

ETSI TS 102 545-3 V1.2.1 (2009-02)

Technical Specification

Broadband Radio Access Networks (BRAN); HiperMAN; Conformance Testing for WiMAX/HiperMAN 1.3.1; Part 3: Abstract Test Suite (ATS)



Reference

RTS/BRAN-004T008-3-R1

Keywords

ATS, broadband, DLC, FWA, HiperMAN,
point to multipoint, radio, testing**ETSI**

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

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

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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

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

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

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

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

Copyright Notification

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

© European Telecommunications Standards Institute 2009.

© WIMAX Forum 2009.

All rights reserved.

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

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

Contents

Intellectual Property Rights	6
Foreword.....	6
1 Scope	7
2 References	7
2.1 Normative references	7
2.2 Informative references.....	8
3 Definitions and abbreviations.....	8
3.1 Definitions.....	8
3.2 Abbreviations	8
4 Abstract Test Method (ATM).....	9
4.1 Test architecture	9
4.1.1 Test method	9
4.1.1.1 What is notional upper tester?.....	9
4.1.2 Test machine operational parameters.....	11
4.1.3 Test machine configuration.....	12
4.1.3.1 Presentation.....	12
4.1.3.2 Test suite TTCN-3 development concept.....	13
4.1.3.3 Test configurations for SS/MS	14
4.1.3.4 Test configurations for BS	16
4.1.4 Re-use of existing test specifications	17
4.1.5 Test architecture.....	18
4.1.5.1 Common test architecture	18
4.1.5.1.1 Single Test Component	18
4.1.5.1.2 Multiple Test Components	19
4.2 Description of the ports and their associated primitives.....	20
4.2.1 The MacMessagePort type.....	20
4.2.1.1 Description	20
4.2.1.2 Primitives of the MacMsg port.....	20
4.2.2 The MacBcMessagePort type	21
4.2.2.1 Description	21
4.2.2.2 Primitives of the BcMacMsg port	21
4.2.3 The MacPduPort type	22
4.2.3.1 Description	22
4.2.3.2 Primitives of the MacPdu port	22
4.2.4 The PhyPort type	24
4.2.4.1 Description.....	24
4.2.4.2 Primitives of the Phy Port	24
4.2.5 The TaPort type	25
4.2.5.1 Description	25
4.2.5.2 Primitives of the TaPort	25
4.3 Port mapping rules.....	26
4.4 PDU sending/receiving rules	26
5 Untestable Test Purposes (TP)	26
6 ATS conventions	27
6.1 Testing conventions.....	27
6.1.1 Testing States.....	27
6.1.2 HiperMAN default values: Reception and transmission at ATS level.....	27
6.1.3 Templates.....	27
6.1.4 Functions	27
6.2 Naming conventions.....	28
6.2.1 General guidelines	28
6.2.2 Test Case (TC) identifier	29
6.3 Service Flow parameter support.....	30

6.3.1	CsSpecification support	30
6.3.2	DataDeliveryServiceType.....	31
6.4	Dispatching of test cases over TTCN modules.....	31
7	External functions	32
8	Test strategies	32
8.1	HARQ testing.....	33
8.1.1	HarqAckNack PHY port message	33
8.1.2	HARQ TA Primitives	33
8.1.3	HARQ Broadcast Message Filter.....	33
8.1.4	HARQ external Functions	33
8.2	Fast Feedback testing	34
8.2.1	CQICH PHY port message	34
8.2.2	CQICH TA Primitives	34
8.2.3	CQICH Broadcast Message Filter	34
8.2.4	CQICH external Functions	34
8.3	Handover testing	35
8.3.1	Testing Serving BS	35
8.3.2	Testing Target BS	36
8.3.2	Testing MS with Handover or MBS procedure	37
8.4	Processing of EAP messages.....	37
8.5	Handling MAC PDU with Fragmentation, Packing or Concatenation.....	38
8.5.1	MS (IUT) Processing DL MAC PDUs with FRAG Subheader.....	38
8.5.2	MS (IUT) Processing DL MAC PDU with PACK Subheaders.....	38
8.5.3	MS (IUT) Processing DL MAC PDU with PACK and FRAG Subheaders.....	39
8.5.4	MS (IUT) Processing concatenated DL MAC PDUs	39
8.5.5	MS (IUT) Generating UL MAC PDU with FRAG Subheader	39
8.5.6	MS (IUT) Generating UL MAC PDU with PACK Subheaders	39
8.5.7	MS (IUT) Generating concatenated UL MAC PDUs	40
8.5.8	BS (IUT) processing UL MAC PDUs with FRAG Subheader	40
8.5.9	BS (IUT) processing concatenated UL MAC PDUs.....	40
8.5.10	BS (IUT) generating DL MAC PDU with FRAG Subheader.....	41
8.5.11	BS (IUT) generating DL MAC PDU with PACK Subheaders	41
8.6	Service Flows management for testing BS.....	41
8.6.1	Introduction.....	41
8.6.2	Handling of service flows in the default.	42
8.6.3	Global variables used for the service flow management.....	43
8.6.4	How to invoke a required SF that is not available	44
Annex A (normative):	WiMAX/HiperMAN 1.3.1 Abstract Test Suite (ATS).....	45
A.1	The TTCN-3 Module.....	45
Annex B (normative):	WiMAX/HiperMAN 1.3.1 Partial PIXIT proforma for IUT BS.....	46
Annex C (normative):	WiMAX/HiperMAN 1.3.1 Partial PIXIT proforma for IUT MS.....	47
Annex D (normative):	WiMAX/HiperMAN 1.3.1 PCTR Proforma for IUT BS.....	48
D.1	Identification summary.....	48
D.1.1	Protocol conformance test report.....	48
D.1.2	IUT identification	48
D.1.3	Testing environment.....	48
D.1.4	Limits and reservation	49
D.1.5	Comments.....	49
D.2	IUT Conformance status	49
D.3	Static conformance summary	49
D.4	Dynamic conformance summary.....	50
D.5	Static conformance review report.....	50
D.6	Test campaign report.....	51

D.7	Observations.....	55
Annex E (normative):	WiMAX/HiperMAN 1.3.1 PCTR Proforma for IUT MS.....	56
E.1	Identification summary.....	56
E.1.1	Protocol conformance test report.....	56
E.1.2	IUT identification.....	56
E.1.3	Testing environment.....	56
E.1.4	Limits and reservation.....	57
E.1.5	Comments.....	57
E.2	IUT Conformance status	57
E.3	Static conformance summary	57
E.4	Dynamic conformance summary.....	58
E.5	Static conformance review report.....	58
E.6	Test campaign report.....	59
E.7	Observations.....	64
Annex F (normative):	HTML documentation.....	65
Annex G (informative):	Bibliography.....	66
History		67

Intellectual Property Rights

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

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

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Broadband Radio Access Networks (BRAN).

The present document was developed on the basis of the Abstract Test Suite (ATS) specification for HiperMAN systems that was in the advanced stage of development when the work was reoriented to produce joint HiperMAN/WiMAX specifications.

The present document is part 3 of a multi-part deliverable covering Broadband Radio Access Networks (BRAN); HiperMAN; Conformance Testing for WiMAX/HiperMAN 1.3.1, as identified below:

- Part 1: "Protocol Implementation Conformance Statement (PICS) proforma";
- Part 2: "Test Suite Structure and Test Purposes (TSS&TP)";
- Part 3: "Abstract Test Suite (ATS)".**

1 Scope

The present document contains the Abstract Test Suite (ATS) to test BRAN HiperMAN1.3.1/WiMAX systems for conformance.

The objective of the present document is to provide a basis for conformance tests for BRAN HiperMAN/WiMAX equipment giving a high probability of air interface inter-operability between different manufacturer's BRAN HiperMAN/WiMAX equipment.

The ISO standard for the methodology of conformance testing (ISO/IEC 9646-1 [5] and ISO/IEC 9646-2 [6]) as well as the ETSI rules for conformance testing (ETS 300 406 [4]) are used as a basis for the test methodology.

Annex A provides the Tree and Tabular Combined Notation (TTCN) part of the ATS.

Annex B provides the Partial Protocol Implementation Extra Information for Testing (PIXIT) Proforma of the SS side ATS.

Annex C provides the Protocol Conformance Test Report (PCTR) Proforma of the SS side ATS.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
 - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
 - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

- [1] ETSI TS 102 178 (V1.2.1): "Broadband Radio Access Networks (BRAN); HiperMAN; Data Link Control (DLC) layer".
- [2] IEEE 802.16-2004: "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems".

- [3] IEEE 802.16e-2005: "IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1".
- [4] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".
- [5] ISO/IEC 9646-1/ITU-T Recommendation X.290: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".
- [6] ISO/IEC 9646-2/ITU-T Recommendation X.291: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification".
- [7] ISO/IEC 9646-6: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 6: Protocol profile test specification".
- [8] ISO/IEC 9646-7: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".
- [9] ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".
- [10] IEEE P802.16-2004/Cor1/D3: "Corrigendum to IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ISO/IEC 9646-7 [8], TS 102 178 [1], IEEE 802.16-2004 [2] and IEEE 802.16e-2005 [3] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 102 178 [1], ISO/IEC 9646-1 [5], ISO/IEC 9646-6 [7], ISO/IEC 9646-7 [8], IEEE 802.16-2004 [2], IEEE 802.16e-2005 [3] and the following apply:

ATS	Abstract Test Suite
BS	Base Station
BW	BandWidth
CID	Connection Identifier
CS	Convergence Sublayer
FDD	Frequency Division Duplexing
IUT	Implementation Under Test
MAC	Medium Access Control Layer
MS	Mobile Station
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PHY	Physical Layer

PIXIT	Partial Protocol Implementation Extra Information for Testing
PMP	Point-to-MultiPoint
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
REQ	REQuest
RNG	RaNGing
RSP	ReSPonse
RTG	Receive/Transmit Transition Gap
SS	Subscriber Station
SUT	System Under Test
TC	Test Case
TLV	Type, Length, Value
TP	Test Purposes
TTCN	Test and Test Control Notation
TTG	Transmit/Receive Transition Gap

4 Abstract Test Method (ATM)

This clause describes the ATM used to test the HiperMAN DLC layer at the BS side and at the SS side.

4.1 Test architecture

4.1.1 Test method

The test method chosen is the remote test method with notional upper tester. Remote test method means that the test tool (the test machine + the executable test suite) shall behave as a BS when the IUT is an SS and shall behave as an SS when the IUT is a BS. Notional upper tester means that it is possible to trigger and to force the IUT to execute predefined actions.

Example: Adding a new service flow with defined parameters, sending data over a known service flow, etc.

This could be done by a specific and proprietary application layer inside the IUT or by other procedures clearly described by the IUT's manufacturer (PIXIT question). As the exchange between the test system and the IUT is the air interface, the PHY layer of the test machine shall be totally conformant with the corresponding PHY layer specification to use the remote test method.

4.1.1.1 What is notional upper tester?

Usually the IUT is not only a plane containing Convergence, MAC and PHY layers, but a real product to be marketed after testing, and therefore the IUT contains also application software to accomplish the purpose of the final product. In that case, the application inside the IUT could be commanded to generate events in direction of the transmission sub layers that shall be used by the testing software as expected IUT's actions. The application layer is the Upper tester as defined in ISO 9646. It is also a notional upper tester, because the test designer cannot determine all of the possible applications that are only market driven.

Considering the explanation of the former paragraph, in terms of source code writing, requesting a notional upper tester action is the combination of the call of an external function and a PIXIT parameter. The external function asks the test laboratory operator to execute the procedure described in the PIXIT parameter. If the action is possible to obtain the external function succeeds, otherwise the test execution becomes inconclusive. The PIXIT parameter is a "how to" question, for which the product manufacturer has to explain the procedure to be used in the IUT to obtain the required action.

Figures 1 to 4 show some examples of possible notional upper tester.

