ETSI
TECHNICAL
REPORT

ETR 193

October 1995

Source: ETSI TC-MTS

Reference: DTR/MTS-00002

ICS: 33.020

Key words: TTCN, NIT, TCP

Methods for Testing and Specification (MTS); Network Integration Testing (NIT); Methodology aspects; Test Co-ordination Procedure (TCP) style guide

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

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

New presentation - see History box

Whilst every care has been taken in the preparation and publication of this document, errors in content, typographical or otherwise, may occur. If you have comments concerning its accuracy, please write to "ETSI Editing and Standards Approval Dept." at the address shown on the title page.

Contents

Foreword 5

Introduction 5

1 Scope 7

2 References 7

3 Definitions 7

4 Abbreviations 7

5 General description 8
5.1 Reasons for Network Integration Testing (NIT) 8
5.2 The NIT concept 9
5.3 End-to-End testing 11
5.4 Node-to-Node testing 12

6 Methodological aspects 12
6.1 Current examples 12
6.2 Overview of ISO 9646 concepts and their applicability to NIT 12
6.2.1 Requirements 12
6.2.2 PICS and PIXIT 12
6.2.3 NIT profile 12
6.2.4 Type of test 12
6.2.5 Test method 13
6.2.6 Test notation 14

7 Test Co-ordination Procedure (TCP) style guide 14
7.1 TCP types 15
7.1.1 Communication TCPs 15
7.1.2 Synchronisation TCPs 15
7.2 Naming convention for TCPs 15
7.3 CM constraint definition 16

8 Example of concurrent TTCN for NIT 16
8.1 Test component declaration 16
8.2 Test component configuration declaration 16
8.2.1 Configuration 1 16
8.2.2 Configuration 2 17
8.2.3 Configuration 3 18
8.2.4 Configuration 4 19
8.3 Co-ordination point declaration 19
8.4 TTCN CM type definition 20
8.5 ASN.1 CM type definition 20
8.6 TTCN CM constraint declaration 21
8.7 ASN.1 CM constraint declaration 21
8.8 Test Case dynamic behaviour 22
8.9 Test Step dynamic behaviour 25

History 32

Blank page

Foreword

This ETSI Technical Report (ETR) has been produced by the Methods for Testing and Specification (MTS) Technical Committee of the European Telecommunications Standards Institute (ETSI).

ETRs are informative documents resulting from ETSI studies which are not appropriate for European Telecommunication Standard (ETS) or Interim European Telecommunication Standard (I-ETS) status. An ETR may be used to publish material which is either of an informative nature, relating to the use or the application of ETSs or I-ETSs, or which is immature and not yet suitable for formal adoption as an ETS or an I-ETS.

Introduction

The Network Integration Testing (NIT) studies start from several kinds of exigencies:

- theoretical, i.e. what is the relationship with the standard ISO 9646, methodology for conformance testing, in terms of test method, Abstract Test Suite (ATS), style and so on;

- practical, i.e. what protocols and tools are used in order to perform the test automatically;

- strategic, i.e. what is the strategy in testing the integration of new networks or new supplementary services.

The purpose of this ETR is not only to provide an answer to the first exigency, giving a method and some hints on how to use Tree and Tabular Combined Notation (TTCN) in this kind of testing, but also to highlight issues to be developed in the future.

Two others work items, covering, respectively, practical and strategic issues are the following:

DTR/MTS-00028: "Methods for Testing and Specification (MTS); Network Integration Testing (NIT); Architectural reference; Test Synchronization Protocol 1 (TSP1) specification".

DTR/MTS-00029: "Methods for Testing and Specification (MTS); Network Integration Testing (NIT); Interconnection; Reasons and goals for a global testing approach".

NOTE: Both of these work items are expected to be published at the end of 1995.

Blank page

1 Scope

This ETSI Technical Report (ETR) defines Network Integration Testing (NIT) and its methodological aspects, and gives some examples of Test Co-ordination Procedures (TCPs) to be applied between two or more testers.

