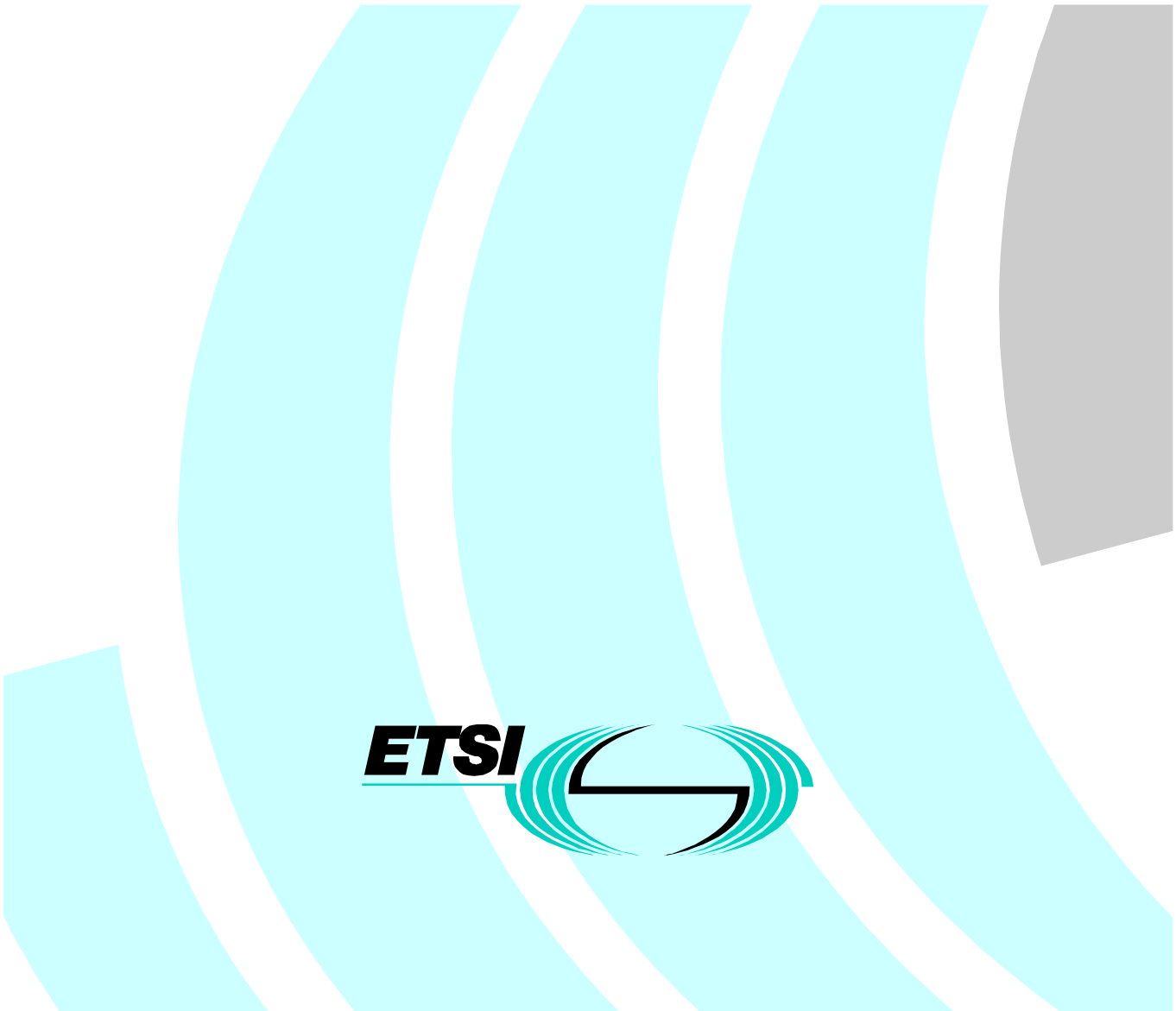
Draft EN 301 005-4 V1.1.1 (1999-06)

European Standard (Telecommunications series)

V interfaces at the digital Service Node (SN); Interfaces at the VB5.1 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 4: Abstract Test Suite (ATS) and partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma specification



Reference
DEN/SPS-09046-4 (9b100ico.PDF)

Keywords
AN, B-ISDN, ISDN, SN, PSTN, V interface, V5 interface, VB5 interface, ATS, PIXIT

ETSI

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - 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

Internet
secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your
comment to: editor@etsi.fr

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 SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

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 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 European Standard (Telecommunications series) has been produced by ETSI Technical Committee Signalling Protocols and Switching (SPS), and is now submitted for the Public Enquiry phase of the ETSI standards Two-step Approval Procedure.

The present document is part 4 of a multi-part EN covering the interfaces at the VB5.1 reference point as described below:

Part 1: "Interface specification";

Part 2: "Protocol Implementation Conformance Statement (PICS) proforma specification";

Part 3: "Test Suite Structure and Test Purposes (TSS&TP) specification";

Part 4: "Abstract Test Suite (ATS) and partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma specification".

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Introduction

General

The work on a new broadband VB reference point concept was initiated by ETSI Technical Committee SPS to consider possible new structures and reference points for the connection of new broadband and combined narrowband/broadband access arrangements to Service Nodes (SN), in co-operation with other TCs.

The VB5 reference point concept, based on ITU-T Recommendation G.902 [8], was split into two variants. The first variant based on an ATM cross-connect with provisioned connectivity, called the VB5.1 reference point, is described in the present document. The other variant which further enables on-demand connectivity within the AN, called the VB5.2 reference point, is covered by EN 301 217-1 [10].

Relationship between the VB5.1 and VB5.2 reference point concepts

VB5.2 extends the capabilities at the VB5.1 reference point to include on-demand connectivity in the AN under the control of SN. The major common features between the VB5.1 and VB5.2 interfaces are:

- both VB5 interfaces support B-ISDN as well as narrowband and other non-B-ISDN customer access types;
- both VB5 interfaces support ATM multiplexing/cross-connecting in the AN at the VP and/or VC level.

It is anticipated that the Real Time Management Co-ordination (RTMC) protocol for the VB5.1 reference point will be a subset of the RTMC protocol for the VB5.2 reference point.

1 Scope

The present document, part 4 of EN 301 005, specifies the Abstract Test Suite (ATS), the Abstract Test Method (ATM), ATS conventions, the partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma for testing the conformity of an implementation to the specification of interfaces at the VB5.1 reference point between an Access Network (AN) and a Service Node (SN). A proforma for the testing report (PCTR) is also included.

There are in fact two separate test suites, as well as two PCTR proformas, one version for testing the AN, the other one for testing the SN.

The ATS is based on the Test Suite Specification & Test Purposes specified in EN 301 005-3 [3].

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] EN 301 005-1 (V1.1): "V interfaces at the digital Service Node (SN); Interfaces at the VB5.1 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 1: Interface specification".

[2] EN 301 005-2 (V1.1): "V interfaces at the digital Service Node (SN); Interfaces at the VB5.1 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 2: Protocol Implementation Conformance Statement (ICS)proforma specification".

