ASP definitions 72
7.3.4.1 PCO Type definitions 72
7.3.4.1.1 PCO type for data transmission and reception in GERAN 72
7.3.4.1.2 PCO type for configuration and control in GERAN 72
7.3.4.2 PCO definitions 72
7.3.4.2.1 PCOs for data transmission and reception in GERAN 72
7.3.4.2.2 PCOs for control primitives transmission and reception in GERAN 73
7.3.4.3 GERAN ASP Definitions 74
7.3.4.3.1 ASPs for data transmission and reception in GERAN 74
7.3.4.3.2 ASPs for control primitive transmission and reception in GERAN 84

8 Design Considerations 101
8.1 Channel mapping 101
8.2 Channel and RB identity 101
8.2.1 Physical Channels 104
8.2.2 Transport Channels 105
8.2.3 Logical Channels 105
8.2.4 Radio bearers 106
8.2.5 Scrambling and channelization codes 106
8.2.6 MAC-d 109
8.3 Channels configurations 109
8.3.1 Configuration of Cell_FACH 109
8.3.2 Configuration of Cell_DCH_StandAloneSRB 110
8.3.3 Configuration of Cell_DCH_Speech 110
8.3.4 Configuration of Cell_DCH_64kCS_RAB_SRB 111
8.3.5 Configuration of Cell_DCH_57_6kCS_RAB_SRB 112
8.3.6 Configuration of Cell_RLC_DCH_ RAB 113
8.3.7 Configuration of Cell_FACH_BMC 114
8.3.8 Configuration of PS Cell_DCH_64kPS_RAB_SRB and Cell_PDCP_AM_RAB 115
8.3.9 Configuration of Cell_Two_DTCH 116
8.3.10 Configuration of Cell_Single_DTCH (CS) 117
8.3.11 Configuration of PS Cell_PDCP_UM_RAB 117
8.3.12 Configuration of PS Cell_PDCP_AM_UM_RAB 118
8.3.13 Configuration of Cell_2SCCPCH_BMC 119
8.3.14 Configuration of Cell_Four_DTCH_CS_PS 120
8.3.15 Configuration of Cell_Two_DTCH_CS_PS 121
8.3.16 Configuration of Cell_Four_DTCH_CS 122
8.3.17 Configuration of Cell_DCH_MAC_SRB 123
8.3.18 Configuration of Cell_FACH_MAC_SRB 124
8.3.19 Configuration of Cell_FACH_MAC_SRB0 125
8.3.20 Configuration of Cell_FACH_2_SCCPCH_StandAlonePCH 125
8.3.21 Configuration of PS Cell_DCH_MAC_2AM_PS 126
8.4 System information blocks scheduling 127
8.4.1 Grouping SIBs for testing 127
8.4.2 SIB configurations 127
8.4.3 Test SIB default schedule 128
8.4.4 Test SIB special schedule 129
8.4.4.1 Test SIB schedule for two S-CCPCH or two PRACH 129
8.4.4.2 Test SIB schedule for Inter-Rat Handover Test 130
8.4.5 Handling the transmission of SIB 130
8.4.5.1 Delivery of System Information content 130
8.4.5.2 Scheduling of System Information Blocks 131
8.4.5.3 Example of usage 131
8.5 Security in testing 132
8.5.1 Authentication 132
8.5.2 Ciphering 132
8.5.3 Integrity 132
8.5.4 Counter check 132
8.5.5 Test USIM configurations 132

8.5.5.1 Test USIM for Idle mode tests 132
8.6 Downlink power control in SS 134
8.7 Test suite operation definitions 134
8.7.1 Test suite operation definitions in the module BasicM....134
8.7.2 Specific test suite operation definitions for Multi RAT Handover testing....146
8.7.3 Specific test suite operation for RLC....147
8.7.3.1 Pseudocode in a C like notation 147
8.7.4 Specific test suite operation for MAC....149
8.8 AT commands 150
8.9 Bit padding....151
8.9.1 The requirements for implementation....151
8.10 Test PDP contexts 152

Annex A (normative): Abstract Test Suites (ATS)....154

A.1 Version of specifications....154

A.2 NAS ATS 154
A.2.1 The TTCN Graphical form (TTCN.GR) 154
A.2.2 The TTCN Machine Processable form (TTCN.MP) 154

A.3 SMS ATS 154
A.3.1 The TTCN Graphical form (TTCN.GR) 154
A.3.2 The TTCN Machine Processable form (TTCN.MP) 154

A.4 RRC ATS 155
A.4.1 The TTCN Graphical form (TTCN.GR) 155
A.4.2 The TTCN Machine Processable form (TTCN.MP) 155

A.5 RLC ATS....155
A.5.1 The TTCN Graphical form (TTCN.GR) 155
A.5.2 The TTCN Machine Processable form (TTCN.MP) 155

A.6 MAC ATS 156
A.6.1 The TTCN Graphical form (TTCN.GR) 156
A.6.2 The TTCN Machine Processable form (TTCN.MP) 156

A.7 BMC ATS 156
A.7.1 The TTCN Graphical form (TTCN.GR) 156
A.7.2 The TTCN Machine Processable form (TTCN.MP) 156

A.8 PDCP ATS 156
A.8.1 The TTCN Graphical form (TTCN.GR) 156
A.8.2 The TTCN Machine Processable form (TTCN.MP) 156

A.9 RAB ATS 157
A.9.1 The TTCN Graphical form (TTCN.GR) 157
A.9.2 The TTCN Machine Processable form (TTCN.MP) 157

Annex B (normative): Partial IXIT proforma 158

B.0 Introduction....158

B.1 Parameter values 159
B.1.1 BasicM Test Suite Parameter Declarations 159
B.1.2 L3M Test Suite Parameters Declarations....162
B.1.3 NAS Test Suite Parameters Declarations 163
B.1.4 SMS Test Suite Parameters Declarations 164
B.1.5 RRC Test Suite Parameters Declarations 164
B.1.6 PDCP Test Suite Parameters Declarations 166
B.1.7 BMC Test Suite Parameters Declarations 168
B.1.8 RRC Test Suite Parameters Declarations 168
B.1.9 RAB Test Suite Parameters Declarations....169
B.1.10 MMI questions 173

Annex C (informative): Additional information to IXIT....175

C.1 Identification Summary 175
C.2 Abstract Test Suite Summary 175
C.3 Test Laboratory 175
C.3.1 Test Laboratory Identification 175
C.3.2 Accreditation status of the test service 176
C.3.3 Manager of Test Laboratory 176
C.3.4 Contact person of Test Laboratory 176
C.3.5 Means of Testing 176
C.3.6 Instructions for Completion 177
C.4 Client 178
C.4.1 Client Identification 178
C.4.2 Client Test Manager 178
C.4.3 Client Contact person 178
C.4.4 Test Facilities Required 178
C.5 System Under Test 179
C.5.1 SUT Information 179
C.5.2 Limitations of the SUT 179
C.5.3 Environmental Conditions 180
C.6 Ancillary Protocols 181
C.6.1 Ancillary Protocols 1 181
C.6.2 Ancillary Protocols 2 181
Annex D (informative): PCTR Proforma 182
Annex E (informative): TTCN style guide for 3GPP ATS 183
E.1 Introduction 183
E.2 ETR 141 rules and applicability 183
E.2.1 Multiple words are separated by upper case letters at the start of each word 184
E.2.2 Identifiers shall be protocol standard aligned 185
E.2.3 Identifiers shall be distinguishing (use of prefixes) 186
E.2.4 Identifiers should not be too long (use standard abbreviations) 189
E.2.5 Test suite operations must not use global data 192
E.3 3GPP ATS implementation guidelines 195
E.3.1 Test case groups shall reflect the TSS&TP document 195
E.3.2 Test case names correspond to the clause number in the prose 196
E.3.3 Use standard template for test case and test step header 196
E.3.4 Do not use identical tags in nested CHOICE constructions 197
E.3.5 Incorrect usage of enumerations 198
E.3.6 Structured type as OCTETSTRING should not be used 198
E.3.7 Wildcards in PDU constraints for structured types should not be used 198
E.3.8 TSOs should be passed as many parameters as meaningful to facilitate their implementation 199
E.3.9 Specification of Encoding rules and variation should be indicated 199
E.3.10 Use of global data should be limited 200
E.3.11 Limit ATS scope to a single layer / sub-layer 200
E.3.12 Place system information in specially designed data structures 200
E.3.13 Place channel configuration in specially designed data structures 200
E.3.14 PICS / PIXIT parameters 200
E.3.15 Dynamic vs. static choices 201
E.3.16 Definition of Pre-Ambles and Post Ambles 201
E.3.17 Use test steps to encapsulate AT and MMI commands 201
E.3.18 Use system failure guard timers 202
E.3.19 Mapping between prose specification and individual test cases 202
E.3.20 Verdict assignment 202
E.3.20.1 General 202
E.3.20.2 Test cases 203
E.3.20.3 Test steps 203
E.3.20.4 Defaults 204

E.3.21 Test suite and test case variables 205
E.3.22 Use of macros is forbidden 205
E.3.23 Support for future Radio Access Technologies 205
E.3.24 Managing multiple representations of the same information 205
E.3.24.1 Predefined types 206
E.3.24.2 Simple types 206
E.3.24.3 Structured types 206
E.3.24.4 Conversion responsibility 206
E.3.24.5 Option 1: Calling party conversions 206
E.3.24.5.1 Advantages 207
E.3.24.5.2 Disadvantages 207
E.3.24.6 Option 2: Called party conversions 207
E.3.24.6.1 Advantages 207
E.3.24.6.2 Disadvantages 207
E.3.25 Assignment using constraint 207
E.3.26 Guidelines for use of timers when tolerances are applicable 207
E.3.26.1 Specific situations 208
E.3.26.2 Example situations 208
E.3.26.2.1 Example of situation 1 208
E.3.26.2.2 Example of situation 2 209
E.3.26.2.3 Example of situation 3 209

Annex F (normative): MMI Command strings 210

F.1 Outgoing Call 210

F.2 Configure UE 210

F.3 PLMN 210

F.4 Power 210

F.5 USIM 211

F.6 SMS 211

F.7 Autocalling 211

F.8 Miscellaneous 211

Annex G (informative): Recommendation of an unique ICS/IXIT electronic exchange format 212

G.1 Syntax 212

G.2 Examples 213

Annex H (informative): Change history 214

History 215

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.

Introduction

The present document is 3rd part of a multi-part conformance test specification for UE. The specification contains a TTCN design frame work and the detailed test specifications in TTCN for UE at the Uu interface.

3GPP TS 34.123-1 [1] contains a conformance test description in prose for UE at the Uu interface.

3GPP TS 34.123-2 [2] contains a pro-forma for the UE Implementation Conformance Statement (ICS).

1 Scope

The present document specifies the protocol conformance testing in TTCN for the 3GPP User Equipment (UE) at the Uu interface.

The document is the 3rd part of a multi-part test specification, 3GPP TS 34.123. The following TTCN test specification and design considerations can be found in the present document:

- the overall test suite structure;
- the testing architecture;
- the test methods and PCO definitions;
- the test configurations;
- the design principles, assumptions, and used interfaces to the TTCN tester (System Simulator);
- TTCN styles and conventions;
- the partial PIXIT proforma;
- the TTCN.MP and TTCN.GR forms for the mentioned protocols tests.

The Abstract Test Suites designed in the document are based on the test cases specified in prose (3GPP TS 34.123-1 [1]).

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 TS 34.123-1: "User Equipment (UE) conformance specification; Part 1: Protocol conformance specification".

[2] 3GPP TS 34.123-2: "User Equipment (UE) conformance specification; Part 2: Implementation Conformance Statement (ICS) proforma specification".

[3] 3GPP TS 34.108: "Common test environments for User Equipment (UE) conformance testing".

[4] 3GPP TS 34.109: "Terminal logical test interface; Special conformance testing functions".

[5] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[6] 3GPP TS 23.003: "Numbering, Addressing and Identification".

[7] 3GPP TS 23.101: "General UMTS Architecture".

[8] 3GPP TS 24.007: "Mobile radio interface signalling layer 3; General aspects".

[9] 3GPP TS 24.008: "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3".

[10] 3GPP TS 24.011: "Point-to-Point (PP) Short Message Service (SMS) support on the mobile radio interface".

[11] 3GPP TS 24.012: "Short Message Service Cell Broadcast (SMSCB) support on the mobile radio interface".

[12] 3GPP TS 25.214: "Physical layer procedures (FDD)".

[13] 3GPP TS 25.224: "Physical layer procedures (TDD)".

[14] 3GPP TS 25.301: "Radio Interface Protocol Architecture".

[15] 3GPP TS 25.303: "Interlayer procedures in Connected Mode".

[16] 3GPP TS 25.304: "UE Procedures in Idle Mode and Procedures for Cell Reselection in Connected Mode".

[17] 3GPP TS 25.321: "Medium Access Control (MAC) protocol specification".

[18] 3GPP TS 25.322: "Radio Link Control (RLC) protocol specification".

[19] 3GPP TS 25.323: "Packet Data Convergence Protocol (PDCP) specification".

[20] 3GPP TS 25.324: "Broadcast/Multicast Control BMC".

[21] 3GPP TS 25.331: "Radio Resource Control (RRC) protocol specification".

[22] 3GPP TS 27.005: "Use of Data Terminal Equipment - Data Circuit terminating; Equipment (DTE-DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)".

[23] 3GPP TS 27.007: "AT command set for User Equipment (UE)".

[24] 3GPP TS 27.060: "Packet domain; Mobile Stations (MS) supporting Packet Switched services".

[25] 3GPP TS 33.102: "3G security; Security architecture".

[26] 3GPP TS 51.010-1: "Mobile Station (MS) conformance specification; Part 1: Conformance specification".

[27] ETSI TR 101 666 (V1.0.0): "Information technology; Open Systems Interconnection Conformance testing methodology and framework; The Tree and Tabular Combined Notation (TTCN) (Ed. 2++)".

[28] ITU-T Recommendation X.691 (1997) "Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)".

[29] ISO/IEC 8824: "Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)".

[30] IETF RFC 2507: "IP Header Compression".

[31] 3GPP TS 45.002: "Multiplexing and multiple access on the radio path".

[32] 3GPP TS 44.060: "General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol".

[33] 3GPP TS 44.064: "Mobile Station - Serving GPRS Support Node (MS-SGSN) Logical Link Control (LLC) Layer Specification".

[34] 3GPP TS 23.038: "Alphabets and language-specific information".

[35] 3GPP TS 23.040: "Technical realization of the Short Message Service (SMS)".

[36] 3GPP TS 23.041: "Technical realization of Cell Broadcast Service (CBS)".

[37] ETSI ETR 141: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; The Tree and Tabular Combined Notation (TTCN) style guide".

[38] ETSI TR 101 101: "Methods for Testing and Specification (MTS); TTCN interim version including ASN.1 1994 support [ISO/IEC 9646-3] (Second Edition Mock-up for JTC1/SC21 Review)".

[39] ITU-T Recommendation X.680: "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[40] 3GPP TS 25.211: "Physical channels and mapping of transport channels onto physical channels (FDD)".

[41] ISO/IEC 9646 (all parts): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework".

[42] 3GPP TS 44.006: "Mobile Station - Base Stations System (MS - BSS) Interface Data Link (DL) Layer Specification".

[43] 3GPP TS 44.018: "Mobile radio interface layer 3 specification; Radio Resource Control Protocol".

[44] 3GPP TR 25.925: "Radio Interface for Broadcast/Multicast Services".

[45] ITU-T Recommendation O.153: "Basic parameters for the measurement of error performance at bit rates below the primary rate".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 34.123-1 [1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TS 34.123-1 [1], 3GPP TS 24.008 [9], 3GPP TS 25.331 [21] and TR 101 666 [27] apply.

4 Requirements on the TTCN development

A number of requirements are identified for the development and production of TTCN specification for 3GPP UE at Uu interface.

1. Top-down design, following 3GPP TS 34.123-1 [1], 3GPP TS 34.108 [3] and 3GPP TS 34.109 [4].
2. A unique testing architecture and test method for testing all protocol layers of UE.
3. Uniform TTCN style and naming conventions.
4. Improve TTCN readability.
5. Using TTCN-2++ (TR 101 666 [27]) for R99, avoid the use of the TTCN 2 features TTCN 3 does not support.
6. TTCN specification feasible, implementable and compilable.
7. Test cases shall be designed in a way for easily adaptable, upwards compatible with the evolution of the 3GPP core specifications and the future Releases.
8. The test declarations, data structures and data values shall be largely reusable.
9. Modularity and modular working method.
10. NAS ATS should be designed being independent from the radio access technologies.

11. Minimising the requirements of intelligence on the emulators of the lower testers. Especially the functionality of the RRC emulator in the TTCN tester should be reduced and simplified, the behaviours should be standardised as the TTCN RRC test steps in the TTCN modular library.

12. Giving enough design freedom to the test equipment manufacturers.

13. Maximising reuse of ASN.1 definitions from the relevant core specifications.

In order to fulfil these requirements and to ensure the investment of the test equipment manufacturers having a stable testing architecture for a relatively long period, a unique testing architecture and test method are applied to the 3GPP UE protocol tests.

5 ATS structure

The total TTCN specification for the UE testing is structured in a number of separate layered ATSs. The number of ATS being produced corresponds to the number of the 3GPP core specifications referred. The separation of ATSs reduces the size of ATSs. The layer-specific test preambles and test data can be confined to one test suite and parallel development of test suites can be facilitated. The separation of ATSs enables also easily to follow the evolution of the core specifications.

NAS ATSs:

1) GSM MAP L3 ATS including MM, CC, GMM, SM test groups;
2) SMS ATS.

AS ATSs:

1) RRC ATS including Singlecell and multicell test group;
2) RLC ATS;
3) MAC ATS;
4) BMC ATS;
5) PDCP ATS;
6) RAB ATS.

5.1 Modularity

The modular TTCN approach is used for the development of the 3GPP ATS specification work. Two modules, BasicM and L3M are installed.

5.1.1 Module structure

The working area is shown in figure 1.

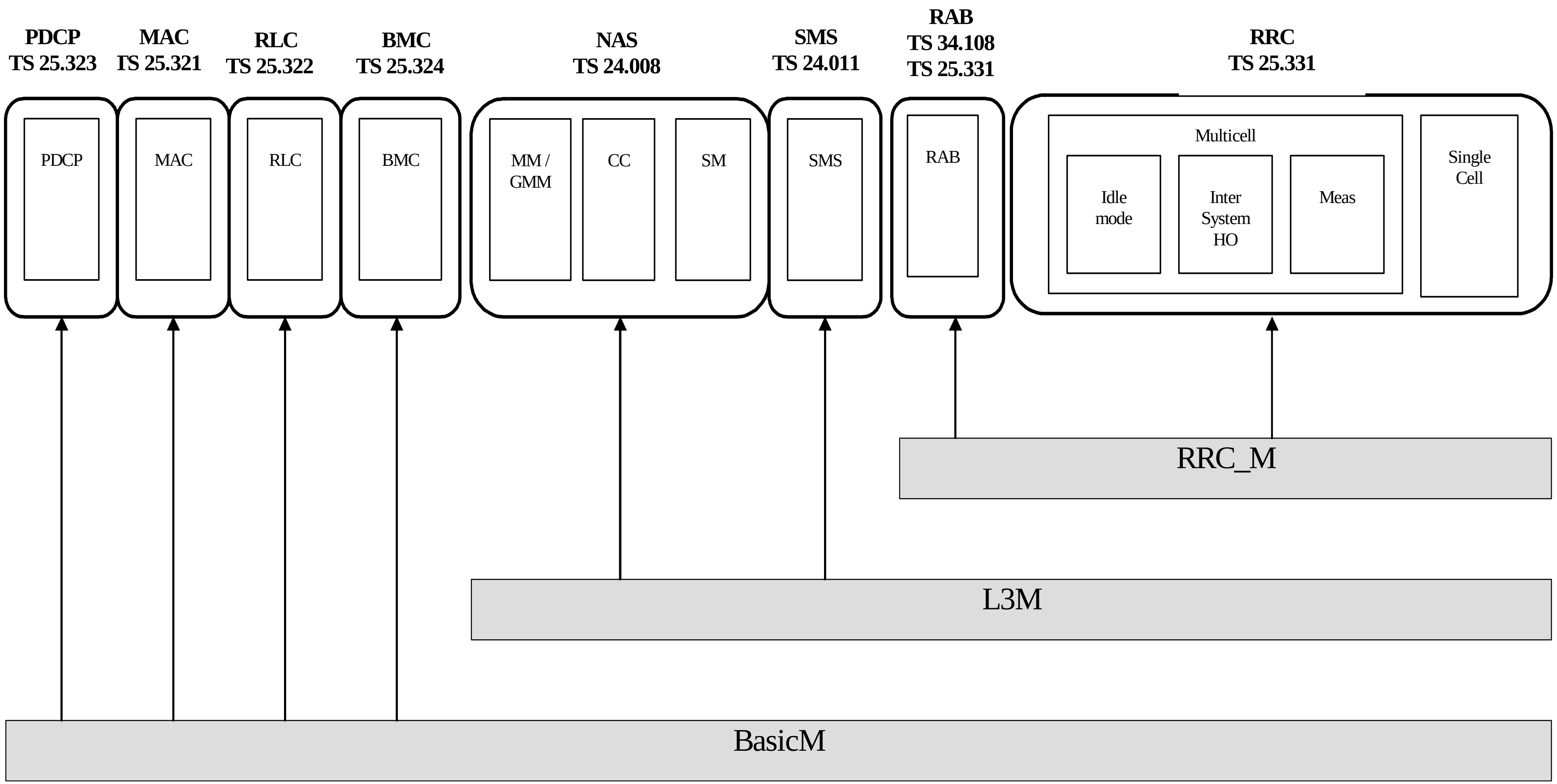

Figure 1: The proposed working area

The BasicM (**Basic M**odule) is a minimum module commonly for the layer 2 and layer 3 testing. The L3M (**L**ayer **3** **M**odule) contains all the items to be shared by the RRC, NAS and SMS ATSs. The RRC_M is a module containing common object for RRC and RAB ATSs.

5.1.2 Contents of the modules

The BasicM module includes objects related to the RRC, the layer 2 and the physical layer. It includes also all test steps needed by the layer 2 and layer 3 test cases for configurations and all objects related to the definition of the steps:

- Common test steps and default test steps defined as generic procedures in 3GPP TS 34.108 [3];
- RRC declarations related to the steps: types, timers, PDU types, ASP type, PCOs, TSOs, constants;
- Related ICS and IXIT parameters needed for testing and respectively defined in 3GPP TS 34.123-2 [2] and the present document;
- Defaults constraints based on the default message contents defined in 3GPP TS 34.108 [3];
- MMI PCO and ASPs;
- All TTCN objects related to the SS configuration, e.g. PCOs, declaration of the components.

The L3M module includes the NAS configuration steps and all related TTCN objects:

- Common test steps and default test steps defined as generic procedures in 3GPP TS 34.108 [3];
- NAS declarations related to these steps: types, PDU, ASP, PCOs, TSOs, constants;
- Related ICS and IXIT parameters needed for testing and respectively defined in 3GPP TS 34.123-2 [2] and the present document;
- Default constraints based on the default message contents defined in 3GPP TS 34.108 [3].

The RRC_M module includes the RRC steps common to RRC and Rab test cases and all related TTCN objects.

5.1.3 Example of a working platform

The figure 2 shows the working platform for the user that is writing the SMS test cases.

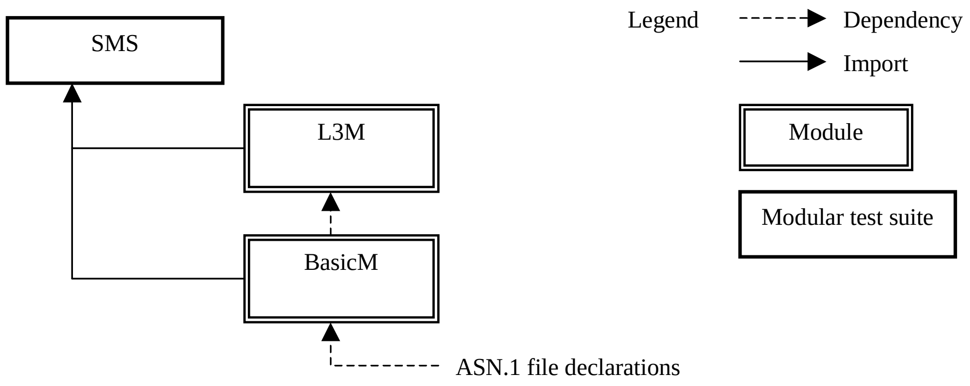

Figure 2: An example of working platform for SMS

6 Test method and Testing Architecture

6.1 Test method

The distributed single party test method is used for the UE testing. The lower tester configures the emulator and communicates with the UE under test via the emulator. An upper tester interfaces UE as (E)MMI.

All common parts in 3GPP TS 34.108 [3], 3GPP TS 34.109 [4] and 3GPP TS 34.123-2 [2] are developed in a TTCN library including the declarations, default constraints, preambles and postambles. They have the following characteristics:

- Very complex;
- Worked in different layers;
- Including data representing the radio parameters for SS setting and the data representing the UE capabilities (PICS parameters);
- Including the generic procedures to bring the UE into certain test states or a test mode (C-plane);
- Setting RABs at U-plane and SRBs in C-plane;
- Being used by every test cases no matter which layer the test case belongs to;
- No affect on the test verdict of PASS or FAIL.

The layer-specific test cases have the characteristics:

- relatively simple and straight forward;
- having narrow test scope and test purposes;
- test scenarios in a single layer (one PCO);
- assigning the test verdict.

6.2 Testing Architecture

A unique testing architecture is shown in figure 3.

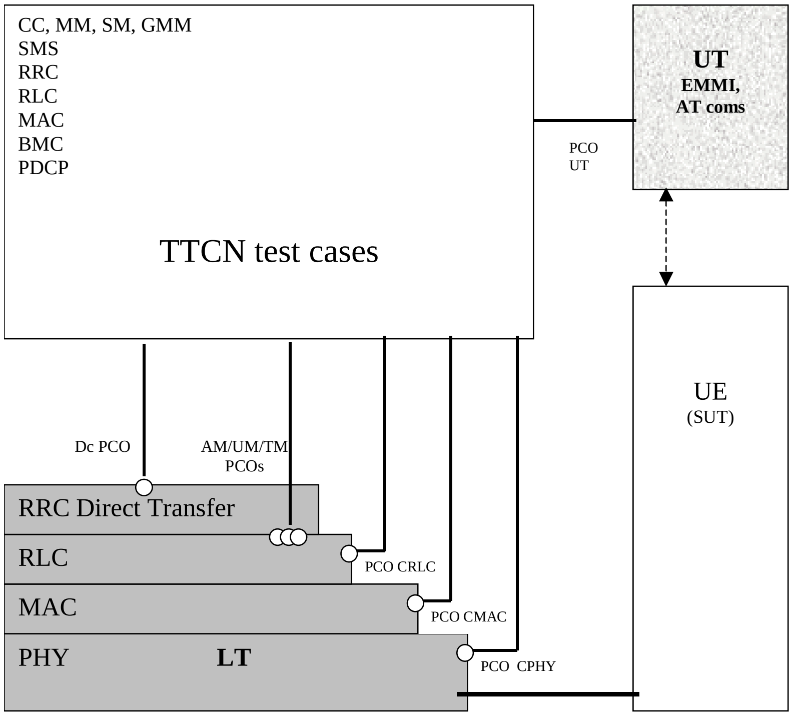

Figure 3: A unique testing architecture

6.2.1 Lower tester

The Lower Tester (LT) provides the test means for the execution of the test cases for CC, SM, MM, GMM, SMS, RRC, RLC, MAC, PDCP or BMC. The LT provides also the RLC, MAC and PHY emulators to communicate with the UE. The configuration and initialisation of the emulators are control by the TTCN via ASPs.

6.2.2 Configuration and initialisation

A number of TTCN test steps are designed for the generic setting.

1) Configuration of L1 of the tester, such as the cells, Physical channels and common transport channels via CPHY-PCO, configuration of MAC via CMAC-PCO and configuration of RLC layer via CRLC-PCO.

2) Sending system information via TR-PCO.

3) Establishment RRC connection via AM or UM-PCO.

4) Assigning a radio bearer via AM-PCO.

5) MM /GMM registration via Dc-PCO.

6) Establishment of a CS call or a PDP context via Dc-PCO.

7) Setting security parameters and control of integrity via CRLC- and ciphering via CRLC- and CMAC-PCO.

6.2.3 Upper tester

An upper tester (UT) exists in the test system. The UT interfaces toward UE with any optional EMMI (3GPP TS 34.109 [4], clause 7). TTCN communicates with the UT by passing coordination primitives via a Ut PCO. The primitives can either contain AT commands aiming at the automatic tests, or some informal commands as MMI, in order to request the UE for certain actions and to provide simple means for observations of UE.

6.2.4 TTCN

TTCN is used as specification language based on TR 101 666 [27] (TTCN 2++). The importation of ASN.1 modules and modular TTCN are two of the most important features used in the design of the ATSs.

The TTCN test suites have been designed to maximise the portability from the language TTCN 2 to TTCN 3.

6.2.5 Model extension

If a test case needs to handle a concurrent situation two or more LTs can be configured at the same time. The following test scenarios identified may require multiple testers in the test configuration.

6.2.6 Multiplexing of RLC services

For the RRC and NAS testing, the TTCN RRC test steps (on RB1 and RB2) and the RRC emulator (on RB3 and RB4 for the NAS messages) share the same service access point (AM SAP). The RLC emulator shall provide separate message queues (buffers) for the TTCN RRC test steps and the RRC emulator for the TTCN NAS test cases, according to the signalling radio bearer identities.

6.3 NAS test method and architecture

6.3.1 Test configuration

The NAS test method is shown in figure 4.

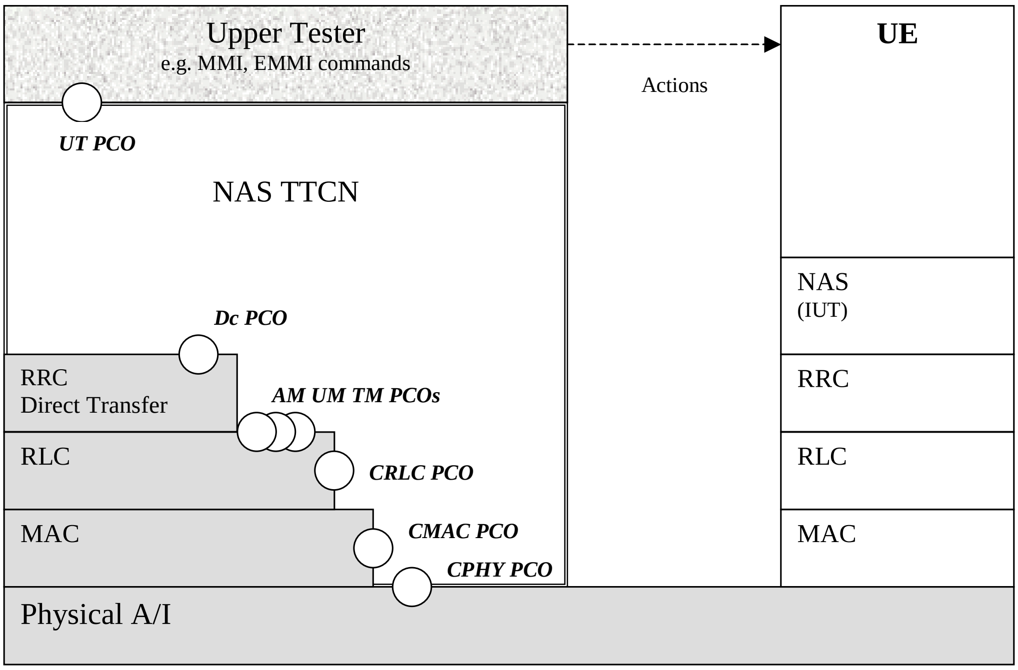

Figure 4: NAS testing architecture

The single layer distributed test method is used.

The Point of Control and Observation (PCO) are defined as the Dc (Dedicated control) SAP. The NAS test verdicts are assigned depending on the behaviours observed at the PCO.

The TTCN tester provides the NAS TTCN test cases and steps with a simple RRC direct transfer function which buffers the NAS PDU data, converts the data from the NAS TTCN table format into ASN.1, or in reverse way, and delivers all lower layer services of AM-SAP for RB3 and RB4.

The NAS TTCN test cases make also intensively use of the RRC TTCN test steps, in order to:

- Configure, initialise and control the L2 emulator;
- Initialise the UE for testing.

The RRC test steps, which are called by the NAS test cases or steps, interface with the RLC PCOs (UM, AM and TR), the control PCOs CRLC, CMAC and CPHY.

The General control (Gc) SAP and the Notification (Nt) SAP are not applied. Messages exchanged via these SAPs will be replaced with the corresponding RRC TTCN test steps.

The Ut PCO (so called logical interface [4]) is served as the interface to the UE EMMI to allow a remote control of operations, which have to be performed during execution of a test case such as to switch the UE on/off, initiate a call, etc.

6.3.2 Routing UL NAS massages in SS

The UL NAS messages are embedded in RRC messages INITIAL / UL DIRECT TRANSFER. In the UE test, the received UL NAS messages can either be routed to the Dc PCO and verified at the NAS message level, or routed to AM PCO and verified at the RRC message level.

1. RBid =3 at the SS side indicates that the UL NAS high priority messages to be routed to Dc PCO. RB3 applies to RRC_DataInd/Req.
2. RBid= -16 at the SS side indicates the received messages to be routed to RLC AM PCO. RB-16 applies to RLC_DataInd/Req.

The RB3 and RB-16 do not coexist. The TTCN writer uses the MAC and RLC reconfigurations to re-map the RB and the corresponding logical channels. If RB3 has been configured, but a test case needs to re-map the logical channel from RB3 to RB-16 the following way is to replace RB3 with RB-16.

CMAC_CONFIG_REQ (reconfiguration, RB-16),

Re-mapping on RB-16 which appears in the transport channel and logical channel mapping list.

CRLC_CONFIG_REQ (reconfiguration, RB-16)

RB-16 appears in the routing info, in order to replace the original mapping on RB3.

Mapping from RB-16 to RB3 is done in the reverse way.

6.4 RRC and RAB test method and architecture

6.4.1 Test configuration

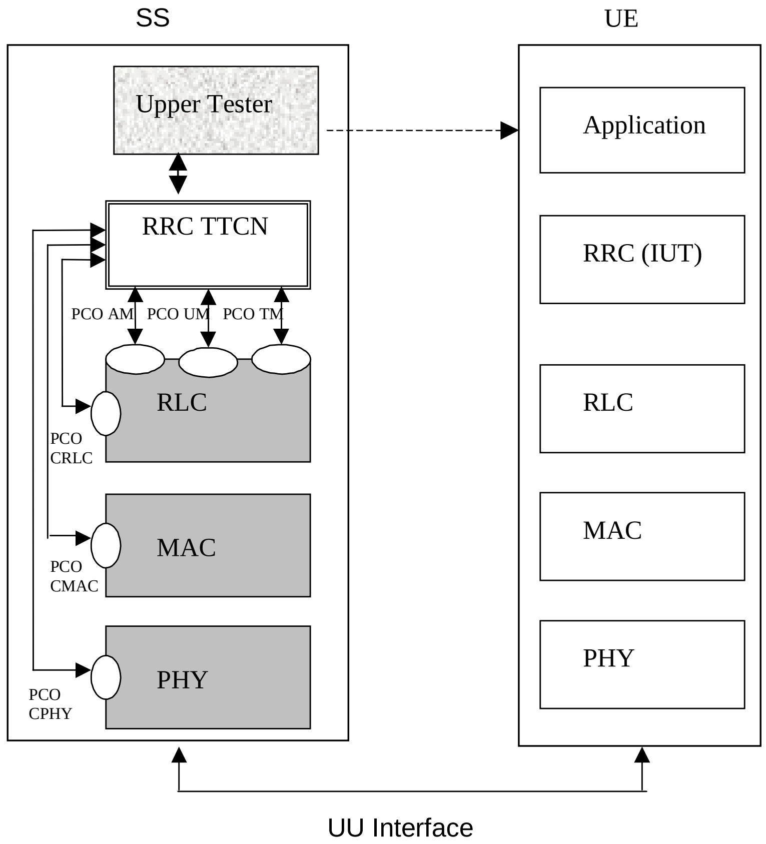

Figure 5: RRC testing architecture

The single layer distributed test method is used.

The PCOs are defined as the AM (Acknowledged Mode), UM (Unacknowledged Mode) and TM (Transparent Mode) SAPs. The RRC test verdicts are assigned depending on the behaviours observed at the PCO. The RRC TTCN interface also with the control PCOs CRLC, CMAC and CPHY, for the configuration, initialisation and control of the System Simulator.

The RRC TTCN test cases also make use of the NAS TTCN test steps in order to:

- Bring UE to Idle state;
- Bring UE to state U10.

The NAS test steps, which are called by the RRC test cases or steps, interface with the Dc PCO.

The Ut PCO (so called logical interface [4]) is served as the interface to the UE EMMI to allow a remote control of operations, which have to be performed during execution of a test case such as to switch the UE on/off, initiate a call, etc.

According to 3GPP TS 25.331 [21] clause 12.1.1, the encoding of RRC PDUs is obtained by applying UNALIGNED PER to the abstract syntax value as specified in ITU-T Recommendation X.691 [28]. The two tables below show the declaration of the encoding rule and an example of the use in the definition of an RRC PDU.

Table 1: PER_Unaligned Encoding Rule

Encoding Rule Name	PER_Unaligned
Reference	X.691 [28]
Default	
Comments	Packet encoding rules (X.691 [28]) unaligned and with adapted padding

Table 2: Definition of the RRC ASN.1 DL_DCCH_Message type by reference

PDU Name	DL_DCCH_Message
PCO Type	DSAP
Type Reference	DL-DCCH-Message
Module Identifier	Class-definitions
Enc Rule	PER_Unaligned
Enc Variation	

6.4.2 RAB test method

6.4.2.1 Sending data on the same TTI

The RAB test requires a specific test method to send the test data on the same TTI. The TFC restriction method is used in this case. A specific TFC subset is allowed to ensure the test data are sent on different RBs on the same TTI. The downlink restriction can be used to ensure that the SS uses a specific TFC for transmission of data, by only allowing the 'No data' TFC, and the 'desired' TFC. It may also be necessary to include one or more 'signalling only' TFCs to allow signalling to occur. The uplink restriction can be used to verify that the UE has used a specific TFC. Any data received by the SS using a forbidden TFCI shall be discarded.

6.4.2.2 Sending continuous data on consecutive TTIs

The RBS ATS is developed using the tabular TTCN notation. In order to test of multiple-RB combinations and simultaneous signalling, the SS shall be capable of sending continues test data in every TTI using the downlink transport format combination under test. A specific TSO is designed to request the SS sending continuous data. The information about the number of RLC SDUs and their sizes for each RAB will be provided to the system simulator through TSO.

6.5 RLC test method and architecture

6.5.1 Testing architecture

Figure 6 illustrates a typical realisation of the RLC ATS.

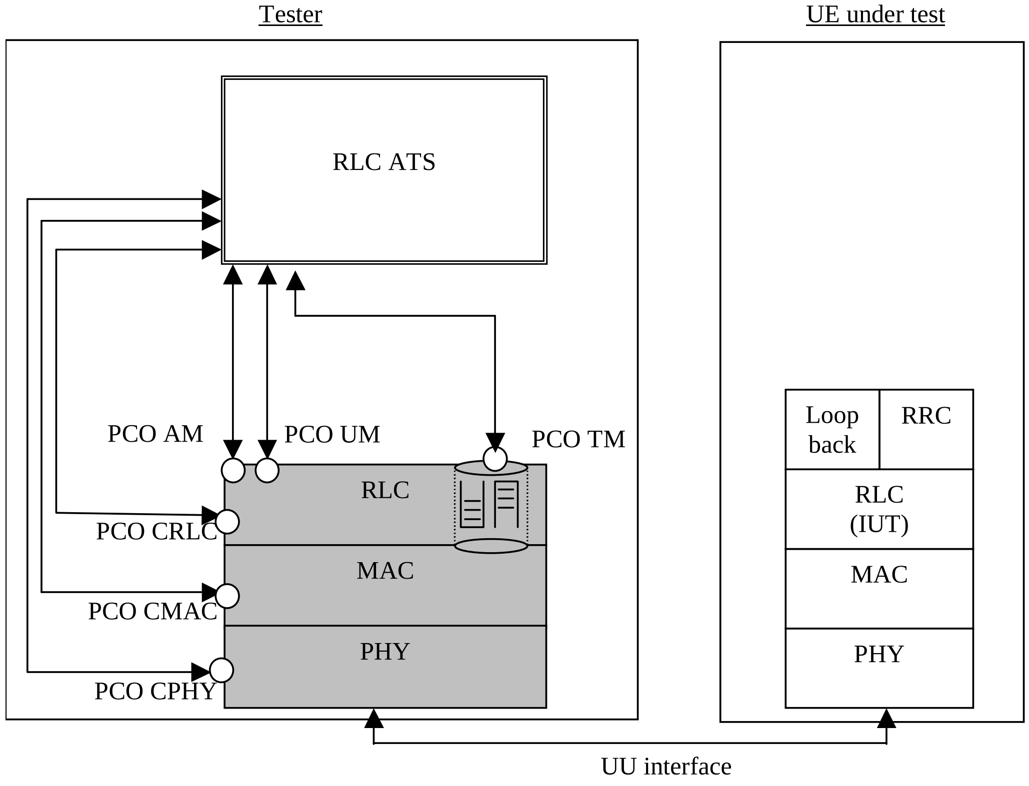

Figure 6: RLC ATS single party test method

The single party test method is used for RLC testing.

Separation of TTCN test cases from the configuration of the tester and initialisation of the UE is achieved by using test steps. For each RLC test case, common test steps will be used to perform the configuration of the tester and the appropriate generic setup procedures as described in 3GPP TS 34.108 [3]. These test steps will make use of PCOs AM, UM, TM, CRLC, CMAC, and CPHY.

Three PCOs are provided at the top of the RLC emulation in the tester, one corresponding to each of the available RLC modes: acknowledged, unacknowledged, and transparent. Routing information for different radio bearers used at these PCOs will be provided in ASP parameters.

The queues shown in the RLC emulation in figure 6 indicate that normal RLC transmit and receive buffering will be used to isolate the TTCN test suite from the real time issues involved if messages are sent directly to the MAC layer.

The RLC TTCN test cases make also use of the NAS TTCN test steps in order to bring UE to Idle state. The NAS test steps, which are called by the RLC test cases or steps, interface with the Dc PCO.

6.5.2 Test method

Figure 7 illustrates an example configuration for downlink UM testing. Uplink and AM tests will use similar configurations. A Tr-Entity is established on the tester side using a CRLC-CONFIG-REQ. A corresponding UM-Entity is created in the UE by sending a Radio Bearer Setup PDU. RLC PDUs are specified in the TTCN test suite, and sent to TM PCO. These PDUs shall be carefully designed so that the Tr-Entity will not perform any segmentation. The system simulator is responsible for direct encoding the abstract representation of transmitted PDUs into a bitstring to be sent by the Transmitting Tr entity. Direct encoding is performed by concatenation of all of the present fields in the abstract representation. It is the TTCN author's responsibility to ensure that the PDU is valid. To test reassembly in the UE side, the segmentation must be explicitly coded in TTCN. To test various aspects of the RLC header (e.g. sequence numbering, length indications etc), the RLC header must be explicitly coded in TTCN. Ciphering will not be tested using this approach, and will be disabled in the UE UM Entity.

The segmentation block in the SS Tr-entity is shown in grey to indicate that the functionality is present in the SS, but the test cases shall be carefully designed to ensure that segmentation is not used in the SS Tr-entity for RLC testing.

The deciphering block in the UE UM-entity is shown in grey to indicate that the functionality may be present in the UE, but shall be disabled for RLC testing.

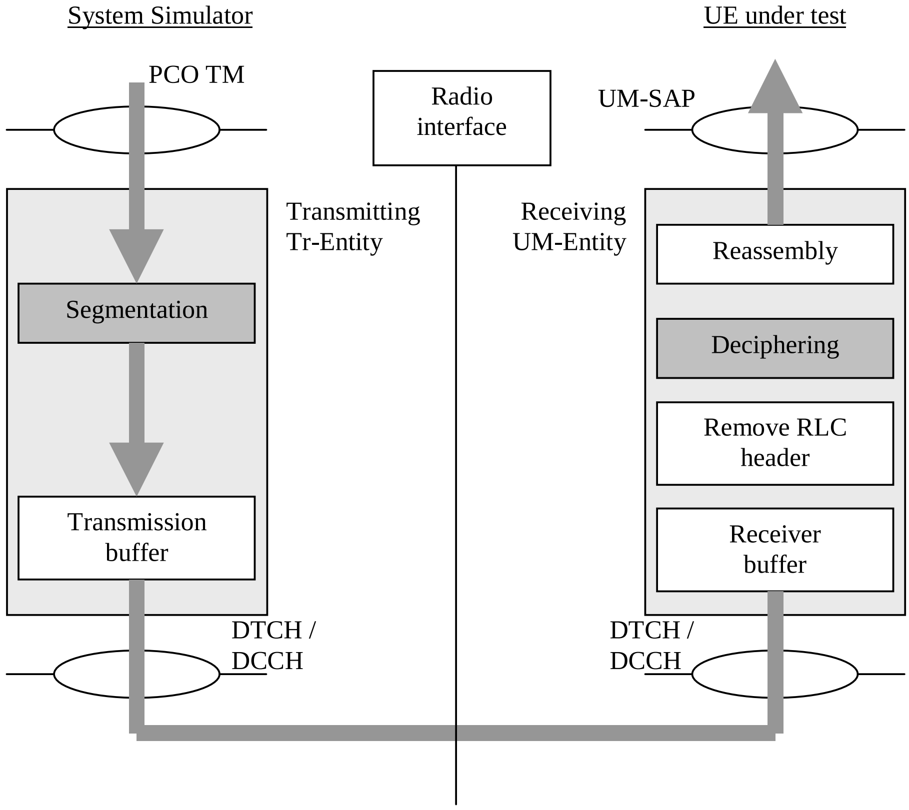

Figure 7: Example configuration for downlink RLC UM testing

The TFCS used for RLC testing must guarantee that Tr mode segmentation will not occur. This is to prevent transmission of more than one Tr PDU per TTI.

All RLC tests that require uplink data will make use of the UE test loop mode 1 defined in 3GPP TS 34.109 [4]. The UE test loop mode 1 function provides all upper tester (UT) functionality required, so an UT PCO is not required for RLC tests. Test Loop mode 1 is only available in the user plane, so all RLC tests will be performed in the user plane, using DTCH and DCCH logical channels mapped to DCH transport channels.

Ciphering will be disabled for all RLC test cases. Ciphering will be tested implicitly by other test cases that have ciphering enabled.

Figure 8 illustrates an example configuration for uplink UM testing, and reception of an example UMD PDU. Figure 9 illustrates an example configuration for uplink AM testing, reception of an example STATUS_PDU, and the use of the superFields and superFieldsRec fields.

The ciphering and deciphering blocks in the UE RLC entities are shown in grey to indicate that the functionality may be present in the UE, but shall be disabled for RLC testing.

The reassembly blocks in the SS Tr-entities are shown in grey to indicate that the functionality is present in the SS, but the test cases shall be carefully designed to ensure that reassembly is not used in the SS Tr-entity for RLC testing.

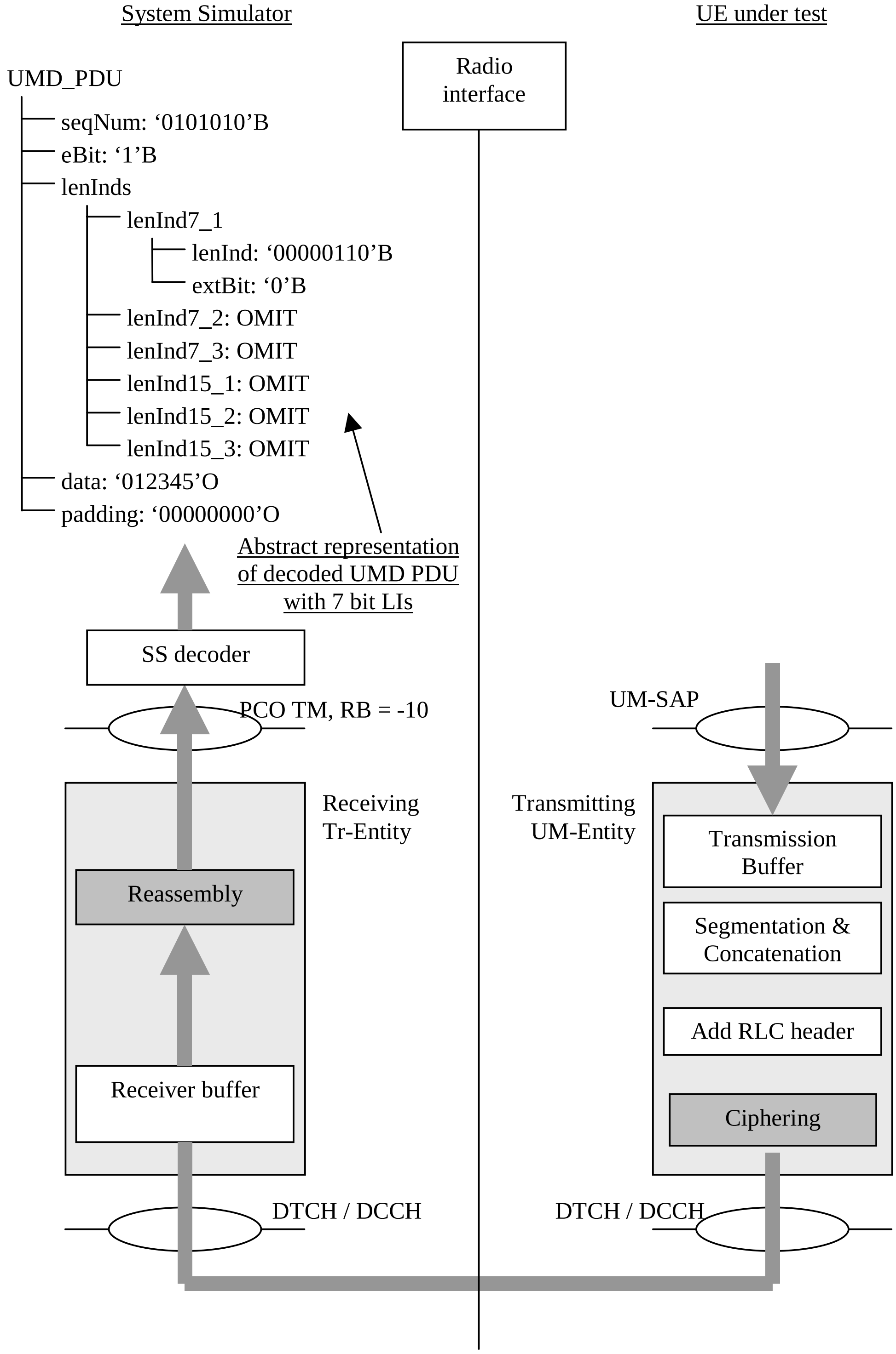

Figure 8: Example configuration for uplink RLC UM testing

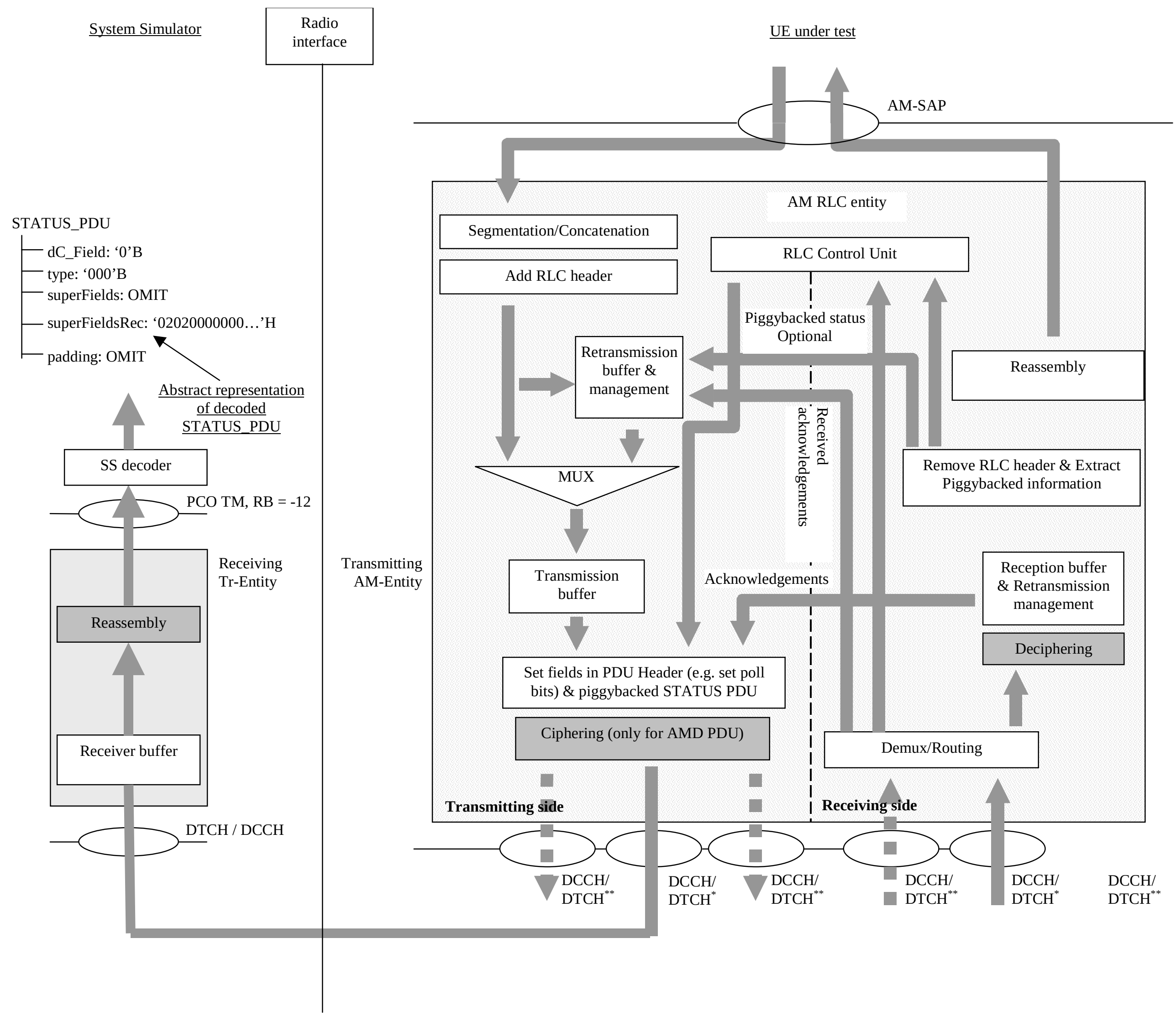

Figure 9: Example configuration for uplink RLC AM testing

Uplink data uses a similar approach to downlink, but the received data must be decoded in the correct way, depending on the current UE configuration. In the example in figure 8, the SS must decode the data received at the TM PCO into an abstract representation of the structure defined in the TTCN for a UMD_PDU, using 7 bit length indicators. This structure is then compared with an abstract representation of the expected data to see if the receive event is successful. Refer to TR 101 666 [27], clause B.5.2.10 for more information.

For RLC testing, the following RB Ids are used within the system simulator, depending on the RLC mode, and length indicator size being simulated.

RLC mode	LI Size	RB Id
UM	7	-10
UM	15	-11
AM	7	-12
AM	15	-13

The SS decoder can use the RB Id to determine which abstract structure to create during the decode process. The SS decoder must also understand the RLC peer-to-peer protocol enough to determine which fields are present.

EXAMPLE 1: The semantics of LI extension bits must be known to determine how many LIs are present.

EXAMPLE 2: The contents of the LIs must be interpreted to determine how many octets of data, and how many octets of padding are present.

