ETSI TS 102 624-3 V1.2.1 (2009-11)

Technical Specification

Broadband Radio Access Networks (BRAN); HiperMAN; Conformance Testing for the Network layer of HiperMAN/WiMAX terminal devices; Part 3: Abstract Test Suite (ATS)

Reference

RTS/BRAN-004T010-3

Keywords

ATS, HiperMAN, layer 3, terminal, 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

Copyright Notification

Contents

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 HiperMAN; Conformance Testing for Network layer of the WiMAX/HiperMAN terminal devices, 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 HiperMAN/WiMAX terminal devices for conformance across WiMAX networks.

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

The ISO standard for the methodology of conformance testing (ISO/IEC 9646-1 [39] and ISO/IEC 9646-2 [40]) as well as the ETSI rules for conformance testing (ETS 300 406 [35] 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 MS 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.

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] WiMAX Forum (Release 1.5): "WiMAX Forum Network Architecture, Stage 2: Architecture Tenets, Reference Model and Reference Points, Base Specification".

[2] WiMAX Forum (Release 1.5): "WiMAX Forum Network Architecture, Stage 3: Detailed Protocols and Procedures, Base Specification".

[3] Void.

[4] ETSI TS 102 624-2: "Broadband Radio Access Networks (BRAN); HiperMAN; onformance Testing for the Network layer of HiperMAN/WiMAX terminal devices; Part 2: Test Suite Structure and Test Purposes (TSS&TP)".

[5] ETSI TS 102 545-3: "Broadband Radio Access Networks (BRAN); HiperMAN; Conformance Testing for WiMAX/HiperMAN 1.3.1 Part 3: Abstract Test Suite (ATS)".

[6] 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".

[7] IEEE 802.16g-2007: "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 3: Management Plane Procedures and Services".

NOTE: Available at http://standards.ieee.org/getieee802/802.16.html.

[8] IETF RFC 768 (August 1980): "User Datagram Protocol".

[9] IETF RFC 791 (September 1981): "Internet Protocol".

[10] IETF RFC 792 (September 1981): "Internet Control Message Protocol".

[11] Void.

[12] IETF RFC 1112 (August 1989): "Host Extensions for IP Multicasting".

[13] IETF RFC 1256 (September 1991): "ICMP Router Discovery Messages ".

[14] IETF RFC 2131 (March 1997): "Dynamic Host Configuration Protocol".

[15] IETF RFC 2132 (March 1997): "DHCP Options and BOOTP Vendor Extensions".

[16] IETF RFC 2460 (December 1998): "Internet Protocol, Version 6 (IPv6) Specification".

[17] IETF RFC 2794 (March 2000): "Mobile IP Network Access Identifier Extension for IPv4".

[18] IETF RFC 3315 (July 2003): "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)".

[19] IETF RFC 3344 (August 2002): "IP Mobility Support for IPv4".

[20] IETF RFC 3543 (August 2003): "Registration Revocation in Mobile IPv4".

[21] IETF RFC 3748 (June 2004): "Extensible Authentication Protocol (EAP)".

[22] IETF RFC 3775 (June 2004): "Mobility Support in IPv6".

[23] IETF RFC 3846 (June 2004): "Mobile IPv4 Extension for Carrying Network Access Identifiers".

[24] IETF RFC 4187 (January 2006): "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)".

[25] IETF RFC 4283 (November 2005): "Mobile Node Identifier Option for Mobile IPv6 (MIPv6)".

[26] IETF RFC 4285 (January 2006): "Authentication Protocol for Mobile IPv6".

[27] IETF RFC 4433 (March 2006): "Mobile IPv4 Dynamic Home Agent (HA) Assignment".

[28] Void.

[29] IETF RFC 4861: "Neighbor Discovery for IP version 6 (IPv6)".

[30] IETF RFC 4862: "IPv6 Stateless Address Autoconfiguration".

[31] IETF RFC 5216: "The EAP-TLS Authentication Protocol".

[32] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[33] IETF RFC 5281 (August 2008): "Extensible Authentication Protocol Tunnelled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)".

[34] IETF draft-ietf-mip6-hiopt-17.txt: "DHCP Option for Home Information Discovery in MIPv6".

[35] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[36] 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".

[37] Void.

[38] ETSI ES 201 873-6: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI)".

[39] ISO/IEC 9646-1 (1994): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts". (See also ITU-T Recommendation X.290 (1991).

[40] ISO/IEC 9646-2 (1994): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification". (See also ITU-T Recommendation X.291 (1991).

[41] ISO/IEC 9646-6 (1994): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 6: Protocol profile test specification".

[42] ISO/IEC 9646-7 (1995): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statement".

[43] IETF RFC 761: "DoD standard Transmission Control Protocol".

[44] IETF RFC 3486: "Compressing the Session Initiation Protocol (SIP)".

[45] IETF RFC 3957: "Authentication, Authorization, and Accounting (AAA) Registration Keys for Mobile IPv4".

[46] Void.

[47] Void.

[48] Void.

[49] Void.

[50] IETF RFC 4286: "Multicast Router Discovery".

[51] IETF RFC 4721: "Mobile IPv4 Challenge/Response Extensions (Revised)".

[52] IETF RFC 4857: "Mobile IPv4 Regional Registration".

[53] IETF RFC 4988: "Mobile IPv4 Fast Handovers".

[54] IETF RFC 2782: "A DNS RR for specifying the location of services (DNS SRV)".

[55] IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1".

NOTE: Available at: http://www.ietf.org/rfc/rfc2616.txt.

[56] Void.

[57] Void.

[58] Void.

[59] Open Mobile Alliance WAP-235-PushOTA-20010425-a: "Push OTA Protocol".

NOTE: Available at: http://www.openmobilealliance.org.

[60] Open Mobile Alliance WAP-251-PushMessage-20010322-a: "Push Message".

NOTE: Available at: http://www.openmobilealliance.org.

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.

[i.1] ETSI TS 102 624-1: "Broadband Radio Access Networks (BRAN); HiperMAN; Conformance Testing for the Network layer of HiperMAN/WiMAX terminal devices; Part 1: Protocol Implementation Conformance Statement (PICS) proforma".

[i.2] ISO/IEC 9646 (all parts): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ISO/IEC 9646-7 [42], TS 102 545-3 [5], IEEE 802.16e-2005 [6] and IEEE 802.16g-2007 [7], [1] and [2] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 102 545-3 [5], ISO/IEC 9646-1 [39], ISO/IEC 9646-6 [41], ISO/IEC 9646-7 [42], IEEE 802.16e-2005 [6], IEEE 802.16g-2007 [7], [1], [2] and the following apply:

AKA	Authentication and Key Agreement
ATS	Abstract Test Suite
AVP	Attribute Value Pair
BS	Base Station
CID	Connection IDentifier
CMAC	Cipher-based MAC
CS	Convergence Sublayer
DHCP	Dynamic Host Configuration Protocol
DIUC	Downlink Interval Usage Code
DLC	Data Link Control
EAP	Extensible Authentication Protocol
ICMP	Internet Control Message Protocol
IGMP	Internet Group Membership Protocol
IP	Internet Protocol
IUT	Implementation Under Test
MIP	Mobile IP
NAP	Network Access Provider
NCT	Network Conformance Tests
NSP	Network Service Provider
NWE	Network Entry
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open Systems Interconnection
PA	Platform Adapter
PCO	Point of Control and Observation
PCT	Protocol Conformance Tests
PCTR	Protocol Conformance Test Report
PHY	Physical layer
PIXIT	Partial Protocol Implementation Extra Information for Testing
REQ	REQuest
RSP	ReSPonse
SA	SUT Adapter
SS	Subscriber Station

SUT	System Under Test
TA	Test Adapter
TC	Test Case
TLS	Transport Layer Security
TLV	Type, Length, Value
TP	Test Purposes
TTCN	Test and Test Control Notation
TTLS	Tunnelled TLS
UIUC	Uplink Interval Usage Code

4 Abstract Test Method (ATM)

This clause describes the ATM used to test the Network layer of the HiperMAN/WiMAX terminal devices.

NOTE: In the present document the normative terms SUT Adapter (SA), Platform Adapter (PA) and CODECS as defined by TTCN-3 standards [36] are used. The non-normative term Test Adaptor (TA) is also used for backward compatibility with TS 102 545-3 [5] and embodies SA, PA and CODECS concepts.

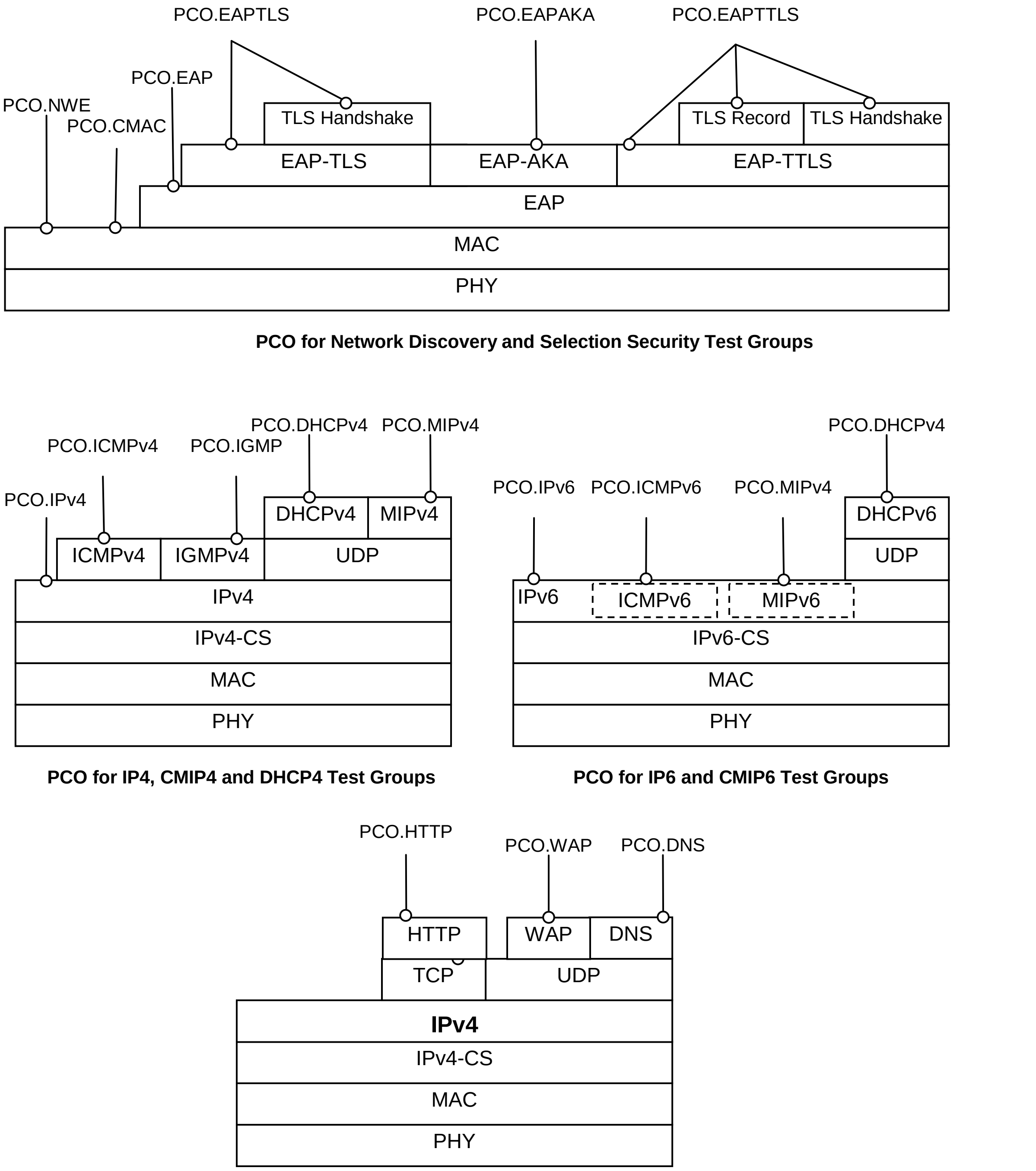
4.1 Test architecture

4.1.1 Points of Observation and Control

Testing BRAN HiperMAN/WiMAX terminal devices for conformance across WiMAX networks involves the participation of many procedures and protocols. In order to delimit the scope of NCT, the identification of Points of Observation and Control (i.e. PCO as defined in ISO/IEC 9646-1 [39]) become of utmost importance.

After an exhaustive analysis of the NWG base standards WiMAX Forum Network Architecture; Stage 2 [1], WiMAX Forum Network Architecture; Stage 3 [2] and TS 102 624-2 [1], the PCO for "Conformance Testing for the Network layer of the HiperMAN/WiMAX terminal devices" are identified.

PCO are depicted in figure 1 on an OSI model representation of the WiMAX terminal device in all the scenarios considered in the present document.



PCO for Network Discovery and Selection Security Test Groups

PCO for IP4, CMIP4 and DHCP4 Test Groups

PCO for IP6 and CMIP6 Test Groups

PCO for OTA Test Groups

Figure 1: Points of Observation and Control

Each PCO identified above is specified in IETF standards (with the exception of PCO.NWE and PCO.CMAC).

Since protocols usually have extensions and options, several standards may be needed to specify one single protocol in its completeness. However, the scope of the present document is limited to the IETF standards applicable in NWG base standards. The list of standards which specifies the protocols involved in the network functionalities to be tested in accordance with TS 102 624-2 [4] is given in table 1.