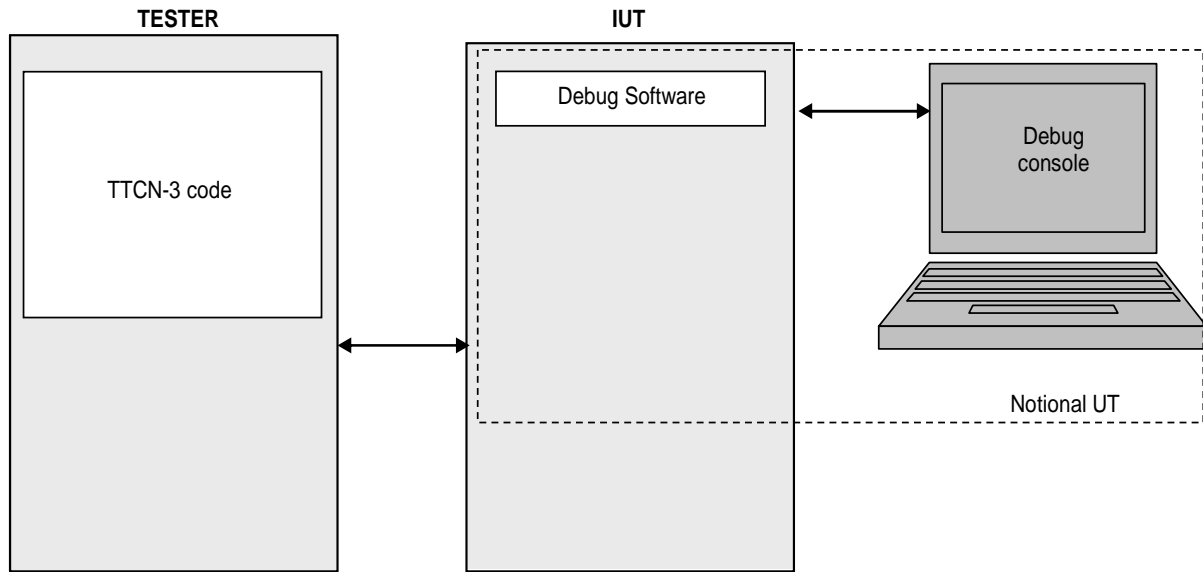


Figure 1: Debug notional upper tester

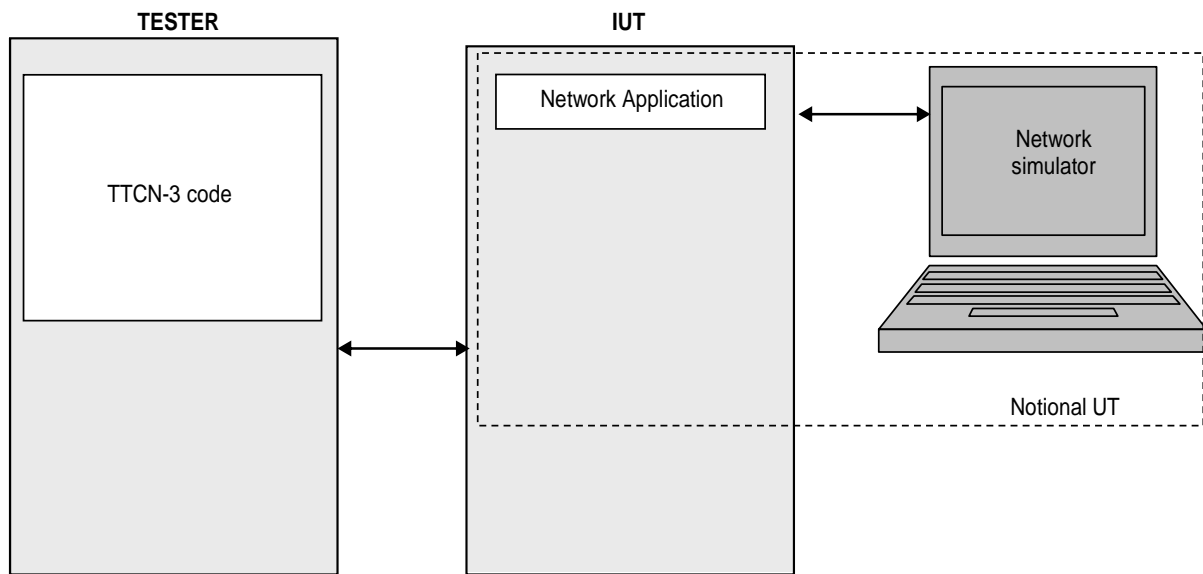


Figure 2: Network driven notional upper tester

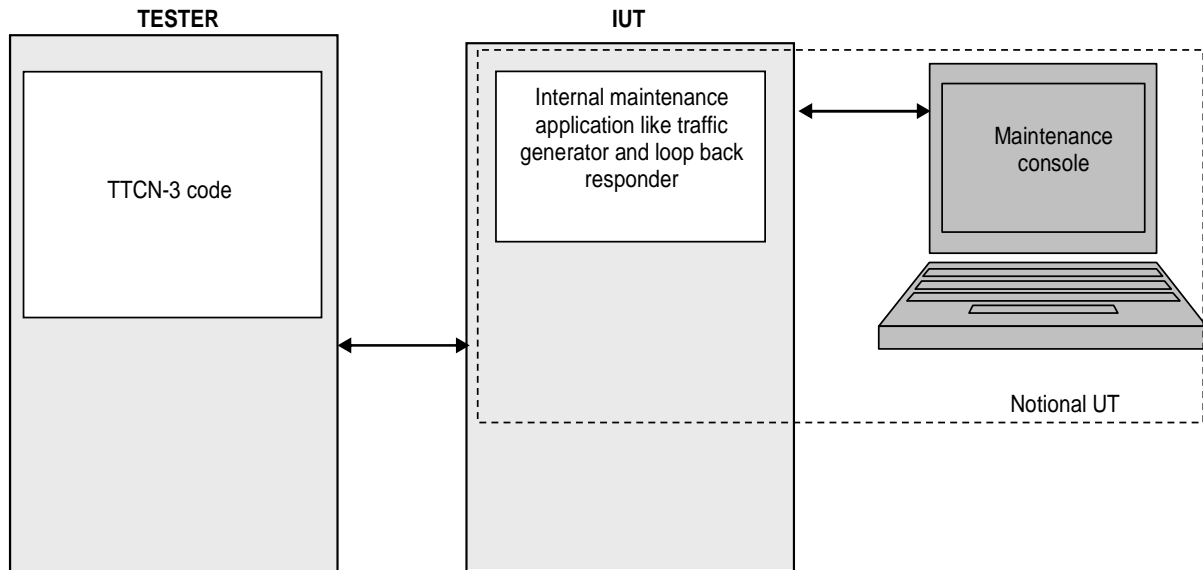


Figure 3: Maintenance application notional upper tester

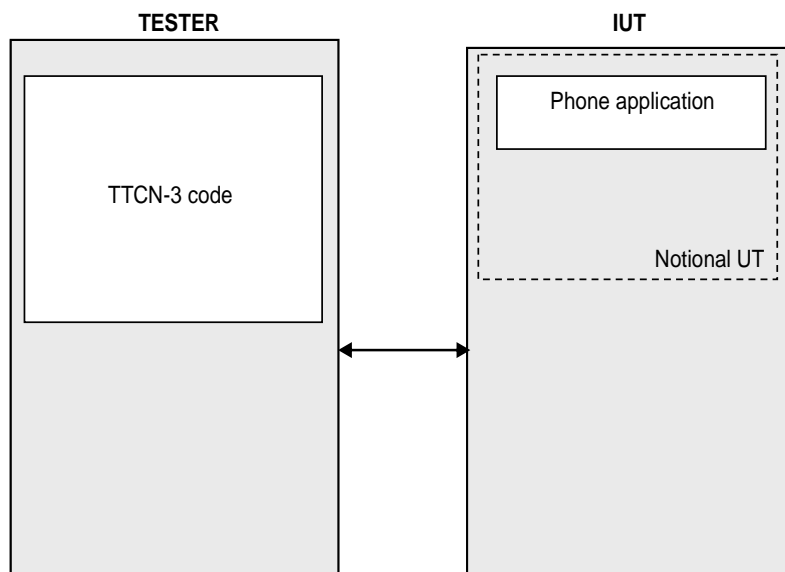


Figure 4: Phone application notional upper tester

4.1.2 Test machine operational parameters

The test machine operational parameters such as frequency, channels, sub channels, power level, etc., could be initialized by static and/or dynamic method.

The static method could be:

- 1) operational parameters included in the firmware or ROM;
- 2) operational parameters included in a configuration file executed at power up;
- 3) other static technique;
- 4) no default or static operational parameters setting.

The dynamic method could be:

- 1) before the test cases execution at the beginning of the test campaign and valid for a list of TCs;
- 2) during the test case execution at the beginning of the test case itself;
- 3) everywhere during test case execution.

The possibility to acquire and to set all of the operational parameters during the test case execution is a main key to cover all of the requirements to be tested by the TTCN-3 test code.

Considering all of the techniques exposed above, it is possible that the configuration of the operational parameters is done either before the beginning of the TTCN-3 environment or during the initialization of the TTCN-3 environment or during the preamble of a test case. The recommended method is the initialization during preamble of the test case.

Another important problem is the reconfiguration on the fly of some operational parameters. To solve this problem, it is recommended that the test case itself shall be able to start and stop the PHY layer and all of its environments during test case execution.

4.1.3 Test machine configuration

4.1.3.1 Presentation

There are six test machine configurations to allow the complete testing of the required functionalities of the specification.

The test machine configurations are:

- 1) test machine simulates a BS with OFDM PHY (IUT is a SS with OFDM PHY);
- 2) test machine simulates a BS with OFDMA PHY (IUT is a SS with OFDMA PHY);
- 3) test machine simulates a SS with OFDM PHY (IUT is a BS with OFDM PHY);
- 4) test machine simulates a SS with OFDMA PHY (IUT is a BS with OFDMA PHY);
- 5) test machine simulates two BS, each of them with OFDM PHY (IUT is an MS with OFDM PHY), This configuration is used for handover and mobility testing;
- 6) test machine simulates two BS, each of them with OFDMA PHY (IUT is an MS with OFDMA PHY), This configuration is used for handover and mobility testing;
- 7) test machine simulates one SS and one BS, each of them with OFDM PHY (IUT is a BS with OFDM PHY);
- 8) test machine simulates one SS and one BS, each of them with OFDMA PHY (IUT is a BS with OFDMA PHY).

NOTE: For a very small number of specification requirements, it may be useful to have a configuration with three simulated BS. This increases the number of test machine configuration by two (one for OFDM and one for OFDMA). Considering the effort of hardware and software development and the corresponding costs, implementation of these configurations should be investigated very carefully, and interoperability testing may be more suitable than conformance testing.

The configurations 1, 2, 3 and 4 can be covered by a single testing approach. The configurations 5, 6, 7 and 8 shall be covered by a concurrent testing approach (it is necessary to monitor and synchronize the two simulated BS test code to obtain a consistent behaviour and a consistent test verdict). The use of the distributed testing possibilities of TTCN-3 is recommended for the physical architecture of the test machine for the test configurations 5 and 6.

The number of physical test machines to cover the eight test configurations could comprise between one and eight depending of the level of flexibility and parameterization of the hardware design made by the test tool manufacturer. A physical test machine could also be constituted by a number greater than one of real hardware machine (example: intelligent PHY plane connected to one or more PC executing the TTCN-3 code).

For similar reasons the number of test suites could comprise between 1 and 8 depending of the level of parameterization, by use of PICS and PIXIT items, used to design the TTCN-3 code. The conditional compilation may be used to have only one source code and many generated test suite. In terms of performance, it is preferable to have static conditional code generation to shorten the length of the test suite and improve the time execution rather than to have dynamic conditional alternatives controlled by PICS or PIXIT items. In terms of readability and maintenance of the test code it is preferable to have a one to one mapping between the test code and the test machine configuration. The use of libraries, packages and other recent technique of source code management are recommended.

4.1.3.2 Test suite TTCN-3 development concept

The possible Test suite TTCN-3 development concepts are shown in figures 5, 6 and 7.

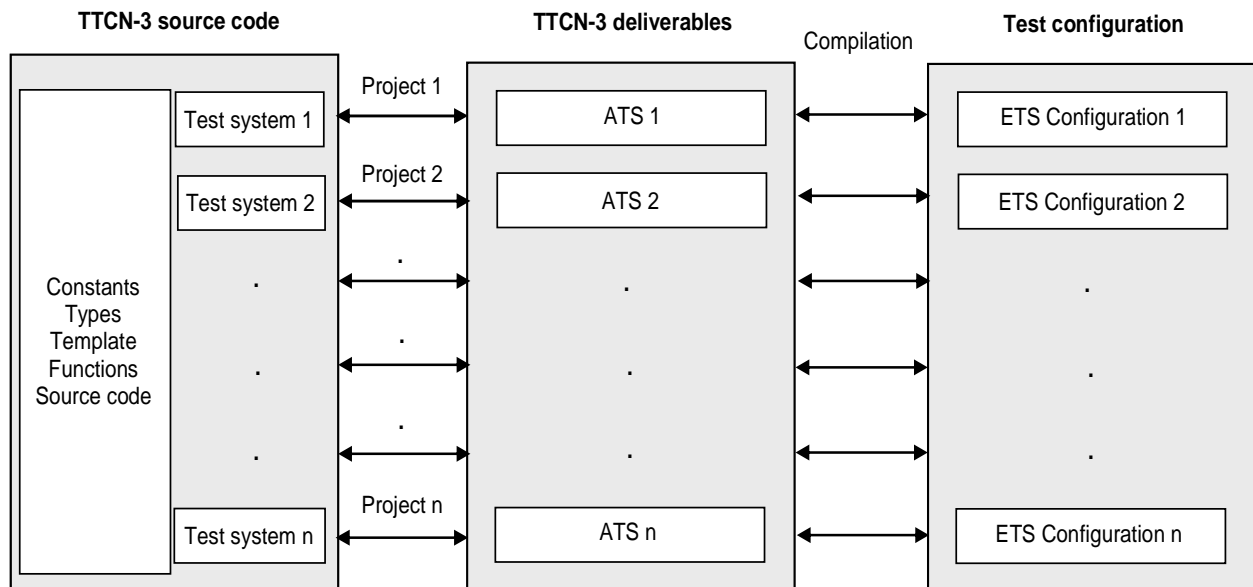


Figure 5: TTCN-3 development concept 1

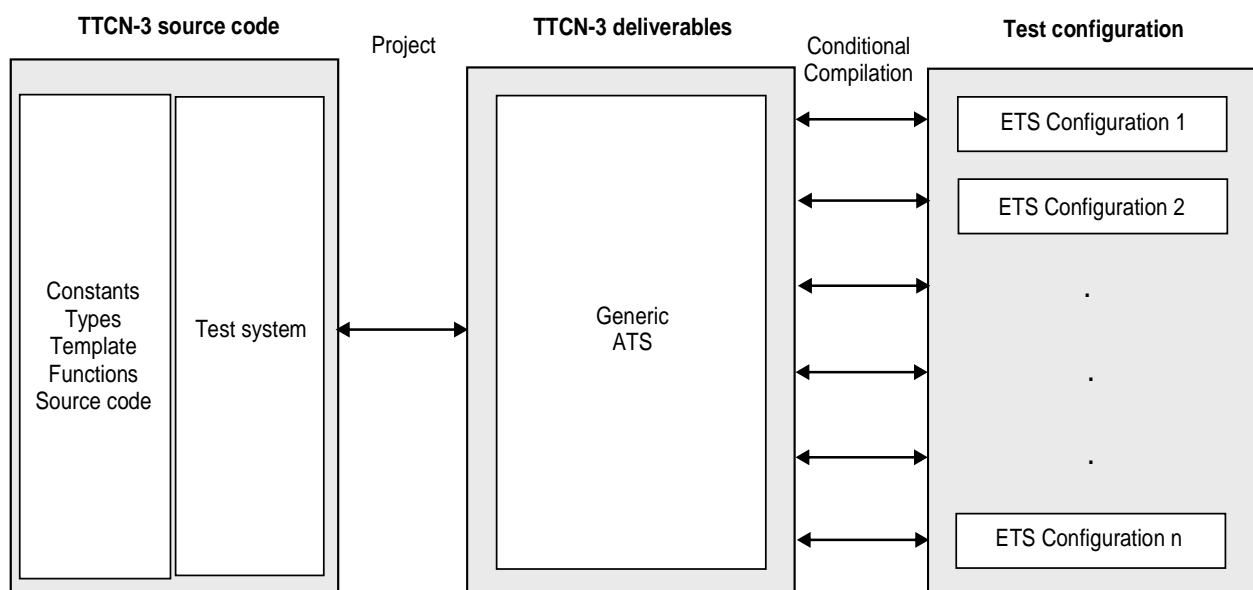


Figure 6: TTCN-3 development concept 2

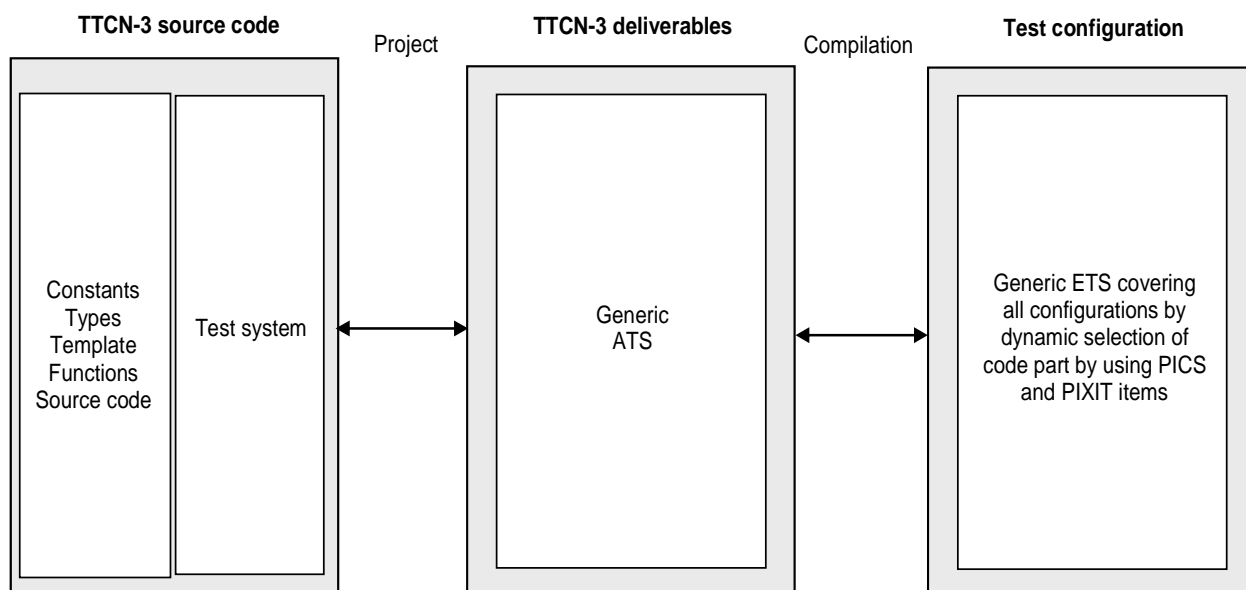


Figure 7: TTCN-3 development concept 3

For all of the three TTCN-3 development concepts, the Test Configuration shall be done dynamically based on PIXIT and PICS parameters.

According to a consensus between the TTCN-3 development team and the Test tool manufacturers, the TTCN-3 development concepts 1 showed above will be used for the real development.

4.1.3.3 Test configurations for SS/MS

There are four normal configurations and two optional configurations for SS/MS testing.

The configuration 1 is defined and used for functionality that requires only interaction between the tested OFDM SS/MS and one OFDM BS. This configuration is shown in figure 8.



Figure 8: Configuration 1 for SS/MS

The configuration 2 is defined and used for functionality that requires only interaction between the tested OFDMA SS/MS and one OFDMA BS. This configuration is shown in figure 9.



Figure 9: Configuration 2 for SS/MS

The configuration 5 is defined and used when an OFDM SS/MS has to interact with two OFDM BSs. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 10.

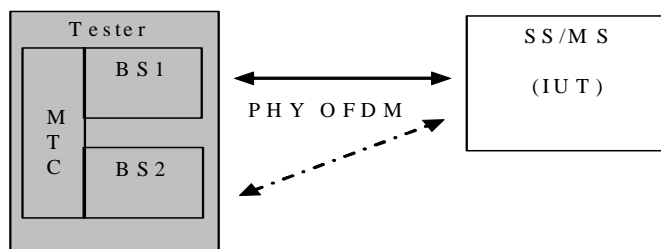


Figure 10: Configuration 5 for SS/MS

The configuration 6 is defined and used when an OFDMA SS/MS has to interact with two OFDMA BSs. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 11.

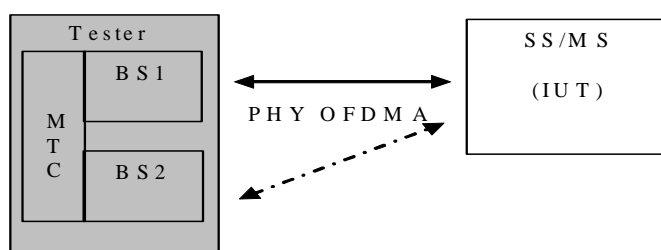


Figure 11: Configuration 6 for SS/MS

The optional configuration Opt1 is defined and used when an OFDM SS/MS has to interact with more than two OFDM BSs. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 12.

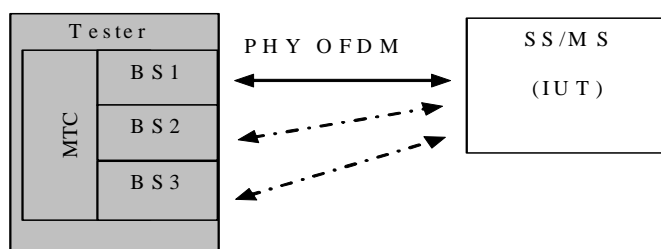


Figure 12: Configuration Opt1 for SS/MS

The optional configuration Opt2 is defined and used when an OFDMA SS/MS has to interact with more than two OFDMA BSs. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 13.

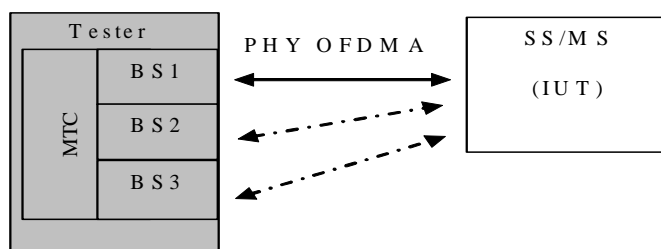


Figure 13: Configuration Opt2 for SS/MS

4.1.3.4 Test configurations for BS

There are two normal configurations and four optional configurations for BS testing.