The SUFI list and any subsequent padding in a received STATUS_PDU or PiggyBackedSTATUS_PDU shall be decoded as a HEXSTRING, and put in the 'superFieldsRec' field of the abstract representation of the STATUS PDU. The 'superFields' and 'padding' fields shall be omitted for received STATUS PDUs. This is illustrated in figure 9.

As in downlink testing, the TFCS must be defined to guarantee that the Tr entity does not perform any reassembly. This is to prevent reception of more than one Tr PDU per TTI so that the TTCN does not need to manage possible interleaving problems due to multiple PDUs received at the same time (i.e. they may be placed on the PCO queue in any order).

6.5.2.1 Handling SUFIs in TTCN

The SUFIs are a very flexible set of information elements contained in the RLC protocol. The order of the fields varies, the existance of a field may depend upon the presence of another one. A field can be present multiple times. For matching received SUFIs, it is convenient to define the SUFIs as an HEXSTRING which is treated by a TSO **o_SUFI_Handler**.

Depending upon which SUFIs and which aspects of SUFIs are to be checked, the TSO is provided with the information (**SUFI_Params**) on what checking it is expected to perform. If the check is successful the result TRUE will be returned, otherwise FALSE. Additionally the TSO will return an object which is structured as the SUFIs used in transmission (SuperFields). This will allow to make use of information received and needed to establish SUFIs to be transmitted.

The input parameters to **o_SUFI_Handler** to be used as checking criteria are collected in tabular data structure **SUFI_Params** which is initialized at the beginning of each test case. These data are to allow the checking of the presence and the value of SUFIs. All entries are initialized to AnyOrOmit, and have to be set to well-defined values if these are to be used by **o_SUFI_Handler**. As a principle values specifically set are used as criteria for checking, values omitted are used as AnyOrOmit values. The resulting SUFI list is established by **o_SUFI_Handler** and can be retrieved in the data structure returned by the TSO. Details have to be defined in the TSO itself.

Tasks **o_SUFI_Handler** has to perform:

- Check mutual exclusiveness of SUFIs ACK and NOMORE.
- Check that one of SUFIs ACK or NOMORE is the last SUFI in the received SUFI string.
- Transfer the SUFIs received into the structure of SuperFields; this is the SUFI list structure existing today.
- If multiple occurrences of SUFI are found then use the **last** one to fill the SuperFields structure.
- Check for all parameters in SUFI_Params set to a specific expected value that one of the SUFIs using this value is present and that the value received matches the specific expected value.
- Check that if SUFIs are received for which an expected value of Any is specified, the SUFI is consistent if that SUFI is received.
- Check that if SUFIs are received for the presence of which no entry is specified in SUFI_Params, the SUFI is consistent.
- Check that sequence numbers are in the range between LB and UB if specific values are set.

Entries in **SUFI_Params**.

Element Name	Sigificance	Comment
UB	Upper bound of sequence number range	Highest SN for checking SNs acknowledged
LB	Lower bound of sequence number range	Lowest SN for checking SNs acknowledged
WSN_presence	Window Size SUFI present	To check the presence of the Window Size SUFI
MRW_presence	Move Receive Window SUFI present	To check the presence of the MRW SUFI
Nack1	SN of 1st PDU negatively acknowledged	For the NackList to check SN to be negatively acknowledged
Nack2	SN of 2nd PDU negatively acknowledged	For the NackList to check SN to be negatively acknowledged
Nack3	SN of 3rdPDU negatively acknowledged	For the NackList to check SN to be negatively acknowledged

More entries may be required in the future if specific SUFI field values are to be checked. The concept allows to add more fields easily. As these will be initialized with the AnyOrOmit value they should not require modifications to existing test cases, except constraints of the SUFI_Params type which may have been specified.

6.6 SMS test method and architecture

6.6.1 SMS CS test method and architecture

The test method used for SMS CS tests is the same as the NAS test method, see clause 6.3, and the same ASPs, see clause 7.1.2.

6.6.2 SMS PS test method and architecture

The test method used for SMS PS tests is the same as the NAS test method, see clause 6.3, and the same ASPs, see clause 7.1.2.

6.6.3 SMS Cell broadcasting test method and architecture

The test method used for SMS CB tests is the same as the BMC test method, see clause 6.8, and the same ASPs, see clause 7.1.2.

6.7 MAC test method and architecture

6.7.1 Testing architecture

Figure 10 illustrates a typical realisation of the MAC ATS.

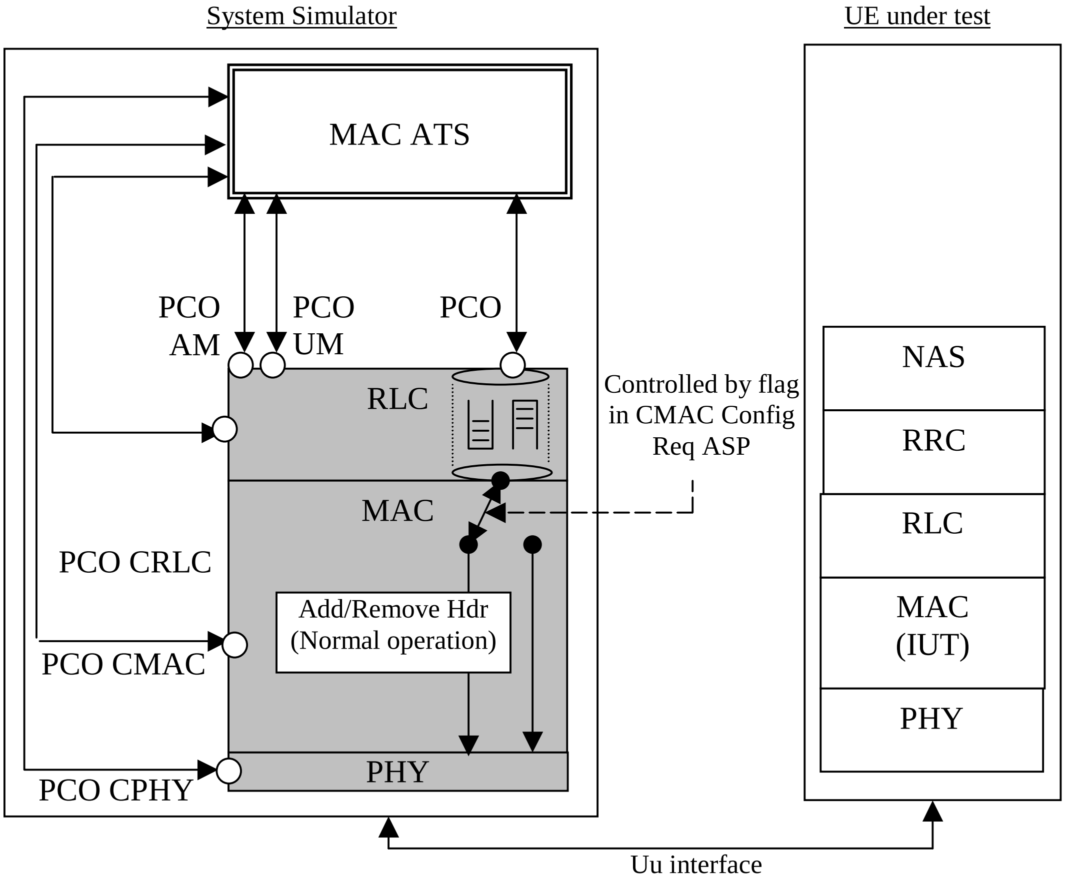

Figure 11: MAC ATS single party test method

6.7.2 Test method

The single party test method is used for MAC testing.

Separation of TTCN test cases from the configuration of the tester and initialisation of the UE is achieved by using test steps. For each MAC test case, common test steps will be used to perform the configuration of the tester and the appropriate generic setup procedures as described in 3GPP TS 34.108 [3]. These test steps will make use of PCOs AM, UM, TM, CRLC, CMAC, and CPHY.

Three PCOs are provided at the top of the RLC emulation in the tester, one corresponding to each of the available RLC modes: acknowledged, unacknowledged, and transparent. Routing information for different radio bearers used at these PCOs will be provided in ASP parameters.

The queues shown in the RLC emulation in figure 8 indicate that normal RLC transmit and receive buffering will be used to isolate the TTCN test suite from the real time issues involved if messages are sent directly to the MAC layer.

A flag is required within the CMAC Config Req to indicate that the SS MAC emulation must not add or remove any MAC header information, even if header fields should be present according to the configured channels. This flag shall allow control of the MAC header on a per logical channel basis. For example, it shall be possible to configure 4 DCCHs and a DTCH mapped to a DCH, such that the MAC will add / remove header information for the DCCHs, but not for the DTCH.

The MAC TTCN test cases make also use of the NAS TTCN test steps in order to bring UE to Idle state. The NAS test steps, which are called by the MAC test cases or steps, interface with the Dc PCO.

For MAC testing, the following RB Ids are used for the high priority NAS RB within the system simulator depending on the MAC configuration being simulated.

RB Id	Simulated configuration
-14	DCCH mapped to FACH
-15	DCCH mapped to DCH
-18	CCCH mapped to FACH

The SS decoder can use the RB Id to determine which MAC header fields are present, and create the appropriate abstract structure during the decode process. The SS decoder must understand enough of the MAC peer-to-peer protocol to determine which fields are present.

For example, the semantics of the UE Id Type field must be known to determine how many bits should be present in the UE Id field.

The MAC PDUs for MAC testing will always contain an AM RLC PDU (data or status) using 7 bit length indicators. See the RLC test method for further information on the SS decoder requirements for RLC PDUs.

6.7.2.1 Abnormal decoding situations

If the SS decoder cannot convert the received data into the supported structure, the SS shall terminate the test case immediately and indicate that a test case error has occurred.

6.8 BMC test method and architecture

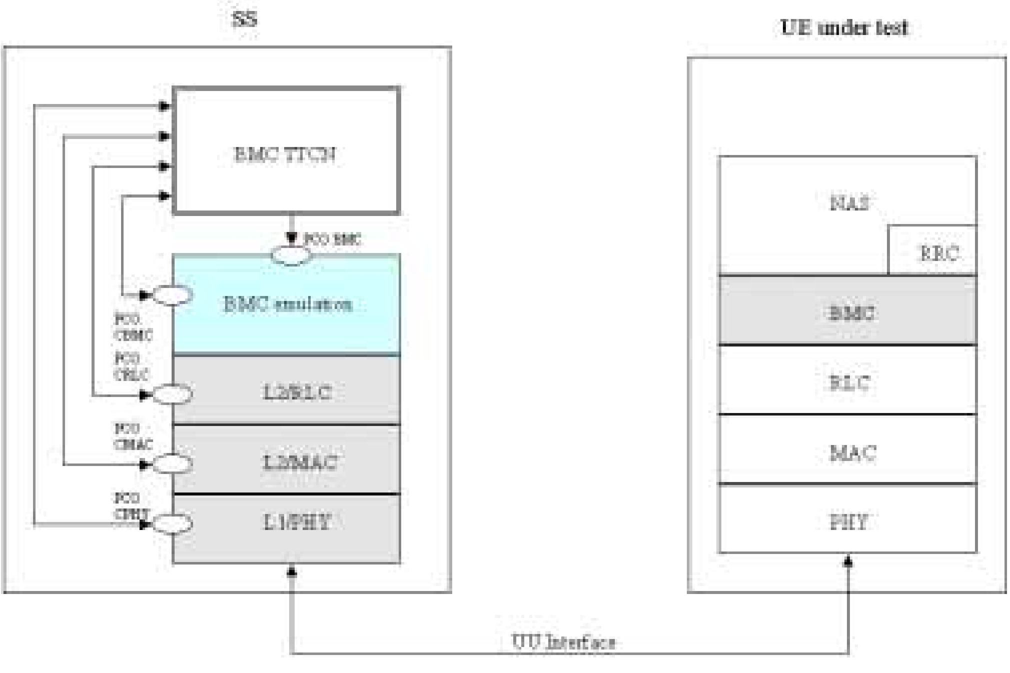

Figure 12: BMC testing architecture single party method

6.8.1 BMC test architecture

The single party test method is used for BMC testing, i.e. it does not exist an Upper Tester. BMC emulation is used as shown in figure 13. The BMC emulation makes use of two PCOs. The CBMC PCO is defined, to pass configuration information for a BMC entity. The BMC PCO is defined for BMC message data transfer.

Separation of TTCN test cases from the configuration of the tester and initialisation of the UE is achieved by using test steps. For BMC test cases, common test steps and newly defined test steps for BMC configuration will be used to perform the configuration of the tester and on UE side. These test steps make use of PCOs, CRLC, CMAC, and CPHY.

The UE shall be able to activate and deactivate a certain CB MessageID according CB data to be sent while testing.

BMC messages are sent in BMC message blocks on the CTCH. For sending BMC messages (BMC Scheduling Message (Level 2, DRX) and BMC CBS Message) a configuration in downlink direction shall be performed to map the CTCH (RB#30) onto the FACH - S-CCPCH.

6.8.2 BMC test method

For BMC testing, only PS Cell Broadcast Service as distributed BMC service is applied. CBS Messages and BMC Schedule Messages are only sent in downlink direction. No uplink is used for BMC testing. The BMC test data with necessary CBS information shall be given by PIXIT parameter with a description of the indication on the display.

This test method uses BMC primitives as defined in 3GPP TS 25.324 [20]. There are two level of BMC scheduling, Level 1 for CTCH configuration and Level 2 for DRX. The BMC scheduling information is conveyed to both BMC and MAC layer.

Level 1 scheduling is used configure the CTCH on the S-CCPCH. For BMC testing Release 99 (FDD), the Level 1 scheduling parameter M_{TTI} contains one radio frame in the TTI of the FACH used for CTCH. Therefore, only Level 1 scheduling information N (period of CTCH allocation on S-CCPCH) and K (CBS frame offset to synchronise to the SFN cycle (0 to 4 095 frames per cycle)) are necessary to configure the CTCH onto the S-CCPCH.

The Level 1 scheduling is done in the SS MAC layer, therefore this information is given by using the primitive "CMAC_BMCscheduling_REQ" to inform the MAC on SS side about K and N. The Level 1 scheduling information, K and N, is broadcast as system information in SIB 5 and SIB 6. After having performed the CTCH configuration as Level 1 scheduling, the SS is configured to send BMC messages and the UE has to listen to each CTCH for a BMC message.

Segmentation of BMC messages is performed by RLC in UM. A RLC segment shall contain BMC message payload as configured in RB#30 with a maximum number of 57 octets. The 57 octets payload is used to calculate the BMC inband scheduling Level 2 in the BMC TTCN (TSO).

If only one CB data as BMC CBS message is sent and repeated for a BMC test case, Level 1 scheduling is adequate, i.e. no BMC Scheduling Message (Level 2) is needed. Therefore, no level 2 scheduling information are included in the "CMAC_BMCscheduling_REQ" primitive. If more then one BMC CBS message are transmitted and repeated, BMC scheduling Level 2 message shall be performed.

Level 2 scheduling is used to predict the sent event of the next BMC message blocks and the BS index contents.

BMC scheduling Level 2 predicts exactly, which information is contained on a certain CTCH block set with an aligned Block Set index number and how many spare CTCH blocks are given as offset, before the next BMC message block will be sent. Figure 12 shows an example, how the message flow shall be done for BMC scheduling Level 2.

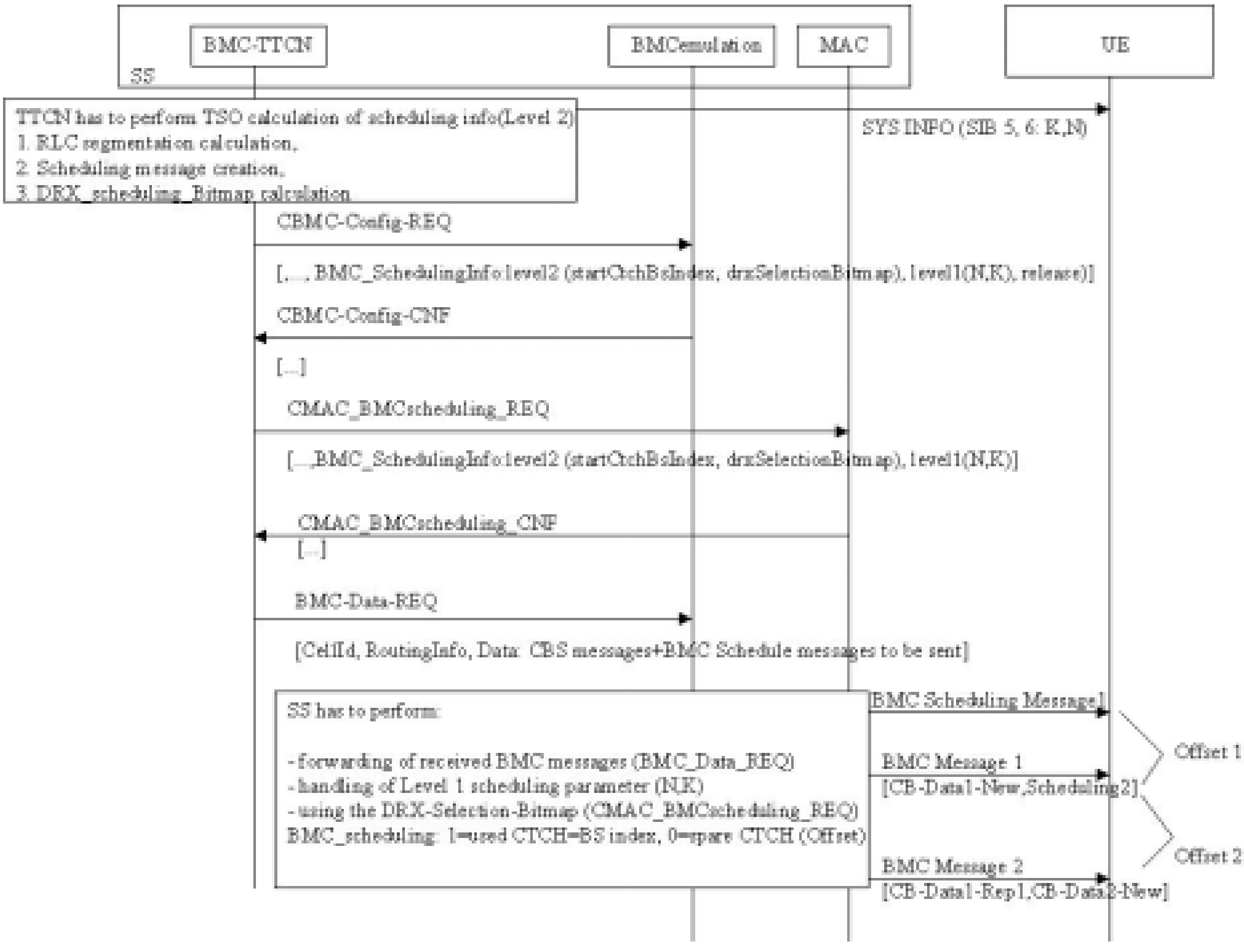

Figure 14: BMC Scheduling

The BMC test method makes use of the primitive: "BMC-Data-REQ" to transmit the BMC Messages to RLC. If BMC Scheduling Level 2 is used, an entire BMC message, including BMC CBS PDUs and a BMC Schedule PDU, to be transmitted is created by the BMC TTCN and forwarded to the BMC emulation. The transmission of BMC PDU is confirmed through the primitive BMC-Data-CNF. The segmentation of the BMC PDU is done at the RLC layer.

According to the K and N value, the MAC layer at SS side determines the CTCH blocks for the BMC use. The CTCH blocks are indexed (i = 1 ... 256). If BMC DRX is needed, the BMC scheduling Level 2 information figures out the occupancy / spare of the available CTCH blocks by using a DRX_Selection_Bitmap. In the bitmap each bit, set to '1', corresponds to an actually available CTCH block belonging to the DRX period for the SS transmission. The all occupied consecutive CTCH blocks constitutes a BMC DRX period, whilst the consecutive spared blocks indicate the DRX offset as spare CTCH slot.

Following the DRX_Selection_Bitmap, the segmented BMC messages are transmitted. Each "BMC-Data-REQ" primitive has its own aligned "CMAC_BMCscheduling _REQ" primitive, where all BMC scheduling information is predicted. An initial CTCH block index is given (startCtchBsIndex) as a start index offset.

An octet string is defined whereas each bit describes one assigned CTCH block, i.e. one BS index on the S-CCPCH.

Bitmap value:

1 (binary) = indicates a used/occupied BS index (CTCH frame, with a payload size of 57 octets) to send BMC message segments for a message block.

0 (binary) = indicates a spare BS index, i.e. unused CTCH frame, to give an UE supporting DRX the necessary information.

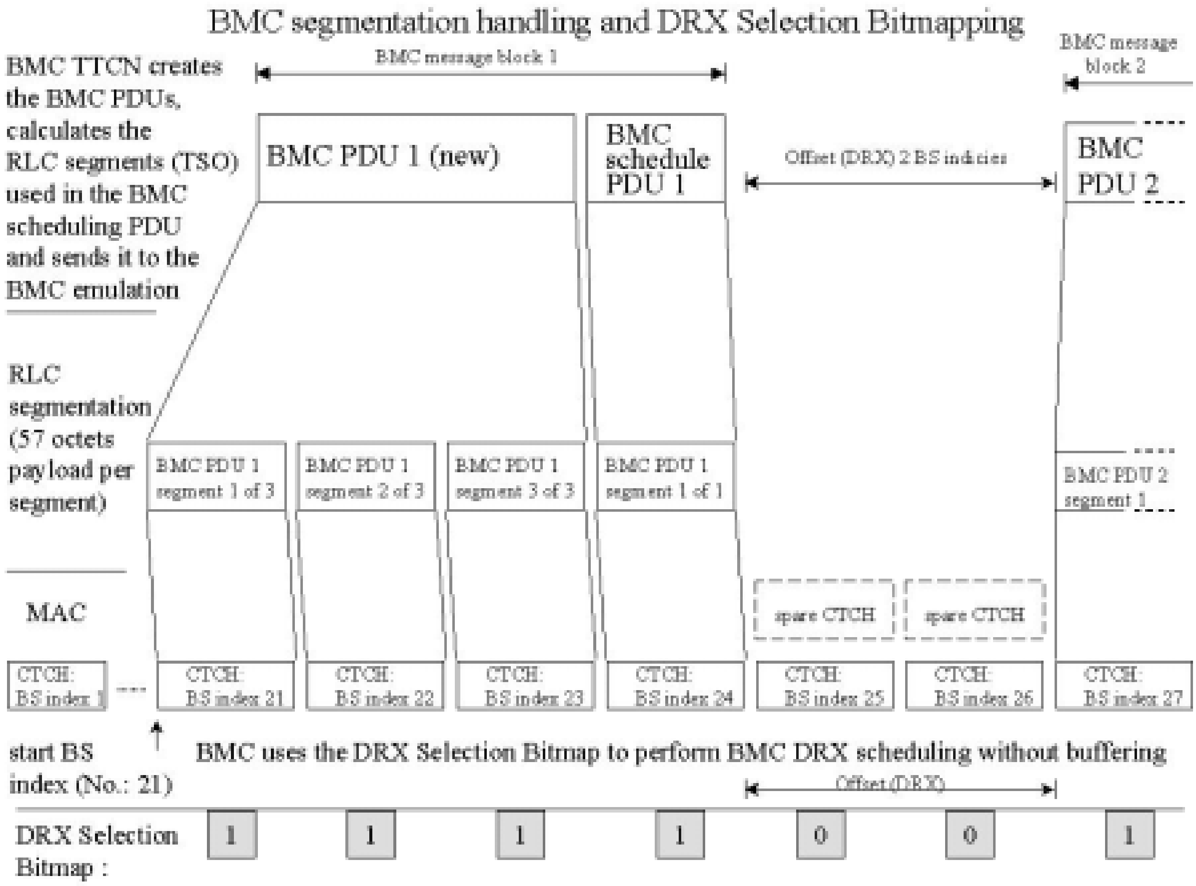

Figure 15: BMC DRX scheduling: segmentation handling

6.9 PDCP test

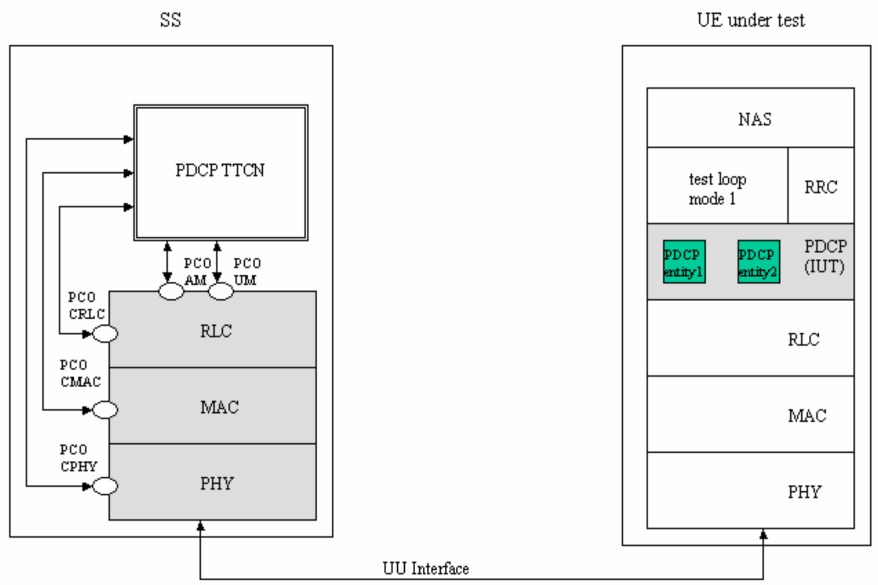

Figure 16: PDCP testing architecture 1: single party test method, with test loop mode 1

6.9.1 PDCP test architecture

The single party test method is used for PDCP testing. All PDCP tests that require uplink data will make use of the UE test loop mode 1 defined in 3GPP TS 34.109 [4]. Test Loop mode 1 is only available in the user plane, so all PDCP tests will be performed in the user plane, using the same logical channels mapped to transport channels as defined in RLC test cases, except for test case, clause 7.3.2.2.4, where a configuration of combined radio bearers used only for this test case is defined.

Separation of TTCN test cases from the configuration of the tester and initialisation of the UE is achieved by using test steps. For PDCP test cases, common test steps and newly defined test steps for PDCP configuration will be used to perform the configuration of the tester and the appropriate generic setup procedures as described in 3GPP TS 34.108 [3] and in clause 7.4 of 3GPP TS 34.123-1 [1]. These test steps will make use of PCOs RLC AM, RLC UM, CRLC, CMAC, and CPHY.

The PDCP TTCN test cases make also use of the NAS TTCN test steps in order to setup a PS session.

For PDCP testing, the IP Header Compression protocol as described in RFC 2507 [30] is used as optimisation method. The IP header compression and decompression mechanisms as described in RFC 2507 [30] is not part of PDCP TTCN. PDCP testing make use of uncompressed, compressed and decompressed TCP/IP header packets of a certain packet stream and uncompressed, compressed and decompressed UDP/IP header packets of a certain generation. This parameters are given as test parameter (PIXIT information).

PDCP testing includes transmission/reception of compressed/decompressed IP header packets, PDCP sequence numbering while lossless SRNS relocation and PID assignment rules as well as PDCP configuration tests as described in 3GPP TS 25.323 [19], Release 99. It does not test optimisation specific protocol behaviour as error recovery and packet reordering as described in RFC 2507 [30].

6.9.2 PDCP test method

For PDCP testing, the RB test mode is used with test loop mode 1. After establishing a PS session with RB in RLC UM or/and AM, the UE is configured to support a negotiated PDCP configuration. UDP/IP header packets are used as Non-TCP/IP header packets as PDCP test data.

There are different input parameter as PIXIT values necessary for PDCP testing.

For TCP/IP header packets, uncompressed TCP/IP header packets shall be defined as PIXIT input parameter. In addition, there are the corresponding RFC 2507 [30] FULL_HEADER packet, COMPRESSED_TCP packet and COMPRESSED_TCP_NONDELTA packet given for each TCP/IP header packet as PIXIT information.

For UDP/IP header packets, uncompressed UDP/IP header packets shall be defined as PIXIT input parameter. In addition, there are the corresponding RFC 2507 [30] FULL_HEADER packet and COMPRESSED_NON_TCP packet given for each UDP/IP header packet as PIXIT information.

To check the use of certain PID values assigned to IP compressed header types, a given IP header packet (PIXIT) will be sent to the UE. The UE shall return a appropriate valid IP header packet type, which corresponds to the previous sent IP header packet. The usage of valid compressed/uncompressed IP header packets shall be checked by comparing the given PIXIT IP header packet types for each IP header packet previously sent.

The IP header packet order as described in RFC 2507 [30] shall be applied within a test case.

If for example an TCP/IP header packet of type "COMPRESSED_TCP" shall be sent, the TTCN uses the given TCP/IP header packet (PIXIT) for transmission to the UE. The UE shall decompress the received packets appropriate, afterwards it will be returned by the loop back entity and it shall be sent by applying IP header compression rules as described in RFC 2507 [30] and as configured. Then, the SS receives returned IP header packets and compares it with all valid IP header packets given as PIXIT parameter corresponding to the previously sent IP header packet. It is checked, whether or not the IP header packet with assigned PID is valid and a configured PDCP PDU where used for transmission. In this way, it is checked, that the UE performs IP header compression as configured and is able to assign the correct PID values.

6.10 Multi-RAT Handover Test Model

6.10.1 Overview

The test model is shown in figure 17. The SS in the model consists of UTRAN emulation part and GERAN emulation part, GERAN emulation part includes protocol emulation modules for GSM CS services and protocol emulation modules for GPRS service. Protocol stack L1 (GERAN), L2 is for GSM CS service function emulation, protocol stack L1, RLC/MAC, LLC, SNDCP is for GPRS service function emulation. SNDCP emulation model and relevant PCO's can be removed if "traffic channel gets through" is not tested.

L1 (GERAN) provides necessary physical layer functionality for both GSM and GPRS. A control PCO and a set of ASP's are defined for configuring and controlling its protocol behaviour required in the test cases. L1 (GERAN) provides services to L2 and RLC/MAC emulation modules, the interfaces between them are not specified in this test model, it is implementation dependent and shall follow the relevant GSM and GPRS specifications.

L2 emulates necessary GSM L2 protocol functionality used in testing. A data PCO and a set of ASP's are defined for this module and used for transmitting and receiving layer 3 signalling messages and use data. The definition of the PCO and these ASP's are based on the logical channel concept of GSM specification. A control PCO and related ASP's are also defined for L2, they are used to introduce abnormal layer 2 behaviour required by the test purposes.

RLC/MAC is emulation module for GPRS Radio Link Control/Medium Access Control protocol. Two PCO's and related ASP's are defined for the module. Control PCO is used to set TBF and assign physical resources to it, actual physical resources (packet channels) are created by L1 (GERAN) ASP's beforehand. Data PCO is for transmitting and receiving RLC control messages (RLC control block). Before any RLC data or control block, except RLC control block on PCCCH or PRACH, or PBCCH, is sent (or received) a proper TBF shall be configured. In addition RLC/MAC module provides service to LLC emulation module, the interface between them is determined by implementation and shall be compliant with relevant core specification.

LLC performs GPRS Logical Link Control protocol emulation. Its data PCO and ASP's are used for exchange GMM signalling messages between TTCN and the UE under test. The current defined ASP's on control PCO are subset of the primitives defined in core specification, they are used to assign, un-assign TLLI and ciphering parameters, or get status report.

6.10.2 ASP function description

6.10.2.1 Identities

- Within the SS, a cell is identified by cell identifier (cellId), which is of TTCN type CellId (INTEGER).
- Within a cell, a basic physical channel is identified by physical channel identifier (physicalChId), which is of TTCN type PhysicalChId (INTEGER).
- Within A a physical channel, logical channel is identified by logical channel type (g_LogicChType), which is of TTCN type G_LogicChType (INTEGER). When multiple logical channels of same type are carried by (mapped to) the same basic physical channel, they are differentiated by sub-channel number (subChannel), which is of TTCN type SubChannelNumber (INTEGER).
- At the top boundary of L2 emulation module two service access points (SAP) are available, they are identified by SAPI. SAPI=3 is used for short message service; SAPI=0 is used for L3 signalling messages and user data.

EXAMPLE: If G_L2_DATA_REQ ASP has the following parameter setting:

- cellId = tsc_CellA;
- sAPI = tsc_SAPI_0;
- physicalChId = tsc_PhyCh0;
- g_LogicChType = tsc_SDCCH4; and
- sunChannel = tsc_SubChannel1;

it sends PDU on the SDCCH4(1) logical channel which is carried by the physical channel tsc_PhyCh0 in cell A.

6.10.2.2 Cell configuration and control

In GSM each base station has a base station identity code BSIC, it consists of network colour code and base station colour code (NCC + BCC). BSIC is continuously broadcasted on the SCH channel, and it shall be used as the training sequence code for broadcast and common control channels.

In the test model the function of G_CL1_CreateCell_REQ ASP is to create a cell and pass parameter BSIC to it. This ASP establishes the cell identifier which shall be used in the ASP's related to this cell.

This is the first step to configure L1 (GERAN) emulation module of the SS.

6.10.2.3 L1 (GERAN) configuration and control

Configuration and control functions identified for L1 (GERAN) of a cell are:

- creation of basic physical channels;
- creation of multislot configuration;
- release of basic physical channel;
- modifications of channel mode, ciphering parameters and transmission power level;
- reporting of L1 header of SACCH channel;
- pickup a frame in near future, which can carry L3 message.

6.10.2.3.1 Basic physical channel configuration

A basic physical channel uses a combination of frequency and time domain resources, therefore, the definition of a particular basic physical channel consists of a description in the frequency domain and a description in the time domain. In time domain the resource is called Time Slot, there are 8 time slots in one frame, numbered from 0 to 7. In frequency domain a basic physical channel may use only one frequency or may use multiple frequencies in frequency hopping.

Basic physical channel carrying FCCH + SCH + BCCH + CCCH (PCH, AGCH, RACH) or FCCH + SCH + BCCH + CCCH + SDCCH4 logical channels shall be located in time slot 0, and uses single frequency (non-hopping). The basic physical channel carrying additional BCCH, CCCH (PCH, AGCH, RACH) logical channels shall be located in time slot 2, 4, 6 and uses the same single frequency as the frequency used by the physical channel carrying FCCH, SCH.

GSM specification defines 24 permitted combinations of different logical channels, which can be mapped on to a basic physical channel. The combination defines which logical channels are carried by a basic physical channel, and it is also an indication of which modulation (GMSK or 8PSK) is used for the basic physical channel.

Training sequence code (TSC) is another parameter needed by physical channel. Common control and broadcast channel have to use BCC as its TSC.

Dedicated control channel and dedicated traffic channel need more parameters to configure. Parameter "Channel Mode" is needed to specify channel coding (therefore the user data rate). Ciphering related parameters are required to define the ciphering behaviour of the channel.

Common control channels need parameters to configure where in the 51-multiframe paging and access grant blocks are located.

Transmission power level is provided as per physical channel parameter, power level of each physical channel can be controlled independently.

The function of ASP G_CL1_CreateBasicPhyCh_REQ is to create a basic physical channel which has the required property defined by all the parameters mentioned above.

In the process of L1 (GERAN) configuration, calling the ASP is the next step after calling G_CL1_CreateCell_REQ.

6.10.2.3.2 Multislot configuration for circuit switched channels

Multislot configuration for circuit switched connection consists of multiple circuit switched traffic channels, in L1 point of view these traffic channels are independent basic physical channels with the same frequency parameters (ARFCN or MA, MAIO, HSN) and the same training sequence code but located in different time slots, one of the basic physical channels is the main channel of the configuration carrying the main signalling (FACCH, SACCH, IACCH) for the configuration. The main channel shall be bi-directional channel and with channelCombanition TCH/F+FACCH/F+SACCH/M or E-TCH/F+E-IACCH/F+E-FACCH/F+E-SACCH/M. When transmitting user data (not signalling message) stream is divided into substreams, each substream is transmitted independently on a channel in the configuration. At the receiving side all substreams are combined back to user stream.

In the test model all traffic channels in a multislot configuration are created separately with G_L1_CreatedBasicPhyCh_REQ, then ASP G_L1_CreateMultiSlotConfig_REQ is called to indicate to the L1 emulation model which channel is the main channel, and which channels are the members of the multislot configuration and their substreams shall be combined together to form the user data stream.

6.10.2.3.3 Frame in the near future

ASP G_CL1_ComingFN_REQ is defined to request L1 (GERAN) return the reduced frame number (FN modulo 42432) which is far enough in the future from current frame number and is able to carry L3 message on the specified channel. "far enough" means that there is enough time left for TTCN to prepare a L3 message to be sent on that frame.

6.10.2.3.4 L1 header

The layer 1 header of SACCH from UE to network carries information of timing advance and UE uplink transmission power level, verifying L1 header contents is required in some test cases, ASP G_CL1_L1Header_REQ and G_CL1_L1Header_CNF are defined for fulfilling this requirement.

6.10.2.4 L2 configuration and control

For normal operation there is no parameter configurable in L2. Some abnormal L2 behaviours are required in test cases. In the test model two ASP's are currently defined to introduce abnormal L2 behaviour.

6.10.2.4.1 Don't response to some handover access bursts

In non-synchronized handover procedure UE/MS, having received handover command, sends handover access bursts on the target channel repeatedly till it receives PHYSICAL INFORMATION message from network or T3124 times out. Normally network replies PHYSICAL INFORMATION as soon as it receives handover access burst. Some test cases require that the SS ignores several incoming handover access bursts then responses to the one that follows. ASP G_CL2_HoldPhyInfo_REQ is defined for fulfilling this requirement. It is used together with and before a data ASP sending PHYSICAL INFORMATION message. When SS receives the G_CL2_HoldPhyInfo_REQ, it does not transmit the PHYSICAL INFORMATION message until n handover access bursts have been received.

6.10.2.4.1 No UA reply to SABM

GSM L2 protocol is adapted from LAPD (HDLC subset). The multiframe operation mode is established through exchange of supervisory frame SABM and unnumbered frame UA between peer entities, and SABM is always sent by UE/MS, UA is always sent by network. UE/MS will repeatedly transmit SABM till it receives UA or retransmission counter is reached. Some handover test cases require that the SS does not response to the incoming SABM, so handover fails. G_CL2_NoUAforSABM_REQ is used for such purpose, it commands the SS not to send UA response to the UE when SABM is received.

6.10.2.5 System Information sending

There are 17 different SYSTEM INFORMATION messages on BCCH and 4 different SYSTEM INFORMATION messages on SACCH defined for circuit switched services in GSM specification. In a particular test case not all of them are required. SYSTEM INFORMATION messages on BCCH shall be broadcasted periodically by the SS, SYSTEM INFORMATION TYPE 5, 6 and optionally 5bis and 5ter messages shall be sent on SACCH by the SS when nothing else has to be sent on that channel.

G_L2_SYSINFO_REQ is defined to deliver a SYSTEM INFORMATION message and its type SysInfoType to the SS, SS shall store the SYSTEM INFORMATION and transmit it periodically according to the scheduling rules specified in 3GPP TS 45.002 [31] clause 6.3.1.3. SYSTEM INFORMATION message newly delivered shall override the same type SYSTEM IFORMATION message previously stored in the SS.

SYSTEM INFORMATION message type 18, 19, 20 are scheduled by scheduling information in SYSTEM INFORMATION type 9. ASP for scheduling these messages has not been defined yet because these messages are not required in current test cases.

6.10.2.6 Paging

Paging message for a particular UE/MS shall be sent on the right CCCH_GROUP (or PCCCH_GROUP) and PAGING_GROUP which are determined by IMSI of the UE/MS and other parameters. In the test model TTCN code is responsible to calculate the value of CCCH_GROUP (or PCCCH_GROUP) and the value of PAGING_GROUP.

TTCN selects the right channel according to the value of CCCH_GROUP (or PCCCH_GROUP), then PAGING REQUEST message and the value of PAGING_GROUP are passed to the SS by using ASP G_L2_Paging_REQ.

The SS shall determine the position where the paging block is located using the value PAGING_GROUP and other CCCH (or PCCCH) parameters configured by G_CL1_CreateBasicPhyCH_REQ, then send the PAGING REQUEST message according the parameter pagingMode in the ASP:

- send the message on the paging block determined by PAGING_GROUP if pagingMode = "normal paging";
- send the message on the paging block determined by PAGING_GROUP and the "next but one" position on the PCH or in the third block period on PCCCH where paging may occur (PPCH) if pagingMode = "extended paging";
- send the message on all paging blocks if pagingMode ="paging reorganization".

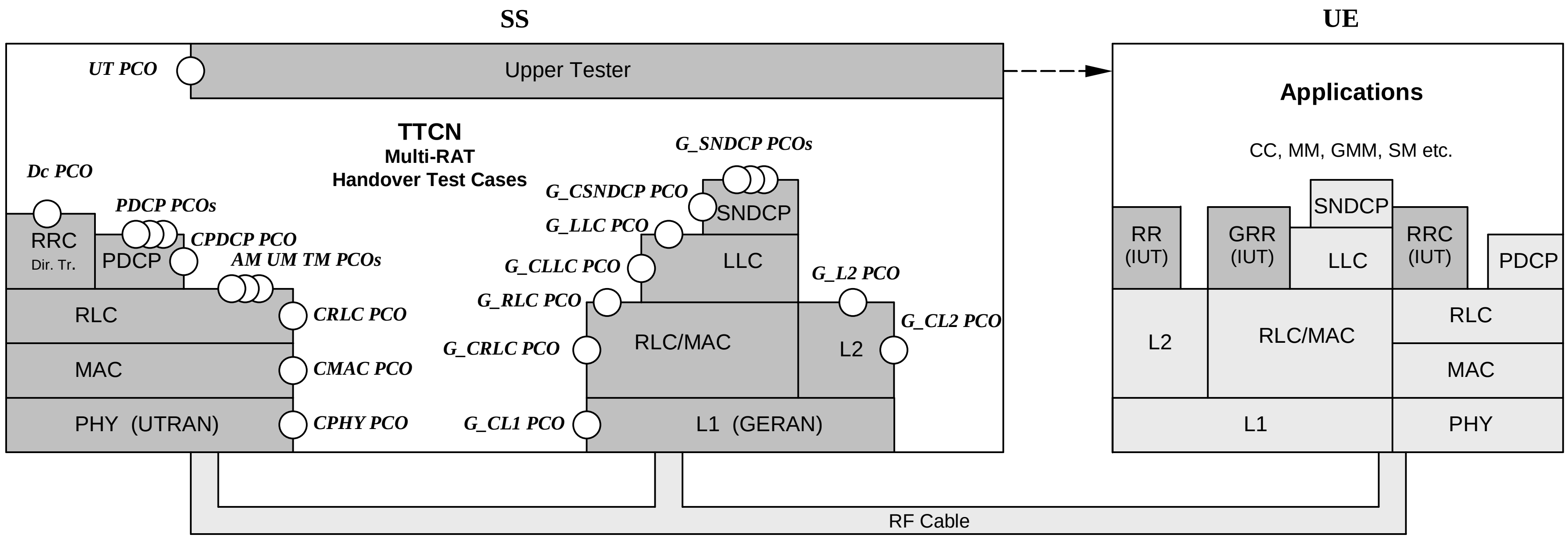

Figure 17: The model of multi-RAT handover testing

7 PCO and ASP definitions

7.1 NAS PCO and ASP definitions

7.1.1 NAS PCO Definitions

Table 3: Dc PCO Type Declarations

PCO Type Declarations	
PCO Type	Dc_SAP
Role	LT
Comments	The PCO type for NAS testing

Table 4: Dc PCO Declarations

PCO Declarations	
PCO Name	Dc
PCO Type	Dc_SAP
Role	LT
Comments	Carry transmission and reception of NAS messages

7.1.2 Primitives used at Dc PCO

The Dc PCO is used to transmit and receive NAS (MM, CC, SM, SS) messages. Two categories of primitives are operated at the Dc PCO:

- RRC_DataReq for transmission of a NAS PDU;
- RRC_DataInd for reception of a NAS PDU.

These primitives are declared in TTCN tabular form, see table 5.

Table 5: Primitives used at the Dc PCO

Primitive	Parameters	Use
RRC_DataInd	Cell identity INTEGER (-31..32) LogicChGSM SapId CN domain id START NAS message	The ASP is used to indicate the receipt of a NAS message using acknowledged operation
RRC_DataReq	Cell identity INTEGER (-31..32) LogicChGSM SapId CN domain id NAS message	The ASP is used to request the transmission of a NAS message using acknowledged operation.

The RB Identity and CN domain parameters defined in the primitives are mandatory for UTRAN and not applicable for GERAN.

The START parameter is mandatory in INITIAL DIRECT TRANSFER; each time when it is received the new START shall be downloaded to the SS to reinitialise counters-C and counters-I.

The LogicChGSM and SapId parameters are mandatory for GERAN and not applicable for UTRAN. They are defined because they may be used for future TTCN test cases.

Except the initial, uplink and downlink direct transfer procedures, the NAS TTCN specification uses the TTCN test steps to realise all RRC functions for testing. The single layer test concept is kept for the NAS tests.

A simple RRC emulation shall be maintained for the NAS tests. It has four functions:

- Emulate the three direct transfer procedures;
- Convert the NAS downlink messages defined in 3GPP TS 24.008 [9] in table format to the NAS message in ASN.1 octet string specified in 3GPP TS 25.331 [21]. Convert the NAS uplink message in the reverse way;
- PER encoding and decoding;
- Have the integrity protection.

RB3 and RB4 are specifically used for the NAS signalling. When an uplink message entered the receiving buffer at AM-SAP from the RLC emulation, either an RRC test step if running will take it out; or the RRC emulation if running will pick the received message from the buffer. Activation of any RRC test steps and activation of any NAS test steps at the same time shall be excluded in TTCN (no concurrency between them).

7.2 Ut PCO and ASP definitions

7.2.1 Ut PCO Declarations

The Ut PCO is served as the interface to the UE EMMI for remote control of operations, which have to be performed during execution of a test case such as to switch the UE on/off, initiate a call, etc.

Table 6: Declaration of the uppertester PCO type

PCO Type Declarations	
PCO Type	MMI
Role	UT
Comments	The PCO type for MMI or EMMI of the upper tester

Table 7: Declaration of the Ut PCO

PCO Declarations	
PCO Name	Ut
PCO Type	MMI
Role	UT
Comments	Carry transmission commands and reception of results for the upper tester

7.2.2 Primitives used at Ut PCO

The Ut PCO is used to indicate to the upper tester actions and to receive the acknowledgement of these actions. The AT commands are used wherever the suitable commands exist within 3GPP TS 27.007 [23], 3GPP TS 27.005 [22] and 3GPP TS 27 060 [24]. An MMI command is used, when AT commands does not exit for the action to performed. The primitives used at the Ut PCO, are declared in TTCN tabular form, see the table 8.

Table 8: Primitives used at the Ut PCO

Primitive	Parameters	Use
AT_CmdReq	Command: IA5String SMS_BlockMode: HEXSTRING	Request an AT command to the upper tester.
AT_CmdInd	Command: IA5String SMS_BlockMode: HEXSTRING	Indication of a result from the upper tester.
AT_CmdCnf	Result: BOOLEAN ResultString: IA5String SMS_BlockMode: HEXSTRING	Return a positive or negative result from the command previously sent. Both the boolean result and String parameter are optional.
MMI_CmdReq	Command: IA5String	Request a command to the upper tester.
MMI_CmdCnf	Result: BOOLEAN ResultString: IA5String	Return a positive or negative result from the command previously sent. The String parameter is optional.

The AT_CmdReq primitive for sending AT commands is mostly used to trigger electronically an uplink access, such as initiating of a call, attaching or detaching, starting packet data transfer etc. The MMI_ primitive is defined mainly for observation of some test events via a test operator, such as checking DTMF tone or checking called party number, etc.

The AT_CmdInd primitive for receiving AT commands is mostly used to transfer unsolicited result codes from the UE to the lower tester.

The SMS_BlockMode parameter is used to control and observe the Block mode procedure for SMS. This parameter is not yet used; it is defined for future development. The Command and SMS_BlockMode parameters are mutually exclusive

For the Command in the AT_CmdReq and AT_CmdInd primitives, the verbose format is used as defined in 3GPP TS 27.007 [23]. For the Command in MMI_CmdReq, just a descriptive IA5 string line, like "Check DTMF tone" is used.