Table 1: IEEE and IETF standards for NCT PCOs

PCO	Standards
PCO.NWE	IEEE P802.16g/ D9, April 2007 [7]
PCO.CMAC	IEEE 802.16e-2005 [6]
PCO.MIPv4	RFC 3344 [19], RFC 3543 [20], RFC 3846 [23] and RFC 4433 [27]
PCO.MIPv6	RFC 3775 [22], RFC 4283 [25] and RFC 4285 [26]
PCO.IPv6	RFC 2460 [16]
PCO.DHCPv4	RFC 2131 [14] and RFC 2132 [15]
PCO.DHCPv6	RFC 3315 [18] and IETF draft draft_ietf_mip6_hiop17.txt [34].
PCO.ICMPv4	RFC 792 [10] and RFC 1256 [13].
PCO.ICMPv6	RFC 4861 [29] and RFC 4862 [30]
PCO.IGMP	RFC 1112 [12]
PCO.EAP	RFC 3748 [21]
PCO.EAPTLS	RFC 5246 [32] and RFC 5216 [31]
PCO.EAPTTLS	RFC 5281 [33]
PCO.AKA	RFC 4187 [24]
PCO.HTTP	RFC 2616 [55]
PCO.WAP	?? OMA DM Protocol
PCO.DNS	RFC 2782 [54]

New PCO may be identified as new network functionalities become target of NCT testing in future releases of TS 102 624-2 [4]. Likewise, new IETF standards may become within the scope of NCT testing.

The present document also deals with the processing of protocols, messages and fields which currently remains outside the scope of NCT testing. See details in clauses 6.1.3 and 6.1.4.

4.1.2 Architectural Requirements

Based on the PCO identification, and considering compatibility with existing standards a desirable system property, the following list of architectural requirements is proposed. These requirements are to be accomplished in the definition of the Abstract Test Method.

4.1.2.1 Reuse of DLC-TTCN

Table 2: TTCN-3 Test Suite Requirements

ID	Description
REQ-1-ATS	NCT TTCN-3 test suite shall use the existing DLC-TTCN (TS 102 545-3 [5]) in order to operate the DLC layer when required by NCT test cases (e.g. test preambles).
REQ-2-ATS	NCT TTCN-3 test suite shall rely on TA (SA part) for the implementation of the WiMAX PHY layer (according to IEEE 802.16e-2005 [6]).
REQ-3-ATS	NCT TTCN-3 test suite shall rely on TA (PA part) for the implementation of the EAP protocol and the upper security methods. Initial implementation will cover mandatory features and optional features may be developed provided that IUT support is available.

Table 3: Test Adaptor Requirements

ID	Description
REQ-1-TA	TA shall use existing DLC-TTCN SA keeping the same TRI interface to serve the DLC-TTCN ports: TA, Phy, MacMsg, MacBcMsg and MacPdu.
REQ-2-TA	TA shall use an upgraded version of the DLC-TTCN Platform Adapter to deal with the EAP protocol and the upper security methods. See REQ-4-TA for upgrade scope.
REQ-3-TA	TA shall use the DLC-TTCN CODECS subsystem for DLC messages in order to code the messages transported by the DLC-TTCN3 ports: TA, Phy, MacMsg, MacBcMsg and MacPdu.

NOTE: In the present document, DLC-TTCN stands for the ATS to test BRAN HiperMAN/WiMAX systems for conformance in accordance with TS 102 545-3 [5]. Likewise, DLC-TTCN SA, PA and CODECS stands for the SA, PA and CODECS implementation respectively of a test machine in compliance with TS 102 545-3 [5].

4.1.2.2 Security test group

Table 4: TTCN-3 Test Suite Requirements

ID	Description
REQ-5-ATS	EAP protocol shall be TTCN-3 typed to handle incoming EAP packets from Test Adaptor in order to set test verdict.
REQ-6-ATS	EAP-TLS protocol shall be TTCN-3 typed to handle incoming EAP packets from Test Adaptor in order to set test verdict.
REQ-7-ATS	EAP-TTLS protocol shall be TTCN-3 typed to handle incoming EAP packets from Test Adaptor in order to set test verdict.
REQ-8-ATS	EAP-AKA protocol shall be TTCN-3 typed to handle incoming EAP packets from Test Adaptor in order to set test verdict.

Table 5: Test Adaptor Requirements

ID	Description
REQ-4-TA	EAP simulator in TA (PA part) shall support the following authentication methods: EAP-TLS, EAP-TTLS and EAP-AKA (according to RFC 3748 [21], RFC 5216 [31], RFC 5246 [32], RFC 4187 [24] and RFC 5281 [33]).
REQ-5-TA	TA shall implement CODECS subsystem for EAP protocol and authentication methods (EAP-TLS, EAP-AKA, EAP-TTLS/MSCHAPv2).
REQ-6-TA	In the case of EAP-TTLS, Test Adaptor (SA part) shall decrypt and encrypt the AVP (Attribute Value Pairs) throughout the inner security method transported in TLS-Record. Therefore, AVP shall be clear coded at the ATS level.

4.1.2.3 IPv4 test groups

Table 6: TTCN-3 Test Suite Requirements

ID	Description
REQ-9-ATS	DHCPv4 protocol shall be fully TTCN-3 typed according to the RFC 2131 [14] and RFC 2132 [15].
REQ-10-ATS	TTCN-3 test code shall act as the network side of the DHCPv4 protocol.
REQ-11-ATS	MIPv4 protocol shall be TTCN-3 typed according to the RFC 3344 [19], RFC 3543 [20], RFC 3846 [23] and RFC 4433 [27].
REQ-12-ATS	TTCN-3 test code shall emulate the network side of the MIPv4 protocol.
REQ-13-ATS	ICMPv4 protocol shall be TTCN-3 typed according to the RFC 792 [10] and RFC 1256 [13].
REQ-14-ATS	TTCN-3 test code shall emulate the network side of the ICMPv4 protocol.
REQ-15-ATS	IGMP protocol shall be TTCN-3 typed according to the RFC 1112 [12].
REQ-16-ATS	TTCN-3 test code shall emulate the network side of the IGMP protocol.
REQ-17-ATS	IPv4 protocol shall be partially TTCN-3 typed, as per NCT test case needs, according to the RFC 791 [9].

Table 7: Test Adaptor Requirements

ID	Description
REQ-7-TA	TA shall provide CODECS subsystems for DHCPv4, MIPv4, ICMPv4 and IGMP.

4.1.2.4 IPv6 test groups

Table 8: TTCN-3 Test Suite Requirements

ID	Description
REQ-18-ATS	DHCPv6 protocol shall be TTCN-3 typed according to the RFC 3315 [18] and draft_ietf_mip6_hiop17.txt [34].
REQ-19-ATS	TTCN-3 test code shall emulate the network side of the DHCPv6 protocol.
REQ-20-ATS	MIPv6 protocol shall be TTCN-3 typed according to the RFC 3775 [22], RFC 4283 [25] and RFC 4285 [26].
REQ-21-ATS	TTCN-3 test code shall emulate the network side of the MIPv6 protocol.
REQ-22-ATS	ICMPv6 protocol shall be TTCN-3 typed according to the RFC 4861 [29] and RFC 4862 [30].
REQ-23-ATS	TTCN-3 test code shall emulate the network side of the ICMPv6 protocol.
REQ-24-ATS	IPv6 protocol shall be partially TTCN-3 typed, as per NCT test case needs, according to the RFC 2460 [16].

Table 9: Test Adaptor Requirements

ID	Description
REQ-8-TA	TA shall provide CODECS subsystems for DHCPv6, MIPv6 and ICMPv6.

4.1.2.5 Miscellaneous requirements

Table 10: TTCN-3 Test Suite Requirements

ID	Description
REQ-25-ATS	TTCN-3 test code shall partially emulate all the counterpart protocols involved in the network functionalities under test (see REQ-[10, 12, 14, 16, 19, 21, and 23]-ATS). More specifically, NCT shall emulate only what SUT expects from its interface with the network (R1 and R2 as defined in WiMAX Forum Network Architecture; Stage 2 [1]).
REQ-26-ATS	TTCN-3 test code shall implement the IEEE P802.16g/ D9 standard [7] as per NCT test case needs.
REQ-27-ATS	TTCN-3 shall provide all attributes for all newly defined types.
REQ-28-ATS	MacMsg port in DLC-TTCN (and types, templates, etc.) shall transport IEEE 802.16g [7] types relevant for NCT testing.

Table 11: Test Adaptor Requirements

ID	Description
REQ-9-TA	TA shall manage all the security keys involved in NCT test cases.
REQ-10-TA	TA shall provide ciphering functionalities to the ATS level, e.g. to compute the "Authenticated Data" field of {AUTH-OPTION-TYPE} MIPv6 option.
REQ-11-TA	TA shall perform traffic encryption and decryption so that MAC PDU (i.e. network packet) is clear coded at the ATS level.
REQ-12-TA	TA shall encode and decode UDP header in compliance with the applicable RFC (outside the scope of NCT testing).
REQ-13-TA	TA shall encode and decode the IPv4 header in compliance with the applicable RFC (outside the scope of NCT testing).
REQ-14-TA	TA shall encode and decode the IPv6 header in compliance with the applicable RFC (outside the scope of NCT testing).
REQ-15-TA	TA shall redirect MAC Management PDU PKMv2 type to the proper test component for further and dedicated processing.
REQ-16-TA	TA shall redirect MAC PDU which contain IP packets, either Ipv4 or Ipv6, to the proper test component for further and dedicated test processing.
REQ-17-TA	TA shall provide CODECS subsystem for IEEE 802.16g [7] types relevant for NCT testing.

4.1.2.6 OTA test group

Table 12: TTCN-3 Test Suite Requirements

ID	Description
REQ-29-ATS	DNS protocol shall be TTCN-3 typed according to the RFC 2782 [54].
REQ-30-ATS	TTCN-3 test code shall emulate the network side of the DNS protocol for OTA bootstrapping.
REQ-31-ATS	HTTP protocol shall be TTCN-3 typed according to the RFC 2616 [55]
REQ-32-ATS	TTCN-3 test code shall emulate the network side of the HTTP protocol for OTA bootstrapping.

Table 13: Test Adaptor Requirements

ID	Description
REQ-18-TA	TA shall provide CODECS subsystems for DNS and HTTP.
REQ-19-TA	OMA server simulator in TA (PA part) shall be integrated.

4.1.3 Test method

The test method chosen is the remote test method with notional upper tester (see clause 4.1.3.1). Remote test method means that the test tool (the test machine + the executable test suite) shall behave as a WiMAX network when the IUT is a WiMAX terminal device. 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 R1 air interface (as defined in WiMAX Forum Network Architecture; Stage 2 [1]), the PHY and MAC layers of the test machine shall be totally conformant with the corresponding PHY and MAC layers specification to use the remote test method.

4.1.3.1 What is notional upper tester?

For description of notional upper tester see TS 102 545-3 [5], clause 4.1.2.1.

4.1.4 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.

4.1.5 Test machine configuration

4.1.5.1 Presentation

Protocol conformance testing methodology is based on ISO/IEC 9646 [i.2]. By nature, the target of NCT is not a protocol but network functionalities which involves several protocols and procedures. Those protocols are not the implementations under test. In fact, the goal of NCT testing is to verify that certain network functionalities are achieved by the IUT by means of such protocols. Therefore, ISO/IEC 9646 [i.2] cannot be directly used as methodology to define the architecture of NCT. However, ISO/IEC 9646 [i.2] principles lay the foundations for defining the NCT test architecture.

In accordance with TS 102 624-2 [4] and the PCO identified in clause 4.1.1, three type of OSI protocol stacks participate in NCT testing.

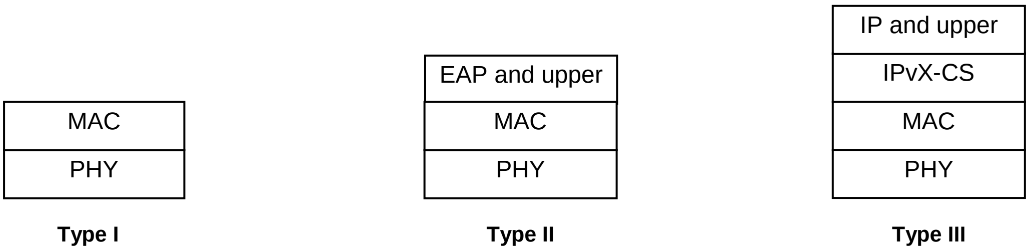


Figure 2: Type of Protocol Stacks on WiMAX terminal devices

Type I: DLC layer signalling procedures with the exception of PKMv2 procedure. Network discovery and selection and security test groups in accordance with TS 102 624-2 [4] define test cases with the purpose of testing on MAC layer. In the other test groups, Type I protocol stack may perform preamble test step.

Type II: Network entry and authentication (PKMv2 procedures). Security test group in accordance with TS 102 624-2 [4] defines test cases with the purpose of testing on EAP layer. In the other test groups, Type II protocol stacks may perform the preamble test step.

Type III: IP based mechanisms. DHCPv4, CMIPv4, IPv6 and CMIPv6 test groups in accordance with TS 102 624-2 [4] define test cases with the purpose of testing on IP protocol stack.

Considering those protocol stacks as the target of the ISO/IEC 9646 [i.2] methodology, the correspondent protocol conformance testers would be as depicted below in accordance with figure 2.

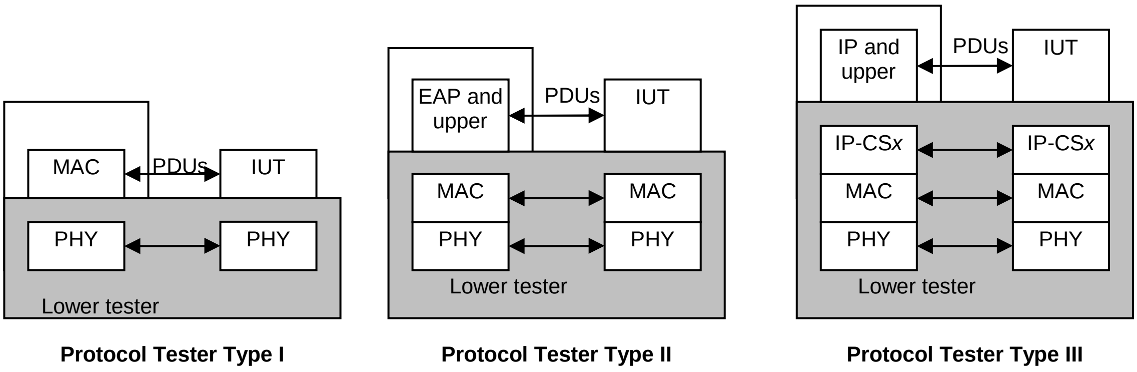


Figure 3: Type of Protocol Testers for WiMAX terminal devices

NOTE: DLC-TTCN as specified in TS 102 545-3 [5] belongs to Protocol Tester Type I.