The configuration 3 is defined and used for functionality that requires only interaction between the tested OFDM BS and one OFDM MS/SS. This configuration is shown in figure 14.

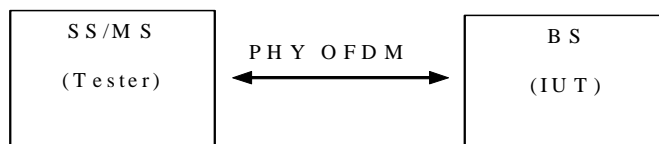


Figure 14: Configuration 3 for BS

The configuration 4 is defined and used for functionality that requires only interaction between the tested OFDMA BS and one OFDMA MS/SS. This configuration is shown in figure 15.



Figure 15: Configuration 4 for BS

The optional configuration Opt3 is defined and used when an OFDM BS has to interact with two OFDM MS/SS. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 16.

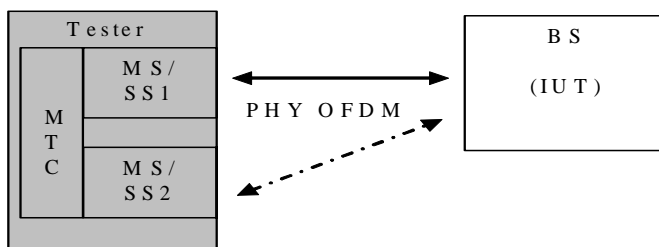


Figure 16: Configuration Opt3 for BS

The optional configuration Opt4 is defined and used when an OFDMA BS has to interact with two OFDMA MS/SS. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 17.

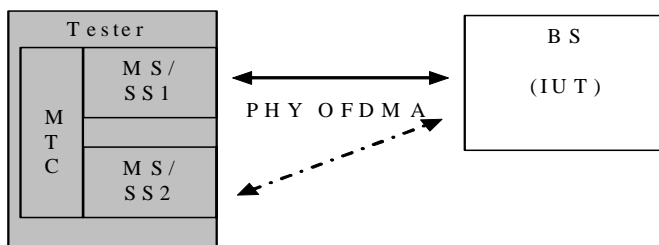


Figure 17: Configuration Opt4 for BS

The optional configuration Opt5 is defined and used when an OFDM BS has to interact with one OFDM BS and one OFDM MS/SS. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 18.

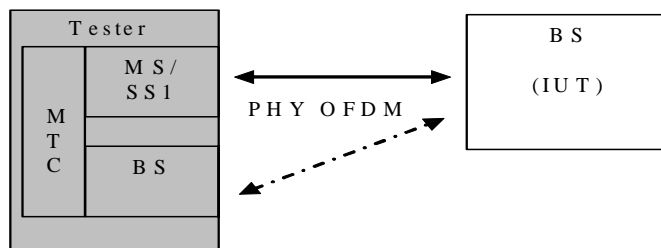


Figure 18: Configuration Opt5 for BS

The optional configuration Opt6 is defined and used when an OFDMA BS has to interact with one OFDMA BS and one OFDMA MS/SS. The concurrent TTCN-3 facilities are used in this configuration. This configuration is shown in figure 19.

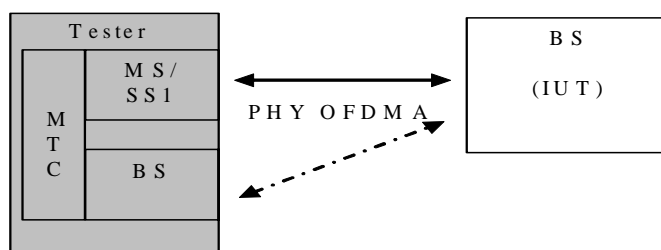


Figure 19: Configuration Opt6 for BS

4.1.4 Re-use of existing test specifications

Due to existing development for IEEE 802.16-2004 [2] and ETSI HiperMAN, it is preferable if not essential to reuse as much of the existing test specifications.

Nevertheless, considering the preceding considerations such as hardware configuration and test configuration, it appears that the existing TTCN code may be only partially re-usable. For TTCN-3 code, the constants, types, templates and internal/external functions could be re-used and extended, but the other parts are certainly not in line with the new hardware and software configuration.

Considering that, there are two possibilities:

- 1) Starting from scratch with small re-use of existing test specifications.
- 2) Defining a test architecture that included the architecture defined for IEEE 802.16-2004 [2] and ETSI HiperMAN as near as possible and adding small changes in the actual TTCN-3 code.

According to a consensus between the TTCN-3 development team and the Test tool manufacturers, the second possibility showed above will be used for the real development.

4.1.5 Test architecture

4.1.5.1 Common test architecture

4.1.5.1.1 Single Test Component

Figures 20 and 21 describe the DLC BS/SS Test Configuration for testing the DLC layer of a product implementing the HiperMAN base standard. More information for these architectures is provided below. Figure 20 is related to single testing. Figure 21 is related to concurrent testing.

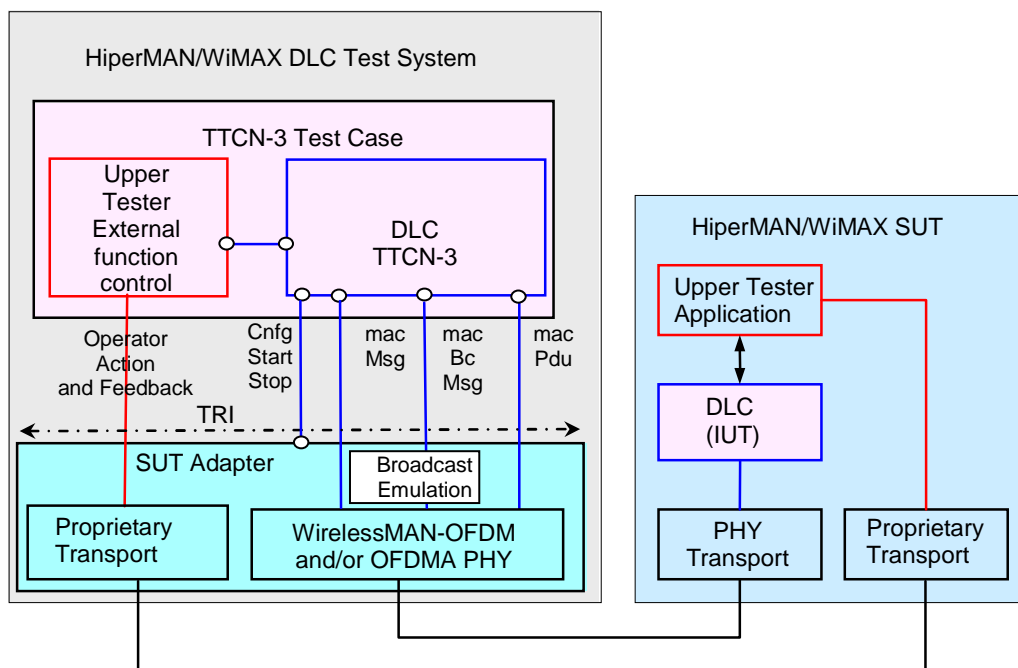


Figure 20: Single DLC BS/SS Test Configuration

DLC TTCN-3 uses macMsg port to send and receive MAC management messages that belong to the Initial Ranging, Basic, and Primary connection. Final verdicts are set on the receive statements (see details on the macMsg port type in clause 4.2.1).

DLC TTCN-3 uses macBcMsg port to receive MAC management messages that belong to the Broadcast connection. Final verdicts are set on the return status of the receive functions (see details on the macMsg port type in clause 4.2.2).

DLC TTCN-3 uses macPdu port to send and receive MAC PDUs. Final verdicts are set on the receive statements (see details on the macMsg port type in clause 4.2.3).

The broadcast emulation handles the sending and reception of the broadcast messages. Broadcast messages from the IUT (BS) are received over the MacBcMsg type port. The setting of the messages contained in the broadcast connection and sent to the IUT (MS) is achieved by using dedicated external functions.

DLC TTCN-3 uses the Phy port to receive information like Ranging Code or Radio Signal measurements, which are not transmitted over MAC PDU but shall be extracted or calculated by the SUT adapter. (See details on the Phy port type in clause 4.2.4).

DLC TTCN-3 controls via external functions the Upper Tester Application. Upper Tester Application allows triggering the IUT. Final verdicts are set on the return status of the external functions.

4.1.5.1.2 Multiple Test Components

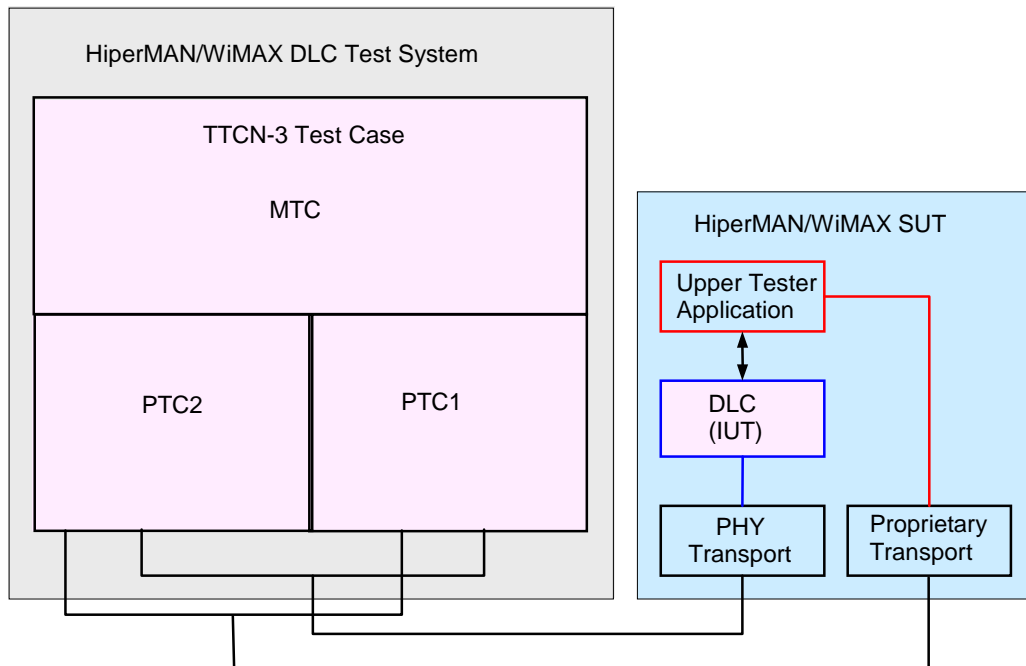


Figure 21: Concurrent DLC BS/SS Test Configuration (logical representation)

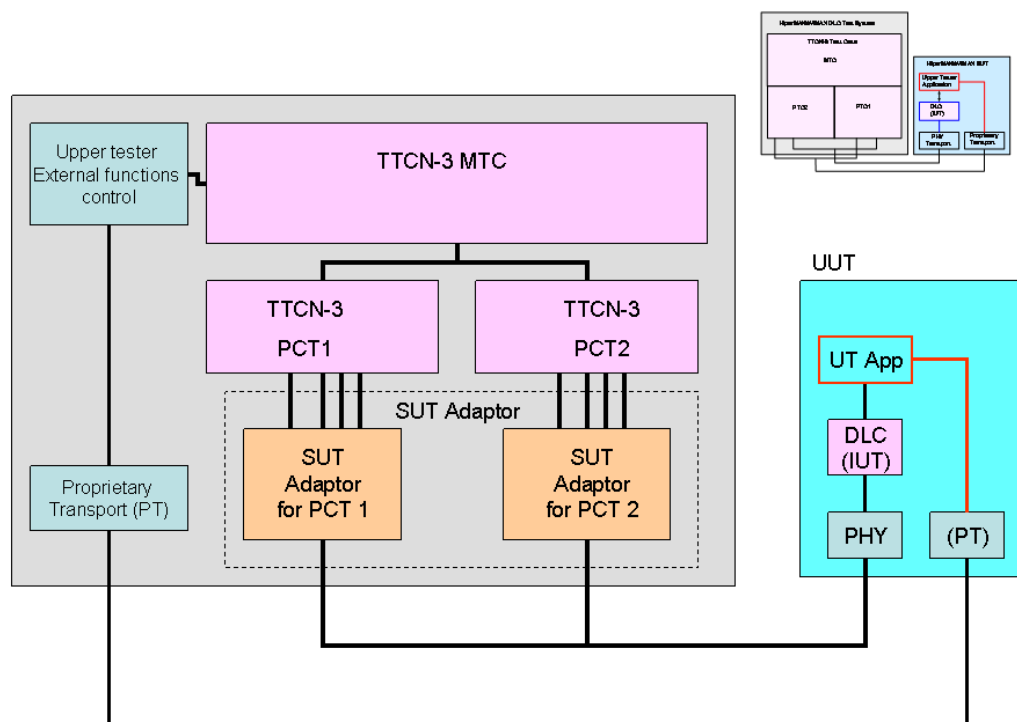


Figure 22: Concurrent DLC BS/SS Test Configuration (physical representation)

The concurrent DLC BS/SS Test Configuration provides 3 test components:

- MTC: Master test component triggers and synchronizes the parallel test components.
- PTC1: Parallel test component 1. Identical to single testing plane to simulate one BS or MS/SS test case part.
- PTC2: Parallel test component 2. Identical to single testing plane to simulate another BS or MS/SS test case part.

NOTE: The number of parallel test components could be extended by adding the corresponding number of single testing plane to perform the required configuration.

All of the parallel test components shall have an identical PHY layer (OFDM or OFDMA).

4.2 Description of the ports and their associated primitives

The test components (either MTC in mono-component configuration or PTCs in multi-components configurations) uses 4 different ports to communicate with the SUT Adapter (see the figures 20 and 21):

- 1) One MacMsg type port.
- 2) One MacBcMsg type port.
- 3) One MacPdu type port.
- 4) One Phy type port.
- 5) One TA type port.

4.2.1 The MacMessagePort type.

4.2.1.1 Description

This port is used to send and receive MAC management messages.

MAC management messages could be fragmented or MAC PDU could contain several MAC management messages (Packed). AS it is difficult to specify defragmentation or unpacking procedures in TTCN, it was decided to create a dedicated port to exchange complete MAC management messages, extracted from the MAC PDU(s) and re-constructed as necessary.

In addition to the MAC messages, several other pieces of information shall be transmitted in the MsgInd primitive:

- 1) Some fields of the generic Header of the MAC PDU that contained the MAC management message.
- 2) A "rawdata" field that actually contains the complete Mac management message (From Message Type until the end) as received from the IUT and before any manipulation of the message by the SUT adapter (i.e. TLV reordering and addition of the default TLVs not sent by the IUT, see clause 4.2.1).
- 3) Any other PHY information related to the receipt of the MAC management message and that can be required in some test cases (e.g. frame number).

4.2.1.2 Primitives of the MacMsg port

Two primitives of type MacMsgPrimitives are currently defined:

- 1) The MsgReq type primitive - to send MAC management messages to the IUT.
- 2) The MsgInd type primitive - to receive MAC management messages from the IUT.

Table 1: Fields of the MsgReq type primitive

Field name	description
cid	Contains the value of the CID field (16 bits) transmitted in the Header of the MAC PDU that contained the MAC management message.
msgInOut	The MAC management message sent on a Basic or Primary connection. This field is a union type of all MAC management message types to be sent or received on the basic and the primary connections.

Table 2: Fields of the MsgInd type primitive

Field name	description
ec	Contains the value of the EC field (1 bit) transmitted in the Header of the MAC PDU that contained the MAC management message.
GenericOrBandwidth	This structured type contains the fields (16 bits) transmitted in the Header of the MAC PDU that contained the MAC management message, which are part of either a generic header or a BandWidth Request header. For the scope of a MAC PDU containing a MAC management message, only the generic header is used, so that this structure type contains the following fields: Type (65 bits), ESF, CI, EKS, Reserved and Length (11 bits).
cid	Contains the value of the CID field (16 bits) transmitted in the Header of the MAC PDU that contained the MAC management message.
msgInOut	The MAC management message received on a Basic or Primary connection. This field is a union type of all MAC management message types to be sent or received on the basic and the primary connections.
rawData	This field contains all bytes of the whole MAC management message that were received by the SUT adapter from the IUT.
phyParams	This field contains the 4 following pieces of information, related to the message received: <ol style="list-style-type: none"> 1. iuc (either the DIUC or the UIUC). 2. Symbol offset. 3. Burst number. 4. Frame number. PhyParams is optional and will be appended to the primitive by the TA (Test Adapter) only when the function <code>f_appendPhyParams</code> is called out.

4.2.2 The MacBcMessagePort type.

4.2.2.1 Description

This port is used to receive MAC management messages on the broadcast connection.

In addition to the MAC messages, several other informations shall be transmitted in the MsgInd primitive:

- 1) Some fields of the generic Header of the MAC PDU that contained the MAC management message.
- 2) A "rawdata" field that actually contains the complete Mac management message (From Message Type until the end) as received from the IUT and before any manipulation of the message by the SUT adapter (i.e. TLV reordering and addition of the default TLVs not sent by the IUT, see clause 4.2.1).
- 3) Any other PHY information related to the receipt of the MAC management message and that can be required in some test cases (e.g. frame number).

4.2.2.2 Primitives of the BcMacMsg port

Two primitives of type MacBcMsgPrimitives are currently defined:

- 1) The BcMsgReq type primitive - to send MAC management messages in the broadcast connection to the IUT.