7.3 RRC PCO and ASP definitions

7.3.1 AM/UM/TM PCO and ASP definitions

7.3.1.1 SAP and PCO for data transmission and reception

Table 9: Declaration of the RRC PCO Type

PCO Type Definition	
PCO Type	DSAP
Role	LT
Comment	DATA transmission and reception

Table 10: PCO TM declaration

PCO Type Definition	
PCO Name	TM
PCO Type	DSAP
Role	LT
Comment	Carry Transparent Mode RLC PDU

Table 11: PCO AM declaration

PCO Type Definition	
PCO Name	AM
PCO Type	DSAP
Role	LT
Comment	Carry Acknowledged Mode RLC PDU

Table 12: PCO UM declaration

PCO Type Definition	
PCO Name	UM
PCO Type	DSAP
Role	LT
Comment	Carry Unacknowledged Mode RLC PDU

Table 13: PCO BMC declaration

PCO Type Definition	
PCO Name	BMC
PCO Type	DSAP
Role	LT
Comment	Provide Unacknowledged Mode BMC data transmission service

7.3.2 Control PCO and ASP

7.3.2.1 SAP and PCO for control primitives transmission and reception

Table 14: SAP declaration

PCO Type Definition	
PCO Type	CSAP
Role	LT
Comment	Control primitives transmission and reception

Table 15: PCO CPHY

PCO Definition	
PCO Name	CPHY
PCO Type	CSAP
Role	LT
Comment	Control Physical Layer

Table 16: PCO CRLC

PCO Type Definition	
PCO Name	CRLC
PCO Type	CSAP
Role	LT
Comment	Control RLC Layer

Table 17: PCO CMAC

PCO Type Definition	
PCO Name	CMAC
PCO Type	CSAP
Role	LT
Comment	Control MAC Layer

Table 18: PCO CBMC

PCO Type Definition	
PCO Name	CBMC
PCO Type	CSAP
Role	LT
Comment	Control BMC Layer

7.3.2.2 Control ASP Type Definition

7.3.2.2.1 CPHY_AICH_AckModeSet

ASN.1 ASP Type Definition	
Type Name	CPHY_AICH_AckModeSet_REQ
PCO Type	CSAP
Comment	To request for setting of AICH Acknowledge Mode
Type Definition	

```
SEQUENCE    {
            cellId              INTEGER(0..63),
            routingInfo         RoutingInfo,
            ratType             RatType,
            aICH_Mode           AICH_Mode
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_AICH_AckModeSet_CNF
PCO Type	CSAP
Comment	To confirm setting of AICH Acknowledge Mode
Type Definition	

```
SEQUENCE    {
                cellId                  INTEGER(0..63),
                routingInfo             RoutingInfo
}
```

ASN.1 Type Definition	
Type Name	AICH_Mode
Comment	Normal operation: The AICH will operate as normal, and will acknowledge or negatively acknowledge on all UE RACH transmission attempts, appropriately. No Acknowledge: The AICH shall not transmit acknowledge or Negative Acknowledge on all UE RACH transmission attempts. Negative Acknowledge: The AICH shall transmit Negative Acknowledge on all UE RACH transmission attempts
Type Definition	

```
ENUMERATED  {
            Normal  (0),
            noAck   (1),
            negACK  (2)
}
```

7.3.2.2.2 CPHY_Cell_Config

ASN.1 ASP Type Definition	
Type Name	CPHY_Cell_Config_CNF
PCO Type	CSAP
Comment	To confirm to setup the cell parameter
Type Definition	

```
SEQUENCE    {
            cellId                          INTEGER(0..63)
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_Cell_Config_REQ
PCO Type	CSAP
Comment	To request to setup the cell parameter. The unit of tcell is chip; the unit of sfnOffset is frame number; the primary scambling code number of the cell is 16*primaryScramblingCode_SS; the unit of dLTxAttenuationLevel is dB.
Type Definition	

```
SEQUENCE    {
            cellId                          INTEGER(0..63),
            tcell                           INTEGER(0..38399),
            sfnOffset                       INTEGER(0..4095),
            frequencyInfo                   FrequencyInfo,
            primaryScramblingCode_SS        INTEGER(0..511),
            cellTxPowerLevel                CellTxPowerLevel,
            dLTxAttenuationLevel            INTEGER(0..30)
}
```

ASN.1 Type Definition	
Type Name	CellTxPowerLevel
Comment	The defaultCellTxPowerLvl is a default setting and is used for the most signalling tests. The real total cell DL Tx power level equals to the sum of the DL Tx power of the individual physical channels configured. The totalCellTxPowerLvl applies to e.g. the idle mode tests in a non-default multi-cell radio environment.
Type Definition	

```
CHOICE {
            defaultCellTxPowerLvl       NULL,
            totalCellTxPowerLvl         DL_TxPower
}
```

7.3.2.2.3 CPHY_Cell_Config

ASN.1 ASP Type Definition	
Type Name	CPHY_Cell_Release_CNF
PCO Type	CSAP
Comment	The confirmation to the CPHY_Cell_Release_Req
Type Definition	

```
SEQUENCE {
            soft_Reset          BOOLEAN,
            cell_ID_List        SEQUENCE (SIZE (1..8)) OF INTEGER(0..63)    -- cell IDs
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_Cell_Release_REQ
PCO Type	CSAP
Comment	1. This Primitive with "Soft_Reset" flag ON gives a common known starting point/state of SS for a test case. The SS performs the following whenever it receives this primitive with "Soft_Reset" flag ON:Releases all configured Channels and cells (if any) irrespective of Cell ID list IE. 2. Releases the associated Memory Buffers (if any). 3. Cancels all active timers (if any) With "Soft_Reset" flag OFF: 1. Releases cells listed in IE Cell_ID_List and associated configured Channels (if any) 2. Releases the Memory Buffers(if any) associated with Cells listed in IE Cell_ID_List 3. Cancels all active timers (if any) associated with Cells listed in IE Cell_ID_List.
Type Definition	

```
SEQUENCE {
            soft_Reset          BOOLEAN,
            cell_ID_List        SEQUENCE (SIZE (1..8)) OF INTEGER(0..63)    -- cell IDs
}
```

7.3.2.2.4 CPHY_Ini

ASN.1 ASP Type Definition	
Type Name	CPHY_Ini_REQ
PCO Type	CSAP
Comment	Request to initialise the test
Type Definition	

```
ENUMERATED {
            defaultRadioEnvironment(0),
            nonDefaultMultiCell(1)
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_Ini_CNF
PCO Type	CSAP
Comment	Confirm the test initialisation
Type Definition	

```
SEQUENCE    {
            confirmation        NULL
}
```

7.3.2.2.5 CPHY_Cell_TxPower_Modify

ASN.1 ASP Type Definition	
Type Name	CPHY_Cell_TxPower_Modify_CNF
PCO Type	CSAP
Comment	To confirm to change the DL power
Type Definition	

```
SEQUENCE    {
            cellId              INTEGER(0..63)
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_Cell_TxPower_Modify_REQ
PCO Type	CSAP
Comment	To request to change the DL power
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            dLTxAttenuationLevel        INTEGER(0..30)
}
```

7.3.2.2.6 CPHY_Frame_Number

ASN.1 ASP Type Definition	
Type Name	CPHY_Frame_Number_CNF
PCO Type	CSAP
Comment	To return the requested connection frame number. The routingInfo indicates a physical channel.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo,
            frameNumber                 INTEGER (0..255)
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_Frame_Number_REQ
PCO Type	CSAP
Comment	To request the physical layer to return a connection frame number on which the next message can be sent at the specified PCO on the specified logical channel. The return frame number shall leave time from current frame number in order to leave some execution time for TTCN preparing next message. The routingInfo indicates a physical channel
Type Definition	

```
SEQUENCE    {
        cellId                  INTEGER(0..63),
        routingInfo             RoutingInfo
}
```

7.3.2.2.7 CPHY_Out_of_Sync

ASN.1 ASP Type Definition	
Type Name	CPHY_Out_of_Sync_IND
PCO Type	CSAP
Comment	To report that the physical channel synchronization (in FDD mode, sync with uplink DPCCH) was lost as detected by the SS receiver.
Type Definition	

```
SEQUENCE    {
        cellId                      INTEGER(0..63),
        routingInfo                 RoutingInfo
}
```

7.3.2.2.8 CPHY_PRACH_Measurement

ASN.1 ASP Type Definition	
Type Name	CPHY_PRACH_Measurement_CNF
PCO Type	CSAP
Comment	To Confirm PRACH Measurement Req
Type Definition	

```
SEQUENCE    {
        cellId              INTEGER(0..63),
        routingInfo         RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_PRACH_Measurement_REQ
PCO Type	CSAP
Comment	To request for Start or Stop of PRACH Measurements to be done every PRACH PREAMBLE or MESSAGE received.
Type Definition	

```
SEQUENCE    {
        cellId                  INTEGER(0..63),
        routingInfo             RoutingInfo,
        ratType                 RatType,
        pRACH_MeasurementInd    PRACH_MeasurementInd
}
```

ASN.1 Type Definition	
Type Name	PRACH_MeasurementInd
Comment	1. Start : The SS shall start the sending PRACH parameters Measurement report on CPHY PCO, for each PRACH Preamble or MESSAGE received from the UE by primitive CPHY_PRACH_Measurement_Report_IND on CPHY PCO. 2. Stop : The SS shall stop sending of PRACH parameters Measurement report on CPHY PCO, for each PRACH Preamble or MESSAGE received from the UE by primitive CPHY_PRACH_Measurement_Report_IND on CPHY PCO.
Type Definition	

```
ENUMERATED  {
            start (0),
            stop  (1)
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_PRACH_Measurement_Report_IND
PCO Type	CSAP
Comment	SS indicates a PRACH parameters measurement report for each PRACH Preambles or MESSAGE received from the UE
Type Definition	

```
SEQUENCE     {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo,
            ratType                     RatType,
            measurementReport           PRACH_MeasurementReport
}
```

ASN.1 Type Definition	
Type Name	PRACH_MeasurementReport
Comment	
Type Definition	

```
SEQUENCE    {
            usedPRACH_AcessSlot         INTEGER (0..14),
            usedPRACH_Signature         INTEGER (0..15) OPTIONAL
}
```

7.3.2.2.9 CPHY_RL_Modify

ASN.1 ASP Type Definition	
Type Name	CPHY_RL_Modify_CNF
PCO Type	CSAP
Comment	To confirm to modify the Radio Link
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_RL_Modify_REQ
PCO Type	CSAP
Comment	To request to modify the Radio Link HardHandover (PhysicalChannelReconfig) ChannelisationCodeChange FrequencyChange PhysicalChannelModifyForTrCHReconfig CompressedMode(PhysicalChannelReconfig) Re_Synchronized HardHandover Softhandover
Type Definition	

```
SEQUENCE        {
                cellId                  INTEGER(0..63),
                routingInfo             RoutingInfo,
                ratType                 RatType,
                modifyMessage           CphyRlModifyReq
}
```

ASN.1 Type Definition	
Type Name	CphyRlModifyReq
Comment	
Type Definition	

```
SEQUENCE    {
        activationTime                  SS_ActivationTime,
        physicalChannelInfo
            CHOICE    {         dpch_CompressedModeStatusInfo
    Dpch_CompressedModeStatusInfo,
                secondaryCCPCHInfo          SecondaryCCPCHInfo,
                pRACHInfo                   PRACHInfo,
                dPCHInfo                    DPCHInfo,
            }
}
```

ASN.1 Type Definition	
Type Name	SS_ActivationTime
Comment	
Type Definition	

```
CHOICE {
        activationCFN                   ActivationTime,
        activateNow                     NULL
}
```

7.3.2.2.10 CPHY_RL_Release

ASN.1 ASP Type Definition	
Type Name	CPHY_ RL_Release_CNF
PCO Type	CSAP
Comment	PHY emulator confirms that a specified physical channel has been released.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_RL_Release_REQ
PCO Type	CSAP
Comment	To request to release the Radio Link
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo
}
```

7.3.2.2.11 CPHY_RL_Setup

ASN.1 ASP Type Definition	
Type Name	CPHY_RL_Setup_CNF
PCO Type	CSAP
Comment	To confirm to setup the Radio Link
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_RL_Setup_REQ
PCO Type	CSAP
Comment	To request to setup the associated transport channels and the Radio Link itself.
Type Definition	

```
SEQUENCE     {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo,
            ratType                     RatType,
            setupMessage                CphyRlSetupReq
}
```

ASN.1 Type Definition	
Type Name	CphyRlSetupReq
Comment	To request to setup the Radio Link
Type Definition	

```
SEQUENCE    {
        physicalChannelInfo         CHOICE      {
            primaryCPICHInfo            PrimaryCPICHInfo,
            secondaryCPICHInfo          SecondaryCPICHInfo,
            primarySCHInfo              PrimarySCHInfo,
            secondarySCHInfo            SecondarySCHInfo,
            primaryCCPCHInfo            PrimaryCCPCHInfo,
            secondaryCCPCHInfo          SecondaryCCPCHInfo,
            pRACHInfo                   PRACHInfo,
            pICHInfo                    PICHInfo,
            aICHInfo                    AICHInfo,
            dPCHInfo                    DPCHInfo
        --  pCPCHInfo                   PCPCHInfo,
        --  aP_ICHInfo                  AP_AICHInfo,
        --  cD_ICHInfo                  CD_ICHInfo,
        --  cD_CA_ichInfo               CD_CA_ICHInfo,
        --  cSICHInfo                   CSICHInfo,
        --  pDSCHInfo                   PDSCHInfo,
        --  pUSCHInfo                   PUSCHinfo
    }
}
```

ASN.1 Type Definition	
Type Name	PrimaryCPICHInfo
Comment	
Type Definition	

```
SEQUENCE    {
            dl_TxPower_PCPICH           DL_TxPower_PCPICH,
            tx_diversityIndicator       BOOLEAN
}
```

ASN.1 Type Definition	
Type Name	SecondaryCPICHInfo
Comment	
Type Definition	

```
SEQUENCE {
            scramblingCode              INTEGER{0..63},
            dl_ChannelizationCode       SF512_AndCodeNumber,
            dl_TxPower                  DL_TxPower
}
```

ASN.1 Type Definition	
Type Name	PrimarySCHInfo
Comment	
Type Definition	

```
SEQUENCE {
            tstdIndicator                   BOOLEAN,
            dl_TxPower                      DL_TxPower
}
```

ASN.1 Type Definition	
Type Name	SecondarySCHInfo
Comment	
Type Definition	

```
SEQUENCE    {
            tstdIndicator                   BOOLEAN,
            dl_TxPower                      DL_TxPower
}
```

ASN.1 Type Definition	
Type Name	PrimaryCCPCHInfo
Comment	
Type Definition	

```
SEQUENCE    {
            sttd_Indicator                  BOOLEAN,
            dl_TxPower                      DL_TxPower
        -- timeSlot                         TimeSlot                 OPTIONAL,
        -- burstType                        BurstType                OPTIONAL,
        -- offset                           Offset                   OPTIONAL,
        --  repetitionPeriod                RepetitionPeriod         OPTIONAL,
        --  repetitionLength                RepetitionLength         OPTIONAL,
}
```

ASN.1 Type Definition	
Type Name	SecondaryCCPCHInfo
Comment	The range for powerOffsetOfTFCI_PO1 and powerOffsetOfPILOT_PO3 is 0-6 dB, 0.25 dB per step.
Type Definition	

```
SEQUENCE {
            scramblingCode                  INTEGER(0..63),
            dl_ChannelizationCode           SF256_AndCodeNumber,
            sCCPCHSlotFormat                SCCPCHSlotFormat,
            timingOffset                    INTEGER (0..149),
            positionFixedOrFlexible         PositionFixedOrFlexible,
            sttd_Indicator                  BOOLEAN,
            dl_TxPower                      DL_TxPower,
            powerOffsetOfTFCI_PO1           INTEGER (0..24),
            powerOffsetOfPILOT_PO3          INTEGER (0..24)
            -- timeSlot                     TimeSlot                     OPTIONAL,
            -- burstType                    BurstType                    OPTIONAL,
            -- midambleShift                MidambleShift                OPTIONAL,
            -- offset                       Offset                       OPTIONAL,
            -- repetitionPeriod             RepetitionPeriod             OPTIONAL,
            -- repetitionLength             RepetitionLength             OPTIONAL,
            -- tFCIPresence                 TFCIPresence                 OPTIONAL,
}
```

ASN.1 Type Definition	
Type Name	PRACHInfo
Comment	
Type Definition	

```
SEQUENCE {
            fdd_tdd                 CHOICE  {
            fdd
                SEQUENCE   {
                    preambleSignature               AvailableSignatures,
                    spreadingFactorForDataPart      SF_PRACH,
                    preambleScramblingCode          PreambleScramblingCodeWordNumber,
                    puncturingLimit                 PuncturingLimit,
                    accessSlot                      AvailableSubChannelNumbers
                },
            tdd
                SEQUENCE   {
                    -- timeSlot                     TimeSlot,
                    -- spreadingCode                SpreadingCode,
                    -- midambleCode                 MidambleCode,
                }
}
}
```

ASN.1 Type Definition	
Type Name	PICHInfo
Comment	
Type Definition	

```
SEQUENCE    {
            pichinfo                    PICH_Info,
            dl_TxPower                  DL_TxPower
}
```

ASN.1 Type Definition	
Type Name	AICHInfo
Comment	
Type Definition	

```
SEQUENCE    {
            aichinfo                    AICH_Info,
            dl_TxPower                  DL_TxPower
}
```

ASN.1 Type Definition	
Type Name	DPCHInfo
Comment	At least one of the fields shall be present.
Type Definition	

```
SEQUENCE    {
            ul_DPCH_Info                UL_DPCH_Info    OPTIONAL,
            dl_DPCHInfo                 DL_DPCHInfo     OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	DL_DPCHInfo
Comment	The range for powerOffsetOfTPC_PO2 and powerOffsetOfTFCI_PO1 and powerOffsetOfPILOT_PO3 is 0-6 dB, 0.25 dB per step.
Type Definition	

```
SEQUENCE    {
            dl_CommonInformation        DL_CommonInformation,
            dl_DPCH_InfoPerRL           DL_DPCH_InfoPerRL,
            powerOffsetOfTFCI_PO1       INTEGER (0..24),
            powerOffsetOfTPC_PO2        INTEGER (0..24),
            powerOffsetOfPILOT_PO3      INTEGER (0..24),
            dl_TxPower                  DL_TxPower,
            dl_TxPowerMax               DL_TxPower,
            dl_TxPowerMin               DL_TxPower
}
```

ASN.1 Type Definition	
Type Name	DL_TxPower_PCPICH
Comment	Absolute Tx Power of PCPICH
Type Definition	
`INTEGER (-60..-30)`	

ASN.1 Type Definition	
Type Name	DL_TxPower
Comment	Downlink Tx Power relative to PCPICH
Type Definition	
`INTEGER (-35..+15)`	

ASN.1 Type Definition	
Type Name	SCCPCHSlotFormat
Comment	Reference to 3GPP TS25.211 [40]
Type Definition	
`INTEGER (0..17)`	

ASN.1 Type Definition	
Type Name	UL_DPCCHSlotFormat
Comment	Reference to 3GPP TS 25.211 [40]
Type Definition	
`INTEGER (0..5)`	

7.3.2.2.12 CPHY_Sync

ASN.1 ASP Type Definition	
Type Name	CPHY_Sync_IND
PCO Type	CSAP
Comment	To indicate that physical channel synchronization (in FDD mode, sync with DPCCH) has been achieved.
Type Definition	

```
SEQUENCE    {
                cellId                      INTEGER(0..63),
                routingInfo                 RoutingInfo
}
```

7.3.2.2.13 CPHY_TrCH_Config

ASN.1 ASP Type Definition	
Type Name	CPHY_TrCH_Config_CNF
PCO Type	CSAP
Comment	To confirm to configure the transport channel
Type Definition	

```
SEQUENCE    {
            cellId                          INTEGER(0..63),
            routingInfo                     RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_TrCH_Config_REQ
PCO Type	CSAP
Comment	To request to configure the transport channel
Type Definition	

```
SEQUENCE    {
            cellId                          INTEGER(0..63),
            routingInfo                     RoutingInfo,
            ratType                         RatType,
            configMessage                   CphyTrchConfigReq
}
```

ASN.1 Type Definition	
Type Name	CphyTrchConfigReq
Comment	To request to configure the transport channel. The same TFCS information should be provided to the PHY and MAC layers at all times. When a CPHY_TrCH_Config_REQ is used to configure the PHY layer, a corresponding CMAC_Config_REQ should be sent to the MAC layer to ensure that the configuration is consistent.
Type Definition	

```
SEQUENCE    {
        activationTime      SS_ActivationTime,
        ulconnectedTrCHList     SEQUENCE (SIZE (0..maxTrCH)) OF SEQUENCE {
                trchid                  TransportChannelIdentity,
                ul_TransportChannelType SS_UL_TransportChannelType,
                transportChannelInfo    CommonOrDedicatedTFS
                                                                    }   OPTIONAL,
        ulTFCS              TFCS    OPTIONAL,
        dlconnectedTrCHList     SEQUENCE (SIZE (0..maxTrCH)) OF SEQUENCE {
                trchid                  TransportChannelIdentity,
                dl_TransportChannelType SS_DL_TransportChannelType,
                transportChannelInfo    CommonOrDedicatedTFS
                                                                    }   OPTIONAL,
        dlTFCS              TFCS    OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	RoutingInfo
Comment	To route between each channels.
Type Definition	

```
CHOICE     {
        physicalChannelIdentity         INTEGER                 {0..31},
        transportChannelIdentity        TransportChannelIdentity,
        logicalChannelIdentity          LogicalChannelIdentity,
        rB_Identity                     INTEGER                 {-31..32},
        cn-DomainIdentity               CN-DomainIdentity
}
```

ASN.1 Type Definition	
Type Name	RatType
Comment	To select route between each channels.
Type Definition	

```
ENUMERATED  {
            fdd (0), tdd (1)
}
```

ASN.1 Type Definition	
Type Name	CommonOrDedicatedTFS
Comment	Transport Format Set
Type Definition	

```
SEQUENCE {
    tti                                 CHOICE {
        tti10                               CommonOrDedicatedTF_InfoList,
        tti20                               CommonOrDedicatedTF_InfoList,
        tti40                               CommonOrDedicatedTF_InfoList,
        tti80                               CommonOrDedicatedTF_InfoList,
        dynamic                         CommonOrDedicatedTF_InfoList_DynamicTTI
    },
    semistaticTF_Information            SemistaticTF_Information
}
```

ASN.1 Type Definition	
Type Name	CommonOrDedicatedTF_InfoList
Comment	Transport Format Set
Type Definition	

```
SEQUENCE (SIZE (1..maxTF)) OF CommonOrDedicatedTF_Info
```

ASN.1 Type Definition	
Type Name	CommonOrDedicatedTF_Info
Comment	Transport Format Set
Type Definition	

```
SEQUENCE {
    tb_Size                 INTEGER (0..5035),
    numberOfTbSizeList      SEQUENCE (SIZE (1..maxTF)) OF NumberOfTransportBlocks,
    logicalChannelList      LogicalChannelList
}
```

ASN.1 Type Definition	
Type Name	CommonOrDedicatedTF_InfoList_DynamicTTI
Comment	Transport Format Set for TDD mode
Type Definition	

```
SEQUENCE {
    tb_Size                 INTEGER (0..5035),
    numberOfTbSizeList      SEQUENCE (SIZE (1..maxTF)) OF NumberOfTransportBlocks,
    logicalChannelList      LogicalChannelList
}
```

7.3.2.2.14 CPHY_TrCH_Release

ASN.1 ASP Type Definition	
Type Name	CPHY_TrCH_Release_REQ
PCO Type	CSAP
Comment	To request to release the Radio Link
Type Definition	

```
SEQUENCE    {
        cellId                      INTEGER(0..63),
        routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CPHY_TrCH_Release_CNF
PCO Type	CSAP
Comment	To confirm to release the Radio Link
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(0..63),
            routingInfo             RoutingInfo
}
```

7.3.2.2.15 CMAC_BMC_Scheduling

ASN.1 ASP Type Definition	
Type Name	CMAC_BMC_Scheduling_CNF
PCO Type	CSAP
Comment	To confirm the BMC scheduling.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_BMC_Scheduling_REQ
PCO Type	CSAP
Comment	Send the BMC scheduling information to the MAC.
Type Definition	

```
SEQUENCE    {
                cellId                      INTEGER(0..63),
                routingInfo                 RoutingInfo,
                ratType                     RatType,
                schedulingInfo              BMC_SchedulingInfo
}
```

ASN.1 Type Definition	
Type Name	BMC_SchedulingInfo
Comment	
Type Definition	

```
SEQUENCE    {
            level1Info                  BMC_SchedulingLevel1Info,
            level2Info                  BMC_SchedulingLevel2Info    OPTIONAL

}
```

ASN.1 Type Definition	
Type Name	BMC_SchedulingLevel2Info
Comment	
Type Definition	

```
SEQUENCE    {
            starCtchBsIndex             INTEGER (1..256)        DEFAULT 1,
            drxSelectionBitmap          OCTET STRING
}
```

ASN.1 Type Definition	
Type Name	BMC_SchedulingLevel1Info
Comment	0<=K<=N-1 (3GPP TS 25.331 [21], clause 8.5.16)
Type Definition	

```
SEQUENCE    {
            ctchAllocationPeriod        INTEGER (1..256),       -- N
            cbsFrameOffset              INTEGER (0..255)        -- K
}
```

7.3.2.2.16 CMAC_Ciphering_Activate

ASN.1 ASP Type Definition	
Type Name	CMAC_Ciphering_Activate_CNF
PCO Type	CSAP
Comment	To confirm to activate or inactivate the ciphering
Type Definition	

```
SEQUENCE {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_Ciphering_Activate_REQ
PCO Type	CSAP
Comment	To request to start, restart or stop downlink ciphering or uplink deciphering. The physicalChannelIdentity of DPCH applies to routingInfo.
Type Definition	

```
SEQUENCE {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo,
            ratType                     RatType,
            cipheringModeInfo           CipheringModeInfo
}
```

7.3.2.2.17 CMAC_Config

ASN.1 ASP Type Definition	
Type Name	CMAC_Config_CNF
PCO Type	CSAP
Comment	For MAC emulator to report that a previous attempt to setup, reconfigure or release a logical channel is successful.
Type Definition	

```
SEQUENCE    {
                cellId                  INTEGER(-1..63),
                routingInfo             RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_Config_REQ
PCO Type	CSAP
Comment	To request to configure MAC entity. Setup is used for creation of the MAC instances or the MAC resources. Release is used for free the all MAC resources. The reconfiguration is to change the MAC parameters, it is not the MAC modification.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo,
            ratType                     RatType,
            configMessage               CHOICE {
                setup                       CmacConfigReq,
                reconfigure                 CmacConfigReq,
                release                     NULL
            }
}
```

ASN.1 Type Definition	
Type Name	CmacConfigReq
Comment	To request to configure MAC
Type Definition	

```
SEQUENCE    {
            activationTime                      SS_ActivationTime,
            uE_Info                                 UE_Info,
            trCHInfo                                TrCHInfo,
            trCH_LogCHMapping                       TrCH_LogCHMappingList1
            -- RACHTrasmissionCtrolElements     TBD,
            -- CPCHTransmissionControlElements  TBD
}
```

ASN.1 Type Definition	
Type Name	UE_Info
Comment	
Type Definition	

```
SEQUENCE {
        u_RNTI                          U_RNTI      OPTIONAL,
        c_RNTI                          C_RNTI      OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	TrCH_LogCHMappingList1
Comment	maxulTrCH = maxdlTrCH = 16
Type Definition	

```
SEQUENCE    {
        ulconnectedTrCHList     SEQUENCE (SIZE (1..maxulTrCH)) OF SEQUENCE {
                trchid                  TransportChannelIdentity,
                trCH_LogCHMappingList   TrCH_LogCHMappingList
                                                        },  OPTIONAL,
        dlconnectedTrCHList     SEQUENCE (SIZE (1..maxdlTrCH)) OF SEQUENCE {
                trchid                  TransportChannelIdentity,
                trCH_LogCHMappingList   TrCH_LogCHMappingList
                                                        },  OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	TrCH_LogCHMappingList
Comment	maxLogCHperTrCH = 15
Type Definition	

```
SEQUENCE (SIZE (1..maxLogCHperTrCH)) OF     TrCH_LogicalChannelMapping
```

ASN.1 Type Definition	
Type Name	TrCHInfo
Comment	The same TFCS information should be provided to the PHY and MAC layers at all times. When a CMAC_Config_REQ is used to configure the MAC layer, a corresponding CPHY_TrCH_Config_REQ should be sent to the PHY layer to ensure that the configuration is consistent.
Type Definition	

```
SEQUENCE    {
        ulconnectedTrCHList     SEQUENCE (SIZE (1..maxulTrCH)) OF SEQUENCE {
                trchid                  TransportChannelIdentity,
                transportChannelInfo    CommonOrDedicatedTFS
                                                        }   OPTIONAL,
        ulTFCS              TFCS    OPTIONAL,
        dlconnectedTrCHList     SEQUENCE (SIZE (1..maxdlTrCH)) OF SEQUENCE {
                trchid                  TransportChannelIdentity,
        transportChannelInfo    CommonOrDedicatedTFS
                                                        }   OPTIONAL,
        dlTFCS              TFCS    OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	TrCH_LogicalChannelMapping
Comment	When map the common transport channels onto DCCH/DTCH in MAC-c/sh, rB_Identity is omitted. It is included in MAC-d mapping.
Type Definition	

```
SEQUENCE    {
            CHOICE  {
                ul_LogicalChannelMapping            SS_UL_LogicalChannelMapping,
                dl_LogicalChannelMapping            SS_DL_LogicalChannelMapping
                    },
            rB_Identity                 INTEGER     {-31..32}   OPTIONAL,
            cn-DomainIdentity           CN-DomainIdentity       OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	SS_UL_LogicalChannelMapping
Comment	If the macHeaderManipulation field is 'NormalMacHeader', then data received on the transport channel supporting this logical channel shall have it's MAC header inspected to determine the appropriate routing, and removed as normal. The MAC SDU shall be passed to the appropriate logical channel. If the macHeaderManipulation field field is 'OmitMacHeader', then data received on the transport channel supporting this logical channel shall have it's MAC header inspected to determine the appropriate routing, but the MAC layer shall not remove the MAC header. Thus the entire MAC PDU shall be passed to the appropriate logical channel, and the MAC header can be checked by the TTCN.
Type Definition	

```
SEQUENCE {
    macHeaderManipulation               MAC_HeaderManipulation,
    ul_TransportChannelType             SS_UL_TransportChannelType,
    logicalChannelIdentity              LogicalChannelIdentity,
    logicalChannelType                  LogicalChannelType
}
```

ASN.1 Type Definition	
Type Name	SS_DL_LogicalChannelMapping
Comment	If the macHeaderManipulation field is 'NormalMacHeader', then data transmitted on this logical channel shall have an appropriate MAC header added before it is sent to lower layers for transmission. If the macHeaderManipulation field is 'OmitMacHeader', then data transmitted on this logical channel shall not have any MAC header information added, even if the logical channel type and mapping indicates that there should be a MAC header present. This allows the entire MAC PDU to be specified in the TTCN, so individual fields in the MAC header can be modified.
Type Definition	

```
SEQUENCE {
    macHeaderManipulation               MAC_HeaderManipulation,
    dlTransportChannelType              SS_DL_TransportChannelType,
    logicalChannelIdentity              LogicalChannelIdentity,
    logicalChannelType                  LogicalChannelType,
    rlc_SizeList                        CHOICE {
            allSizes                            NULL,
            configured                          NULL,
            explicitList                        RLC_SizeExplicitList},
    mac_LogicalChannelPriority          MAC_LogicalChannelPriority  OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	SS_UL_TransportChannelType
Comment	
Type Definition	

```
ENUMELATED {
    dch (0),
    rach (1),
    cpch (2),
    usch (3)
}
```

ASN.1 Type Definition	
Type Name	MAC_LogicalChannelPriority
Comment	
Type Definition	

```
INTEGER (1..8)
```

ASN.1 Type Definition	
Type Name	SS_DL_TransportChannelType
Comment	
Type Definition	

```
ENUMELATED {
    dch     (0),
    fach    (1),
    bch     (2),
    pch     (3),
    dsch    (4)
}
```

ASN.1 Type Definition	
Type Name	LogicalChannelType
Comment	
Type Definition	

```
ENUMERATED {
    BCCH    (0),
    PCCH    (1),
    CCCH    (2),
    CTCH    (3),
    DCCH    (4),
    DTCH    (5),
    SHCCH   (6)
}
```

ASN.1 Type Definition	
Type Name	MAC_HeaderManipulation
Comment	
Type Definition	

```
ENUMERATED  {
            NormalMacHeader (0),
            OmitMacHeader   (1)
}
```

7.3.2.2.18 CMAC_PAGING_Config

ASN.1 ASP Type Definition	
Type Name	CMAC_PAGING_Config_CNF
PCO Type	CSAP
Comment	To confirm to setup the paging message
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_PAGING_Config_REQ
PCO Type	CSAP
Comment	To request MAC layer to send the Paging message on the specified configuration.
Type Definition	

```
SEQUENCE     {
                cellId                      INTEGER(0..63),
                routingInfo                 RoutingInfo,
                ratType                     RatType,
                configMessage               CmacPagingConfigReq
}
```

ASN.1 Type Definition	
Type Name	CmacPagingConfigReq
Comment	
Type Definition	

```
SEQUENCE  {
            pI_BitMapInfo       CHOICE {
                e18                 BIT STRING (SIZE (18)),
                e36                 BIT STRING (SIZE (36)),
                e72                 BIT STRING (SIZE (72)),
                e144                BIT STRING (SIZE (144))
                                        },
            dRX_CycleLength     INTEGER {3..9},
            iMSI                SEQUENCE (SIZE (6..15)) OF  Digit,
            t_pich_T_sccpch     BOOLEAN         --  T_pich>T_sccpch then FALSE
}
```

7.3.2.2.19 CMAC_Restriction

ASN.1 ASP Type Definition	
Type Name	CMAC_Restriction_CNF
PCO Type	CSAP
Comment	For MAC emulator to report that a previous attempt of restricting TFCs have been successful.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_Restriction_REQ
PCO Type	CSAP
Comment	To request to configure MAC entity. The field restrictAllowedTFCs is provided to allow the UL and/or DL SS TFCS to be restricted for a specific transport channel. This information only needs to be sent to the MAC layer, since it is the MAC layer's responsibility to determine the set of valid TFCs each TTI.
Type Definition	

```
SEQUENCE {
        cellIdentity                INTEGER (-1..63),
        routingInfo                 RoutingInfo,
        ratType                     RatType,
        restrictAllowedTFCs         TFC_Restriction
}
```

<table>
<tr><th colspan="2">ASN.1 Type Definition</th></tr>
<tr><td>Type Name</td><td>TFC_Restriction</td></tr>
<tr><td>Comment</td><td>This type is used to specify the allowed TFCs within the current TFCS. A TFC restriction is applicable until a subsequent TFC restriction is applied. TFC restrictions are not cumulative, so each TFC restriction completely replaces the previous TFC restriction.
The downlink restriction can be used to ensure that the SS uses a specific TFC for transmission of data, by only allowing the 'No data' TFC, and the 'desired' TFC. It may also be necessary to include one or more 'signalling only' TFCs to allow signalling to occur.
The uplink restriction can be used to verify that the UE has used a specific TFC. Any data received by the SS using a forbidden TFCI shall be discarded.</td></tr>
<tr><th colspan="2">Type Definition</th></tr>
<tr><td colspan="2"><pre>SEQUENCE {
        ulTFCI_Restriction   TFC_Subset OPTIONAL,
        dlTFCI_Restriction   TFC_Subset OPTIONAL
}</pre></td></tr>
<tr><td>Detailed Comments</td><td>SS requirements for downlink.
1. The SS MAC layer shall not use a restrictednon-allowed TFC for DL.
2. The SS MAC layer shall not use a TFC that requires the SS RLC layer to provide padding PDUs (3GPP TS 25.322 [18])
3. In the case that there is data pending on one or more RLC entities, but not enough to use one of the allowed TFCs:
a. The SS MAC layer shall use the 'No data' TFC until there is enough data in the RLC to use another allowed TFC.
b. The SS RLC layer shall buffer the data until there is enough data in the RLC entities for the MAC layer to use an allowed TFC other than the 'No data' TFC for transmission of the data.
NB: The TTCN author is responsible for ensuring:
1. The SDU discard function is not configured for TM and UM entities in the UE, and is configured to no_discard for AM entities in the UE.
2. That RLC SDUs that are expected to be sent in the same TTI (due to a TFC restriction) are sent as quickly as possible to minimise the number of 'no data' TFCs used by the MAC layer, and the amount of buffering that must be performed by the RLC layer.
SS requirements for uplink:
The SS shall discard all data received using a restricted non-allowed TFC.</td></tr>
</table>

7.3.2.2.20 CMAC_SecurityMode_Config

<table>
<tr><th colspan="2">ASN.1 ASP Type Definition</th></tr>
<tr><td>Type Name</td><td>CMAC_SecurityMode_Config_CNF</td></tr>
<tr><td>PCO Type</td><td>CSAP</td></tr>
<tr><td>Comment</td><td>To confirm to configure the MAC security mode</td></tr>
<tr><th colspan="2">Type Definition</th></tr>
<tr><td colspan="2"><pre>SEQUENCE   {
            cellId   INTEGER(-1..63)
}</pre></td></tr>
</table>

<table>
<tr><th colspan="2">ASN.1 ASP Type Definition</th></tr>
<tr><td>Type Name</td><td>CMAC_SecurityMode_Config_REQ</td></tr>
<tr><td>PCO Type</td><td>CSAP</td></tr>
<tr><td>Comment</td><td>To request to configure the MAC security mode</td></tr>
<tr><th colspan="2">Type Definition</th></tr>
<tr><td colspan="2"><pre>SEQUENCE {
            cellId   INTEGER(-1..63),
            macCipheringInfo   SecurityInfo
}</pre></td></tr>
</table>

7.3.2.2.21 CMAC_SequenceNumber

ASN.1 ASP Type Definition	
Type Name	CMAC_Sequence_Number_CNF
PCO Type	CSAP
Comment	To return the requested counter sequence number on MAC-d DCH. The physicalChannelIdentity of DPCH applies to routingInfo.
Type Definition	

```
SEQUENCE    {
            cellId                   INTEGER(-1..63),
            routingInfo              RoutingInfo,
            count_C_MSB_UL           COUNT_C_MSB ,
            count_C_MSB_DL           COUNT_C_MSB
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_SequenceNumber_REQ
PCO Type	CSAP
Comment	To request the MAC layer to return current counter sequence numbers. The physicalChannelIdentity of DPCH applies to routingInfo.
Type Definition	

```
SEQUENCE    {
            cellId                       INTEGER(-1..63),
            routingInfo                  RoutingInfo
}
```

7.3.2.2.22 CMAC_SYSINFO_Config

ASN.1 ASP Type Definition	
Type Name	CMAC_SYSINFO_Config_CNF
PCO Type	CSAP
Comment	To confirm to setup the system information block
Type Definition	

```
SEQUENCE    {
            cellId                       INTEGER(0..63),
            routingInfo                  RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CMAC_SYSINFO_Config_REQ
PCO Type	CSAP
Comment	To request MAC layer to send the BCCH message on the specified configuration.
Type Definition	

```
SEQUENCE     {
                cellId                      INTEGER(0..63),
                routingInfo                 RoutingInfo,
                ratType                     RatType,
                configMessage               CmacSysinfoConfigReq
}
```

ASN.1 Type Definition	
Type Name	CmacSysinfoConfigReq
Comment	
Type Definition	

```
SEQUENCE    {

        sg_REP                  INTEGER     (2..12),
                                -- Repetition period is the sg_REP-th power of 2.
        sg_POS                  INTEGER     (0..2047),
                                -- The position of each segment is 2 * sg_POS.
        bcch_ModificationTime   BCCH_ModificationTime   OPTIONAL
}
```

7.3.2.2.23 CRLC_Ciphering_Activate

ASN.1 ASP Type Definition	
Type Name	CRLC_Ciphering_Activate_CNF
PCO Type	CSAP
Comment	To confirm to activate or inactivate the ciphering
Type Definition	

```
SEQUENCE {
            cellId                  INTEGER(-1..63)

}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_Ciphering_Activate_REQ
PCO Type	CSAP
Comment	To request to start, restart or stop downlink ciphering or uplink deciphering. Each call of the ASP includes one RLC SN in rb-DL-CiphActivationTimeInfo for the corresponding rb-identity.
Type Definition	

```
SEQUENCE {
            cellId                      INTEGER(-1..63),

            ratType                     RatType,
            ciphActivationInfo          CiphActivationInfo}
```

ASN.1 Type Definition	
Type Name	CiphActivationInfo
Comment	DL or UL ciphering activation info
Type Definition	

```
CHOICE {
            cipheringModeInfo               CipheringModeInfo,
            rb_UL_CiphActivationTimeInfo    RB_ActivationTimeInfoList
}
```

7.3.2.2.24 CRLC_Config

ASN.1 ASP Type Definition	
Type Name	CRLC_Config_CNF
PCO Type	CSAP
Comment	For RLC emulator to confirm that a previous attempt to establish, re_configure or release a radio bearer has been successful.
Type Definition	

```
SEQUENCE  {
             cellId                    INTEGER(-1..63),
             routingInfo               RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_Config_REQ
PCO Type	CSAP
Comment	To request to setup, reconfigure or release RLC entity
Type Definition	

```
SEQUENCE     {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo,
            ratType                     RatType,
            configMessage               CrlcConfigReq
}
```

ASN.1 Type Definition	
Type Name	CrlcConfigReq
Comment	To request to setup, re_configure release RLC entity The Stop parameter indicates that the RLC entity shall not transmit or receive RLC PDUs. The Continue parameter indicates that the RLC entity shall continue transmission and reception of RLC PDUs. When the RLC entity is stopped, the all protocol parameters, such as the protocol variables, RLC timers and status are not affected. Triggered polls and status transmissions are delayed until the RLC entity is continued.
Type Definition	

```
CHOICE  {
        setup                           RBInfo,
        reconfigure                     RBInfo,
        release                         NULL,
        stop                            NULL,
        continue                        NULL
}
```

ASN.1 Type Definition	
Type Name	RBInfo
Comment	
Type Definition	

```
SEQUENCE (
              sS_rlc_Info               SS_RLC_Info         OPTIONAL,
              rB_LogCH_Mapping          RB_LogCH_Mapping
}
```

ASN.1 Type Definition	
Type Name	RB_LogCH_Mapping
Comment	Provide mapping information between RB, logical channel and CN domain.
Type Definition	

```
SEQUENCE {
    uLlogicalChannelIdentity                LogicalChannelIdentity  OPTIONAL,
    dLlogicalChannelIdentity                LogicalChannelIdentity  OPTIONAL,
    logicalChannelType                      LogicalChannelType      OPTIONAL,
    cn-DomainIdentity                       CN-DomainIdentity       OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	SS_RLC_Info
Comment	UL and DL have been swapped intentionally in this type definition. This is to maximise re-use of the type definitions in 3GPP TS 25.331 [21] which are intended to configure a UE, where UL is transmission, and DL is reception. For the SS, UL is reception, and DL is transmission. For example, consider configuring a DL AM RLC entity (transmitter) in the SS. The transmission parameters to be configured include PollingInformation, Transmission-RLC-Discard etc. If the DL-AM-RLC-Mode type definition is used to configure this entity, it is only possible to configure reception parameters such as StatusInformation, and receiving window size. By swapping UL and DL, it is possible to configure the DL AM RLC entity using the existing type definition UL-AM-RLC-Info, which contains all of the required transmission parameters.
Type Definition	

```
SEQUENCE {
            sS_ul_RLC_Mode          DL_RLC_Mode             OPTIONAL,
            sS_dl_RLC_Mode          SS_DL_RLC_Mode          OPTIONAL
}
```

ASN.1 Type Definition	
Type Name	SS_DL_RLC_Mode
Comment	
Type Definition	

```
SEQUENCE    {
            dl_PayloadSize              PayloadSize     OPTIONAL,
            dl_RLCModeInfo              UL_RLC_Mode
}
```

ASN.1 Type Definition	
Type Name	PayloadSize
Comment	
Type Definition	
`INTEGER (0..4992)`	

7.3.2.2.25 CRLC_Integrity_Activate

ASN.1 ASP Type Definition	
Type Name	CRLC_integrity_Activate_CNF
PCO Type	CSAP
Comment	To confirm to activate or inactivate the integrity protection
Type Definition	

```
SEQUENCE {
        cellId              INTEGER(-1..63)
}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_Integrity_Activate_REQ
PCO Type	CSAP
Comment	To request to start or to modifythe the downlink or uplink integrity protection. The ASP shall be called before send SECURITY MODE COMMAND. It activates the integrity on all SRBs in DL. Not to call the ASP if wishing to switch off the integrity in the test case.
Type Definition	

```
SEQUENCE {
        cellId                          INTEGER(-1..63),
        integrityActivationInfo         IntegrityActivationInfo
}
```

ASN.1 Type Definition	
Type Name	IntegrityActivationInfo
Comment	DL or UL integrity activation info
Type Definition	

```
CHOICE {
        integrityProtectionModeInfo     IntegrityProtectionModeInfo,
        ul-IntegProtActivationInfo      IntegrityProtActivationInfo
}
```

7.3.2.2.26 CRLC_Integrity_Failure

ASN.1 ASP Type Definition	
Type Name	CRLC_Integrity_Failure_IND
PCO Type	CSAP
Comment	RLC emulator reports the occurrences of a failure in integrity protection, i.e. reception of an integrity-protected RLC AM/UM SDU containing a non-matching X-MAC value compared to the desired.
Type Definition	

```
SEQUENCE    {
        cellId                  INTEGER(-1..63),
        routingInfo             RoutingInfo,
        failureCause            ENUMERATED { codeNotMatched(0) }
        -- the enumerated types of failure cause field is ffs
}
```

7.3.2.2.27 CRLC_Resume

ASN.1 ASP Type Definition	
Type Name	CRLC_Resume_CNF
PCO Type	CSAP
Comment	To confirm the resume request
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo
}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_Resume_REQ
PCO Type	CSAP
Comment	To request to resume data transmission
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo
}
```

7.3.2.2.28 CRLC_SecurityMode_Config

ASN.1 ASP Type Definition	
Type Name	CRLC_SecurityMode_Config_CNF
PCO Type	CSAP
Comment	To confirm to configure the RLC security mode
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63)
}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_SecurityMode_Config_REQ
PCO Type	CSAP
Comment	To request to configure the RLC security mode
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63),
            rlcSecurityInfo         SecurityInfo}
```

ASN.1 Type Definition	
Type Name	SecurityInfo
Comment	The integrityKey is not applicable to MAC
Type Definition	

```
SEQUENCE{
            cn-DomainIdentity               CN-DomainIdentity,
            startValue                      START_VALUE,
            cipheringKey                    BITSTRING(128)          OPTIONAL,
            integrityKey                    BITSTRING(128)          OPTIONAL,
            gsmCiperingKey                  BITSTRING(64)           OPTIONAL
}
```

Detailed Comments	SecurtiyInfo contains either a new START_VALUE for the existing security keys, or a set of new security keys with the zero value for START. The START value is activated at the activation time.

7.3.2.2.29 CRLC_SequenceNumber

ASN.1 ASP Type Definition	
Type Name	CRLC_Sequence_Number_CNF
PCO Type	CSAP
Comment	To return the requested counter sequence number
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63),
            routingInfo             RoutingInfo,
            count_C_MSB_UL          COUNT_C_MSB,
            count_C_LSB_UL          RLC_SequenceNumber,
            count_C_MSB_DL          COUNT_C_MSB,
            count_C_LSB_DL          RLC_SequenceNumber
}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_SequenceNumber_REQ
PCO Type	CSAP
Comment	To request the RLC layer to return current counter sequence numbers
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63),
            routingInfo             RoutingInfo
}
```

7.3.2.2.30 CRLC_Status

ASN.1 ASP Type Definition	
Type Name	CRLC_Status_IND
PCO Type	CSAP
Comment	To report the occurrence of certain events to RRC. Note: the possible event types to be defined for this ASP is FFS.
Type Definition	

```
SEQUENCE    {
            cellId              INTEGER(-1..63),
            routingInfo         RoutingInfo,
            ratType             RatType,
            statusInd           CrlcStatusInd
}
```

ASN.1 Type Definition	
Type Name	CrlcStatusInd
Comment	
Type Definition	

```
ENUMERATED  {
            DataLinkFailure (0)
            MaxRESET        (1),
            SDUDiscarded    (2)
            -- __ More event types are to be added here
}
```

7.3.2.2.31 CRLC_Suspend

ASN.1 ASP Type Definition	
Type Name	CRLC_Suspend_CNF
PCO Type	CSAP
Comment	To confirm to suspend data transmission
Type Definition	

```
SEQUENCE    {
            cellId              INTEGER(-1..63),
            routingInfo         RoutingInfo,
            n                   RLC_SequenceNumber
}
```

ASN.1 ASP Type Definition	
Type Name	CRLC_Suspend_REQ
PCO Type	CSAP
Comment	To request to suspend data transmission
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63),
            routingInfo             RoutingInfo,
            n                       RLC_SequenceNumber
}
```

7.3.2.2.32 CBMC_Config

ASN.1 ASP Type Definition	
Type Name	CBMC_Config_CNF
PCO Type	CSAP
Comment	To confirm the BMC configuration, reconfiguration or release.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(0..63),
            routingInfo                 RoutingInfo         -- RBid
}
```

ASN.1 ASP Type Definition	
Type Name	CBMC_Config_REQ
PCO Type	CSAP
Comment	To request the configuration, reconfiguration or release of BMC.
Type Definition	

```
SEQUENCE    {
        cellId                      INTEGER(0..63),
        routingInfo                 RoutingInfo,            -- RBid
        configMessage               CHOICE {
            setup                       BMC_SchedulingInfo,
            release                     NULL}
}
```

7.3.2.2.33 RLC_TR_DATA

ASN.1 ASP Type Definition	
Type Name	RLC_TR_DATA_REQ
PCO Type	DSAP
Comment	To request to transmit DATA using transparent mode.
Type Definition	

```
SEQUENCE    {
                cellId                  INTEGER(-1..63),
                routingInfo             RoutingInfo,
                tM_Message              CHOICE  {
                        dL_DCCH_Message             DL_DCCH_Message,
                        dL_CCCH_Message             DL_CCCH_Message,
                        pCCH_Message                PCCH_Message,
                        dL_SHCCH_Message            DL_SHCCH_Message,
                        bCCH_FACH_Message           BCCH_FACH_Message,
                        bCCH_BCH_Message            BCCH_BCH_Message,
                        invalid_dL_DCCH_Message     Invalid_DL_DCCH_Message,
                        invalid_dL_CCCH_Message     Invalid_DL_CCCH_Message,
                        invalid_dL_SHCCH_Message    Invalid_DL_SHCCH_Message}
}
```

ASN.1 ASP Type Definition	
Type Name	RLC_TR_DATA_IND
PCO Type	DSAP
Comment	To indicate to receive DATA using transparent mode.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo,
            tM_Message                  CHOICE  {
                        uL_DCCH_Message             UL_DCCH_Message,
                        uL_CCCH_Message             UL_CCCH_Message,
                        uL_SHCCH_Message            UL_SHCCH_Message}
}
```

7.3.2.2.34 RLC_AM_DATA

ASN.1 ASP Type Definition	
Type Name	RLC_AM_DATA_REQ
PCO Type	DSAP
Comment	To request to transmit DATA using acknowledged mode.
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER (-1..63),
            routingInfo         RoutingInfo,
            confirmationRequest     AmConfirmationRequest,
            aM_Message              CHOICE  {
                        dL_DCCH_Message             DL_DCCH_Message,
                        dL_CCCH_Message             DL_CCCH_Message,
                        pCCH_Message                PCCH_Message,
                        dL_SHCCH_Message            DL_SHCCH_Message,
                        bCCH_FACH_Message           BCCH_FACH_Message,
                        bCCH_BCH_Message            BCCH_BCH_Message,
                        invalid_dL_DCCH_Message     Invalid_DL_DCCH_Message,
                        invalid_dL_CCCH_Message     Invalid_DL_CCCH_Message,
                        invalid_dL_SHCCH_Message    Invalid_DL_SHCCH_Message}
}
```

ASN.1 Type Definition	
Type Name	AmConfirmationRequest
Comment	If the noConfirmationRequested option is used, then an RLC_AM_DATA_CNF is not expected from the RLC AM entity. If the confirmationRequested option is used, then the RLC AM entity is being requested to provide an RLC_AM_DATA_CNF primitive containing the same Mui value.
Type Definition	

```
CHOICE {
            noConfirmationRequest       NULL,
            confirmationRequested       Mui
}
```

ASN.1 Type Definition	
Type Name	Mui
Comment	
Type Definition	

```
INTEGER {0..4095}
```

ASN.1 ASP Type Definition	
Type Name	RLC_AM_DATA_IND
PCO Type	DSAP
Comment	To indicate to receive DATA using acknowledged mode.
Type Definition	

```
SEQUENCE    {
            cellId                      INTEGER(-1..63),
            routingInfo                 RoutingInfo,
            integrityResult             IntegrityResult,
            aM_Message                  CHOICE  {
                        uL_DCCH_Message             UL_DCCH_Message,
                        uL_CCCH_Message             UL_CCCH_Message,
                        uL_SHCCH_Message            UL_SHCCH_Message}
}
```

ASN.1 Type Definition	
Type Name	IntegrityResult
Comment	
Type Definition	

```
CHOICE  {
            integrityNotUsed        NULL,
            integrityUsed           IntegrityStatus
}
```

ASN.1 Type Definition	
Type Name	IntegrityStatus
Comment	
Type Definition	

```
ENUMERATED {
            i_pass(0), i_fail(1)
}
```

ASN.1 ASP Type Definition	
Type Name	RLC_AM_DATA_CNF
PCO Type	DSAP
Comment	For RLC emulator to report to the upper layer that a previously transmitted SDU has been acknowledged correctly by the UE
Type Definition	

```
SEQUENCE    {
        cellId              INTEGER(-1..63),
        routingInfo         RoutingInfo,
        mui                 Mui
}
```

7.3.2.2.35 RLC_UM_DATA

ASN.1 ASP Type Definition	
Type Name	RLC_UM_DATA_REQ
PCO Type	DSAP
Comment	To request to transmit DATA using unacknowledged mode.
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63),
            routingInfo             RoutingInfo,
            uM_Message               CHOICE {
                        dL_DCCH_Message             DL_DCCH_Message,
                        dL_CCCH_Message             DL_CCCH_Message,
                        pCCH_Message                PCCH_Message,
                        dL_SHCCH_Message            DL_SHCCH_Message,
                        bCCH_FACH_Message           BCCH_FACH_Message,
                        bCCH_BCH_Message            BCCH_BCH_Message,
                        invalid_dL_DCCH_Message     Invalid_DL_DCCH_Message,
                        invalid_dL_CCCH_Message     Invalid_DL_CCCH_Message,
                        invalid_dL_SHCCH_Message    Invalid_DL_SHCCH_Message}
}
```

ASN.1 ASP Type Definition	
Type Name	RLC_UM_DATA_IND
PCO Type	DSAP
Comment	To indicate to receive DATA using unacknowledged mode.
Type Definition	

```
SEQUENCE    {
            cellId                  INTEGER(-1..63),
            routingInfo             RoutingInfo,
            integrityResult         IntegrityResult,
            uM_Message               CHOICE {
                        uL_DCCH_Message             UL_DCCH_Message,
                        uL_CCCH_Message             UL_CCCH_Message,
                        uL_SHCCH_Message            UL_SHCCH_Message}
}
```

7.3.3 TTCN Primitives

7.3.3.1 UTRAN TTCN Primitives

Table 19 shows the primitives that are used for RLC, BMC ,RB and PDCP tests, these primitives are defined in TTCN tabular form.

Table 19: Primitives for RLC, BMC and RB tests

Primitive	Parameters	Use
RLC_TR_TestDataReq	Cell identity INTEGER (-31..32) Data (Meta type PDU)	The ASP is used to request the transmission of unstructured data using transparent mode in the downlink direction
RLC_TR_TestDataInd	Cell identity INTEGER (-31..32) Data (Meta type PDU)	The ASP is used to indicate the reception of unstructured data using transparent mode in the uplink direction
RLC_UM_TestDataReq	Cell identity INTEGER (-31..32) Data (Meta type PDU)	The ASP is used to request the transmission of unstructured data using unacknowledged mode in the downlink direction
RLC_UM_TestDataInd	Cell identity INTEGER (-31..32) Data (Meta type PDU)	The ASP is used to indicate the reception of unstructured data using unacknowledged mode in the uplink direction
RLC_AM_TestDataReq	Cell identity INTEGER (-31..32) Data (Meta type PDU)	The ASP is used to request the transmission of unstructured data using acknowledged mode in the downlink direction
RLC_AM_TestDataInd	Cell identity INTEGER (-31..32) Data (Meta type PDU)	The ASP is used to indicate the reception of unstructured data using acknowledged mode in the uplink direction
BMC_DataReq	Cell identity, INTEGER (-31..32), Data (Meta type PDU	The ASP is used to request the transmission of unstructured BMC data or scheduling message, using unacknowledged mode in the downlink direction.
BMC_DataCnf	CellId, INTEGER (-31..32)	The ASP is used to confirm the reception of BMC CBS data

The TTCN tabular format applies to the primitive definitions.

7.3.4 GERAN PCO and ASP definitions

7.3.4.1 PCO Type definitions

7.3.4.1.1 PCO type for data transmission and reception in GERAN

Table 20: Declaration of the G_DSAP PCO Type

PCO Type Definition	
PCO Type	G_DSAP
Role	LT
Comment	DATA transmission and reception

7.3.4.1.2 PCO type for configuration and control in GERAN

Table 21: Declaration of the G_CSAP PCO Type

PCO Type Definition	
PCO Type	G_CSAP
Role	LT
Comment	Transmission and reception of control primitives

7.3.4.2 PCO definitions

7.3.4.2.1 PCOs for data transmission and reception in GERAN

7.3.4.2.1.1 PCO for data transmission and reception through GERAN L2

Table 22: Declaration of G_L2 PCO

PCO Type Definition	
PCO Name	G_L2
PCO Type	G_DSAP
Role	LT
Comment	Control and observation point of GERAN L3 messages and user data

7.3.4.2.1.2 PCO for data transmission and reception through GPRS RLC

Table 23: Declaration of G_RLC PCO

PCO Type Definition	
PCO Name	G_RLC
PCO Type	G_DSAP
Role	LT
Comment	Control and observation point of GPRS GRR signalling messages

7.3.4.2.1.3 PCO for data transmission and reception through GPRS LLC

Table 24: Declaration of LLC PCO

PCO Type Definition	
PCO Name	G_LLC
PCO Type	G_DSAP
Role	LT
Comment	Control and observation point of GPRS GMM signalling messages

7.3.4.2.1.4 PCO for data transmission and reception through GPRS SNDCP

Table 25: Declaration of SNDCP PCO

PCO Type Definition	
PCO Name	G_SNDCP
PCO Type	G_DSAP
Role	LT
Comment	Control and observation point of GPRS user packet data

7.3.4.2.2 PCOs for control primitives transmission and reception in GERAN

7.3.4.2.2.1 PCO for GERAN L1control primitives transmission and reception

Table 26: Declaration of G_CL1 PCO

PCO Type Definition	
PCO Name	G_CL1
PCO Type	G_CSAP
Role	LT
Comment	Control GERAN Physical Layer (L1)

7.3.4.2.2.2 PCO for GERAN L2 control primitives transmission and reception

Table 27: Declaration of G_CL2 PCO

PCO Type Definition	
PCO Name	G_CL2
PCO Type	G_CSAP
Role	LT
Comment	Control GERAN L2

7.3.4.2.2.3 PCO for GPRS RLC control primitives transmission and reception

Table 28: Declaration of G_CRLC PCO

PCO Type Definition	
PCO Name	G_CRLC
PCO Type	G_CSAP
Role	LT
Comment	Control GPRS RLC/MAC layer

7.3.4.2.2.4 PCO for GPRS LLC control primitives transmission and reception

Table 29: Declaration of G_CLLC PCO

PCO Type Definition	
PCO Name	G_CLLC
PCO Type	G_CSAP
Role	LT
Comment	Control GPRS LLC layer

7.3.4.2.2.5 PCO for GPRS SNDCP control primitives transmission and reception

Table 30: Declaration of G_CSNDCP PCO

PCO Type Definition	
PCO Name	G_CSNDCP
PCO Type	G_CSAP
Role	LT
Comment	Control GPRS SNDCP layer

7.3.4.3 GERAN ASP Definitions

7.3.4.3.1 ASPs for data transmission and reception in GERAN

7.3.4.3.1.1 ASPs for data transmission and reception through GERAN L2

ASP Name	G_L2_DATA_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send L3 signalling message on the signalling channels or user data on the traffic channels to the UE/MS in acknowledged mode.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
sAPI	SAPI	0
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
rfn	RFN	The reduced frame number of the first frame on which this message is sent. This field is not applicable and the SS shall ignore it if the field t2 of rfn is coded as '11111'B.
msg	PDU	Signalling message or user data to be sent
Detailed Comments	Parameter fn is only used in the test cases that require specific L3 message to be sent on specified frame number.	

ASP Name	G_L2_DATA_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive a L3 signalling message on the signalling channels or user data on the traffic channels from the UE/MS in acknowledged mode.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
sAPI	SAPI	0 or 3
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
rfn	RFN	The reduced frame number of the first frame carrying the message
msg	PDU	Signalling message or user data received
Detailed Comments		

ASP Name	G_L2_L2Estab_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive an indication of that L2 multiple frame operation on the specified channel has been established.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4, This field shall be coded as 15 if it is not applicable.
sAPI	SAPI	0,3
establish_mode	OCTETSTRING[1]	
rfn	RFN	The reduced frame number of the first frame carries the L2 SABM frame
msg	PDU	this field is present only when the establidg mode is CoRes (collision resolution)
Detailed Comments	see 3GPP TS 44.006 clause 7.1.1 and 7.1.3	

ASP Name	G_L2_UNITDATA_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send L3 signalling message on the signalling channels or send user data on the traffic channels to the UE/MS in unacknowledged mode.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
sAPI	SAPI	0
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
rfn	RFN	The reduced frame number of the first frame on which this message is sent. This field is not applicable and the SS shall ignore it if the field t2 of rfn is coded as '11111'B.
msg	PDU	Signalling message or user data to be sent
Detailed Comments	Parameter fn is only used in the test cases that require specific L3 message to be sent on specified frame number.	

ASP Name	G_L2_UNITDATA_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive a L3 signalling message on the signalling channels or user data on the traffic channels from the UE/MS in unacknowledged mode.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
sAPI	SAPI	0 or 3
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
rfn	RFN	The reduced frame number of the first frame carrying the message
msg	PDU	Signalling message or user data received
Detailed Comments		

ASP Name	G_L2_ACCESS_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive a random access or handover access burst on the specified channel.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	RACH, FACCH, SDCCH/8, SDCCH/4. RACH is used for random access burst; others are used for handover access burst
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8, SDCCH/4. This field is not applicable and the SS shall ignore it if this field is coded as 15.
rfn	RFN	The reduced frame number of the first frame carrying the burst
burst	PDU	Random access burst or handover access burst
Detailed Comments		

ASP Name	G_L2_Paging_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send a paging message on the specified paging group of the specified paging channel to the UE/MS.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
sAPI	SAPI	0
physicalChId	PhysicalChId	Channel identifier of the right CCCH_GROUP or PCCCH_GROUP
g_LogicChType	G_LogicChType	PCH or PPCH
pagingGroup	PAGING_GROUP	
pagingMode	PagingMode	0—normal paging; 1—extended paging; 2—paging reorganization.
msg	PDU	Paging message
Detailed Comments	The SS is required to send valid layer 3 messages continuously on all paging subchannels on CCCH and is required to send valid RLC data blocks or RLC/MAC control blocks continuously on all subchannels on PCCCH where paging can appear. For "normal paging" the SS send the paging message in the specified pagingGroup; For "extended paging" " the SS send the paging message in the specified pagingGroup and in the "next but one" position on the PCH or in the third block period on PCCCH where paging may occur (PPCH), following the block corresponding to pagingGroup; For "paging reorganization" the SS send the paging message in all paging subchannels.	