[3] EN 301 005-3 (V1.1): "V interfaces at the digital Service Node (SN); Interfaces at the VB5.1 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 3: Test Suite Structure and Test Purposes (TSS&TP) specification".

[4] ISO/IEC 9646-1: "Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[5] ISO/IEC 9646-2: "Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 2: Abstract test suite specification".

[6] ISO/IEC 9646-3: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 3: The Tree and Tabular Combined Notation (TTCN)".

[7] ISO/IEC 9646-7: "Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 7: Protocol Implementation Conformance Statement".

[8] ITU-T Recommendation G.902: "Framework Recommendation on functional access networks (AN) – Architecture and functions, access types, management and service node aspects".

[9] ITU-T Recommendation M.3010: "Principles for a Telecommunications management network".

[10] EN 301 217-1: "V interfaces at the digital Service Node (SN); Interfaces at the VB5.2 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 1: Interface specification".

3 Definitions and abbreviations

3.1 Definitions

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

- terms defined in EN 301 005-1 [1];
- terms defined in ISO/IEC 9646-1 [4] and in ISO/IEC 9646-2 [5].

In particular, the following terms defined in ISO/IEC 9646 apply:

Abstract Test Suite (ATS)
Abstract Test Method (ATM)
Implementation Conformance Statement (ICS)
Implementation Under Test (IUT)
Implementation eXtra Information for Testing (IXIT)
Lower Tester (LT)
PICS proforma
PIXIT proforma
Point of Control and Observation (PCO)
Protocol Implementation Conformance Statement (PICS)
Protocol Implementation eXtra Information for Testing (PIXIT)
Service Access Point (SAP)
Single Party Testing (SPyT)
System Under Test (SUT)
Upper Tester (UT)
TTCN.GR
TTCN.MP
Protocol Conformance Test Report (PCTR)
PCTR proforma

3.2 Abbreviations

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

AAL	ATM Adaptation Layer
AAL-SAP	AAL - Service Access Point
AN	Access Network
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband ISDN
B-ISUP	Broadband ISDN Signalling User Part
B-UNI	Broadband UNI
BA	Basic (rate) Access
CPE	Customer Premises Equipment
CPN	Customer Premises Network
ET	Equipment Terminal
FSM	Finite State Machine
ID	Identity
IE	Information Element
INI	Inter-Network Interface
ISDN	Integrated Services Digital Network
LAN	Local Area Network
LE	Local Exchange
LME	Layer Management Entity
LMI	Local Management Interface
LSP	Logical Service Port
LUP	Logical User Port
MIB	Management Information Base

MSC	Message Sequence Chart
N-ISDN	Narrowband ISDN
NNI	Network-to-Network Interface
OAM	Operations Administration and Maintenance
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Units
PSP	Physical Service Port
PSTN	Public Switched Telephone Network
PUP	Physical User Port
Q3	"Q" management interface reference point as ITU-T Recommendation M.3010 [9]
RTMC	Real Time Management Co-ordination
SAAL	Signalling ATM Adaptation Layer
SAP	Service Access Point
SAR	Segmentation and Reassembly
SDH	Synchronous Digital Hierarchy
SDL	Specification and Description Language
SDU	Service Data Units
SN	Service Node
SNI	Service Node Interface
SP	Service Port
SPS	Signalling Protocols and Switching
SSCF	Service Specific Co-ordination Function
SSCOP	Service Specific Connection Oriented Protocol
TC	Technical Committees
TE	Terminal Equipment
TMN	Telecommunication Management Network
TP	Transmission Path
UNI	User-Network Interface
VB	Broadband "V" reference point
VC	Virtual Channel (ATM)
VCE	Virtual Channel Entity
VP	Virtual Path
VPC	VP Connection
VPCI	VP Connection Identifier
VPCI-CC	VP Connection Identifier - Consistency Check
VPI	VP Identifier
VPL	VP Link

4 Test architecture

4.1 Abstract Test Method (ATM)

This subclause describes the Abstract Test Method (ATM) and the Point of Control and Observation (PCO) used to test the VB5.1 RTMC protocol for the AN and SN components.

The remote test method is used for VB5.1 RTMC conformance testing, since the VB5.1 implementations are not mandated to offer a direct access to the upper service boundary (i.e. to the "mee" service primitives). The co-ordination procedures can only be expressed in an informal way.

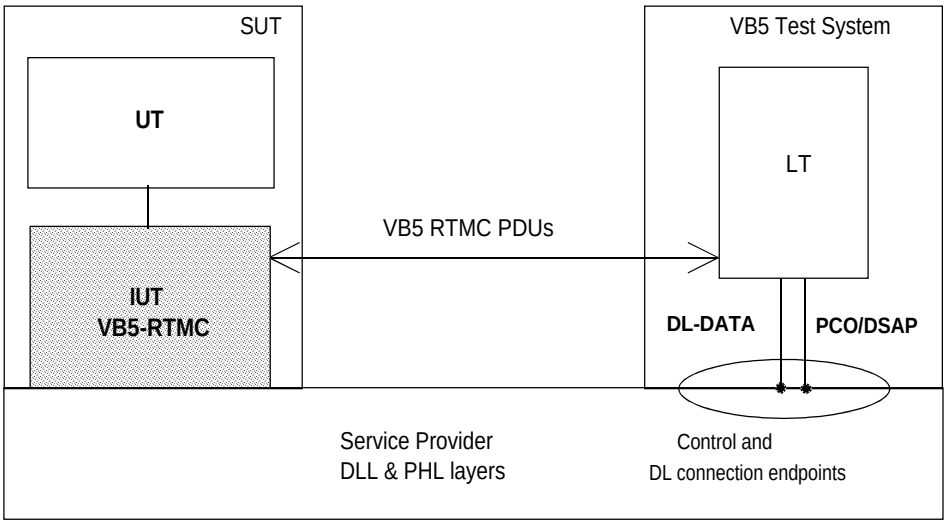


Figure 1: Remote single layer test method applied to the VB5.1 RTMC testing

LT: A Lower Tester (LT) is located in the VB5.1 test system. It controls and observes the behaviour of the IUT.

DSAP: A unique Data link Service Access Point (DSAP) is defined at the VB5.1 interface and commonly used for exchanging service data of the RTMC protocol functional entities.

PCO: The PCO for RTMC testing is located on the DSAP. All test events at the PCO are specified in terms of data link Abstract Service Primitives (ASPs) and network layer PDUs.

UT: No explicit Upper Tester (UT) exists in the test system. However, the SUT needs to carry out some UT functions to achieve some effects of test co-ordination procedures. Designing ATS, the capability of the system management functions, such as controls of the IUT, its interactions with the Q3 interface may be taken into account. The controls of the IUT will be implied or informally expressed in the ATS, but no assumption shall be made regarding their feasibility or realization. An example of such controls could be to provoke start up of the IUT through the Q3 interface.

VB5-RTMC: The PDU conveying this information will be transferred to and from the tester via a single, dedicated virtual connection.