NOTE: This primitive is actually not used. Some external functions are currently used to set-up the content of the broadcast connection.

- 2) The BcMsgInd type primitive - to receive MAC management messages from the IUT.

Table 3: Fields of the BcMsgReq type primitive

Field name	description
cid	Contains the value of the CID field (16 bits) transmitted in the Header of the MAC PDU that contained the MAC management message.
bcMsgInOut	The MAC management message sent on a broadcast connection. This field is a union type of all MAC management message types to be sent on the broadcast connection.

Table 4: Fields of the BcMsgInd type primitive

Field name	description
cid	Contains the value of the CID field (16 bits) transmitted in the Header of the MAC PDU that contained the MAC management message.
bcMsgInOut	The MAC management message received on a broadcast connection. This field is a union type of all MAC management message types to be received on the broadcast connection.
phyParams	This field contains the 4 following pieces of information, related to the message received: <ol style="list-style-type: none"> 1. iuc (either the DIUC or the UIUC) 2. Symbol offset 3. Burst number 4. Frame number PhyParams is optional and will be appended to the primitive by the TA (Test Adapter) only when the function <code>f_appendPhyParams</code> is called out.

4.2.3 The MacPduPort type.

4.2.3.1 Description

This port is used to send or receive MAC PDUs.

4.2.3.2 Primitives of the MacPdu port

Two primitives of type MacPduPrimitives are currently defined:

- 1) The PduReq type primitive - to send MAC PDUs to the IUT.
- 2) The PduInd type primitive - to receive MAC PDUs from the IUT.
- 3) The UIFbkReq type primitive - to send Type II Feedback header to the IUT.
- 4) The UIFbkInd type primitive - to receive Type II Feedback header from the IUT.

The primitives PduReq and PduInd contain a list of MacPdu type elements.

Table 5: Fields of the PduReq type

Field name	description
macPduList	List of (record of) MacPdu type fields (see below).

Table 6: Fields of the PduInd type

Field name	description
macPduList	List of (record of) MacPdu type fields (see below).
phyParams	This field contains the 4 following pieces of information, related to the message received: <ol style="list-style-type: none"> 1. iuc (either the DIUC or the UIUC). 2. Symbol offset. 3. Burst number. 4. Frame number. PhyParams is optional and will be appended to the primitive by the TA (Test Adapter) only when the function <code>f_appendPhyParams</code> is called out.

Table 7: Fields of the MacPdu type

Field name	description
macHeader	Contains the PDU header.
perPduSubHeader	Contains the set of Per PDU Subheader (Mesh, Grant management, Fragmentation and Fast feedback).
macSduList	Contains a list of SDUs (Packing Subheader + payload).
crc	Contains the CRC code (32 bits).

The primitives UIFbkReq and UIFbkInd contain fields as described in tables 8 and 9:

Table 8: Fields of the UIFbkReq type

Field name	description
macUIFbk	"MacSignalingHeaderType2" record type that collects the fields of a Feedback Header MPDU (with or without CID).

Table 9: Fields of the UIFbkInd type

Field name	description
macUIFbk	"MacSignalingHeaderType2" record type that collects the fields of a Feedback Header MPDU (with or without CID).
phyParams	This field contains the 4 following pieces of information, related to the message received: <ol style="list-style-type: none"> 1. iuc (either the DIUC or the UIUC). 2. Symbol offset. 3. Burst number. 4. Frame number. PhyParams is optional and will be appended to the primitive by the TA (Test Adapter) only when the function <code>f_appendPhyParams</code> is called out.

4.2.4 The PhyPort type

4.2.4.1 Description

This port is used to receive information from the MS that is not transmitted in PDUs. These pieces of information, which are use for ranging purpose, are:

- 1) Information transmitted by the MS in the OFDMA Ranging Region: CDMA Ranging Code.
- 2) Symbol number and frame number of the ranging code in the ranging region.
- 3) Power level, timing and offset frequency adjust value as determined by the SUT adapter.
- 4) HARQ Ack from feedback channels (MS).
- 5) CQICH from Feedback channels (MS).

4.2.4.2 Primitives of the Phy Port

One primitive is currently defined to be sued with the Phy port: the PhyInd primitive. The Phy port is only intended to receive from the TA (Test Adapter) information concerning PHY values, which are themselves generated by the test system.

The Phy port uses only one primitive: phyInd , which is of one of the following type, according to the type of PHY information expected from the test system:

- 1) OFDMACodeParams; or
- 2) StuffBytesParams; or
- 3) HarqAckNack; or
- 4) CQIChannel.

Table 10: Fields of the PhyInd primitive type (union type)

Field name	description
ofdmaCodeParams	Contains OFDMA parameters: <ul style="list-style-type: none"> • ranging code (as received in the Ranging Region); • power level adjust; • timing adjust; • offset frequency adjust; • symbol number (of the ranging Code in the ranging region); • frame number (of the ranging Code in the ranging region).
stuffBytesParams	Contains the following fields: <ul style="list-style-type: none"> • stuffBytes (octetstring); • symbolOffset, (8 bits) // Symbol offset from the start of the UL subframe; • frameNr (24 bits) //Frame number of the frame in which the stuff bytes were received.
harqAckNack	Contains the following fields: <ul style="list-style-type: none"> • ackBitmap; • acid; • frameNumber of the received ACK message.
cqiChannel	Contains the following fields: <ul style="list-style-type: none"> • fastFeedbackChannelBits; • frameNumber of the received CQICH message; • channelNumber.

4.2.5 The TaPort type.

4.2.5.1 Description

The Test Adapter port (TaPort) is used to enable the TTCN execution to communicate with the test adapter. This will allow to enforce the test adapter, to proceed with actions or transmit information to the TTCN execution environment, which are not available by invoking other port, as for instance:

- 1) Insert values or wrong encodings in the broadcast messages that are automatically sent by the Test Adapter.
- 2) Change burst profile.
- 3) Provides UL allocation.
- 4) Get Frame number of the PDU sent by TTCN send command over the TA.

The Ta type port handles the communication between TTCN and the test adapter (TA) via a simple command/response protocol as shown in figure 23:

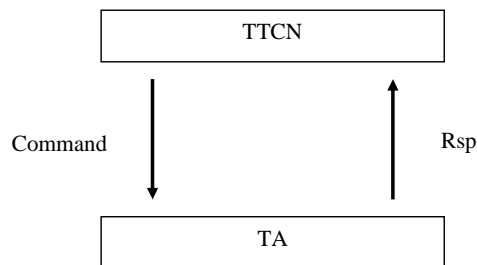


Figure 23: HO Command / Response mechanism between TTCN and TA

Example of using TaPort to get the frame number of a PDU sent by the TTCN execution:

```

// frame Number of the next sent Pdu required
ta.send(m_retrievePhyParamsOfNextSentPdu_Cmd(e_macBcMsgPort));

t_wait.start;
macPdu.send(pdu_req(v_macPduListToBeSent));

alt
{
  [] ta.receive(mw_retrievePhyParamsOfNextSentPdu_Rsp)-> value v_taPrimitives {
    // Store Frame Nr
    v_startFrameNr := v_taPrimitives.retrievePhyParamsOfNextSentPdu_Rsp.phyParams.frameNr;
  }
  [] t_wait.timeout { ... }
}
  
```

4.2.5.2 Primitives of the TaPort

Only primitive type is currently defined: TaPrimitive, which is a union of several structured (Record) types, each one dedicated to the action required from the Test Adapter. All primitives using at least one field to identify the type of primitive: taPrimitiveMsgType. There are two kind of primitives: Cmd (command) and Rsp (Response).

The Cmd primitive are used to request the TA to proceed with specific actions, the Rsp primitives confirms that the Cmd primitives were proceeded correctly and pass some values to the TTCN TE as necessary.

For example:

```

type record UlMap_FastFeedbackChannelIe_Cmd {
  TaPrimitiveMsgType taPrimitiveMsgType
}
  
```

Used to request TA to include a FasFeedback Channel IE in the UL-MAP message.

and

```

type record UMap_FastFeedbackChannelIe_Rsp {
  TaPrimitiveMsgType taPrimitiveMsgType,
  FncRetCode fncRetCode,
  FrameNumber frameNr1
}

```

This primitive is used in receive events for the TA to send the frame number of the UL-MAP message, where the FastFeedback Channel IE was included (frameNr1 formal parameter).

All TA Commands and Responses are part of the HTML documentation (T3doc) provided with this Word document (see annex F).

4.3 Port mapping rules

TTCN-3 enables activation and mapping of ports on a very flexible manner. As MAC messages are also contained in MAC PDUs, some rules apply to manage the mapping of received PDU to the right port, accordingly to the port configuration used in the test case.

Actually the mapping of message to ports need to be different depending if both MacPdu and MacMsg (and BcMacMsg) or only one of both are used. This gives 3 possible cases:

- 1) If only a MacPdu port has been mapped in the TC (and no MacMsg nor BcMacMsg ports), then all PDU, including PDUs containing a MAC management message, are directly sent to the MacPdu port. MAC PDU, with non-generic header is also sent to the MacPdu port. When mapping only the MacPdu port, the function `f_setBcMsgFilter` shall be used as necessary in order to indicate to the test adapter which broadcast messages shall be received.
- 2) If a MacPdu and a MacMsg/BcMacMsg are mapped in the TC, then all PDU are sent to the MacPdu port, except the MAC PDU containing MAC management messages, which are sent resp. to the MacMsg or the BcMacMsg port. MAC PDU, with non-generic header is also sent to the MacPdu port.
- 3) If only a MacMsg or a BcMacMsg port is mapped in the TC, then MAC PDUs, not containing a MAC management message are not received in the TTCN. This avoids in particular to get MAC PDUs matching in the default behaviour.

4.4 PDU sending/receiving rules

The rules for sending (resp. receiving) MAC PDU with MacPduList from TE (TTCN execution) to SA (SUT adapter) is as follows:

- 1) From TE to SA: the MAC PDUs from a MacPduListy can be sent concatenated on the Air interface.
- 2) From SA to TE: the MAC PDU shall be provided one by one from the SA to the TE.

NOTE: There are no test cases required to receive concatenated PDUs (i.e. to check the IUT sends concatenated data).

5 Untestable Test Purposes (TP)

This clause gives a list of TP, which are not implemented in the ATS due to the chosen ATM or other restrictions.

Table 11: Untestable TP

Test Case Name	Reason
void	

6 ATS conventions

The ATS conventions are intended to give a better understanding of the ATS but they also describe the conventions made for the development of the ATS. These conventions shall be considered during any later maintenance or further development of the ATS.

The ATS conventions contain two clauses, the naming conventions and the implementation conventions. The naming conventions describe the structure of the naming of all ATS elements. The implementation conventions describe the functional structure of the ATS.

To define the ATS, the guidelines of the document ETS 300 406 [4] were considered.

6.1 Testing conventions

6.1.1 Testing States

BS Null: The BS is switched on and sends broadcast messages.

SS Null: The SS is switched on and is ready to receive broadcast messages.

6.1.2 HiperMAN default values: Reception and transmission at ATS level

IEEE P802.16-2004 [10] lists many default TLV values. The spec says that devices SHOULD NOT transmit TLVs if the default value applies. However, this is NOT a requirement. Thus, one tested device may not transmit the default TLVs (or a subset of these default TLVs) while another may transmit all TLVs including the defaults. Including all the possible combinations of sent and received default TLVs in an ATS is problematic. Therefore:

- For ATS purposes, all TLVs are assumed to be sent and received at the ATS level.
- The Test Adapter will fill in the missing received TLVs with a TLV containing the default value and pass it up to the ATS.
- The Test Adapter may or may not transmit default TLVs received from the ATS to the IUT. This is a test equipment vendor decision.

6.1.3 Templates

Separate templates are defined for use in sending and receiving operations.

Template definitions should avoid using matching attributes such as "*" or "?" for complete structured values, e.g. record or set of values.

PIXIT parameter values are passed as parameters into templates.

6.1.4 Functions

The WMx ATS differentiates between external functions for which only the signature is specified and functions completely defined in the ATS. The completely defined functions are separated according to their use for SS or BS testing and preamble and postamble functions.

The SS and BS testing functions are grouped in a general configurations functions group and separate groups with functions used for testing different types of functionality.

Each type of function is implemented in a separate module, although there may be multiple modules for each function type. The following general rules apply:

- Functions use the "runs on" statement wherever this is possible.
- Each function provides a return value wherever this is possible. The return value used is the enumeration type "FncRetCode" defined in the WMx_Types.ttcn file.

EXAMPLE: `WMx_Types.FncRetCode`.

- The *stop* statement is used only for controlled test component shutdown.

6.2 Naming conventions

6.2.1 General guidelines

The naming convention is based on the following underlying principles:

- In most cases, identifiers should be prefixed with a short alphabetic string (specified in table 12) indicating the type of TTCN-3 element it represents.
- Suffixes should not be used except in those specific cases identified in table 7.
- Prefixes and suffixes should be separated from the body of the identifier with an underscore ("_"):

EXAMPLE 1: `c_sixteen, t_wait_max`.

- Only module names, data type names and module parameters should begin with an upper-case letter. All other names (i.e. the part of the identifier following the prefix) should begin with a lower-case letter.
- The start of second and subsequent words in an identifier should be indicated by capitalizing the first character. Underscores should not be used for this purpose.

EXAMPLE 2: `f_authenticateUser`.

Table 12 specifies the naming guidelines for each element of the TTCN-3 language indicating the recommended prefix, suffixes (if any) and capitalization.

Table 12: TTCN-3 naming convention

Language element	Naming convention	Prefix	Suffix	Example	Notes
Module	Use upper-case initial letter	<i>none</i>	<i>none</i>	WMx_Templates	
TSS grouping	Use all upper-case letters	<i>none</i>	<i>none</i>	TP_RT_PS_TR	
Item group within a module	Use lower-case initial letter	<i>none</i>	<i>none</i>	messageGroup	
Data type	Use upper-case initial letter	<i>none</i>	<i>none</i>	SetupContents	
List type identifiers	Use upper-case initial letter	<i>none</i>	<i>none</i>	DIMapleList	
Message template	Use lower-case initial letter	m_	<i>none</i>	m_setupInit	
Message template with wildcard or matching expression	Use lower-case initial letters	mw_	<i>none</i>	mw_setupBasic	
Port instance	Use lower-case initial letter	<i>none</i>	<i>none</i>	signallingPort	
Test component ref	Use lower-case initial letter	<i>none</i>	<i>none</i>	userTerminal	
Signature	Use lower-case initial letter	s_	<i>none</i>	s_callSignature	
External function	Use lower-case initial letter	xf_	<i>none</i>	xf_calculateLength()	
Constant	Use lower-case initial letter	c_	<i>none</i>	c_maxRetransmission	
Function	Use lower-case initial letter	f_	<i>none</i>	f_authentication()	
Altstep	Use lower-case initial letter	a_	<i>none</i>	a_receiveSetup()	
Altstep (Default)	Use lower-case initial letter	d_	<i>none</i>	d_receiveOtherMessages()	
Variable	Use lower-case initial letter	v_	<i>none</i>	v_basicCid	
Variable, global to component	Use lower-case initial letter	g_	<i>none</i>	g_ssSimu.basicCid	
Timer	Use lower-case initial letter	t_	_min _max	t_wait t_auth_min	Note 1
Module parameters PICS values PIXIT values	Use all upper case letters	<i>none</i>	<i>none</i>	PIC_T7PXT_TNOAC	Note 2
External constant	Use lower-case initial letter	xc_	<i>none</i>	xc_macId	
Parameterization	Use lower-case initial letter	p_	<i>none</i>	p_macId	
Enumerated Value	Use lower-case initial letter	e_	<i>none</i>	e_synCpk	
NOTE 1: If a time window is needed, the suffixes "_min" and "_max" should be appended.					
NOTE 2: In this case it is acceptable to use underscore as a word delimiter.					

6.2.2 Test Case (TC) identifier

Table 13: TC naming convention

TC <st> <pg> <fg> <sg> <ini> <x> H<nnn>		
<st> = side type	BS	Base Station
	SS	Subscriber Station
<pg> = protocol group	CDM	Channel Descriptors and Maps
	RLC	Radio Link Control
	INI	Registration, IP Connectivity, and Parameter Transfer
	PKM	Privacy and Key Management
	DS	Dynamic Services
	BWA	Bandwidth Allocation and Polling
	RER	Reset and Re-registration
	CCC	Clock Comparison
	MAC	MAC PDU Construction
	PCS	Packet CS
<fg> = function group	MAP	Map and Frame Structure
	CD	Channel Descriptors
	CDC	Channel Descriptor Change
	IRNG	Initial Ranging
	PRNG	Periodic Ranging
	DBPC	Downlink Burst Profile Management
	SBC	Negotiate Basic Capabilities
	REG	Registration
	IPC	IP Connectivity
	AUTH	Authentication/Authorization

TC <st> <pg> <fg> <sg> <ini> <x> H<nnn>		
	TEK	Encryption Key Transfer
	SAM	Security Association Management
	EKS	Encryption and Key Scheduling
	DSA	Dynamic Service Addition
	DSC	Dynamic Service Change
	DSD	Dynamic Service Deletion
	REQ	Request/Grant
	MCP	Multicast Polling
	PACK	Packing
	FRAG	Fragmentation
	CAT	PDU Concatenation
	CRC	Cyclic Redundancy Check (CRC)
	ARQ	ARQ
	PCU	Packet CS Usage
	CLS	Classification
	CDS	Classifier DSx Signalling
	PHS	Payload Header Suppression
<sg> = subfunction group	INIT	Initialization
	OPN	Operation
	RLV	Relevance
	KU	Key Usage
	ENC	Encryption
	DEC	Decryption
<ini> = initiator of procedure or direction of flow	Bslni	Procedure is initiated by BS
	Sslni	Procedure is initiated by SS
	DL	Downlink
	UL	Uplink
<x> = type of testing	BV	Valid Behaviour Tests
	BI	Invalid Syntax or Behaviour Tests
	BO	Inopportune Behaviour Tests
	TI	Timer and Counter Tests
<nnn> = sequential number	Hnnn	(H000, H001, etc.)