NCT tester is not a pure protocol conformance tester as defined in ISO/IEC 9646 [i.2] wherein the tester implements the protocol counterpart. NCT tester must implement "to some degree" all the counterpart protocols involved in the network functionalities under test. However, NCT tester must emulate only what IUT expects from its interface with the network (R1 and R2 as defined by WiMAX Forum Network Architecture; Stage 2 [1]), so internal network behaviours are hidden to the IUT and therefore need not be implemented by NCT tester.

Consequently, NCT tester becomes from merging the three types of protocol testers into one single conformance tester. Thus, the architectures above become the test configurations for NCT tester. All type I, type II and type III test configurations shall be tailored by a concurrent testing approach (it is necessary to monitor and synchronize the network protocols simulated test code to obtain a consistent behaviour and a consistent test verdict).

Specifically according to TTCN-3 test system architecture principles, the three TTCN-3 test configurations presented would develop as given in figure 4.

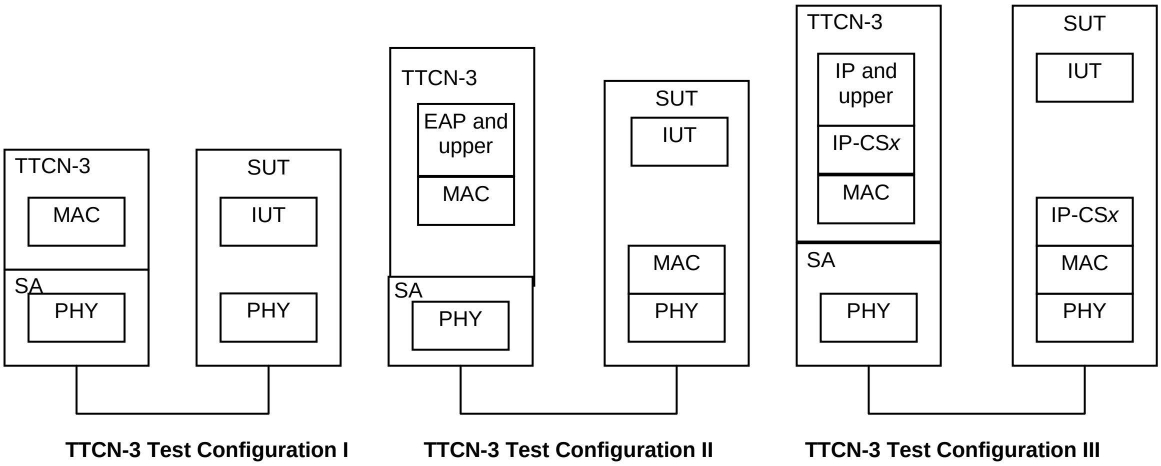


Figure 4: Test Configurations for NCT testing

NCT test configuration I (both SA and TTCN-3 test code) shall be provided by DLC-TTCN (see clause 4.1.6). Provisioning of NCT test configurations II and III is detailed in clause 4.1.7.

For similar reasons the number of test suites could be comprised between 1 and 3 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.

4.1.5.2 Test suite TTCN-3 development concept

See TS 102 545-3 [5], clause 4.1.4.1.

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.

4.1.6 Re-use of existing test specifications

Due to existing development for IEEE 802.16e-2005 [6], 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 802.16e-2005 [6] 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.

The re-use of existing specification in the present document consist of:

1) NCT test configuration I (both SA and TTCN-3 test code) shall be provided by DLC-TTCN specified in TS 102 545-3 [5].

2) NCT test configuration II implementation shall be strongly based on DLC-TTCN specified in TS 102 545-3 [5], namely PKMv2 related test code.

4.1.7 Test architecture

The NCT test architecture for testing network layer of a product implementing the HiperMAN and WiMAX Network Reference Architecture standards is shown in figure 5.

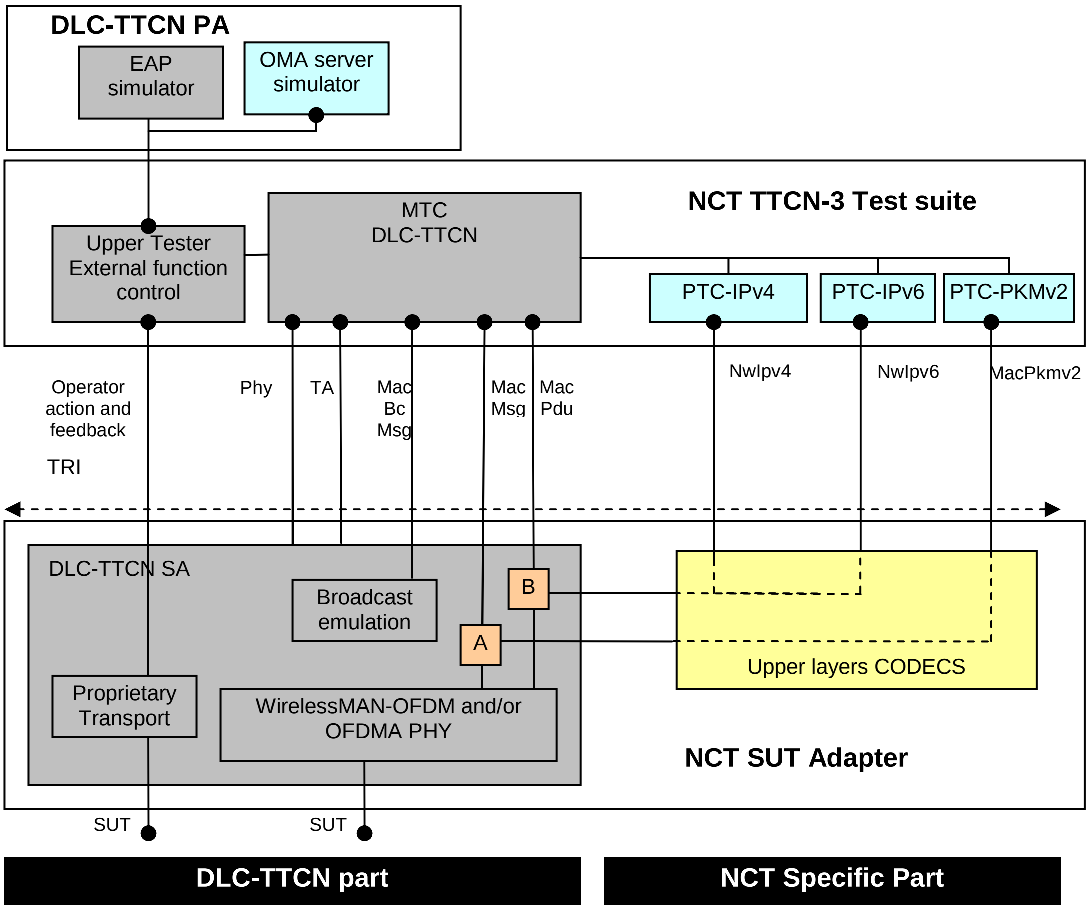


Figure 5: TTCN-3 Network Conformance Tester Architecture

The provisioning of functionalities for the NCT test configuration I is provided by existing DLC-TTCN specified in TS 102 545-3 [5].

The provisioning of functionalities for the NCT test configurations II and III is specified as follows:

- SA for test configuration II: provided by an upgrade of the available DLC-TTCN SA, hereinafter module A. Module A should redirect to/from MacPkmv2 port the MAC messages containing PKMv2 messages when authentication or re-authentication mechanisms occur. In NCT, PKMv2 signalling is processed in a new PTC at the ATS level. See details in clause 8.1.

- TTCN-3 test configuration II: provided by PTC-PKMv2 in the specific TTCN-3 test code for NCT. PTC-PKMv2 is strongly based on the existing PKM Version 2 implementation on DLC-TTCN specified in TS 102 545-3 [5]. However, PTC-PKMv2 shall perform conformance testing at EAP and upper security methods level. See details in clauses 4.1.7.1 and 4.2.

- SA for test configuration III: provided by an upgrade of the available DLC-TTCN SA, hereinafter module B. Module B intercepts incoming packets from PHY module, decrypt it, and examine whether or not the CID corresponds to IP-CS service flow (v4 or v6 depending on the running test case) and then extract MAC Payload and send it to NCT CODECS. On the reverse direction, module B encrypts the IP packet, encapsulates it in a MAC PDU and then sends it to the PHY module. See details in clause 8.2.

- TTCN-3 test configuration III: provided by new PTCs in the specific the TTCN-3 test suite for NCT. New ports (Nw prefix) are defined in order to transport network messages. Additionally, some IP information is also attached on these ports on demand. See details in clauses 4.1.7.1 and 4.2.

The TTCN-3 main test component, NTC-MTC, shall perform the coordination activities (Test Coordination Procedures according to ISO/IEC 9646) by orchestrating network nodes (implemented as TTCN-3 code) and therefore by emulating the network against which the IUT is connected.

As existing TTCN-3 code is used, the DLC-TTCN CODECS shall encode and decode the existing TTCN-3 types specified in TS 102 545-3 [5].

To sum up, the following test components constitute the NCT architecture:

- NCT TTCN-3 test suite: NCT-MTC (extending the existing DLC-TTCN) and the set of PTC which emulates the protocols on which the network functionalities under test are based.

- NCT TA:

    - SA: DLC-TTCN SA, module A and module B.

    - PA: an upgrade of EAP simulator to support EAP-TLS, EAP-TTLS and EAP-AKA authentication methods. An OMA DM simulator is needed to be integrated for OTA tests group.

    - CODECS: DLC-TTCN CODECS and NCT CODECS.

4.1.7.1 NCT specific part

The concurrent NCT Test Configuration provides the following test components:

- NCT-MTC: Main test component triggers and synchronizes the parallel test components. NCT-MTC orchestrates both the new PTC and the existing DLC-TTCN test suite in order to command the protocols involved in the network functionality under test. NTC-MTC also contains the DLC-TTCN test code in accordance with TS 102 545-3 [5] to emulate the DLC layer of HiperMAN/WiMAX terminal devices.

- PTC-IPv4: Parallel test component IPv4. PTC-IPv4 emulates the network side of the test cases belonging to IPv4, DHCPv4 and CMIPv4 test groups in accordance with the test suite structure defined in TS 102 624-2 [4]. PTC-IPv4 embodies both correspond node and foreign and home agents on the same TTCN-3 component when CMIPv4 is under test. PTC-IPv4 uses NwIpv4 port to send and receive DHCPv4 messages, ICMPv4 messages, MIPv4 messages, IGMP messages, HTTP messages, DNS messages and WAP Push messages. Final verdicts are set on the receive statements. (See details of NwIpv4 port type in clause 4.2).

- PTC-IPv6: Parallel test component IPv6. PTC-IPv6 emulates the network side of the test cases belonging to IPv6, and CMIPv6 test groups in accordance with the test suite structure defined in TS 102 624-2 [4]. PTC-IPv6 uses NwIpv6 port to send and receive DHCPv6 messages, ICMPv6 messages, and MIP6 messages. When CMIPv6 is under test, PTC-IPv6 uses NwIpv6 port to send and receive DHCPv6 messages of Home Address assignment when stateful configuration supported. Final verdicts are set on the receive statements. (See details of NwIpv6 port type in clause 4.2).

- PTC-PKMv2: Parallel test component PKMv2. PTC-PKMv2 emulates the EAP entity at the network side. PTC-PKMv2 embodies both authenticator and authentication server on the same TTCN-3 component. PTC-PKMv2 uses MacPkmv2 port to send and receive PKM Version 2 and inner EAP messages that belong to the Network authentication and re-authentication procedures. Final verdicts are set on the receive statements. (See details of the MacPkmv2 port type in clause 4.2).

The definition of the ATS level specified above demands the tester machine vendors to provide a TTCN-3 CODECS in accordance with ES 201 873-6 [38]. Notwithstanding the scope of the present document, tester machine vendors may need recommendations regarding CODECS implementation. See details in clause 6.1.6.

4.1.7.2 DLC-TTCN extension part

TTCN-3 test code of NCT uses DLC-TTCN test code as described in clause 4.1.6 for all MAC signalling except for PKMv2 procedures. Additionally, in order to emulate IEEE P802.16g/D9 [7] required for Network Entry and Discovery test group, an upgrade of DLC-TTCN test code is implemented for this purpose.

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

4.1.7.3 Requirements accomplishment

This clause summarizes how functional blocks of the test architecture proposed address the list of requirements given in clause 4.1.2.

Table 12: TTCN-3 test code - Requirements mapping

Test Component	Requirements
NTC-MTC	REQ-01-ATS, REQ-02-ATS, REQ-03-ATS, REQ-25-ATS, REQ-26-ATS, REQ-27-ATS, REQ-28-ATS
PTC-PKMv2	REQ-05-ATS, REQ-06-ATS, REQ-07-ATS, REQ-08-ATS
PTC-IPv4	REQ-09-ATS, REQ-10-ATS, REQ-11-ATS, REQ-12-ATS, REQ-13-ATS, REQ-14-ATS, REQ-15-ATS, REQ-16-ATS, REQ-17-ATS, REQ-29-ATS, REQ-30-ATS, REQ-31-ATS, REQ-32-ATS
PTC-IPv6	REQ-18-ATS, REQ-19-ATS, REQ-20-ATS, REQ-21-ATS, REQ-22-ATS, REQ-23-ATS, REQ-24-ATS

Table 13: Test Adaptor - Requirements mapping

Test Component	Description
DLC-TTCN	REQ-1-TA, REQ-2-TA, REQ-3-TA
NCT PA	REQ-4-TA, REQ-19-TA
NCT CODECS	REQ-5-TA, REQ-6-TA, REQ-7-TA, REQ-8-TA, , REQ-12-TA, REQ-13-TA, REQ-14-TA, REQ-17-TA, REQ-18-TA
NCT SA (Module A)	REQ-9-TA, REQ-10-TA, REQ-15-TA
NCT SA (Module B)	REQ-11-TA, REQ-16-TA

4.2 Description of the ports and their associated primitives

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;

6) one NwIpv4 type port;

7) one NwIpv6type port;

8) one MacPkmv2 type port.