4.2 Scope of test purposes and additional testing

Behaviour which cannot be fully controlled and observed via the RTMC protocol alone, such as the checking of completion of certain actions of the system on the managed resources, has been included, when appropriate, in individual VB5.1 test purposes in an additional requirement section.

Taking the example of blocking, a need may arise to verify that the status of a resource has actually been changed, as specified in the standard, by checking that it can no longer be allocated to a call. This may require the execution of a call attempt via the signalling protocols at a UNI or NNI. Such test purposes are not testable in the scope of the RTMC protocol alone. An architecture has to be defined for system level tests requiring the combination of two or more interfaces or protocols. Such tests are outside the scope of the ATS for RTMC. The approach taken is to append a short description of such requirements to the related RTMC test purposes.

The actual testing of these requirements can only be performed if additional means are provided to access internal data which cannot be interrogated via RTMC procedures. One way of performing this is to use an ad-hoc tester loaded into the SUT, if available. This is not typically the case. A generic approach is to consider that any system of the Network Element type comprises several interfaces and protocols, each of which is first tested at the individual protocol level. These tests typically have to leave out a number of requirements which are not testable within a single protocol.

A second hierarchical level of testing (see fig. B2) could cover a substantial number of such requirements by checking interactions between two or more protocols which have passed individual conformance testing. This is however outside the scope of the present testing standard.

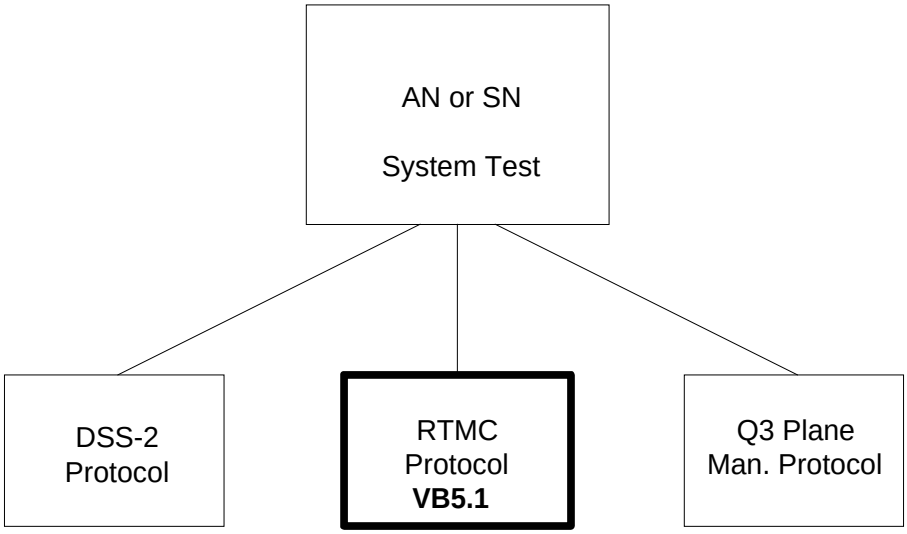


Figure 2: Example of hierarchical test architecture applied beyond VB5.1 testing

NOTE: The example of blocking described above is an illustration of such an approach. It could also be applied in extending the testing of the Q3 management protocol beyond the mere reception of an acknowledgement at the Q3 interface: the full execution a blocking request issued from the Operation System could be checked via DDS-2 to verify that the resource is actually blocked, and via RTMC to verify that real time co-ordination between AN and SN has taken place.

5 ATS conventions

This clause describes the conventions applied to define the ATS and gives the naming conventions chosen for the different elements of the ATS.

The ATS conventions are intended to give a better understanding of the ATS but they describe also the conventions made for the development of the ATS, thus for any later maintenance purposes or further development of the ATS, the conventions described in this clause shall be considered.

5.1 Naming conventions

5.1.1 Declarations part

This subclause describes the naming conventions chosen for the elements of the ATS declarations part.

5.1.1.1 Test suite type and structured type definitions "By Reference"

In order to avoid misalignment problems with the standard, all the types used in this ATS have been defined by reference. The same ASN1 file (RTMCMessages) containing the standard type definitions has been used as module containing the referenced type definitions.

EXAMPLE:

TypeName	*TypeReference*	*ModuleIdentifier*
AwaitClear	AwaitClear	RTMCMessages

5.1.1.2 Test suite constant declarations By Reference

In order to avoid misalignment problems with the standard, all the constants used in this ATS have been defined by reference. The same ASN.1 file RTMCMessages containing the standard constant definitions has been used as module containing the referenced type definitions.

EXAMPLE:

ConstantName	*Type*	Value Reference	*ModuleIdentifier*
connLUP	BIT_STRING	connLUP	RTMCMessages

5.1.1.3 Test suite parameter declarations

The test suite parameter identifiers are composed of strings in uppercase letters and separated by underscores.

If the test suite parameter references a PIXIT item, the prefix "PIX_" is used.

EXAMPLE: PIX_LSP_VPCI_1

5.1.1.4 Test case selection expression definitions

The test case selection expressions are based on PICS or PIXIT test suite parameter definitions.

EXAMPLE: PIX_automatic_loopback_results

5.1.1.5 Test suite variable declarations

The test suite variable identifiers are composed of string in lowercase letters starting by the lowercase string "tsv_".

EXAMPLE: TSV_transId

5.1.1.6 Timer declarations

Two kinds of timers can be distinguished:

1) standardized:

   Those defined in the standard, e.g. T_block., use the same name as in the standard, beginning with a capital "T". their duration is defined in the PIXIT under same name, for example PIX_T_block.

2) for testing supervision:

   In order to control the duration of individual test related to timeout of standardized timer tests, a specific timer and its duration , PIX_T_supervision, is defined.

5.1.1.7 ASP type definitions

ASP definitions follow the specification when a corresponding definition exists. For example RESET_RSC is used. If not specified, a free name is used.

EXAMPLE:

msg_too_short.

5.1.1.8 Alias definitions

No alias definitions are used in the test suite.

5.1.2 Constraints naming

This subclause describes the naming conventions chosen for the elements of the ATS constraints part.

Constraint identifiers commence with lowercase c, followed by the ASP_TYPE, possibly followed by other mnemonics which complete its definition, like LSP or LUP when based on a some PDU, followed by a numerical value like 001, 002, etc.

When the constraint is based on a PDU, the transaction Id value contained in the PDU is considered as a variable and this value is not taken into account for the definition of the constraint name.

Table 1

Type Definition	Constraint Definition Example
UNBLOCK_RSC	c_UNBLOCK_RSC_LSP_002
UNBLOCK_RSC	c_UNBLOCK_RSC_LUP_004
REQ_LSPID	c_REQ_LSPID_002

5.1.3 Dynamic part

This subclause describes the naming conventions chosen for the elements of the ATS dynamic part.