NIT is applicable, in general, to any network configuration. However, the NIT approach is very suitable for testing of the international networks to facilitate automatic execution of a Test Suite between two or more Public Network Operators (PNO). Examples in the following clauses will refer mainly to this situation.

The objective of this ETR is to produce simple results in accordance with ISO 9646 [1] to [7] (in particular taking into account the Multi-Party Testing Method (MPTM)) in order to supply a framework for future project planning.

2 References

For the purposes of this ETR, the following references apply:

[1] ISO/IEC 9646-1 (1992): "Information technology - OSI conformance testing methodology and framework - Part 1: General concepts".

[2] ISO/IEC 9646-1/DAM.1 (1993): "Information technology - OSI conformance testing methodology and framework - Multi-party testing".

[3] ISO/IEC 9646-2 (1992): "Information technology - OSI conformance testing methodology and framework - Part 2: Abstract test suite specification".

[4] ISO/IEC 9646-3 (1992): "Information technology - OSI conformance testing methodology and framework - Part 3: Tree and tabular combined notation".

[5] ISO/IEC 9646-3 AM. 1 (1993): "Information technology - OSI conformance testing methodology and framework - Concurrent TTCN".

[6] ISO/IEC 9646-4 (19924): "Information technology - OSI conformance testing methodology and framework - Part 4: Test realisation".

[7] ISO/IEC 9646-5 (1992): "Information technology - OSI conformance testing methodology and framework - Part 5: Requirements on test laboratories and clients for the conformance assessment process".

[8] ETR 141 (October 1994): "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specification; The Tree and Tabular Combined Notation (TTCN) style guide".

3 Definitions

For the purpose of this ETR, all the definitions in ISO/IEC 9646 and its amendments apply [1] to [7].

4 Abbreviations

For the purpose of this ETR, all the symbols and abbreviations defined in ISO/IEC 9646 and its amendments [1] to [7] to apply.

ATS	Abstract Test Suite
B - ISDN	Broadband - ISDN
CM	Co-ordination Message
CP	Co-ordination Point
ETS	Executable Test Suite
EURESCOM	European institute for Research and Strategic studies in telecommunication
IN	Intelligent Network
ISDN	Integrated Services Digital Network

ISO	International Standard Organization
ISUP	ISDN User Part
IUT	Implementation Under Test
LT	Lower Tester
LTCF	Lower Tester Control Function
MSC	Message Sequence Chart
MTC	Master Test Component
MPTM	Multi-Party Testing Method
NE	Network Element
NIT	Network Integration Testing
N-ISDN	Narrow Band - ISDN
PCO	Point of Control and Observation
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation eXtra Information for Testing
PLMN	Public Land Mobile Network
PNO	Public Network Operator
PSTN	Public Switched Telephone Network
PT	Protocol Tester
PTC	Parallel Test Component
SUT	System Under Test
TCP	Test Co-ordination Procedure
TTCN	Tree and Tabular Combined Notation
TMN	Telecommunication Management Network
UT	Upper Tester

5 General description

5.1 Reasons for Network Integration Testing (NIT)

In order to provide customers with modern telecommunication services in a homogeneous and reliable way, the network is becoming more and more a complex, "global network", based on the interaction of many highly co-operating network nodes. Distributed call processing also takes place in some cases.

Different kinds of users, services (bearer and/or supplementary) and transport technologies might interoperate in the establishment of future telecommunication services. For example, Narrow Band Integrated Services Digital Network (N-ISDN), Broadband - ISDN (B-ISDN), and mobile users, might negotiate the quality of services, call each other using different bearer connectivities, and/or require access to specialised service nodes, e.g. to provide "Intelligent Network" services. Management of Network Elements (NE) of services using "external" management networks (e.g. Telecommunication Management Network (TMN)) could also be in the scenario. In this complex scenario, the services should however be offered in a reliable and homogeneous way and no undesired interactions or side effects between all such services and/or service features should, in principle, exist.