4.2.1 The MacMessagePort type

4.2.1.1 Description

See TS 102 545-3 [5], clause 4.2.1.1.

4.2.1.2 Primitives of the MacMsg port

See TS 102 545-3 [5], clause 4.1.2.2.

4.2.1.3 NCT specific extension

New TLV are defined in IEEE P802.16g/D9 [7] which participates in NCT testing:

- NSP List TLV;
- NAP List TLV;
- NSP change count; and
- NAP-ID.

MacMessagePort shall support these TLV.

4.2.2 The MacBcMessagePort type

4.2.2.1 Description

See TS 102 545-3 [5], clause 4.2.2.1.

4.2.2.2 Primitives of the MacMsg port

See TS 102 545-3 [5], clause 4.2.2.2.

4.2.2.3 NCT specific extension

Not needed.

4.2.3 The MacPduPort type

4.2.3.1 Description

See TS 102 545-3 [5], clause 4.2.3.1.

4.2.3.2 Primitives of the MacPdu port

See TS 102 545-3 [5], clause 4.2.3.2.

4.2.3.3 NCT specific extension

The network layer messages which are not mapped to NwIpv4 or NwIpv6 shall be sent to MacPDU port (see details in clause 4.3).

4.2.4 The PhyPort type

4.2.4.1 Description

See TS 102 545-3 [5], clause 4.2.4.1.

4.2.4.2 Primitives of the Phy port

See TS 102 545-3 [5], clause 4.2.4.2.

4.2.4.3 NCT specific extension

Not needed.

4.2.5 The TAPort type

4.2.5.1 Description

See TS 102 545-3 [5], clause 4.2.5.1.

4.2.5.2 Primitives of the TA port

See TS 102 545-3 [5], clause 4.2.5.2.

4.2.5.3 NCT specific extension

Primitives are added to this port in order to enable the TA to perform the test strategies required in the present document (see details in clauses 8).

4.2.6 The NwIpv4Port type

4.2.6.1 Description

This port is used to send and receive:

1) DHCPv4 messages in accordance with the following RFC standards: RFC 2131 [14].

2) MIPv4 messages in accordance with the following RFC standards: RFC 2794 [17], RFC 3344 [19], RFC 3543 [20], RFC 3486 [44], RFC 3957 [45] and RFC 4433 [27].

3) ICMPv4 messages in accordance with the following RFC standards: RFC 792 [10] and RFC 1256 [13].

4) IGMP messages in accordance with the following RFC standard: RFC 1112 [12].

5) DNS SRV (over UDP or TCP) messages in accordance with the following RFC standard: RFC 2782 [54].

6) HTTP messages in accordance with the following RFC standard: RFC 2616 [55].

7) WAP Push messages in accordance with the standard [60].

8) IPv4 messages other than DHCPv4, MIPv4, ICMPv4, IGMP, DNS, HTTP and WAP Push shall be sent to NwIpv4 port as raw data.

In addition to the messages above, other information shall be transmitted in the NwIp4Ind and NwIp4Req primitives:

1) Some fields of the generic header of the IP datagram: source address, destination addresses, fragment offset, TTL and DF flag.

4.2.6.2 Primitives of the NwIpv4 Port

Two primitives of type NwIp4Primitives are currently defined:

1) The NwIpv4Req type primitive - to send IPv4 messages to the IUT.

2) The NwIpv4Ind type primitive - to receive IPv4 messages from the IUT.

Table 14: Fields of the NwIpv4Req type primitive

Field name	Description
NwIpv4Header	This field contains the following pieces of information: - The value of the source and destination addresses of the IP datagram that contains the outgoing IPv4 message. - The value of the Time to Live field of the IP datagram that contains the outgoing IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the Identification field of the IP datagram that contains the outgoing IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the Fragment Offset of the IP datagram that contains the outgoing IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the DF Flags of the IP datagram that contains the outgoing IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value.
NwIpv4Message	This field contains one of the following pieces of information: - The DHCPv4 sent on the Initial Service Flow established. This field is a record type that fully identifies the MIPv4 message. - The MIPv4 message sent on the Initial Service Flow established. This field is a record type that fully identifies the MIPv4 message. - The ICMPv4 message sent on the Initial Service Flow established. This field is a record type that fully identifies the ICMPv4 message. - The IGMP message sent on the Initial Service Flow established. This field is a record type that fully identifies the IGMP message. - The DNS UDP/TCP message sent on the Initial Service Flow established. This field is a record type that fully identifies the DNS message. - The HTTP message sent on the Initial Service Flow established. This field is a record type that partially identifies the HTTP message. - The WAP Push message sent on the Initial Service Flow established. This field is an octect string. Not typed. - Any raw data. This is used to send fragmented IP packets.

Table 15: Fields of the NwIpv4Ind type primitive

Field name	Description
NwIpv4Header	This field contains the following pieces of information: - The value of the source and destination addresses of the IP datagram that contains the incoming IPv4 message. - The value of the Time to Live field of the IP datagram that contains the incoming IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the Fragment Offset of the IP datagram that contains the incoming IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the DF Flags of the IP datagram that contains the incoming IPv4 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value.
NwIpv4Message	This field contains one of the following pieces of information: - The DHCPv4 message received from the Initial Service Flow established. This field is a record type that fully identifies the MIPv4 message. - The MIPv4 message received from the Initial Service Flow established. This field is a record type that fully identifies the MIPv4 message. - The ICMPv4 message received from the Initial Service Flow established. This field is a record type that fully identifies the ICMPv4 message. - The IGMP message received from the Initial Service Flow established. This field is a record type that fully identifies the IGMP message. - The DNS UDP/TCP message received from the Initial Service Flow established. This field is a record type that fully identifies the DNS message. - The HTTP message received from the Initial Service Flow established. This field is a record type that fully identifies the HTTP message. - The WAP Push message received from the Initial Service Flow established. This field is an octect string. Not typed. - Any raw data. This is used either to receive fragmented IP packets or to receive unknown messages that the Test Adaptor is able to process (see details in clause 8.2).
phyParams	This field contains the 3 following pieces of information, related to the message received: 1. iuc (either the DIUC or the UIUC); 2. symbol offset; 3. frame number.

4.2.7 The NwIpv6Port type

4.2.7.1 Description

This port is used to send and receive:

1) DHCPv6 messages in accordance with RFC 3315 [18] and "draft_ietf_mip6_hiop17.txt" [34].

2) MIPv6 messages in accordance with RFC 3775 [22], RFC 4283 [25] and RFC 4285 [26].

3) ICMPv6 messages in accordance RFC 4861 [29] and RFC 4862 [30].

4) IPv6 messages other than DHCPv6, MIPv6 and ICMPv6 SHALL be sent to NwIpv6 port as raw data.

In addition to the messages above, other information shall be transmitted in the NwIp4Ind and NwIp4Req primitives:

1) Some fields of the generic Header of the IP datagram: source address, destination addresses, Hop Limit, and Fragment Header.

4.2.7.2 Primitives of the NwIpv6 port

Two primitives of type NwIpv6Primitives are currently defined:

1) The NwIpv6Req type primitive - to send IPv6 messages to the IUT.

2) The NwIpv6Ind type primitive - to receive IPv6 messages from the IUT.

Table 16: Fields of the NwIpv6Req type primitive

Field name	Description
NwIpv6Header	This field contains the following pieces of information: - The value of the source and destination addresses of the IP datagram that contains the outgoing IPv6 message. - The value of the Hop Limit field of the IP datagram that contains the outgoing IPv6 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the Fragment Header of the IP datagram that contains the outgoing IPv6 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value.
NwIpv6Message	This field contains one of the following pieces of information: - The DHCPv6 sent on the Initial Service Flow established. This filed is a record type that fully identifies the MIPv6 message. - The MIPv6 message sent on the Initial Service Flow established. This filed is a record type that fully identifies the MIPv6 message. - The ICMPv6 message sent on the Initial Service Flow established. This filed is a record type that fully identifies the ICMPv6 message. - Any raw data. This is used to send fragmented IP packets.

Table 17: Fields of the NwIpv6Ind type primitive

Field name	Description
NwIpv6Header	This field contains the following pieces of information: - The value of the source and destination addresses of the IP datagram that contains the incoming IPv6 message. - The value of the Hop Limit field of the IP datagram that contains the incoming IPv6 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value. - The value of the Fragment Header of the IP datagram that contains the incoming IPv6 message. This sub-field is optional at the ATS level. If omitted, Test Adapter shall assume a valid default value.
NwIpv6Message	This field contains one of the following pieces of information: - The DHCPv6 received from the Initial Service Flow established. This filed is a record type that fully identifies the MIPv6 message. - The MIPv6 message received from the Initial Service Flow established. This filed is a record type that fully identifies the MIPv6 message. - The ICMPv6 message received from the Initial Service Flow established. This filed is a record type that fully identifies the ICMPv6 message. - Any raw data. This is used either to receive fragmented IP packets or to receive unknown messages that the Test Adaptor is able to process (see details in clause 8.2).
phyParams	This field contains the 3 following pieces of information, related to the message received: 1. iuc (either the DIUC or the UIUC); 2. symbol offset; 3. frame number.

4.2.8 The MacPkmv2Port type

4.2.8.1 Description

This port is used to send and receive PKM Version 2 messages in accordance with the following IETF RFC standards RFC 5246 [32], RFC 5216 [31], RFC 3748 [21], RFC 5281 [33] and RFC 4187 [24] and IEEE 802.16e-2005 [6].

4.2.8.2 Primitives of the MacPkmv2 port

Two primitives of type MacPkm2Primitives are currently defined:

1) The Pkmv2Req type primitive - to send PKM Version 2 messages to the IUT.

2) The Pkmv2Ind type primitive - to receive PKM Version 2 messages from the IUT.

Table 18: Fields of the Pkmv2Req type primitive

Field name	Description
Pkmv2Msg	The PKM Version 2 message received during authentication procedure. This filed is a record type that fully identifies the PKMv2 message and inner authentication method (i.e. EAP-TLS, EAP-TTLS and EAP-AKA).

Table 19: Fields of the Pkmv2Ind type primitive

Field name	Description
Pkmv2Msg	The EAP message sent during authentication procedure. This filed is a record type that fully identifies the PKMv2 message and inner authentication method (i.e. EAP-TLS, EAP-TTLS and EAP-AKA).
phyParams	This field contains the 3 following pieces of information, related to the message received: 1. iuc (either the DIUC or the UIUC); 2. symbol offset; 3. frame number.

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, EAP messages are contained in MAC messages and IPv4 and IPv6 packets (i.e. on NwIp4 and NwIp6 respectively) 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 all MacPdu, MacMsg (and BcMacMsg), MacPkm2 and "Nw" ports or only some of them are used. This gives 3 possible cases:

1) If only 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 respectively to the MacMsg or the BcMacMsg port. MAC PDUs with non-generic header are also sent to the MacPdu port.

2) If a MacPdu, a MacMsg/BcMacMsg and a MacPkmv2 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 respectively to the MacMsg or the BcMacMsg port, except the MAC management messages containing PKMv2 messages, which are sent to the MacPkmv2 port during authentication. MAC PDUs with non-generic header are also sent to the MacPdu port.

3) If a MacPdu, a MacMsg/BcMacMsg, MacPkmv2 and either NwIpv4 or NwIpv6 port 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 respectively to the MacMsg or the BcMacMsg or the MacPkm2 port, and except the MAC PDUs containing IP messages, which are sent to the appropriate Nw port. Furthermore, in the case of IPv4, IP Version 4 packets other than DHCP4, ICMP4, MIP4 and IGMP are sent to NwIpv4 port as "Payload" message type. In the case of IPv6, IP Version 6 packets other than DHCP6, ICMP6 and MIP6 are sent to NwIpv6 port as "Payload" message type.

Figure 6 illustrates the port mapping rules in NCT.

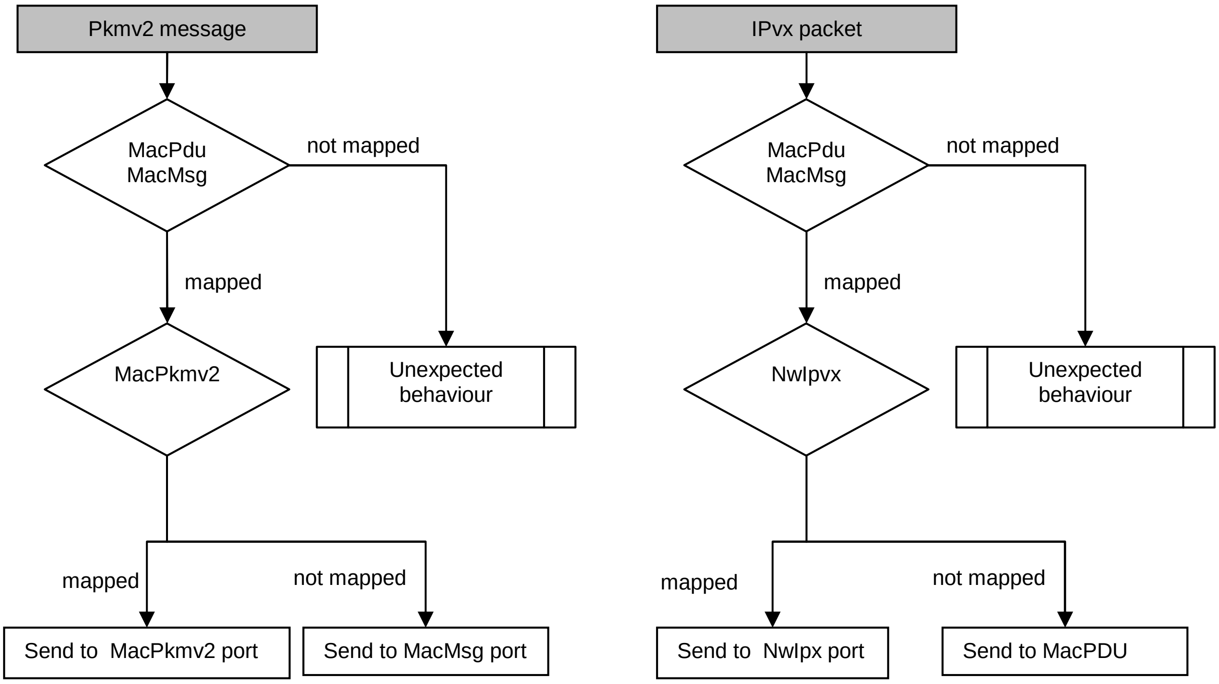


Figure 6: Port mapping rules flowchart

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 20: Untestable TP

Test Case Name	Reason

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 [35] were considered.