Type Name	CellId
Type Definition	INTEGER
Type Encoding	
Comments	

Type Name	SAPI
Type Definition	INTEGER
Type Encoding	
Comments	Service access point identifier for GERAN L2 and LLC

Type Name	PhysicalChId
Type Definition	INTEGER(0..31)
Type Encoding	
Comments	Physical channel identifier in GERAN

Type Name	G_LogicChType
Type Definition	INTEGER
Type Encoding	
Comments	GERAN logical channel type: 0—BCCH; 1—RACH; 2—PCH; 3—AGCH; 4—SDCCH/4; 5—SACCH/C4; 6—SDCCH/8; 7—SACCH/C8; 8—TCH/F; 9—FACCH/F; 10—SACCH/TF; 11—TCH/H; 12—FACCH/H; 13—SACCH/TH; 14—PBCCH; 15—PRACH; 16—PPCH; 17—PAGCH; 18—PDTCH/F; 19—PACCH/F; 20—PTCCH/F; 21—E-TCH/F; 22—E-IACCH/F; 23—E-FACCH/F; 24—SACCH/M; 25—SACCH/MD

Type Name	SubChannelNumber
Type Definition	INTEGER
Type Encoding	
Comments	Subchannel number for TCH/H, FACCH/H, SACCH/TH, SDCCH/4, SDCCH/C4, SDCCH/8 and SDCCH/C8. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); For SDCCH/4 and SACCH/C4 value is (0..3).

Type Name	PAGING_GROUP
Type Definition	INTEGER
Type Encoding	
Comments	3GPP TS 45.002 [31] clauses 6.5.2 and 6.5.6

Type Name	PagingMode
Type Definition	INTEGER
Type Encoding	
Comments	0 – normal paging; 1 – extended paging; 2 – paging reorganization.

Type Name	RFN		
Encoding Variation			
Comments	The reduced frame number, its range is 0 -- 42431 (FN modulo 42432) about 195.8 s		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
t1_	BITSTRING[5]		(FN div 1326) mod 32
t2	BITSTRING[5]		FN mod 26
t3	BITSTRING[6]		FN mod 51
Detailed Comments	see 3GPP TS 44.018 clause 10.5.38. The reduced frame number, FN modulo 42432 can be calculated in the following formula: 51 * ((t3 - t2) mod 26) + t3 + 1326 * t1_. RFN is used for starting time and TBF starting time.		

ASP Name	G_L2_SYSINFO_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send system information messages to the lower layer emulator.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
sAPI	SAPI	0
physicalChId	PhysicalChId	
g_LogicChType	G_LogicChType	BCCH or SACCH
instanceIndex	INTEGER	To indicate the instance of the system information messages. For SYSTEM INFORMATION Type 2ter, 18, 19, 20 the value is (0..7); for type 14, 15 the value is (0..3); for type 2quater the value is (0..15); for all other type the value is 0.
sysInfoType	SysInfoType	SYSTEM INFORMATION Type 5, 5bis, 5ter, and 6 are sent on SACCH, the other SYSTEM INFORMATION 's are sent on BCCH.
msg	PDU	This field contains SYSTEM INFORMATION message. See 3GPP TS 44.018 clause 9.1.31 to clause 9.1.43h for SYSTEM INFORMATION message definitions.
Detailed Comments	The lower layer emulator shall store the SYSTEM INFORMATION's, and transmit them periodically according to the rules specified in clause 6.3.1.3 of 3GPP TS 45.002 [31]. The msg shall override the same type system information message previous stored in the lower layer emulator.	

Type Name	SysInfoType
Type Definition	INTEGER
Type Encoding	
Comments	25--SYSTEM INFORMATION TYPE 1 26--SYSTEM INFORMATION TYPE 2 2 -- SYSTEM INFORMATION TYPE 2bis 3 -- SYSTEM INFORMATION TYPE 2ter 7 -- SYSTEM INFORMATION TYPE 2quater 27--SYSTEM INFORMATION TYPE 3 28--SYSTEM INFORMATION TYPE 4 29--SYSTEM INFORMATION TYPE 5 5 -- SYSTEM INFORMATION TYPE 5bis 6 -- SYSTEM INFORMATION TYPE 5ter 30--SYSTEM INFORMATION TYPE 6 31--SYSTEM INFORMATION TYPE 7 24--SYSTEM INFORMATION TYPE 8 4 -- SYSTEM INFORMATION TYPE 9 0 -- SYSTEM INFORMATION TYPE 13 61--SYSTEM INFORMATION TYPE 16 62--SYSTEM INFORMATION TYPE 17 64--SYSTEM INFORMATION TYPE 18 65--SYSTEM INFORMATION TYPE 19 66--SYSTEM INFORMATION TYPE 20

7.3.4.3.1.2 ASPs for data transmission and reception through GERAN RLC

ASP Name	G_RLC_PSI_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send packet system information messages to the lower layer emulator.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	
g_LogicChType	G_LogicChType	PBCCH or PACCH or PCCCH
packetSysInfoCategory	PSI_Category	PSI1 or high repetition rate or low repetition rate. Type of this field is INTEGER: 0-- PSI1; 1--high repetition category; 2--low repetition category.
positionInList	PositionInList	Position in the high repetition rate list or the low repetition rate list, for PSI1 this field is not applicable and set to 31. Type of this field is INTEGER, the order of the position is from 0, 1, ... 0 indicates the first position, 1 the second, and so on.
msg	PDU	This field contains PACKET SYSTEM INFORMATION message, see 3GPP TS 44.060 [32] clause 11.2.18 to clause 11.2.25 for the message definitions
Detailed Comments	On PBCCH, the lower layer emulator shall store the PACKET SYSTEM INFORMATION's, and transmit them periodically according to the rules specified in clause 6.3.2.4 of 3GPP TS 45.002 [31]. The msg shall override the same type packet system information message previous stored in the lower layer. Multiple instances of a PSI shall be put in the same list and in ascending order of the message instance number	

Type Name	PSI_Category
Type Definition	INTEGER
Type Encoding	
Comments	3GPP TS 45.002 [31] clause 6.3.2.4

Type Name	PositionInList
Type Definition	INTEGER
Type Encoding	
Comments	0 is the first position; 1 is the second, and so on.

ASP Name	G_RLC_ControlMsg_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to transmit a RLC/MAC control message to the UE/MS on the specified channel.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Valid for PCCCH only
g_LogicChType	G_LogicChType	PCCCH or PACCH or PTCCH
tBF_Direction	INTEGER	0—downlink; 1--uplink
tFI	TFI	Temporary flow identity
rRBP	RRBP	Relative reserved block period
s_P_Bit	S_P_Bit	Supplementary/polling bit
rfn	RFN	The reduced frame number of the first frame on which this message is sent. This field is not applicable and the SS shall ignore it if the field t2 of rfn is coded as '11111'B.
msg	PDU	Down link RLC/MAC control message
Detailed Comments	PTCCH is valid for PACKET TIMING ADVANCE/POWER CONTROL message only	

Type Name	RRBP
Type Definition	BITSTRING[2]
Type Encoding	
Comments	3GPP TS 44.060 [32] clause 10.4.5

Type Name	S_P_Bit
Type Definition	BITSTRING[1]
Type Encoding	
Comments	0 – RRBP field is not valid; 1 – RRBP field is valid.

ASP Name	G_RLC_ControlMsg_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive an uplink RLC/MAC control block sent by the UE/MS on the specified channel.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	
g_LogicChType	G_LogicChType	PACCH or PDTCH
tBF_Direction	INTEGER	0--downlink; 1--uplink
tFI	TFI	Temporary flow identity
retryBit	BITSTRING[1]	For access bursts on PRACH, RACH and PACCH, this field is no meaning
rfn	RFN	The reduced frame number of the frame carrying the message
msg	PDU	Uplink RLC/MAC control message
Detailed Comments	Logical channel type PDTCH is valid for PACKET ENHANCED MEARSUREMENT REPORT message only.	

ASP Name	G_RLC_ACCESS_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive an access burst sent by the UE/MS on the specified channel.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	
g_LogicChType	G_LogicChType	PRACH or PACCH or PTCCH
rfn	RFN	The reduced frame number of the frame carrying the burst
burst	PDU	8-bit or 11-bit access burst
Detailed Comments	PACKET CHANNEL REQUEST, EGPRS PACKET CHANNEL REQUEST and burst format of PACKET CONTROL ACKNOWLEDGEMENT are access bursts.	

7.3.4.3.1.3 ASPs for data transmission and reception through GERAN LLC

ASP Name	G_LLC_UNITDATA_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send L3 PDU to the UE/MS in LLC unconfirmed transmission.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
tLLI	TLLI	
sAPI	SAPI	
protectMode	BITSTRING[1]	0 -- unprotected; 1 -- protected
cipherMode	BITSTRING[1]	0 -- no encryption; 1 -- encrypted
msg	PDU	L3 PDU
Detailed Comments	3GPP TS 44.064 [33] clause 8.4.1	

ASP Name	G_LLC_UNITDATA_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive a L3 PDU from the UE/MS in LLC unconfirmed transmission.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
tLLI	TLLI	
sAPI	SAPI	
msg	PDU	L3 PDU
Detailed Comments	3GPP TS 44.064 [33] clause 8.4.2	

7.3.4.3.1.4 ASPs for data transmission and reception through GERAN SNDCP

ASP Name	G_SN_DATA_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send a valid IP datagram on the specified NSAPI to the UE/MS by acknowledged transmission.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
nSAPI	NSAPI	5-15
n_PDU_Number	N_PDU_Number	
n_PDU	N_PDU	Valid IPv4 or IPv6 datagram
Detailed Comments	Acknowledged transmission mode	

ASP Name	G_SN_DATA_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive an IP datagram on the specified NASPI from the UE/MS in acknowledged transmission mode.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
nSAPI	NSAPI	5-15
n_PDU	N_PDU	IPv4 or IPv6 datagram
Detailed Comments	Acknowledged transmission mode	

ASP Name	G_SN_UNIDATA_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send a valid IP datagram on the specified NSAPI to the UE/MS by unacknowledged transmission.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
nSAPI	NSAPI	5-15
n_PDU	N_PDU	Valid IPv4 or IPv6 datagram
Detailed Comments	Unacknowledged transmission mode	

ASP Name	G_SN_UNITDATA_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive an IP datagram on the specified NASPI from the UE/MS in unacknowledged transmission mode.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
nSAPI	NSAPI	5-15
n_PDU	N_PDU	IPv4 or IPv6 datagram
Detailed Comments	Unacknowledged transmission mode	

ASP Name	G_SN_XID_REQ	
PCO Type	G_DSAP	
Comments	The ASP is used to send the requested XID parameters to the UE/MS.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
xID_Info	XID_Info	XID parameters requested
Detailed Comments		

ASP Name	G_SN_XID_IND	
PCO Type	G_DSAP	
Comments	The ASP is used to receive the XID parameters requested by the UE/MS.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
xID_Info	XID_Info	XID parameters requested by the UE/MS
Detailed Comments		

ASP Name	G_SN_XID_CNF	
PCO Type	G_DSAP	
Comments	The ASP is used to receive the negotiated XID parameters agreed by the UE/MS.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
xID_Info	XID_Info	The negotiated XID parameters agreed by the UE/MS
Detailed Comments		

ASP Name	G_SN_XID_RES	
PCO Type	G_DSAP	
Comments	The ASP sends to the UE/MS the negotiated XID parameters agreed by the SS.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
xID_Info	XID_Info	The negotiated XID parameters agreed by the SS
Detailed Comments		

7.3.4.3.2 ASPs for control primitive transmission and reception in GERAN

7.3.4.3.2.1 ASPs for configuration and control of GERAN L1

ASP Name	G_CL1_CreateCell_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to create a cell in GERAN	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
baseId	BITSTRING[6]	base transceiver station identity code = NCC+BCC. see 3GPP TS 23.003 [6]
Detailed Comments		

ASP Name	G_CL1_CreateCell_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CL1_CreateCell_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell created
Detailed Comments		

ASP Name	G_CL1_DeleteCell_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to delete a cell in GERAN	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell to be deleted
Detailed Comments		

ASP Name	G_CL1_DeleteCell_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a CG_L1_DeleteCell_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell deleted
Detailed Comments		

ASP Name	G_CL1_CreateBasicPhyCh_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to create a basic physical channel in GERAN	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell which the channel to be created belongs to
physicalChId	PhysicalChId	identifier of the physical channel in the SS.
channelCombination	ChannelCombination	Logical channels combined onto the basic physical channel.
frqInfo	FrqInfo	Parameters for Description of the physical channel in frequency domain
timeSlot	TN	The timeslot number of the physical channel
tsc	TSC	Training sequence code. For common control and broadcast channels the value of tsc must be equal to BCC (base station colour code)
channelSpecificInfo	ChannelSpecificInfo	Specific parameters related to individual channel
txPower	TX_Power	The transmission power level in dBµVemf()
Detailed Comments	The value of channelCombination permitted currently: 1 TCH/F + FACCH/F + SACCH/TF 2 TCH/H(0,1) + FACCH/H(0,1) + SACCH/TH(0,1) 3 TCH/H(0,0) + FACCH/H(0,1) + SACCH/TH(0,1) + TCH/H(1,1) 4 FCCH + SCH + BCCH + CCCH 5 FCCH + SCH + BCCH + CCCH + SDCCH/4(0..3) + SACCH/C4(0..3) 6 BCCH + CCCH 7 SDCCH/8(0..7) + SACCH/C8(0.. 7) 8 TCH/F + FACCH/F + SACCH/M 9 TCH/F + SACCH/M 10 TCH/FD + SACCH/MD 11 PBCCH+PCCCH+PDTCH/F+PACCH/F+PTCCH/F 12 PCCCH+PDTCH/F+PACCH/F+PTCCH/F 13 PDTCH/F+PACCH/F+PTCCH/F 18 E-TCH/F + E-IACCH/F + E-FACCH/F + SACCH/TF 19 E-TCH/F + E-IACCH/F + E-FACCH/F + SACCH/M 20 E-TCH/F + E-IACCH/F + SACCH/M 21 E-TCH/FD + E-IACCH/F + SACCH/MD	

ASP Name	G_CL1_CreateBasicPhyCh_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a CG_L1_CreateBasicPhyCh_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell which the created channel belongs to
physicalChId	PhysicalChId	The physical channel created.
Detailed Comments		

Type Name	FrqInfo		
Encoding Variation			
Comments	Parameters for Description of basic physical channel in frequency domain.		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
h	BITSTRING[1]		h=1:hopping channel h=0: non-hopping channel
spr	BITSTRING [3]		'000'B
spr1	BITSTRING [2]		'00'B if h = 0, otherwise OMIT
maio	BITSTRING [6]		mobile allocation index offset if h = 1, otherwise OMIT
hsn	BITSTRING [6]		hopping sequence number if h = 1, otherwise OMIT
arfcn	BITSTRING [10]		absolute RF channel number if h = 0, otherwise OMIT
hoppingFreqList	FrequencyList		hopping frequency list if h = 1, otherwise OMIT. The definition see 3GPP TS 44.018 clause 10.5.2.13
Detailed Comments			

Type Name	ChannelSpecificInfo		
Encoding Variation			
Comments	Parameters for individual channel		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
presence	BITSTRING[4]		4 bits field indicating which fields below are presented in the constraint of this structured type. B3 = 1indicating dedCh_Info presence, B2 = 1 indicating cCCH_Info presence, B1 = 1 indicating pCCCH_Info presence, B0 = 1 indicading pBCCH_Info presence.
dedCH_Info	DedCH_Info		Parameters for dedicated channel. Valid for combination:1, 2, 3, 5, 7, 8, 9, 10
cCCH_Info	CCCH_Info		Parameters for common control channels: PCH, SCH,… Valid for combination: 4, 5, 6
pCCCH_Info	PCCCH_Info		Parameters for packet common control channels: PCCCH, PPCH,… Valid for combination: 11, 12
pBCCH_Info	PBCCH_Info		Parameters for packet broadcast channels: PBCCH Valid for combination: 11
Detailed Comments			

Type Name	DedCH_Info		
Encoding Variation			
Comments	Parameters for dedicated channel		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
chMod	CHMOD		Definition see 3GPP TS 44.018 clause 10.5.2.6
cipherMode	CPHMS		Definition see 3GPP TS 44.018 clause 10.5.2.9
cipherKey	BITSTRING[64]		
Detailed Comments			

Type Name	CCCH_Info		
Encoding Variation			
Comments	Parameters for common control channels		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
bS_PA_MFRMS	BITSTRING[3]		the number of 51-multiframes between transmissions of paging messages. Definition see 3GPP TS 44.018 clause 10.5.2.11
bS_AG_BLKS_RES	BITSTRING[3]		the number of blocks on each common control channel reserved for access grant messages. Definition see 3GPP TS 44.018 clause 10.5.2.11
splitOnCCCH	BITSTRING[1]		0 -- no split pa cycle on CCCH; 1—split pg cycle on CCCH 3GPP TS 45.002 [31] clause 6.5.6
Detailed Comments			

Type Name	PCCCH_Info		
Encoding Variation			
Comments	Parameters for packet common control channels		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
bS_PBCCH_BLKS	BITSTRING[2]		3GPP TS 44.060 [32] clause 12.25
bS_PAG_BLKS_RES	BITSTRING[4]		3GPP TS 44.060 [32] clause 12.25
bS_PRACH_BLKS	BITSTRING[4]		3GPP TS 44.060 [32] clause 12.25
Detailed Comments			

Type Name	PBCCH_Info		
Encoding Variation			
Comments	Parameters for packet broadcast channel		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
pSI1_REPEAT_PERIOD	PSI1_REPEAT_PERIOD		The repeat period of packet system information Type 1. See 3GPP TS 44.060 [32] clause 11.2.18
pSI_COUNT_HR	PSI_COUNT_HR		The number of PSI message instances sent with high repetition rate. See 3GPP TS 44.060 [32] clause 11.2.18
pSI_COUNT_LR	PSI_COUNT_LR		The number of PSI message instances sent with low repetition rate. See 3GPP TS 44.060 [32] clause 11.2.18
Detailed Comments			

ASP Name	G_CL1_CreateMultiSlotConfig_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to create an multi-slot configuration in GERAN	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell which the configuration to be created belongs to
mainChannel	PhysicalChId	identifier of the main physical channel of this multi-slot configuration.
multiSlotAllocation	MultiSlotAllocation	The timeslot allocation of the configuration
Detailed Comments	This ASP is to create an multi-slot configuration with combination of TCH/F+FACCH/F+SACCH/M, TCH/F+SACCH/M and TCH/FD+SACCH/MD or combination of E-TCH/F+E-IACCH/F+E-FACCH/F+SACCH/M, E-TCH/F+E-IACCH/F+SACCH/M and E-TCH/FD+E-IACCH/F+SACCH/MD	

ASP Name	G_CL1_CreateMoultiSlotConfig_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a CG_L1_CreateMultiSlotConfig_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The cell which the created multi-slot configuration belongs to
mainChannel	PhysicalChId	The main channel identifier.
Detailed Comments		

Type Name	MultiSlotAllocation		
Encoding Variation			
Comments	Used in multi-slot configuration		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
tN0	BOOLEAN		TRUE – time slot 0 is allocated; FALSE -- not allocated
physicalChId0	PhysicalChId		Physical channel of time slot 0; not applicable if tN0 = FALSE
tN1	BOOLEAN		TRUE – time slot 1 is allocated; FALSE -- not allocated
physicalChId1	PhysicalChId		Physical channel of time slot 1; not applicable if tN1 = FALSE
tN2	BOOLEAN		TRUE – time slot 2 is allocated; FALSE -- not allocated
physicalChId2	PhysicalChId		Physical channel of time slot 2; not applicable if tN2 = FALSE
tN3	BOOLEAN		TRUE – time slot 3 is allocated; FALSE -- not allocated
physicalChId3	PhysicalChId		Physical channel of time slot 3; not applicable if tN3 = FALSE
tN4	BOOLEAN		TRUE – time slot 4 is allocated; FALSE -- not allocated
physicalChId4	PhysicalChId		Physical channel of time slot 4; not applicable if tN4 = FALSE
tN5	BOOLEAN		TRUE – time slot 5 is allocated; FALSE -- not allocated
physicalChId5	PhysicalChId		Physical channel of time slot 5; not applicable if tN5 = FALSE
tN6	BOOLEAN		TRUE – time slot 6 is allocated; FALSE -- not allocated
physicalChId6	PhysicalChId		Physical channel of time slot 6; not applicable if tN6 = FALSE
tN7	BOOLEAN		TRUE – time slot 7 is allocated; FALSE -- not allocated
physicalChId7	PhysicalChId		Physical channel of time slot 7; not applicable if tN7 = FALSE
Detailed Comments			

ASP Name	G_CL1_ComingFN_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to request lower layer return the reduced frame number (FN modulo 42432) which is far enough in the future from current frame number and is able to carry L3 message on the specified channel. The requirement of "far enough" is that there is enough time left for TTCN to prepare a L3 message to send before that frame.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL1_ComingFN_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to receive the result of G_CL1_ComingFN_REQ.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
rfn	RFN	the reduced frame number (FN modulo 42432) which is about 5 seconds later than current frame number and is able to carry L3 message on the channel specified by "physicalChId"+"G_LogicChType"+"subChannel"
Detailed Comments		

ASP Name	G_CL1_L1Header_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to request lower layer return the L1 header of SACCH.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	SACCH
subChannel	SubChannelNumber	Valid only for logical channel types: SACCH/TH, SACCH/C8, and SACCH/C4 This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL1_L1Header_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to receive the result of G_CL1_L1Header_REQ.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	SACCH
subChannel	SubChannelNumber	Valid only for logical channel types: SACCH/TH, SACCH/C8, and SACCH/C4 This field is not applicable and the SS shall ignore it if this field is coded as 15.
l1Header	L1HD	Power level and timing advance
Detailed Comments		

ASP Name	G_CL1_DeleteChannel_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to delete a basic physical channel or an multi-slot configuration	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell which the channel to be deleted belongs to
physicalChId	PhysicalChId	The physical channel or the multi-slot configuration to be deleted.
Detailed Comments		

ASP Name	G_CL1_DeleteChannel_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CL1_DeleteChannel_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell which the deleted channel belongs to
physicalChId	PhysicalChId	The physical channel or multi-slot configuration deleted.
Detailed Comments		

ASP Name	G_CL1_ChModeModify_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to modify the channel mode of a dedicated channel	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
chMode	CHMOD	Definition see 3GPP TS 44.018 clause 10.5.2.1b
Detailed Comments		

ASP Name	G_CL1_ChModeModify_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CL1_ChModeModify_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL1_SetNewKey_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to set new cipher key for a dedicated channel	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	The channel which uses the new key
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
cipherKey	BITSTRING[64]	
Detailed Comments		

ASP Name	G_CL1_SetNewKey_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CL1_SetNewKey_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL1_CipherModeModify_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to modify cipher mode of a dedicated channel	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
cipherMode	CPHMS	The new cipher mode. Definition see 3GPP TS 44.018 clause 10.5.2.9
Detailed Comments		

ASP Name	G_CL1_CipherModeModify_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CL1_CipherModeModify_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: TCH/H, FACCH/H, SACCH/TH, SDCCH/8, SACCH/C8, SDCCH/4, and SACCH/C4. For TCH/H, FACCH/H and SACCH/TH value is (0..1); For SDCCH/8 and SACCH/C8 value is (0..7); for SDCCH/4 and SACCH/C4 value is (0..3). This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL1_ChangePowerLevel_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to change the transmission power level of a physical channel	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell which the physical channel belongs to
physicalChId	PhysicalChId	Channel using the new transmission power level
txPower	TX_Power	The new transmission power level in dBµVemf()
Detailed Comments		

ASP Name	G_CL1_ChangePowerLevel_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CL1_ChangePowerLevel_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
physicalChId	PhysicalChId	The physical channel which uses the new transmission power level
Detailed Comments		

7.3.4.3.2.2 ASPs for configuration and control of GERAN L2

ASP Name	G_CL2_HoldPhyInfo_REQ	
PCO Type	G_CSAP	
Comments	The ASP commands the SS to hold the PHYSICAL INFORMATION message, which will be sent on PCO G_L2 following the current ASP. The PHYSICAL INFORMATION message shall be sent to the UE/MS within T3124 from the time when the SS has received n handover access bursts.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4, This field is not applicable and the SS shall ignore it if this field is coded as 15.
n	INTEGER	The number of handover access bursts to be received
Detailed Comments	T3124 is defined in 3GPP TS 44.018 clause 3.4.4.2.2 and clause 11.1.1	

ASP Name	G_CL2_HoldPhyInfo_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get a confirmation of the G_CL2_HoldPhyInfo_REQ.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4. This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL2_NoUAforSABM_REQ	
PCO Type	G_CSAP	
Comments	The ASP commands the SS not to send UA response to the UE when it receives SABM from the UE on the specified channel.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4, This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL2_NoUAforSABM_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get a confirmation of the G_CL2_NoUAforSABM_REQ.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4. This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL2_ResumeUAforSABM_REQ	
PCO Type	G_CSAP	
Comments	The ASP commands the SS to send UA response to the UE when it receives SABM from the UE on the specified channel. This ASP is used after G_CL2_NoUAforSABM_REQ to resume the normal multiframe operation of L2	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4, This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

ASP Name	G_CL2_ResumeUAforSABM_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get a confirmation of the G_CL2_ResumeUAforSABM_REQ.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
physicalChId	PhysicalChId	Channel identifier
g_LogicChType	G_LogicChType	
subChannel	SubChannelNumber	Valid only for logical channel types: FACCH/H, SDCCH/8 and SDCCH/4. This field is not applicable and the SS shall ignore it if this field is coded as 15.
Detailed Comments		

7.3.4.3.2.3 ASPs for configuration and control of GERAN RLC/MAC

ASP Name	G_CRLC_UL_TBF_Config_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to configure a TBF used for uplink packet data transfer	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
tFI	TFI	
tBF_Mode	BITSTRING[1]	0 – GPRS; 1 – EGPRS
channelCoding	ChannelCoding	
tLLI_BlockChannelCoding	BITSTRING[1]	0 – CS-1 or MCS-1(EGPRS); 1 – same as channelCoding
rLC_Mode	BITSTRING[1]	0 – acknowledged mode; 1 – unacknowledged mode
startingTime	RFN	This field is not applicable and the SS shall ignore it if the field t2 of rfn is coded as '11111'B.
resourceAllocation	ResourceAllocation	Fixed, dynamic or single allocation and other parameters.
Detailed Comments	For GPRS channel coding can be: CS-1, CS-2, CS-3 and CS-4; For EGPRS channel coding can be : MCS-1, MCS-2, MCS-3, MCS-4, MCS-5, MCS-6, MCS-7, MCS-8, MCS-9, MCS-5-7 and MCS-6-9.	

ASP Name	G_CRLC_UL_TBF_Config_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CRLC_UL_TBF_Config_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
tFI	TFI	
Detailed Comments		

Type Name	ChannelCoding
Type Definition	INTEGER
Type Encoding	
Comments	1 – CS-1; 2 – CS-2; 3 – CS-3; 4 -- CS-4; 5 – MCS-1; 6 – MCS-2; 7 – MCS-3; 8 – MCS-4; 9 – MCS-5; 10 – MCS-6; 11 – MCS-7; 12 – MCS-8; 13 – MCS-9; 14 – MCS-5-7; 15 – MCS-6-9

Type Name	ResourceAllocation		
Encoding Variation			
Comments	Used for up link TBF		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
dynamicAllocation	DynamicAllocation		Dynamic allocation or extended dynamic allocation
fixedAllocation	FixedAllocation		
singleBlockAllocation	SingleBlockAllocation		
Detailed Comments			

Type Name	DynamicAllocation		
Encoding Variation			
Comments	Used for up link TBF; dynamic allocation or extended dynamic allocation		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
extendedAllocation	BITSTRING[1]		0 – dynamic allocation; 1 – extended dynamic allocation
uSFGranularity	BITSTRING[1]		0 – one block; 1 – four blocks
tN0	BOOLEAN		TRUE – time slot 0 is allocated; FALSE -- not allocated
uSF_TN0	BITSTRING[3]		USF value for slot 0
physicalChId0	PhysicalChId		Physical channel of timeslot 0; not applicable if tN0 = FALSE
tN1	BOOLEAN		TRUE – time slot 1 is allocated; FALSE -- not allocated
uSF_TN1	BITSTRING[3]		USF value for slot 1
physicalChId1	PhysicalChId		Physical channel of timeslot 1; not applicable if tN1 = FALSE
tN2	BOOLEAN		TRUE – time slot 2 is allocated; FALSE -- not allocated
uSF_TN2	BITSTRING[3]		USF value for slot 2
physicalChId2	PhysicalChId		Physical channel of timeslot 2; not applicable if tN2 = FALSE

Type Name	DynamicAllocation		
Encoding Variation			
Comments	Used for up link TBF; dynamic allocation or extended dynamic allocation		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
tN3	BOOLEAN		TRUE – time slot 3 is allocated; FALSE -- not allocated
uSF_TN3	BITSTRING[3]		USF value for slot 3
physicalChId3	PhysicalChId		Physical channel of timeslot 3; not applicable if tN3 = FALSE
tN4	BOOLEAN		TRUE – time slot 4 is allocated; FALSE -- not allocated
uSF_TN4	BITSTRING[3]		USF value for slot 4
physicalChId4	PhysicalChId		Physical channel of timeslot 4; not applicable if tN4 = FALSE
tN5	BOOLEAN		TRUE – time slot 5 is allocated; FALSE -- not allocated
uSF_TN5	BITSTRING[3]		USF value for slot 5
physicalChId5	PhysicalChId		Physical channel of timeslot 5; not applicable if tN5 = FALSE
tN6	BOOLEAN		TRUE – time slot 6 is allocated; FALSE -- not allocated
uSF_TN6	BITSTRING[3]		USF value for slot 6
physicalChId6	PhysicalChId		Physical channel of timeslot 6; not applicable if tN6 = FALSE
tN7	BOOLEAN		TRUE – time slot 7 is allocated; FALSE -- not allocated
uSF_TN7	BITSTRING[3]		USF value for slot 7
physicalChId7	PhysicalChId		Physical channel of timeslot 7; not applicable if tN7 = FALSE
Detailed Comments	The uSF_TNx field is not applicable when tNx = FALSE.		

Type Name	FixedAllocation		
Encoding Variation			
Comments	Used for up link TBF		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
downlinkControlSlot	BITSTRING[3]		Time slot for downlink control messages
timeSlotAllocation	TimeSlotAllocation		
blocksOrBlockPeriods	BITSTRING[1]		0 -- blocks; 1 -- block periods
allocationBitMap	BITSTRING		See 3GPP TS 44.060 [32] clause 12.4
Detailed Comments			

Type Name	SingleBlockAllocation		
Encoding Variation			
Comments	Used for up link TBF		
Element Name	Type Definition	Field Encoding	Comments
physicalChId	PhysicalChId		The physical channel of the allocated block
Detailed Comments	Time slot number is implicitly indicated by the physical channel identifier.		

ASP Name	G_CRLC_DL_TBF_Config_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to configure a TBF used for down link packet data transfer	
Parameter Name	Parameter Type	Comments
cellId	CellId	
tFI	TFI	
tBF_Mode	BITSTRING[1]	0 – GPRS; 1 – EGPRS
channelCoding	ChannelCoding	
rLC_Mode	BITSTRING[1]	0 – acknowledged mode; 1 – unacknowledged mode
timeSlotAllocation	TimeSlotAllocation	Downlink TBF time slot allocation
startingTime	RFN	This field is not applicable and the SS shall ignore it if the field t2 of rfn is coded as '11111'B.
Detailed Comments	For GPRS channel coding can be: CS-1, CS-2, CS-3 and CS-4; For EGPRS channel coding can be : MCS-1, MCS-2, MCS-3, MCS-4, MCS-5, MCS-6, MCS-7, MCS-8, MCS-9, MCS-5-7 and MCS-6-9.	

ASP Name	G_CRLC_DL_TBF_Config_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CRLC_DL_TBF_Config_REQ	
Parameter Name	Parameter Type	Comments
cellId	CellId	
tFI	TFI	
Detailed Comments		

Type Name	TimeSlotAllocation		
Encoding Variation			
Comments	Used for downlink and up link TBF		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
tN0	BOOLEAN		Timeslot 0; TRUE— allocated; FALSE— not allocated.
physicalChId0	PhysicalChId		Physical channel of timeslot 0; not applicable if tN0 = FALSE
tN1	BOOLEAN		Timeslot 1; TRUE — allocated; FALSE— not allocated.
physicalChId1	PhysicalChId		Physical channel of timeslot 1; not applicable if tN1 = FALSE
tN2	BOOLEAN		Timeslot 2; TRUE— allocated; FALSE— not allocated.
physicalChId2	PhysicalChId		Physical channel of timeslot 2; not applicable if tN2 = FALSE
tN3	BOOLEAN		Timeslot 3; TRUE— allocated; FALSE— not allocated.
physicalChId3	PhysicalChId		Physical channel of timeslot 3; not applicable if tN3 = FALSE
tN4	BOOLEAN		Timeslot 4; TRUE— allocated; FALSE— not allocated.
physicalChId4	PhysicalChId		Physical channel of timeslot 4; not applicable if tN4 = FALSE
tN5	BOOLEAN		Timeslot 5; TRUE— allocated; FALSE— not allocated.
physicalChId5	PhysicalChId		Physical channel of timeslot 5; not applicable if tN5 = FALSE
tN6	BOOLEAN		Timeslot 6; TRUE— allocated; FALSE— not allocated.
physicalChId6	PhysicalChId		Physical channel of timeslot 6; not applicable if tN6 = FALSE
tN7	BOOLEAN		Timeslot 7; TRUE— allocated; FALSE— not allocated.
physicalChId7	PhysicalChId		Physical channel of timeslot 7; not applicable if tN7 = FALSE
Detailed Comments			

Declaration of G_CRLC_TBF_Reconfig_REQ ASP

TBD

ASP Name	G_CRLC_TBF_Reconfig_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CRLC_TBF_Reconfig_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	
tFI	TFI	
Detailed Comments		

7.3.4.3.2.4 ASPs for configuration and control of GERAN LLC

ASP Name	G_CLLC_Assign_REQ	
PCO Type	G_CSAP	
Comments	The ASP is used to assign, change, or unassign the TLLI, the ciphering key (Kc) and the ciphering algorithm of GERAN LLC emulation module.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
oldTLLI	TLLI	OCTETSTRING[4]
newTLLI	TLLI	
cipherKey	BITSTRING[64]	
cipherAlgorethm	GPRS_CipherAlg	BITSTRING[3], see 3GPP TS 24.008 [9] clause 10.5.5.3
Detailed Comments		

7.3.4.3.2.5 ASPs for configuration and control of GERAN SNDCP

Declaration of G_CSNDCP_Activate_REQ ASP

ASP Name	G_CSNDCP_Activate_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CSNDCP_Activate_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
Detailed Comments		

ASP Name	G_CLLC_Assign_CNF	
PCO Type	G_CSAP	
Comments	The ASP is used to get the confirmation of a G_CLLC_Assign_REQ	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
Detailed Comments		

ASP Name	G_CLLC_Status_IND	
PCO Type	G_CSAP	
Comments	The ASP is used to get the LLC status report when an LLC error that cannot be recovered by the LLC layer has occurred.	
Parameter Name	**Parameter Type**	**Comments**
cellId	CellId	The identifier of the cell
tLLI	TLLI	32 bits
cause	Cause	
Detailed Comments	This ASP may be used in default tree to prevent dead lock when un-recoverable protocol error occurred in LLC emulator.	

8 Design Considerations

8.1 Channel mapping

The figure 18 shows the channel type mapping that is used for the configuration of the SS.

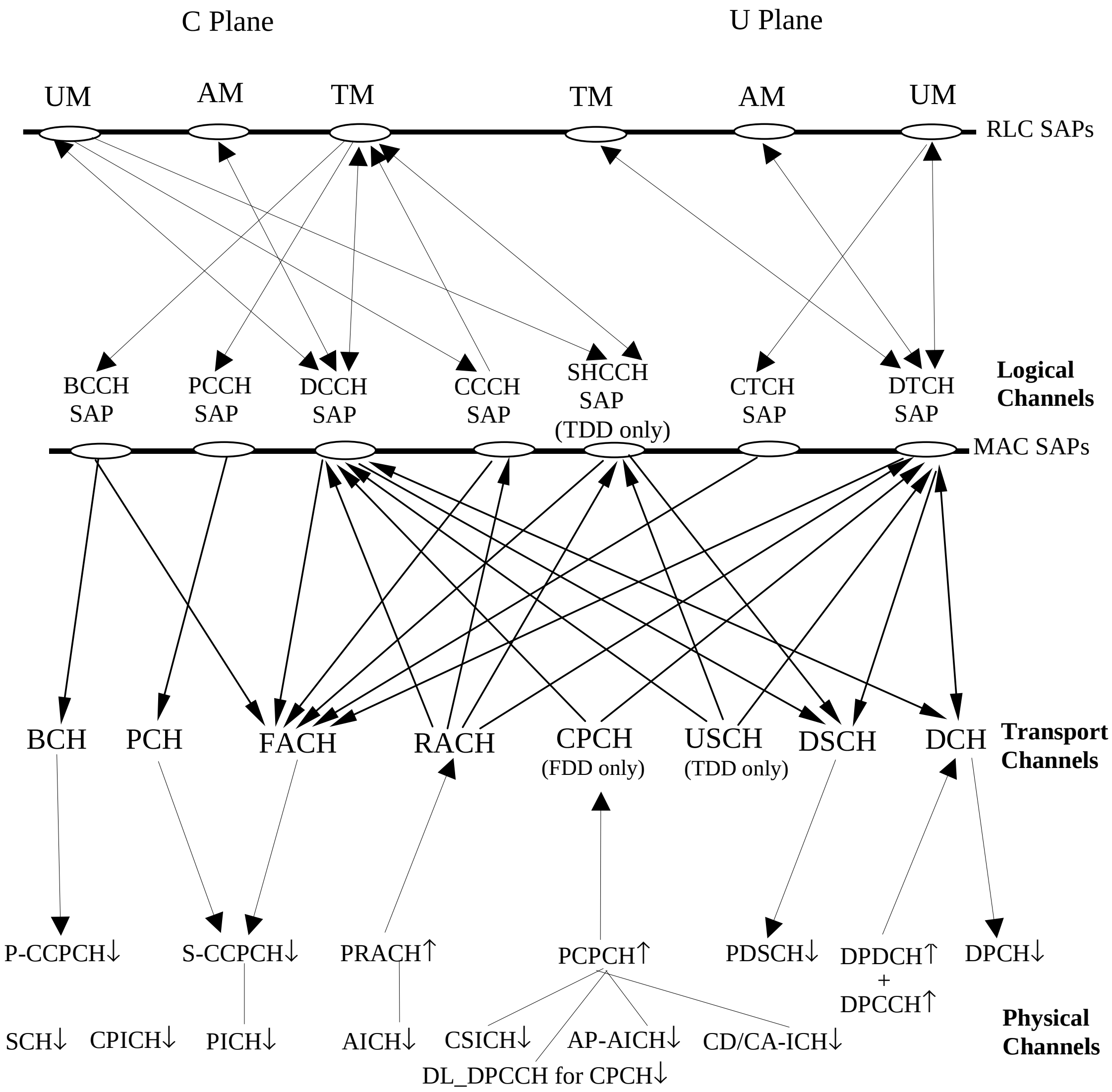

Figure 18: Channel mapping in SS

8.2 Channel and RB identity

The TTCN addresses the TTCN tester by using a channel identifier:

- Either Physical channel identifier (PhyCh id); or
- Transport channel identifier (TrCh id); or
- Radio bearer identifier (RB id).

The selected channel identifier identifies uniquely:

- a channel within a cell;
- a total path of the address in the lower layers concerned.

Having taken out the cell id and PCO id (AM, UM and TM), a complete address, as RoutingInfo in the RRC ASP definition, should have at least five fields, CN domain id, RB id, LogCH id, TrCH id and PhyCH id. For simplified application of CHOICE of the routing information, a TTCN writer must carefully follow a number of rules assigning the channel identifiers.

General requirements:

- a structured scheme of planning all channel identifiers assigned;
- the scheme shall meet the requirements for all test cases in 3GPP TS 34.123-1 [1] including TDD channels;
- the scheme can apply to all radio bearer configurations in 3GPP TS 34.108 [3], clause 6.10;
- a clear multiplex mapping between a PhyCH id to TrCH ids and a TrCH id to LogCH ids, RB ids is needed.

Requirements on identification of RB in a test case:

- unique identification of the individual SRBs;
- unique identification of the individual sub-flows of a RABs in CS and PS domain.;
- an assigned RB id can represent UL and DL.

Requirements on identification of Logical Channel in a test case:

- it is an instance number of the individual logical channel; and
- uniquely identifies among all the Logical Channel mapped onto a Transport Channel.

Requirements on identification of Transport Channel in a test case:

- unique identification of the individual Transport Channel;
- assign different identities for UL and DL of a same Transport Channel type;
- the order of the Transport Channel id assigned in a cell shall follow the TFCS definitions in the 3GPP TS 34.108 [3], clause 6.10.

EXAMPLE: Transport Channel ids are assigned in the ascending order for (RABsubflow#1, RABsubflow#2, RABsubflow#3, 64kRAB, DCCH).

Requirements on identification of Physical Channel in a test case:

- unique identification of the individual Physical Channel;
- assign different identities for UL and DL of a same Physical Channel type;
- each S-CCPCH or PRACH has a unique identifier;
- for 2 Mbps PS data radio link (in case of demux of a Transport Channel), three DPCH are needed for high-speed data. A single Physical Channel id is assigned to a bundle of the three physical channels.

Table 31 shows which type of channel identity is chosen for the individual primitives. In table 31, the ASN.1 primitives use a CHOICE type for channel identity, while TTCN primitives use an explicit channel identity.

Table 31: Primitives and the associated channel identity type

Primitive name	Channel Idientity
ASN.1 Primitives	
CPHY_AICH_AckModeSet_CNF	Physical Channel Identity
CPHY_AICH_AckModeSet_REQ	Physical Channel Identity
CPHY_Cell_Config_CNF	No Routing Info Field Present
CPHY_Cell_Config_REQ	No Routing Info Field Present
CPHY_Cell_Ini_CNF	No Routing Info Field Present
CPHY_Cell_Ini_REQ	No Routing Info Field Present
CPHY_Cell_TxPower_Modify_CNF	No Routing Info Field Present
CPHY_Cell_TxPower_Modify_REQ	No Routing Info Field Present

Primitive name	Channel Idientity
CPHY_Commit_CNF	Physical Channel Identity
CPHY_Commit_REQ	Physical Channel Identity
CPHY_Frame_Number_CNF	Physical Channel Identity
CPHY_Frame_Number_REQ	Physical Channel Identity
CPHY_Out_of_Sync_IND	Physical Channel Identity
CPHY_PRACH_Measurement_CNF	Physical Channel Identity
CPHY_PRACH_Measurement_REQ	Physical Channel Identity
CPHY_RL_Modify_CNF	Physical Channel Identity
CPHY_RL_Modify_REQ	Physical Channel Identity
CPHY_RL_Release_CNF	Physical Channel Identity
CPHY_RL_Release_REQ	Physical Channel Identity
CPHY_RL_Setup_CNF	Physical Channel Identity
CPHY_RL_Setup_REQ	PhysicalChannelIdentity
CPHY_Sync_IND	Physical Channel Identity
CPHY_TrCH_Config_CNF	Physical Channel Identity
CPHY_TrCH_Config_REQ	PhysicalChannelIdentity
CPHY_TrCH_Release_CNF	Physical Channel Identity
CPHY_TrCH_Release_REQ	Physical Channel Identity
CMAC_BMC_Scheduling_CNF	Physical Channel Identity
CMAC_BMC_Scheduling_REQ	Physical Channel Identity
CMAC_Ciphering_Activate_CNF	Physical Channel Identity of DPCH
CMAC_Ciphering_Activate_REQ	Physical Channel Identity of DPCH
CMAC_Config_CNF	Physical Channel Identity
CMAC_Config_REQ	PhysicalChannelIdentity
CMAC_PAGING_Config_CNF	Physical Channel Identity
CMAC_PAGING_Config_REQ	Physical Channel Identity
CMAC_Restriction_CNF	PhysicalChannelIdentity
CMAC_Restriction_REQ	PhysicalChannelIdentity
CMAC_SecurityMode_Config_CNF	No Routing Info Field Present (applies to all RB Ids)
CMAC_Sequence_Number_CNF	Physical Channel Identity
CMAC_SequenceNumber_REQ	Physical Channel Identity
CMAC_SYSINFO_Config_CNF	RB Identity
CMAC_SYSINFO_Config_REQ	RB Identity
CRLC_Ciphering_Activate_CNF	No Routing Info Field Present (applies to all RB Ids)
CRLC_Ciphering_Activate_REQ	No Routing Info Field Present (applies to all RB Ids)
CRLC_Config_CNF	RB Identity
CRLC_Config_REQ	RB_Identity
CRLC_Integrity_Activate_CNF	No Routing Info Field Present (applies to all RB Ids)
CRLC_Integrity_Activate_REQ	No Routing Info Field Present (applies to all RB Ids)
CRLC_Integrity_Failure_IND	RB Identity
CRLC_Resume_CNF	RB Identity (applies to all suspended RB Ids)
CRLC_Resume_REQ	RB Identity (applies to all suspended RB Ids)
CRLC_SecurityMode_Config_CNF	No Routing Info Field Present (applies to all RB Ids)
CRLC_SecurityMode_Config_REQ	No Routing Info Field Present (applies to all RB Ids)
CRLC_SequenceNumber_CNF	RB Identity
CRLC_SequenceNumber_REQ	RB Identity
CRLC_Status_Ind	RB Identity
CRLC_Suspend_CNF	RB Identity
CRLC_Suspend_REQ	RB Identity
CBMC_Config_CNF	RB Identity
CBMC_Config_REQ	RB Identity
RLC_AM_DATA_CNF	RB Identity
RLC_AM_DATA_IND	RB Identity
RLC_AM_DATA_REQ	RB Identity
RLC_TR_DATA_IND	RB Identity
RLC_TR_DATA_REQ	RB Identity
RLC_UM_DATA_IND	RB Identity
RLC_UM_DATA_REQ	RB Identity
TTCN Primitives	
RLC_AM_TestDataInd	RB Identity
RLC_AM_TestDataReq	RB Identity
RLC_TR_TestDataInd	RB Identity
RLC_TR_TestDataReq	RB Identity
RLC_UM_TestDataInd	RB Identity

Primitive name	Channel Idientity
RLC_UM_TestDataReq	RB Identity
BMC_DataReq	RB Identity

8.2.1 Physical Channels

Table 32: Physical channel identities

Type	Min. No.	Current Config.	Identities (value assigned)	Direction	Comment
P-CCPCH	1	1	tsc_P_CCPCH (4)	downlink	Primary Common Control Physical Channel. For Broadcasting System Information messages, using the Primary Scrambling Code for the Cell.
P-CPICH	1	1	tsc_P_CPICH (0)	downlink	Primary Common Pilot Channel using the Primary Scrambling Code for the Cell.
S-CPICH	1	FFS	tsc_S_CPICH (3)	downlink	Secondary Common Pilot Channel, used as the phase reference for some RF tests.
P-SCH	1	1	tsc_P_SCH (1)	downlink	Primary Synchronisation Channel
S-SCH	1	1	tsc_S_SCH (2)	downlink	Secondary Synchronisation Channel
S-CCPCH	2	1	tsc_S_CCPCH1 (5) tsc_S_CCPCH2 (10)	downlink	Secondary Common Control Physical Channel.
PICH	1	1	tsc_PICH1 (6) tsc_PICH2 (11)	downlink	To identify whether the UE should access the PCCH for Paging Messages.
AICH	1	1	tsc_AICH1 (7) tsc_AICH2 (12)	downlink	General Acquisition Indicator Channel, can be used for: - Aquisition Indicator Channel, for PRACH - Access Preamble Acquisition Indicator Channel (AP-ICH), for PCPCH - Collision-Detection/Channel-Assignment Indicator Channel (CD/CA-ICH), for PCPCH
DPCH	3	1	tsc_DL_DPCH1 (26) tsc_DL_DPCH2 (27)	downlink	Downlink Physical Data Channel. Layer 1 signalling is transmitted only on the first DPCH. This number is for the First Cell. Additional Cells may define a lower number which should be at least 1.
PDSCH	1	FFS		downlink	Physical Downlink Shared Channel.
DPDCH	1	1	tsc_UL_DPCH1 (20) tsc_UL_DPCH2 (21)	uplink	Uplink Dedicated Physical Channel. A single DPCCH associated with all the DPDCHs used for Layer 1 signalling.
PRACH	2	1	tsc_PRACH1 (8) tsc_PRACH2 (9)	uplink	Physical Random Access Channel.
PCPCH	1	FFS		uplink	Physical Common Packet Channel.
CSICH	1	FFS		downlink	CPCH Status Indicator Channel

The Physical Channel values 20 to 25 are assigned to uplink DPCHs and the values 26 to 31 are assigned to downlink DPCHs.

8.2.2 Transport Channels

Table 33: Transport channel identities

Type	Min. No.	Current Config.	Identities (value assigned)	Direction	Comments
BCH	1	1	tsc_BCH1 (11)	downlink	
FACH	1	1	tsc_FACH1 (13) tsc_FACH2 (14)	downlink	
PCH	1	1	tsc_PCH1 (12) tsc_PCH2 (30)	downlink	
DCH	n	4	tsc_UL_DCH1 (1) tsc_UL_DCH2 (2) tsc_UL_DCH3 (3) tsc_UL_DCH4 (4) tsc_UL_DCH5 (5)	uplink	tsc_UL_DCH1 for RAB subflow#1, tsc_UL_DCH2 for RAB subflow#2, tsc_UL_DCH3 for RAB subflow#3, tsc_UL_DCH4 for future use, tsc_UL_DCH5 for SRB.
DCH	n	4	tsc_DL_DCH1 (6) tsc_DL_DCH2 (7) tsc_DL_DCH3 (8) tsc_DL_DCH4 (9) tsc_DL_DCH5 (10)	downlink	tsc_DL_DCH1 for RAB subflow#1, tsc_DL_DCH2 for RAB subflow#2, tsc_DL_DCH3 for RAB subflow#3, tsc_DL_DCH4 for future use, tsc_DL_DCH5 for SRB.
USCH	1	N/A	tsc_USCH1(20)	uplink	TDD only
DSCH	1	N/A	tsc_DSCH (19)	downlink	
RACH	2	1	tsc_RACH1 (15) tsc_RACH2 (16)	uplink	
CPCH	1	N/A	tsc_CPCH1(17)	uplink	
FAUSCH	N/A	N/A	tsc_FAUSCH1(18)	uplink	Not in Release 99

The TrCH values 20 - 29 are assigned to the TDD TrCH.

8.2.3 Logical Channels

Table 34 shows the logical channels identities.

Table 34: Logical channel identities

Type	Min. No.	Current Config.	Identities (value assigned)	Direction	Comments
BCCH_BCH	1	1	tsc_BCCH1 (1)	downlink	
BCCH_FACH	1	1	tsc_BCCH6 (6)	downlink	
CCCH	1	1	tsc_DL_CCCH5 (5)	downlink	
CCCH	1	2	tsc_UL_CCCH5 (5) tsc_UL_CCCH6 (6)	uplink	
DCCH	4	4	tsc_DL_DCCH1 (1) tsc_DL_DCCH2 (2) tsc_DL_DCCH3 (3) tsc_DL_DCCH4 (4)	downlink	tsc_DL_DCCH1 for SRB1, tsc_DL_DCCH2 for SRB2, tsc_DL_DCCH3 for SRB3, tsc_DL_DCCH4 for SRB4
DCCH	4	4	tsc_UL_DCCH1 (1) tsc_UL_DCCH2 (2) tsc_UL_DCCH3 (3) tsc_UL_DCCH4 (4)	uplink	tsc_UL_DCCH1 for SRB1, tsc_UL_DCCH2 for SRB2, tsc_UL_DCCH3 for SRB3, tsc_UL_DCCH4 for SRB4
PCCH	1	2	tsc_PCCH1 (1) tsc_PCCH2 (2)	downlink	
DTCH	n	4	tsc_UL_DTCH1 (7) tsc_UL_DTCH2 (8) tsc_UL_DTCH3 (9) tsc_UL_DTCH4 (10)	uplink	tsc_UL_DTCH1for RAB subflow#1, tsc_UL_DTCH2 for RAB subflow#2, tsc_UL_DTCH3 for RAB subflow#3' tsc_UL_DTCH4 for future use
DTCH	n	4	tsc_DL_DTCH1 (7) tsc_DL_DTCH2 (8) tsc_DL_DTCH3 (9) tsc_DL_DTCH4 (10)	downlink	tsc_DL_DTCH1for RAB subflow#1, tsc_DL_DTCH2 for RAB subflow#2, tsc_DL_DTCH3 for RAB subflow#3, tsc_DL_DTCH4 for future use
CTCH	1	2	tsc_CTCH1 (11) tsc_CTCH2 (12)	downlink	

8.2.4 Radio bearers

Table 35: Radio bearer identities

Identities (value assigned)	Direction	Type	RLC mode	Service domain	Comments
tsc_RB_BCCH (-1)	downlink		TM	NA	BCCH-BCH
tsc_RB_PCCH (-2)	downlink		TM	NA	PCCH PCH
tsc_RB_BCCH_FACH (-3)	downlink		TM	NA	BCCH FACH
tsc_RB_2ndPCCH (-4)	downlink		TM	NA	Second PCCH PCH SCPCCH
tsc_RB_2ndCCCH (-5)	uplink		TM	NA	Second CCCH RACH PRACH
tsc_RB_UM_7_RLC (-10)	downlink	RAB	TM	CS	For UM RLC tests using 7 bit LIs
tsc_RB_UM_7_RLC (-10)	uplink	RAB	TM	CS	For UM RLC tests using 7 bit LIs
tsc_RB_UM_15_RLC (-11)	downlink	RAB	TM	CS	For UM RLC tests using 15 bit LIs
tsc_RB_UM_15_RLC (-11)	uplink	RAB	TM	CS	For UM RLC tests using 15 bit LIs
tsc_RB_AM_7_RLC (-12)	downlink	RAB	TM	CS	For AM RLC tests using 15 bit LIs
tsc_RB_AM_7_RLC (-12)	uplink	RAB	TM	CS	For AM RLC tests using 7 bit LIs
tsc_RB_AM_15_RLC (-13)	downlink	RAB	TM	CS	For AM RLC tests using 15 bit LIs
tsc_RB_AM_15_RLC (-13)	uplink	RAB	TM	CS	For AM RLC tests using 15 bit LIs
tsc_RB_DCCH_FACH_MAC (-14)	downlink	SRB3	TM	CS	For MAC tests using DCCH mapped to FACH
tsc_RB_DCCH_FACH_MAC (-14)	uplink	SRB3	TM	CS	For MAC tests using DCCH mapped to FACH
tsc_RB_DCCH_DCH_MAC (-15)	downlink	SRB3	TM	CS	For MAC tests using DCCH mapped to DCH
tsc_RB_DCCH_FACH_MAC (-15)	uplink	SRB3	TM	CS	For MAC tests using DCCH mapped to DCH
tsc_RB3_DCCH_RRC_(-16)	uplink	SRB3	AM	CS or PS	For RRC test cases to route UL NAS messages
tsc_RB_CCCH_FACH_MAC (-18)	downlink	SRB0	TM	CS or PS	For MAC test using donwlink SRB0 on TM
tsc_RB0 (0)	uplink	SRB0	TM	CS or PS	The service domain for which the most recent security negotiation took place. CCCH
tsc_RB0 (0)	downlink	SRB0	UM	CS or PS	CCCH
tsc_RB1 (1)	uplink	SRB1	UM	CS or PS	DCCH
tsc_RB1 (1)	downlink	SRB1	UM	CS or PS	DCCH
tsc_RB2 (2)	uplink	SRB2	AM	CS or PS	DCCH
tsc_RB2 (2)	downlink	SRB2	AM	CS or PS	DCCH
tsc_RB3 (3)	uplink	SRB3	AM	CS or PS	DCCH
tsc_RB3 (3)	downlink	SRB3	AM	CS or PS	DCCH
tsc_RB4 (4)	uplink	SRB4	AM	CS or PS	DCCH
tsc_RB4 (4)	downlink	SRB4	AM	CS or PS	DCCH
tsc_RB5 (5)	uplink		TM		DCCH
tsc_RB5 (5)	downlink		TM		DCCH
tsc_RB10 (10)	uplink	RAB#1	TM	CS	
tsc_RB10 (10)	downlink	RAB#1	TM	CS	
tsc_RB11 (11)	uplink	RAB#2	TM	CS	
tsc_RB11 (11)	downlink	RAB#2	TM	CS	
tsc_RB12 (12)	uplink	RAB#3	TM	CS	
tsc_RB12 (12)	downlink	RAB#3	TM	CS	
tsc_RB20 (20)	uplink	RAB#1	AM	PS	
tsc_RB20 (20)	downlink	RAB#1	AM	PS	
tsc_RB21 (21)	uplink	RAB#2	UM	PS	
tsc_RB21 (21)	downlink	RAB#2	UM	PS	
tsc_RB30 (30)	downlink		UM		CTCH FACH

The RB values 0-5 are used for the signalling bearers. The values 10-15 are assigned to the CS RAB sub-flows. The values 20-25 are assigned to the PS RAB sub-flows. The value 30 is assigned to the CBSMS/BMC service.