Actually, potential problems could arise, due to the complexity of the new services and of the network elements or because the occurrence of major technical changes in sub-parts of the "global network" will occur (for example, when a new set of basic or supplementary services is introduced, or when a new version of a network signalling protocol (e.g. a new version of ISDN User Part (ISUP)) starts to be operated in a sub-part of the global network).

To prevent or limit the occurrence of such problems, the network behaviour should be testable and monitorable using "not-only-domestic" approaches and techniques. This is necessary to check, for example, that the bearer and supplementary ISDN services, as implemented in the sub-parts of the global networks, are actually capable to inter-work in the global network, are compatible, and are provided to customers in an homogeneous way. In case of problems, efficient and reliable technical methods and managerial procedures and processes to investigate the reasons of the possible failures are also appreciable.

There is also a need to avoid multiplication of efforts and resources in the achievement of the above results, simplify the achievement of agreements among the different organisations on common methods for testing the global network behaviour(s). So, general guidelines and preliminary technical references to address this process should be found opportune and useful.

The expected general result is a distributed testing methodology and process (NIT) that could be applied to several network configurations, for example; Integrated Services Digital Network (ISDN), Public Land Mobile Network (PLMN), Public Switched Telephone Network (PSTN), Intelligent Network (IN) and so on.

5.2 The NIT concept

From the technical point of view, NIT is the set of all the checks necessary in order to verify that a given network works as expected, and to verify the compatibility of the single network components (Network Elements). Conformance testing of each network component is assumed as a pre-requisite.

The network complexity may vary, including interworking of protocols and/or national networks. In any case, this technique is concerned only with the external behaviour of the network in the case of "End-to-End" testing (see figure 1), and it is concerned also with the internal behaviour in the case of "Node-to-Node". End-to-End means testing the network as seen at user accesses and Node-to-Node means testing the network as seen from network trunks. The principal distinction within NIT should be flexibility and a high level of confidence in the network behaviour, because the network itself can be tested from the point of view of the network protocols, as well as from the point of view of the access protocol(s).

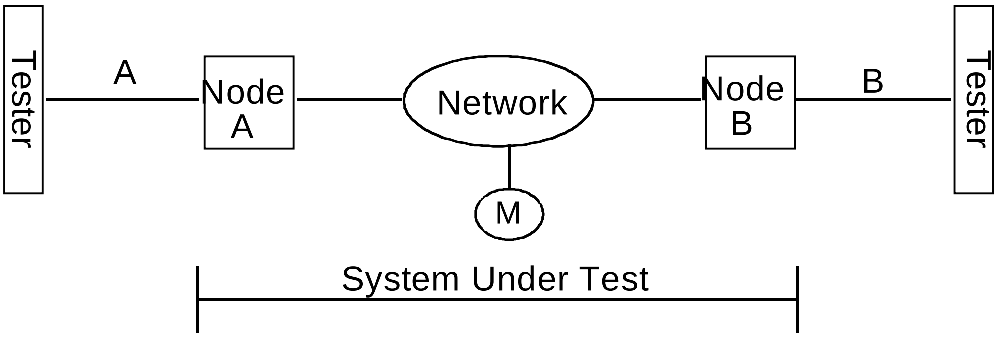


Figure 1: The System Under Test (SUT) for NIT

The System Under Test (SUT) is composed of all the network components that are placed between the interfaces where the testers are to be connected.

NIT should be conceived in order to allow it to be executed in two different situations:

- in a controlled situation, i.e. in a local or distributed test plant, before the new functions and services are deployed into the real network elements;

- in a real situation, when the new functions and services have already been deployed in the real network elements, which are therefore in operational states.

NOTE: The second configuration implies that the Test Suite and the related procedures should be designed so as not to disturb the normal network behaviour.