EXAMPLE: TP identifier: TP/SS/RLC/IRNG/BV-H002
 TC identifier: TC_SS_RLC_IRNG_BV_H002.

6.3 Service Flow parameter support

This clause describes which values of the service flow parameters: CS specification and Data Delivery services, are supported in the ATS.

6.3.1 CsSpecification support

CsSpecification	Supported
e_packetIpv4(1)	Yes
e_packetIpv6(2)	Yes
e_packetIeee8023Ethernet(3)	Yes
e_packetIeee802IqVlan(4)	No
e_packetIpv4OverIeee8023Ethernet(5)	Yes
e_packetIpv6OverIeee8023Ethernet(6)	No
e_packetIpv4OverIeee802IqVlan(7)	No
e_packetIpv6Over802IqVlan(8)	No
e_atm(9)	No
e_packetIpv4OverIeee8023EthernetRohc(10)	No
e_packetIpv6OverIeee8023EthernetEcrt(11)	No
e_packetIp2Rohc(12)	No
e_packetIp2Ecrt(13)	No

6.3.2 DataDeliveryServiceType

DataDeliveryServiceType	Supported
e_unsolicited_grant_service(0)	Yes
e_realtime_variable_rate_service(1)	No
e_non_realtime_variable_rate_service(2)	No
e_best_effort_service(3)	Yes
e_xtded_realtime_variable_rate_service(4)	No

6.4 Dispatching of test cases over TTCN modules

In order to maintain a reasonable size of modules containing the test case definitions, the test cases are defined in different module according to their groups.

Each new test case is always defined in either the WMx_BsTestcases_16e or WMx_SsTestcases_16e.

After the validation of a test case, the signedOff test case will be move to a dedicated module accordingly to its TP group.

The module structure is as indicated in tables 14 and 15.

Table 14: Module names for signedoff SS test cases

TP groups	Module names	description
SS_CDM_CDC	WMx_SsTestcases_ChannelDescriptorsAndMaps_16e	Channel Descriptors and Maps
SS_CDM_MAP		
SS_RLC_FBK	WMx_SsTestcases_NetworkEntry_16e	Network Entry
SS_RLC_IRNG		
SS_RLC_PRNG		
SS_RLC_SBC		
SS_INI_REG		
SS_RER		
MS_IDM_LOC	WMx_SsTestcases_Mobility_16e	Mobility
MS_SLM_PW1		
MS_GHF_NTA_SCAN		
MS_GHF_HO_INI_MS		
MS_GHF_HO_TER		
MS_GHF_NWR		
SS_DS_DSA	WMx_SsTestcases_DynamicServices_16e	Dynamic Services
SS_DS_DSC		
SS_DS_DSD		
SS_BWA_REQ	WMx_SsTestcases_BandwidthAllocation_16e	Bandwidth Allocation
SS_BWA_CBR		
SS_MAC_BV	WMx_SsTestcases_MacPduFormat_16e	Mac Pdu Format
SS_MAC_CAT		
SS_MAC_CRC		
SS_MAC_FRAG		
SS_MAC_PACK		
SS_CSOE_IPV4		
SS_CSOE_IPV6		
SS_CSOC		
SS_PHS		
SS_ARQ	WMx_SsTestcases_ArqOrHarq_16e	Arq and Harq
SS_ARQ_RXD		
SS_ARQ_TXD		
SS_ARQ_RE		
SS_RLC_HARQ		
SS_SEC_PKMv1	WMx_SsTestcases_Security_16e	Security
SS_SEC_PKMv2		

Table 15: Module names for signedoff BS test cases

TP groups	Module names	description
BS_CDM_MFS	WMx_BsTestcases_ChannelDescriptorsAndMaps_16e	Channel Descriptors and Maps
BS_CDM_CD		
BS_CDM_CDC		
BS_CDM_MAP		
BS_RLC_FBK	WMx_BsTestcases_NetworkEntry_16e	Network Entry
BS_RLC_ACQ		
BS_RLC_IRNG		
BS_RLC_SBC		
BS_INI_REG		
BS_RER		
BS_IDM_LOC	WMx_BsTestcases_Mobility_16e	Mobility
BS_IDM_NWR		
BS_IDM_PG		
BS_IDM_TIDM		
BS_GHF_NTA		
BS_GHF_HO		
BS_SLM_PW1		
BS_DS_DSA	WMx_BsTestcases_DynamicServices_16e	Dynamic Services
BS_DS_DSC		
BS_DS_DSD		
BS_DS_QPS		
BS_BWA_CBR	WMx_BsTestcases_BandwidthAllocation_16e	Bandwidth Allocation
BS_BWA_REQ		
BS_MAC	WMx_BsTestcases_MacPduFormat_16e	Mac Pdu Format
BS_MAC_CAT		
BS_MAC_CRC		
BS_MAC_FRAG		
BS_CSOE_IP4	WMx_BsTestcases_ConvergenceSublayer_16e	Convergence sublayer
BS_CSOE_IP6		
BS_CSOC		
BS_CSOC_CDS		
BS_CSOC_IP4C		
BS_CSOC_IP6C		
BS_PHS		
BS_ARQ		
BS_ARQ_RXD		
BS_ARQ_TXD		
BS_RLC_HARQ		
BS_SEC_PKMv2	WMx_BsTestcases_Security_16e	Security

7 External functions

The document concerning the external functions is provided in HTML format as an output from the T3Doc Open Source Tool.

To look at this HTML documentation, please refer to the instructions in annex F.

8 Test strategies

Due the combination of PHY and MAC procedures, the protocol specified in [2] and [3] requires, by nature, using dedicated strategies for testing.

In particular, the WiMAX/Hiperman processes can not only be tested by simply invoking SEND or RECEIVE events over a communication port (using PDU for instance),but requires to set up test configurations on the IUTs or in the Test system, and invoking external functions from the Test Adapter (TA).

In this clause test strategies applying to some WiMAX processes are described in details.

8.1 HARQ testing

Testing HARQ requires using some dedicated testing features described in the following clauses.

8.1.1 HarqAckNack PHY port message

A dedicated PhyPort primitive is used to receive the HARQ ACK or NCAK messages from the test adapter (TA). This messages from the MS cannot be received on the usual MAC or PDU port, but requires the TA to decode them from the UL frames and to pass the messages to TTCN over the PHY port.

The following type is added to the union in the PhyMsgIn type:

```
type record HarqAckNack {
  AckBitmap  ackBitmap,
  AcId       acId,
  FrameNumber frameNumber
}
```

The filter on phyPort shall be used in order to indicate the test adapter which kind of messages it shall enqueue.

```
f_bsMapPhyPort({e_harqAckNackParams });
```

8.1.2 HARQ TA Primitives

The following TA commands types are used to test HARQ:

DIMap_Harq_Cmd.

DIMap_Harq_Rsp.

UIMap_Harq_Cmd.

UIMap_Harq_Rsp.

WrongHarqCrcOfNextSentPdu_Cmd.

WrongHarqCrcOfNextSentPdu_Rsp.

MseHarq_Cmd.

MseHarq_Rsp.

8.1.3 HARQ Broadcast Message Filter

The following Broadcast Message filter type is required to test HARQ:

```
type enumerated BcMacMngtMsgType {
...
  e_dlMapWithHarqDlMapIe          ( 8),
...
  e_ulMapHarqAckChRegionAllocIE  (11),
  e_ulMapWithHarqUlMapIe         (12),
...
}
```

8.1.4 HARQ external Functions

The following external function is required to test HARQ:

```
cf_bsSimuAssignUlBurst_HARQACKCH
```

NOTE: Create a burst for HARQ ACKCH in the UL subframe.

8.2 Fast Feedback testing

Testing Fast FeedBack requires to use some dedicated testing features described in the following clauses.

8.2.1 CQICH PHY port message

A dedicated PhyPort primitive is used to receive CQICH messages from the Test Adapter (TA). This messages from the MS cannot be received on the usual MAC or PDU port, but requires the TA to decode them from the UL frames and to pass the messages to TTCN over the PHY port.

The following type is added to the union in the PhyMsgIn type::

```
type record CQICHannel
{
  FastFeedbackChannelBits fastFeedbackChannelBits,
  FrameNumber frameNumber,
  ChannelNumber channelNumber
}
```

The filter on phyPort shall be used in order to indicate the test adapter which kind of messages it shall enqueue.

```
f_bsMapPhyPort({e_cqiChannel});
```

8.2.2 CQICH TA Primitives

The following TA commands types are required to test CQICH fast feedback channel:

```
UIMap_CqichAllocationIe_Cmd,
UIMap_CqichAllocationIe_rsp,
SendCqichCodeword_Cmd,
SendCqichCodeword_Rsp,
```

8.2.3 CQICH Broadcast Message Filter

The following Broadcast Message filter type is required to test CQICH and Fast Feedback

```
type enumerated BcMacMngtMsgType {
...
  e_ulMapWithCqichAllocationIe      (13),
  e_ulMapWithFastFeedbackAllocIe    (14)
...
}
```

8.2.4 CQICH external Functions

The following external function is required to test CQICH:

```
cf_bsSimuAssignULBurst_CQICH
```

NOTE: Allocates Fast Feedback with UIUC0.

8.3 Handover testing

8.3.1 Testing Serving BS

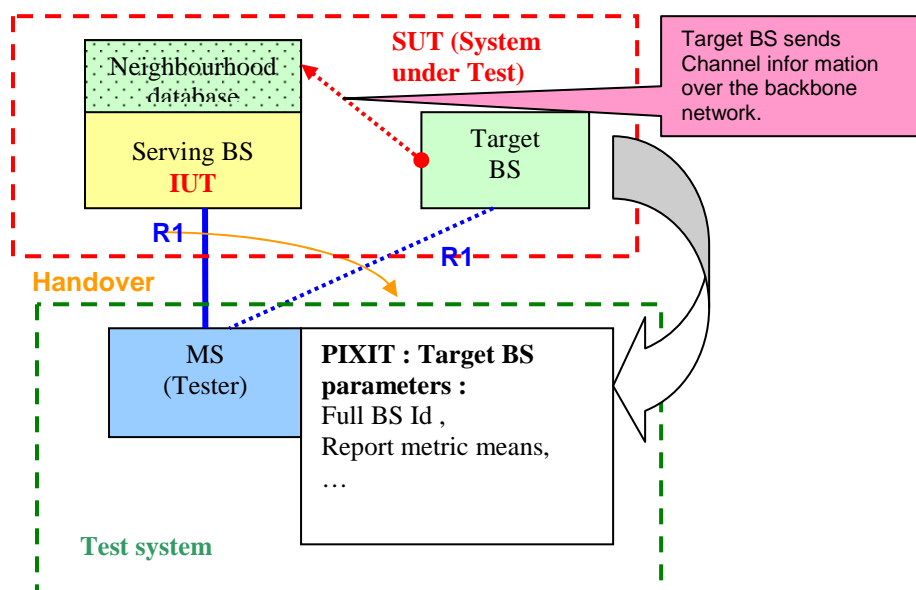


Figure 24: HO testing for the Serving BS

In this configuration the IUT is the Serving BS. But to test handover procedures, a Target BS is also required. This Target BS shall be provided as part of the System Under Test (SUT).

In order to enable testing the different HO procedures specified in IEEE 802.16 [3], the following requirements are needed:

- the Target BS channel information shall be made available to the neighbourhood database of the Serving BS (e.g. over the backbone network or any other mean);
- Target BS parameter shall be stored in PIXIT parameters to enable the Tester to send HO request with the required Target BS information (BS id, metric values, etc.).

8.3.2 Testing Target BS

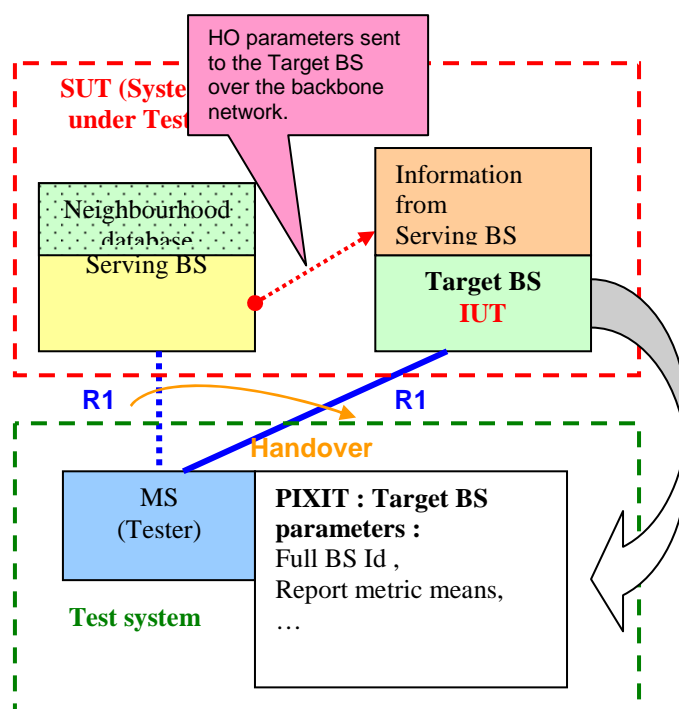


Figure 25: HO testing for the Target BS

In this configuration the IUT is the Target BS. But to test handover procedures, a Serving BS is also required. This Serving BS shall be provided as part of the System Under Test (SUT).

In order to enable testing the different HO procedures specified in IEEE 802.16, the following requirements are needed:

- as applicable, the MS information to enable optimized HO procedures shall be sent from the Serving BS to the Target BS (e.g. over the backbone network or any other mean);
- Target BS parameter shall be stored in PIXIT parameters to enable the Tester to send HO request with the required Target BS information (BS id, metric values, etc.).

8.3.2 Testing MS with Handover or MBS procedure

For testing the MS in an Handover or multiple BS situation, the test system architecture in figure 26 is used

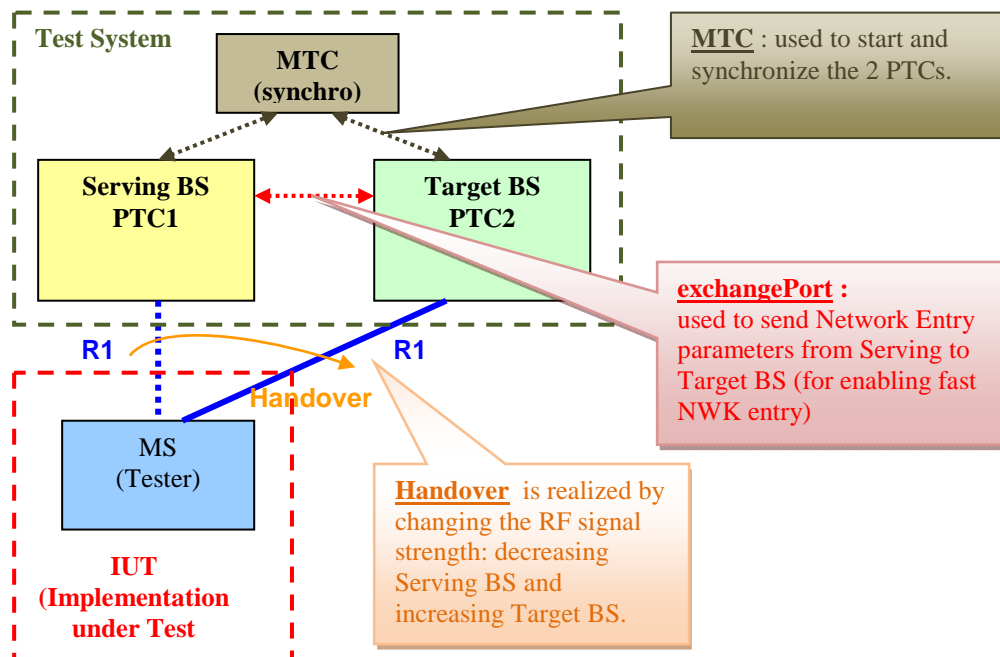


Figure 26: System architecture for testing the MS

The above architecture is used for testing the Handover and the MBS procedures, when the MS is connected to 2 BSs. This architecture is also use for testing PKM procedures over 2 BSs.

The IUT (MS) is initially connected to the Serving BS.

The initial preamble: `f_ssSvgBsNwkEntryAndData`, enables the IUT registration, and then establishes 2 service flows (1 uplink and 1 downlink). Actually the MS may not use Handover if no service flow is active.

Then as necessary the preamble may require to send data from the MS. When the network entry has succeeded the preambles sends some network entry parameters (transport CIDs, SFIDs, SBC and REG TLVs) over the exchange port.

These parameters are received in the PTC2, simulating Target BS, inside the `f_ssGetInfoFromSvgBs` target BS preamble.