6.1 Testing conventions

6.1.1 Testing States

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

6.1.2 HiperMAN default values: Reception and transmission at ATS level

IEEE P802.16-2005e [6] lists many default TLV values. The specification states that devices should not transmit TLV if the default value applies. However, this is NOT a requirement. Thus, one tested device may not transmit the default TLV (or a subset of these default TLV) while another may transmit all TLV including the defaults. Including all the possible combinations of sent and received default TLV in an ATS is problematic.

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

6.1.3 IPv4 and UDP protocol headers: Reception and transmission at ATS level

UDP [8] and IPv4 [9] are not target of NCT testing. The correspondent standards states that IP nodes may, should or must transmit certain protocol fields to certain values. However, this is NOT a requirement at ATS level. Thus, one tested device may not transmit the optional protocol fields while another may transmit all protocol fields including the optional ones. Including all the possible combinations of sent and received protocol fields in an ATS is problematic. TA and ATS shall deal with each of the IPv4 and UDP protocol headers as specified in tables 21 and 22.

Table 21: Reception and transmission of IPv4 headers

IPv4 Header Fields	Transmission	Reception
Version	Fixed value set by TA	Discarded by TA
IHL	Variable value set by TA (see note 1)	Discarded by TA
Type of Service	Fixed value set by TA	Discarded by TA
Total Length	Variable value set by TA (see note 1)	Discarded by TA
Identification	Provided by the ATS level, if omitted, fixed value	Decoded by TA and passed up to the ATS level
Flags	Provided by the ATS level, if omitted, fixed value	Decoded by TA and passed up to the ATS level
TTL	Provided by the ATS level, if omitted, fixed value	Decoded by TA and passed up to the ATS level
Protocol	Variable value set by TA (see note 1)	Decoded by TA and passed up to the ATS level
Header checksum	Variable value set by TA (see note 1)	Checked by TA for validation but not pass it up to the ATS level
Source Address	Always provided by the ATS level	Decoded by TA and passed up to the ATS level
Destination Address	Always provided by the ATS level	Decoded by TA and passed up to the ATS level
Optional (see note 2)	Never encoded	Discarded by TA
NOTE 1: The value is assessed and set by the TA in each instance of message encoding. NOTE 2: RFC 761 [43] states: "*The options may appear or not in datagram. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation.*" RFC 761 [43] quote entails that NCT tester need not send optional fields and, NCT tester shall discard received optional fields.		

Table 22: Reception and transmission at ATS level of UDP headers

UDP Header	Transmission	Reception
Source Port	Variable value set by TA	Discarded by TA
Destination Port (see note)	Variable value set by TA	Discarded by TA
Length	Variable value set by TA	Discarded by TA
Checksum	Variable value set by TA	Checked by TA for validation but not pass it up to the ATS level
NOTE: In reception, destination port is the protocol field needed to select the application layer decoder. In transmission TA will set "destination port "to the value of "source port" provided by the IUT in the previous UDP datagram.		

6.1.4 Network protocol messages: Reception and transmission at ATS level

NCT testing does not test the entire list of related RFC thoroughly, but those parts specified in WiMAX Forum Network Architecture, Stage 3: Detailed Protocols and Procedures [2]. Those are hereinafter referred as the standards inside scope of NCT testing. The ATS level shall process the protocol types and messages inside the scope of NCT testing. TA therefore will encode and decode them accordingly.

As for the standards or part of them which remain outside the scope of NCT testing, rules of reception and transmission are defined in the present clause.

More specifically, the present clause addressees encoding and decoding part of the standards. Therefore, levels of encoding are defined in order to organize the rules: protocol type (e.g. DHCP), message type (e.g. DHCP-Discover) and optional fields (e.g. Renewal Time).

The rules of reception and transmission which shall apply both ATS and TA are specified below.

- For **protocols** over IPv4 or IPv6 **outside the scope** of NCT:
  - In transmission, ATS shall not send PDUs outside the scope of NCT testing.
  - In reception, TA shall decode the PDU as `Payload` TTCN-3 type in the "NwIpvXMessage" on "NwIpvx" port (where "x" is either 4 or 6). The PDU is ignored at the ATS level.
- For **unknown message types** of protocols over IPv4 or IPv6 inside the scope of NCT testing:
  - In transmission, ATS shall not send message types outside the scope of NCT testing.
  - In reception, TA shall decode the unknown message types as `Payload` TTCN-3 type in the "NwIpvXMessage" on "NwIpvx" port (where "x" is either 4 or 6). The PDU is ignored at the ATS level.
- For **optional fields** of protocols messages over IPv4 or IPv6 inside the scope of NCT testing;
  - In transmission the ATS shall not send optional fields outside the scope of NCT testing.
  - In reception, TA shall remove the optional fields outside the scope of NCT testing from the IUT and then pass the message up to the ATS.

The list of protocols, messages and optional fields inside the scope on NCT is compiled below. Otherwise, the rules presented above shall apply.

The **protocols inside the scope of NCT testing:** ICMPv4, ICMPv6, DHCPv4, DHCPv6, MIPv4, MIPv6, IGMP, DNS, HTTP and WAP Push (last three in the context of OTA testing).

The **protocol messages inside the scope of NCT testing** are listed below.

- ICMPv4: All the message types defined in RFC 792 [10] are known at the ATS level. Additionally, table 23 compiles other known ICMPv4 options at the ATS level specified in other RFCs.

Table 23: ICMPv4 known message types at the ATS level

Type	Name	Reference
3 to 5,11 to 16	All message types defined in RFC 792 [10]	RFC 792 [10]
9	Router Advertisement	RFC 1256 [13]
10	Router Solicitation	RFC 1256 [13]

- DHCPv4: All the message types defined in RFC 3315 [18] are known at the ATS level.
- MIPv4: All the message types defined in RFC 3344 [19] and 3543 are known at the ATS level.
- IPV6: All the header types defined in RFC 2460 [16] and 3775 are known at the ATS level.
- ICMPv6: table 24 compiles all the ICMPv6 message types known at the ATS level.

Table 24: ICMPv6 known message types at the ATS level

Type	Name	Reference
1	Destination Unreachable	RFC 4433 [27]
2	Packet Too Big	RFC 4433 [27]
3	Time Exceeded	RFC 4433 [27]
4	Parameter Problem	RFC 4433 [27]
128	Echo Request	RFC 4433 [27]
129	Echo Response	RFC 4433 [27]
133	Router Solicitation	RFC 4861 [29]
134	Router Advertisement	RFC 4861 [29]
135	Neighbour Solicitation	RFC 4861 [29]
136	Neighbour Advertisement	RFC 4861 [29]
137	Redirect Message	RFC 4861 [29]
144	Home Agent Address Discovery Request	RFC 3775 [22]
145	Home Agent Address Discovery Reply	RFC 3775 [22]
146	Mobile Prefix Solicitation	RFC 3775 [22]
147	Mobile Prefix Advertisement	RFC 3775 [22]
151	Multicast Router Advertisement	RFC 4286 [50]
152	Multicast Router Solicitation	RFC 4286 [50]
153	Multicast Router Termination	RFC 4286 [50]

- DHCPv6: All the message types defined in RFC 3315 [18] are known at the ATS level.
- MIPv6: All the mobility headers defined in RFC 3775 [22] are known at the ATS level.
- DNS and HTTP: Message types defined in RFC 2782 [54] and in RFC 2616 [55] are partially known at the ATS level (only the necessary parts involved in OTA testing)
- WAP Push: Messages are not defined at ATS level. These messages MUST be handled by their correspondant server in the TA but they MUST pass through the ATS.

The **protocol optional fields inside the scope of NCT testing** are listed below:

- DHCPv4: table 25 compiles all the known DHCPv4 options at the ATS level.

Table 25: DHCPv4 known options at the ATS level

Code	Name	Reference
1	Subnet Mask	RFC 2132 [15]
28	Broadcast Address	RFC 2132 [15]
43	Vendor specific information	RFC 2132 [15]
50	Requested Ip Address	RFC 2132 [15]
51	Ip Address Lease Time	RFC 2132 [15]
53	Message type	RFC 2132 [15]
54	Server identifier	RFC 2132 [15]
57	Maximum DHCP message size	RFC 2132 [15]
58	Renew time value	RFC 2132 [15]
59	Rebinding time value	RFC 2132 [15]
61	Client Identifier	RFC 2132 [15]
-	Magic Cookie	RFC 2132 [15]
-	End	RFC 2132 [15]

- MIPv4: tables 26 and 27 compile all the known MIPv4 extensions at the ATS level.

Table 26: MIPv4 known extension types at the ATS level

Type	Name	Reference
32	Mobile-Home Authentication	RFC 3344 [19]
33	Mobile-Foreign Authentication	RFC 3344 [19]
34	Foreign-Home Authentication	RFC 3344 [19]
26	Generalized Mobile IP Authentication	RFC 4721 [51]
40	MA-FA-KeyGen Request	RFC 3957 [45]
41	MA-FA-KeyGen Reply	RFC 3957 [45]
42	MA-HA-KeyGen Request	RFC 3957 [45]
43	MA-HA-KeyGen Reply	RFC 3957 [45]
131	Mobile Node NAI	RFC 2794 [17]
132	MN-FA Challenge Extension	RFC 4721 [51]
136	NAI Carrying Extension	RFC 3846 [23]
137	Revocation Support	RFC 3543 [20]
139	Dynamic HA Extension	RFC 4433 [27]

Table 27: MIPv4 known extension types at the ATS level for ICMPv4 router discovery messages

Type	Name	Reference
0	One-byte padding	RFC 3344 [19]
16	Mobility Agent Advertisement	RFC 3344 [19]
19	Prefix Length	RFC 3344 [19]

In order to conserver the type number space, in MIPv4 extension types may have sub-types. Table 28 compiles all the MIPv4 extensions sub-type known at the ATS level for the generalized authentication extension type. The value of any MIPv4 extension sub-type unknown to the ATS level shall be decoded as raw octetstring (and therefore no further decoded) and then pass up to the ATS level. No extension sub-types shall be discarded at TA level.

Table 28: MIPv4 known extension sub-types at the ATS level for generalized authentication type

Sub-type	Name	Reference
1	MN-AAA Authentication	RFC 4721 [51]
2	FA-FA Authentication	RFC 4857 [52]
3	MN-GFA Authentication	RFC 4857 [52]
4	MN-PAR Authentication	RFC 4988 [53]

- DHCPv6: All the options defined in RFC 3315 [18] are known at the ATS level. Additionally, table 29 compiles other known DHCPv6 options at the ATS level specified in other RFCs.

Table 29: DHCPv6 known options at the ATS level

Code	Name	Reference
1 to 20	All options defined in RFC 3315 [18]	RFC 3315 [18]
49	MIP6 Home Network Identifier	ietf-mip-hiop-17.txt [34]
50	MIP6 Relay	RFC 2132 [15]

- MIPv6: All the mobility options defined in RFC 3775 [22], 4283 and 4285 are known at the ATS level.

6.1.5 EAP-TTLS encrypted tunnel: Reception and transmission at ATS level

EAP-TTLS [33] consists in a TLS handshake protocol to establish an end to end tunnel to secondly transmit encrypted user credentials. Tested devices therefore provide user credentials to the Tester in a second handshake procedure transported on an encrypted channel (TLS record layer) by means of AVP protocol fields. Some test cases checks these AVP, therefore all incoming AVP from the tested device shall be available unencrypted at the ATS level in such test cases. However, as cipher suite management is NOT a requirement at the ATS level. Test Adaptor shall be responsible for AVP decryption. See details in clause 8.3.

6.1.6 NCT CODECS Implementation (Informative)

Most of the Point of Observation and Control for NCT testing are on upper layer protocols (see clause 4.1.1), and according to TTCN-3 standards, TA should be in charge of CODECS implementation for all the required PCO:

- Security protocols: EAP, EAP-TLS, EAP-TTLS, EAP-AKA.
- Network protocols: ICMPv4, DHCPv4, MIPv4, IGMP, IPv6, ICMPv6, DHCPv6 and MIPv6.
- DLC layer.

NCT CODECS should implement TTCN-3 CODECS in accordance with ES 201 873-6 [38] for security and network protocols.

Existing DLC-TTCN CODECS should implement TTCN-3 CODECS in accordance with ES 201 873-6 [38] for IEEE 802.16e-2005 [6] and IEEE 802.16g [7]/D9 standards.

The SA may be in charge of PKMv2 header encoding and decoding as that procedure is not target of testing in NCT. Actually, this MAC layer processing is required to address the MAC PDU to the proper CODECS (either DLC-TTCN or NCT). See details in clause 8.1.