5.1.3.1 Test case identifier

There is a one-to-one mapping between the Test Purposes and the test cases. The identifier of a TC and its corresponding Test Purpose is described in Table 2:

Table 2: TP identifier naming convention scheme

Identifier: **VB5_<i>_<IUT>_<pp>_<cc>_<nn>**

VB5 VB5.1 reference point specification

<i> RTMC protocol at interface

<IUT> AN Access Network is the IUT
SN Service Node is the IUT

<pp> = procedure identifier like

ST	StartUp
BR	Blocking Resource
UR	Unblocking Resource
SR	Shut down Resource
CC	VPCI Consistency Check
VLS	Verify LSPId (SN initiated)
VLA	Verify LSPId (AN initiated)
RLS	Reset LSPId (SN initiated)
RLA	Reset LSPId (AN initiated)
RVS	Reset VPC (SN initiated)
RVA	Reset VPC (AN initiated)
CEH	Common Error Handling

<cc> = test category:

CA	Capability tests
BV	Valid Behaviour tests
BI	Invalid Behaviour tests
BO	Inopportune Behaviour tests
TI	Timer tests

<nn> = sequential number: (01-99)

Example of test case name: **VB5_RTMC_AN_CC_BV_02**

5.1.3.2 Preamble identifier

A preamble defines a set of TTCN statements needed to initialize the IUT then carry the IUT to a particular state or a particular call configuration situation from which the test case shall start. Preambles are called StartupAN or StartupSN. In addition, a StartbuSN is defined when the test requires to set a specific VPCI busy prior to the test.

5.1.3.3 Postamble identifier

There is no postamble defined, as each new test is beginning by a complete reset, it cannot be corrupted by side effects of the previous executed test.

5.1.3.4 Default identifier

A default is a set of TTCN statements called here OtherwiseFail, which is used to complete the behaviour tree related to the test case. It covers some different alternatives bringing the test case to a FAIL or INCONCLUSIVE verdict.

Annex A (normative): ATS for VB5.1 interface

This ATS has been produced using the Tree and Tabular Combined Notation (TTCN) according to ISO/IEC 9646-3 [6].

The ATS was developed on a separate TTCN software tool and therefore the TTCN tables are not completely referenced in the table of contents. The ATS itself contains a test suite overview part which provides additional information and references.

A.1 The TTCN Graphical form (TTCN.GR)

The TTCN.GR representation of this ATS is contained in Adobe Portable Document Format™ files (AN_Suite.PDF and SN_Suite.PDF contained in archive 9b100ico.ZIP) which accompanies the present document.

A.2 The TTCN Machine Processable form (TTCN.MP)

The TTCN.MP representation corresponding to this ATS is contained in ASCII files (AN_Suite.MP and SN_Suite.MP contained in archive 9b100ico.ZIP) which accompanies the present document.

NOTE: According to ISO/IEC 9646-3 [6], in case of a conflict in interpretation of the operational semantics of TTCN.GR and TTCN.MP, the operational semantics of the TTCN.GR representation takes precedence.

Annex B (normative): Partial PIXIT proforma for VB5.1 interface

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

B.1 Identification summary

Table B.1

PIXIT number:	
Test laboratory name:	
Date of issue:	
Issued to:	

B.2 ATS summary

Table B.2

Protocol specification:	EN 301 005-1
Protocol to be tested:	
ATS specification:	EN 301 005-4
Abstract test method:	Remote test method

B.3 Test laboratory

Table B.3

Test laboratory identification:	
Test laboratory manager:	
Means of testing:	
SAP address:	

B.4 Client identification

Table B.4

Client identification:	
Client test manager:	
Test facilities required:	

B.5 SUT

Table B.5

Name:	
Version:	
SCS number:	
Machine configuration:	
Operating system identification:	
IUT identification:	
PICS reference for IUT:	
Limitations of the SUT:	
Environmental conditions:	

B.6 Protocol layer information

B.6.1 Protocol identification

Table B.6

Name:	EN 301 005-1: VB5.1 interface
Version:	
PICS references:	EN 301 005-2

B.6.2 IUT information

B.6.2.1 Implicit send events

Table B.7: Implicit send events

Item	PIXIT (See note)	Related implicit send message (PDU)	Invocation description
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
NOTE:			

B.6.2.2 Parameter values

Table B.8: Parameter values

Item	Parameter name	Parameter type (ASN1)	Explanation	Value in MSCs
1.	PIX_LUP_address	LogId	LUP of PUP01	
2.	PIX_LUP_unkaddr	LogId	Unknown LUP of PUP01	
3.	PIX_LUP_VPCI_1	VPCI	First VPCI of LUP01 of PUP01	
4.	PIX_LUP_VPCI_2	VPCI	2nd VPCI of LUP01 of PUP01	
5.	PIX_LUP_VPCI_unk	VPCI	unknown VPCI of LUP01 of PUP01	
6.	PIX_LUP_VPCI_busy	VPCI	Busy VPCI of LUP01 of PUP01	
7.	PIX_LUP VPCI_last	VPCI	Last VPCI of LUP01 of PUP01	
8.	PIX_LSP_address	LogId	LSP id	
9.	PIX_LSP_unkaddr	LogId	unknown LSP id	
10.	PIX_LSP_VPCI_1	VPCI	First VPCI of LSP81 of PSP81	
11.	PIX_LSP_VPCI_2	VPCI	2nd VPCI of LSP	
12.	PIX_LSP_VPCI_unk	VPCI	unknown VPCI of LSP	
13.	PIX_LSP_VPCI_busy	VPCI	Busy VPCI of LSP	
14.	PIX_LSP VPCI_last	VPCI	Last VPCI of LSP	
15.	PIX_automatic_loopback_results	Boolean	choice dictated by implementation	
	PIX_T_acl	integer	Timer value	
	PIX_T_block	integer	Timer value	
	PIX_T_consreq	integer	Timer value	
	PIX_T_consend	integer	Timer value	
	PIX_T_lspid	integer	Timer value	
	PIX_T_reset	integer	Timer value	
	PIX_T_start	integer	Timer value	
	PIX_T_unblock	integer	Timer value	
	PIX_T_supervision	integer	supervision timer	

Annex C (normative): Protocol Conformance Test Report (PCTR) proforma when testing AN

The PCTR Proforma is based on ISO/IEC 9646-6 [7]. Any additional information needed can be found in the present document.