8.4 Processing of EAP messages

TTCN test cases are handling PkmV2 messages, which carry on EAP messages. The EAP behaviour is complex. To avoid increasing tremendously the complexity of the TTCN code, the EAP procedures shall rather be simulated by an EAP Server/Client (depending on testing BS or SS). The EAP messages are passed to the EAP emulation, and back, through external.

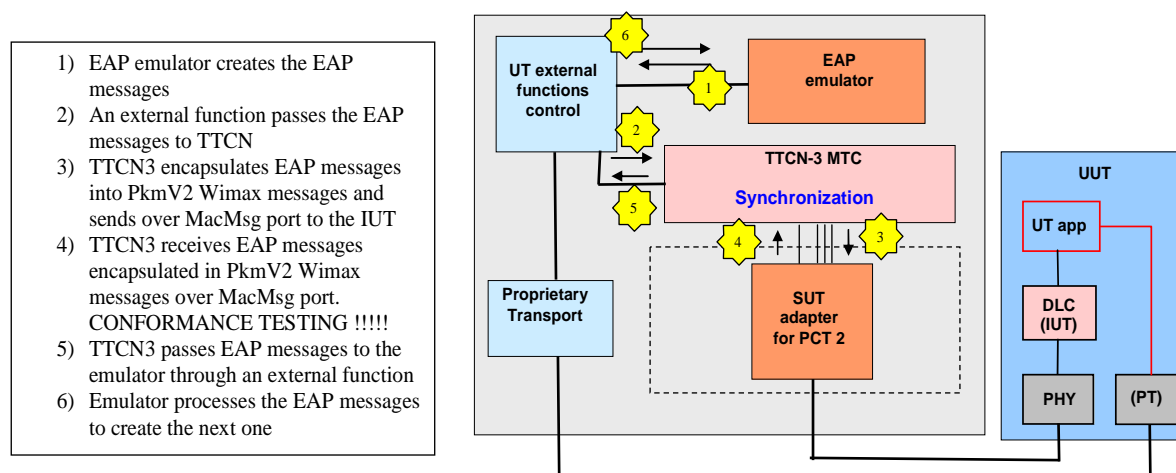


Figure 27: Description of the required EAP procedures

8.5 Handling MAC PDU with Fragmentation, Packing or Concatenation

Correctly handling MAC PDU with either packing, fragmentation or concatenation requires establishing the service flows with ad hoc TLV values. The setting of the TLV for the corresponding service flows as well as other MAC PDU settings is described in the following clauses.

8.5.1 MS (IUT) Processing DL MAC PDUs with FRAG Subheader

- Frag/No Pack in DL SF.
- No Frag/No Pack in UL SF.
- The test system (BS Simu) sends DL MAC PDUs. Each DL MAC PDU contains a FRAG Subheader. The fragmented data is created from 1 EchoRequest.
- The EchoRequest is small enough to allow MSUT to send 1 MAC PDU containing the EchoReply.
- If MSUT replies correctly it can be assumed that it reassembled the received fragmented data correctly.

8.5.2 MS (IUT) Processing DL MAC PDU with PACK Subheaders

- No Frag/ Pack in DL SF.
- No Frag/No Pack in UL SF.
- The test system (BS Simu) sends 1 DL MAC PDU with 2 PACK Subheaders / 2 SDU packets. Each SDU packet contains 1 EchoRequest.
- Each EchoRequest is small enough to allow MSUT to send 1 MAC PDU containing the EchoReply.
- If MSUT replies correctly with 2 Echo Replies it can be assumed that it treated the received packed data correctly.

8.5.3 MS (IUT) Processing DL MAC PDU with PACK and FRAG Subheaders

- Frag/ Pack in DL SF.
- No Frag/No Pack in UL SF.
- The test system (BS Simu) sends 1 DL MAC PDU with 1 FRAG Subheader1 / 1 SDU and 1 DL MAC PDU with 2 PACK Subheaders.
- The EchoRequest is small enough to allow MSUT to send 1 MAC PDU containing the EchoReply.
- If MSUT replies correctly with 1 Echo Reply it can be assumed that it treated the received packed data correctly.

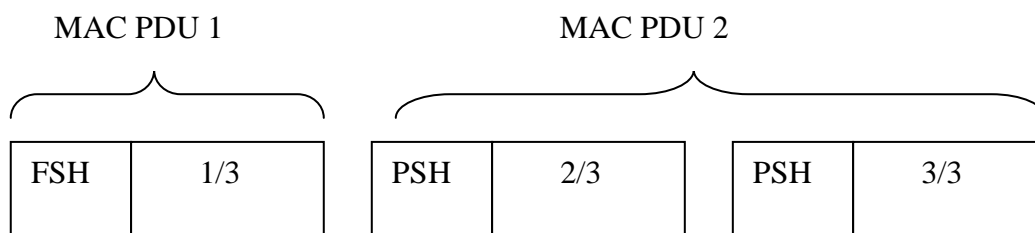


Figure 28: Description of the packed and fragmented PDUs

8.5.4 MS (IUT) Processing concatenated DL MAC PDUs

- No Frag/ No Pack in DL SF.
- No Frag/No Pack in UL SF.
- The test system(BS Simu) sends 1 DL MAC PDU LIST in the same frame. Each MAC PDU contains 1 EchoRequest.
- Each EchoRequest is small enough to allow MSUT to send 1 MAC PDU containing the EchoReply.
- If MSUT replies correctly with 2 Echo Replies it can be assumed that it processed the received concatenated MAC PDUs correctly.

8.5.5 MS (IUT) Generating UL MAC PDU with FRAG Subheader

- DL SF not established.
- Frag/No Pack in UL UGS SF.
- $e_qpskCc1over2/e_qpskCtc1over2$.
- TTCN instructs the test adapter with `xf_grantUiuc` to grant N number of slots.
- TTCN instructs test operator to use packet generator to create UL packets. Each packet is bigger than the granted N number of slots.
- If an UL MAC PDU with FRAG subheader is received in TTCN , then verdict PASS.
- If no UL MAC PDU with FRAG subheader is received in TTCN , then verdict FAIL.

8.5.6 MS (IUT) Generating UL MAC PDU with PACK Subheaders

- DL SF not established.

- No Frag/ Pack in UL UGS SF.
- $e_qpskCc1over2/e_qpskCtc1over2$.
- TTCN instructs the test adapter with `xf_grantUiuc` to grant N number of slots.
- TTCN instructs test operator to use packet generator to create UL packets. Each packet is small enough to fit 2 packets in the granted N number of slots.
- If during a period of 10 seconds an UL MAC PDU with PACK subheaders is received in TTCN , then verdict PASS.
- If during a period of 10 seconds no UL MAC PDU with PACK subheaders is received in TTCN , then verdict INCONC.

8.5.7 MS (IUT) Generating concatenated UL MAC PDUs

- DL SF not established.
- No Frag/ No Pack in UL UGS SF.
- $e_qpskCc1over2/e_qpskCtc1over2$.
- TTCN instructs the test adapter with `xf_grantUiuc` to grant N number of slots.
- TTCN instructs test operator to use packet generator to create UL packets. Each packet is small enough to fit 2 packets in the granted N number of slots.
- If during a period of 10 seconds 2 UL MAC PDUs with same frame number are received in TTCN , then verdict PASS.
- If during a period of 10 seconds no 2 UL MAC PDUs with same frame number are received in TTCN , then verdict INCONC.

8.5.8 BS (IUT) processing UL MAC PDUs with FRAG Subheader

- No Frag/No Pack in DL SF.
- Frag/No Pack in UL SF.
- The test system (MS Simu) sends UL MAC PDUs. Each UL MAC PDU contains a FRAG Subheader. The fragmented data is created from 1 EchoRequest.
- The EchoRequest is small enough to allow BS (IUT) to send 1 MAC PDU containing the EchoReply.
- If BS (IUT) replies correctly it can be assumed that it reassembled the received fragmented data correctly.

8.5.9 BS (IUT) processing concatenated UL MAC PDUs

- No Frag/ No Pack in DL SF.
- No Frag/No Pack in UL SF.
- The test system (MS Simu) sends 1 UL MAC PDU LIST in the same frame. Each MAC PDU contains 1 EchoRequest.
- The EchoRequest is small enough to allow MSUT to send 1 MAC PDU containing the EchoReply.
- If MSUT replies correctly with 2 Echo Replies it can be assumed that it processed the received concatenated MAC PDUs correctly.

8.5.10 BS (IUT) generating DL MAC PDU with FRAG Subheader

- UL SF not needed to be established, but if BS (IUT) is allowed to establish it, if it wished to.
- Frag/No Pack in DL UGS SF.
- TTCN instructs test operator to use packet generator to create DL packets. Each packet is bigger than what can be sent in 1 frame so that BS (IUT) must fragment.
- If an DL MAC PDU with FRAG subheader is received in TTCN , then verdict PASS.
- If no DL MAC PDU with FRAG subheader is received in TTCN , then verdict FAIL.

8.5.11 BS (IUT) generating DL MAC PDU with PACK Subheaders

- UL SF not needed to be established, but if BS (IUT) is allowed to establish it, if it wished to.
- No Frag/ Pack/ SduTypeIndicator = e_varLengthSDUs (11.13.15) in DL UGS SF.
- TTCN instructs test operator to use packet generator to create DL packets. Each packet is small enough to fit 2 packets in the granted N number of slots, i.e. DL packets of 2 bytes.
- If during a period of 10 seconds a DL MAC PDU with PACK subheaders is received in TTCN , then verdict PASS.
- If during a period of 10 seconds no DL MAC PDU with PACK subheaders is received in TTCN , then verdict INCONC.

8.6 Service Flows management for testing BS

8.6.1 Introduction

For testing certain functionalities, the establishment of service flows with specific features is required. For example, for testing fragmentation or packing , the corresponding bits in the Request/transmission Policy shall be set. So that for the TTCN code shall verify that the service flows are suited to the feature to be tested.

For testing MS, it is easy to establish service flows from the tester (BS simulation) by invoking DSA messages with the ad hoc parameters.

But for testing BS, the situation is quite different because:

- Some BSs do not support establishment of service flows initiated by the MS.
- Many BSs establish at least two service flows (one UL and one DL), as soon as the MS has registered to the network.
- When the above service flows, automatically invoked by the BS, are not successfully established, then the BS discards any MS action.

The BS test cases needs to take the above statements into account. The following clauses clarify the procedures implemented in the TTCN code, in order to enable the service flow conditions, applying to the BSs, to be followed.

Because BS service flows are often unsolicited and require to be successful, the handling of DSA invocation is treated in the default behaviour. This use of the TTCN default behaviour avoid mixing in the test case the procedure related to the test purpose and those related to service flows.

For test cases where specific service flows are required, the TTCN code will find the required service flows as following:

- Either the SFs can be found among the ones created by the BS.
- Or the service flow can be created by from the TTCN if the BS supports MS initiated service flow (mostly not).
- Or the TTCN code can enforce the IUT (BS) to create a suitable service flow.

The procedures associated with handling of service flows for testing BSs are described in the following clauses:

- Handling of the Service Flows events in the default procedure: `d_bsMacMsg` works.
- Meaning and usage of the variables used by TTCN for handling service flows.
- How to invoke or trigger the invocation of a specific service flow that has not already been established by the BS.

8.6.2 Handling of service flows in the default.

`d_bsMacMsg` consists of some alt statements which handle messages on the `macMsg` port. The ones related to service flow management may be spread into three groups:

- To receive `DsaReq` messages.
- To receive `DsaAck` messages.
- To handle T8 and T10 timers.

For the first group there are two alt statements, one for `DsaReq` for downlink and one for uplink, but they are essentially implemented in the same way.

First of all, they will check whether the `DsaReq` message received is a retransmission or not by checking transaction identifier. If transaction identifier was already received then `DsaReq` message will be treated as a retransmission, therefore `DsaRsp` message which was sent previously will be sent again and T8 timer will be started.

If transaction identifier was not received yet, `DsaReq` message will be checked and afterwards `DsaRsp` message will be sent with no TLVs because main features received in `DsaReq` message were already checked and all the rest must be accepted. Then, finally, T8 timer will be started.

In order to check `DsaReq` messages, `f_checkAndSaveDsaBsReq` function is used. This function must:

- a) check that main features are correct (range of some identifiers, presence of some TLVs, etc.);
- b) save information received into the system variables;
- c) check service flow received satisfies TC needs and set related system variables (`firstULSfIndex`, `secondULSfIndex`, `firstDLSfIndex`, `secondDLSfIndex`).

For item c), `f_checkDsaBsReqNeeded` function is checking that SF characteristics (CS layer + Packet Classification, Data Delivery Services, PHS and ARQ) match TC needs.

For the second group, there is only one alt statement which is receiving `DsaAck` messages, where messages from the existing SF transactions will be matched. Afterwards, T10 timer will be started using the index from matched `DsaAck` message. Only when T10 expires respective service flow is either to be used (marked by `firstUISrvFlowUp`, `secondUISrvFlowUp`, `firstDISrvFlowUp` and `secondDISrvFlowUp`) or to be ignored because it is a service flow which does not correspond to TC needs.

T8 timeouts are used for the retransmissions of `DsaRsp` messages.

T10 timeouts are used for setting up variables above mentioned (if appropriate) to indicate that service flow is ready.

8.6.3 Global variables used for the service flow management

- ssDsaIni, Mse-Initiated Service Flow is supported.
- firstDISrvFlowUp, First downlink SF which satisfies TC needs is completely established.
- secondDISrvFlowUp, Second downlink SF which satisfies TC needs is completely established.
- firstUISrvFlowUp, First uplink SF which satisfies TC needs is completely established.
- secondUISrvFlowUp, Second uplink SF which satisfies TC needs is completely established.
- firstDLSfIndex, Index for the First downlink SF which satisfies TC needs. It is setup when f_checkDsaBsReqNeeded has succeeded. Otherwise, it will be omitted.
- firstULSfIndex, Index for the First uplink SF which satisfies TC needs. It is setup when f_checkDsaBsReqNeeded has succeeded. Otherwise, it will be omitted.
- secondDLSfIndex, Index for the Second downlink SF which satisfies TC needs. It is setup when f_checkDsaBsReqNeeded has succeeded. Otherwise, it will be omitted.
- secondULSfIndex, Index for the Second uplink SF which satisfies TC needs. It is setup when f_checkDsaBsReqNeeded has succeeded. Otherwise, it will be omitted.
- transactionIdList, List of transaction identifiers managed during the TC execution. Concerning service flow management when testing Bs, each element is a transaction identifier received in a new DsaReq message, so it is setup in f_checkAndSaveDsaBsReq function.
- transportCidList, List of transport connection identifiers managed during the TC execution. When testing Bs, each element is a transport connection identifier received in a new DsaReq message, so it is setup in f_checkAndSaveDsaBsReq function.
- sfidList, List of transaction identifiers managed during the TC execution. Concerning service flow management when testing Bs, each element is a transaction identifier received in a new DsaReq message, so it is setup in f_checkAndSaveDsaBsReq function.
- targetSaidList, List of target SA identifiers managed during the TC execution. When testing Bs, each element is a target SA identifier received in a new DsaReq message, so it is setup in f_checkAndSaveDsaBsReq function.
- dsxRspRetriesList, List of DsaRsp messages retries.
- dsaReqList, List of DsaReq messages. When testing Bs, DsaReq messages received.
- dsaRspList, List of DsaRsp messages. When testing Bs, DsaRsp messages sent. Variable to be used for retransmissions.
- firstDsaAckReceivedList,
- directionServiceFlowList, List of service flow directions. That allows to checking the service flow direction in an easier and quicker way.

Indexes are managed by f_checkAndSaveDsaBsReq function. This function will take next free index in order to save properly features above described. Also, this function will update the next free index variable for the next messages to be handled.

f_checkAndSaveDsaBsReq function will return the index used so that it can be used by other functions.

All lists are synchronized by the same index. For example, first DsaReq message received will be saved in dsaReqList[0], the response to such message will be saved in dsaRspList[0], service flow direction in directionServiceFlowList[0], and so on. Timer lists will use same indexing.

8.6.4 How to invoke a required SF that is not available

The function which checks that a required service flow is established is `f_bsServiceFlowEstablished`. This function consists of three main parts.

- First part waits for the service flow establishments initiated by the BS.
- Next one is checking service flows established. That is done by checking.
- `SrvFlowUp` system variables. If a SF required by a specific TC is not come available, the BS will be triggered to create it or it will be initiated by Mse depending on `ssDsaIni` system variable.
- Last part is very similar to the first one, but if any of the SF needed by TC has not been established, TC execution will be stopped and verdict set up to inconclusive.

By default, all TCs are waiting for the establishment of 2 SFs, one uplink and one downlink. That is done by the following system variables initialization:

- `vc_simu.firstDISrvFlowUp := false.`
- `vc_simu.secondDISrvFlowUp:= true.`
- `vc_simu.firstUISrvFlowUp := false.`
- `vc_simu.secondUISrvFlowUp:= true.`

Tester allows BS to establish as many SFs as it wants, but for successful test execution this second Sf usually is not needed and it will not be used and, therefore, marked as true. Only small number of TCs use the second SF and it will be marked as false at the beginning of the TC, after the initialization of system variables (`f_init` function).

Annex A (normative): WiMAX/HiperMAN 1.3.1 Abstract Test Suite (ATS)

This ATS has been produced using the Testing and Test Control Notation (TTCN-3) according to ES 201 873-1 [9].

A.1 The TTCN-3 Module

The TTCN-3 code corresponding to the ATS is contained in an archive named `ts_10254503v010201p0.zip` which accompanies the present document.