The SA may be in charge of IPv4 and UDP header encoding and decoding as those protocols are not target of testing in NCT. Actually, some MAC layer processing is required to address the MAC PDU to the proper CODECS (either DLC-TTCN or NCT). See details in clause 8.2.

6.1.7 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.

6.1.8 Functions

The 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 MS testing and preamble and postamble functions.

The MS 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 LibCommon_VerdictControl.ttcn file.

- 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 30) indicating the type of TTCN-3 element it represents.

- Suffixes should not be used except in those specific cases identified in table 30.

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

Table 30: TTCN-3 naming convention

Language element	Naming convention	Prefix	Suffix	Example	Notes
Module	Use upper-case initial letter	*none*	*none*	WMx_NCT_CommonFns	
TSS grouping	Use all upper-case letters	*none*	*none*	TP_RT_PS_TR	
Item group within a module	Use lower-case initial letter	*none*	*none*	messageGroup	
Data type	Use upper-case initial letter	*none*	*none*	SetupContents	
List type identifiers	Use upper-case initial letter	none	*none*	DlMapIeList	
Message template	Use lower-case initial letter	m_	none	m_setupInit	
Message template with wildcard or matching expression	Use lower-case initial letters	mw_	none	mw_setupBasic	
Port instance	Use lower-case initial letter	*none*	*none*	signallingPort	
Test component ref	Use lower-case initial letter	*none*	*none*	userTerminal	
Signature	Use lower-case initial letter	s_	*none*	s_callSignature	
External function	Use lower-case initial letter	xf_	*none*	xf_calculateLength()	
Constant	Use lower-case initial letter	c_	*none*	c_maxRetransmission	
Function	Use lower-case initial letter	f_	*none*	f_authentication()	
Altstep	Use lower-case initial letter	a_	*none*	a_receiveSetup()	
Altstep (Default)	Use lower-case initial letter	d_	*none*	d_receiveOtherMessages()	
Variable	Use lower-case initial letter	v_	*none*	v_basicCid	
Variable, global to component	Use lower-case initial letter	g_	*none*	g_ssSimu.basicCid	
Timer	Use lower-case initial letter	t_	_min _max	t_wait t_auth_min	(see note 1)
Module parameters PICS values PIXIT values	Use all upper case letters	*none*	*none*	PIC_T7PXT_TNOAC	(see note 2)
External constant	Use lower-case initial letter	xc_	*none*	xc_macId	
Parameterization	Use lower-case initial letter	p_	*none*	p_macId	
Enumerated Value	Use lower-case initial letter	e_	*none*	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

The identifier of the test cases is built according to table 31 as specified in TS 102 624-2 [4].

Table 31: TC naming convention

Identifier	TP/<pg>/<fg>/<sg>/<x>-H<nnn>		
	<st> = side type	MS	Mobile Station
	<pg> = protocol group	CMIPv4	Client Mobile IP v4
		DHCP	Dynamic Host Configuration Protocol
		QoS	Quality of Service
		SEC	Security
		IPv6	IP v6
		CMIPv6	Client Mobile IP v6
	<fg> = function group		To be added in subsequent releases
	<sg> = subfunction group		To be added in subsequent releases
	<x> = type of testing		To be added in subsequent releases
	<nnn> = sequential number	Hnnn	(H000, H001, etc.)

EXAMPLE: TP/MS/NWE/BV-H000
TP/MS/DHCP/TI-H001

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_packetIpv6OverIeee8023EthernetEcrtp(11)	No
e_packetIp2Rohc(12)	No
e_packetIp2Ecrtp(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 defined in the proper module according to table 32.

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

Table 32: Module names for signedoff MS test cases

TC groups	Module names	Description
NWE	WMx_NCT_Testcases_Nwe	Network Discovery and Network Selection/re-selection
SEC	WMx_NCT_Testcases_Security	EAP, EAP-TLS, EAP-TTLS, EAP-AKA
CMIPv4 DHCPv4	WMx_NCT_Testcases_IPv4	DHCPv4, MIPv4
CMIPv6 IPv6	WMx_NCT_Testcases_IPv6	IPv6, DHCPv6, MIPv6

6.5 Reuse of TTCN modules

The present document aims to maximize the reuse of TTCN test code given in TS 102 545-3 [5]. Actually, the objective is to import the maximum unmodified TTCN test code from TS 102 545-3 [5]. Thus, subsequent releases of TS 102 545-3 [5] can be imported to the present document without demanding validation. This goal cannot be completely achieved since IEEE P802.16g [7]/D9 standard is inside scope of NCT testing and require to modify existing TTCN code.

Accordingly, the organization of the TTCN code is as follows:

- NCT modules: TTCN code which is for the exclusive use of the present document (see note 1).

- PCT (see note 2) modules: Unmodified DLC-TTCN test code from TS 102 545-3 [5].
- PCT modified modules: modified DLC-TTCN test code from TS 102 545-3 [5]. In the present deliverable the following modules belong to this group (see note 3):
  - WMx_TestSystem_16e.ttcn: due to test system interface modification for NCT.
  - WMx_Messages_16e.ttcn: IEEE P802.16g [7]/D9 standard included.
  - WMx_Templates_16e.ttcn: IEEE P802.16g [7]/D9 standard included and unused templates removed.

NOTE 1: The new EAP types are not defined in NCT modules in order to avoid incompatibility issues with existing EAP types of TTCN code in TS 102 545-3 [5]. Rather, the new EAP types will be defined, but never used, in subsequent releases of TS 102 545-3 [5] in order to allow forward compatibility with the present document.

NOTE 2: In the present document PCT (Protocol Conformance Test) tester is the non-normative term to refer a test machine built in compliance with TS 102 545-3 [5].

NOTE 3: If IEEE P802.16g [7]/D9 standard is adopted in subsequent releases of TS 102 545-3 [5], PCT updated modules will only consist of WMx_TestSystem_16e.ttcn.

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 TS 102 545-3 [5], annex F.

8 Test strategies

Due to the combination of PHY and MAC procedures, IP and upper layers, the protocols involved in NCT testing require by nature using dedicated strategies for testing.

NOTE 1: The distribution of TA processing tasks between SA, PA and CODECS subsystems is informative in this clause.

NOTE 2: Since test strategies are specified at execution level, the term ETS is used instead of ATS in this clause.

8.1 Processing of Pkmv2 messages

Unlike in DLC-TTCN test code wherein Pkmv2 messages are processed like any other MAC management message at the ETS level, in NCT testing the Pkmv2 messages are processed by the dedicated test component MacPmv2. At the TA level, in DLC-TTCN the Pkmv2 messages are nor specially processed but treated as MAC management messages. In NCT however, TA shall process MAC management messages and identify which are Pkmv2 type and then encoded/decoded properly. This NCT specific functionality is carried out by the module A which belongs to the TA.

The main goal of module A is to handle the exchange of PKMv2 messages at TA level. In transmission (i.e. TE to IUT), the module A addresses the proper encoder and provides the correspondent primary CID. In reception, (i.e. IUT to TE), the module A checks that the message is PKMv2 type.

Figure 7 illustrates processing of Pkmv2 messages.

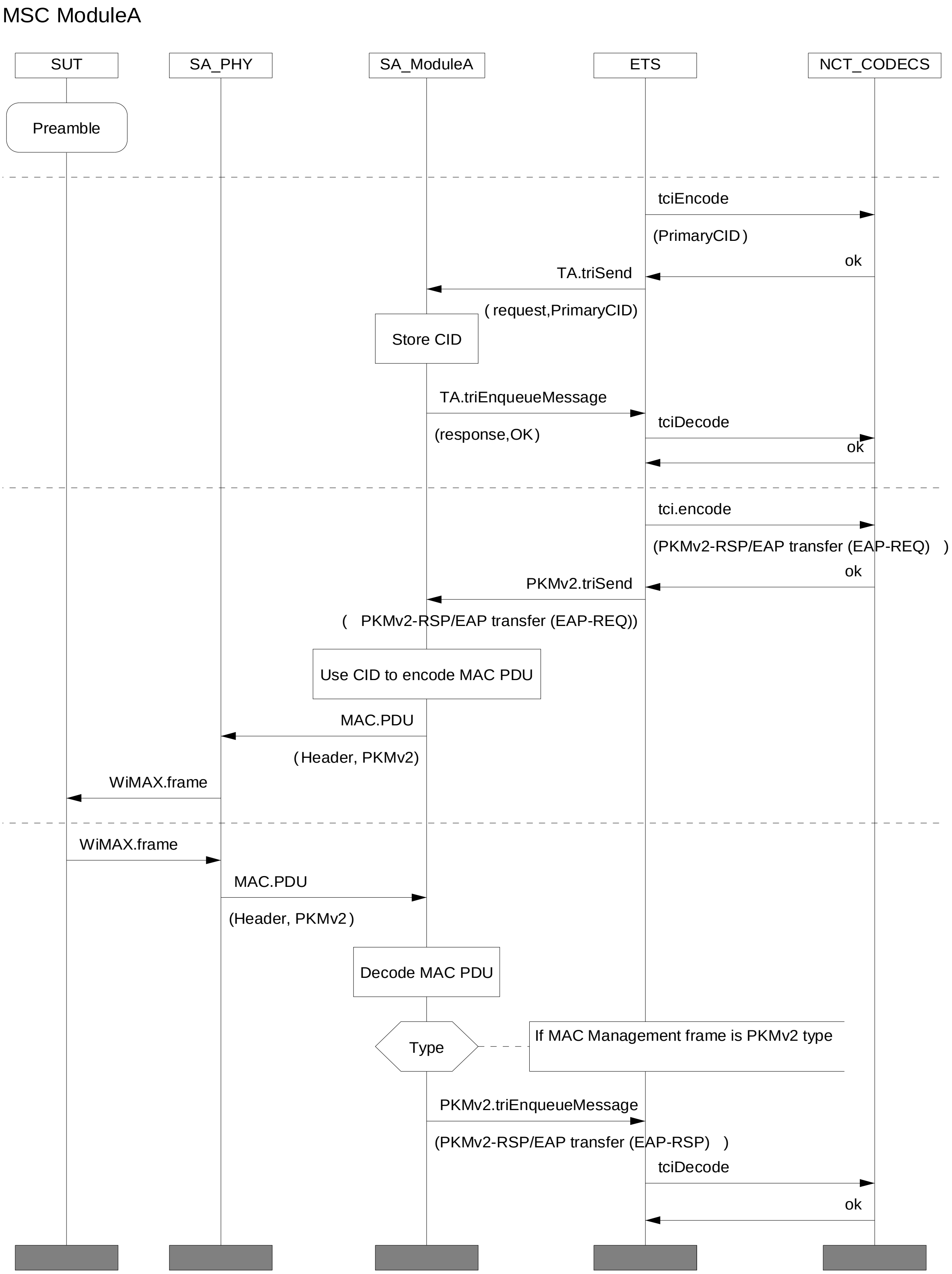


Figure 7

Test Step 1: the test case preamble consists in Ranging and SS capability negotiation procedures.

Test Step 2: the ETS sends the Primary CID to the module A through a new TA port primitive. Module A will keep this value until the end of the test case.

Test Step 3: The EAP part of the PKMv2 messages outgoing from MacPkmv2 port is encoded by the NCT CODECS. By using DLC-TTCN CODECS, TA encodes PKMv2 header and the rest of the MAC PDU frame: using Generic MAC header (i.e. as in DLC-TTCN) and using the stored primary CID reported in test step 2. Then, the encoded WiMAX frame is sent to the IUT via SA PHY module.

Test Step 4: the incoming WiMAX frame from PHY module is initially decoded by the DLC-TTCN CODECS. If WiMAX frame is MAC management message: check "Type" field in the MAC header. If the type is PKMv2, the PKMv2 message (header) is decoded by the DLC-TTCN CODECS whereas the EAP part is decoded by the NCT CODECS. Finally, the completely decoded PKMv2 message is sent to the NCT port MacPkmv2. Otherwise, if the WiMAX frame is not PKMv2 type, then the frame is processed like in DLC-TTCN.

The Pkmv2 decoding algorithm is described in figure 8.

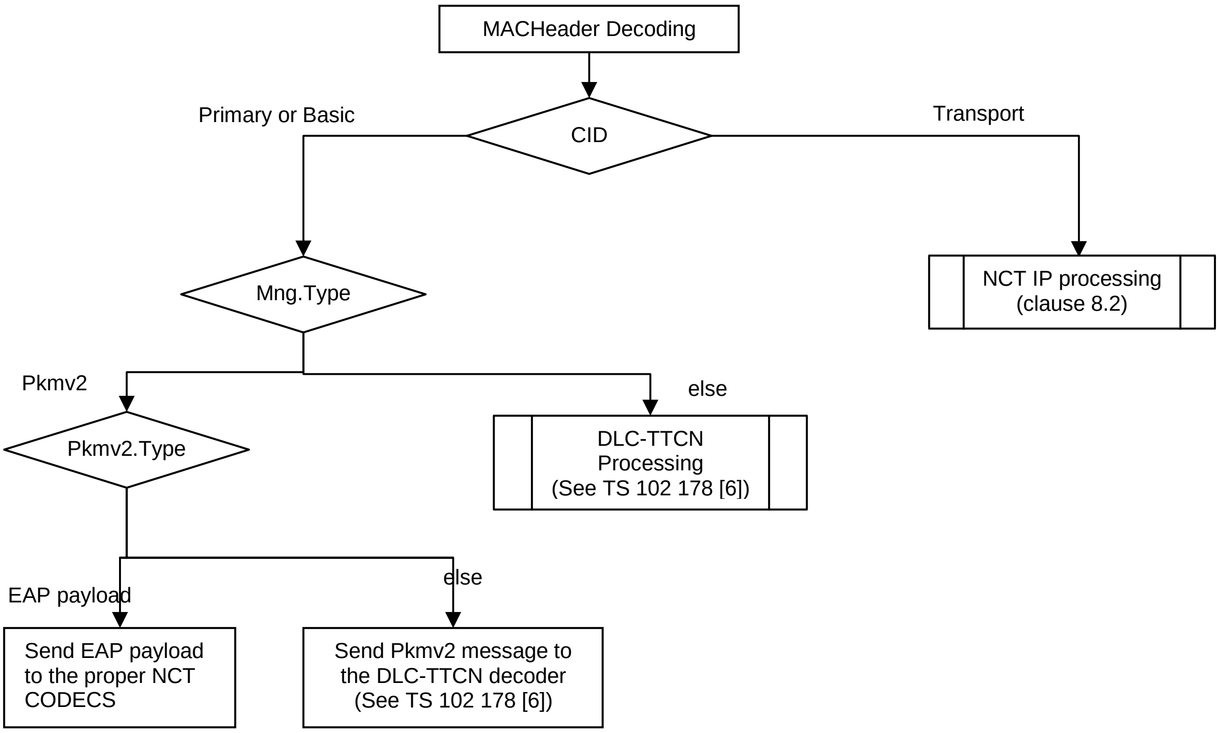

MACHeader Decoding
CID
Primary or Basic
Transport
Mng.Type
NCT IP processing (clause 8.2)
Pkmv2
else
Pkmv2.Type
DLC-TTCN Processing (See TS 102 178 [6])
EAP payload
else
Send EAP payload to the proper NCT CODECS
Send Pkmv2 message to the DLC-TTCN decoder (See TS 102 178 [6])

Figure 8: Pkmv2 messages decoding flowchart

The existing DLC-TTCN CODECS subsystem are valid to process at PKMv2 level. For NCT purposes though, a new CODECS subsystem should be provided to process EAP itself and the authentication method.