C.1 Identification summary

C.1.1 Protocol conformance test report

Table C.1

PCTR number:	
PCTR 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:	
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 in this report 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 of this report 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 subclause itemizes the mismatches between the PICS and the static conformance requirements of the specified protocol specification.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

C.6 Test campaign report

Table C.4

ATS reference	Selected	Run	Verdict	Observations
VB5_RTMC_AN_ST_CA_01	Yes/No	Yes/No		
VB5_RTMC_AN_ST_CA_02	Yes/No	Yes/No		
VB5_RTMC_AN_ST_CA_03	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_02	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_03	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_04	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_05	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_06	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_07	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_08	Yes/No	Yes/No		
VB5_RTMC_AN_BR_TI_01	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_11	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_12	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_13	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_14	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_15	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_16	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_17	Yes/No	Yes/No		
VB5_RTMC_AN_BR_BV_18	Yes/No	Yes/No		
VB5_RTMC_AN_BR_TI_11	Yes/No	Yes/No		
VB5_RTMC_AN_UR_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_UR_BV_02	Yes/No	Yes/No		
VB5_RTMC_AN_UR_BV_03	Yes/No	Yes/No		
VB5_RTMC_AN_UR_TI_01	Yes/No	Yes/No		
VB5_RTMC_AN_UR_BV_11	Yes/No	Yes/No		
VB5_RTMC_AN_UR_BV_12	Yes/No	Yes/No		
VB5_RTMC_AN_UR_BV_13	Yes/No	Yes/No		
VB5_RTMC_AN_UR_TI_11	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_02	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_03	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_04	Yes/No	Yes/No		
VB5_RTMC_AN_SR_TI_01	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_11	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_12	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_13	Yes/No	Yes/No		
VB5_RTMC_AN_SR_BV_14	Yes/No	Yes/No		
VB5_RTMC_AN_SR_TI_11	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BI_01	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BV_02	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BO_01	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BI_02	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BV_11	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BV_12	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BO_11	Yes/No	Yes/No		
VB5_RTMC_AN_CC_BI_12	Yes/No	Yes/No		
VB5_RTMC_AN_VLS_BV_01	Yes/No	Yes/No		

ATS reference	Selected	Run	Verdict	Observations
VB5_RTMC_AN_VLA_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_VLA_TI_01	Yes/No	Yes/No		
VB5_RTMC_AN_RLS_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_RLS_BI_01	Yes/No	Yes/No		
VB5_RTMC_AN_RLA_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_RLA_TI_01	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BV_02	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BI_01	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BI_02	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BI_03	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BV_11	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BV_12	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BI_11	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BI_12	Yes/No	Yes/No		
VB5_RTMC_AN_RVS_BI_13	Yes/No	Yes/No		
VB5_RTMC_AN_RVA_BV_01	Yes/No	Yes/No		
VB5_RTMC_AN_RVA_BV_02	Yes/No	Yes/No		
VB5_RTMC_AN_RVA_TI_01	Yes/No	Yes/No		
VB5_RTMC_AN_RVA_BV_11	Yes/No	Yes/No		
VB5_RTMC_AN_RVA_BV_12	Yes/No	Yes/No		
VB5_RTMC_AN_RVA_TI_11	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_01	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_02	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_03	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_04	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_05	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_06	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_07	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_11	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_12	Yes/No	Yes/No		
VB5_RTMC_AN_CEH_BI_13	Yes/No	Yes/No		

C.7 Observations

Additional information relevant to the technical content of the PCTR are given here.

.....
.....
.....
.....
.....
.....
.....
.....

Annex D (normative): Protocol Conformance Test Report (PCTR) proforma when testing SN

The PCTR Proforma is based on ISO/IEC 9646-6 [7]. Any additional information needed can be found in the present document.

D.1 Identification summary

D.1.1 Protocol conformance test report

Table D.1

PCTR number:	
PCTR date:	
Corresponding SCTR number:	
Corresponding SCTR date:	
Test laboratory identification:	
Test laboratory manager:	
Signature:	

D.1.2 IUT identification

Table D.2

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

D.1.3 Testing environment

Table D.3

PIXIT number:	
ATS specification:	
Abstract test method:	
Means of testing identification:	
Date of testing:	
Conformance log reference(s):	
Retention date for log reference(s):	

D.1.4 Limits and reservation

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

.....

.....

.....

.....

.....

D.1.5 Comments

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

.....

.....

.....

.....

.....

D.2 IUT conformance status

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

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

D.3 Static conformance summary

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

Strike the appropriate words in this sentence.

D.4 Dynamic conformance summary

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

Strike the appropriate words in this sentence. If there are no "FAIL" verdicts to be recorded in clause D.6 of this report strike the words "did or" otherwise strike the words "or did not".

Summary of the results of groups of test:

.....

.....

.....

.....

.....

D.5 Static conformance review report

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

D.6 Test campaign report

Table D.4

ATS reference	Selected	Run	Verdict	Observations
VB5_RTMC_SN_ST_CA_01	Yes/No	Yes/No		
VB5_RTMC_SN_ST_CA_02	Yes/No	Yes/No		
VB5_RTMC_SN_ST_CA_03	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_02	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_03	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_04	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_05	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_06	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_07	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BI_02	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BI_03	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_11	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_12	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_13	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_14	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_15	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_16	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BV_17	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BI_11	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BI_12	Yes/No	Yes/No		
VB5_RTMC_SN_BR_BI_13	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BV_02	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BV_03	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BI_02	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BI_03	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BV_11	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BV_12	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BV_13	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BI_11	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BI_12	Yes/No	Yes/No		
VB5_RTMC_SN_UR_BI_13	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_02	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_03	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_04	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_05	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_06	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_07	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_08	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_SR_TI_01	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_11	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_12	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_13	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_14	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_15	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_16	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BV_17	Yes/No	Yes/No		

ATS reference	Selected	Run	Verdict	Observations
VB5_RTMC_SN_SR_BV_18	Yes/No	Yes/No		
VB5_RTMC_SN_SR_BI_11	Yes/No	Yes/No		
VB5_RTMC_SN_SR_TI_11	Yes/No	Yes/No		
VB5_RTMC_SN_CC_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_CC_BV_02	Yes/No	Yes/No		
VB5_RTMC_SN_CC_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_CC_TI_01	Yes/No	Yes/No		
VB5_RTMC_SN_CC_TI_02	Yes/No	Yes/No		
VB5_RTMC_SN_VLS_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_VLS_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_VLS_TI_01	Yes/No	Yes/No		
VB5_RTMC_SN_VLA_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_RLS_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_RLS_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_RLS_TI_01	Yes/No	Yes/No		
VB5_RTMC_SN_RLA_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_RLA_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_RVS_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_RVS_BV_02	Yes/No	Yes/No		
VB5_RTMC_SN_RVS_TI_01	Yes/No	Yes/No		
VB5_RTMC_SN_RVS_BV_11	Yes/No	Yes/No		
VB5_RTMC_SN_RVS_BV_12	Yes/No	Yes/No		
VB5_RTMC_SN_RVS_TI_11	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BV_01	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BV_02	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BI_02	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BV_11	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BV_12	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BI_11	Yes/No	Yes/No		
VB5_RTMC_SN_RVA_BI_12	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_01	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_02	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_03	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_04	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_05	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_06	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_07	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_11	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_12	Yes/No	Yes/No		
VB5_RTMC_SN_CEH_BI_13	Yes/No	Yes/No		

D.7 Observations

Additional information relevant to the technical content of the PCTR are given here.

.....

.....

.....

.....
.....
.....
.....
.....
.....