From the point of view of the Implementation Under Test (IUT) a main distinction within NIT is between End-to-End testing and Node-to-Node testing, as previously mentioned:

End-to-End testing: the network is tested as it is seen from the user's terminal equipments (e.g. A = B = ISDN Basic Access protocol as shown in figure 1), i.e. taking the user-network interfaces as Points of Control and Observation (PCO);

Node-to-Node testing: the network is tested as it is seen from other network components (e.g. A = B = ISUP protocol as shown in figure 1), i.e. taking as PCOs the external network-network interfaces. M is a generic monitor point that is used to check the internal behaviour of the network.

The IUT is composed of all the parts in the SUT that contribute to perform the expected functionality, i.e. connection and transport of data between the external gates:

- protocols that manage the external entities connected to the SUT (access protocols in the case of End-to-End testing, network protocols (e.g. ISUP in the case of SS#7) in the case of Node-to-Node);

- in both cases each network component involved in a call between A-side and B-side is part of the IUT.

5.3 End-to-End testing

Figure 2 shows an example of an End-to-End test configuration for testing the ISDN international network.

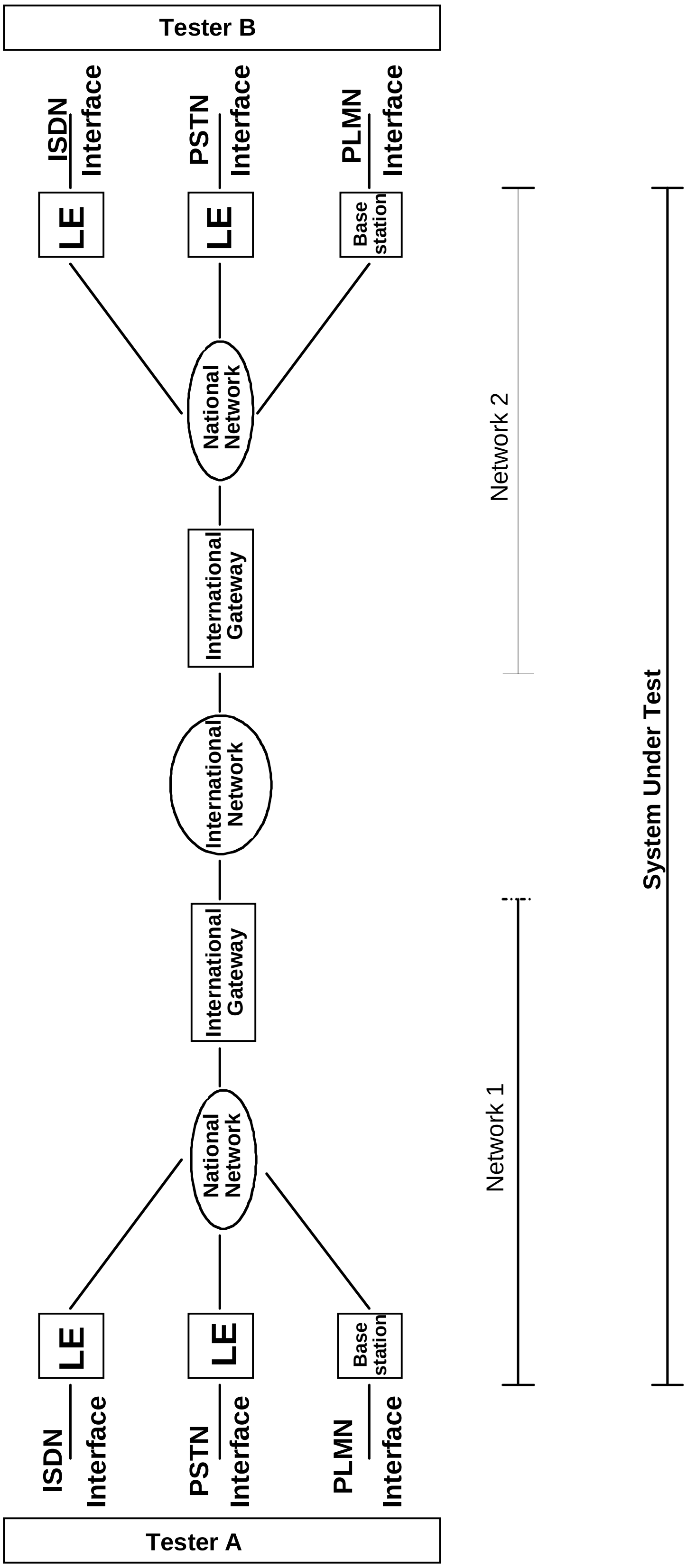
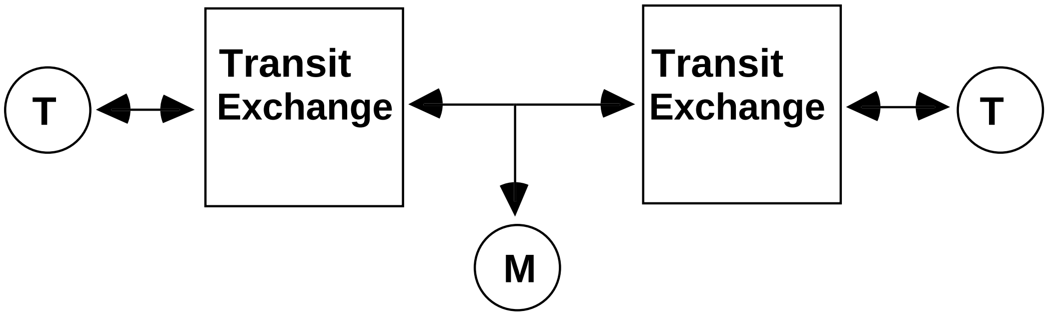