8.2 Processing of IP packets

Unlike in DLC-TTCN test code wherein IP packets are indistinctly processed as MAC PDU messages at the ETS level, in NCT testing the IP packets are processed by the dedicated test components either NwIpv4 or NwIPv6.

At the TA level, in DLC-TTCN the IP packets messages are nor specially processed but treated as MAC PDU messages. In NCT however, TA shall process MAC PDU messages and identify which are IP packet and then encoded/decoded them properly. This NCT specific functionality is carried out by the module B which belongs to the TA.

The main goal of module B is to handle the exchange of network layer messages at system adaptor level. In transmission (i.e. TE to IUT), the module B addresses the proper encoder and provides the correspondent transport CID. In reception, (i.e. IUT to TE), the module B checks that the payload of the MAC message is an IP message.

Figure 8 illustrates the processing of IP packets.

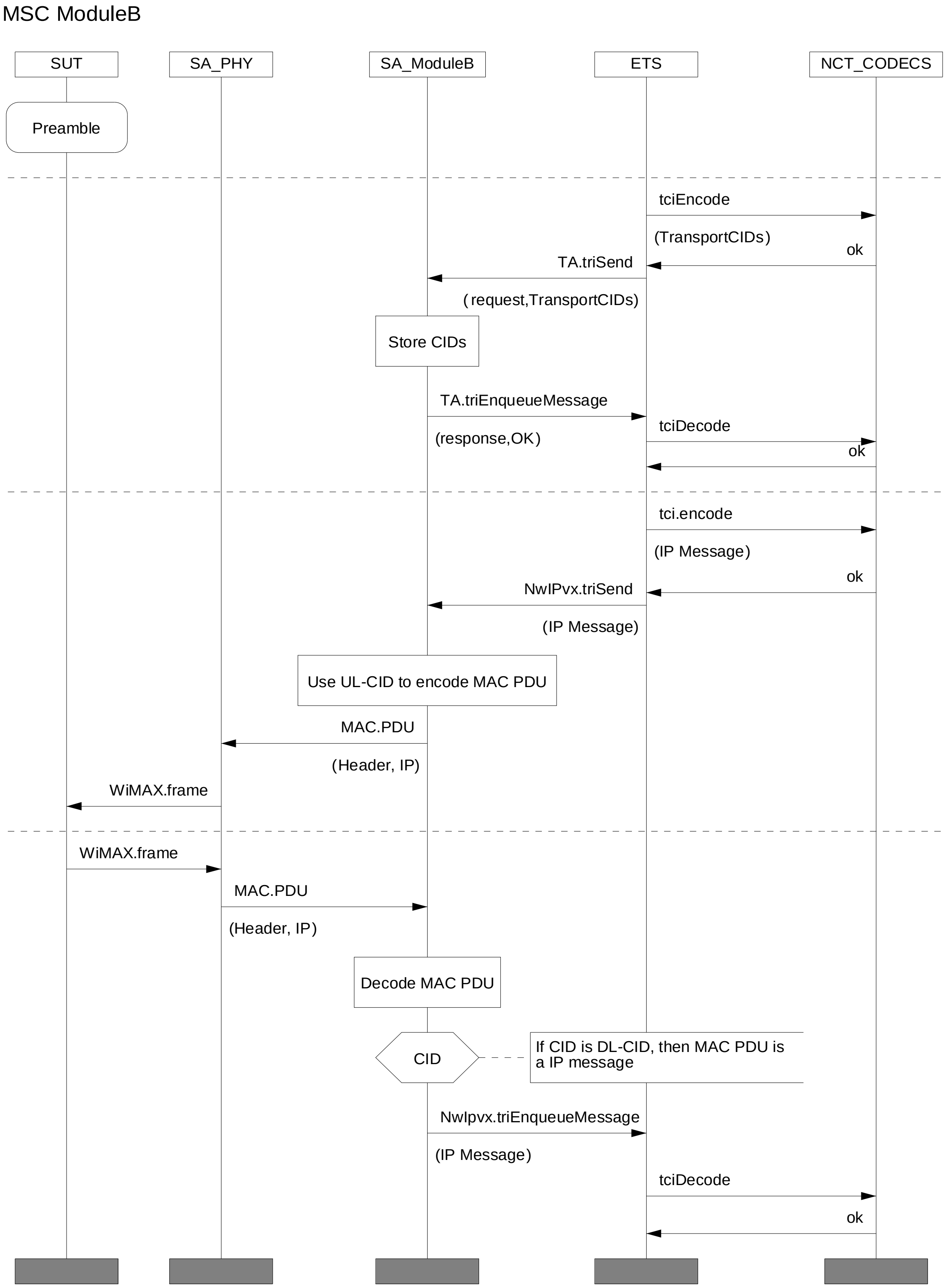


Figure 9: Module "B" MSC

Test Step 1: the test case preamble consists in Ranging and SS capability negotiation procedures and initial service flows (ISF) establishment.

Test Step 2: the ETS sends the Transport CIDs to the module B through a new TA port primitive: CID for uplink ISF (UL-CID) and CID for downlink ISF (DL-CID). Module B should keep these values until the end of the test case.

Test Step 3: The outgoing IP message is encoded by the NCT CODECS as the WiMAX MAC PDU payload. The WiMAX MAC header is encoded by the DLC-TTCN CODECS but using the UL-CID which sent the ETS in the test step 2. Then, the encoded WiMAX frame is sent to the IUT via SA PHY module.

Test Step 4: the incoming WiMAX frame from PHY module is decoded by the DLC-TTCN CODECS. Module B checks CID field in the MAC header. If the CID is the DL-CID, the MAC payload is decoded by the NCT CODECS whereas the MAC header is not decoded and then discarded.

The IP packet decoding algorithm is described in figure 9.

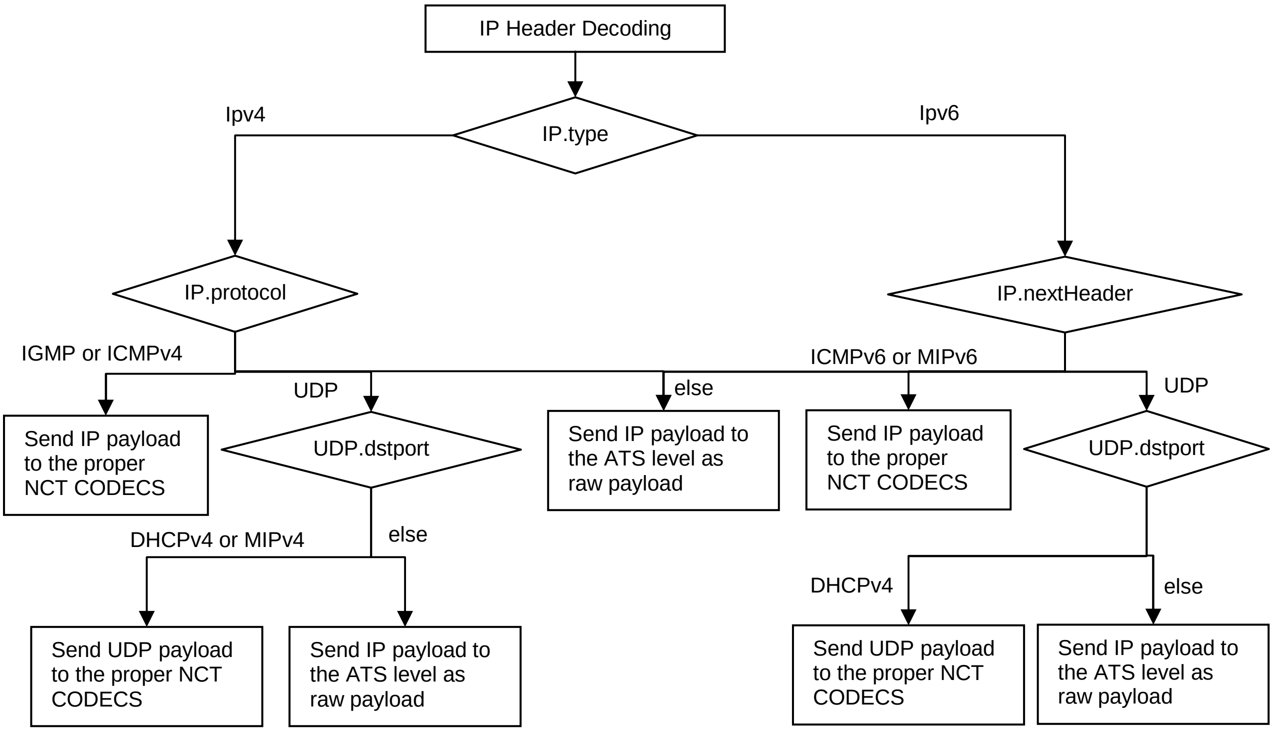

IP Header Decoding
Ipv4
IP.type
Ipv6
IP.protocol
IP.nextHeader
IGMP or ICMPv4
ICMPv6 or MIPv6
UDP
else
UDP
Send IP payload to the proper NCT CODECS
UDP.dstport
Send IP payload to the ATS level as raw payload
Send IP payload to the proper NCT CODECS
UDP.dstport
DHCPv4 or MIPv4
else
DHCPv4
else
Send UDP payload to the proper NCT CODECS
Send IP payload to the ATS level as raw payload
Send UDP payload to the proper NCT CODECS
Send IP payload to the ATS level as raw payload

Figure 10: IP Packets Decoding flowchart

NOTE: Sending the IP payload as "raw payload" means to send the data to the ATS level via "NwIpvx" port and "NwIpv4Message" field with the union TTCN-3 type set to `Payload` (see clause 4.2) not decoded.

8.3 Processing of EAP-TTLS AVPs

According to EAP-TTLS [33], an EAP-TTLS negotiation comprises two phases: the TLS handshake (phase 1) and the TLS tunnel (phase 2). Phase 1 results in the activation of a cipher suite, allowing phase 2 to tunnel information between client (tested device) and TTLS server (tester) to perform user authentication. Information between client and TTLS server is exchanged encrypted via attribute-value pairs (AVP). Since some test cases checks AVP, these must be available unencrypted at the ATS level.

TA shall be responsible for AVP decryption. The Message Sequence Chart depicted below shows the processing of EAP-TTLS messages and how AVPs are obtained in clear format at the ATS level.

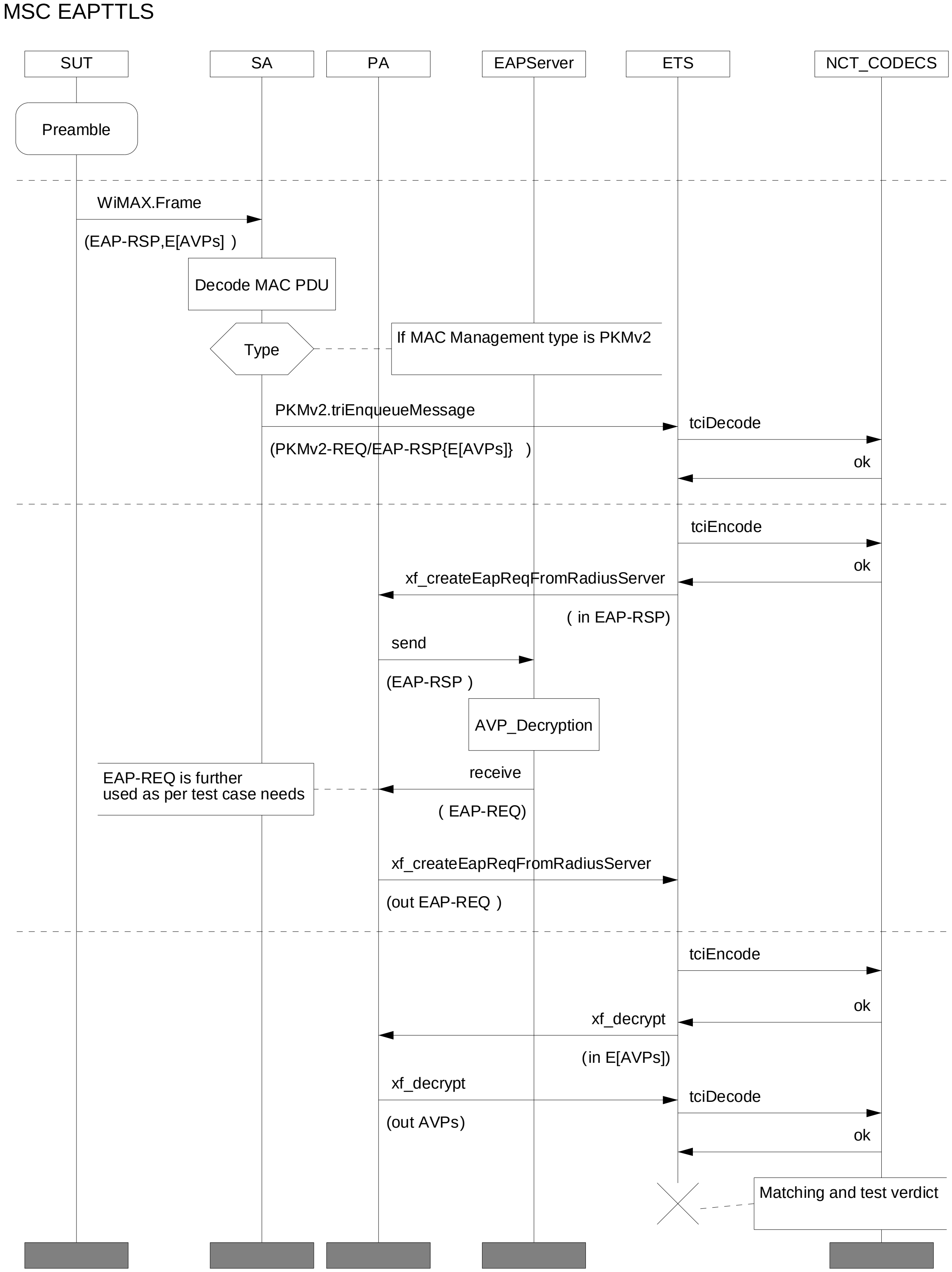


Figure 11

Test Step 1: the test case preamble consists in Ranging, SS capability negotiation procedures and phase 1 of EAP-TTLS authentication.