8.2.5 Scrambling and channelization codes

Table 36 shows the primary/secondary scrambling codes and the channelization codes for downlink channels.

Table36: Primary/seondary scrambling codes and channelization codes for downlink channels

Type	Identities (value assigned)	Primary scrambling code	Secondary scrambling code	Channelization Code
P-CCPCH	tsc_P_CCPCH (4)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA	tsc_P_CCPCH_ChC (256:1)
P-CPICH	tsc_P_CPICH (0)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA	tsc_P_CPICH_ChC (256:0)
S-CCPCH	tsc_S_CCPCH1 (5)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA (carrying PCH)	tsc_S_CCPCH1_ChC (64:1)
	tsc_S_CCPCH2 (10)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA (carrying PCH)	tsc_S_CCPCH2_ChC (64:2)
PICH	tsc_PICH1 (6)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA	tsc_PICH1_ChC (256:2)
	tsc_PICH2 (11)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA	tsc_PICH2_ChC (256:12)
AICH	tsc_AICH1 (7)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA	tsc_AICH1_ChC (256:3)
	tsc_AICH2 (12)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	NA	tsc_AICH2_ChC (256:13)
DPCH	tsc_DL_DPCH1 (26)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	tsc_DL_DPCH1_2ndScrC (1) This value is related to the primary scrambling code of the cell	Depending on the configuration: tsc_DL_DPCH1_ChC_SRB (256:0) tsc_DL_DPCH1_ChC_Speech (128:0) tsc_DL_DPCH1_ChC_Streaming (32:0) tsc_DL_DPCH1_ChC_64k_CS (32:0) tsc_DL_DPCH1_ChC_64k_PS (32:0)
	tsc_DL_DPCH2 (27)	(px_PrimaryScramblingCode + 50*(cell No –1)) mod 512	tsc_DL_DPCH2_2ndScrC (1) This value is related to the primary scrambling code of the cell	Depending on the configuration: tsc_DL_DPCH2_ChC_SRB (256:1) tsc_DL_DPCH2_ChC_Speech (128:1) tsc_DL_DPCH2_ChC_Streaming (32:1) tsc_DL_DPCH2_ChC_64k_CS (32:1) tsc_DL_DPCH2_ChC_64k_PS (32:1)

Table 37 shows the scrambling codes, the signatures and the spreading factors for uplink channels.

Table 37: Scrambling codes, signatures and spreading factor for uplink channels

Type	Identities (value assigned)	Scrambling code	Signature	Spreading factor
DPDCH	tsc_UL_DPCH1 (20)	(px_UL_ScramblingCode + 1000*(cell No –1)) MOD 16777216	NA	If only one DPDCH and depending on the configuration tsc_UL_DPDCH_SF_SRB (256) tsc_UL_DPDCH_SF_Speech (64) tsc_UL_DPDCH_SF_Streaming (16) tsc_UL_DPDCH_SF_64k_CS (16) tsc_UL_DPDCH_SF_64k_PS (16) If more than one DPDCH tsc_UL_DPDCH_SF_4 (4:1)
	tsc_UL_DPCH2 (21)	(px_UL_ScramblingCode + 1000*(cell No –1)) MOD 16777216	NA	If only one DPDCH and depending on the configuration tsc_UL_DPDCH_SF_SRB (256) tsc_UL_DPDCH_SF_Speech (64) tsc_UL_DPDCH_SF_Streaming (16) tsc_UL_DPDCH_SF_64k_CS (16) tsc_UL_DPDCH_SF_64k_PS (16) If more than one DPDCH tsc_UL_DPDCH_SF_4 (4:1)
PRACH	tsc_PRACH1 (8)	tsc_PRACH1_ScrC (0)	tsc_PRACH1_Signatures ('0000000011111111'B)	tsc_PRACH1_SF (64)
	tsc_PRACH2 (9)	tsc_PRACH2_ScrC (1)	tsc_PRACH2_Signatures ('0000000011111111'B)	tsc_PRACH2_SF (64)

8.2.6 MAC-d

MAC-d and the served RLC are cell-independent and are configured by using the cell-id = -1. During reconfigurations, cell changes and state transitions, the relevant counters in the RLC and MAC-d are maintained.

For the active set updating, the DL DCH with the same channel Id in the different cells are implicitly connected to form the DL multiple paths.

8.3 Channels configurations

8.3.1 Configuration of Cell_FACH

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 for downlink and 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1 for uplink. The configuration is applied to the RRC tests related in the states CELL_FACH, CELL_PCH and URA_PCH. They need a minimum radio configuration for testing.

Table 38: Uplink configuration of Cell_FACH

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH
LogCh Identity	Tsc_UL_DTCH1 (7)	tsc_UL_CCCH5 (5)	tsc_UL_DCCH1 (1)	tsc_UL_DCCH2 (2)	tsc_UL_DCCH3 (3)	tsc_UL_DCCH4 (4)
RLC mode	AM	TM	UM	AM	AM	AM
TrCH Type	RACH					
TrCH identity	tsc_RACH1 (15)					
PhyCh Type	PRACH					
PhyCH identity	tsc_PRACH1 (8)					

Table 39: Downlink configuration of Cell_FACH

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB_BCCH_FACH (-3)	tsc_RB_PCCH (-2)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH	BCCH	PCCH
LogCh Identity	tsc_DL_DTCH1 (6)	tsc_DL_CCCH5 (5)	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_BCCH6 (6)	tsc_PCCH1 (1)
RLC mode	AM	UM	UM	AM	AM	AM	TM	TM
MAC priority	1	1	2	3	4	5	6	1
TrCH Type	FACH	FACH						PCH
TrCH identity	tsc_FACH2 (14)	tsc_FACH1 (13)						tsc_PCH1 (12)
PhyCh Type	Secondary CCPCH							
PhyCH identity	tsc_S_CCPCH1 (5)							

8.3.2 Configuration of Cell_DCH_StandAloneSRB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.2. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to the RRC and NAS signalling tests in the DCH state without RAB.

Table 40: Uplink configuration of Cell_DCH_StandAloneSRB

RB Identity	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB0 (0)	
LogCh Type	DCCH	DCCH	DCCH	DCCH	CCCH	
LogCh Identity	tsc_UL_DCCH1 (1)	tsc_UL_DCCH2 (2)	tsc_UL_DCCH3 (3)	tsc_UL_DCCH4 (4)	tsc_UL_CCCH5 (5)	
RLC mode	UM	AM	AM	AM	TM	AM
TrCH Type	DCH				RACH	
TrCH identity	tsc_UL_DCH5 (5)				tsc_RACH1 (15)	
PhyCh Type	DPDCH				PRACH	
PhyCH identity	tsc_UL_DPCH1 (20)				tsc_PRACH1 (8)	

Table 41: Downlink configuration of Cell_DCH_StandAloneSRB

RB Identity	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB0 (0)	tsc_RB_PCCH (-2)	
LogCh Type	DCCH	DCCH	DCCH	DCCH	CCCH	PCCH	
LogCh Identity	tsc_DL_DCCH 1 (1)	tsc_DL_DCCH 2 (2)	tsc_DL_DCCH 3 (3)	tsc_DL_DCCH 4 (4)	tsc_DL_CCCH 5 (5)	tsc_PCCH1 (1)	
RLC mode	UM	AM	AM	AM	UM	TM	AM
MAC priority	1	2	3	4	1	1	1
TrCH Type	DCH				FACH	PCH	FACH
TrCH identity	tsc_DL_DCH5 (10)				tsc_FACH1 (13)	tsc_PCH1 (12)	tsc_FACH2 (14)
PhyCh Type	DPCH				Secondary CCPCH		
PhyCH identity	tsc_DL_DPCH1 (26)				tsc_S_CCPCH1 (5)		

8.3.3 Configuration of Cell_DCH_Speech

The configuration is based on 3GPP TS 34.108 [3], clauses 6.10.2.4.1.4 and 6.10.2.4.1.5. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to those RRC and NAS signalling tests in the DCH state where a CS voice service, such as narrowband speech, emergency speech call or TS 61 for speech, is established.

Table 42: Uplink configuration of Cell_DCH_Speech

RB Identity	tsc_RB10 (10)	tsc_RB11 (11)	tsc_RB12 (12)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH	DTCH	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)	tsc_UL_DTCH2 (8)	tsc_UL_DTCH3 (9)		
RLC mode	TM	TM	TM		
TrCH Type	DCH	DCH	DCH		
TrCH identity	tsc_UL_DCH1 (1)	tsc_UL_DCH2 (2)	tsc_UL_DCH3 (3)		
PhyCh Type	DPDCH				PRACH
PhyCH identity	tsc_UL_DPCH1 (20)				tsc_PRACH1 (8)

Table 43: Downlink configuration of Cell_DCH_Speech

RB Identity	tsc_RB10 (10)	tsc_RB11 (11)	tsc_RB12 (12)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH	DTCH	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)	tsc_DL_DTCH2 (8)	tsc_DL_DTCH3 (9)		
RLC mode	TM	TM	TM		
MAC priority	1	1	1		
TrCH Type	DCH	DCH	DCH		
TrCH identity	tsc_DL_DCH1 (6)	tsc_DL_DCH2 (7)	tsc_DL_DCH3 (8)		
PhyCh Type	DPCH				Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)				tsc_S_CCPCH1 (5)

8.3.4 Configuration of Cell_DCH_64kCS_RAB_SRB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.13 for the conversational unknown quality class. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to those RRC and NAS signalling tests in the DCH state where one of the following CS transparent data services is established:

- Multimedia call 28,8 kbit/s, 3,1 kHz Audio;
- Multimedia call 32 kbit/s, UDI;
- Multimedia call 33,6 kbit/s, 3,1 kHz Audio;
- Multimedia call 56 kbit/s, RDI;
- Multimedia call 64 kbit/s, UDI;
- Asynchronous 3,1 kHz Audio 28,8 kbit/s;
- Synchronous 3,1 kHz Audio 28,8 kbit/s;

- Synchronous V.110 UDI up to 56 kbit/s;
- BTM RDI 56 kbit/s;
- BTM UDI 64 bit/s.

Table 44: Uplink configuration of Cell_DCH_64kCS_RAB_SRB

RB Identity	tsc_RB10 (10)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)		
RLC mode	TM		
TrCH Type	DCH		
TrCH identity	tsc_UL_DCH1 (1)		
PhyCh Type	DPDCH		PRACH
PhyCH identity	tsc_UL_DPCH1 (20)		tsc_PRACH1 (8)

Table 45: Downlink configuration of Cell_DCH_64kCS_RAB_SRB

RB Identity	tsc_RB10 (10)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH		
LogCh Identity	tsc_DL_DTCH 1 (7)		
RLC mode	TM		
MAC priority	1		
TrCH Type	DCH		
TrCH identity	tsc_DL_DCH1 (6)		
PhyCh Type	DPCH		Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)		tsc_S_CCPCH1 (5)

8.3.5 Configuration of Cell_DCH_57_6kCS_RAB_SRB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.17 for the streaming unknown quality class. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to those RRC and NAS signalling tests in the DCH state where one of the following CS non-transparent data services is established:

- Asynchronous 3,1 kHz Audio up to 19,2 kbit/s;
- Asynchronous 3,1 kHz Audio modem auto-bauding;
- Asynchronous V.110 UDI up to 38,4 kbit/s, except 28,8 kbit/s;
- Asynchronous V.120 up to 56 kbit/s;
- Asynchronous PIAFS up to 64 kbit/s;
- Asynchronous FTM up to 64 kbit/s;
- Synchronous 3,1 kHz Audio up to 19,2 kbit/s;

- Synchronous V.110 UDI up to 56 kbit/s, except 28,8 kbit/s;
- Synchronous X.31 Flags Stuffing UDI up to 56 kbit/s;
- Synchronous V.120 up to 56 kbit/s;
- Synchronous BTM up to 64 kbit/s;
- TS61 FAX.

Table 46: Uplink configuration of Cell_DCH_57_6kCS_RAB_SRB

RB Identity	tsc_RB10 (10)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)		
RLC mode	TM		
TrCH Type	DCH		
TrCH identity	tsc_UL_DCH1 (1)		
PhyCh Type	DPDCH		PRACH
PhyCH identity	tsc_UL_DPCH1 (20)		tsc_PRACH1 (8)

Table 47: Downlink configuration of Cell_DCH_57_6kCS_RAB_SRB

RB Identity	tsc_RB10 (10)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)		
RLC mode	TM		
MAC priority	1		
TrCH Type	DCH		
TrCH identity	tsc_DL_DCH1 (6)		
PhyCh Type	DPCH		Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)		tsc_S_CCPCH1 (5)

8.3.6 Configuration of Cell_RLC_DCH_ RAB

The configuration is based on 3GPP TS 34.108 [3], clauses 6.11.1, 6.11.2, 6.11.3, and 6.11.4 for the RLC AM and UM tests with 7 and 15 bit length indicators. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1.

The RB Ids used for the DTCH depend on the RLC mode and length indicator size being simulated (reference clause 6.5.2, RLC test method). Table 48 shows the test suite constants used for each RLC mode, and length indicator size.

Table 48: RB Ids used for DTCH depending on RLC mode and LI size

RLC mode	LI Size	TSC	RB Id
UM	7	tsc_RB_UM_7_RLC	-10
UM	15	tsc_RB_UM_15_RLC	-11
AM	7	tsc_RB_AM_7_RLC	-12
AM	15	tsc_RB_AM_15_RLC	-13

Table 49: Uplink configuration of Cell_RLC_DCH_RAB

RB Identity	See table 48	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)		
RLC mode	TM		
TrCH Type	DCH		
TrCH identity	tsc_UL_DCH1 (1)		
PhyCh Type	DPDCH		PRACH
PhyCH identity	tsc_UL_DPCH1 (20)		tsc_PRACH1 (8)

Table 50: Downlink configuration of Cell_RLC_DCH_RAB

RB Identity	See table 48	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)		
RLC mode	TM		
MAC priority	1		
TrCH Type	DCH		
TrCH identity	tsc_DL_DCH1 (6)		
PhyCh Type	DPCH		Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)		tsc_S_CCPCH1 (5)

8.3.7 Configuration of Cell_FACH_BMC

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 for downlink and 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1 without RAB/DTCH for uplink. A RB30/CTCH is configured. The configuration is applied to the BMC and CBSMS tests.

The uplink configuration of Cell_FACH_BMC is the same as the uplink configuration of Cell_FACH.

Table 51: Downlink configuration of Cell_FACH_BMC

RB Identity		tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB_BCCH_FACH (-3)	Tsc_RB30 (30)	tsc_RB_PCCH (-2)
LogCh Type		CCCH	DCCH	DCCH	DCCH	DCCH	BCCH	CTCH	PCCH
LogCh Identity		tsc_DL_CCCH5 (5)	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_BCCH6 (6)	Tsc_CTCH (11)	tsc_PCCH1 (1)
RLC mode	AM	UM	UM	AM	AM	AM	TM	UM	TM
MAC priority	1	1	2	3	4	5	6	7	1
TrCH Type	FACH	FACH							PCH
TrCH identity	tsc_FACH2 (14)	tsc_FACH1 (13)							tsc_PCH1 (12)
PhyCh Type	Secondary CCPCH								
PhyCH identity	tsc_S_CCPCH1 (5)								

8.3.8 Configuration of PS Cell_DCH_64kPS_RAB_SRB and Cell_PDCP_AM_RAB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.26. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to those RRC and NAS signalling tests in the DCH state where a PS RAB on DTCH is setup for the interactive or background service class. The configuration is applied to PDCP test cases in acknowledge mode.

Table 52: Uplink configuration of PS Cell_DCH_64kPS_RAB_SRB SRB and Cell_PDCP_AM_RAB

RB Identity	tsc_RB20 (20)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)		
RLC mode	AM		
TrCH Type	DCH		
TrCH identity	tsc_UL_DCH1 (1)		
PhyCh Type	DPDCH		PRACH
PhyCH identity	tsc_UL_DPCH1 (20)		tsc_PRACH1 (8)

Table 53: Downlink configuration of PS Cell_DCH_64kPS_RAB_SRB SRB and Cell_PDCP_AM_RAB

<table>
<tr><th>RB Identity</th><td>tsc_RB20
(20)</td><td rowspan="7">Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH</td><td rowspan="7">Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH</td></tr>
<tr><th>LogCh Type</th><td>DTCH</td></tr>
<tr><th>LogCh Identity</th><td>tsc_DL_DTCH1
(7)</td></tr>
<tr><th>RLC mode</th><td>AM</td></tr>
<tr><th>MAC priority</th><td>1</td></tr>
<tr><th>TrCH Type</th><td>DCH</td></tr>
<tr><th>TrCH identity</th><td>tsc_DL_DCH1
(6)</td></tr>
<tr><th>PhyCh Type</th><td colspan="2">DPCH</td><td>Secondary CCPCH</td></tr>
<tr><th>PhyCH identity</th><td colspan="2">tsc_DL_DPCH1
(26)</td><td>tsc_S_CCPCH1
(5)</td></tr>
</table>

8.3.9 Configuration of Cell_Two_DTCH

The configuration is based on 3GPP TS 34.108 [3], clauses 6.10.2.4.1.6 to 6.10.2.4.1.11. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to RB tests.

Table 54: Uplink configuration of Cell_Two_DTCH

<table>
<tr><th>RB Identity</th><td>tsc_RB10
(10)</td><td>tsc_RB11
(11)</td><td rowspan="6">Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH</td><td rowspan="6">Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH</td></tr>
<tr><th>LogCh Type</th><td>DTCH</td><td>DTCH</td></tr>
<tr><th>LogCh Identity</th><td>tsc_UL_DTCH1
(7)</td><td>tsc_UL_DTCH2
(8)</td></tr>
<tr><th>RLC mode</th><td>TM</td><td>TM</td></tr>
<tr><th>TrCH Type</th><td>DCH</td><td>DCH</td></tr>
<tr><th>TrCH identity</th><td>tsc_UL_DCH1
(1)</td><td>tsc_UL_DCH2
(2)</td></tr>
<tr><th>PhyCh Type</th><td colspan="3">DPCH</td><td>PRACH</td></tr>
<tr><th>PhyCH identity</th><td colspan="3">tsc_UL_DPDCH1
(20)</td><td>tsc_PRACH1
(8)</td></tr>
</table>

Table 55: Downlink configuration of Cell_Two_DTCH

<table>
<tr><th>RB Identity</th><td>tsc_RB10
(10)</td><td>tsc_RB11
(11)</td><td rowspan="7">Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH</td><td rowspan="7">Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH</td></tr>
<tr><th>LogCh Type</th><td>DTCH</td><td>DTCH</td></tr>
<tr><th>LogCh Identity</th><td>tsc_DL_DTCH1
(7)</td><td>tsc_DL_DTCH2
(8)</td></tr>
<tr><th>RLC mode</th><td>TM</td><td>TM</td></tr>
<tr><th>MAC priority</th><td>1</td><td>1</td></tr>
<tr><th>TrCH Type</th><td>DCH</td><td>DCH</td></tr>
<tr><th>TrCH identity</th><td>tsc_DL_DCH1
(6)</td><td>tsc_DL_DCH2
(7)</td></tr>
<tr><th>PhyCh Type</th><td colspan="3">DPCH</td><td>Secondary CCPCH</td></tr>
<tr><th>PhyCH identity</th><td colspan="3">tsc_DL_DPCH1
(26)</td><td>tsc_S_CCPCH1
(5)</td></tr>
</table>

8.3.10 Configuration of Cell_Single_DTCH (CS)

The configuration is based on 3GPP TS 34.108 [3], clauses 6.10.2.4.1.12 to 6.10.2.4.1.22. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to RB tests.

Table 56: Uplink configuration of Cell_Single_DTCH (CS)

RB Identity	tsc_RB10 (10)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)		
RLC mode	TM		
TrCH Type	DCH		
TrCH identity	tsc_UL_DCH1 (1)		
PhyCh Type	DPDCH		PRACH
PhyCH identity	tsc_UL_DPCH1 (20)		tsc_PRACH1 (8)

Table 57: Downlink configuration of Cell_Single_DTCH (CS)

RB Identity	tsc_RB10 (10)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)		
RLC mode	TM		
MAC priority	1		
TrCH Type	DCH		
TrCH identity	tsc_DL_DCH1 (6)		
PhyCh Type	DPCH		Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)		tsc_S_CCPCH1 (5)

8.3.11 Configuration of PS Cell_PDCP_UM_RAB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.26. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to PDCP test cases in unacknowledge mode.

Table 58: Uplink configuration of PS Cell_PDCP_UM_RAB

RB Identity	tsc_RB21 (21)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)		
RLC mode	UM		
TrCH Type	DCH		
TrCH identity	tsc_UL_DCH1 (1)		
PhyCh Type	DPDCH		PRACH
PhyCH identity	tsc_UL_DPCH1 (20)		tsc_PRACH1 (8)

Table 59: Downlink configuration of PS Cell_PDCP_UM_RAB

RB Identity	tsc_RB21 (21)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)		
RLC mode	UM		
MAC priority	1		
TrCH Type	DCH		
TrCH identity	tsc_DL_DCH1 (6)		
PhyCh Type	DPCH		Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)		tsc_S_CCPCH1 (5)

8.3.12 Configuration of PS Cell_PDCP_AM_UM_RAB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.26. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to PDCP test cases using both the acknowledged and unacknowledged mode.

Table 60: Uplink configuration of PS Cell_PDCP_AM_UM_RAB

RB Identity	tsc_RB20 (20)	tsc_RB21 (21)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)	tsc_UL_DTCH2 (8)		
RLC mode	AM	UM		
TrCH Type	DCH			
TrCH identity	tsc_UL_DCH1 (1)			
PhyCh Type	DPDCH			PRACH
PhyCH identity	tsc_UL_DPCH1 (20)			tsc_PRACH1 (8)

Table 61: Downlink configuration of PS Cell_PDCP_AM_UM_RAB

RB Identity	tsc_RB20 (20)	tsc_RB21 (21)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)	tsc_DL_DTCH2 (8)		
RLC mode	AM	UM		
MAC priority	1	1		
TrCH Type	DCH			
TrCH identity	tsc_DL_DCH1 (6)			
PhyCh Type	DPCH			Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (26)			tsc_S_CCPCH1 (5)

8.3.13 Configuration of Cell_2SCCPCH_BMC

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 for downlink and 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1 without RAB/DTCH for uplink. RB30/CTCH and RB31/CTCH as well as two PCCH are configured. The configuration is applied to the BMC and CBSMS tests.

Table 62: Uplink configuration of Cell_2SCCPCH_BMC

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	Tsc_RB3 (3)	tsc_RB4 (4)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH
LogCh Identity	Tsc_UL_DTCH1 (7)	tsc_UL_CCCH5 (5)	tsc_UL_DCCH1 (1)	tsc_UL_DCCH2 (2)	tsc_UL_DCCH3 (3)	tsc_UL_DCCH4 (4)
RLC mode	AM	TM	UM	AM	AM	AM
TrCH Type	RACH					
TrCH identity	tsc_RACH1 (15)					
PhyCh Type	PRACH					
PhyCH identity	tsc_PRACH1 (8)					

Table 63: Downlink configuration of Cell_2SCCPCH_BMC: second S-CCPCH

RB Identity	Tsc_RB31 (31)	tsc_RB_2ndPCCH (-4)
LogCh Type	CTCH	PCCH
LogCh Identity	Tsc_CTCH2 (12)	tsc_PCCH2 (2)
RLC mode	UM	TM
MAC priority	1	1
TrCH Type	FACH	PCH
TrCH identity	tsc_FACH1 (13)	tsc_PCH2 (30)
PhyCh Type	Secondary CCPCH	
PhyCH identity	tsc_S_CCPCH2 (10)	

Table 64: Downlink configuration of Cell_2SCCPCH_BMC: first S-CCPCCH

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB_BCCH_FACH (-3)	Tsc_RB30 (30)	tsc_RB_PCCH (-2)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH	BCCH	CTCH	PCCH
LogCh Identity	tsc_DL_DTCH1 (6)	tsc_DL_CCCH5 (5)	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_BCCH6 (6)	Tsc_CTCH1 (11)	tsc_PCCH1 (1)
RLC mode	AM	UM	UM	AM	AM	AM	TM	UM	TM
MAC priority	1	1	2	3	4	5	6	7	1
TrCH Type	FACH	FACH							PCH
TrCH identity	Tsc_FACH2 (14)	tsc_FACH1 (13)							tsc_PCH1 (12)
PhyCh Type	Secondary CCPCH								
PhyCH identity	tsc_S_CCPCH1 (5)								

8.3.14 Configuration of Cell_Four_DTCH_CS_PS

The configuration is based on 3GPP TS 34.108 [3], clauses 6.10.2.4.1.40. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to RB tests.

Table 65: Uplink configuration of Cell_Four_DTCH_CS_PS

RB Identity	tsc_RB10 (10)	tsc_RB11 (11)	tsc_RB12 (12)	tsc_RB20 (20)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAlone SRB on PRACH
LogCh Type	DTCH	DTCH	DTCH	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)	tsc_UL_DTCH2 (8)	tsc_UL_DTCH3 (9)	tsc_UL_DTCH4 (10)		
RLC mode	TM	TM	TM	AM		
MAC priority	1	1	1	1		
TrCH Type	DCH	DCH	DCH	DCH		
TrCH identity	tsc_UL_DCH1 (6)	tsc_UL_DCH2 (7)	tsc_UL_DCH3 (8)	tsc_UL_DCH4 (9)		
PhyCh Type	DPDCH					Secondary CCPCH
PhyCH identity	tsc_UL_DPCH1 (20)					tsc_S_CCPCH1 (5)

Table 66: Downlink configuration of Cell_Four_DTCH_CS_PS

RB Identity	tsc_RB10 (10)	tsc_RB11 (11)	tsc_RB12 (12)	tsc_RB20 (20)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAlone SRB on sCCPCH
LogCh Type	DTCH	DTCH	DTCH	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)	tsc_DL_DTCH2 (8)	tsc_DL_DTCH3 (9)	tsc_DL_DTCH4 (10)		
RLC mode	TM	TM	TM	AM		
MAC priority	1	1	1	1		
TrCH Type	DCH	DCH	DCH	DCH		
TrCH identity	tsc_DL_DCH1 (6)	tsc_DL_DCH2 (7)	Tsc_DL_DCH3 (8)	tsc_DL_DCH4 (9)		
PhyCh Type	DPCH					Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (20)					tsc_S_CCPCH1 (5)

8.3.15 Configuration of Cell_Two_DTCH_CS_PS

The configuration is based on 3GPP TS 34.108 [3], clauses 6.10.2.4.1.51 and 6.10.2.4.1.53. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to RB tests.

Table 67:Uplink configuration of Cell_Two_DTCH_CS_PS

RB Identity	tsc_RB10 (10)	tsc_RB20 (20)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)	tsc_UL_DTCH2 (8)		
RLC mode	TM	AM		
TrCH Type	DCH	DCH		
TrCH identity	tsc_UL_DCH1 (1)	tsc_UL_DCH2 (2)		
PhyCh Type	DPDCH			PRACH
PhyCH identity	tsc_UL_DPCH1 (20)			tsc_PRACH1 (8)

Table 68: Downlink configuration of Cell_Two_DTCH_CS_PS

RB Identity	tsc_RB10 (10)	tsc_RB20 (20)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH
LogCh Type	DTCH	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)	tsc_DL_DTCH2 (8)		
RLC mode	TM	AM		
MAC priority	1	1		
TrCH Type	DCH	DCH		
TrCH identity	tsc_DL_DCH1 (6)	tsc_DL_DCH2 (7)		
PhyCh Type	DPCH			Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (20)			tsc_S_CCPCH1 (5)

8.3.16 Configuration of Cell_Four_DTCH_CS

The configuration is based on 3GPP TS 34.108 [3], clauses 6.10.2.4.1.49. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to RB tests.

Table 69: Uplink configuration of Cell_Four_DTCH_CS

RB Identity	tsc_RB10 (10)	tsc_RB11 (11)	tsc_RB12 (12)	tsc_RB13 (13)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH	DTCH	DTCH	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)	tsc_UL_DTCH2 (8)	tsc_UL_DTCH3 (9)	tsc_UL_DTCH4 (10)		
RLC mode	TM	TM	TM	TM		
MAC priority	1	1	1	1		
TrCH Type	DCH	DCH	DCH	DCH		
TrCH identity	tsc_UL_DCH1 (6)	tsc_UL_DCH2 (7)	tsc_UL_DCH3 (8)	tsc_UL_DCH4 (9)		
PhyCh Type	DPDCH					Secondary CCPCH
PhyCH identity	tsc_UL_DPCH1 (20)					tsc_S_CCPCH1 (5)

Table 70: Downlink configuration of Cell_Four_DTCH_CS

RB Identity	tsc_RB10 (10)	tsc_RB11 (11)	tsc_RB12 (12)	tsc_RB13 (13)	Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as downlink configuration of Cell_DCH_StandAlone SRB on sCCPCH
LogCh Type	DTCH	DTCH	DTCH	DTCH		
LogCh Identity	tsc_DL_DTCH1 (7)	tsc_DL_DTCH2 (8)	tsc_DL_DTCH3 (9)	tsc_DL_DTCH4 (10)		
RLC mode	TM	TM	TM	TM		
MAC priority	1	1	1	1		
TrCH Type	DCH	DCH	DCH	DCH		
TrCH identity	tsc_DL_DCH 1 (6)	tsc_DL_DCH 2 (7)	tsc_DL_DCH 3 (8)	tsc_DL_DCH 4 (9)		
PhyCh Type	DPCH					Secondary CCPCH
PhyCH identity	tsc_DL_DPCH1 (20)					tsc_S_CCPCH1 (5)

8.3.17 Configuration of Cell_DCH_MAC_SRB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.2. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1; except that RB3 is mapped on TM mode.

The configuration is applied to the MAC tests.

Table 71: Uplink configuration of Cell_DCH_MAC_SRB

RB Identity	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB_DCCH_DCH_MAC (-15)	tsc_RB4 (4)	tsc_RB0 (0)	
LogCh Type	DCCH	DCCH	DCCH	DCCH	CCCH	
LogCh Identity	tsc_UL_DCCH1 (1)	tsc_UL_DCCH2 (2)	tsc_UL_DCCH3 (3)	tsc_UL_DCCH4 (4)	tsc_UL_CCCH5 (5)	
RLC mode	UM	AM	TM	AM	TM	AM
TrCH Type	DCH				RACH	
TrCH identity	tsc_UL_DCH5 (5)				tsc_RACH1 (15)	
PhyCh Type	DPDCH				PRACH	
PhyCH identity	tsc_UL_DPCH1 (20)				tsc_PRACH1 (8)	

Table 72: Downlink configuration of Cell_DCH_MAC_SRB

RB Identity	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB_DCCH_DCH_MAC (-15)	tsc_RB4 (4)	tsc_RB0 (0)	tsc_RB_PCCH (-2)	
LogCh Type	DCCH	DCCH	DCCH	DCCH	CCCH	PCCH	
LogCh Identity	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_DL_CCCH5 (5)	tsc_PCCH1 (1)	
RLC mode	UM	AM	TM	AM	UM	TM	AM
MAC priority	1	2	3	4	1	1	1
TrCH Type	DCH				FACH	PCH	FACH
TrCH identity	tsc_DL_DCH5 (10)				tsc_FACH1 (13)	tsc_PCH1 (12)	tsc_FACH2 (14)
PhyCh Type	DPCH				Secondary CCPCH		
PhyCH identity	tsc_DL_DPCH1 (26)				tsc_S_CCPCH1 (5)		

8.3.18 Configuration of Cell_FACH_MAC_SRB

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 for downlink and 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1 for uplink; except that RB3 is mapped on TM mode.

The configuration is applied to the MAC tests.

Table 73: Uplink configuration of Cell_FACH_MAC_SRB

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB_DCCH_FACH_MAC (-14)	tsc_RB4 (4)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH
LogCh Identity	Tsc_UL_DTCH1 (7)	tsc_UL_CCCH5 (5)	tsc_UL_DCCH1 (1)	tsc_UL_DCCH2 (2)	tsc_UL_DCCH3 (3)	tsc_UL_DCCH4 (4)
RLC mode	AM	TM	UM	AM	TM	AM
TrCH Type	RACH					
TrCH identity	tsc_RACH1 (15)					
PhyCh Type	PRACH					
PhyCH identity	tsc_PRACH1 (8)					

Table 74: Downlink configuration of Cell_FACH_MAC_SRB

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB_DCCH_FACH_MAC (-14)	tsc_RB4 (4)	tsc_RB_BCCH_FACH (-3)	tsc_RB_PCCH (-2)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH	BCCH	PCCH
LogCh Identity	tsc_DL_DTCH1 (6)	tsc_DL_CCCH5 (5)	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_BCCH6 (6)	tsc_PCCH1 (1)
RLC mode	AM	UM	UM	AM	TM	AM	TM	TM
MAC priority	1	1	2	3	4	5	6	1
TrCH Type	FACH	FACH						PCH
TrCH identity	tsc_FACH2 (14)	tsc_FACH1 (13)						tsc_PCH1 (12)
PhyCh Type	Secondary CCPCH							
PhyCH identity	tsc_S_CCPCH1 (5)							

8.3.19 Configuration of Cell_FACH_MAC_SRB0

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 for downlink and 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1 for uplink; except that the downlink SRB0 is mapped on TM mode.

The configuration is applied to the MAC tests.

The uplink configuration of Cell_FACH_MAC_SRB0 is the same as the uplink configuration of Cell_FACH.

Table 75: Downlink configuration of Cell_FACH_MAC_SRB0

RB Identity	tsc_RB20 (20)	tsc_RB_CCCH_FACH_MAC (-18)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB_BCCH_FACH (-3)	tsc_RB_PCCH (-2)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH	BCCH	PCCH
LogCh Identity	tsc_DL_DTCH1 (6)	tsc_DL_CCCH5 (5)	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_BCCH6 (6)	tsc_PCCH1 (1)
RLC mode	AM	TM	UM	AM	AM	AM	TM	TM
MAC priority	1	1	2	3	4	5	6	1
TrCH Type	FACH	FACH						PCH
TrCH identity	tsc_FACH2 (14)	tsc_FACH1 (13)						tsc_PCH1 (12)
PhyCh Type	Secondary CCPCH							
PhyCH identity	tsc_S_CCPCH1 (5)							

8.3.20 Configuration of Cell_FACH_2_SCCPCH_StandAlonePCH

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 for downlink and 3GPP TS 34.108 [3] except the mapping of PCH, clause 6.10.2.4.4.1.1.1 for uplink.

The configuration is applied to the MAC tests.

The uplink configuration of Cell_FACH_2_SCCPCH_StandAlonePCH is the same as the uplink configuration of Cell_FACH.

Table 76: Downlink configuration of Cell_FACH_2_SCCPCH_StandAlonePCH

RB Identity	tsc_RB20 (20)	tsc_RB0 (0)	tsc_RB1 (1)	tsc_RB2 (2)	tsc_RB3 (3)	tsc_RB4 (4)	tsc_RB_BCCH_FACH (-3)	tsc_RB_PCCH2 (-19)
LogCh Type	DTCH	CCCH	DCCH	DCCH	DCCH	DCCH	BCCH	PCCH
LogCh Identity	tsc_DL_DTCH1 (6)	tsc_DL_CCCH5 (5)	tsc_DL_DCCH1 (1)	tsc_DL_DCCH2 (2)	tsc_DL_DCCH3 (3)	tsc_DL_DCCH4 (4)	tsc_BCCH6 (6)	tsc_PCCH1 (1)
RLC mode	AM	UM	UM	AM	AM	AM	TM	TM
MAC priority	1	1	2	3	4	5	6	1
TrCH Type	FACH	FACH						PCH
TrCH identity	tsc_FACH2 (14)	tsc_FACH1 (13)						tsc_PCH1 (12)
PhyCh Type	Secondary CCPCH							Secondary CCPCH
PhyCH identity	tsc_S_CCPCH1 (5)							tsc_S_CCPCH2 (10)

8.3.21 Configuration of PS Cell_DCH_MAC_2AM_PS

The configuration is based on 3GPP TS 34.108 [3], clause 6.10.2.4.1.26. The RB0/UM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.3.2.1.2 with 2 AM RAB and RB0/TM-CCCH is referred to 3GPP TS 34.108 [3], clause 6.10.2.4.4.1.1.1. The configuration is applied to MAC test cases.

Table 77: Uplink configuration of Cell_DCH_MAC_2AM_PS

RB Identity	tsc_RB20 (20)	tsc_RB21 (21)	Same as uplink configuration of Cell_DCH_StandAloneSRB on DPCH	Same as uplink configuration of Cell_DCH_StandAloneSRB on PRACH
LogCh Type	DTCH	DTCH		
LogCh Identity	tsc_UL_DTCH1 (7)	tsc_UL_DTCH2 (8)		
RLC mode	AM	AM		
TrCH Type	DCH			
TrCH identity	tsc_UL_DCH1 (1)			
PhyCh Type	DPDCH			PRACH
PhyCH identity	tsc_UL_DPCH1 (20)			tsc_PRACH1 (8)

Table 78: Downlink configuration of Cell_DCH_MAC_2AM_PS

<table>
<tr><td>RB Identity</td><td>tsc_RB20
(20)</td><td>tsc_RB21
(21)</td><td rowspan="7">Same as downlink configuration of Cell_DCH_StandAloneSRB on DPCH</td><td rowspan="7">Same as downlink configuration of Cell_DCH_StandAloneSRB on sCCPCH</td></tr>
<tr><td>LogCh Type</td><td>DTCH</td><td>DTCH</td></tr>
<tr><td>LogCh Identity</td><td>tsc_DL_DTCH
1
(7)</td><td>tsc_DL_DTCH
2
(8)</td></tr>
<tr><td>RLC mode</td><td>AM</td><td>AM</td></tr>
<tr><td>MAC priority</td><td>1</td><td>1</td></tr>
<tr><td>TrCH Type</td><td colspan="2">DCH</td></tr>
<tr><td>TrCH identity</td><td colspan="2">tsc_DL_DCH1
(6)</td></tr>
<tr><td>PhyCh Type</td><td colspan="3">DPCH</td><td>Secondary CCPCH</td></tr>
<tr><td>PhyCH identity</td><td colspan="3">tsc_DL_DPCH1
(26)</td><td>tsc_S_CCPCH1
(5)</td></tr>
</table>

8.4 System information blocks scheduling

All SIBs specified in 3GPP TS 34.108 [3] are broadcast for all test cases in the present document. The repeat period of broadcasting of a complete SIB configuration is 64 frames (0,64 s) as the defualt configuration.

Except MIB and SB1, they have the highest scheduling rates, SIB 7 has also a higher scheduling rate.

According to the default SIB contents in 3GPP TS 34.108 [3], SIB 11 and SIB12 have 3 segments. SIB 5 and SIB 6 have 4 segments. MIB, SB1, SIB1, SIB 2, SIB 3, SIB 4, SIB 7 and SIB18 are not segmented, i.e. one segment for each. For the PDCP tests, SIB16 has 7 segments.

Use CMAC_SYSINFO_CONFIG_REQ, CMAC_SYSINFO_CONFIG_CNF and RLC_TR_DATA_REQ as interface to SS for broadcasting.

Two TSOs are defined, one for PER encoding function, the other for segmentation function. The TSOs shall be implemented in the tester.

8.4.1 Grouping SIBs for testing

<table>
<tr><td rowspan="2">Mandatory in 3GPP TS 34.108 [3]</td><td>Used in Idle Mode</td><td>MIB, SB1, (SB2), SIB1, SIB2, SIB3, SIB5, SIB7, SIB11</td></tr>
<tr><td>Used in Connected Mode</td><td>SIB4, SIB6, SIB12</td></tr>
<tr><td colspan="2">Mandatory for FDD CPCH</td><td>SIB8, SIB9</td></tr>
<tr><td colspan="2">Mandatory for FDD DRAC</td><td>SIB10</td></tr>
<tr><td colspan="2">Mandatory for TDD</td><td>SIB14, SIB17</td></tr>
<tr><td colspan="2">Mandatory for LCS</td><td>SIB15, SIB15.1, SIB15.2, SIB15.3</td></tr>
<tr><td colspan="2">Mandatory for ANSI-41 system</td><td>SIB13, SIB13.1, SIB13.2, SIB13.3, SIB13.4</td></tr>
<tr><td colspan="2">Mandatory for InterSys HO</td><td>SIB16</td></tr>
<tr><td colspan="2">Mandatory for Cell reselection</td><td>SIB18</td></tr>
</table>

8.4.2 SIB configurations

Currently the ATS contains three SIB configurations, Configuration 1 is default for both UTRAN/FDD SYSTEM and UTRAN/FDD. Configuration 2 is for test cases which need two S_CCPCH or two PRACH. Configuration 3 is for inter-RAT handover test cases.

Configuration 1	MIB, SB1, SIB1, SIB2, SIB3, SIB4, SIB5, SIB6, SIB7, SIB11, SIB12, SIB18
Configuration 2	MIB, SB1, SIB1, SIB2, SIB3, SIB4, SIB5, SIB7, SIB11, SIB12, SIB18
Configuration 3	MIB, SB1, SIB1, SIB2, SIB3, SIB4, SIB5, SIB7, SIB11, SIB16, SIB18

8.4.3 Test SIB default schedule

Frame No.	0	2	4	6	8	10	12	14
REP-POS	0	1	2	3	4	5	6	7
Block Type	MIB	SB1	SIB7	SIB6	MIB	SIB6	SIB6	SIB6

Frame No.	16	18	20	22	24	26	28	30
REP-POS	8	9	10	11	12	13	14	15
Block Type	MIB	SB1	SIB7/SIB3	SIB1/SIB2	MIB	SIB12	SIB12	SIB12

Frame No.	32	34	36	38	40	42	44	46
REP-POS	16	17	18	19	20	21	22	23
Block Type	MIB	SB1	SIB7/SIB18	SIB5	MIB	SIB5	SIB5	SIB5

Frame No.	48	50	52	54	56	58	60	62
REP-POS	24	25	26	27	28	29	30	31
Block Type	MIB	SB1	SIB7/SIB4		MIB	SIB11	SIB11	SIB11

SIB-repeat period (in frame)

Block Type	MIB	SB1	SIB1	SIB2	SIB3	SIB4	SIB5	SIB6	SIB7	SIB11	SIB12	SIB18
SIB Rep	8	16	64	64	64	64	64	64	16	64	64	64
Max. No of seg.	1	1	1	1	1	1	4	4	1	3	3	1

8.4.4 Test SIB special schedule

8.4.4.1 Test SIB schedule for two S-CCPCH or two PRACH

Frame No.	0	2	4	6	8	10	12	14
REP-POS	0	1	2	3	4	5	6	7
Block Type	MIB	SB1	SB1		MIB	SIB1	SIB18	SIB2

Frame No.	16	18	20	22	24	26	28	30
REP-POS	8	9	10	11	12	13	14	15
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB3		SIB4

Frame No.	32	34	36	38	40	42	44	46
REP-POS	16	17	18	19	20	21	22	23
Block Type	MIB	SB1	SB1	SIB5	MIB	SIB5	SIB5	SIB5

Frame No.	48	50	52	54	56	58	60	62
REP-POS	24	25	26	27	28	29	30	31
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB11	SIB11	SIB11

Frame No.	64	66	68	70	72	74	76	78
REP-POS	32	33	34	35	36	37	38	39
Block Type	MIB	SB1	SB1	SIB5	MIB	SIB5	SIB5	SIB5

Frame No.	80	82	84	86	88	90	92	94
REP-POS	40	41	42	43	44	45	46	47
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB3		SIB4

Frame No.	96	98	100	102	104	106	108	110
REP-POS	48	49	50	51	52	53	54	55
Block Type	MIB	SB1	SB1		MIB			

Frame No.	112	114	116	118	120	122	124	126
REP-POS	56	57	58	59	60	61	62	63
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB12	SIB12	SIB12

SIB-repeat period (in frame)

Block Type	MIB	SB1	SIB1	SIB2	SIB3	SIB4	SIB5	SIB7	SIB11	SIB12	SIB18
SIB Rep	8	16	128	128	64	64	128	32	128	128	128
Max. No of seg.	1	2	1	1	1	1	8	1	3	3	1

8.4.4.2 Test SIB schedule for Inter-Rat Handover Test

Frame No.	0	2	4	6	8	10	12	14
REP-POS	0	1	2	3	4	5	6	7
Block Type	MIB	SB1	SB1		MIB	SIB1	SIB18	SIB2

Frame No.	16	18	20	22	24	26	28	30
REP-POS	8	9	10	11	12	13	14	15
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB3		SIB4

Frame No.	32	34	36	38	40	42	44	46
REP-POS	16	17	18	19	20	21	22	23
Block Type	MIB	SB1	SB1	SIB5	MIB	SIB5	SIB5	SIB5

Frame No.	48	50	52	54	56	58	60	62
REP-POS	24	25	26	27	28	29	30	31
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB11	SIB11	SIB11

Frame No.	64	66	68	70	72	74	76	78
REP-POS	32	33	34	35	36	37	38	39
Block Type	MIB	SB1	SB1	SIB16	MIB	SIB16	SIB16	SIB16

Frame No.	80	82	84	86	88	90	92	94
REP-POS	40	41	42	43	44	45	46	47
Block Type	MIB	SB1	SB1	SIB7	MIB	SIB3		SIB4

Frame No.	96	98	100	102	104	106	108	110
REP-POS	48	49	50	51	52	53	54	55
Block Type	MIB	SB1	SB1	SIB16	MIB	SIB16	SIB16	SIB16

Frame No.	112	114	116	118	120	122	124	126
REP-POS	56	57	58	59	60	61	62	63
Block Type	MIB	SB1	SB1	SIB7	MIB			

SIB-repeat period (in frame)

Block Type	MIB	SB1	SIB1	SIB2	SIB3	SIB4	SIB5	SIB7	SIB11	SIB16	SIB18
SIB Rep	8	16	128	128	64	64	128	32	128	128	128
Max. No of seg.	1	2	1	1	1	1	4	1	3	8	1

8.4.5 Handling the transmission of SIB

According to the SIB repeat periods, SIBs need to be transmitted on a very regular basis during the operation of a test case. This transmission usually has no direct bearing on the operation of the test case, although the carried information ensures the correct configuration and operation of the UE during the test case.

To send this information repeatedly directly from each test case would make the test cases very complex to implement, difficult to understand and place real-time requirements upon them that are beyond the capabilities of most TTCN driven test engines.

Management of scheduling of System Information messages is performed by the system simulator. The SIB contents, usually determined in part by the individual tests, come from the TTCN test cases.

8.4.5.1 Delivery of System Information content

The content of the System Information messages is delivered as a fully encoded bit string to the TM-RLC SAP from the message content defined in the TTCN test case.

The IE 'SFNprime' in the SI messages is set to 0 by the TTCN, and the correct value of 'SFNprime' shall be inserted by the System Simulator prior to transmission of a SI message.

SI messages are ASN.1 packed encoded through a TTCN TSO and segmented another TTCN TSO into SIBs in the TTCN and sent only once to the TM-RLC SAP. Repetition of the SIB is the responsibility of the System Simulator lower layers.

SIBs are considered to be cached. That is, sending a SIB to the TM-RLC SAP will cause a previously sent copy of the SIB to be lost, and all future transmissions of the SIB will be the most recently sent version. This allows for the updating of System Information during the operation of a test case.

8.4.5.2 Scheduling of System Information Blocks

The schedule for the transmission of SIBs is provided by the TTCN test case. It is sent using the CMAC_SYSINFO_CONFIG_REQ primitive sent to the CMAC SAP (CMAC_PCO).

Each CMAC_SYSINFO_CONFIG_REQ primitive carries scheduling information for the next SIB sent from the TTCN. Each primitive is followed by an associated SIB. Sending two CMAC_SYSINFO_CONFIG_REQ primitives in succession may cause an unspecified result.

8.4.5.3 Example of usage

The following example shows how the MIB, SB1 and all SIBs in subclause 8.4.3 are sent to the System Simulator lower layers for broadcasting. The 1st parameter in CMAC_SYSINFO_CONFIG_REQ represents the repeat period in power of 2. The 2nd parameter represents the repetition position. Two consecutive frames represent an available repetition position.

```
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (3, 0)
TM_PCO:     MIB
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (4, 1)
TM_PCO:     SB1
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 2)
TM_PCO:     SIB7
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 3)
TM_PCO:     SIB6 (segment 1 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 5)
TM_PCO:     SIB6 (segment 2 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 6)
TM_PCO:     SIB6 (segment 3 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 7)
TM_PCO:     SIB6 (segment 4 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 10)
TM_PCO:     SIB7 + SIB3 (concatenation)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 11)
TM_PCO:     SIB1 + SIB2 (concatenation)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 13)
TM_PCO:     SIB12 (segment 1 of 3)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 14)
TM_PCO:     SIB12 (segment 2 of 3)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 15)
TM_PCO:     SIB12 (segment 3 of 3)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 18)
TM_PCO:     SIB7 + SIB18 (concatenation)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 19)
TM_PCO:     SIB5 (segment 1 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 21)
TM_PCO:     SIB5 (segment 2 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 22)
TM_PCO:     SIB5 (segment 3 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 23)
TM_PCO:     SIB5 (segment 4 of 4)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 26)
TM_PCO:     SIB7 + SIB4 (concatenation)
CMAC_PCO:   CMAC_SYSINFO_CONFIG_REQ (6, 27)
TM_PCO:     No segment
```

CMAC_PCO: CMAC_SYSINFO_CONFIG_REQ (6, 29)
TM_PCO: SIB11 (segment 1 of 3)
CMAC_PCO: CMAC_SYSINFO_CONFIG_REQ (6, 30)
TM_PCO: SIB11 (segment 3 of 3)
CMAC_PCO: CMAC_SYSINFO_CONFIG_REQ (6, 31)
TM_PCO: SIB11 (segment 3 of 3)

8.5 Security in testing

The security functions at the SS side are implemented in RLC and MAC layers. When the AM or UM RLC entities and a MAC(d) entity are created, the TTCN will download a security context for each CN domain used. The two ASPs CMAC_SecurityMode_Config_REQ & CRLC_SecurityMode_Config_REQ configues the SS security contexts and associate the contexts to the created entities. The SS sahll support one activate security contexts and one context pending activation for each CN domain.

A security context at the SS consists of the security parameter START, 20 bits long and a pair of integrity key and a ciphering key, each 128 bits long. All these security parameters belong to a CS or a PS domain. The SS shall have the ability to store these values till the new vlaues are downloaded and activated. $START_{cs}$ is used for intitialisation of all counters-C and counters-I (32 bits long each) of all DL and UL radio bearers for ciphering and intergrity protection in the CS domain. The same is for $START_{ps}$ in the PS domain. The TTCN downloads the new START value whenever it is received from the UE. In the case of a succeeded authentication procedure, the START value is reset to zero by the TTCN.

Once the START is downloaded the SS inialises the 20 most significant bits of the RRC HFN (for integrity protection), the RLC HFN (for ciphering) and the MAC-d HFN (for ciphering) to the START value of the corresponding service domain; the remaining bits are initialised to 0.

Upon the concerned RLC entities and the MAC(d) entity release in the SS, the associated security contexts are no longer used and shall be removed as well. The RLC and the MAC(d) entities are addressed by the TTCN with the cell id = -1.

8.5.1 Authentication

A GMM or MM authentication test step makes use of a number of TSOs to generate an authentication vector:

AV := {RAND, XRES, CK, IK, AUTN}

8.5.2 Ciphering

The ciphering in the SS is activated through the ASP CRLC_Ciphering_Activate_REQ for the AM or UM mode and through CMAC_Ciphering_Activate_REQ for the TM mode.

8.5.3 Integrity

The integrity protection in the SS is activated through the ASP CRLC_Integrity_Activate_REQfor all SRB.

8.5.4 Counter check

TBD

8.5.5 Test USIM configurations

The default test USIM is defined in 3GPP TS 34.108 [3]. This clause specifies a number of specific test USIM configurations which are used for the concerned test cases.

8.5.5.1 Test USIM for Idle mode tests

The PLMN 1-12 identities used below have been defined in 3GPP TS 34.123-1 [1], table 6.2. Clause numbers refer to 3GPP TS 34.123-1 [1].

Test USIM for PLMN selection of RPLMN, HPLMN, UPLMN and OPLMN in TC_6_1_1_1 and TC_6_1_1_4.

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}		PLMN 1	
$EF_{HPLMNwAcT}$	1^{st}	PLMN 2	UTRAN
$EF_{PLMNwAcT}$	1^{st}	PLMN 3	UTRAN
	2^{nd}	PLMN 4	UTRAN
$EF_{OPLMNwAcT}$	1^{st}	PLMN 5	UTRAN
	2^{nd}	PLMN 6	UTRAN
EF_{FPLMN}	PLMN 3		

Test USIM for PLMN selection of PLMN selection of Other PLMN with access technology combinations in TC_6_1_1_2 and TC_6_1_1_5.

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}		PLMN 1	
$EF_{HPLMNwAcT}$	1^{st}	PLMN 2	UTRAN
$EF_{PLMNwAcT}$	1^{st}	PLMN 3	UTRAN
	2^{nd}	PLMN 4	UTRAN
$EF_{OPLMNwAcT}$	1^{st}	PLMN 5	UTRAN
	2^{nd}	PLMN 6	UTRAN
EF_{FPLMN}	PLMN 10		

Test USIM for PLMN selection of PLMN selection of PLMN selection; independence of RF level and preferred PLMN; Manual mode in TC_6_1_1_3.

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}			
$EF_{HPLMNwAcT}$	1^{st}	PLMN 1	UTRAN
$EF_{PLMNwAcT}$	1^{st}	PLMN 3	UTRAN

Test USIM for emergency calls requires that all the BCCH cells belong to the same PLMN, which is not the UE's home PLMN and is in the USIM's forbidden PLMN's list. The test USIM applies to TC_6_1_2_6.

Test USIMs for Selection of the correct PLMN and associated RAT in TC_6_2_1_1. Two test USIMs are needed for the test.

USIM A:

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}			
$EF_{HPLMNwAcT}$	1^{st}	PLMN 1	GSM
	2^{nd}		UTRAN

USIM B:

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}			
$EF_{HPLMNwAcT}$	1^{st}	PLMN 2	UTRAN
	2^{nd}		GSM

Test USIMs for Selection of RAT for HPLMN in TC_6_2_1_2 and TC_6_2_1_6. Two test USIMs are needed for the test.