Annex E (normative): ASN.1 data definitions for RTMC protocol

```
RTMCMessages

DEFINITIONS ::=

BEGIN

 --definition of RTMC messages

MessageTypes ::= CHOICE --Table 37
 {  awaitClear         [1] AwaitClear,
    awaitClearAck      [2] AwaitClearAck,
    awaitClearComp     [3] AwaitClearComp,
    awaitClearCompAck  [4] AwaitClearCompAck,
    blockRsc           [5] BlockRsc,
    blockRscAck        [6] BlockRscAck ,
    consCheckEnd       [7] ConsCheckEnd ,
    consCheckEndAck    [8] ConsCheckEndAck ,
    consCheckReq       [9] ConsCheckReq ,
    consCheckReqAck   [10] ConsCheckReqAck ,
    lSPId             [11] LSPId ,
    protocolError     [12] ProtocolError ,
    reqLSPId          [13] ReqLSPId ,
    resetRsc          [14] ResetRsc ,
    resetRscAck       [15] ResetRscAck ,
    unblockRsc        [16] UnblockRsc ,
    unblockRscAck     [17] UnblockRscAck ,
    unknownMessage    [18] UnknownMessage
 }

AwaitClear ::= SEQUENCE --Table 40
 {  commonMsgInfo        CommonMsgInfo,
    rscList              RscListIEs
 }

AwaitClearAck ::= SEQUENCE --Table 41
 {  commonMsgInfo        CommonMsgInfo,
    result               ResultIE,
    rscList              RscListIEs  OPTIONAL
 }

AwaitClearComp ::= SEQUENCE --Table 42
 {  commonMsgInfo        CommonMsgInfo,
    rscList              RscListIEs
 }

AwaitClearCompAck ::= SEQUENCE
 {  commonMsgInfo        CommonMsgInfo
 }

BlockRsc ::= SEQUENCE --Table 38
 {  commonMsgInfo        CommonMsgInfo,
    blRscList            BlRscListIEs
 }

BlockRscAck ::= SEQUENCE --Table 39
 {  commonMsgInfo        CommonMsgInfo,
    result               ResultIE,
    rscList              RscListIEs  OPTIONAL
 }

ConsCheckEnd ::= SEQUENCE --Table 47
 {  commonMsgInfo        CommonMsgInfo,
    rscId                RscIdIE
 }

ConsCheckEndAck ::= SEQUENCE --Table 48
 {  commonMsgInfo        CommonMsgInfo,
    result               ResultIE,
    rscId                RscIdIE  OPTIONAL
 }

ConsCheckReq ::= SEQUENCE --Table 45
 {  commonMsgInfo        CommonMsgInfo,
    rscId                RscIdIE
 }

ConsCheckReqAck ::= SEQUENCE --Table 46
```

```
 {  commonMsgInfo        CommonMsgInfo,
    result               ResultIE,
    rscId                RscIdIE  OPTIONAL
 }

LSPId ::= SEQUENCE --Table 49
 {  commonMsgInfo        CommonMsgInfo,
    rscId                RscIdIE  OPTIONAL
 }

ProtocolError ::= SEQUENCE --Figure 71
 {  commonMsgInfo        CommonMsgInfo,
    protErrCause         ProtErrCauseIE  OPTIONAL
 }

ReqLSPId ::= SEQUENCE
 {  commonMsgInfo        CommonMsgInfo
 }

ResetRsc ::= SEQUENCE --Table 50
 {  commonMsgInfo        CommonMsgInfo,
    rscList              RscListIEs
 }

ResetRscAck ::= SEQUENCE --Table 51
 {  commonMsgInfo        CommonMsgInfo,
    result               ResultIE,
    rscList              RscListIEs  OPTIONAL
 }

UnblockRsc ::= SEQUENCE
 {  commonMsgInfo        CommonMsgInfo,
    rscList              RscListIEs
 }

UnblockRscAck ::= SEQUENCE
 {  commonMsgInfo        CommonMsgInfo,
    result               ResultIE,
    rscList              RscListIEs  OPTIONAL
 }

UnknownMessage ::= SEQUENCE
 {  commonMsgInfo        CommonMsgInfo
 }

CommonMsgInfo ::= SEQUENCE
 {  protDiscr            ProtDiscr DEFAULT '49'H,
    transId              TransId,
    msgType              MsgType,
    msgCompatInd         MsgCompatInd,
    msgLength            MsgLength
 }

TransId ::= SEQUENCE      --Transaction Identifier
 {  sparebits4          [1] SpareBits4 DEFAULT '0000'B,
    lengthOfTransIdVal  [2] BIT STRING (SIZE (4)) DEFAULT '0011'B,
    transIdFlag         [3] TransIdFlag,
    transIdVal          [4] TransIdVal
 }

MsgCompatInd ::= SEQUENCE
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    spareBits2          [2] SpareBits2 DEFAULT '00'B,
    compatFlag          [3] CompatFlag,
    spareBits21         [4] SpareBits2 DEFAULT '00'B,
    msgActInd           [5] MsgActInd
 }

 --definition of RTMC message parameters

BlRscListIEs ::= SEQUENCE --RTMC Blocked Resource Identifier List
 {  repeatInd            [1] RepeatIndIE  OPTIONAL,
    multBlRscId          [2] SEQUENCE OF BlRscIdIE
 }

RscListIEs ::= SEQUENCE --RTMC Resource Identifier List
 {  repeatInd            [1] RepeatIndIE  OPTIONAL,
    multRscId            [2] SEQUENCE OF RscIdIE
 }

BlRscIdIE ::= SEQUENCE --RTMC Blocked Resource Identifier
 {  commonIEInfo          CommonIEInfo,
    blRscIdContent        BlRscIdContent
```

```
 }

ProtErrCauseIE ::= SEQUENCE --RTMC Protocol Error Cause
 {  commonIEInfo         CommonIEInfo,
    protErrCause         ProtErrCauseContent
 }

RepeatIndIE ::= SEQUENCE --RTMC Repeat Indicator
 {  commonIEInfo         CommonIEInfo,
    repeatIndOctet       RepeatIndOctet
 }

ResultIE ::= SEQUENCE --RTMC Result Indicator
 {  commonIEInfo         CommonIEInfo,
    resultIndOctet       ResultIndOctet
 }

RscIdIE ::= SEQUENCE --RTMC Resource Identifier
 {  commonIEInfo         [1] CommonIEInfo,
    rscIdContent         [2] RscIdContent
 }

CommonIEInfo ::= SEQUENCE
 {  iEType               IEType,
    iECompatInd          IECompatInd,
    iELength             IELength
 }

IECompatInd ::= SEQUENCE
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    spareBits2          [2] SpareBits2 DEFAULT '00'B,
    compatFlag          [3] CompatFlag,
    spareBits1          [4] SpareBits1 DEFAULT '0'B,
    iEActInd            [5] IEActInd
 }

CompatFlag ::= BIT STRING (SIZE(1)) --Table 35
  compatFlagNotSignif     CompatFlag ::= '0'B
  compatFlagExplicit      CompatFlag ::= '1'B

MsgActInd ::= BIT STRING (SIZE(2)) --Table 35