Test Step 2: the WiMAX message sent by the IUT is processed by the TA (i.e. module A). The EAP part of the PKMv2 message is decoded by the NCT CODECS and then available at the ETS level. Note that the TTLS AVP are encrypted at this step of execution.

Test Step 3: ETS invokes the external function that sends the EAP-RSP message from the IUT in the test step 1 to the EAP server and receive the following EAP message according to the EAP protocol behaviour. At this point, EAP Server silently decrypts the AVP carried by the EAP-RSP message and stores the clear AVP. Then, the EAP Server is ready to serve the external function `xf_decrypAVP` function.

The AVP decryption functionality in the EAP server may be available by nature. According to the EAP-TTLS standard [33], the tested device begins the phase 2 exchange by encoding information in a sequence of AVP, passing this sequence to the TLS record layer for encryption, and sending the resulting data to the TTLS server. Then, the server recovers the AVP in clear text from the TLS record layer. So, recovering clear AVP is a normative behaviour of the TTLS server. Moreover, if the AVP sequence includes authentication information, as in NCT test case, it forwards this information to the AAA/H server using the AAA carrier protocol. Even though the test machine does not need an AAA server, the interface may be available in the EAP-TTLS server. In such a case, clear AVP may be acquired at this point by simple interception.

Test Step 4: ETS invokes the NCT external function `xf_decryptAVP` in order to obtain the clear coded AVP.

If the test purpose is accomplished by matching the EAP-TTLS message (regardless of AVP) received from the IUT, test step 2 and 3 are not required.

8.4 IP Fragmentation testing

Testing fragmentation at network level (i.e. IPv4 or IPv6 fragmentation) requires special procedures in order to avoid additional upper tester requirements. As ICMP is already a requirement at the ETS level, the proposed approach in the present clause is based on ICMP protocol. The procedure is valid regardless of IP version.

IP fragmentation and packing capabilities in the tested device can be tested following the same procedure:

1) At the ETS level two or more Ipv4Request primitives containing a fragmented ICMP Request message are sent to the TA. Fragmentation is therefore performed at ETS level.

2) Upon receiving these packets, if correctly received and reassembled, the tested device must respond with two or more IP packet containing an ICMP Reply message with the same total payload length as the ICMP Request carried.

3) TA must pass the packets up to the ETS level which shall set the verdict accordingly.

ETS test logic shall define which capability is tested in each test case.

8.5 Processing OTA messages

According to OTA General [59] a provisioning and subscription phase of an OTA procedure comprises three phases: the bootstrap procedure (phase 1), the order of subscription and account setup (phase 2) and the provisioning device procedure and the device management (phase 3). Phase 1 results in the selection of the device management protocol, allowing phase 2 and 3 to proceed with the correct device management protocol.

TA shall be responsible for replying all messages involved in OTA provisioning procedure. The Message Sequence Chart depicted below shows the processing of these messages.

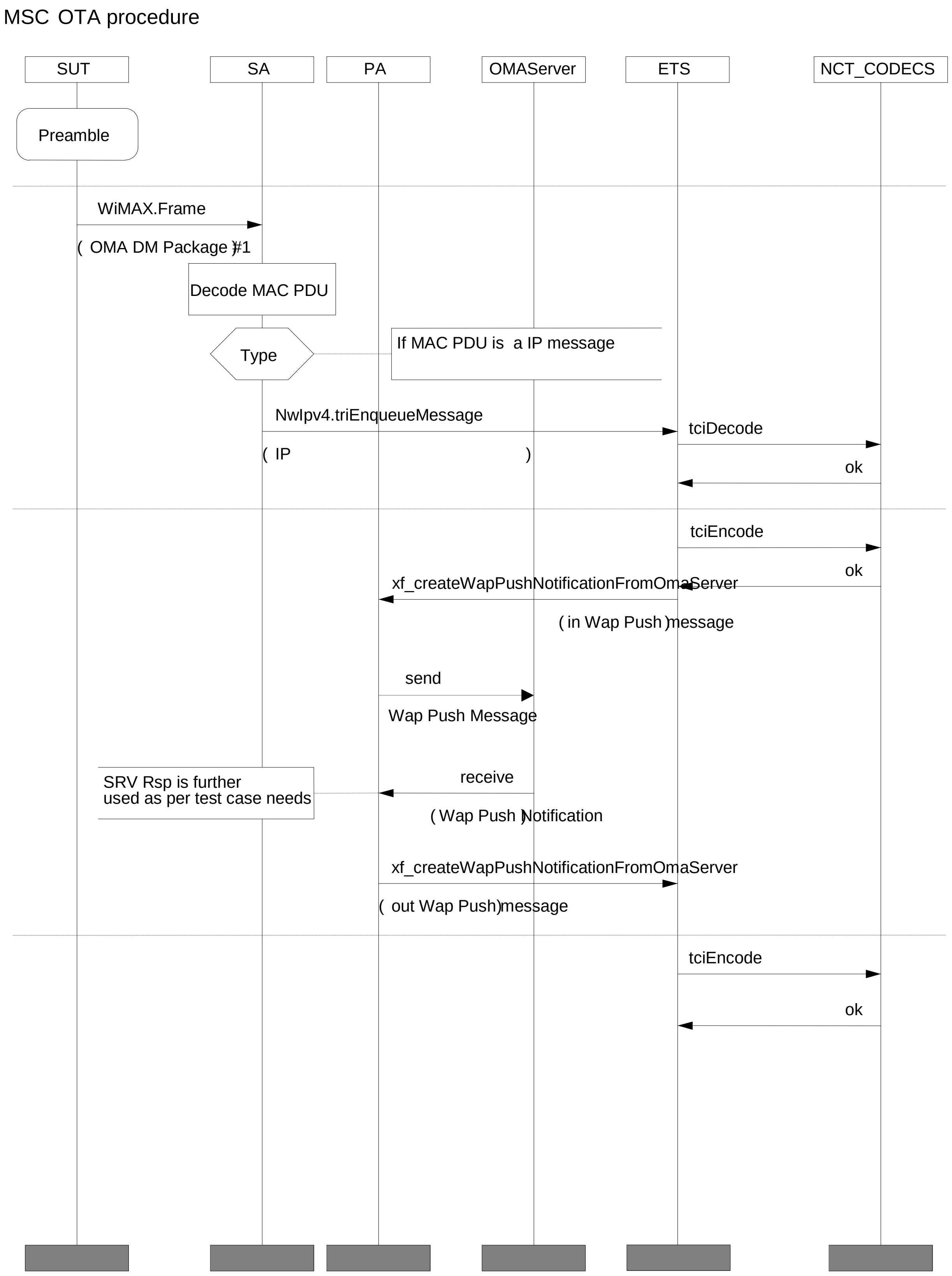


Figure 12

Test Step 1: the test case preamble consists in network entry procedure, initial service flows establishment and phase 1 of OTA procedure.

Test Step 2: the WiMAX message sent by the IUT is processed by the TA (i.e. module A). The device initiates an OMA DM session by sending the first package, which is decoded by the NCT CODECS and then available at the ETS level. Note that these messages are decoded as octetstring.

Test Step 3: ETS invokes the external function that sends the message from the IUT to the OMA server and receive the response message (Wap Push notification).

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

This ATS has been produced using the Testing and Test Control Notation (TTCN-3) according to ES 201 873-1 [36].

A.1 The TTCN-3 Module

The TTCN-3 code corresponding to the ATS is contained in an archive named ts_10262403v010201p0.zip which accompanies the present document.

Annex B (normative): WiMAX/HiperMAN NCT Partial PIXIT proforma for IUT MS

The PIXIT Proforma is based on ISO/IEC 9646-6 [41]. 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 D.

Annex C (normative): WiMAX/HiperMAN NCTR 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 NCTR proforma in this annex so that it can be used for its intended purposes and may further publish the completed NCTR.

The NCTR proforma is based on ISO/IEC 9646-6 [41]. Any needed additional information can be found in this International standard document.

C.1 Identification summary

C.1.1 Protocol conformance test report

Table C.1

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

C.1.2 IUT identification

Table C.2

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

C.1.3 Testing environment

Table C.3

PIXIT Number:	
ATS Specification:	
Abstract Test Method:	TS 102 624-3, clause 4
Means of Testing identification:	
Date of testing:	
Conformance Log reference(s):	
Retention Date for Log reference(s):	

C.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.

....

....

....

....

....

....

C.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.

....

....

....

....

....

....

C.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) and there are no "FAIL" verdicts to be recorded (in clause C.6) strike the words "has or", otherwise strike the words "or has not".

C.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.

C.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) strike the words "did or" otherwise strike the words "or did not".

Summary of the results of groups of test:

....

....

....

....

....

....

....

....

C.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.

....

....

....

....

....

....

....

....

C.6 Test campaign report

Table C.4: BS test cases

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause C.7)
TP/MS/NWE/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/NWE/BV-H0013	**Yes/No**	**Yes/No**		
TP/MS/NWE/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H004	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H005	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H006	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H007	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H008	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H009	**Yes/No**	**Yes/No**		
TP/MS/DHCP/BV-H010	**Yes/No**	**Yes/No**		
TP/MS/DHCP/TI-H000	**Yes/No**	**Yes/No**		
TP/MS/DHCP/TI-H001	**Yes/No**	**Yes/No**		
TP/MS/DHCP/TI-H002	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H004	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H005	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H006	**Yes/No**	**Yes/No**		
TP/MS/CMIPV4/BV-H007	**Yes/No**	**Yes/No**		
TP/MS/QoS/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/QoS/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/QoS/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/QoS/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/SEC/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/BV-H004	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/FRAG/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/ FRAG/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTLS/ FRAG/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPAKA/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPAKA/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPAKA/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPAKA/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPAKA/BV-H004	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPAKA/BV-H005	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H004	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H005	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H006	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H007	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H008	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/BV-H009	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/FRAG/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/EAPTTLSv0/FRAG/BV-H001	**Yes/No**	**Yes/No**		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause C.7)
TP/MS/SEC/EAPTTLSv0/FRAG/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/SEC/CMAC/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/SEC/CMAC/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/IPv6 /BV-H000	**Yes/No**	**Yes/No**		
TP/MS/IPv6 /BV-H001	**Yes/No**	**Yes/No**		
TP/MS/IPv6 /BV-H002	**Yes/No**	**Yes/No**		
TP/MS/IPv6 /BV-H003	**Yes/No**	**Yes/No**		
TP/MS/IPv6 /BV-H004	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H000	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H001	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H002	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H003	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H004	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H005	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H006	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H007	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H008	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H009	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H010	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H011	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H012	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H013	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H014	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H015	**Yes/No**	**Yes/No**		
TP/MS/CMIPv6/BV-H016	**Yes/No**	**Yes/No**		

C.7 Observations

Additional information relevant to the technical content of the MCTR is given here.

....

....

....

....

....

....

....

....

....

Annex D (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.
- PIXIT.
- External functions.
- TA (Test Adapter) Command and Responses.
- All test Cases Modules.

The HTML files are compressed in an archive named ts_10262403v010101p0.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 E (informative): Bibliography

IETF RFC 1035 (November 1987): "Domain Names - Implementation and Specification".

IETF RFC 4443 (March 2006): "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) specification".

ETSI ES 201 873-5: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI)".

WiMAX Forum (Release 1.5): "WiMAX Forum Network Architecture; Stage 3: Architecture, detailed Protocols and Procedures: WiMAX Over-The-Air General Provisioning System Specification".

WiMAX Forum (Release 1.5): "WiMAX Forum Network Architecture; Stage 3: Architecture, detailed Protocols and Procedures: Over-The-Air Provisioning & Activation Protocol based on TR-069 Specification".

WiMAX Forum (Release 1.5): "WiMAX Forum Network Architecture; Stage 3: Architecture, detailed Protocols and Procedures; WiMAX Over-The-Air Provisioning & Activation Protocol based on OMA DM Specifications".

History

Document history		
V1.1.1	March 2009	Publication
V1.2.1	November 2009	Publication