USIM A:

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}		PLMN 1	
$EF_{HPLMNwAcT}$	1st	PLMN 2	UTRAN
	2nd		GSM

USIM B:

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}		PLMN 1	
$EF_{HPLMNwAcT}$	1st	PLMN 2	UTRAN
	2nd		

Test USIM for Selection of RAT for UPLMN or OPLMN in TC_6_2_1_3, TC_6_2_1_4, TC_6_2_1_7, TC_6_2_1_8 and for Selection of Other PLMN with access technology combinations"; Automatic mode in TC_6_2_1_9.

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}		PLMN 1	
$EF_{HPLMNwAcT}$	1st	PLMN 2	UTRAN
	2nd		GSM
$EF_{PLMNwAcT}$	1st	PLMN 3	UTRAN
	2nd	PLMN 4	GSM
$EF_{OPLMNwAcT}$	1st	PLMN 5	UTRAN
	2nd	PLMN 6	GSM

Test USIM for Selection of Other PLMN with access technology combinations"; Manual mode in TC_6_2_1_5.

USIM field	Priority	PLMN	Access Technology Identifier
EF_{LOCI}		PLMN 1	
$EF_{HPLMNwAcT}$	1st	PLMN 2	UTRAN
	2nd		GSM
$EF_{PLMNwAcT}$	1st	PLMN 3	UTRAN
	2nd	PLMN 4	GSM
$EF_{OPLMNwAcT}$	1st	PLMN 5	UTRAN
	2nd	PLMN 6	GSM
EF_{FPLMN}	PLMN 7		
	PLMN 12		

Test USIM for Cell reselection if cell becomes barred or for Cell reselection timings requires that the USIM does not contain any preferred RAT. The test USIM applies to TC_6_2_2_1, TC_6_2_2_2 and TC_6_2_2_3.

8.6 Downlink power control in SS

TBD

8.7 Test suite operation definitions

8.7.1 Test suite operation definitions in the module BasicM

Table 79: TSO definitions in BasicM

TSO Name	Description
o_AuthRspChk	**Type of the result:** BOOLEAN **Parameters:** p_AuthRsp : AuthRsp

TSO Name	Description

	p_AuthRspExt : AuthRspExt
	p_K : BITSTRING
	p_RAND : BITSTRING
	p_Ext : BOOLEAN
	
	Description
	Checks the input parameter p_AuthRsp and p_AuthRspExt, both received in an Authentication Response, according to the authentication algorithm defined in the following procedure.
	The extension, p_AuthRspExt, is optional. Its presence is indicated by p_Ext.
	Returns TRUE if the Authentication Response contained in parameters p_AuthRsp and eventually p_AuthRspExt is correct, FALSE otherwise.
	The value of tcv_Auth_n indicates whether the AuthRspExt has been provided by the UE or not (n=31, or 31 < n < 128). See 3GPP TS 34.108 [3] clause 8.1.2.
	If not the parameter p_AuthRspExt is not to be used.
	
	Algorithm (without the knowledge of tcv_Auth_n):
	==
	if NOT p_Ext EvaluateAuthRsp else EvaluateAuthRspAndAuthRspExt
	EvaluateAuthRsp:
	==============
	resultbitstring = o_BitstringXOR(XRES, AuthRsp)
	if resultbitstring is all 0s then there is a match.
	
	EvaluateAuthRspAndAuthRspExt:
	============================
	XREShigh = o_BitstringXtract(XRES, 32, 32, 0)
	/* XRES divides into 2 parts: the higher part of 32 bits related to AuthRsp and the lower part related to AuthRspExt */
	/* SourceLength of 32 is only to ensure usage of the procedure */
	resultbitstring = o_BitstringXOR(XREShigh, AuthRsp)
	if resultbitstring is all 0s then there is a match for the first 32 bits:EvaluateAuthRspExt else Authentication failed.
	
	EvaluateAuthRspExt:
	=================
	/* As AuthRespExt may not be octet aligned the last octet indicated in AuthRspExt is not used for checking */
	if (AuthRspExt.iel = 1)
	then Authentication passed
	/* there was only 1 possibly incomplete octet which is not used */
	else
	{
	AuthRspExthigh = o_BitstringXtract(AuthRspExt.authRsp, ((AuthRspExt.iel -1)* 8), (AuthRspExt.iel -1)* 8, 0)
	/* extract (AuthRspExt.iel -1)* 8 bits starting from bit 0 */
	XRESlow = o_BitstringXtract(XRES, ((AuthRspExt.iel -1)* 8 + 32), (AuthRspExt.iel -1)* 8, 32)
	/* extract (AuthRspExt.iel -1)* 8 bits starting from bit 32 */
	resultbitstring = o_BitstringXOR(XRESlow, AuthRspExthigh, (AuthRspExt.iel -1)* 8)
	if resultbitstring is all 0s then there is a match for the bits following the first 32 bits else Authentication failed

TSO Name	Description
o_BCD_ToInt	**Type of the result:** INTEGER **Parameters:** p_bcdstring:HEXSTRING **Description** The operation OC_BCDtoInt converts an HEXSTRING containing BCD coded digits to an integer representation of these relevant digits. EXAMPLE: OC_BCDtoInt('12345'H) := 12345
o_BitstringChange	**Type of the result:** BITSTRING **Parameters:** P_Str: BITSTRING p_Len: INTEGER p_Offset: INTEGER **Description** Performs the manipulation of a bitstring by toggling the bit identified by p_Offset. The length of the string to be manipulated is specified in p_Len. This is only provided to help ensure that the p_Offset is less than p_Len. Returns a resulting bitstring of length p_Len. Examples: o_BitstringChange('010101'B, 6, 5) produces '010100'B. o_BitstringChange('010101'B, 6, 0) produces '110101'B.
o_BitstringConcat	**Type of the result:** BITSTRING **Parameters:** P_Str1: BITSTRING p_Str2: BITSTRING p_Len1: INTEGER p_Len2: INTEGER **Description** Performs the concatenation of 2 bitstrings of possibly different lengths. The bit significance is from left to right, ie the MSB is at the lefthand side. Returns a resulting bitstring p_Str1 || p_Str2 of length p_ Len1 + p_Len. Example: o_BitstringConcat('010101'B,'11'B) produces '01010111'B of length 6 + 2 = 8.
o_BitstringXOR	**Type of the result:** BITSTRING **Parameters:** P_Str1: BITSTRING p_Str2: BITSTRING p_Len: INTEGER **Description** Performs an XOR operation using 2 bitstrings of the same length (p_Len). Returns a resulting Bitstring of length p_Len. Example: o_BitstringXOR('0011'B, '0101'B, 4) produces '0110'B
o_BitstringXtract	**Type of the result**: BITSTRING **Parameters:** P_Str: BITSTRING p_SrcLen: INTEGER p_TargetLen: INTEGER p_Offset: INTEGER **Description** Performs the wrap around extract of a bitstring. The length of the string from which extraction is to be made is specified in p_SrcLen. The length of the bitstring to be extracted is indicated as p_TargetLen, the offset in the original string is indicated in p_Offset. The bit position 0 is at the left, the MSB is at the righthand side. Returns a resulting bitstring of length p_TargetLen. Examples: o_BitstringXtract('101010'B, 6, 2, 1) produces '01'B. o_BitstringXtract('101010'B, 6, 4, 3) produces '0101'B, wrapping around. o_BitstringXtract('111000'B, 6, 4, 3) produces '0111'B, wrapping around.

TSO Name	Description
o_BitToOct	**Type of the result**: OCTETSTRING **Parameters:** p_Str: BITSTRING **Description** This TSO is used to convert the given BITSTRING into an OCTETSTRING. If the bitstring length is not a multiple of 8, 1 to 7 padding bits are added at the end to fill the final octet.
o_BMC_DrxScheduling	**Type of the result**: BMC_ResultOfSchedulingLevel2 **Parameters:** p_BMC_CBS_Message1 : BMCCBSMESSAGE p_BMC_CBS_Message2 : BMCCBSMESSAGE p_BMC_CB_RepPeriod : INTEGER p_BMC_NoOfBroadcast_Req : INTEGER p_Offset : BMC_DRX_Offset **Description** This TSO shall calculate all BMC CBS schedule Messages for the CBS messages as described in 3GPP TS 34.123-1, clause 7.4.3.1. The TSO has to precalculate the CTCH Block SETs needed, i.e. it shall have all necessary knowledge (RLC segmentation, MAC handling, if needed) to predict the CTCH with BMC contents for the given input to be sent. The TSO shall consider the BMC CBS Scheduling Level2 as described in 3GPP TS 25.324 [20], 3GPP TR 21.925 [44] and the description of BMC test architecture and test method in the present document, clause 6.8. The TSO calculates the BMC CBS Schedule messages to predict its next BlockSet to be sent. In addition, a DRX scheduling Bitmap is created for each CTCH allocated TTI alligned to the pre-calculated offset in between 2 CTCH Block Sets. The prinziple of DRX shall be followed by this TSO. I.e. BMC Messages shall be sent blockwise (CTCH Block Set) with predicted offset in between 2 Block Sets. The TSO shall consider the following aspects to calculate the DRX Selection Bitmap and to create the BMC CBS Schedule messages: 1. The first CTCH Block Set consists of the first BMC CBS Schedule message predicting the offset, length and content of the following Block Set where the BMC CBS Message1 shall be send as new message. 2. The BMC CBS Message1 shall be repeated for p_BMC_CB_RepPeriod multiplied by p_BMC_NoOfBroadcast_Req times before the BMC CBS Message2 is broadcasted. 3. The BMC CBS Schedule Messages shall be the last message of a CTCH Block Set, i.e. on the end of a Block Set. 4. If no further repetition of BMC CBS Messages is needed, no further BMC CBS Schedule message shall be created. output parameter: DrxSelectionBitmap: The TSO creates a Bitmap as Octetstring for scheduled CTCH allocated TTI as described in 3GPP TS 34.123-3: clause 6.8.2 BMC test method and architecture. CBS_Schedule_Message01, CBS_Schedule_Message02, CBS_Schedule_Message03:Considering the given BMC PDUs BMC_DRX_Offset and BMCCBSMESSAGE to be sent, the BMC Schedule messages have to be created according the given parameter.

TSO Name	Description
o_CheckStringStartWith	**Type of the result:**BOOLEAN **Parameters:** p_SourceString: IA5String p_StartString : IA5String **Description** o_CheckStringStartWith returns TRUE if the p_sourceString start with the p_StartString. Otherwise it returns FALSE. EXAMPLE: o_CheckStringStartWith ("+CLCC:1,0,0,2,0;", "+CLCC:1,0,0")=TRUE */
o_ComputeSM_Contents	**Type of the result:** OCTETSTRING **Parameters:** p_NumOfChars: INTEGER **Description** This operation provides a short message's contents with a specified number of characters 'p_NumOfChars', each represented by 7 bits. As possibly different characters are sent, the characters are those corresponding to the 7-bit representation of 0, 1, 2, ... up to ('p_NumOfChars' - 1). If more than 128 characters are sent, the rest of the characters is the corresponding to 0, 1, ... up to ('p_NumOfChars' - 128 - 1), e.g. for 160 characters: 0, 1, ..., 127, 0, 1, ..., 31. The bits are arranged acc. to 3GPP TS 23.038 [34], clause 6.1.2.1.1. max. 160 characters, i.e. 140 octets.
o_ComputeSM_ContentsSpec	**Type of the result:** OCTETSTRING **Parameters:** p_NumOfChars: INTEGER p_Text: IA5String **Description** This operation provides a short message's contents with a specified number of characters 'p_NumOfChars', each represented by 7 bits. 'p_Text' is used as contents of the short message. If 'p_Text' contains less than 'p_NumOfChars' characters, 'p_Text' is repeated until the short message reaches the 'p_NumOfChars' characters long.The bits are arranged acc. to 3GPP TS 23.038 [34], clause 6.1.2.1.1. max. 160 characters, i.e. 140 octets.
o_ConcatStrg	**Type of the result:** IA5String **Parameters:** P_String1: IA5String p_String2: IA5String **Description** o_ConcatString concatenates 'p_String1' and 'p_String2' and returns the resulting string. EXAMPLE: o_ConcatString ("AT+CBST=0" , ",0") = "AT+CBST=0,0"
o_ConvertIMSI	**Type of the result:** IMSI_GSM_MAP **Parameters:** P_Imsi : HEXSTRING The input parameter \`p_Imsi\` is a BCD string (subset of HEXSTRING), the result is of type IMSI_GSM_MAP.
o_ConvertTMSI	**Type of the result:**TMSI_GSM_MAP **Parameters:** p_Tmsi : OCTETSTRING **Description** The input parameter 'p_Tmsi' is an OCTETSTRING; the result is of type TMSI_GSM_MAP.
o_ConvertPTMSI	**Type of the result:** P_TMSI_GSM_MAP **Parameters:** p_PTMSI : OCTETSTRING **Description** The input parameter \`PTMSI\` is a OCTETSTRING, the result is of type P_TMSI_GSM_MAP.

TSO Name	Description
o_ConvtPLMN	**Type of the result:**TMSI_GSM_MAP **Parameters:** OCTETSTRING p_MCC, p_MNC : HEXSTRING **Description** the functions of o_ConvtPLMN are as following: 1. The least significant HEX of p_MNC is removed from p_MNC and inserted into p_MCC in the position left to the third HEX to form a new p_MCC of 4 HEXs, then swap the first HEX (left most, most siginificant Hex) with the second HEX of the new p_MCC. 2. Swap the first Hex with the second HEX of the remaining part of p_MNC and append it to the new p_MCC formed in Step1 above. EXAMPLE 1: o_ConvtPLMN('123'H, '456'H) = '216354'O EXAMPLE 2: o_ConvtPLMN ('234'H, '01F'H) = '32F410'O
o_ConvtAndConcatStr	**Type of the result:**OCTETSTRING **Parameters:** p_MCC, p_MNC : HEXSTRING; p_LAC : OCTETSTRING; p_RAC : OCTETSTRING **Description** functions of o_ConvtAndConcatStr are as following: 1. The least significant HEX of p_MNC is removed from p_MNC and inserted into p_MCC in the position left to the third HEX to form a new p_MCC of 4 HEXs, then swap the first HEX (left most, most siginificant Hex) with the second HEX of the new p_MCC. 2. Swap the first Hex with the second HEX of the remaining part of p_MNC and append it to the new p_MCC formed in Step1 above. 3. Append p_LAC to the result of Step 2, this is the final result if p_RAC is omitted. 4. Append p_RAC to the result of Step 3, this is the final result. NOTE 1: Steps 1 and 2 are identical to o_ConvtPLMN. NOTE 2: If p_RAC is omitted, 5 octets of Location Area Identification are produced (for SysInfo sending). If p_RAC is not omitted, 6 octets of Routing Area Identification are produced (for SysInfo sending). EXAMPLE 1: o_ConvtAndConcatStr ('123'H, '456'H, '0001'O, '01'O) = '216354000101'O EXAMPLE 2: o_ConvtAndConcatStr ('234'H, '01F'H, '0005'O, OMIT) = '32F4100005'O
o_DrawRandomNo	**Type of the result:** INTEGER **Parameters:** p_LowerBound, p_UpperBound: INTEGER **Description** This operation draws a random number in the range of p_LowerBound and p_UpperBound.The result is in the range p_LowerBound, p_LowerBound+1, ..., p_UpperBound.
o_FirstDigit	**Type of the result:** B4 **Parameters:** p_BCDdigits : HEXSTRING **Description** The input parameter p_BCDdigits shall be a BCD string (subset of HEXSTRING), the resut is a BITSTRING[4] of a binary representation of one BCD digit. The function of the o_FirstDigit is to return the first (most significant) digit of the input parameter 'p_BCDdigits'. EXAMPLE 1: o_FirstDigit('12345') = '0001'B, EXAMPLE 2: o_FirstDigit('012345678') = '0000'B.

TSO Name	Description
o_GetBit	**Type of the result:** BITSTRING **Parameters:** p_Source: BITSTRING p_DataLength:INTEGER **Description** o_GetBit returns the BITSTRING of length p_DataLength extracted from p_Source.
o_GetN_OctetsFromPRBS	**Type of the result:**OCTETSTRING **Parameters:** p_Start, p_N: INTEGER **Description** This operation returns N octets from a repeated pseudo random bit sequence, starting with octet position p_Start. The PRBS is the 2047 bit pseudo random test pattern defined in ITU-T Recommendation O.153 [45] for measurements at 64 kbit/s and N x 64 kbit/s o_GetN_OctetsFromPRBS(p_Start, p_N) generates an OCTETSTRING containing p_N octets starting from octet number p_Start in the PRBS. Requirements p_Start >= 0 p_N >= 1 Definition Define the 2 047 bit PRBS sequence b(i) as an m-sequence produced by using the following primitive (over GF(2)) generator polynomial of degree 11: X^11 + X^9 + 1 This sequence is defined recursively as: b(i) = 1 , i = 0,1,...,10 b(i) = b(i - 2) + b(i - 11) modulo 2 , i = 11,16,...,2046 The OCTETSTRING, o(j) generated by the present TSO is produced by extracting p_N octets from the repeated sequence b(i) as follows: o(j,k) = b(((n_Start + j) * 8 + k) modulo 2047) where: j = 0,1,..,p_N - 1 k = 0,1,..7 o(j,k) is the kth bit of the jth octet in o(j), o(j,0) is the MSB of the jth octet in o(j), o(j,7) is the LSB of the jth octet in o(j), Example results: o_GetN_OctetsFromPRBS(0, 25) and o_GetN_OctetsFromPRBS(2047, 25) both return: 'FFE665A5C5CA3452085408ABEECE4B0B813FD337873F2CD1E2'O o_GetN_OctetsFromPRBS(255, 25) and o_GetN_OctetsFromPRBS(255 + 2047, 25) both return '01FFCCCB4B8B9468A410A81157DD9C9617027FA66F0E7E59A3'O
o_GetPI	**Type of the result:** BITSTRING **Parameters:** p_Imsi : **HEXSTRING** p_Np: INTEGER **Description** The PI is calculated as following: PI = drx_index mod np The drx_index is calculated as described hereafter: drx_index = (p_Imsi / 8192) This calculation is defined in 3GPP TS 25.304 [16] clause 8.3. NOTE: The IMSI is passed as HEXSTRING, the relevant conversion shall be done.

TSO Name	Description
o_GetSC_TimeStamp	**Type of the result:** TP_ServCentreTimeSt **Parameters:** p_timezone : TZONES This operation provides the hexstring containing the service center time stamp (SCTS) according to 3GPP TS 23.040 [35], clauses 9.2.2.1 and 9.2.3.11. The TSO reads the current time of the test systems clock and transforms the time in combination with the input parameter 'timezone' into a service center time stamp. Example: 2002 April 18, 15:32:46, timezone=4 o_GetSC_TimeStamp returns 20408151236440 TPSCTS is HEXSTRING[14]
o_HexToDigitsMCC	**Type of the result:**MCC **Parameters:** p_BCDdigits : HEXSTRING **Description** The input parameter p_BCDdigits shall be a BCD string (subset of HEXSTRING), the result is a SEQUENCE (SIZE(3)) OF digit (MCC). NOTE: The length of p_BCDdigits shall be 3. User shall take the responsibility of fulfilling this requirement. For example: o_HexToDigitsMCC('111'H) = {1, 1, 1} o_HexToDigitsMCC('123'H) = {1, 2, 3}.
o_HexToDigitsMNC	**Type of the result:**MNC **Parameters:** p_BCDdigits : HEXSTRING **Description** The function of this operation is: 1. The least significant HEX is removed if it is 'F' and the operation returns SEQUENCE (SIZE(2)) OF Digit. 2. The operation returns SEQUENCE (SIZE(3)) OF Digit if all 3 HEX digits in p_BCDdigits are BCD Digit. EXAMPLE 1: o_HexToDigitsMNC('123'H) = {1, 2, 3} EXAMPLE 2: o_HexToDigitsMNC('13F'H) = {1, 3}.
o_HexToIA5	**Type of the result:** IA5String **Parameters:** p_String: HEXSTRING **Description** o_HEX_TO_IA5 converts hexadecimal string 'p_String' to an IA5 String For example: o_HEX_TO_IA5 ('15A'H) = "15A"
o_IA5_ToOct	**Type of the result:**OCTETSTRING **Parameters:** p_String : IA5String **Description** o_IA5_ToOct converts the string p_String from IA5String type to OCTETSTRING. Each character is mapped onto an octet, and bit 8 is set to 0. This TSO shall be used to convert Access Point Numbers for example. See 3GPP TS 24008, clause 10.5.6.1 EXAMPLE: o_IA5_ToOct ("15A") = '313541'O

<table>
<tr><th>TSO Name</th><th>Description</th></tr>
<tr><td>o_IA5_BMC_ToOct</td><td>Type of the result:OCTETSTRING
Parameters:
p_String :IA5String_BMC
p_DCS: TP_DataCodingScheme

Description
o_IA5_BMC_ToOct converts the string p_String from IA5String_BMC type to OCTETSTRING.
p_DCS determines how this is done (refer to 3GPP TS 23.038 [34] clause 5).
If a 7 bit packing is to be applied then proceed as described in 3GPP TS 23.038 [34] clause 6.1.2.2.1 and clause 6.2.1. This is the default case.

If 8bit data is to be used then proceed as described in 3GPP TS 23.038 [34] clause 6.2.2.
If UCS2is to be used then proceed as described in 3GPP TS 23.038 [34] clause 6.2.3.

The type IA5_BMC implies that the length of p_String is restricted to 1 246 octets.
(Refer to 3GPP TS 23.041 [36], 3GPP TS 23.038 [34], 3GPP TS 25.324 [20])

EXAMPLE 1: o_IA5_ BMC_ToOct ("15A", '0F'O) = 'B15A10'O ('0F'O is the default codepoint, GSM 7 bit packed).
EXAMPLE 2: o_IA5_ BMC_ToOct ("15A", '00'O) = 'B15A10'O (German Language, GSM 7 bit packed).
EXAMPLE 3: o_IA5_ BMC_ToOct ("15A", '01'O) = 'B15A10'O (English Language, GSM 7 bit packed).
EXAMPLE 4: o_IA5_ BMC_ToOct ("15A", 'F0'O) = 'B15A10'O (Data coding, no msg class, GSM 7 bit packed).
EXAMPLE 5: o_IA5_ BMC_ToOct ("15A", 'F1'O) = 'B15A10'O (Data coding, class 1, GSM 7 bit packed).
EXAMPLE 6: o_IA5_ BMC_ToOct ("15A", 'F2'O) = <8 bit data is user defined> (Data coding, no msg class, 8 bit data).</td></tr>
<tr><td>o_IA5_IP_ToOct</td><td>Type of the result:OCTETSTRING
Parameters:
p_String: IA5String
p_IP_V4: BOOLEAN

Description
o_IA5_IP_ToOct converts the string p_String from IA5String type to OCTETSTRING.
p_String represents an IP address consisting of a number of fields of digits, separated by dots. Each one of the numbers of which the IP address consists is converted into one octet. The dots separating the numbers are ignored.
p_IP_V4 is a BOOLEAN. When TRUE, an IP Version 4 address is to be converted, the maximum length of which is 4 octets, otherwise an IP Version 6 address is to be converted, the maximum length of which is 16 octets. See 3GPP TS 24.008 [9], clause 10.5.6.4.

EXAMPLE 1: o_IA5_IP_ToOct ("200.1.1.80", TRUE) = 'C8010150'O.
EXAMPLE 2: o_IA5_IP_ToOct ("200.1.1.80.100", TRUE) should result in an appropriate error message.
EXAMPLE 3: o_IA5_IP_ToOct ("300.1.1.80", TRUE) should result in an appropriate error message.</td></tr>
<tr><td>o_IA5_DigitsToOct</td><td>Type of the result:OCTETSTRING
Parameters:
p_String: IA5String

Description
o_IA5_DigitsToOct converts the string p_String from IA5String type to OCTETSTRING.
Each pair of characters is considered a pair of numbers to be mapped onto 1 octet.
Each character of p_String shall represent a digit (0..9).
In case the number of characters is odd, then a filler '1111'B is used to fill the last octet required to represent the digits. See 3GPP TS 24.008 [9], clause 10.5.4.7.

EXAMPLE 1: o_IA5_DigitsToOct ("0613454120") = '6031541402'O.
EXAMPLE 2: o_IA5_DigitsToOct ("06134541209") = '6031541402F9'O.
EXAMPLE 3: o_IA5_DigitsToOct ("A6134541209") should result in an appropriate error message.</td></tr>
</table>

TSO Name	Description
o_IntToOct	**Type of the result:**OCTETSTRING **Parameters:** p_N : INTEGER p_L: INTEGER **Description** o_IntToOct converts the INTEGER `p_N` into OCTETSTRING with length = 'p_L'. EXAMPLE 1: o_IntToOct(14,1) = '0E'O. EXAMPLE 2: o_IntToOct(18,1) = '12'O. EXAMPLE 3: o_IntToOct(18,2) = '0012'O.
o_IntToIA5	**Type of the result:**IA5String **Parameters:** p_N : INTEGER; p_L: INTEGER **Description** o_IntToIA5 converts the INTEGER `p_N` into IA5 String with length = 'p_L'. EXAMPLE 1: o_IntToIA5(160,3) = "160"; EXAMPLE 2: o_IntToIA5(160,4) = " 160"; EXAMPLE 3: o_IntToIA5(160,2) = "60".
o_OctetstringConcat	**Type of the result:**OCTETSTRING **Parameters:** p_Str1, p_Str2: OCTETSTRING **Description** o_OctetstringConcat Performs the concatenation of 2 octetstrings of possibly different lengths. The octet significance is from left to right, i.e. the MSB is at the lefthand side. Returns a resulting octetstring p_Str1 || p_Str2. EXAMPLE: o_OctetstringConcat('135'O, '9A38'O) = '1359A38'O.
o_OctToBit	**Type of the result:** BITSTRING **Parameters:** p_octetStr: OCTETSTRING **Description** Converts an OCTETSTRING into a BITSTRING. The size of the resulting BITSTRING is 8 times the size of the input OCTETSTRING.
o_OctToInt	**Type of the result:** INTEGER **Parameters:** p_oct : OCTETSTRING **Description** Transform an OCTETSTRING of length 1 to 4 into an unsigned 32 bits IINTEGER value. If the input octet string is larger than 4, then only the first 4 octets shall be considered.
o_OctToIA5	**Type of the result:** IA5String **Parameters:** p_String: OCTETSTRING **Description** o_OctToIA5 converts hexadecimal string 'p_String' to an IA5 String EXAMPLE: o_OctToIA5 ('2A15AF'O) = "2A15AF".

TSO Name	Description
o_OeBit	**Type of the result:**BITSTRING **Parameters:** p_BCDdigits: HEXSTRING **Description** The input parameter 'p_BCDdigits' is a BCD string (subset of HEXSTRING), the result is BITSTRING[1]. The function of the o_OeBit is as the follows: 1. It returns '1'B, if the length of the 'p_BCDdigits' is odd. 2. It returns '0'B, if the length of the 'p_BCDdigits' is even. EXAMPLE 1: o_OeBit('12583') = '1'B. EXAMPLE 2: o_OeBit('87259957') ='0'B.
o_OtherDigits	**Type of the result:**OCTETSTRING **Parameters:** p_BCDdigits : HEXSTRING The input parameter ` p_BCDdigits ` is a BCD string (subset of HEXSTRING), the result is an even string of BCD digits, with eventually a filler 'F'H used. */ The function of the o_OtherDigits is as the follows: 1. If the number of the 'p_BCDdigits' is odd, the operation removes the most significant digit, and then reverses the order of each pair of digits. 2. If the number of the 'p_BCDdigits' is even, first the operation suffixes the `bcddigits` with 'F'H, then removes the most significant digit, and then reverses the order of each pair of digits. EXAMPLE 1: o_OtherDigi('12345') = '3254', EXAMPLE 2: o_OtherDigi('12345678') ='325476F8'. See o_FirstDigit for the handling of the first digit.
o_SendInSameFrame	**Type of the result:** BOOLEAN **Parameters:** p_NumberMsg : INTEGER **Description** o_SendInSameFrame is called to request SS to send the p_NumberMsg messages in the same frame. Then it returns TRUE.
o_SIB_PER_Encoding	**Type of the result:**BITSTRING **Parameters:** p_SIB : SIB **Description** It returns the unaligned PER encoding (BIT STRING) of the input system information block p_SIB (without "Encoder added (1-7) bits padding"). The bits corresponding to the encoding of the CHOICE of the SIB type shall be removed. Example: for the following SIBType1 value: SysInfoType1 ::= { cn-CommonGSM-MAP-NAS-SysInfo '32F4100001'H, cn-DomainSysInfoList { { cn-DomainIdentity ps-domain, cn-Type gsm-MAP : '0000'H, cn-DRX-CycleLengthCoeff 7}, {cn-DomainIdentity cs-domain, cn-Type gsm-MAP : '0001'H, cn-DRX-CycleLengthCoeff 7}}, ue-ConnTimersAndConstants { t-304 ms100, n-304 7, t-308 ms40, t-309 8, t-313 15, n-313 s200, t-314 s20,

TSO Name	Description
	t-315 s1800, n-315 s1000}, ue-IdleTimersAndConstants { t-300 ms400, n-300 7, t-312 10, n-312 s200}, nonCriticalExtensions { } } The operation returns BITSTRING: "1000011001011110100000100000000000000000001011000100000000000000000100 0010000000000000001010000110011000001111100000111001111111111111111110 0101111010011"
o_SIB_Segmentation	**Type of the result:** SegmentsOfSysInfoBlock **Parameters:** p_SIBBitString : BITSTRING **Description** The function of the o_SIB_Segmentation is as following: 1. If the p_SIBBitString is less than or equal to 226 bits, the bit string is fit into a complete segment. If the segment is less than 226 bits but more than 214 bits,the segment shall be padded to 226 bits long with padding bits set to '0'B. 2. If the input operand p_SIBBitString is longer than 226 bits it is segmented from left to right into segments, each segment except the last one is 222 bits. The last segment may be 222 bits or shorter. If the length of last segment is greater than 214 bits pad it to 222 bits with padding bits set to '0'B. 3. The number of segments is assigned to segCount field of the result. 4. The first segment is assigned to seg1 field of the result, the second segment is assigned to the seg2 field of the result, the third segment is assigned to the seg3 field of the result, and so on till the last segment.
o_CheckPDUsAcknowledged	**Type of the result:** BOOLEAN **Parameters:** p_NackList: NackList Contains a list of integers (possibly empty), each of which corresponds to a PDU SN. Negative acknowledgement is expected for each of these PDUs. p_FSN: INTEGER Contains an integer representing the first SN expected to be acknowledged. p_LSN: INTEGER Contains an integer representing the last SN expected to be acknowledged. p_SUFI_List: SuperFields This parameter contains the received SUFI list to be checked. **Description:** This TSO is used to check that the given SUFI list contains any combination of SUFIs that fulfils the following requirements: 1. Negatively acknowledges all PDUs whose sequence numbers are in p_NackList. Note that the list may be empty. 2. Positively acknowledges all other PDUs with sequence numbers greater thatn or equal to p_FSN, and less than or equal to p_LSN. **Output:** This TSO returns a BOOLEAN value of TRUE if the SUFI list meets all of the requirements based on the given parameters. Otherwise the TSO returns FALSE.

8.7.2 Specific test suite operation definitions for Multi RAT Handover testing

Table 80: TSO definitions for Multi RAT handover

TSO Name	Description
o_GetEstCauRandomRef	**Type of the result:** B_8 **Parameters:** p_msg : CHANNELREQUEST **Description** Returns the Eight bits of the EstCauRandomRef of the PDU CHANNELREQUEST
o_PagingGroupCalculate	**Type of the result:** INTEGER **Parameters:** p_IMSI : HEXSTRING p_CCCH_Conf : B_3 p_N : INTEGER **Description** Calculate the PAGING_GROUP (0 .. N?1) = ((IMSI mod 1000) mod (BS_CC_CHANS x N)) mod N where : N = number of paging blocks "available" on one CCCH = (number of paging blocks "available" in a 51-multiframe on one CCCH) x BS_PA_MFRMS. IMSI = International Mobile Subscriber Identity, as defined in 3GPP TS 23.003 [6]. mod = Modulo. div = Integer division.
o_SecondDigit	**Type of the result:** B4 **Parameters:** p_digits : HEXSTRING **Description** The input parameter bcddigits shall be a BCD string (subset of HEXSTRING) except the third digit can take value 'F'H, the resut is a BITSTRING[4] of a binary representation of one digit in the input string. The function of the o_SecondDigit is to return the second digit of the input parameter p_digits. EXAMPLE 1: o_G_FirstDigit('123') = '0010'B. EXAMPLE 2: o_G_FirstDigit('01F') = '0001'B.
o_ThirdDigit	**Type of the result:** B4 **Parameters:** p_digits : HEXSTRING **Description** The input parameter bcddigits shall be a BCD string (subset of HEXSTRING) except the third digit can take value 'F'H, the resut is a BITSTRING[4] of a binary representation of one digit in the input string. The function of the o_ThirdDigit is to return the third digit of the input parameter p_digits. EXAMPLE 1: o_G_FirstDigit('123') = '0011'B. EXAMPLE 2: o_G_FirstDigit('01F') = '1111'B.
o_TTCN_HO_CommandToBitstring	**Type of the result:** BITSTRING **Parameters:** p_PDU : PDU Description The function of the o_TTCN_HOCommandToBitstring is as the follows: - It returns the bitstring representation of the input HANDOVERCOMMAND p_PDU.

8.7.3 Specific test suite operation for RLC

Table 81: TSO definitions for RLC SUFI handling

TSO Name	Description
o_SUFI_Handler	**Type of the result:** ResAndSUFIs **Parameters:** p_SUFI_Params: SUFI_Params p_SUFI_String: HEXSTRING **Conditions**: Inputs: p_SUFI_Params: the list of checking criteria to be applied by the TSO p_SUFI_String: the HEXSTRING received containing the SUFIs Outputs: the BOOLEAN result of the TSO: TRUE if all checking and the filling of the SuperFields structure were successful; FALSE otherwise; in this case the TSO shall produce sufficient output to allow problem analysis

Table 82: ResAndSUFIs type and Processing of the SUFI parameters input to the TSO

Parameter	Type	Setting	Meaning	Comment
Lower Bound (**LB**) Upper Bound (**UB**)	BITSTRING [12]	OMIT	Do not use !	
		AnyOrOmit	Do not use !	
		Any	Do not use !	
		Value	Use !	
NackList Element i (**Nacki)**	BITSTRING [12]	OMIT	Do not use !	
		AnyOrOmit	Do not use !	
		Any	Do not use !	
		Value	Use !	Check negative ack
Window Size SUFI presence (**WSN_ presence)**	BOOLEAN	OMIT	Use !	Check absence
		AnyOrOmit	Do not use !	
		Any	Use !	Check presence
		Value	Use !	Check presence
MRW SUFI presence (**MRW_ presence)**	BOOLEAN	OMIT	Use !	Check absence
		AnyOrOmit	Do not use !	
		Any	Use !	Check presence
		Value	Use !	Check presence

8.7.3.1 Pseudocode in a C like notation

The pseudocode defined below can be written in a more compact fashion. The code herafter is to allow easy identification of the TSO's tasks. All situations leading to a FALSE result must produce a log. This is not shown in the code hereafter. Possible wrap arounds are not shown in this clause. These have to be accounted for at the appropriate places.

```
/* INITIALIZATION */
Initialize_ResAndSUFIs();                       /* RESULT := TRUE, all SUFI fields are AnyOrOmit */

/* EXTRACTION OF SUFIs AND TRANSFER INTO THE TTCN SUFI STRUCUTRE */
i = 0;
if (p_SUFI_String == NULL)
{
RESULT := FALSE;                                /* No SUFIs -> Result is FALSE */
RETURN;
}
SUFI := Extract_SUFI(i);                        /* Let n SUFI be numbered from 0 to n-1 */
while (SUFI != NULL)                            /* TRUE when there is a SUFI */
{
    Set_SUFI_ListRec(SUFI);                     /* Put the SUFI at the correct place in the
resulting */
/* SUFI structure; overwrite if the SUFI type has */
```

```
/* already been extracted */
    i++;
    SUFI := Extract_SUFI(i);                    /* Get next SUFI */
}


/* CHECK MUTUAL EXCLUSIVENESS OF ACK AND NO_MORE */
/* to be checked if needed */
if Exists_SUFI (ACK) AND Exists_SUFI (NO_MORE)
    RESULT := FALSE;                            /* Exists_SUFI (SUFI_type) is TRUE when the */
/* specified type has been extracted */


/* CHECK ONE OF SUFIs ACK OR NO_MORE IS THE LAST SUFI */
/* check that only one of the SUFIs ACK or NO_MORE has been received and is the last SUFI */
/* FOR ALL SUFI TYPES: IF EXISTING, PERFORM CONSISTENCY CHECK */
if Exists_SUFI (ACK) AND NOT CheckConsistency (ACK)
RESULT := FALSE;                                /* ACK SUFI inconsistent -> Result is FALSE */
.....
if Exists_SUFI (WINDOW) AND NOT CheckConsistency (WINDOW)
RESULT := FALSE;                                /* WINDOW SUFI inconsistent -> Result is FALSE */


/* TAKE THE INDIVIDUAL CHECKING PARAMETERS & PERFORM THE EXPECTED CHECKING */
/* PART 1: EXISTENCE CHECKS */
if (WSN_presence) AND NOT Exists_SUFI(WINDOW)
RESULT := FALSE;                                /* WINDOW not ex. but should -> Result is FALSE */
if (MRW_presence) AND NOT Exists_SUFI(MRW)
RESULT := FALSE;                                /* MRW not ex. but should -> Result is FALSE */


/* PART 2: RANGE AND NACK CHECKS OF SUFI CONTENTS*/
/* ACK: LB <= LSN received <= UB */
if NOT (LB <= Extract_SUFI_Value(ACK) -1 AND Extract_SUFI_Value(ACK) -1 <= UB)
RESULT := FALSE;                                /* ACK value not in the expected range */
                                                /* LB: first SN acceptable as LSN received */
                                                /* UB: last SN acceptable as LSN received */
                                                /* LSN received acks SNs upto LSN received –1 */


/* Bitmap */
/* for all SNs between between LB and UB */
{
if (ExtractBitmap(FSN extracted, LENGTH extracted, Bitmap extracted, SN) == 1) AND (SN in NackList)
RESULT := FALSE;                                /* if the bit in the Bitmap is not 0 */
if (ExtractBitmap(FSN extracted, LENGTH extracted, Bitmap extracted, SN) == 0) AND (SN NOT in
NackList)
RESULT := FALSE;                                /* if the bit in the Bitmap is not 0 */
}


/* LIST */
/* The (SNi,Li) pairs identify AMD PDUs which have not been correctly received. */
/* Therefore the (SNi,Li) pairs have to be consistent with the NackList. */

/* RLIST */

/* The CWs represent the distance between the previous indicated erroneous AMD PDU */
/* up to and including the next erroneous AMD PDU, starting from the FSN contained in the RLIST
SUFI. */
/* Therefore the FSN and the Codewords have to be consistent with the NackList. */
/* Error burst indicator has to be treated as a separate case. May not have to be implemented
currently. */
/* MRW */
/* LENGTH = 0 */
/* 1 SN_MRWi is present and the RLC SDU to be discarded extends above the configured transmission
window in the sender */
/* LENGTH = 1 ... 15 */
/* 1 ...15 SN_MRWi */
/* a) MRW configured ➔ an SN_MRWi indicates the end of each discarded RLC SDU */
/* n SN_MRWs ➔ n RLC SDUs discarded */
/* b) MRW not configured ➔ an SN_MRWi indicates end of last RLC SDU to be discarded */
/* in the receiver */

/* To be implemented as far as required by the RLC ATS */
/* MRW ACK */
/* The SN_ACK must be consistent with the information sent in a previous MRW SUFI upon which the */
/* MRW_ACK represents the answer. */
```

```
/* NO MORE */
/* no checking required */
/* SUBFUNCTIONS USED*/
Check_Consistency (SUFI_type)                    /* returns TRUE when the type fulfills the */

/* requirements of the spec. TS 25.322*/
Exists_SUFI (SUFI_type)                          /* returns TRUE when the specified */

/* type has been extracted, therefore exists*/
ExtractBitmap(FSN extracted, LENGTH extracted, Bitmap extracted, Criterion)
                                        /* Extract the value in the Bitmap at position Criterion */
                                        /* Calculation based on information receivd in the */
                                        /* Bitmap SUFI */
Extract_SUFI (Counter)                  /* returns the SUFI extracted at position counter */

/* from the input p_SUFI_String; */
/* n SUFIs from positions 0 to n-1 */
/* returns NULL if there is no further SUFI */
Extract_SUFI_Value (SUFI_type, field_type )     /* extract the value of specific field type */

/* contained in a specific SUFI type */
/* There will be several flavours depending upon the */
/* result (field) type */
Initialize_ResAndSUFIs ()                        /* Initialize RESULT and all SUFI fields */
Set_SUFI_ListRec(SUFI)                           /* set return values RESULT and */
                                                 /* SUFI structure SUFI_ListRec */
```

8.7.4 Specific test suite operation for MAC

Table 83: TSO definitions for RLC SUFI handling

TSO Name	Description
o_SendContinuousData	**Type of the result:** BOOLEAN **Parameters:** p_RAB_Tx_Info : RAB_Tx_Info **Conditions**: Inputs: p_RAB_Tx_Info: test data, number of RBs, and RB info of each RB (RB id, SDU size and number of SDUs to be transmitted in consecutive TTIs **Outputs:** The BOOLEAN result of the TSO: TRUE if system simulator accepts the information sent from TTCN FALSE if system simulator rejects the information sent from TTCN.

Table 84: RAB_Tx_Info type

Structure Type Definition			
Type Name: RAB_Tx_Info			
Encoding Variation:			
Comments: To provide the information to SS to send data in every TTI on each RAB. Number of RBs depends on specific requirement. SS shall take care about all kind of discard info in all RLC modes and final aim is DL TFCs under test shall be selected in downlink for each TTI.			
Element name	**Type Definition**	**Field Encoding**	**Comments**
test data	BITSTRING		The raw test data buffer
no_of_rbs	INTEGER		No of Radio Bearers
rb_tx_info1	RB_Tx_Info		Info about RB id, SDU size and number of SDUs
rb_tx_info2	RB_Tx_Info		Info about RB id, SDU size and number of SDUs
rb_tx_info3	RB_Tx_Info		Info about RB id, SDU size and number of SDUs
rb_tx_info4	RB_Tx_Info		Info about RB id, SDU size and number of SDUs
rb_tx_info5	RB_Tx_Info		Info about RB id, SDU size and number of SDUs
rb_tx_info6	RB_Tx_Info		Info about RB id, SDU size and number of SDUs

Table 85: RB_Tx_Info type

Structure Type Definition			
Type Name: RB_Tx_Info			
Encoding Variation:			
Comments:			
Element name	Type Definition	Field Encoding	Comments
rb_id	INTEGER		
sdu_size	INTEGER		
no_of_sdus	INTEGER		

8.8 AT commands

Table 68 shows a list of AT commands. By using these commands the ATSs communicate with the SS for an automatic execution. The column 'ATS' indicates in which ATS the command is used.

Table 86: AT commands used in 3GPP ATSs

Command	Reference	ATS
+CGACT	3GPP TS 27.007 [23]	NAS
+CGATT	3GPP TS 27.007 [23]	NAS
+CGCMOD	3GPP TS 27.007 [23]	NAS
+CGDCONT	3GPP TS 27.007 [23]	NAS
+CGDSCONT	3GPP TS 27.007 [23]	NAS
+CGEQREQ	3GPP TS 27.007 [23]	NAS
+CGEREQMIN	3GPP TS 27.007 [23]	NAS
+CLCC	3GPP TS 27.007 [23]	NAS
+VTS	3GPP TS 27.007 [23]	NAS
H	3GPP TS 27.007 [23]	NAS
+CBST	3GPP TS 27.007 [23]	RRC, NAS, SMS
+CMOD	3GPP TS 27.007 [23]	RRC, NAS, SMS
A	3GPP TS 27.007 [23]	RRC, NAS, SMS
D	3GPP TS 27.007 [23]	RRC, NAS, SMS
+CGMD	3GPP TS 27.005 [22]	SMS
+CGMF	3GPP TS 27.005 [22]	SMS
+CGMR	3GPP TS 27.005 [22]	SMS
+CMGW	3GPP TS 27.005 [22]	SMS
+CMSS	3GPP TS 27.005 [22]	SMS
+CNMI	3GPP TS 27.005 [22]	SMS
+CPMS	3GPP TS 27.005 [22]	SMS
+CSCA	3GPP TS 27.005 [22]	SMS
+CSCS	3GPP TS 27.005 [22]	SMS
+CSMP	3GPP TS 27.005 [22]	SMS
+CSMS	3GPP TS 27.005 [22]	SMS

8.9 Bit padding

Three different kinds of bit padding at the RRC layer are defined in 3GPP TS 25.331 [21].

If a bit string is defined in ASN.1 and is an output from a (PER) encoder, it may need the segmentation and padding. One example is that each SIB message is PER-encoded and becomes a (PER) bit-string. A long bit-string is segmented in fixed length, for example with 222 bits. The (1 ... 7) padding bits shall be added at the last segment if it's lengh is between 215 - 211.

No bit padding shall be generated by the PER encoder. Contrary to ITU-T Recommendation X.691 [28], the unaligned PER encoder shall not generate any padding bit to achieve octet alignment at the end of a PER bit string.

RRC padding. The RRC padding bits shall be generated after PER encoder. If the PER bit strings are exchanged via AM or UM SAP, the (1 ... 7) padding bits shall be added to ensure the octed alignment. If the PER bit strings are exchanged via TR SAP, before the exchanges, RRC shall select the smallest transport format that fits the RRC PDU and shall add the lowest number of padding bits required to fit the size specified for the selected transport format. The RRC padding bits shall be taken into account at the calculation of the integrity checksum.

8.9.1 The requirements for implementation

The different kinds of bit padding occur at the different places in the testing architecture. Care must be taken, in order to ensure the correct implementation.

The bit padding for the embedded bit string in ASN.1shall be resolved in TTCN. It is under the responsibility of the TTCN writer. Several TSO defined can resolve the necessary bit padding in the downlink direction.

The unaligned PER encoder used for TTCN shall not implement the octet alignment at the end of a PER bit string in the downlink direction.

The RRC padding should be implemented at the SS in the downlink direction both for AM/UM and TR modes according to 3GPP TS 25.331 [21], clause 12.1.3.

The SS PER decoder compliant with R99 has no need to distinguish the extension and padding parts in the UL direction, and shall match and accept RRC PDUs with any bit string in the extension and padding parts. The remaining part of the received bit string shall be discarded regardless of the RLC mode.

8.10 Test PDP contexts

The following table defines test PDP contexts used in the generic procedures for the PS establishment and other SM tests. The test PDP context1 is the default Test PDP context used in the test cases where no particular Test PDP contexts are specified and UE is in DCH state. The test PDP context2 is the default Test PDP context used in the test cases where no particular Test PDP contexts are specified and UE is in FACH state.

Table 87: Test PDP contexts

	PDP Context1	PDP Context2	PDP Context3
NSAPI	Selected by UE in Activate PDP Context Request	Selected by UE in Activate PDP Context Request	Selected by UE in Activate PDP Context Request
LLC SAPI	0	0	0
QoS	QoS-UL64kAM-DL64kAM	QoS- UL32kAM-DL32kAM	QoS- UL8kAM-DL8kAM
PDP address	PIXIT	PIXIT	PIXIT
Radio Priority	1	1	1
Access Point Name	PIXIT	PIXIT	PIXIT
Protocol configuration options	TBD	TBD	TBD
Packet Flow Identifier	Best Effort	Best Effort	Best Effort

Table 88: Test QoS

	QoS-UL64kAM-DL64kAM	QoS- UL32kAM-DL32kAM	QoS- UL8kAM-DL8kAM
Reliability class	'001' Acknowledged GTP, LLC, and RLC; Protected data	'001' Acknowledged GTP, LLC, and RLC; Protected data	'001' Acknowledged GTP, LLC, and RLC; Protected data
Delay class	'100' Best effort	'100' Best effort	'100' Best effort
Precedence class	'100' Normal Class	'100' Normal Class	'100' Normal Class
Peak throughput	'0111' 64 kbps	'0110' Up to 32 000 octet/s	'0110' Up to 32 000 octet/s
Mean throughput	'11111'B Best Effort	'11111'B Best Effort	'11111'B Best Effort
Delivery of erroneous SDU	'010' B Erroneous SDUs are delivered ('yes')	'010' B Erroneous SDUs are delivered ('yes')	'010' B Erroneous SDUs are delivered ('yes')
Delivery order	'01'B With delivery order ('yes')	'01'B With delivery order ('yes')	'01'B With delivery order ('yes')
Traffic class	'011' B Interactive class	'011' B Interactive class	'011' B Interactive class
Maximum SDU size	'20' O 320 bits]	'20'O 320 bits	'20'O 320 bits
Maximum bit rate for uplink	'40' O	'20'O 32 kbps	'08'O 32 kbps
Maximum bit rate for downlink	'40' O	'20'O 32 kbps	'08'O 32 kbps
Residual BER	'1001' 6X10E-3	'1001' 6X10E-3	'1001' 6X10E-3
SDU error ratio	'0011' 1X10E-3	'0011' 1X10E-3	'0011' 1X10E-3
Traffic Handling priority	'11' B Needs to be neglected by UE	'11' B Needs to be neglected by UE	'11' B Needs to be neglected by UE
Transfer delay	'111111' B spare (not applicable for Interactive / Background)	'111111' B spare (not applicable for Interactive / Background)	'111111' B spare (not applicable for Interactive / Background)
Guaranteed bit rate for uplink	'40' O 64 kbps	'20'O 32 kbps	'08'O 32 kbps
Guaranteed bit rate for downlink	'40' O 64 kbps	'20'O 32 kbps	'08'O 8 kbps

Annex A (normative): Abstract Test Suites (ATS)

This annex contains the approved ATSs.

The ATSs have been produced using the Tree and Tabular Combined Notation (TTCN) according to TR 101 666 [27].

The ATSs were developed on a separate TTCN software tool and therefore the TTCN tables are not completely referenced in the table of contents. Each ATS contains a test suite overview part which provides additional information and references.

A.1 Version of specifications

Table A.1 shows the version of the test specifications which the delivered ATSs are referred to.

Table A.1: Versions of the test and Core specifications

Test specifications	3GPP TS 34.123-1 [1] (V5.0.1)
	3GPP TS 34.123-2 [2] (V5.0.0)
	3GPP TS 34.108 [3] (V3.8.0)
	3GPP TS 34.109 [4] (V3.6.0)

A.2 NAS ATS

A.2.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.2.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.3 SMS ATS

A.3.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.3.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.4 RRC ATS

The approved RRC test cases are listed.

Table A.4: RRC TTCN test cases

Test case	Description
Singlecell	
8.1.1.1	RRC / Paging for Connection in idle mode
8.1.2.1	RRC / RRC Connection Establishment in CELL_DCH state: Success
8.1.3.1	RRC / RRC Connection Release in CELL_DCH state: Successful

A.4.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (RRCv300.PDF contained in archive 34123c300ATS.ZIP) which accompanies the present document.

A.4.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (RRCv300.MP contained in archive 34123c300ATS.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.5 RLC ATS

A.5.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.5.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.6 MAC ATS

A.6.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.6.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.7 BMC ATS

A.7.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.7.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.8 PDCP ATS

A.8.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.8.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

A.9 RAB ATS

A.9.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in an Adobe Portable Document Format™ file (<any_name>.PDF contained in archive <Shortfilename>.ZIP) which accompanies the present document.

A.9.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in an ASCII file (<any_name>.MP contained in archive <Shortfilename>.ZIP) which accompanies the present document.

NOTE: Where an Abstract Test Suite (in TTCN) is published in both .GR and .MP format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

Annex B (normative): Partial IXIT proforma

B.0 Introduction

This partial IXIT proforma contained in the present document is provided for completion, when the related Abstract Test Suite is to be used against the Implementation Under Test (IUT).

Text in *italics* is comments for guidance for the production of a IXIT, and is not to be included in the actual IXIT.

The completed partial IXIT will normally be used in conjunction with the completed ICS, as it adds precision to the information provided by the ICS.

B.1 Parameter values

B.1.1 BasicM Test Suite Parameter Declarations

The following parameters are common to all ATSs.