Annex B (normative): WiMAX/HiperMAN 1.3.1 Partial PIXIT proforma for IUT BS

Notwithstanding the provisions of the copyright clause related to the text of the present document, ETSI grants that users of the present document may freely reproduce the Partial PIXIT proforma in this annex so that it can be used for its intended purposes and may further publish the completed Partial PIXIT.
--

The PIXIT Proforma is based on ISO/IEC 9646-6 [7]. Any needed additional information can be found in this international standard document.

The document concerning the Partial PIXIT Proforma for IUT BS is provided in HTML format with the T3Doc Open Source Tool.

To look at this documentation provided with T3Doc, please refer to the instructions in annex F.

Annex C (normative): WiMAX/HiperMAN 1.3.1 Partial PIXIT proforma for IUT MS

Notwithstanding the provisions of the copyright clause related to the text of the present document, ETSI grants that users of the present document may freely reproduce the Partial PIXIT proforma in this annex so that it can be used for its intended purposes and may further publish the completed Partial PIXIT.
--

The PIXIT Proforma is based on ISO/IEC 9646-6 [7]. Any needed additional information can be found in this international standard document.

The document concerning the Partial PIXIT Proforma for IUT MS is provided in HTML format with the T3Doc Open Source Tool.

To look at this documentation provided with T3Doc, please refer to the instructions in annex F.

Annex D (normative): WiMAX/HiperMAN 1.3.1 PCTR Proforma for IUT BS

Notwithstanding the provisions of the copyright clause related to the text of the present document, ETSI grants that users of the present document may freely reproduce the PCTR proforma in this annex so that it can be used for its intended purposes and may further publish the completed PCTR.

The PCTR proforma is based on ISO/IEC 9646-6 [7]. Any needed additional information can be found in this International standard document.

D.1 Identification summary

D.1.1 Protocol conformance test report

Table D.1

PCTR Number:	
PCTR Date:	
Corresponding SCTR Number:	
Corresponding SCTR Date:	
Test Laboratory Identification:	
Test Laboratory Manager:	
Signature:	

D.1.2 IUT identification

Table D.2

Name:	
Version:	
Protocol specification:	
PICS:	
Previous PCTR if any:	

D.1.3 Testing environment

Table D.3

PIXIT Number:	
ATS Specification:	
Abstract Test Method:	TS 102 545-3 clause 4.
Means of Testing identification:	
Date of testing:	
Conformance Log reference(s):	
Retention Date for Log reference(s):	

D.1.4 Limits and reservation

Additional information relevant to the technical contents or further use of the test report, or the rights and obligations of the test laboratory and the client, may be given here. Such information may include restriction on the publication of the report.

.....

.....

.....

.....

.....

.....

D.1.5 Comments

Additional comments may be given by either the client or the test laboratory on any of the contents of the PCTR, for example, to note disagreement between the two parties.

.....

.....

.....

.....

.....

.....

D.2 IUT Conformance status

This IUT has or has not been shown by conformance assessment to be non-conforming to the specified protocol specification.

Strike the appropriate words in this sentence. If the PICS for this IUT is consistent with the static conformance requirements (as specified in clause C.3 in the present document) and there are no "FAIL" verdicts to be recorded (in clause C.6 in the present document) strike the words "has or", otherwise strike the words "or has not".

D.3 Static conformance summary

The PICS for this IUT is or is not consistent with the static conformance requirements in the specified protocol.

Strike the appropriate words in this sentence.

D.4 Dynamic conformance summary

The test campaign did or did not reveal errors in the IUT.

Strike the appropriate words in this sentence. If there are no "FAIL" verdicts to be recorded (in clause C.6 of the present document) strike the words "did or" otherwise strike the words "or did not".

Summary of the results of groups of test:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

D.5 Static conformance review report

If clause C.3 indicates non-conformance, this clause itemizes the mismatches between the PICS and the static conformance requirements of the specified protocol specification.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

D.6 Test campaign report

Table D.4: BS test cases

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_BS_ARQ_BV_H000	Yes/No	Yes/No		
TC_BS_ARQ_BV_H002	Yes/No	Yes/No		
TC_BS_ARQ_BV_H005	Yes/No	Yes/No		
TC_BS_ARQ_BV_H006	Yes/No	Yes/No		
TC_BS_ARQ_BV_H007	Yes/No	Yes/No		
TC_BS_ARQ_BV_H008	Yes/No	Yes/No		
TC_BS_ARQ_RE_BV_H000	Yes/No	Yes/No		
TC_BS_ARQ_RE_BV_H002	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H001b	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H002	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H003	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H004	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H005	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H007	Yes/No	Yes/No		
TC_BS_ARQ_RXD_BV_H010	Yes/No	Yes/No		
TC_BS_ARQ_RXD_TI_H000	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H000	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H001	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H002	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H003	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H004	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H007	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H010	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H010a	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H011	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H011a	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H013	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H014	Yes/No	Yes/No		
TC_BS_ARQ_TXD_BV_H014a	Yes/No	Yes/No		
TC_BS_ARQ_TXD_TI_H001	Yes/No	Yes/No		
TC_BS_ARQ_TXD_TI_H003	Yes/No	Yes/No		
TC_BS_BWA_BV_H010	Yes/No	Yes/No		
TC_BS_BWA_CBR_BV_H000	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H000	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H001	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H005	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H006	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H007	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H009	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H010	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H011	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H012	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H013	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H014	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H031	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H032y	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H035a	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H036	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H037	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H038	Yes/No	Yes/No		
TC_BS_BWA_REQ_BV_H039	Yes/No	Yes/No		
TC_BS_CDM_CD_BV_H000	Yes/No	Yes/No		
TC_BS_CDM_CD_BV_H003	Yes/No	Yes/No		
TC_BS_CDM_CD_BV_H004	Yes/No	Yes/No		
TC_BS_CDM_CD_BV_H005	Yes/No	Yes/No		
TC_BS_CDM_CD_BV_H006	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_BS_CDM_CD_BV_H007	Yes/No	Yes/No		
TC_BS_CDM_CDC_DL_BV_H000	Yes/No	Yes/No		
TC_BS_CDM_CDC_DL_BV_H001	Yes/No	Yes/No		
TC_BS_CDM_CDC_DL_BV_H002	Yes/No	Yes/No		
TC_BS_CDM_CDC_DL_BV_H004	Yes/No	Yes/No		
TC_BS_CDM_CDC_DL_BV_H005	Yes/No	Yes/No		
TC_BS_CDM_CDC_DL_BV_H007	Yes/No	Yes/No		
TC_BS_CDM_CDC_UL_BV_H000	Yes/No	Yes/No		
TC_BS_CDM_CDC_UL_BV_H001	Yes/No	Yes/No		
TC_BS_CDM_CDC_UL_BV_H002	Yes/No	Yes/No		
TC_BS_CDM_MAP_BV_H000	Yes/No	Yes/No		
TC_BS_CDM_MAP_BV_H005	Yes/No	Yes/No		
TC_BS_CDM_MAP_BV_H006	Yes/No	Yes/No		
TC_BS_CDM_MAP_BV_H007	Yes/No	Yes/No		
TC_BS_CDM_MAP_BV_H008	Yes/No	Yes/No		
TC_BS_CDM_MAP_BV_H009	Yes/No	Yes/No		
TC_BS_CDM_MFS_OPN_BV_H000a	Yes/No	Yes/No		
TC_BS_CDM_MFS_OPN_BV_H001a	Yes/No	Yes/No		
TC_BS_CDM_MFS_OPN_BV_H005	Yes/No	Yes/No		
TC_BS_CSOC_BV_H006	Yes/No	Yes/No		
TC_BS_CSOC_CDS_BV_H005	Yes/No	Yes/No		
TC_BS_CSOC_CDS_BV_H006	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H000	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H001	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H002	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H003	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H004	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H005	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H006	Yes/No	Yes/No		
TC_BS_CSOC_IP4C_BV_H007	Yes/No	Yes/No		
TC_BS_CSOC_IP6C_BV_H001	Yes/No	Yes/No		
TC_BS_CSOC_IP6C_BV_H003	Yes/No	Yes/No		
TC_BS_CSOC_IP6C_BV_H004	Yes/No	Yes/No		
TC_BS_CSOC_IP6C_BV_H005	Yes/No	Yes/No		
TC_BS_CSOC_IP6C_BV_H006	Yes/No	Yes/No		
TC_BS_CSOC_IP6C_BV_H007	Yes/No	Yes/No		
TC_BS_CSOE_IP4_BV_H000	Yes/No	Yes/No		
TC_BS_CSOE_IP4_BV_H001	Yes/No	Yes/No		
TC_BS_CSOE_IP6_BV_H000	Yes/No	Yes/No		
TC_BS_CSOE_IP6_BV_H001	Yes/No	Yes/No		
TC_BS_DS_DSA_BI_H001	Yes/No	Yes/No		
TC_BS_DS_DSA_BI_H002	Yes/No	Yes/No		
TC_BS_DS_DSA_BO_H003	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H000	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H001	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H002	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H004	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H006	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H007	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H010	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H011	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H014	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H015	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H017	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H019	Yes/No	Yes/No		
TC_BS_DS_DSA_BV_H020	Yes/No	Yes/No		
TC_BS_DS_DSA_TI_H000	Yes/No	Yes/No		
TC_BS_DS_DSA_TI_H002	Yes/No	Yes/No		
TC_BS_DS_DSA_TI_H003	Yes/No	Yes/No		
TC_BS_DS_DSA_TI_H006	Yes/No	Yes/No		
TC_BS_DS_DSA_TI_H008	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_BS_DS_DSA_TI_H014	Yes/No	Yes/No		
TC_BS_DS_DSC_BI_H001	Yes/No	Yes/No		
TC_BS_DS_DSC_BO_H018	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H002	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H003	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H006	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H008	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H009	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H012	Yes/No	Yes/No		
TC_BS_DS_DSC_BV_H018	Yes/No	Yes/No		
TC_BS_DS_DSC_TI_H000	Yes/No	Yes/No		
TC_BS_DS_DSC_TI_H002	Yes/No	Yes/No		
TC_BS_DS_DSD_BV_H000	Yes/No	Yes/No		
TC_BS_DS_DSD_BV_H001	Yes/No	Yes/No		
TC_BS_DS_DSD_BV_H003	Yes/No	Yes/No		
TC_BS_DS_DSD_BV_H004	Yes/No	Yes/No		
TC_BS_DS_DSD_TI_H004	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H002	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H003	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H018	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H020	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H021	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H022	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H023	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H024	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H025	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H026	Yes/No	Yes/No		
TC_BS_DS_QPS_BV_H032	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_CCL_BV_H001	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_INI_BS_BV_H000	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_INI_BS_BV_H001	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_INI_BS_BV_H002	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_INI_MS_BV_H000	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_INI_MS_BV_H001	Yes/No	Yes/No		
TC_BS_GHF_HO_SVG_INI_MS_BV_H003	Yes/No	Yes/No		
TC_BS_GHF_NTA_NWA_BV_H000	Yes/No	Yes/No		
TC_BS_GHF_NTA_SCAN_BV_H000	Yes/No	Yes/No		
TC_BS_GHF_TGT_NWR_BV_H000	Yes/No	Yes/No		
TC_BS_GHF_TGT_NWR_BV_H001	Yes/No	Yes/No		
TC_BS_GHF_TGT_NWR_BV_H002	Yes/No	Yes/No		
TC_BS_GHF_TGT_NWR_BV_H004	Yes/No	Yes/No		
TC_BS_GHF_TGT_NWR_BV_H006	Yes/No	Yes/No		
TC_BS_GHF_TGT_NWR_BV_H010	Yes/No	Yes/No		
TC_BS_GHF_TGT_SAR_BV_H000	Yes/No	Yes/No		
TC_BS_IDM_LOC_BV_H000	Yes/No	Yes/No		
TC_BS_IDM_LOC_BV_H001	Yes/No	Yes/No		
TC_BS_IDM_NWR_BV_H000	Yes/No	Yes/No		
TC_BS_IDM_NWR_BV_H003	Yes/No	Yes/No		
TC_BS_IDM_NWR_BV_H005	Yes/No	Yes/No		
TC_BS_IDM_NWR_BV_H009	Yes/No	Yes/No		
TC_BS_IDM_NWR_BV_H013	Yes/No	Yes/No		
TC_BS_IDM_PG_BV_H000	Yes/No	Yes/No		
TC_BS_IDM_PG_BV_H005	Yes/No	Yes/No		
TC_BS_IDM_PWD_BV_H000	Yes/No	Yes/No		
TC_BS_IDM_TIDM_BV_H000	Yes/No	Yes/No		
TC_BS_IDM_TIDM_BV_H001	Yes/No	Yes/No		
TC_BS_IDM_TIDM_BV_H002	Yes/No	Yes/No		
TC_BS_IDM_TIDM_TI_H000	Yes/No	Yes/No		
TC_BS_IDM_TIDM_TI_H001	Yes/No	Yes/No		
TC_BS_INI_REG_BI_H000	Yes/No	Yes/No		
TC_BS_INI_REG_BO_H001	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_BS_INI_REG_BV_H000	Yes/No	Yes/No		
TC_BS_INI_REG_BV_H003	Yes/No	Yes/No		
TC_BS_INI_REG_BV_H004	Yes/No	Yes/No		
TC_BS_MAC_BV_H000	Yes/No	Yes/No		
TC_BS_MAC_BV_H000a	Yes/No	Yes/No		
TC_BS_MAC_BV_H001	Yes/No	Yes/No		
TC_BS_MAC_BV_H001b	Yes/No	Yes/No		
TC_BS_MAC_BV_H002	Yes/No	Yes/No		
TC_BS_MAC_BV_H003	Yes/No	Yes/No		
TC_BS_MAC_BV_H004	Yes/No	Yes/No		
TC_BS_MAC_BV_H009	Yes/No	Yes/No		
TC_BS_MAC_BV_H010	Yes/No	Yes/No		
TC_BS_MAC_BV_H011	Yes/No	Yes/No		
TC_BS_MAC_CAT_BV_H000	Yes/No	Yes/No		
TC_BS_MAC_CAT_BV_H001	Yes/No	Yes/No		
TC_BS_MAC_CRC_BV_H000	Yes/No	Yes/No		
TC_BS_MAC_CRC_BV_H001	Yes/No	Yes/No		
TC_BS_MAC_CRC_BV_H002	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H005	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H006	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H007	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H008	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H010	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H011	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H012	Yes/No	Yes/No		
TC_BS_MAC_FRAG_BV_H013	Yes/No	Yes/No		
TC_BS_MAC_PACK_BV_H000	Yes/No	Yes/No		
TC_BS_MAC_PACK_BV_H005	Yes/No	Yes/No		
TC_BS_MAC_PACK_BV_H006	Yes/No	Yes/No		
TC_BS_MBS_BV_H002	Yes/No	Yes/No		
TC_BS_MBS_BV_H006	Yes/No	Yes/No		
TC_BS_MBS_BV_H007	Yes/No	Yes/No		
TC_BS_MBS_BV_H008	Yes/No	Yes/No		
TC_BS_MBS_BV_H009	Yes/No	Yes/No		
TC_BS_PHS_BV_H000	Yes/No	Yes/No		
TC_BS_PHS_BV_H003y	Yes/No	Yes/No		
TC_BS_PHS_BV_H003z	Yes/No	Yes/No		
TC_BS_PHS_BV_H004y	Yes/No	Yes/No		
TC_BS_PHS_BV_H004z	Yes/No	Yes/No		
TC_BS_PHS_BV_H008y	Yes/No	Yes/No		
TC_BS_PHS_BV_H008z	Yes/No	Yes/No		
TC_BS_PHS_BV_H009	Yes/No	Yes/No		
TC_BS_PHS_BV_H011y	Yes/No	Yes/No		
TC_BS_PHS_BV_H011z	Yes/No	Yes/No		
TC_BS_RER_BV_H006	Yes/No	Yes/No		
TC_BS_RLC_ACQ_BV_H000	Yes/No	Yes/No		
TC_BS_RLC_FBK_BV_H003	Yes/No	Yes/No		
TC_BS_RLC_HARQ_BV_H000B	Yes/No	Yes/No		
TC_BS_RLC_HARQ_BV_H000C	Yes/No	Yes/No		
TC_BS_RLC_HARQ_BV_H000D	Yes/No	Yes/No		
TC_BS_RLC_HARQ_BV_H003	Yes/No	Yes/No		
TC_BS_RLC_HARQ_BV_H008	Yes/No	Yes/No		
TC_BS_RLC_HARQ_BV_H009	Yes/No	Yes/No		
TC_BS_RLC_IRNG_BI_H000	Yes/No	Yes/No		
TC_BS_RLC_IRNG_BV_H015	Yes/No	Yes/No		
TC_BS_RLC_IRNG_BV_H016	Yes/No	Yes/No		
TC_BS_RLC_IRNG_BV_H017	Yes/No	Yes/No		
TC_BS_RLC_IRNG_TI_H003	Yes/No	Yes/No		
TC_BS_RLC_PRNG_BV_H013	Yes/No	Yes/No		
TC_BS_RLC_PRNG_BV_H014	Yes/No	Yes/No		
TC_BS_RLC_SBC_BV_H000	Yes/No	Yes/No		

Annex E (normative): WiMAX/HiperMAN 1.3.1 PCTR Proforma for IUT MS

Notwithstanding the provisions of the copyright clause related to the text of the present document, ETSI grants that users of the present document may freely reproduce the PCTR proforma in this annex so that it can be used for its intended purposes and may further publish the completed PCTR.