  msgActIndReject           MsgActInd ::= '00'B --reject by PROTO_ERROR msg
  msgActIndDiscardAndIgnore MsgActInd ::= '01'B --discard  and ignore
  msgActIndDiscardAndReport MsgActInd ::= '10'B --not used by this VB5.1 version
  --All other values are reserved

IEActInd ::= BIT STRING (SIZE (3)) --Table 36
  iEActIndReject                  IEActInd ::= '000'B
  iEActIndDiscardAndProceed       IEActInd ::= '001'B
  iEActIndDiscardProceedAndReport IEActInd ::= '010'B
  iEActIndDiscardMsgAndIgnore     IEActInd ::= '101'B
  iEActIndDiscardMsgAndReport     IEActInd ::= '110'B
  --All other values are reserved

ExtBit ::= BIT STRING (SIZE(1))
  extBitContinues   ExtBit ::= '0'B --another octet of this group follows
  extBitLast        ExtBit ::= '1'B --last octet in group

BlRscIdContent ::= SEQUENCE --RTMC blocked resource identifier
 {  blReasonIndOctet     [1] BlReasonIndOctet,
    blRscIndOctet        [2] BlRscIndOctet,
    logRsc               [3] LogId,
    vpci1                [4] VPCI  OPTIONAL,
    vpci2                [5] VPCI  OPTIONAL
 }

RscIdContent ::= SEQUENCE --RTMC resource identifier
 {  rscIndOctet          [1] RscIndOctet,
    logRsc               [2] LogId,
    vpci1                [3] VPCI  OPTIONAL,
    vpci2                [4] VPCI  OPTIONAL
 }

LogId ::= OCTET STRING (SIZE (3))     --logical port identifier
  --LogicalServicePortId  (0 .. 16777215)
  --LogicalUserPortId     (0 .. 16777215)

ProtErrCauseContent ::= SEQUENCE
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    protErrCauseVal     [2] ProtErrCauseVal,
    protErrCauseDiagn   [3] ProtErrCauseDiagn OPTIONAL
 }
```

```
RepeatIndOctet ::= SEQUENCE
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    spareBits3          [2] SpareBits3 DEFAULT '000'B,
    repeatInd           [3] RepeatInd
 }

RepeatInd ::= BIT STRING (SIZE (4))
  repeatIndMultList       RepeatInd ::= '0011'B

ResultIndOctet ::= SEQUENCE --figure 73
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    spareBits3          [2] SpareBits3 DEFAULT '000'B,
    resultInd           [3] ResultInd     --result indication values
 }

ResultInd ::= BIT STRING (SIZE(4)) --Table 57
  rTMCSuccess             ResultInd ::= '0000'B --accepted or successful
  rTMCUnkRsc              ResultInd ::= '0001'B --resource(s) unknown
  rTMCOpRej               ResultInd ::= '0010'B --operation rejected
  rTMCOpFail              ResultInd ::= '0011'B --operation failed
  rTMCNotPerf             ResultInd ::= '0100'B --operation not performed

BlReasonIndOctet ::= SEQUENCE --Blocking reason indicator octet
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    errReason           [2] ErrReason, --RTMC error reason value
    admReason           [3] AdmReason  --RTMC administrative reason value
 }

ErrReason ::= BIT STRING (SIZE (2))
  rTMCErrNone             ErrReason ::= '00'B
  rTMCErr                 ErrReason ::= '01'B

AdmReason ::= BIT STRING (SIZE (5)) --table 54 Blocked resource id IE
  rTMCAdmNone             AdmReason ::= '00000'B
  rTMCAdmPart             AdmReason ::= '00001'B
  rTMCAdmFull             AdmReason ::= '00010'B

BlRscIndOctet ::= SEQUENCE
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    spareBits4          [2] SpareBits4 DEFAULT '0000'B,
    rscInd              [3] RscInd
 }

RscIndOctet ::= SEQUENCE
 {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
    unkRscInd           [2] UnkRscInd,
    aCCTreatment        [3] ACCTreatment,
    spareBits1          [4] SpareBits1 DEFAULT '0'B,
    rscInd              [5] RscInd
 }

RscInd ::= BIT STRING (SIZE (3)) --RTMC resource indicator values Table 58 octet N+ 5
  compLSP                 RscInd ::= '000'B
  connLSP                 RscInd ::= '001'B  --connection(s) at LSP
  connLUP                 RscInd ::= '101'B  --connection(s) at LUP

ACCTreatment ::= BIT STRING (SIZE(1)) --'1' if resource is used for on-demand connections at SN

UnkRscInd ::= BIT STRING (SIZE(2))       --unknown resource indicator
  logIdUnk                UnkRscInd ::= '00'B   --LSP Id or LUP Id unknown
  vpciUnk                 UnkRscInd ::= '11'B   --vpciUnknown

VPCI ::= OCTET STRING (SIZE (2))

ProtErrCauseVal ::= BIT STRING (SIZE (7))
  unkMsgType                 ProtErrCauseVal ::= '0000001'B
  mandIEMiss                 ProtErrCauseVal ::= '0000011'B
  unrecogIE                  ProtErrCauseVal ::= '0000100'B
  iEContErr                  ProtErrCauseVal ::= '0000101'B
  iENotAllowed               ProtErrCauseVal ::= '0000110'B
  msgNotCompatWithProtState  ProtErrCauseVal ::= '0000111'B
ProtErrCauseDiagn ::= OCTET STRING (SIZE (2))

ExtBitLast ::= BIT STRING (SIZE (1)) --Always defaulted to '1'B (last octet in group)

SpareBits4 ::= BIT STRING (SIZE(4))    --Always defaulted to '0000'B
SpareBits3 ::= BIT STRING (SIZE(3))    --Always defaulted to '000'B
SpareBits2 ::= BIT STRING (SIZE(2))    --Always defaulted to '00'B
SpareBits1 ::= BIT STRING (SIZE(1))    --Always defaulted to '0'B

ProtDiscr ::= OCTET STRING (SIZE(1))  --Always defaulted to '049'H

TransIdFlag ::= BIT STRING (SIZE(1))
```

```
TransIdVal ::= BIT STRING (SIZE (23))

MsgType ::= OCTET STRING (SIZE (1))

MsgLength ::= OCTET STRING (SIZE (2))

IEType ::= OCTET STRING (SIZE (1))

IELength ::= OCTET STRING (SIZE (2))

 -- definition of mee primitives

PrimitiveAnSn ::= CHOICE --RTMC primitive type definitions common for AN and SN
 {  meeErrorInd         [1]  MEEErrorInd,
    meeLspFailureInd    [2]  MEELspFailureInd,
    meeResetRscReq      [3]  MEEResetRscReq,