Table B.1: BasicM PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_AccessPtNameDCH	The logical name for the GGSN or the external packet world for the DCH PDP context	IA5String	"ABCDEF"	
px_AccessPtNameFACH	The logical name for the GGSN or the external packet world for the FACH PDP context	IA5String	"GHIJK"	
px_PDP_IP_AddrInfoDCH	A string parameter that identifies the MT in the address space applicable to the PDP for DCH.	IA5String	"200.1.1.80"	
px_PDP_IP_AddrInfoFACH	A string parameter that identifies the MT in the address space applicable to the PDP for FACH.	IA5String	"200.1.1.90"	
px_AuthAMF	Authentication Management Field (16 bits). The value shall be different from '1111 1111 1111 1111'B (AMFresynch).	BITSTRING	See note 2	
px_AuthK	Authentication Key (128 bits)	BITSTRING	'0101111001001 0101011001101 0110001001000 1001101110101 1101001010101 1101110100000 0100101110011 0011111000011 0000100110100 11000101001'B	
px_AuthN	Value of n to initialize tcv_Auth_n (length of extended response) min 31, max 127 (3GPP TS 34.108 [3] clause 8.1.2)	INTEGER	127	
px_AuthRAND	Random Challenge (128 bits)	BITSTRING	'01010101…01'B	
px_CC_CallDiallingDigits	Dialling digits used to initiate a CC MO call (used with the AT dial D command).	IA5String	"0123456902"	
px_Cg01	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	BITSTRING[4]	"Test_cg1"	
px_Cg02	Data to be sent in TC 7.4.2.1	BITSTRING[4]	"Test_cg2"	
px_CipheringOnOff	Security mode - TRUE if ciphering is applicable	BOOLEAN	TRUE	
px_CN_DomainTested	CN domain to be tested. This parameter is used in test cases that handle both PS and CS domains.	CN_DomainIdentity	cs_domain	
px_Code01	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	BITSTRING[4]	"Test_ code01"	
px_Code02	Data to be sent in TC 7.4.2.1	BITSTRING[4]	"Test_ code02"	
px_CRNTI	C RNTI	C_RNTI	'0000000000000001'B	
px_DL_TxPower_DPCH	Down link transmit power level of DPCH	DL_TxPower	-5	
px_FRESH	Value for FRESH	Fresh	See note 1	
px_IMEI_Def	Default IMEI value	HEXSTRING	See note 1	
px_IMEISV_Def	Default IMEISV value	HEXSTRING	See note 1	

Parameter Name	Description	Type	Default Value	Supported Value
px_IMSI_Def	Default IMSI value	HEXSTRING	'001010123456063'H	
px_IMSI_Diff	Different IMSI from the IMSI stored in the USIM	HEXSTRING	'001010654321063'H	
px_IntegrityOnOff	Integrity mode – Shall be set to TRUE, it is possible to set to FALSE in order to test several protoypes of UE which have not yet implemented the integrity function. Default value: TRUE	BOOLEAN	TRUE	
px_KeySeqDef	Default Key Sequence	Keyseq	'101'B	
px_MSClsmkA5_1	Default Algorithm A5/1 supported	B1	'0'B	
px_MSClsmkESIND	Default Early Sending Indication	B1	'0'B	
px_MSClsmkRevLvl	Default Revision Level	B2	'10'B	
px_MSClsmkRF_PwrCap	Default RF Power Capability	B3	'000'B	
px_NMO	This parameter is used to specify network operation mode. Valid values: '00'O and '01'O	OCTETSTRING	'00'O	
px_OperationBandSupp	Operating Band supported (1, 2 or 3).	INTEGER	1	
px_PowerAICH	Transmission power level of AICH	DL_TxPower	-65	
px_PowerpCCPCH	Transmission power level of primary CCPCH	DL_TxPower	-2	
px_PowerpCPICH	Transmission power level of primary CPICH	DL_TxPower_PCPICH	-60	
px_PowerPICH	Transmission power level of PICH	DL_TxPower	-65	
px_PowerpSCH	Transmission power level of primary SCH	DL_TxPower	-5	
px_PowersCCPCH1	Transmission power level of secondary CCPCH1	DL_TxPower	-2	
px_PowersSCH	Transmission power level of secondary SCH	DL_TxPower	-5	
px_PriScrmCode	Primary scrambling code	PrimaryScramblingCode	100	
px_PTMSI_Def	default PTMSI	OCTETSTRING	'12345678'O	
px_PTMSI_SigDef	default PTMSI signature (3 octets, 3GPP 24.008 [9], clause 10.5.5.8).	OCTETSTRING	'AB123466'O	
px_PuncLimit	Puncturing limit for PRACH	PuncturingLimit	pl1	
px_RAT	This parameter is used to specify which radio access technology is being used for the current test execution. Valid values: fdd and tdd	RatType	fdd	
px_RB_Background_64	Data to be sent for RB test TC_14_2_26.	BITSTRING	INT_TO_BIT (17378987476987465213313265 0, 1344)	
px_RB_DataConversational_64	Data to be sent for RB test TC_14_2_13.	BITSTRING	INT_TO_BIT (8941203214580965478932211684654654, 2560)	
px_RB_DataSpeech_12_2	Data to be sent for RB test TC_14_2_4.	BITSTRING	INT_TO_BIT (15896423213132132, 103)	
px_RB_DataStreaming_57_6	Data to be sent for RB test TC_14_2_17.	BITSTRING	INT_TO_BIT (12358987456987465213213265 0, 2304)	
px_RB_Interactive_64	Data to be sent for RB test TC_14_2_26.	BITSTRING	INT_TO_BIT (15358987456987465213313265 0, 1344)	
px_RRC_CS_ServTested	CS service to be tested for RRC test cases.	RRC_ServTested	Speech	

Parameter Name	Description	Type	Default Value	Supported Value
px_RRC_PS_ServTested	PS service to be tested for RRC test cases.	RRC_ServTested	Speech	
px_SFN_OffsetA	SFN offset values for cell A	INTEGER	0	
px_SFN_OffsetB	SFN offset values for cell B	INTEGER	0	
px_SFN_OffsetC	SFN offset values for cell C	INTEGER	0	
px_SFN_OffsetD	SFN offset values for cell D	INTEGER	15624	
px_SFN_OffsetE	SFN offset values for cell E	INTEGER	15624	
px_SFN_OffsetF	SFN offset values for cell F	INTEGER	678	
px_SFN_OffsetG	SFN offset values for cell G	INTEGER	1356	
px_SFN_OffsetH	SFN offset values for cell H	INTEGER	2034	
px_SlotFormatsCCPCH1	Channelization code for secondary CCPCH1 when spreading factor = 64	SCCPCHSlotFormat	4	
px_SRNC_Id	SRNC Id	SRNC_Identity	'0000 0000 0001'B	
px_SRNC_IdDiff	Different value for SRNC Id than in px_SRNCId	SRNC_Identity	'0000 0000 0010'B	
px_SRNTI	S RNTI	S_RNTI	'0000 0000 0000 0000 0001'B	
px_SRNTI_Diff	Different value for S RNTI than in px_SRNTI	S_RNTI	'0000 0000 0000 0000 0010'B	
px_TCellA	TCell value for cell A	Tcell	0	
px_TCellB	TCell value for cell B	Tcell	512	
px_TCellC	TCell value for cell C	Tcell	1536	
px_TCellD	TCell value for cell D	Tcell	321	
px_TCellE	TCell value for cell E	Tcell	833	
px_TCellF	TCell value for cell F	Tcell	6577	
px_TCellG	TCell value for cell G	Tcell	7253	
px_TCellH	TCell value for cell H	Tcell	4351	
px_TimingsCCPCH1	Timing offset for secondary CCPCH1	INTEGER	0	
px_TMSI_Def	Default TMSI	OCTETSTRING	'12345678'O	
px_UARFCN_D_Mid	Downlink UARFCN number	INTEGER	10700	
px_UARFCN_D_Low	Another value for downlink UARFCN number	INTEGER	10563	
px_UARFCN_D_High	downlink UARFCN for Ch2	INTEGER	10837	
px_UE_OpModeDef	Default UE operation mode (either opModeA or opModeC). (For most UEs this corresponds class-A or class-C, and can not be changed by the user)	UE_OperationMode	opModeA	
px_UL_ScramblingCode	UL scrambling code value to be used by UE.	UL_ScramblingCode	0	
px_UTRAN_GERAN	This parameter is used to specify for which environment region the system information blocks are broadcast in the test execution. Valid values: "UTRAN only" and "UTRAN and GERAN".	Region	"UTRAN and GERAN"	
NOTE 1: No default value can be proposed (Manufacturer defined value). NOTE 2: No default value can be proposed, because not enough information is available in 3GPP TS 34.109 [4] clause 8.1.2.				

B.1.2 L3M Test Suite Parameters Declarations

The following parameters are commonly used in the RRC and NAS ATSs.

Table B.2: L3M PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_BcapDataCompression	Data compression supported (used in the Bearer Capability)	B1	'0'B	
px_BcapFNUR	Fixed Network User rate supported: '00001'B: FNUR 9.6 kbit/s '00010'B: FNUR 14.4 kbit/s '00011'B: FNUR 19.2 kbit/s '00100'B: FNUR 28.8 kbit/s '00101'B: FNUR 38.4 kbit/s '00110'B: FNUR 48.0 kbit/s '00111'B: FNUR 56.0 kbit/s '01000'B: FNUR 64.0 kbit/s '01001'B: FNUR 33.6 kbit/s '01010'B: FNUR 32.0 kbit/s	B5	'00001'B	
px_BcapITC	Information transfer capability supported (used for the generation of the Bearer Capability) 0 - UDI 1 - RDI 2 - 31 kHz Audio 3 - Other	ItcInt	2	
px_BcapModemType	Modem type supported (used in the Bearer Capability)	B5	'00110'B	
px_BcapNumberDataBits	Number of data bits supported (used in the Bearer Capability)	B1	'1'B	
px_BcapNumberStopBits	Number of Stops bits supported (used in the Bearer Capability)	B1	'1'B	
px_BcapOtherModemType	Other modem type supported (used in the Bearer Capability)	B2	'10'B	
px_BcapParity	Parity supported (used in the Bearer Capability)	B3	'011'B	
px_BcapSACP	Signalling access protocol supported (used in the Bearer Capability)	B3	'001'B	
px_BcapSyncAsync	Synchronous '0'B or Asynchronous '1'B mode supported by IUT	B1	'1'B	
px_BcapUeFlowControl	UE flow control. 0-outband, 1-inband, 2-no flow control. 3- X.25 4- X.75 Default: 0, outband flow control	FlowControl	0	
px_CC_Serv	Service selected for Mobile Originated calls and Mobile Terminated calls. The possible values are ("Telephony", "EmergencyCall", "31kHz", "V110", "V120", "PIAFS", "FTM", "X31", "BTM", "MmediaCall")	Services	"31kHz"	
px_MS_ClsmkA5_2	Default Algorithm A5/2 supported	B1	'0'B	
px_MS_ClsmkA5_3	Default Algorithm A5/3 supported	B1	'0'B	
px_MS_ClsmkCM3	Default Classmark 3 Indicator	B1	'0'B	
px_MS_ClsmkCMSP	Default CM Service Prompt Support	B1	'0'B	
px_MS_ClsmkFreqCap	Default Frequency Capability	B1	'0'B	
px_MS_ClsmkLCSVA_Cap	Default LCSVA Capabilities Support	B1	'0'B	
px_MS_ClsmkPS_Cap	Default Pseudo Synchronisation Capability	B1	'0'B	
px_MS_ClsmkSM_Cap	Default Short Message Capability	B1	'1'B	
px_MS_ClsmkSoLSA	Default SoLSA supported	B1	'0'B	
px_MS_ClsmkSSSI	Default SS Screen Indicator	B2	'01'B	
px_MS_ClsmkUCS2	Default UCS2 encoding supported	B1	'0'B	
px_MS_ClsmkVBS	Default VBS Capability	B1	'0'B	
px_MS_ClsmkVGCS	Default VGCS Capability	B1	'0'B	

Parameter Name	Description	Type	Default Value	Supported Value
px_NwOrgPDP_Support	This indicates if the UE implementation supports network originated PDP Context. TRUE indicates, supported FALSE indicate, not supported	BOOLEAN	FALSE	
px_PDP_TypeNo	Indicates IP v4 or IP v6	PDP_TypeNo	'00100001'O	
px_PDP_TypeOrg	A string parameter which specifies the type of packet data protocol	B4	'0000'B	
px_UARFCN_D_B	RF frequency number for downlink Cell B	INTEGER	10650	
px_UARFCN_U_B	RF frequency number for uplink Cell B	INTEGER	9700	

B.1.3 NAS Test Suite Parameters Declarations

The following parameters are commonly used in the NAS ATS.

Table B.3: NAS PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_AuthRAND_2	A second Random Challenge (128 bits)	BITSTRING	'1010101…10'B	
px_AutocallingBlacklistNumber	Number of B-party numbers that can be stored in the list of blacklisted numbers	INTEGER	20	
px_AutocallingCause1or2	Cause value of category 1 or 2 to be used in TC_17_1_3	INTEGER	18	
px_AutocallingNumber	Called number to be used for auto calling	IA5String	"0613454120"	
px_AutocallingRepeatCat1or2	Number of repeat attempt done for the category 1 or 2 to be used in TC_17_1_3	INTEGER	10	
px_CC_ServNotSupp	Not supported service selected for Mobile Originated calls and Mobile Terminated calls. The possible values are ("Telephony", "EmergencyCall", "31kHz", "V110", "V120", "PIAFS", "FTM", "X31", "BTM", "MmediaCall")	Services	"BTM"	
px_DTMF_BasicCharSet	TRUE if DMTF Chars 0-9, *, # supported	BOOLEAN	TRUE	
px_DTMF_OtherCharSet	TRUE if DMTF Chars A, B, C, D supported	BOOLEAN	TRUE	
px_DTMF_ToneInd	TRUE if UE support DTMF tone indication	BOOLEAN	TRUE	
px_EmergencyCallNumber	Emergency Number used by UE to initiate an emergency call	EmergencyNumber	"112"	
px_KeySeq2	Second key sequence	KeySeq	'000'B	
px_NoNwOrgPDP_ContextSupp	This indicates the number of network originated PDP context supported by the UE	INTEGER (0..7)	7	
px_SupportOpModeC	Paramter is TRUE if UE supports operation mode C. Operation mode C means UE offers PS services only (see 3GPP 23.060 clause 4.1 and 3GPP 24.008 [9])	BOOLEAN	TRUE	
px_TMSI_2	Second TMSI value	OCTETSTRING	'09876543'O	
px_UARFCN_D_C	RF frequency number for downlink Cell C	INTEGER	10750	
px_UARFCN_U_C	RF frequency number for uplink Cell C	INTEGER	9800	
px_UARFCN_D_D	RF frequency number for downlink Cell D	INTEGER	5000	
px_UARFCN_U_D	RF frequency number for uplink Cell D	INTEGER	5950	

Parameter Name	Description	Type	Default Value	Supported Value
px_UuInfo	User-user information for TC 10_3	OCTETSTRING	'01020304'O	
px_Uupd	User-user protocol discriminator for TC 10_3	B8	'00000100'B	
px_PTMSI_2	Second PTMSI used for testing.	OCTETSTRING	'09876543'O	
px_PTMSI_Sig2	Second PTMSI signature used for testing.	OCTETSTRING	'AB123467'O	
px_VTS_AT_CommandSupp	TRUE if the AT command +VTS is supported	BOOLEAN	TRUE	

B.1.4 SMS Test Suite Parameters Declarations

These parameters are used in the SMS ATS.

Table B.4: SMS PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_BMC_CB_RepPeriod01	CB repetition period for CB message 1	INTEGER	2	
px_BMC_CB_RepPeriod02	CB repetition period for CB message 2	INTEGER	2	
px_BMC_NoOfBC_Req01	No of broadcasts requested for CB message 1	INTEGER	2	
px_BMC_NoOfBC_Req02	No of broadcasts requested for CB message 2	INTEGER	2	
px_MaxCP_DataRetx	max. number of CP data retransmissions for SMS	INTEGER	3	
px_SMS_CB_Data01	Contents of the first Cell Broadcast Message sent will be converted to an OCTETSTRING	IA5String	"First Cell Broadcast Message"	
px_SMS_CB_Data02	Contents of the second Cell Broadcast Message sent will be converted to an OCTETSTRING	IA5String	"Second Cell Broadcast Message"	
px_SMS_CB_MsgId01	Message Id to be used for the first Cell Broadcast Message sent	B16	'0000000000000001'B	
px_SMS_CB_MsgId02	Message Id to be used for the second Cell Broadcast Message sent	B16	'0000000000000010'B	
px_TC1M	Value for timer TC1M, to be declared by the manufacturer	INTEGER	10000	

B.1.5 RRC Test Suite Parameters Declarations

These parameters are used in the RRC and RAB ATS.

Table B.5: RRC and RAB PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_DL_MaxCC_TB_bits	Maximum sum of number of bits of all convolutionally coded transport blocks being received at an arbitrary time instant.	MaxNoBits	b163840	
px_DL_MaxCCTrCH	Maximum number of Simultaneous CCTrCH for downlink	MaxSimultaneousCCTrCH_Count	8	
px_DL_MaxTB_bits	Maximum sum of number of bits of all transport blocks being received at an arbitrary time instant.	MaxNoBits	b163840	

Parameter Name	Description	Type	Default Value	Supported Value
px_DL_MaxTC_TB_bits	Maximum sum of number of bits of all turbo coded transport blocks being received at an arbitrary time instant.	MaxNoBits	b163840	
px_DL_MaxTF	Maximum number of TF for downlink	MaxNumberOfTF	tf1024	
px_DL_MaxTFS	Maximum number of TFC in the TFCS for downlink	MaxNumberOfTFC_DL	tfc1024	
px_DL_MaxTrCHs	Maximum number of simultaneous transport channels for downlink.	MaxSimultaneousTransChsDL	e32	
px_DL_MaxTTI_TB	Maximum total number of transport blocks received within TTIs that end within the same 10 ms interval.	MaxTransportBlocksDL	tb512	
px_DL_TC	Support for turbo decoding for downlink.	BOOLEAN	TRUE	
px_MaxAM_EntityNumberRLC_Cap	Maximum AM Entity Number for RLC.	MaximumAM_EntityNumberRLC_Cap	am30	
px_MaxHcContextSpace	MaxHcContextSpace if RFC 2507 [30] is supported.	MaxHcContextSpace	by512	
px_MaxNoDPCH_PDSCH_Codes	Part of DL_PhysChCapabilityFDD. INTEGER (1..8).	INTEGER	8	
px_MaxNoDPDCH_BitsTransmitted	Part of UL_PhysChCapabilityFDD.	MaxNoDPDCH_BitsTransmitted	b57600	
px_MaxNoPhysChBitsReceived	Part of DL_PhysChCapabilityFDD.	MaxNoPhysChBitsReceived	b76800	
px_MaxNoSCCPCH_RL	Part of SimultaneousSCCPCH_DPCH_Reception.	MaxNoSCCPCH_RL	rl1	
px_MaxRLC_WindowSize	Maximum RLC window size.	MaximumRLC_WindowSize	mws4095	
px_RRC_CS_ServTested	RRC_ServTested	CS service to be tested for RRC test cases	Speech	
px_SupportOfGSM	GSM supported by UE	BOOLEAN	TRUE	
px_SupportOfMulticarrier	Part of MultiRAT_Capability.	BOOLEAN	TRUE	
px_TotalRLC_AM_BufferSize	Total RLC AM buffer size.	TotalRLC_AM_BufferSize	NA	
px_TxRxFrequencySeparation	TxRxFrequencySeparation value.	TxRxFrequencySeparation	mhz190	
px_UE_PowerClass	UE_PowerClass value.	UE_PowerClass	1	
px_UL_MaxCC_TB_bits	Maximum sum of number of bits of all convolutionally coded transport blocks being transmitted at an arbitrary time instant.	MaxNoBits	b163840	
px_UL_MaxTB_bits	Maximum sum of number of bits of all transport blocks being transmitted at an arbitrary time instant.	MaxNoBits	b163840	
px_UL_MaxTC_TB_bits	Maximum sum of number of bits of all turbo coded transport blocks being transmitted at an arbitrary time instant.	MaxNoBits	b163840	
px_UL_MaxTF	Maximum number of TF for uplink.	MaxNumberOfTF	tf1024	
px_UL_MaxTFS	Maximum number of TFC in the TFCS for uplink.	MaxNumberOfTFC_DL	tfc1024	
px_UL_MaxTrCHs	Maximum number of simultaneous transport channels for uplink.	MaxSimultaneousTransChsUL	e32	

Parameter Name	Description	Type	Default Value	Supported Value
px_UL_MaxTTI_TB	Maximum total number of transport blocks transmitted within TTIs that start at the same time.	MaxTransportBlocksUL	tb512	
px_UL_TC	Support for turbo encoding for uplink.	BOOLEAN	TRUE	
px_UE_PositioningNetworkAssistedGPS_Sup	UE positioning capability: supports network assisted by GPS	NetworkAssistedGPS_Supported	networkBased	
px_UE_PositioningIPDL_Sup	UE positioning capability: support for IPDL	BOOLEAN	TRUE	
px_UE_PositioningGPS_TimingOfCellFramesSup	UE positioning capability: the UE supports the GPS timing of cell frames	BOOLEAN	TRUE	
px_UE_PositioningBasedOTDOA_Sup	UE positioning capability: the Based OTDOA is supporting by UE	BOOLEAN	TRUE	
px_UE_PositioningStandaloneLocMethodsSup	UE positioning capability: the standalone location method is supporting by UE	BOOLEAN	TRUE	

B.1.6 PDCP Test Suite Parameters Declarations

These parameters are used in the PDCP ATS.

Table B.6: PDCP PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_PDCP_TCPIP_Packet1	Data to be sent for each PDCP test	OCTETSTRING	"Test_PDCP_TCPIP_Packet1"	
px_PDCP_TCPIP_Packet2	Data to be sent for each PDCP test	OCTETSTRING	"Test_PDCP_TCPIP_Packet2"	
px_PDCP_TcpIpCompressedTcpNonDeltaPacket01	IP header compressed packet type (PID=3) of px_PDCP_TcpIpUncompressedPacket01	IP_Packet	0000 0000 0000 0a00 0000 0050 1000 0026 3400 006a 6e6e 206a 6e6e 206a 6e6e	
px_PDCP_TcpIpCompressedTcpNonDeltaPacket02	IP header compressed packet type (PID=3) of px_PDCP_TcpIpUncompressedPacket02	IP_Packet	"Test_PDCP_TCPIP_Packet2_PID_Type3"	
px_PDCP_TcpIpCompressedTcpPacket01	IP header compressed packet type (PID=2) of px_PDCP_TcpIpUncompressedPacket01	IP_Packet	0028 2634 0a00 0000 6a6e 6e20 6a6e 6e	
px_PDCP_TcpIpCompressedTcpPacket02	IP header compressed packet type (PID=2) of px_PDCP_TcpIpUncompressedPacket02	IP_Packet	"Test_PDCP_TCPIP_Packet2_PID_Type2"	
px_PDCP_TcpIpFullHeaderPacket01	IP header compressed packet type (PID=1) of px_PDCP_TcpIpUncompressedPacket01	IP_Packet	c500 0000 0000 0000 4006 7ac6 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 5010 0000 263e 0000 6a6e 6e20 6a6e 6e	
px_PDCP_TcpIpFullHeaderPacket02	IP header compressed packet type (PID=1) of px_PDCP_TcpIpUncompressedPacket02	IP_Packet	"Test_PDCP_TCPIP_Packet2_PID_Type1"	

Parameter Name	Description	Type	Default Value	Supported Value
px_PDCP_TcpIpUncompressedPacket01	uncompressed TCP/IP Packet01	IP_Packet	4500 0033 0000 0000 4006 7ac6 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 5010 0000 263e 0000 6a6e 6e20 6a6e 6e	
px_PDCP_TcpIpUncompressedPacket02	uncompressed TCP/IP Packet02	IP_Packet	"Test_PDCP_TCPIP_Packet2"	
px_PDCP_UDPIP_Packet1	Data to be sent for each PDCP test, except TC 7.3.3.1 and 7.3.3.2	OCTETSTRING	"Test_PDCP_UDPIP_Packet1"	
px_PDCP_UDPIP_Packet2	Data to be sent for each PDCP test, except TC 7.3.3.1 and 7.3.3.2	OCTETSTRING	"Test_PDCP_UDPIP_Packet2"	
px_PDCP_UdpIpCompressedTcpNonTcpPacket01	IP header compressed packet type (PID=4) of px_PDCP_UdpIpUncompressedPacket01	IP_Packet	0001 0000 763c 6a6e 6e20 6a6e 6e20 6a6e 6e	
px_PDCP_UdpIpCompressedTcpNonTcpPacket02	IP header compressed packet type (PID=4) of px_PDCP_UdpIpUncompressedPacket02	IP_Packet	"Test_PDCP_UDPIP_Packet2_PID_Type4"	
px_PDCP_UdpIpFullHeaderPacket01	IP header compressed packet type (PID=1) of px_PDCP_UdpIpUncompressedPacket01	IP_Packet	8500 0100 0000 0000 4011 7ac7 0000 0000 0000 0000 0000 0000 0013 763c 6a6e 6e20 6a6e 6e20 6a6e 6e	
px_PDCP_UdpIpFullHeaderPacket02	IP header compressed packet type (PID=1) of px_PDCP_UdpIpUncompressedPacket02	IP_Packet	"Test_PDCP_UDPIP_Packet2_PID_Type1"	
px_PDCP_UdpIpUncompressedPacket01	uncompressed UDP/IP Packet01	IP_Packet	4500 0027 0000 0000 4011 7ac7 0000 0000 0000 0000 0000 0000 0013 763c 6a6e 6e20 6a6e 6e20 6a6e 6e	
px_PDCP_UdpIpUncompressedPacket02	uncompressed UDP/IP Packet02	IP_Packet	"Test_PDCP_UDPIP_Packet2"	

B.1.7 BMC Test Suite Parameters Declarations

These parameters are used in the BMC ATS.

Table B.7: BMC PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_CB_Data1	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	IA5String [1..1246]	"CB Data1"	
px_CB_Data2	Data to be sent in TC 7.4.2.1	IA5String [1..1246]	"CB Data2"	
px_SMS_CB_MsgId01	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	HEXSTRING[4]	'0000'H	
px_SMS_CB_MsgId02	Data to be sent in TC 7.4.2.1	HEXSTRING[4]	'0000'H	
px_GS01	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	BITSTRING[2]	"Test_gS1"	
px_GgS02	Data to be sent in TC 7.4.2.1	BITSTRING[2]	"Test_gS2"	
px_MsgCode01	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	BITSTRING[10]	"Test_msgCode 01"	
px_MsgCode02	Data to be sent in TC 7.4.2.1	BITSTRING[10]	"Test_msgCode 02"	
px_UpdateNumber01	Data to be sent for each PDCP test, except TC 7.4.1.4, 7.4.1.5 and 7.4.1.6	BITSTRING[4]	"Test_ updateNumber0 1"	
px_UpdateNumber02	Data to be sent in TC 7.4.2.1	BITSTRING[4]	"Test_ updateNumber0 2"	

B.1.8 RRC Test Suite Parameters Declarations

These parameters are used in the RRC ATS.

Table B.8: RRC PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_CipherAlg	Cipher algorithm.	B_3	'000'B	
px_CipherKey	Cipher Key (64 bits).	B_64	'01011110010010 10101100110101 10001001000100 11011101011101 00101010'B	
px_CRNTI_Diff	different value for C RNTI than in px_CRNTI.	C_RNTI	'0000 0000 0000 0010'B	
px_G_TimeSlot	time slot 3GPP TS 24.008 [9], clause 10.5.2.5, BITSTRING[3] suitable for Single slot operation	BITSTRING [3]	'001'B	
px_MS_TXPWR_MAX_CCH	MS_TXPWR_MAX_CCH.	B_5	'01010'B	
px_RXLEV_ACCESS_MIN	minimum received signal level at MS.	B_6	'000000'B	
px_SplitOnCCCH	Split pg cycle on CCCH supported indication (1 bit)	B_1	'0'B not supported	
px_TSC	Training sequence code for traffic channels.	B_3	'011'B	

B.1.9 RAB Test Suite Parameters Declarations

These parameters are used in the RAB ATS.

Table B.9: RAB PIXIT

Parameter Name	Description	Type	Default Value	Supported Value
px_RB_Background_128	Data to be sent for RB test TC_14_2_28.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 2688)	
px_RB_Background_128_2048	Data to be sent for RB test TC_14_2_36.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 41984)	
px_RB_Background_128_384	Data to be sent for RB test TC_14_2_33.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 8064)	
px_RB_Background_144	Data to be sent for RB test TC_14_2_30.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 3024)	
px_RB_Background_32_64	Data to be sent for RB test TC_14_2_25.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 1344)	
px_RB_Background_32_8	Data to be sent for RB test TC_14_2_23.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 672)	
px_RB_Background_384	Data to be sent for RB test TC_14_2_34.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 8064)	
px_RB_Background_384_2048	Data to be sent for RB test TC_14_2_37	BITSTRING	INT_TO_BIT (173789874769874652133132650, 41984)	
px_RB_Background_64_128	Data to be sent for RB test TC_14_2_27.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 2688)	
px_RB_Background_64_144	Data to be sent for RB test TC_14_2_29.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 3024)	
px_RB_Background_64_2048	Data to be sent for RB test TC_14_2_35.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 41984)	
px_RB_Background_64_256	Data to be sent for RB test TC_14_2_31.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 5376)	
px_RB_Background_64_384	Data to be sent for RB test TC_14_2_32.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 8064)	
px_RB_Background_64_8	Data to be sent for RB test TC_14_2_24.	BITSTRING	INT_TO_BIT (173789874769874652133132650, 1344)	

Parameter Name	Description	Type	Default Value	Supported Value
px_RB_ConvUnknown_64_ConvUnknown_64	Data to be sent for RB test TC_14_2_50	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2560)	
px_RB_DataConversational_14_4	Data to be sent for RB test TC_14_2_15.	BITSTRING	INT_TO_BIT (2473304159874563214258, 576)	
px_RB_DataConversational_28_8	Data to be sent for RB test TC_14_2_12.	BITSTRING	INT_TO_BIT (5896632514789541144447788454777, 1152)	
px_RB_DataConversational_32	Data to be sent for RB test TC_14_2_14.	BITSTRING	INT_TO_BIT (124578963254124555488512323565565465, 1280)	
px_RB_DataSpeech_10_2	Data to be sent for RB test TC_14_2_5.	BITSTRING	INT_TO_BIT (123456789, 99)	
px_RB_DataSpeech_4_75	Data to be sent for RB test TC_14_2_11.	BITSTRING	INT_TO_BIT (9007195689745888, 53)	
px_RB_DataSpeech_5_15	Data to be sent for RB test TC_14_2_10.	BITSTRING	INT_TO_BIT (1523402589632104555, 54)	
px_RB_DataSpeech_5_9	Data to be sent for RB test TC_14_2_9.	BITSTRING	INT_TO_BIT (1234564787987987901247, 64)	
px_RB_DataSpeech_6_7	Data to be sent for RB test TC_14_2_8.	BITSTRING	INT_TO_BIT (258964758964546546546, 76)	
px_RB_DataSpeech_7_4	Data to be sent for RB test TC_14_2_7.	BITSTRING	INT_TO_BIT (78945612345604, 87)	
px_RB_DataSpeech_7_95	Data to be sent for RB test TC_14_2_6.	BITSTRING	INT_TO_BIT (987654256987456987455, 84)	
px_RB_DataStreaming_128_0	Data to be sent for RB test TC_14_2_21	BITSTRING	INT_TO_BIT (123589874569874652132132650, 576)	
px_RB_DataStreaming_28_8	Data to be sent for RB test TC_14_2_16.	BITSTRING	INT_TO_BIT (1238974566954102315468754654654654654654, 1152)	
px_RB_DataStreaming_64_0	Data to be sent for RB test TC_14_2_19	BITSTRING	INT_TO_BIT (123589874569874652132132650, 576)	
px_RB_Interactive_128	Data to be sent for RB test TC_14_2_28.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 2688)	
px_RB_Interactive_128_2048	Data to be sent for RB test TC_14_2_36.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 20992)	

Parameter Name	Description	Type	Default Value	Supported Value
px_RB_Interactive_128_384	Data to be sent for RB test TC_14_2_33.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 4032)	
px_RB_Interactive_144	Data to be sent for RB test TC_14_2_30.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 3024)	
px_RB_Interactive_32_64	Data to be sent for RB test TC_14_2_25.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 1344)	
px_RB_Interactive_32_8	Data to be sent for RB test TC_14_2_23.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 336)	
px_RB_Interactive_384	Data to be sent for RB test TC_14_2_34.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 4032)	
px_RB_Interactive_384_2048	Data to be sent for RB test TC_14_2_37	BITSTRING	INT_TO_BIT (153589874569874652133132650, 20992)	
px_RB_Interactive_64_128	Data to be sent for RB test TC_14_2_27.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 2688)	
px_RB_Interactive_64_144	Data to be sent for RB test TC_14_2_29.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 3024)	
px_RB_Interactive_64_2048	Data to be sent for RB test TC_14_2_35.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 20992)	
px_RB_Interactive_64_256	Data to be sent for RB test TC_14_2_31.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 2688)	
px_RB_Interactive_64_384	Data to be sent for RB test TC_14_2_32.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 4032)	
px_RB_Interactive_64_8	Data to be sent for RB test TC_14_2_24.	BITSTRING	INT_TO_BIT (153589874569874652133132650, 1344)	
px_RB_Speech_12_2_ConvUnknown_64	Data to be sent for RB test TC_14_2_49.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2560)	
px_RB_Speech_12_2_StreamUnknown_57_6	Data to be sent for RB test TC_14_2_45.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2304)	
px_RB_DataStreaming_0_64	Data to be sent for RB test TC_14_2_18.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2560)	

Parameter Name	Description	Type	Default Value	Supported Value
px_RB_DataStreaming_0_128	Data to be sent for RB test TC_14_2_20.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 5120)	
px_RB_DataStreaming_0_384	Data to be sent for RB test TC_14_2_22.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 15360)	
px_RB_Speech_12_2_Interactive_32_8	Data to be sent for RB test TC_14_2_38.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 672)	
px_RB_Speech_12_2_Background_32_8	Data to be sent for RB test TC_14_2_38.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 672)	
px_RB_Speech_12_2_Interactive_32_64	Data to be sent for RB test TC_14_2_39.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 1344)	
px_RB_Speech_12_2_Background_32_64	Data to be sent for RB test TC_14_2_39.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 1344)	
px_RB_Speech_12_2_Interactive_64_64	Data to be sent for RB test TC_14_2_40.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 1344)	
px_RB_Speech_12_2_Background_64_64	Data to be sent for RB test TC_14_2_40.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 1344)	
px_RB_Speech_12_2_Interactive_64_128	Data to be sent for RB test TC_14_2_41.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_Speech_12_2_Background_64_128	Data to be sent for RB test TC_14_2_41.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_Speech_12_2_Interactive_64_256	Data to be sent for RB test TC_14_2_42.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 5376)	
px_RB_Speech_12_2_Background_64_256	Data to be sent for RB test TC_14_2_42.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 5376)	
px_RB_Speech_12_2_Interactive_64_384	Data to be sent for RB test TC_14_2_43.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 8064)	
px_RB_Speech_12_2_Background_64_384	Data to be sent for RB test TC_14_2_43.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 8064)	
px_RB_Speech_12_2_Interactive_128_2048	Data to be sent for RB test TC_14_2_44.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 41984)	
px_RB_Speech_12_2_Background_128_2048	Data to be sent for RB test TC_14_2_44.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 41984)	

Parameter Name	Description	Type	Default Value	Supported Value
px_RB_Speech_12_2_StreamUnknown_0_64	Data to be sent for RB test TC_14_2_46.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2560)	
px_RB_Speech_12_2_StreamUnknown_0_128	Data to be sent for RB test TC_14_2_47.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 5120)	
px_RB_Speech_12_2_StreamUnknown_0_384	Data to be sent for RB test TC_14_2_48.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 15360)	
px_RB_ConvUnknown_64_Interactive_64	Data to be sent for RB test TC_14_2_51.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2560)	
px_RB_ConvUnknown_64_Background_64	Data to be sent for RB test TC_14_2_51.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2560)	
px_RB_ConvUnknown_64_Interactive_64_128	Data to be sent for RB test TC_14_2_52.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_ConvUnknown_64_Background_64_128	Data to be sent for RB test TC_14_2_52.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_ConvUnknown_64_Interactive_128_128	Data to be sent for RB test TC_14_2_53.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_ConvUnknown_64_Background_128_128	Data to be sent for RB test TC_14_2_53.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_Interactive_64_128StreamingUnknown_0k_64k	Data to be sent for RB test TC_14_2_54.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_Background_64_128_StreamingUnknown_0k_64k	Data to be sent for RB test TC_14_2_54.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 2688)	
px_RB_Interactive_64_128StreamingUnknown_0k_128k	Data to be sent for RB test TC_14_2_55.	BITSTRING	INT_TO_BIT (123589874569874652132132650, 5120	
px_RB_Background_64_128_StreamingUnknown_0k_128k	Data to be sent for RB test TC_14_2_55	BITSTRING	INT_TO_BIT (123589874569874652132132650, 5120)	

B.1.10 MMI questions

Table B.10 requests additional information needed for the excution of the MMI commands used in the ATSs, the column 'ATS' indicates in which ATS the question is used.

Table B.10: MMI questions

Required information for MMI question	ATS
How to switch the PLMN selection mode of the UE to automatic selection?	All ATSs
How to switch the PLMN selection mode of the UE to manual selection?	All ATSs
How to select a given PLMN manually?	All ATSs
How to power off the UE?	All ATSs
How to power on the UE?	All ATSs
How to switch off the UE?	All ATSs
How to switch on the UE?	All ATSs
How to insert the USIM card into the UE?	All ATSs
How to remove the USIM card from the UE?	All ATSs
How to check that DTCH is trough connected ?	RRC, SMS, NAS
How to configure UE for a MO telephony call?	RRC, SMS, NAS
How to configure UE for an emergency call?	RRC, SMS, NAS
How to configure UE for a MT telephony call?	RRC, SMS, NAS
How to send any NAS message in order for RRC to receive data?	RRC, SMS, NAS
How to initiate a non call related supplementary service which is supported by the UE?	NAS
How to initiate sending of a mobile originated short message from the UE?	NAS
How to insert 2nd SIM card with short IMSI?	NAS
How to initiate an autocalling call with a given number?	NAS
How to initiate an autocalling call for a number that will be put in the blacklisted list?	NAS
How to reset the autocalling list of blacklisted numbers?	NAS
How to check that the DTMF tone indication has been generated?	NAS
How to enable call refusal on the UE?	NAS
How to check the contents of the received CBS?	SMS
How to check that the Memory Capacity Exceeded Flag has been set to the USIM simulator?	SMS
How to check if the Memory Capacity Exceeded Flag has been unset on the USIM simulator?	SMS
How to check the length and the contents of a given received Short Message ?	SMS
How to check whether the USIM simulator indicated an attempt made by the ME to store the short message in the USIM and return the status response 'Memory Problem'('92 40')?	SMS
How to check whether the USIM simulator indicates an attempt made by the ME to store the short message in the USIM and returns the status response 'OK' ('90 00')?	SMS
How to connect the USIM simulator to the UE?	SMS
How to send an SMS COMMAND message containing a request to delete the previously submitted Short Message?	SMS
How to send an SMS COMMAND message containing an enquiry about the previously submitted SM?	SMS
How to check that NO recalled short Message is displayed?	SMS
How to reply to a short Message with a given length?	SMS
How to insert a USIM card of type B into the UE?	MAC

Annex C (informative): Additional information to IXIT

Additional information may be provided when completing the IXIT questions listed in annex A.

C.1 Identification Summary

Table C.1 is completed by the test laboratory. The item "Contract References" is optional.

Table C.1: Identification Summary

IXIT Reference Number	
Test Laboratory Name	
Date of Issue	
Issued to (name of client)	
Contract References	

C.2 Abstract Test Suite Summary

In table C.2 the test laboratory provides the version number of the protocol specification and the version number of ATS which are used in the conformance testing.

Table C.2: ATS Summary

Protocol Specification	3GPP TS 25.331
Version of Protocol Specification	
Test Specification in prose	3GPP TS 34.123-1
Version of TSS & TP Specification	
ATS Specification	TS 34.123-3
Version of ATS Specification	
Abstract Test Method	Distributed Test Method

C.3 Test Laboratory

C.3.1 Test Laboratory Identification

The test laboratory provides the following information.

Table C.3: Test Laboratory Identification

Name of Test Laboratory	
Postal Address	
Office address	
e-mail address	
Telephone Number	
FAX Number	

C.3.2 Accreditation status of the test service

The test laboratory provides the following information.

Table C.4: Accreditation status of the test service

Accreditation status	
Accreditation Reference	

C.3.3 Manager of Test Laboratory

The test laboratory provides the information about the manager of test laboratory in table C.5.

Table C.5: Manager of Test Laboratory

Name of Manager of Test Laboratory	
e-mail address	
Telephone Number	
FAX Number	
E-mail Address	

C.3.4 Contact person of Test Laboratory

The test laboratory provides the information about the contact person of test laboratory in table C.6.

Table C.6: Contact person of Test Laboratory

Name of Contact of Test Laboratory	
e-mail address	
Telephone Number	
FAX Number	
E-mail Address	

C.3.5 Means of Testing

In table C.7, the test laboratory provides a statement of conformance of the Means Of Testing (MOT) to the reference standardized ATS, and identifies all restrictions for the test execution required by the MOT beyond those stated in the reference standardized ATS.

Table C.7: Means of Testing

Means of Testing

C.3.6 Instructions for Completion

In table C.8, the test laboratory provides any specific instructions necessary for completion and return of the proforma from the client.

Table C.8: Instruction for Completion

Instructions for Completion

C.4 Client

C.4.1 Client Identification

The client provides the identification in table C.9.

Table C.9: Client Identification

Name of Client	
Postal Address	
Office Address	
Telephone Number	
FAX Number	

C.4.2 Client Test Manager

In table C.10 the client provides information about the test manager.

Table C.10: Client Test Manager

Name of Client Test Manager	
Telephone Number	
FAX Number	
E-mail Address	

C.4.3 Client Contact person

In table C.11 the client provides information about the test contact person.

Table C.11: Client Contact person

Name of Client contact person	
Telephone Number	
FAX Number	
E-mail Address	

C.4.4 Test Facilities Required

In table C.12, the client records the particular facilities required for testing, if a range of facilities is provided by the test laboratory.

Table C.12: Test Facilities Required

Test Facilities Required

C.5 System Under Test

C.5.1 SUT Information

The client provides information about the SUT in table C.13.

Table C.13: SUT Information

System Name	
System Version	
SCS Reference	
Machine Configuration	
Operating System Identification	
IUT Identification	
ICS Reference for the IUT	

C.5.2 Limitations of the SUT

In table C.14, the client provides information explaining if any of the abstract tests cannot be executed.

Table C.14: Limitation of the SUT

Limitations of the SUT

C.5.3 Environmental Conditions

In table C.15 the client provides information about any tighter environmental conditions for the correct operation of the SUT.

Table C.15: Environmental Conditions

Environmental Conditions

C.6 Ancillary Protocols

This clause is completed by the client in conjunction with the test laboratory.

In the following tables, the client identifies relevant information concerning each ancillary protocol in the SUT other than the IUT itself. One table for one ancillary protocol.

Based on the MOT the test laboratory should create question proforma for each ancillary protocol in the blank space following each table. The information required is dependent on the MOT and the SUT, and covers all the addressing, parameter values, timer values and facilities (relevant to ENs) as defined by the ICS for the ancillary protocol.

C.6.1 Ancillary Protocols 1

Table C.16: Ancillary Protocol 1

Protocol Name	
Version number	
ICS Reference (optional)	
IXIT Reference (optional)	
PCTR Reference (optional)	

C.6.2 Ancillary Protocols 2

Table C.17: Ancillary Protocol 2

Protocol Name	
Version number	
ICS Reference (optional)	
IXIT Reference (optional)	
PCTR Reference (optional)	

Annex D (informative): PCTR Proforma

PROTOCOL

Conformance Test Report

(PCTR)

Universal Mobile Telecommunication System, UMTS,
User Equipment-Network Access

Layer 3 Signalling Functions

Test Candidate	
Name :	SUT name
Model :	model
H/W version :	hw
S/W version :	sw
Serial No. :	serienr

Client	
Name :	
Street / No. :	
Postal Code / City:	
Country :	

This Test Report shall not be reproduced except in full without the written permission of TEST LAB REFERENCE, and shall not be quoted out of context.

Annex E (informative): TTCN style guide for 3GPP ATS

E.1 Introduction

This annex provides a set of coding standards and development guidelines for use in the development of TTCN abstract test suites for ensuring that user equipment for the 3GPP standard conforms to the relevant core specifications.

The following items are assumed to exist, but their specification is outside the scope of this annex.

- A complete unambiguous prose detailing all test cases to be implemented.
- A complete unambiguous set of core specifications.
- A complete unambiguous detailed description of all the messages that are to be sent.
- A tool or human process that can convert Test Suite Operation Definitions to physical processes within the test system or unit under test.
- An abstracted or generic application programmers interface to all hardware components in the system.
- A tool for the translation and/or compilation of ISO/IEC 9646 [41] series TTCN to run on a test platform.

It is recognised within the context of the 3GPP User Terminal that some of these items are not yet stabilised.

The structure of the present annex maps directly to the guidelines provided in ETR 141 [37]. Rules are repeated in the present annex for convenience, with additional information specific to 3GPP test suite development provided where relevant. For more detailed information or examples about the rules, see ETR 141 [37].

In the present annex, the terms 'should' and 'shall' are frequently used. For the purpose of this annex, the following definitions apply:

- **Shall** means that the rule must be adhered to for all ATS development. If a rule expressed in terms of 'shall' is not followed, either the ATS must be updated so that the rule is followed, or the rule in the coding conventions must be updated to resolve the difference.
- **Should** means that the rule is a guideline. If a rule expressed in terms of 'should' is broken, a brief comment should be provided describing why the guideline does not apply.

E.2 ETR 141 rules and applicability

RULE 1: Statement of naming conventions
Naming conventions should be explicitly stated. Naming conventions should not exist only for a single ATS, and the reader of an ATS should not be forced to "derive" the rules implicitly. The naming conventions should be part of the ATS conventions contained in the ATS specification document.

Names used in the present annex are comprised of a prefix part and a name body part. Conventions for deriving prefixes and name bodies are described after Rule 3 in the present annex.

RULE 2: Coverage of naming conventions
Naming conventions stated should, as a minimum, cover the following TTCN objects: - test suite parameters/constants/variables; - test case variables; - formal parameters; - timers; - PDU/ASP/structured types; - PDU/ASP/structured types constraints; - test suite operations; - aliases; - test case/test step identifiers.

RULE 3: General properties of naming conventions
a) Protocol standard aligned When there is a relationship between objects defined in the ATS and objects defined in the protocol standard, e.g. PDU types, the same names should be used in the ATS if this does not conflict with the character set for TTCN identifiers or with other rules. In case of a conflict, similar names should be used. **b) Distinguishing** The naming conventions should be defined in such a way, that objects of different types appearing in the same context, e.g. as constraint values, can be easily distinguished. **c) Structured** When objects of a given type allow a grouping or structuring into different classes, the names of these objects should reflect the structuring, i.e. the names should be composed of 2 or more parts, indicating the particular structure elements. **d) Self-explaining** The names should be such that the reader can understand the meaning (type/value/contents) of an object in a given context. When suffixes composed of digits are used, it is normally useful to have some rule expressed explaining the meaning of the digits. **e) Consistent** The rules stated should be used consistently throughout the document, there should be no exceptions. **f) Appropriate name length** Following the above rules extensively may occasionally lead to very long names, especially when structuring is used. The names should still be easily readable. When TTCN graphical form (TTCN.GR) is used, very long names are very inconvenient. NOTE: Also, test tools may not be able to implement very long identifier names, which is an important aspect in this context.

E.2.1 Multiple words are separated by upper case letters at the start of each word

Many names consist of more words, and it shall be easy to distinguish the different words building up the same name. For all TTCN Object classes this is done using the case of the letters.

This rule is mandatory for all names appearing in the body of a dynamic behaviour table, and is recommended for all other TTCN object classes.

Generally every word a name consists of shall start with an upper case letter and the rest of this word shall be in lower case letters.

E.g.: "channel" + "description" -> "ChannelDescription".

This rule also applies if a word starts after another upper case letter.

E.g:. "px" + "Cell" + "A" + "Cell" + "Id" -> px_CellACellId.

This rule also applies if the name has a prefix, which is always lower case.

E.g.: A test case variable "sequence" + "number" -> tcv_SequenceNumber.

This rule does not apply if the word is a unit, in which case the word retains it's original case.

E.g.: Power level 1.5 dBm ->PowerLvl1_5dBm.

This rule does not apply if the word in the name is an acronym, in which case the word retains it's normal case.

- If an acronym is followed by another word, an underscore shall be used to separate the acronym from the following word. If an acronym is followed by a number in order to represent an identity (e.g. channel or radio bearer identity) then this acronym is not followed by an underscore.

  E.g.: "this" + "Is" + "SIM" + "Message" + "With" + "CC" + "And" + "RR" + "Things" + "In" + "It" -> "thisIsSIM_MessageWithCC_AndRR_ThingsInIt".

- An exception to acronyms retaining their case is if the name is a field / element / parameter in a structured type / PDU / ASP, in which case it must start with a lower case letter.

  E.g.: "SCH" + "info" + "element" -> "sCH_InfoElement".

- A further exception to acronyms retaining their case is if the name is an ASN.1 constraint, in which case, in which case the first letter is upper case, and the remaining letters are lower case.

For all objects used in the body of dynamic behaviour tables, use of underscores is forbidden, except for the following situations:

- As a replacement for a '.'. E.g. Test case that maps to prose clause 7.2.3.1 -> tc_7_2_3_1.
- To separate prefixes from names.
- To separate acronyms from the following word.
- To separate a number from the following word.
- To replace hyphens when types are re-used / imported from core specifications. This applies to types imported from ASN.1 definitions, and to names derived from table definitions in core specifications.
- To separate an ASP name from the embedded PDU name when the metatype PDU is not used.
  E.g RRC_DataInd_ConnAck for an RRC data indication ASP with an embedded CONNECT ACKNOWLEDGE PDU.

E.2.2 Identifiers shall be protocol standard aligned

To support rule 3(a), the mapping guidelines in table E1 shall be used. This mapping table also supports rule 6.

Table E.1: Mapping guidelines between protocol standards and identifiers

Type	Naming rule
Objects of Structured Type	Shall be derived from the name of the Information Element in the standard, if it corresponds to this (use standard acronyms where appropriate). E.g.: "Window Size super-field" -> "WindowSizeSUFI"
Fields in a Structured Type	Shall be derived from the name of the same field in the corresponding Information Element in the standard. (Acronyms for the entire field name shall not be used) E.g.: "Header Extension Type" -> "headerExtensionType" (not "HE")
Objects of ASP type	Shall be derived from the name of the corresponding Service Primitive in the Standard, using any relevant abbreviations from the present annex. The full name as it appears in the core specification shall be included in parentheses after the name. E.g.: "CRLC-SUSPEND-Conf" -> "CRLC_SuspendCnf (CRLC-SUSPEND-Conf)" If the metatype PDU is not used, the ASP name shall reflect both the ASP, and the embedded PDU name, using an underscore to separate the ASP part from the PDU part. E.g.: DataReq_StartDTMF_Ack for an RRC-DATA-Req with an embedded START DTMF ACKNOWLEDGE PDU
Objects of PDU type	Shall have exactly the same name as the Message it corresponds to in the standard. If this Message is named by more words, they shall be joined, leaving the blanks out E.g.: "AMD PDU" -> "AMDPDU".

E.2.3 Identifiers shall be distinguishing (use of prefixes)

To support rules 2, 3(b), 4, and 5, the prefixes shown in table E2 shall be used for TTCN objects. Prefixes are separated from the name by an underscore to improve readability by clearly separating the prefix from the name. This convention will also support searching operations. For example, a search for all uses of PIXIT parameters in the test suite is possible by searching for 'px_'.

The optional <*protocol*> part shall be included in the name when the object is closely related to the protocol (e.g. PICS, some PIXIT parameters), it is necessary to be unambiguous or improves comprehension significantly (e.g. no need to think about protocol stacks on all used interfaces during reading). The optional <*protocol*> part shall be used for types defined in common modules.

Table E.2: Prefixes used for TTCN objects

TTCN object	Case of first character	Prefix	Comment
Test Suite	Upper	-	
TTCN Module	Upper	-	
Simple Type	Upper	[<protocol>_]	Note 8
Structured Type	Upper	[<protocol>_]	Note 8
Element in Structured Type	Lower	-	
ASN.1 Type	Upper	[<protocol>_]	Note 8
Element in ASN.1 Type	Lower	-	
Test Suite Operation	Upper	o_[<protocol>_]	Notes 1 and 8
TSO Procedural Definition	Upper	o_[<protocol>_]	Notes 1 and 8
Formal Parameter to TSO or TSOP	Upper	p_	
Test Suite Parameter (PICS)	Upper	pc_[<protocol>_]	Note 8
Test Suite Parameter (PIXIT)	Upper	px_[<protocol>_]	Note 8
Test Case Selection Expression	Upper	[<protocol>_]	Note 8
Test Suite Constant	Upper	tsc_[<protocol>_]	Note 8
Test Suite Variable	Upper	tsv_[<protocol>_]	Note 8
Test Case Variable	Upper	tcv_[<protocol>_]	Note 8
PCO Type	Upper	-	
PCO	Upper	-	Note 2
CP	Upper	cp_	Note 2
Timer	Upper	t_[<protocol>_]	Note 8
Test Component	Upper	mtc_[<protocol>_] or ptc_[<protocol>_]	Notes 3 and 8
Test Component Configuration	Upper	-	
ASP Type	Upper	[<protocol>_]	Notes 4 and 8
Parameters within ASP Type	Lower	-	Note 4
PDU Type	Upper	[<protocol>_]	Notes 4 and 8
Fields within PDU Type	Lower	-	Note 4
Encoding Definition	Upper	enc_	
Encoding Variation	Upper	var_	
Invalid Field Encoding Variation	Upper	inv_	
CM Type	Upper	cm_	
Field within CM Type	Lower	-	
Alias	Upper	a_	
ASP constraint	Upper	ca[b|d][s|r|w]_[<protocol>_]	Notes 5 and 8
PDU constraints	Upper	c[b|d][s|r|w]_[<protocol>|AA|108]	Notes 5, 8 and 10
Constraint (other types)	Upper	c[b|d][s|r|w]_[<protocol>_]	Notes 5 and 8
Formal Parameter for a Constraint	Upper	p_	
Test Case Group	Upper	<protocol>/	Note 8
Test Step Group	Upper		
Test Case	Upper	tc_	Note 6
Test Step	Upper	(ts_|pr_|po_)<CN domain>_<protocol>_	Notes 7, 8 and 9
Local tree	Upper	lt_	
Defaults	Upper	<protocol>_	Note 8

NOTE 1: Coding rules are not specified for test suite operation procedural definitions at this stage. These rules will be defined when the need arises

NOTE 2: A prefix is not used for PCO declarations, but is used for CP declarations. This is because PCOs and CPs will only be used in send and receive statements, and PCOs will be used more frequently than CPs. Since a PCO name or a CP name will be used on most behaviour lines, PCO names should be as short as possible – E.g. 2 to 3 characters.

NOTE 3: The prefix is mtc if the component role is MTC, or ptc if the component role is PTC. If multiple PTCs are used, the rest of the identifier will clarify which PTC is being referred to. E.g. ptc_Cell1, ptc_Cell2.

NOTE 4: This applies for both tabular and ASN.1 definitions.

NOTE 5: Constraint prefixes are built up from the following regular expression. c[a][b|d][s|r|w].

- 'c' shall always be present to indicate that the object is a constraint.
- 'a' shall be present for ASP constraints to distinguish them from PDU constraints.
- 'b' shall be present if and only if the constraint is used as a base constraint. (i.e. included in the derivation path of any other constraint).
- 'd' shall be present if the constraint is derived from another constraint.(i.e. has an entry in it's derivation path field)
- 'b' and 'd' cannot both be used in the same constraint, thereby limiting the derivation path to 1.
- For the purpose of the present note, the following definitions are required (see TR 101 666 [27] clause 12.6.2):
    - The term 'field' is used to represent a structured type element, an ASP parameter, or a PDU field.
    - A 'bound field' is a field that either contains a SpecificValue, or is Omitted (-).
    - An 'unbound field' is a field that contains any of the following matching mechanisms: Complement, AnyValue (?), AnyOrOmit (*), ValueList, Range, SuperSet, SubSet, AnyOne (?), AnyOrNone (*), Permutation, Length, or IfPresent.
- 's' may optionally be present if the constraint is only used in send statements. 's' shall not be present if the constraint contains any unbound fields, or any fields chained to a constraint whose prefix includes 'w' or 'r'.
- 'r' may optionally be present if the constraint is only used in receive statements.
- 'w' may optionally be present to indicate that the constraint contains fields that are unbound. Before these constraints are used in SEND events, all unbound fields must either be bound by using a derived constraint, or explicitly assigned a value in the SEND event behaviour line.
- Either 'w' or 'r' shall be used if any fields in the constraint are unbound or are chained to a constraint whose prefix includes 'w' or 'r'.

NOTE 6: Test case names will correspond to the clause in the prose that specifies the test purpose. E.g. tc_7_2_23_2. An additional digit may be specified if more than one test case is used to achieve the test purpose. If an additional digit is required, this probably means that the test prose are not well defined.

NOTE 7: Test steps may optionally use the prefixes pr_ or po_ to indicate that the test step is a preamble or postamble respectively.

NOTE 8: Protocol abbreviations are provided in table E3. Protocol abbreviations may optionally be used to clarify the scope of TTCN objects, or to resolve conflicts when the same name is required by multiple protocols within the ATS. The protocol abbreviation indicates that the object is related to a particular procedure (e.g. an MM procedure). This does not prevent the object from being used by an ATS testing a different protocol. If an object is specific to one ATS, this should be indicated in comments, rather than using a protocol abbreviation (e.g. if a timer is only used in RLC tests this should be stated in the comments, rather than using the abbreviation RLC in the timer name).If two different types exist in the ATS that represent the same information (e.g. IMSI) conversion operations shall be used to ensure consistency between the types. Also, conversion operations shall be used to avoid asking the same PIXIT question twice. For example, if a type is defined as an OCTETSTRING[4] for a NAS protocol, and the same type is represented as a BITSTRING[32] for RRC, a single PIXIT question shall be asked, and conversion operations shall be used to ensure that the same value is used for both types.

NOTE 9: The prefixes CS and PS may optionally be used to indicate that a test step is specific to circuit switched, or packet switched signalling respectively. For test steps specific to the Upper Tester, the prefixes AT or MMI or UT shall be used to indicate that, respectively, AT or MMI or both types of commands are used.

NOTE 10: The prefix AA shall be used for RRC PDU constraints to indicate that it is defined in 3GPP TS 34.123-1 [1] annex A. The prefix 108 shall be used for RRC PDU constraints to indicated that it is defined in 3GPP TS 34.108 [3] clause 9.

Table E.3: Protocol abbreviations for prefixes

Protocol / prefix
BMC
CC
CS
GMM
MAC
MM
PDCP
RLC
RRC
SMS
SS
SUS (Supplementary services)
TC

E.2.4 Identifiers should not be too long (use standard abbreviations)

To assist in keeping TTCN identifiers shorter, table E.4provides a non-exhaustive set of standard abbreviations that shall be used when naming objects that are used in the body of dynamic behaviour tables. Consistent use of abbreviations will improve test suite readability, and assist maintenance.