The PCTR proforma is based on ISO/IEC 9646-6 [7]. Any needed additional information can be found in this International standard document.

E.1 Identification summary

E.1.1 Protocol conformance test report

Table E.1

PCTR Number:	
PCTR Date:	
Corresponding SCTR Number:	
Corresponding SCTR Date:	
Test Laboratory Identification:	
Test Laboratory Manager:	
Signature:	

E.1.2 IUT identification

Table E.2

Name:	
Version:	
Protocol specification:	
PICS:	
Previous PCTR if any:	

E.1.3 Testing environment

Table E.3

PIXIT Number:	
ATS Specification:	
Abstract Test Method:	TS 102 545-3, clause 4.
Means of Testing identification:	
Date of testing:	
Conformance Log reference(s):	
Retention Date for Log reference(s):	

E.1.4 Limits and reservation

Additional information relevant to the technical contents or further use of the test report, or the rights and obligations of the test laboratory and the client, may be given here. Such information may include restriction on the publication of the report.

.....

.....

.....

.....

.....

E.1.5 Comments

Additional comments may be given by either the client or the test laboratory on any of the contents of the PCTR, for example, to note disagreement between the two parties.

.....

.....

.....

.....

.....

E.2 IUT Conformance status

This IUT has or has not been shown by conformance assessment to be non-conforming to the specified protocol specification.

Strike the appropriate words in this sentence. If the PICS for this IUT is consistent with the static conformance requirements (as specified in clause C.3 in the present document) and there are no "FAIL" verdicts to be recorded (in clause C.6 in the present document) strike the words "has or", otherwise strike the words "or has not".

E.3 Static conformance summary

The PICS for this IUT is or is not consistent with the static conformance requirements in the specified protocol.

Strike the appropriate words in this sentence.

E.4 Dynamic conformance summary

The test campaign did or did not reveal errors in the IUT.

Strike the appropriate words in this sentence. If there are no "FAIL" verdicts to be recorded (in clause C.6 of the present document) strike the words "did or" otherwise strike the words "or did not".

Summary of the results of groups of test:

.....

.....

.....

.....

.....

.....

.....

E.5 Static conformance review report

If clause C.3 indicates non-conformance, this clause itemizes the mismatches between the PICS and the static conformance requirements of the specified protocol specification.

.....

.....

.....

.....

.....

.....

E.6 Test campaign report

Table E.4: MS test cases

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_MS_GHF_HO_CCL_BV_H000	Yes/No	Yes/No		
TC_MS_GHF_HO_CCL_BV_H001	Yes/No	Yes/No		
TC_MS_GHF_HO_CCL_BV_H002	Yes/No	Yes/No		
TC_MS_GHF_HO_CCL_BV_H003	Yes/No	Yes/No		
TC_MS_GHF_HO_INI_BS_BV_H000	Yes/No	Yes/No		
TC_MS_GHF_HO_INI_BS_BV_H003	Yes/No	Yes/No		
TC_MS_GHF_HO_INI_MS_BV_H000	Yes/No	Yes/No		
TC_MS_GHF_HO_INI_MS_BV_H001	Yes/No	Yes/No		
TC_MS_GHF_HO_INI_MS_BV_H002	Yes/No	Yes/No		
TC_MS_GHF_HO_TER_BV_H000	Yes/No	Yes/No		
TC_MS_GHF_NTA_SCAN_BV_H000	Yes/No	Yes/No		
TC_MS_GHF_NTA_SCAN_BV_H001	Yes/No	Yes/No		
TC_MS_GHF_NTA_SCAN_BV_H001a	Yes/No	Yes/No		
TC_MS_GHF_NTA_SCAN_BV_H004	Yes/No	Yes/No		
TC_MS_GHF_NWR_BV_H000	Yes/No	Yes/No		
TC_MS_GHF_NWR_BV_H002	Yes/No	Yes/No		
TC_MS_GHF_NWR_BV_H010	Yes/No	Yes/No		
TC_MS_GHF_NWR_BV_H011	Yes/No	Yes/No		
TC_MS_IDM_LOC_BV_H000	Yes/No	Yes/No		
TC_MS_IDM_LOC_BV_H001	Yes/No	Yes/No		
TC_MS_IDM_LOC_BV_H002	Yes/No	Yes/No		
TC_MS_IDM_NWRI_BV_H002	Yes/No	Yes/No		
TC_MS_IDM_NWRI_BV_H008	Yes/No	Yes/No		
TC_MS_IDM_PG_BV_H000	Yes/No	Yes/No		
TC_MS_IDM_PG_BV_H001	Yes/No	Yes/No		
TC_MS_IDM_PWD_BV_H000	Yes/No	Yes/No		
TC_MS_IDM_TIDM_BV_H000	Yes/No	Yes/No		
TC_MS_IDM_TIDM_BV_H001	Yes/No	Yes/No		
TC_MS_IDM_TIDM_BV_H002	Yes/No	Yes/No		
TC_MS_IDM_TIDM_TI_H000	Yes/No	Yes/No		
TC_MS_IDM_TIDM_TI_H002	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H003	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H003a	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H003b	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H007	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H008	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H009	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H010	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H011	Yes/No	Yes/No		
TC_MS_SLM_PW1_BV_H014	Yes/No	Yes/No		
TC_MS_SLM_PW1_TI_H001	Yes/No	Yes/No		
TC_MS_SLM_PW1_TI_H001a	Yes/No	Yes/No		
TC_SS_ARQ_BV_H000	Yes/No	Yes/No		
TC_SS_ARQ_BV_H001	Yes/No	Yes/No		
TC_SS_ARQ_BV_H002	Yes/No	Yes/No		
TC_SS_ARQ_BV_H004	Yes/No	Yes/No		
TC_SS_ARQ_BV_H006	Yes/No	Yes/No		
TC_SS_ARQ_BV_H007	Yes/No	Yes/No		
TC_SS_ARQ_RE_BV_H000	Yes/No	Yes/No		
TC_SS_ARQ_RE_BV_H003	Yes/No	Yes/No		
TC_SS_ARQ_RE_TI_H000	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H001	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H002	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H003	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H004	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H005	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_SS_ARQ_RXD_BV_H006	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H008	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H012	Yes/No	Yes/No		
TC_SS_ARQ_RXD_BV_H013	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H000	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H001	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H003	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H004	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H005	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H006	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H007	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H007a	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H007b	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H011	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H012	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H012a	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H012b	Yes/No	Yes/No		
TC_SS_ARQ_TXD_BV_H014	Yes/No	Yes/No		
TC_SS_ARQ_TXD_TI_H001	Yes/No	Yes/No		
TC_SS_ARQ_TXD_TI_H002	Yes/No	Yes/No		
TC_SS_BWA_CBR_BV_H000	Yes/No	Yes/No		
TC_SS_BWA_CBR_BV_H001	Yes/No	Yes/No		
TC_SS_BWA_CBR_BV_H002	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H001	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H002	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H008	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H009	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H015	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H016	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H017	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H018	Yes/No	Yes/No		
TC_SS_BWA_REQ_BV_H021	Yes/No	Yes/No		
TC_SS_CDM_CD_BV_H001a	Yes/No	Yes/No		
TC_SS_CDM_CDC_DL_BV_H000	Yes/No	Yes/No		
TC_SS_CDM_CDC_DL_BV_H002	Yes/No	Yes/No		
TC_SS_CDM_CDC_UL_BV_H000	Yes/No	Yes/No		
TC_SS_CDM_CDC_UL_BV_H003	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H001	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H002	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H003	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H004	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H007	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H008	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H009	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H010	Yes/No	Yes/No		
TC_SS_CDM_MAP_BV_H011	Yes/No	Yes/No		
TC_SS_CDM_MFS_OPN_BV_H002	Yes/No	Yes/No		
TC_SS_CDM_MFS_RLV_BV_H000	Yes/No	Yes/No		
TC_SS_CSOC_CDS_BV_H000	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H000	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H001	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H002	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H003	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H004	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H005	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H006	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H007	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H008	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_IP4oE_BV_H009	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_PETC_BV_H000	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_PETC_BV_H001	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_SS_CSOC_ENTC_PETC_BV_H002	Yes/No	Yes/No		
TC_SS_CSOC_ENTC_PETC_BV_H003	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H000	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H001	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H003	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H004	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H005	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H006	Yes/No	Yes/No		
TC_SS_CSOC_IP4C_BV_H007	Yes/No	Yes/No		
TC_SS_CSOC_IP6C_BV_H001	Yes/No	Yes/No		
TC_SS_CSOC_IP6C_BV_H003	Yes/No	Yes/No		
TC_SS_CSOC_IP6C_BV_H004	Yes/No	Yes/No		
TC_SS_CSOC_IP6C_BV_H005	Yes/No	Yes/No		
TC_SS_CSOC_IP6C_BV_H006	Yes/No	Yes/No		
TC_SS_CSOC_IP6C_BV_H007	Yes/No	Yes/No		
TC_SS_CSOE_IP4_BV_H000	Yes/No	Yes/No		
TC_SS_CSOE_IP4_BV_H001	Yes/No	Yes/No		
TC_SS_CSOE_IP6_BV_H000	Yes/No	Yes/No		
TC_SS_CSOE_IP6_BV_H001	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H000	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H001	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H002	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H003	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H004	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H005	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H006	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H007	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H008	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H009	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H010	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H011	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H012	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H013	Yes/No	Yes/No		
TC_SS_DS_DSA_BV_H015	Yes/No	Yes/No		
TC_SS_DS_DSA_TI_H003	Yes/No	Yes/No		
TC_SS_DS_DSA_TI_H005	Yes/No	Yes/No		
TC_SS_DS_DSA_TI_H010	Yes/No	Yes/No		
TC_SS_DS_DSA_TI_H014	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H000	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H001	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H002	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H003	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H010	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H011	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H013	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H017	Yes/No	Yes/No		
TC_SS_DS_DSC_BV_H020	Yes/No	Yes/No		
TC_SS_DS_DSC_TI_H007	Yes/No	Yes/No		
TC_SS_DS_DSD_BV_H000	Yes/No	Yes/No		
TC_SS_DS_DSD_BV_H001	Yes/No	Yes/No		
TC_SS_DS_DSD_BV_H002	Yes/No	Yes/No		
TC_SS_DS_DSD_BV_H003	Yes/No	Yes/No		
TC_SS_DS_DSD_BV_H004	Yes/No	Yes/No		
TC_SS_DS_DSD_TI_H001	Yes/No	Yes/No		
TC_SS_DS_DSD_TI_H003	Yes/No	Yes/No		
TC_SS_DS_DSD_TI_H004	Yes/No	Yes/No		
TC_SS_DS_QPS_BV_H001	Yes/No	Yes/No		
TC_SS_INI_REG_BI_H000	Yes/No	Yes/No		
TC_SS_INI_REG_BV_H001	Yes/No	Yes/No		
TC_SS_INI_REG_TI_H001	Yes/No	Yes/No		
TC_SS_MAC_BV_H000	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_SS_MAC_BV_H001	Yes/No	Yes/No		
TC_SS_MAC_BV_H005	Yes/No	Yes/No		
TC_SS_MAC_BV_H006	Yes/No	Yes/No		
TC_SS_MAC_BV_H007	Yes/No	Yes/No		
TC_SS_MAC_BV_H008	Yes/No	Yes/No		
TC_SS_MAC_BV_H010	Yes/No	Yes/No		
TC_SS_MAC_BV_H011	Yes/No	Yes/No		
TC_SS_MAC_CAT_BV_H000	Yes/No	Yes/No		
TC_SS_MAC_CAT_BV_H001	Yes/No	Yes/No		
TC_SS_MAC_CRC_BV_H001	Yes/No	Yes/No		
TC_SS_MAC_CRC_BV_H002	Yes/No	Yes/No		
TC_SS_MAC_CRC_BV_H004	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H000	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H003	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H005	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H006	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H007	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H008	Yes/No	Yes/No		
TC_SS_MAC_FRAG_BV_H009	Yes/No	Yes/No		
TC_SS_MAC_PACK_BV_H000	Yes/No	Yes/No		
TC_SS_MAC_PACK_BV_H001	Yes/No	Yes/No		
TC_SS_MAC_PACK_BV_H002	Yes/No	Yes/No		
TC_SS_MAC_PACK_BV_H003	Yes/No	Yes/No		
TC_SS_MBS_BV_H001	Yes/No	Yes/No		
TC_SS_MBS_BV_H002	Yes/No	Yes/No		
TC_SS_MBS_BV_H003	Yes/No	Yes/No		
TC_SS_PHS_BV_H000x	Yes/No	Yes/No		
TC_SS_PHS_BV_H003a	Yes/No	Yes/No		
TC_SS_PHS_BV_H004x	Yes/No	Yes/No		
TC_SS_PHS_BV_H007	Yes/No	Yes/No		
TC_SS_PHS_BV_H008	Yes/No	Yes/No		
TC_SS_PHS_BV_H009y	Yes/No	Yes/No		
TC_SS_PHS_BV_H009z	Yes/No	Yes/No		
TC_SS_PHS_BV_H010y	Yes/No	Yes/No		
TC_SS_PHS_BV_H010z	Yes/No	Yes/No		
TC_SS_RER_BV_H001	Yes/No	Yes/No		
TC_SS_RER_BV_H007	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H000	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H003	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H004	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H005	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H006	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H007	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H008	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H009	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H011	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H012	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H014	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H015	Yes/No	Yes/No		
TC_SS_RLC_FBK_BV_H018	Yes/No	Yes/No		
TC_SS_RLC_HARQ_BV_H002b	Yes/No	Yes/No		
TC_SS_RLC_HARQ_BV_H002e	Yes/No	Yes/No		
TC_SS_RLC_HARQ_BV_H006	Yes/No	Yes/No		
TC_SS_RLC_HARQ_BV_H007	Yes/No	Yes/No		
TC_SS_RLC_HARQ_BV_H009	Yes/No	Yes/No		
TC_SS_RLC_HARQ_BV_H010	Yes/No	Yes/No		
TC_SS_RLC_IRNG_BV_H015	Yes/No	Yes/No		
TC_SS_RLC_IRNG_BV_H016	Yes/No	Yes/No		
TC_SS_RLC_IRNG_BV_H017	Yes/No	Yes/No		
TC_SS_RLC_IRNG_BV_H018	Yes/No	Yes/No		
TC_SS_RLC_IRNG_BV_H019	Yes/No	Yes/No		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause E.7)
TC_SS_RLC_IRNG_BV_H019a	Yes/No	Yes/No		
TC_SS_RLC_IRNG_BV_H025	Yes/No	Yes/No		
TC_SS_RLC_IRNG_TI_H005	Yes/No	Yes/No		
TC_SS_RLC_PRNG_BV_H017	Yes/No	Yes/No		
TC_SS_RLC_PRNG_BV_H043	Yes/No	Yes/No		
TC_SS_RLC_PRNG_BV_H045	Yes/No	Yes/No		
TC_SS_RLC_PRNG_BV_H046	Yes/No	Yes/No		
TC_SS_RLC_PRNG_BV_H047	Yes/No	Yes/No		
TC_SS_RLC_PRNG_TI_H001	Yes/No	Yes/No		
TC_SS_RLC_SBC_BV_H000a	Yes/No	Yes/No		
TC_SS_RLC_SBC_BV_H001	Yes/No	Yes/No		
TC_SS_RLC_SBC_BV_H001a	Yes/No	Yes/No		
TC_SS_RLC_SBC_TI_H000	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H004	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H009	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H011	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H016	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H018	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H020	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H025	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_EAP_FSM_BV_H027	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_HAN_BV_H000	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_HAN_BV_H001	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_HAN_BV_H003	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_IDM_BV_H000	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_IDM_BV_H001	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_IDM_BV_H003	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_NWE_BV_H003	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_NWE_BV_H008	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_NWE_BV_H009	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_REA_BV_H003	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_REA_BV_H008	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_AUTH_REA_BV_H009	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H000	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H001	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H002	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H003	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H004	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H005	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H006	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H007	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H008	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H009	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H010	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H016	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H017	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H018	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H020	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H021	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H022	Yes/No	Yes/No		
TC_SS_SEC_PKMv2_TEK_FSM_BV_H023	Yes/No	Yes/No		

Annex F (normative): HTML documentation

An additional documentation in HTML is also available to extend the current ATS documentation. This HTML documentation can be viewed using a regular web browser. This HTML documentation contains structured information, which provide details on TTCN definitions for the following modules:

- Test Configuration.
- PICs.
- PIXITs.
- External functions.
- TA (Test Adapter) Command and Responses.
- All test Cases Modules.

The HTML files corresponding to the TTCN documentation are contained in an archive named `ts_10254503v010201p0.zip` which accompanies the present document.

To look at this HTML documentation you need:

- 1) to unpack the zip file in any empty directory;
- 2) to start browsing the files with the "index.htm" file;
- 3) to follow the different links to reach the desired item;
- 4) Comment and description of the items is provided at different levels of the html files.

Annex G (informative): Bibliography

- IETF RFC 2131: "Dynamic Host Configuration Protocol".
- IETF RFC 868: "Time Protocol".
- IETF RFC 1123: "Requirements for Internet Hosts - Application and Support".
- IETF RFC 2349: "TFTP Timeout Interval and Transfer Size Options".

History

Document history		
V1.1.1	September 2007	Publication
V1.2.1	February 2009	Publication