    meeResetRscConf     [4]  MEEResetRscConf,
    meeResetRscInd      [5]  MEEResetRscInd,
    meeStartupReq       [6]  MEEStartupReq,
    meeStartupConf      [7]  MEEStartupConf,
    meeStartupInd       [8]  MEEStartupInd,
    meeStopTrafficReq   [9]  MEEStopTrafficReq,
    meeStopTrafficConf [10]  MEEStopTrafficConf,
    meeVerifyLspIdReq  [11]  MEEVerifyLspIdReq,
    meeVerifyLspIdConf [12]  MEEVerifyLspIdconf,
    meeVpcCreateReq    [13]  MEEVPCCreateReq,
    meeVpcCreateConf   [14]  MEEVPCCreateConf,
    meeVpcDeleteReq    [15]  MEEVPCDeleteReq,
    meeVpcDeleteConf   [16]  MEEVPCDeleteConf
}

PrimitiveAn ::= CHOICE --RTMC primitive type definitions specific for AN
 {  meeAwaitClearConf   [1]  MEEAwaitClearConf,
    meeAwaitClearReq    [2]  MEEAwaitClearReq,
    meeBlockRscReq      [3]  MEEBlockRscReq,
    meeBlockRscConf     [4]  MEEBlockRscConf,
    meeConsCheckInd     [5]  MEEConsCheckInd,
    meeConsCheckRes     [6]  MEEConsCheckRes,
    meeConsCheckEndInd  [7]  MEEConsCheckEndInd,
    meeConsCheckEndRes  [8]  MEEConsCheckEndRes,
    meeUnblockRscReq    [9]  MEEUnblockRscReq,
    meeUnblockRscConf  [10]  MEEUnblockRscConf
 }

PrimitiveSn ::= CHOICE --RTMC primitive type definitions specific for SN
 {  meeAwaitClearInd    [1]  MEEAwaitClearInd,
    meeAwaitClearRes    [2]  MEEAwaitClearRes,
    meeBlockRscInd      [3]  MEEBlockRscInd,
    meeConsCheckReq     [4]  MEEConsCheckReq,
    meeConsCheckConf    [5]  MEEConsCheckConf,
    meeConsCheckEndReq  [6]  MEEConsCheckEndReq,
    meeConsCheckEndConf [7]  MEEConsCheckEndConf,
    meeUnblockRscInd    [8]  MEEUnblockRscInd
 }

MEEAwaitClearConf ::= SEQUENCE
 {  result              [1]  EnvResult,
    rscList             [2]  RscList OPTIONAL  --resources with shutting down completed or unknown
resources
 }

MEEAwaitClearInd ::= SEQUENCE
 {  rscList             RscList
 }

MEEAwaitClearReq ::= SEQUENCE
 {  rscList             RscList
 }

MEEAwaitClearRes ::= SEQUENCE
 {  rscList             RscList
 }

MEEBlockRscConf ::= SEQUENCE
 {  result              EnvResult,
    unkRscList          RscList  OPTIONAL --reporting of unknown resources
 }

MEEBlockRscInd ::= SEQUENCE
 {  blRscList           BlRscList
 }

MEEBlockRscReq ::= SEQUENCE
```

```
 {  blRscList            BlRscList
 }

MEEConsCheckConf ::= SEQUENCE
 {  result               EnvResult,
    unkRscId             RscId  OPTIONAL --reporting of unknown resources
 }

MEEConsCheckEndConf ::= SEQUENCE
 {  result               EnvResult,
    unkRsc               RscId  OPTIONAL --reporting of unknown resources
 }

MEEConsCheckEndInd ::= SEQUENCE
 {  rscId                RscId
 }

MEEConsCheckEndReq ::= SEQUENCE
 {  rscId                RscId
 }

MEEConsCheckEndRes ::= SEQUENCE
 {  result               EnvResult,
    unkRsc               RscId  OPTIONAL --reporting of unknown resources
 }

MEEConsCheckInd ::= SEQUENCE
 {  rscId                RscId
 }

MEEConsCheckReq ::= SEQUENCE
 {  rscId                RscId
 }

MEEConsCheckRes ::= SEQUENCE
 {  result               EnvResult,
    unkRsc               RscId
 }

MEEErrorInd ::= SEQUENCE { }

MEELspFailureInd ::= SEQUENCE { }

MEEResetRscConf ::= SEQUENCE
 {  result               EnvResult,
    unkRscList           RscList  OPTIONAL --reporting of unknown resources
 }

MEEResetRscInd ::= SEQUENCE
 {  rscList              RscList
 }

MEEResetRscReq ::= SEQUENCE
 {  rscList              RscList
 }

MEEStartupConf ::= SEQUENCE
 {  result               EnvResult
 }

MEEStartupInd ::= SEQUENCE { }

MEEStartupReq ::= SEQUENCE { }

MEEStopTrafficConf ::= SEQUENCE { }

MEEStopTrafficReq ::= SEQUENCE { }

MEEUnblockRscConf ::= SEQUENCE
 {  result               EnvResult,
    unkRscList           RscList  OPTIONAL  --reporting of unknown resources
 }

MEEUnblockRscInd ::= SEQUENCE
 {  rscList              RscList
 }

MEEUnblockRscReq ::= SEQUENCE
 {  rscList              RscList
 }

MEEVerifyLspIdconf ::= SEQUENCE
 {  result               EnvResult
```

```
 }

MEEVerifyLspIdReq ::= SEQUENCE { }

MEEVPCCreateConf ::= SEQUENCE { }

MEEVPCCreateReq ::= SEQUENCE
 {  rscId               RscId
 }

MEEVPCDeleteConf ::= SEQUENCE { }

MEEVPCDeleteReq ::= SEQUENCE
 {  rscId               RscId
 }

 --supporting productions

EnvResult ::= BIT STRING (SIZE(4))
  envSuccess   EnvResult ::= '0000'B
  envUnkRsc    EnvResult ::= '0001'B
  envOpRejRem  EnvResult ::= '0010'B
  envOpFail    EnvResult ::= '0011'B
  envNotPerf   EnvResult ::= '0100'B
  envOpRejLoc  EnvResult ::= '1000'B
  envTransErr  EnvResult ::= '1001'B
  envMismatch  EnvResult ::= '1010'B
  envSaal      EnvResult ::= '1011'B

RscList       ::= SEQUENCE OF RscId   --environment resource list
BlRscList     ::= SEQUENCE OF BlRscId --environment block resource list
RTMCBlRscList ::= SEQUENCE OF BlRscIdContent

RscId ::= SEQUENCE
 {  rscIdContent         RscIdContent
 }

BlRscId ::= SEQUENCE
 {  envBlReason          EnvBlReason,
    rscIdContent         RscIdContent
 }

EnvBlReason ::= BIT STRING (SIZE(8))
  admPartEnv     EnvBlReason ::= '00000001'B
  admFullEnv     EnvBlReason ::= '00000010'B
  errorEnv       EnvBlReason ::= '00000100'B
  admPartErrEnv  EnvBlReason ::= '00001000'B
  admFullErrEnv  EnvBlReason ::= '00010000'B

END
```

Bibliography

The following material, though not specifically referenced in the body of the present document (or not publicly available), gives supporting information.

ETS 300 406 (1995): "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

History

Document history				
V1.1.1	June 1999	Public Enquiry	PE 9945:	1999-06-09 to 1999-11-05