Table E.4: Standard abbreviations

Abbreviations	Meaning
Acs	access
Acp	accept
Ack	acknowledge
act	activation
addr	address
(re)alloc	(re)allocated, (re)allocation
arg	argument
ass	assignment
auth	authentication
ava	avail, available
bCap	bearer capability
cau	cause
clg	calling
ch	channel
chk	check
ciph	cipher, ciphering
cld	called
clsmk	classmark
cmd	command
cmpl	complete
cnf	confirm
cfg	configuration
conn	connect
ctrl	control
def	default
descr	description
disc	disconnect
enq	enquiry
err	error
(re)est	(re)establish
ext	extended
fail	failure
ho	handover
id	identity / identification

Abbreviations	Meaning
ie	information element
iel	information element length
ind	indication
info	information
init	initialize
lvl	level
loc	location
locUpd	location update
max	maximum
mgmt	management
min	minimum
misc	miscellaneous
mod	modification
ms	mobile station
msg	message
mt	mobile terminal
neigh	neighbour
ntw	network
num	number
orig	origin/-al
pag	page/-ing
params	parameters
perm	permission
phy	physical
qual	quality
rand	random
ref	reference
reg	register
rej	reject
rel	release
req	request
rsp	response
rx	receiver
sel	selection
seq	sequence
serv	service
st	state
sysInfo	system information
sync	synchronization
sys	system
tx	transmitter

RULE 4: Specific naming rules for test suite parameters/constants/variables test case variables and formal parameters
a) The name should reflect the purpose/objective the object is used for. b) If the type is not a predefined one, it is useful that the name reflects the type, too. c) It could be useful, that the individual naming conventions are not the same for all object classes this rule applies to. e.g. use upper case letters for test suite parameters/constants, and use one of the other possibilities presented in ETR 141 [37] example 1 for other object classes.

See also ETR 141 [37] clauses 5.1 to 5.4 for further discussion on naming test suite parameters.

RULE 5: Specific naming rule for timers
If the timer is not defined in the protocol to be tested, the name should reflect the objective of the timer used for testing. NOTE: There is no need to indicate the object type "timer" in the name, since timers only occur together with timer operations

RULE 6: Specific naming rule for PDU/ASP/structured types
As far as applicable, derivation rules or mapping tables should be used to relate the names of the types to the corresponding objects in the protocol or service definition. NOTE: There may be types, e.g. erroneous PDU types, that do not relate to an object in the protocol or service definition.

Whenever names of types are derived from ASN.1 type definitions provided in the core specifications, the names shall remain the same as the ASN.1 specifications, and references shall be provided in the comment fields.

RULE 7: Specific naming rule for PDU/ASP/structured types constraints
Rules should be stated to derive the names from the names of the corresponding type definitions. It is often possible to use the type name plus an appropriate suffix reflecting the specific constraint value. In case of lengthy names, useful abbreviations or a defined numbering scheme can be chosen.

Constraint names begin with the appropriate prefix, followed by the first letter of each word in the type, followed by words describing the peculiarity of the constraint. E.g. Type = RadioBearerSetupPDU, constraint name could be cb_RBSP_GenericUM_DTCH.

RULE 8: Specific naming rule for test suite operations
The name should reflect the operation being performed. i.e. the name should indicate an activity, not a status. This can be achieved e.g. by using appropriate prefixes like "check", "verify", etc.

RULE 9: Specific naming rule for aliases
The name should reflect that aspect of its expansion, that is important in the situation where the alias is used. Derivation rules should be provided to derive the alias name from its macro expansion or from the name of an embedded ASP / PDU.

See also ETR 141 [37] clauses 6.3.6 and 9 for further guidelines on naming aliases.

RULE 10: Specific naming rule for test steps
The name should reflect the objective of the test step.

RULE 11: Selecting the ASN.1 format for type definitions
a) If the protocol standard uses ASN.1 to specify the PDUs, the ATS specifier should also use ASN.1. b) If the protocol standard does not use ASN.1, check carefully whether features of ASN.1 that the tabular format of type definition does not present are necessary in the ATS, or could ease the design and understanding of the definitions as a whole. Check especially whether fields or parameters have to be specified, the order of appearance of which, in a received ASP/PDU, cannot be predicted. If any of these conditions apply, use ASN.1 for type and ASP/PDU type declarations. c) Use the option of "ASN.1 ASP/PDU type Definitions by Reference" whenever applicable. d) Example 14 shows a compatibility problem that could occur, when ASN.1 type declarations as well as tabular type declarations are used in an ATS. Use the ATS Conventions to describe how this compatibility problem is handled in the ATS, i.e. whether in expressions and assignments entities defined in ASN.1 are only related to entities defined in ASN.1 or not.

Names of ASN.1 objects shall be kept the same as the core specifications in this case, even where the names are at odds with the naming conventions adopted for other TTCN objects.

RULE 12: Further guidelines on type definitions
a) Use simple type or ASN.1 type definitions whenever an object of a base type with given characteristics (length, range, etc.) will be referenced more often than once.
b) Use the optional length indication in the field type or parameter type column of structured type and ASP/PDU type definitions whenever the base standard/profile restricts the length.
NOTE 1: This can often be achieved by references to simple types.
c) Map the applicable ASPs/PDUs from the service/protocol standard to corresponding ASP/PDU type definitions in the ATS.
NOTE 2: It may happen that not all ASPs/PDUs of a service/protocol standard are applicable to a particular ATS for the related protocol. It may also happen that additional ASP/PDU type declarations are necessary, e.g. to create syntactical errors.
d) Map the structure of ASPs/PDUs in the service/protocol standard to a corresponding structure in the ATS.
NOTE 3: This mapping is not always one-to-one, e.g. because a field in the PDU definition of the protocol standard is always absent under the specific conditions of an ATS. But it should normally not happen, that a structured element in the protocol standard is expanded using the "<-" macro expansion, so that the individual fields are still referenced, but the structure is lost in the ATS.

RULE 13: Specification of test suite operations
a) Use a test suite operation only if it cannot be substituted by other TTCN constructs.
b) Write down the rationale/objective of the test suite operation. Reference standards if applicable.
c) Classify and simplify algorithm. Split test suite operation if too complex.
d) Choose an appropriate specification language depending on the rationale/objective: - predicates for Boolean tests; - abstract data types for manipulation of ASN.1 objects; - programming languages for simple calculation.
e) Check/proof the test suite operation: - is the notation used known/explained; - are all alternative paths fully specified; - is the test suite operation returning a value in all circumstances; - are error situations covered (empty input variables, etc.).
f) State some evident examples.

E.2.5 Test suite operations must not use global data

All information required by test suite operations must be passed as formal parameters. This includes test suite variables, test case variables, test suite parameters, and constraints.

RULE 14: General aspects of specifying constraints
a) Develop a design concept for the complete constraints part, particularly with respect to the "conflicting" features as indicated in items i) to iv) and including naming conventions (see ETR 141 [37] clause 6).
b) Make extensive use of the different optional "Comment" fields in the constraint declaration tables to highlight the peculiarity of each constraint.

RULE 15: Relation between base constraints and modified constraints
a) Define different base constraints for the send- and receive direction of a PDU (when applicable).
b) Use modified constraints preferably when only a small number of fields or parameter values are altered with respect to a given base.
NOTE 1: For SEND events the creation of a further modified constraint can sometimes be avoided, if an assignment is made in the SEND statement line, thus overwriting a particular constraint value.
c) Design the relation between base constraints and modified constraints always in connection with parameterization of constraints (see the two subsequent subclauses).
NOTE 2: Additional parameters in a constraint, introduced to avoid the declaration of further base/modified constraints can reduce the amount of constraints needed in an ATS, but then the constraint reference is getting more and more unreadable.
d) When modified constraints are used, keep the length of the derivation path small. The length of the derivation path (resulting from the number of dots in it) is a kind of nesting level, and it is known from experience that a length greater than 2 is normally difficult to overview and maintain.

Modified constraints should not have a derivation path longer than 1. A modified constraint should not alter more than 5 values with respect to a given base constraint. If a constraint is used as a base constraint, it must have the prefix 'cb', to warn test suite maintainers / developers that any changes to this constraint may cause side effects.

Note that if an existing constraint without the 'cb' prefix is to be used as a base constraint, either a new, identical constraint with an 'cb' prefix must be created, or the existing constraint must be renamed to include the 'cb' prefix in all places it is referenced in the test suite.

RULE 16: Static and dynamic chaining
a) Make a careful evaluation of which embedded PDUs are needed in ASPs/PDUs, in which (profile) environment the ATS may operate and which kind of parameterization for other parameters/fields is needed, to find an appropriate balance between the use of static and/or dynamic chaining in a particular ATS. b) When the ATS is used in different profile environments and the types and values of embedded PDUs cannot be predicted, dynamic chaining is normally the better choice. c) When static chaining is used, chose the name of the ASP/PDU constraint such that it reflects the peculiar value of the embedded PDU (see also the clause on naming conventions in ETR 141 [37]).

RULE 17: Parameterization of constraints
a) Make a careful overall evaluation of which field/parameter values are needed in ASPs and PDUs to find an appropriate balance between the aim of a comparably small number of constraint declarations and readable and understandable constraint references. b) Keep the number of formal parameters small. Keep in mind, that the number of formal parameters in structured/ASN.1 types Constraints will add up to the total number of ASP/PDU constraints. A clear border for the number of formal parameters cannot be stated, but it is known from experience that a number bigger than 5 normally cannot be handled very well.

Constraints should not be passed more than five parameters. Instead, more constraints should be defined. Related parameters can be grouped in new structured types to reduce the number of parameters that must be passed to constraints.

NOTE 1: The value five has been selected based on the recommendation in ETR 141 [37] rule 17. If more parameters are required, we can update this rule, or use more than 5 parameters, and provide documentation indicating why more parameters are required.

A constraint should not be passed parameters to that are not processed in that constraint. If for example a parameter is to be passed from a PDU constraint to a structured type constraint then the PDU constraint should be made specific and not have that parameter passed. The reason for this is that no editors as yet can trace through this mechanism and it becomes very difficult in a complex suite to see exactly what is being passed.

For example:

```
PduA ::= SEQUENCE {
  infoElement1  InformationElementType1,
  infoElement2  INTEGER
}

InformationElementType1 ::= SEQUENCE {
  field1  INTEGER,
  field2  INTEGER
}

cb_PATypical( p_Field1: INTEGER; p_Field2: INTEGER ) ::= {
  infoElement1  c_IET1Typical( p_Field1 ),
  infoElement2  pField2
}

c_IET1Typical( p_Field1: INTEGER ) ::= {
  field1  p_Field1,
  field2  5
}
```

In the example constraint cb_PATypical, passing p_Field1 through to a nested constraint is not allowed, but the use of p_Field2 is acceptable.

RULE 18: Constraint values
a) Use comments to highlight the peculiarity of the value, especially when the value is a literal, whose meaning is not apparent. b) Use test suite constants instead of literals, when appropriate. Normally not all literals can be defined as Test Suite Constants, but a rule by thumb is: if a literal value of a given type occurs more than once (as a constraint value or more generally in an expression), then it is useful to define it as a Test Suite Constant, letting the name reflect the value. c) Use the length attribute when possible and when the length is not implicit in the value itself or given by the type definition (e.g. for strings containing "*").

RULE 19: Verdict assignment in relation to the test body
Make sure that verdict assignment within a default tree is in relation to the test body. If an unsuccessful event arising in the test body is handled by the default tree, then assign a preliminary result "(FAIL)" within the corresponding behaviour line of the default tree. If the position of the unsuccessful event is not in the test body, assign a preliminary result "(INCONCLUSIVE)". If the behaviour line handling the unsuccessful event is a leaf of the default tree, assign a final verdict instead.

RULE 20: Test body entry marker
The entry of the test body should be marked.

RULE 21: State variable
For realizing test purposes dependent on protocol states, use a variable to reflect the current state of the IUT.

RULE 22: State checking event sequences
Combine event sequences used for checking a state of the IUT within test steps.

RULE 23: Easy adaptation of test steps to test cases
For easy adaptation of a test step to test case needs, parameterize the constraints used within a test step.

Test steps may be parameterised, but with no more than five parameters. See also ETR 141 [37] clause12.2 and rule 28. Related parameters can be grouped in new structured types to reduce the number of parameters that must be passed to constraints.

NOTE 2: Again, the value five has been selected based on the recommendation in ETR 141 [37] rule 17. If more parameters are required, we can update this rule, or use more than 5 parameters, and provide documentation indicating why more parameters are required.

RULE 24: Minimizing complexity of test steps
Minimize the complexity of test steps either by restricting the objective of a test step to atomic confirmed service primitives or by separating event sequences, which build different "logical" units into different test steps.

RULE 25: Nesting level of test steps
Keep the nesting level of test steps to a minimum.

RULE 26: Recursive tree attachment
Avoid recursive tree attachment. Where possible, use loops instead of recursive tree attachments.

RULE 27: Verdict assignment within test steps
If verdicts are assigned within a test step, guarantee at least the partial (i.e. not general) re-use of the test step.

RULE 28: Parameterized test steps
Use parameterized test steps to ensure re-use of test steps within test cases for different needs.

RULE 29: Combining statements in a sequence of alternatives
If there is no Boolean expression included in an alternative sequence, a statement of type UCS (unconditional statement) should never be followed by a statement of type UCS or CS (conditional statement) within a sequence of alternatives.

RULE 30: Using relational expressions as alternatives
a) A relational expression should never restrict the value range of a preceding relational expression in the same alternative sequence using the same variable. b) The value range of a relational expression should be different from the whole value range of all preceding relational expressions in the same alternative sequence using the same variable.

RULE 31: Loop termination
Do not use conditions for terminating loops, which depend only on the behaviour of the IUT.

RULE 32: Avoiding deadlocks
a) Make sure that each alternative sequence of receive events contains an OTHERWISE statement (without any qualifier) for each PCO. b) Make sure that each alternative sequence of receive events contains at least one TIMEOUT event (implying that a corresponding timer was started).

A set of alternatives using qualifiers shall always include an alternative containing the qualifier [TRUE], to provide a default behaviour if none of the qualifiers match.

For example:

```
[ tcv_Value = 1 ]
  AM ! ASP_ForValue1
    ...
[ tcv_Value = 2 ]
  AM ! ASP_ForValue2
    ...
[ TRUE ]
  AM ! ASP_ForOtherValues
    ...
```

RULE 33: Straightforward specification of test cases
a) Use only event sequences leading to the test body within a preamble. b) Handle all event sequences not leading to the test body within the default tree of the test case/step. c) If the very same event sequence can be used to transfer the IUT from each possible state to the idle state, then realize this event sequence as a postamble.

RULE 34: Test component configuration declaration
Avoid recursive test component configuration declarations.

RULE 35: Default trees with RETURN statement
Special care should be taken by using a RETURN statement within a default tree in order to avoid an endless loop resulting from the expansion of the default tree.

E.3 3GPP ATS implementation guidelines

This clause provides a set of guidelines that must be followed during ATS development. In general, these guidelines are intended to prevent developers from making common errors, or discuss considerations that must be taken into account before using specific features of the TTCN language.

E.3.1 Test case groups shall reflect the TSS&TP document

Test groups shall be used to organise the test cases in the same way as the test purposes are structured in the prose specification.

The general structure of the test groups should be in the following format.

<protocol>/<group>/<subgroup>

E.g. RLC/UM/Segmentation/LengthIndicator7bit/

E.3.2 Test case names correspond to the clause number in the prose

Test case names are derived directly from the clause number in the prose specification. Decimal points between digits in the clause number are replaced with underscores. E.g. the test case name for the test purpose specified in clause 7.2.3.2 of 3GPP TS 34.123-1 [1] is tc_7_2_3_2. If more than one test case is required to achieve a test purpose, an additional digit may be added. See also ETR 141 [37] clause 6.3.7

E.3.3 Use standard template for test case and test step header

Table E.5 illustrates how the Test Case dynamic behaviour header fields should be used.

Table E.5: Template for TTCN test case table header

<table>
<tr><th colspan="2">Field</th><th colspan="4">Contents</th></tr>
<tr><td colspan="2">Test Case Name:</td><td colspan="4">tc_NUMBER_OF_TESTCASE
The number of the test case, which is used in the name of the test case, is the number it has in the prose specification.
e.g.: "tc_26_13_1_3_1"</td></tr>
<tr><td colspan="2">Group:</td><td colspan="4">Is automatically filled and cannot be changed</td></tr>
<tr><td colspan="2">Purpose:</td><td colspan="4">This is taken directly from the prose specifications.</td></tr>
<tr><td colspan="2">Configuration:</td><td colspan="4">As required if concurrent TTCN is being used.</td></tr>
<tr><td colspan="2">Default</td><td colspan="4">The appropriate default</td></tr>
<tr><td colspan="2">Comments:</td><td colspan="4">First line contains:
Specification: The names and clauses of relevant core specifications.
Next line contains:
Status: OK / NOT OK (+explanation if not ok) / Version number / Validated / Reviewed etc…
E.g.: Status: OK
Rest of lines give comments as:
What has to be done before running this test?
E.g.: 1. Generic setup procedure must be completed before running this test.
Any special information about what might be needed for the testing system, like specific requirements for the testing system, specific hacks, certain settings etc. This field should be short (if long description is needed it must be put into Detailed Comments)</td></tr>
<tr><td colspan="2">Selection Ref:</td><td colspan="4">The appropriate test case selection expression.</td></tr>
<tr><td colspan="2">Description:</td><td colspan="4">Optional. Max 4 lines. If available, this should be the title of the prose clause. Note 1</td></tr>
<tr><td>Nr</td><td>Label</td><td>Behaviour Description</td><td>Constraints Ref</td><td>Verdict</td><td>Comments</td></tr>
<tr><td>1</td><td></td><td>Note 3</td><td>Note 3</td><td></td><td>Note 2</td></tr>
<tr><td colspan="2">Detailed Comments</td><td colspan="4">Contains detailed information about test steps + additional information Note 2</td></tr>
<tr><td colspan="6">NOTE 1: The description field in the test case / step header is used to generate the test suite overview, and should only include a brief overview of the test case / step with a maximum of 4 lines. For a more detailed description of the test case / step algorithm / parameters etc, the comments or detailed comments fields should be used.
NOTE 2: The comments field for each behaviour line should usually consist of a number that is a reference to a specific numbered comment in the detailed comments field. If this extra level of indirection reduces readability, brief comments can be used in the comments field for each behaviour line.
NOTE 3: If entries in the behaviour description or constraints reference column contain lists with more than one element, carriage returns should be used between list elements to prevent the line from becoming too long.</td></tr>
</table>

Table E.6 illustrates how the Test Case dynamic behaviour header fields should be used.

Table E.6: Template for TTCN test step table header

<table>
<tr><td colspan="2">Test Step Name</td><td colspan="4">ts_TestStepName(p_Param1: Param1Type; p_Param2: Param2Type)</td></tr>
<tr><td colspan="2">Group</td><td colspan="4">Is automatically filled and cannot be changed</td></tr>
<tr><td colspan="2">Objective</td><td colspan="4">The objective of the test case. Provides a brief summary of the functionality of the test step.</td></tr>
<tr><td colspan="2">Default</td><td colspan="4">The appropriate default</td></tr>
<tr><td colspan="2">Comments</td><td colspan="4">A detailed description of the test step, including the relevant items from the following categories:

Algorithm
A detailed description of the algorithm / principles used within the test step

Parameters:
A description of each of the parameters passed to the test step, including the purpose of the parameter, valid values, restrictions etc.

Preconditions
The required state of the UE and / or SS before using this test step, including test steps that should be executed before using the present test step, and a description of all test case variables that must contain appropriate values before using this test step.

Postcondidions
The expected state of the UE and / or SS after using this test step, including a description of all test case variables that will be modified by this test step.

NOTE: It is too difficult to maintain the list of variables required / affected by nested test steps, so it is the users responsibility to check which variables are required / affected by nested test steps.</td></tr>
<tr><td colspan="2">Description</td><td colspan="4">Optional. Max 4 lines. Note 1</td></tr>
<tr><td>Nr</td><td>Label</td><td>Behaviour Description</td><td>Constraints Ref</td><td>Verdict</td><td>Comments</td></tr>
<tr><td>1</td><td></td><td>Note 3</td><td>Note 3</td><td></td><td>Note 2</td></tr>
<tr><td colspan="2">Detailed Comments</td><td colspan="4">Contains detailed information about test steps + additional information Note 2</td></tr>
<tr><td colspan="6">NOTE 1: The description field in the test case / step header is used to generate the test suite overview, and should only include a brief overview of the test case / step with a maximum of 4 lines. For a more detailed description of the test case / step algorithm / parameters etc, the comments or detailed comments fields should be used.
NOTE 2: The comments field for each behaviour line should usually consist of a number that is a reference to a specific numbered comment in the detailed comments field. If this extra level of indirection reduces readability, brief comments can be used in the comments field for each behaviour line.
NOTE 3: If entries in the behaviour description or constraints reference column contain lists with more than one element, carriage returns should be used between list elements to prevent the line from becoming too long.</td></tr>
</table>

E.3.4 Do not use identical tags in nested CHOICE constructions

A nested CHOICE requires tags in the different alternative type lists to differ (see ISO/IEC 8824 [29], clause 24.4, example 3, INCORRECT). 'The tag shall be considered to be variable, ... becomes equal to the tag of the "Type" ... from which the value was taken'.

EXAMPLE: components are defined in a nested CHOICE construction, but no distinguishing tags are used to make the difference between component types, i.e. tags for different types turn out to be identical.

```
Component ::=   CHOICE {
  gSMLocationRegistration_Components   GSMLocationRegistration_Components,
  gSMLocationCancellation_Components   GSMLoactionCancellation_Components,
  ...
}

GSMLocationRegistration_Components ::= CHOICE {
  gSMLocationRegistration_InvokeCpt    [1] IMPLICIT GSMLocationRegistration_InvokeCpt,
  gSMLocationRegistration_RRCpt        [2] IMPLICIT GSMLocationRegistration_RRCpt,
  gSMLocationRegistration_RECpt        [3] IMPLICIT GSMLocationRegistration_RECpt,
  gSMLocationRegistration_RejectCpt    [4] IMPLICIT RejectComponent
}
```

```
GSMLocationCancellation_Components ::=   CHOICE {
  gSMLocationCancellation_InvokeCpt  [1] IMPLICIT GSMLocationCancellation_InvokeCpt,
  gSMLocationCancellation_RejectCpt  [4] IMPLICIT RejectComponent
}
```

gSMLocationRegistrationInvokeCpt and gSMLocationCancellation_InvokeCpt have the same tag and can therefore not distinguished anymore. Note that ITEX 3.5 does not report this error.

E.3.5 Incorrect usage of enumerations

Enumerations may contain distinct integers only (see ISO/IEC 8824 [29], clause 15.1)

EXAMPLE: TypeOfNumber containing a NamedValueList in which there are non-distinct values.

```
TypeOfNumber ::=    ENUMERATED {
.....,
  internationalnumber  (1),
  level2RegionalNumber (1),
  nationalNumber       (2),
  level1RegionalNumber (2),
......
 }
```

E.3.6 Structured type as OCTETSTRING should not be used

"It is required to declare all fields of the PDUs that are defined in the relevant protocol standard, …"
TR 101 101 [38] TTCN specification clause 11.15.1

EXAMPLE 1: The ISDN Bearer Capability Information Element (BCAP) contents is defined as OCTETSTRING.

EXAMPLE 2: Usage of data type BITSTRING [7..15] as data type of the Call Reference (= 7 bits or =15 bits, but not 8 bits for example) does not correspond to the specification !!).

E.3.7 Wildcards in PDU constraints for structured types should not be used

Contrary to popular belief, TR 101 666 [27] does not support the use of wildcards for TTCN ASP parameters, or TTCN PDU fields whose type is structured. It is not clearly stated if wildcards are permitted for TTCN structured type elements whose type is structured but it is assumed that they are not permitted because the semantics for this are not clearly specified.

Note that this does not apply to ASN.1 Type definitions, ASPs, or PDUs.

Most tools do support wildcards for TTCN ASP parameters / TTCN PDU fields / TTCN structured type elements whose type is structured, but there is ambiguity between implementations since the semantics are not clearly specified in the core specification.

This feature is commonly used by TTCN developers, and is present in many existing test suites, including the 3GPP test suite, and in constraints that are being re-used from GERAN tests.

One problem with values '?' and '*' in constraints where they are used to indicate values of structured types, is that they would allow any combinations of values - even incorrect ones - which is not admissible according to the specifications. It is to be kept in mind that in tabular form each field is optional! It would be better to create and use an "any"-constraint which would deal with all the fields in detail (mandatory, IF PRESENT, etc.).

For the purpose of the present annex, the following rules shall apply:

1. '?' shall not be used to indicate values of TTCN ASP parameters / TTCN PDU fields / TTCN structured type elements whose type is structured. Known TTCN implementations differ significantly in their implementation of this feature.

2. '*' shall not be used for TTCN PDU fields, or TTCN ASP parameters whose type is structured (i.e. at the top level).

3. '*' is permitted but discouraged for structured type elements whose type is structured. Note that this may result in ambiguous behaviour between TTCN implementations because the semantics are not specified in TR 101 666 [27].

4. One of the following two options shall be used as an alternative to using a '?' for a TTCN ASP parameter / TTCN PDU field / TTCN structured type element whose type is structured.

   4.1 Option 1: Use '*' instead (only applicable to structured type elements due to rules 2 and 3 above).

WARNING: This may result in the situation where a UE omits a mandatory field, but passes the test anyway, and / or different behaviour depending on the TTCN tool used.

   4.2 Option 2 (preferred option; supported by TR 101 666 [27]): Use an 'any' constraint, in conjunction with IF PRESENT if appropriate (whole TTCN ASP parameters / TTCN PDU fields / TTCN structured type elements may be omitted according to TR 101 666 [27]). This means that the constraint value specified for the parameter / field / element shall be a reference to another constraint of the appropriate structured type, which may in turn use wildcards for each of it's elements according to the rules specified in the present annex.

E.3.8 TSOs should be passed as many parameters as meaningful to facilitate their implementation

Parameters should be passed to TSOs to facilitate the TSO realization. If a TSO is used in various contexts, this should be reflected in the parameters passed to the TSO. Specifically, TSOs operating on well-defined (parameterized) constraints should take these constraints (including relevant parameters) as parameters if required.

BAD EXAMPLE: In this example, the TSO may be used in many contexts, but no information is passed to the TSO, which makes TSO realization difficult.

		L?SETUPr (... tcv_invokeId := TSO_GET_INVOKEID ( ), ...)	Sr (SU_GR3(GSM_IncomingCallMMInfo_In voke(...)))		

GOOD EXAMPLE: In this case, the TSO is provided with information about the data object from which the invoke Id is to be extracted, and the type of component from which the invoke Id is to be extracted is identified by passing the component constraint.

		L?SETUPr (... tcv_invokeId := TSO_GET_INVOKEID (DL_DataInd_Setup.msg, GSM_IncomingCallMMInfo_Invoke(...)), ...)	Sr (SU_GR3(GSM_IncomingCallMMInfo_In voke(...)))		

To calculate the invocation identification and store the result in variable tcv_invokeId the TSO has to be provided with information about the data object from which the invoke Id is to be extracted. PDU constraint SU_GR3 may contain several components. In the specific situation only one of these components is relevant.

Depending on the nature of the TSO, passing the received value, or a subcomponent of the received value may be more appropriate than passing the constraint.

E.3.9 Specification of Encoding rules and variation should be indicated

TTCN does not mandate encoding rules, although TTCN foresees that applicable encoding rules and encoding variations can be indicated for the data structures used in a test suite.

There are standards defining encoding rules, e.g. the ITU-T Recommendation X.680 [39] series. However, the type of encoding called "Direct Encoding" - a bit-by-bit-mapping from the data definitions onto the data stream to be transmitted - is not defined anywhere. It therefore needs a "home".

TTCN should therefore define which encoding rules may legally be used by TTCN test suite specifiers. All the encoding rules defined in the ITU-T Recommendation X.680 [39] series should be contained in this repertoire. Additionally an encoding rule called Direct Encoding is needed in particular for tabular TTCN.

ITU-T Recommendation X.680 [39] allows to encode data objects using different length forms (short, long, indefinite). These could be used alternatively as encoding variations. Another encoding variation could be the "minimum encoding", accepting any of the length forms in reception, and using the shortest of the available forms in sending. The variation actually used has to be described somewhere (in the ATS).

E.3.10 Use of global data should be limited

The Phase 2 ATS became extremely complex due to the global definition of data. Data should be defined locally where possible if the language allows, alternatively the names of global constraints could be given prefixes to indicate their use.

E.3.11 Limit ATS scope to a single layer / sub-layer

Separate ATSs should be produced to test each Layer and perhaps sub Layer. By doing this preambles and common areas particular to one sub Layer can be confined to one test suite and parallel development of test suites can be facilitated.

E.3.12 Place system information in specially designed data structures

System Information data could be stored in specially defined data structures, use of these structures to build PDUs may help to ensure that a consistent set of data is transmitted in all the channels in a cell.

E.3.13 Place channel configuration in specially designed data structures

Likewise the configuration of a 'channel' could be stored in similar structures. This data can then be used to configure the test system and to build Assignment messages to the UE under test. This may help avoid the situation where the TTCN creates one channel and unintentionally commands the mobile to a different, non-existent, channel.

E.3.14 PICS / PIXIT parameters

It is desirable to limit the scope of PICS / PIXIT parameters.

A default value shall be provided in the PIXIT document for all PIXIT parameters.

PICS / PIXIT parameters shall not include structured types. If a structured parameter is required, several parameters shall be used, one for each simple element within the type, and a constraint shall be created to combine the simple parameters into a structured type.

For example, to use the following structured type as a parameter.

Type Name	LocAreaId_v		
Encoding Variation			
Comments	Location Area Identification Value 3GPP TS 24.008 [9] clause 10.5.1.3		
Element Name	**Type Definition**	**Field Encoding**	**Comments**
mcc	HEXSTRING[3]		MCC 3 digits
mnc	HEXSTRING[3]		MNC 3 digits
lac	OCTETSTRING[2]		LAC
Detailed Comments			

The following three PIXIT parameters should be defined: Parameter Name	Type	PICS/PIXIT Ref	Comments
px_LACDef	OCTETSTRING	PIXIT TC	default LAC
px_MCCDef	HEXSTRING	PIXIT TC	default MCC
px_MNCDef	HEXSTRING	PIXIT TC	default MNC

And then the following constraint can be used to combine the simple parameters into a structured parameter.

Constraint Name	cb_LocAreaIdDef_v		
Structured Type	LocAreaId_v		
Derivation Path			
Encoding Variation			
Comments			
Element Name	**Element Value**	**Element Encoding**	**Comments**
mcc	px_MCCDef		
mnc	px_MNCDef		
lac	px_LACDef		
Detailed Comments			

E.3.15 Dynamic vs. static choices

Don't use wildcards for static choice constraints. For example, a type that is similar for FDD and TDD should have 2 type definitions, rather than a single type that uses an ASN.1 choice. Then in the TTCN, the correct type should be selected based on test suite parameters.

E.g.:

```
[ pxUseTddMode ] AM ! TddSpecificAsp
  AM ?
    ...
[ pxUseFddMode ] AM ! FddSpecificAsp
  AM ? …
    ...
```

E.3.16 Definition of Pre-Ambles and Post Ambles

Test cases should, as far as possible, use one of a set of standard pre-ambles to place the user equipment in its initial conditions. These pre-ambles should align with the generic setup procedures in the conformance specification. All non-standard pre-ambles should be identified and added to the pre-amble library.

With pre-ambles readability is very important so they should not use other test steps to send message sequences, and they should be passed as few parameters as possible. This also makes the results log easier to read.

The prose message sequence charts should be analysed, and a catalogue of common ways in which the test cases can terminate (correctly or incorrectly) created. This catalogue should be used to create a set of post-ambles. All final verdicts should be assigned in the post-ambles.

Wherever possible, a post-amble should return the test system and the User Equipment under test to a known idle state.

E.3.17 Use test steps to encapsulate AT and MMI commands

When the same AT or MMI command is to be used more than once within a test suite, the command should be placed within a test step, to ensure that the same information is provided consistently. The main intention of this guideline is to ensure that MMI commands provided to the user are consistent, and can be changed easily if required.

For example, a test step similar to the one illustrated in table E.7 should be created and attached so that the same information is provided to the user each time the test step is used, and the string to be sent only exists in one place within the test suite.

Table E.7: Example test step to encapsulate AT / MMI commandsDefault behaviour

Test Step Name		ts_AT_MMI_Example			
Group					
Objective		Send an MMI command instructing the user to insert the USIM card into the UE.			
Default					
Comments		Encapsulate an AT / MMI command within a test step to ensure that the same information is used consistently, and the information only exists in one place within the test suite.			
Description					
Nr	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		Ut ! MMI_CmdReq	ca_MMICmdReq (" Please insert the USIM card into the UE ")		
2		Ut ? MMI_CmdCnf	ca_MMICmdCnf		

Defaults are test steps that are executed when ever a receive event occurs that is not expected. Not expected means that it does not match any of the defined ASP constraints at that point in the test case. The default behaviour used in test case is defined in the test case declaration. They can be defined to stop the test case by calling a standard post-amble or receive the event as OTHERWISE and RETURN back to step where the unexpected event occurred.

A strategy for dealing with unexpected behaviour involving consistent use of defaults should be developed, and applied to test cases wherever possible.

If during a test case or test step it is necessary to change the default behaviour, the ACTIVATE statement may be used.

E.3.18 Use system failure guard timers

A timer should be set at the beginning of each test case to guard against system failure. Behaviour on expiry of this timer should be consistent for all test cases.

E.3.19 Mapping between prose specification and individual test cases

The ATS should map one-to-one between test cases and tests as described in 3GPP TS 34.123-1 [1]. A method for ensuring that the two specifications track each other needs to be defined.

E.3.20 Verdict assignment

E.3.20.1 General

Final verdicts shall only be used to indicate test case errors, or when unexpected UE behaviour occurs such that it not sensible to continue the test. When a test case reaches a leaf node, the test case ends, and the current preliminary verdict is assigned. At least one preliminary verdict shall be assigned for every test case. If a test case terminates and no final or preliminary verdicts have been assigned, the current value of the predefined variable R will be 'none', and a test case error is recorded instead of a final verdict.

Labels shall be used for every line in which a verdict is posted to improve the traceability of the conformance log produced when the test case is executed. These labels should be kept short, since they appear in the dynamic behaviour tables.

All test suites shall make use of a global boolean variable, defined in the common module, called tcv_TestBody. tcv_TestBody is updated within each test case to indicate if the test body is currently being executed. tcv_TestBody is referenced in defaults and test steps to assign a preliminary inconclusive verdict when unexpected events occur outside of the test body, or a preliminary failure verdict when unexpected events occur within the test body.

The initial value in the declaration of the test case variable tcv_TestBody shall be FALSE. The variable will be bound to this value when the ATS is initialised, and will be re-bound to this value after termination of each test case, ready for execution of the next test case.

E.3.20.2 Test cases

A line similar to line 3 in table E.8 shall be used in all test cases to set tcv_TestBody to TRUE. This line shall have the label TBS to indicate the Test Body Start point.

A line similar to line 6 in table E.8 shall be used in all test cases to set tcv_TestBody to FALSE. This line shall have the label TBE[N] to indicate the Test Body End point. A number N (with one or more digits) may optionally be appended to the label to distinguish between multiple test body end points. If the number of possible test sequences makes management of the tcv_TestBody variable too difficult, the variable can be set to TRUE at the beginning of the test. In this case, a comment shall be added to the test case noting that tcv_TestBody is not updated, so verdicts assigned within preambles and postambles will be treated as if they are part of the test body.

Within the test body, preliminary verdicts shall be used to indicate the result of the test purpose. Each behaviour line within the test body containing a preliminary verdict shall have a label of the form TBXN, where X is one of P, F, I for pass, fail, and inconclusive respectively, and N is a number (with one or more digits) used to distinguish multiple TBPs, TBFs, or TBIs in the same test case.

If an unexpected event occurs corresponding to a test case error, a final inconclusive verdict shall be assigned, and the behaviour line shall have a label ERRN, where N is a number used to distinguish multiple ERRs, and ERR indicates that a test case error has occurred. An example of this is provided in the test step clause.

Table E.8 contains an example test case illustrating these concepts.

Table E.8: Example test case illustrating use of verdicts, labels and tcv_TestBody test case variable

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		+ts_Preambles			
2	TBS	(tcv_TestBody := TRUE)			1
3		L ! Stimulus	cs_Stimulus1		
4		+lt_Response			
5	TBE	(tcv_TestBody := FALSE)		(P)	2
6		+ts_Postambles			
		lt_Response			
7	TBP1	L ? Response	cr_ValidResponse1	(P)	3
8	TBP2	L ? Response	cr_ValidResponse2	(P)	3
9	TBF1	L ? Response	cr_InvalidResponse	(F)	4
10	TBI1	L ? Response	cr_OtherResponse	(I)	5
Detailed comments	1. The behaviour line setting tcv_TestBody to TRUE shall have the label TBS. 2. The behaviour line setting tcv_TestBody to FALSE shall have the label TBE, and can optionally be used to assign a verdict indicating that the test purpose has passed or failed (i.e. if the final behaviour statement in the test body is a tree attachment). 3. The label TBPN is used to indicate that the test purpose has been achieved via the Nth possible valid UE behaviour. 4. The label TBFN is used to indicate that the test purpose has not been achieved, due to the Nth possible failure cause. 5. The label TBIN is used to indicate that the test result is inconclusive for the Nth possible unexpected / unknown event.				

E.3.20.3 Test steps

To promote re-use, test steps shall only assign preliminary verdicts (I) and (F). (P) verdicts shall be managed at the test case level in general, but may be used sparingly within test steps. ETR 141 [37] clause 12.4 recommends that a preliminary pass verdict should be assigned at the leaf of each passing event sequence of the test step. If a test step includes an alternative for unexpected / invalid behaviour, then either a preliminary inconclusive verdict shall be assigned if tcv_TestBody is FALSE, or a preliminary failure verdict shall be assigned if tcv_TestBody is TRUE.

Each behaviour line within the test step containing a preliminary verdict shall have a label of the form TSXN, where X is one of P, F or I for pass, fail, and inconclusive respectively, and N is a number (with one or more digits) used to distinguish multiple TSPs, TSFs, or TSIs in the same test step.

If an unexpected event occurs corresponding to a test case error, a final inconclusive verdict shall be assigned, and the behaviour line shall have a label ERRN, where N is a number used to distinguish multiple ERRs, and ERR indicates that a test case error has occurred.

Table E.9 contains an example test step illustrating these concepts.

Table E.9: Example test step illustrating use of verdicts, labels and tcv_TestBody test case variable

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		[p_Mode = tsc_Mode1]			
2		L ! Stimulus	cs_Stimulus1		
3		+lt_Response			
4		[p_Mode = tsc_Mode2]			
5		L ! Stimulus	cs_Stimulus2		
6		+lt_Response			
7	ERR1	[TRUE]		I	1
		lt_Response			
8		L ? Response	cr_ValidResponse1		2
9		L ? Response	cr_InvalidResponse		
10	TSI1	[tcv_TestBody = FALSE]		(I)	3
11	TSF1	[tcv_TestBody = TRUE]		(F)	4
Detailed comments	1. An invalid value for the parameter p_Mode has been passed to this test step, so a final inconclusive verdict is assigned, with a label indicating that a test case error has occurred. 2. If the expected behaviour occurs, then the test step completes at the leaf node, and the current preliminary verdict is not changed. 3. If unexpected / invalid behaviour occurs, and the current test step is being used as a preamble or postamble (tcv_TestBody = FALSE) then a preliminary inconclusive verdict is assigned. 4. If unexpected / invalid behaviour occurs, and the current test step is being used as part of the test purpose(tcv_TestBody = TRUE) then a preliminary failure verdict is assigned.				

E.3.20.4 Defaults

Each behaviour line within a default behaviour table containing a preliminary verdict shall have a label of the form DFXN, where X is one of F or I for fail, and inconclusive respectively, and N is a number (with one or more digits) used to distinguish multiple DFFs, or DFIs in the same test step.

tcv_TestBody shall be referenced from within default behaviour tables to assign the appropriate verdict when unexpected events occur.

Table E.10 contains an example default behaviour table illustrating these concepts.

TableE.10: Example default behaviour table illustrating use of verdicts, labels and tcv_TestBody test case variable

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		L ? Response	cr_IgnoredResponse		1
2		RETURN			
3	DFI1	L ? OTHERWISE [tcv_TestBody = FALSE]		(I)	2
4	DFF1	L ? OTHERWISE [tcv_TestBody = TRUE]		(F)	3
Detailed comments	1. Valid events that are to be ignored can be included in the default behaviour, but should have no preliminary verdict assigned. 2. If unexpected data is received in the preambles or postambles, a preliminary inconclusive verdict is assigned, and the test case is terminated. 3. If unexpected data is received in the test body, a preliminary failure verdict is assigned, and the test case is terminated.				

See also ETR 141 [37] clauses 11.2, 12.4 and 14.3.

E.3.21 Test suite and test case variables

A default value shall be provided for all test suite and test case variables.

E.3.22 Use of macros is forbidden

The use of macros is forbidden, to support migration to TTCN3.

E.3.23 Support for future Radio Access Technologies

To allow existing test cases to be updated in future to support other radio access technologies, test suites shall make use of a PIXIT parameter px_RAT of type RatType as shown in the following example.

Test Case Name	tc_RAT_Example1				
Nr	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		START t_Guard(300)			
2		[px_RAT = fdd]			
3		PCO ! FDD_PDU	c_FDD_PDU1		FDD specific behaviour
4	TBP1	PCO ? COMMON_PDU	c_COMMON_PDU1	(P)	
5		[px_RAT = tdd]			
6		PCO ! TDD_PDU	c_TDD_PDU1		TDD specific behaviour
7	TBP2	PCO ? COMMON_PDU	c_COMMON_PDU1	(P)	
8		[px_RAT = other_rat]		I	Tests for this RAT not implemented yet
9	TCE1	[TRUE]		I	Unexpected px_RAT value
Detailed Comments					

In general, alternatives should be used to separate behaviour specific for each RAT, and common behaviour should be re-used as much as possible. A final inconclusive verdict shall be used for any alternatives that have not been implemented yet.

Local trees may be used as shown in the following example to improve re-use of common behaviour.

Test Case Name	tc_RAT_Example2				
Nr	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		START t_Guard(300)			
2		+lt_RAT_SpecificPart			
3	TBP1	PCO ? COMMON_PDU	c_COMMON_PDU1	(P)	Common behaviour
		lt_RAT_SpecificPart			
4		[px_RAT = fdd]			
5		PCO ! FDD_PDU	c_FDD_PDU1		FDD specific behaviour
6		[px_RAT = tdd]			
7		PCO ! TDD_PDU	c_TDD_PDU1		TDD specific behaviour
8	TCE1	[TRUE]		(I)	Unexpected px_RAT value
Detailed Comments					

E.3.24 Managing multiple representations of the same information

When the same information is represented using multiple types within the same test suite, it is necessary to manage conversions between the types, and ensure that the information remains consistent across all of the representations.

For example, IMSI is represented as 'SEQUENCE (SIZE (6..15)) OF Digit' in the RRC ASN.1 definitions, as a HEXSTRING for input as a PIXIT parameter, and as an information element defined in TTCN tabular format for MM.

E.3.24.1 Predefined types

Conversion operations are not required to convert the following TTCN predefined types to their counterparts in ASN.1.

a) INTEGER predefined type.

b) BOOLEAN predefined type.

c) BITSTRING predefined type.

d) HEXSTRING predefined type.

e) OCTETSTRING predefined type.

f) OBJECTIDENTIFIER predefined type.

g) R_TYPE predefined type.

h) CharacterString predefined types.

Therefore it is valid to pass a value of type BIT STRING (ASN.1) as a formal parameter of type BITSTRING (TTCN predefined).

E.3.24.2 Simple types

TR 101 666 [27] clause 11.2.1 states:

- 'TTCN is a weakly typed language, in that values of any two types which have the same base type are considered to be type compatible (e.g. for the purposes of performing assignments or parameter passing)'.

When simple types have restrictions, it is the TTCN author's responsibility to ensure that the restrictions are compatible. The TTCN compiler provides some assistance with this, but the extent of the checking is compiler specific.

E.3.24.3 Structured types

For conversion between more complex representations, test suite operations will generally be required. If the mapping is simple enough, it may be possible to perform the conversion using a test step, which takes the common representation as a parameter, and stores the required representation in a test case variable. This may avoid the need for an extra test suite operation.

E.3.24.4 Conversion responsibility

Two design approaches are possible for deciding where the responsibility of conversion lies: Calling party conversion and called party conversion.

The appropriate option should be selected on a case-by-case basis with the following restrictions:

- If one representation of the information is a PIXIT parameter, and this information must be passed to a test step, the called party conversion option shall be used, and the formal parameter to the test step shall always have the same type as the PIXIT parameter.

- If a test step provides multiple alternatives for different radio access technologies, which require different representations of the same information, the called party conversion convention shall be used. In this case a technology independent representation of the information shall be passed as a parameter, and the test step shall perform the conversion to the appropriate type depending on which RAT is being used.

E.3.24.5 Option 1: Calling party conversions

For this approach, each test step provides an interface based on its internal representation. It is the responsibility of the test case / step attaching the test step to perform the conversion before the attachment.

E.3.24.5.1 Advantages

- The number of calls to conversion operations is minimised.
- The complexity of the attached test steps is reduced because fewer conversions are required than for the called party conversion approach.

E.3.24.5.2 Disadvantages

- Different types are used to transfer the same information across the test step interfaces.
- The complexity of the attaching test steps / cases may be increased because conversions are required before attaching a test step.
- The attaching test steps / cases are responsible for ensuring that multiple representations contain consistent information.

E.3.24.6 Option 2: Called party conversions

In this case, the same representation is used wherever the information must be used as a formal parameter value to a test step, and it is the responsibility of the test step to perform any conversions required.

E.3.24.6.1 Advantages

- The complexity in the attaching test case / step is reduced, which will often improve readability.
- The test step interfaces are cleaner, because the same representation is always passed as a formal parameter.
- Internal representations may be hidden within test steps so that calling parties do not need to have any knowledge of them.

E.3.24.6.2 Disadvantages

- Conversion operations may be called more times than necessary, for example if the same test step is attached twice within one test case.

E.3.25 Assignment using constraint

According to TR 101 666 [27], the Right Hand Side (RHS) of an assignment shall not contain any unbound variables. This implies that the constraints, which are appearing the RHS, shall follow the rules :

1 If the field is of TTCN base type (Simple Type definition):

1.1 the value * is not allowed, it has to be '*'B (or'*'H or '*'O) appropriately.

1.2 the value ? is not allowed, it has to be replaced by '?'B (or'?'H or '?'O) appropriately.

2 If the field is of Structure/ASP/PDU type and the value * or ? are not allowed, it shall be replaced by a constraint of appropriate type (Structute/ASP/PDU). This constraint shall have, all the field values defined properly, satisfying these two rules.

3 The above two rules, have to be applied recursively, if a Structure/ASP/PDU embeds another Struct/ASP/PDU.

E.3.26 Guidelines for use of timers when tolerances are applicable

Timed events within the test suite should implement the timer tolerances specified in 3GPP TS 34.108 [3], clause 4.2.3. It is the TTCN author's responsibility to ensure that appropriate tolerance checks and tolerance values are being used.

NOTE: Tolerances are not applicable to guard timers as described in clause E.3.18 of the present document.

E.3.26.1 Specific situations

The present clause provides recommendations for how to implement timers with tolerances for the following situations:

a) The timed event must occur before a given time.

b) The timed event must occur after a given time.

c) The timed event must occur between two given times.

NOTE: A specific case of this situation is when the desired event occurs at a specific time, plus or minus a tolerance.

E.3.26.2 Example situations

The examples below assume:

a) The test case variable tcv_Duration contains the timer duration (in terms of the units used in the timer declaration).

b) The test case variable tcv_Tolerance has been initialised using one of the following assignments (it is the TTCN author's responsibility to select the calculation resulting in the greatest value of tcv_Tolerance. Reference 3GPP TS 34.108 [3], clause 4.2.3):

1) (tcv_Tolerance := tcv_Duration / 10)

2) (tcv_Tolerance := 2 * tcv_TTI + tsc_T_Delta)
Where tcv_TTI contains the applicable TTI (in ms), and tsc_T_Delta is 55 ms.

NOTE: The timer value parameters used when starting the timers in the examples are recommendations only. Other timer value parameter expressions may be used if appropriate.

E.3.26.2.1 Example of situation 1

Test Step Name	ts_TimerSituation1Example				
Purpose	To demonstrate implementation of a timed event that must occur before a given time.				
Nr	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		START t_UpperBound (tcv_Duration + tcv_Tolerance)			1.
2		+lt_TimedEvent			2.
3	TSP1	CANCEL t_UpperBound		(P)	3.
4	TSF1	? TIMEOUT t_UpperBound		(F)	4.
		lt_TimedEvent			
5		[TRUE]			2.
Detailed Comments	1. Start the timer, allowing tcv_Tolerance extra units for the timed event to arrive. 2. The timed event is observed. 3. The timed event occurred before the timeout, so cancel the timer, and assign a preliminary pass verdict. 4. The timer expired before the timed event occurred, so assign a preliminary failure verdict.				

E.3.26.2.2 Example of situation 2

Test Step Name	ts_TimerSituation2Example				
Purpose	To demonstrate implementation of a timed event that must occur after a given time.				
Nr	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		START t_LowerBound (tcv_Duration - tcv_Tolerance)			1.
2		? TIMEOUT t_LowerBound			2.
3		+lt_TimedEvent			3.
4	TSP1	[TRUE]		(P)	3.
5		+lt_TimedEvent			4.
6	TSF1	CANCEL t_LowerBound		(F)	4.
		lt_TimedEvent			
7		[TRUE]			
Detailed Comments	1. Start the timer, allowing tcv_Tolerance extra units for the timed event to arrive. 2. The timeout is observed before the timed event. 3. The timed event is observed, so assign a preliminary pass verdict. 4. The timed event occurred before the timeout, so cancel the timer, and assign a preliminary failure verdict.				

E.3.26.2.3 Example of situation 3

Test Step Name	ts_TimerSituation3Example				
Purpose	To demonstrate implementation of a timed event that must occur between two given times.				
Nr	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		START t_UpperBound (tcv_Duration + tcv_Tolerance), START t_LowerBound (tcv_Duration - tcv_Tolerance)			1.
2		? TIMEOUT t_LowerBound			2.
3		+lt_TimedEvent			3
4	TSP1	CANCEL t_UpperBound		(P)	3.
5	TSF1	? TIMEOUT t_UpperBound		(F)	4.
6		+lt_TimedEvent			5.
7	TSF2	CANCEL t_LowerBound , CANCEL t_UpperBound		(F)	
		lt_TimedEvent			
8		[TRUE]			
Detailed Comments	1. Start the upper and lower bound timers, allowing tcv_Tolerance extra units each side of the expected time for the timed event to arrive. 2. The lower bound timeout is observed before the timed event. 3. The timed event is observed, so cancel the upper bound timer, and a preliminary pass verdict is assigned. 4. The upper bound timer expired before the timed event occurred, so a preliminary failure verdict is assigned. 5. The timed event occurred before the lower bound timer expired, so a preliminary failure verdict is assigned.				

Annex F (normative): MMI Command strings

This annex lists MMI command strings which are transmitted from the TTCN test steps to the SS.

F.1 Outgoing Call

Please initiate an outgoing Conversational call.

Please initiate an outgoing Streaming call.

Please initiate an outgoing Interactive call.

Please initiate an outgoing Background call.

Please initiate an outgoing Subscribed traffic call.

F.2 Configure UE

Please Configure UE for a MO Telephony call.

Please Configure UE for an MT Telephony call.

Please Configure UE for an Emergency call.

Please Enable call refusal on the UE.

Please configure UE to use the following emmergency number.

F.3 PLMN

Please switch the PLMN selection mode of the UE to automatic selection.

Please switch the PLMN selection mode of the UE to manual selection.

Please select the following PLMN manually: <PLMN ID>.

F.4 Power

Please power on the UE.

Please power off the UE.

Please switch on the UE.

Please switch off the UE.

F.5 USIM

Please insert the USIM card, with information give in table <TABLE NUMBER> into the UE.

Please remove the USIM card from the UE.

Please check if the Memory Capacity Exceeded Flag has been set on the USIM simulator.

Please check if the Memory Capacity Exceeded Flag has been **reset** on the USIM simulator.

Please connect the USIM simulator to the UE.

Please check whether the USIM simulator indicates an attempt made by the ME to store the short message in the USIM and returns the status response 'OK' ('90 00').

Please check whether the USIM simulator indicates an attempt made by the ME to store the short message in the USIM and returns the status response 'Memory Problem' ('92 40').

F.6 SMS

Please check that the reception of a received Short Message is indicated.

Please check that the reception of a received Short Message is NOT indicated.

Please check that NO recalled Short Message is displayed.

Please send an SMS COMMAND message containing a request to delete the previously submitted Short Message.

Please send an SMS COMMAND message containing an enquiry about the previously submitted ~~SM~~ **Short Message**.

Please check the length of the received Short Message: <LENGTH> and please check the contents of the received Short Message: <MESSAGE>.

Please reply to the Short Message of length: <LENGTH> and of the contents: <MESSAGE>.

Please check the contents of the received CBS Message: <MESSAGE>.

F.7 Autocalling

Please initiate an autocalling call with the number: <NUMBER>.

Please initiate an autocalling call with a number that will be put in the blacklisted list. The following number shall not be used: <NUMBER>.

Please reset the autocalling list of blacklisted numbers.

F.8 Miscellaneous

Please check that the DTCH is trough connected by generating a noise.

The guard timer has run out. Please take appropriate measures.

Read the data status of UE.

Please check that the DTMF tone indication has been generated.

Please initiate a non call related supplementary service, which is supported by the UE.

Annex G (informative): Recommendation of an unique ICS/IXIT electronic exchange format

With standardization of ICS/IXIT file format, same test suite parameter (TSP) files can be used across different System Simulators. The ICS/PIXIT will be simple ASCII text files. The assumption is that the test uite parameters are of simple type definitions only and do not include structured types (clause E.3.14).

G.1 Syntax

The proposed format of the ICS/IXIT file is as follows:

[<Parameter Name> <Parameter Type> <Value>] [<#Comment>]

- At the most one TSP value can be defined in a line.
- The comment starts with # and ends with new line.
- [..] represent OPTIONAL field(s).
- **<..>** represent MANDATORY field(s).
- Fields will be separated by one or more space characters.

The syntax for different Parameter Types will be as follows:

- **INTEGER**

```
<Parameter Name>    INTEGER        <Integer Value>
```

- **BOOLEAN**

```
<Parameter Name>    BOOLEAN        <Value>
```

NOTE 1: Here Value will be either 'TRUE' or 'FALSE'.

- **BITSTRING**

```
<Parameter Name>    BITSTRING      <Value>
```

- **HEXSTRING**

```
<Parameter Name>    HEXSTRING      <Value>
```

- **OCTETSTRING**

```
<Parameter Name>    OCTETSTRING    <Value>
```

- **ENUMERATED**

```
<Parameter Name>    ENUMERATED     <Integer Value>
```

- **IA5String**

```
<Parameter Name>    IA5String      "<Value>"
```

NOTE 2: Here Value will be string and is mandatory to put the actual value in double quotes.

G.2 Examples

This clause gives an example of ICS/IXIT file format.

```
# TSP file version 1.0.0

px_CS          BOOLEAN       TRUE            # TRUE if Circuit Switched is applicable

px_PTMSI_Def   OCTETSTRING   12345678         #Default PTMSI

px_RAT         ENUMERATED    0                #px_RAT is of Type RatType and is of Type of ENUMERATED
                                             {fdd(0), tdd(1)}.

px_Region      IA5String     "Europe"         #px_Region is of Type Region and is of Type IA5String
("Europe", Japan").

px_PriScrmCodeA              INTEGER         100    #px_PriScrmCodeA is of Type PrimaryScramblingCode
and is of Type
                                             INTEGER (0..511).

px_SRNC_Id     BITSTRING     000000000001    #px_SRNC_Id is of Type SRNC_Identity and is of Type BIT
STRING
                                             (SIZE(12)).

px_IMSI_Def    HEXSTRING     001010123456063    #Default IMSI
```

Annex H (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
12/2002	TP-18	TP-020301	-		Approved as v3.0.0	-	3.0.0

History

Document history		
V3.0.0	December 2002	Publication