Figure 2: Example of End-to-End test configuration for testing the ISDN international network

5.4 Node-to-Node testing

Figure 3 shows an example of a Node-to-Node test configuration where two testers and a monitor are used. This allows compatibility testing to be performed, e.g. according to ITU-T Recommendations Q.784 and Q.785.



T = Tester

M = Monitor

Figure 3: Example of Node-to-Node test configuration

6 Methodological aspects

6.1 Current examples

A first example of Abstract Test Suites (ATSs) concerning ISDN End-to-End checks was performed by the European Institute for Research and Strategic studies in telecommunications (EURESCOM) (Project P.104). But this ATS does not follow MPTM. Another ATS for End-to-End testing has been performed by EURESCOM Project P.412, of which the actual release is Version 2. This version, also, does not follow MPTM. A first example of Node-to-Node ATS has been performed by EURESCOM Project P.412. This ATS follows MPTM and it is written in concurrent Tree and Tabular Combined Notation (TTCN).

The TTCN tables used in the following clauses are based on the examples above.

6.2 Overview of ISO 9646 concepts and their applicability to NIT

6.2.1 Requirements

NIT Requirements have to be based not on a single set of reference specifications but on many, at least one for each interconnected sub-network, from which the compatibility requirements must be extracted.

6.2.2 PICS and PIXIT

A Protocol Implementation Conformance Statement (PICS) for NIT can be derived from the PICS of each reference specification.

A Protocol Implementation eXtra Information for Testing (PIXIT) can be derived from the PIXIT of each implementation.

6.2.3 NIT profile

A NIT profile can be derived from the PICS of different base standards or profiles. This allows the inclusion of a requirement list for the specific interworking under test.

6.2.4 Type of test

As the purpose of NIT is not to check the conformance of the network to the standard, but its functionality, that is to check how the information related to the establishment, usage and release of a call is carried between the network components, **basic interconnection** category is definitely in its scope.

The type of test for NIT is an open issue. In fact there could be other types of tests in NIT scope.

6.2.5 Test method

The chosen method for NIT is MPTM. In the case of End-to-End testing, an applicable method is MPTM without Upper Tester (UT) (see figure 4). In the case of the Node-to-Node testing, it is necessary to cause within the IUT some conditions necessary to continue the test. In this case an applicable method is MPTM with an UT (see figure 5).

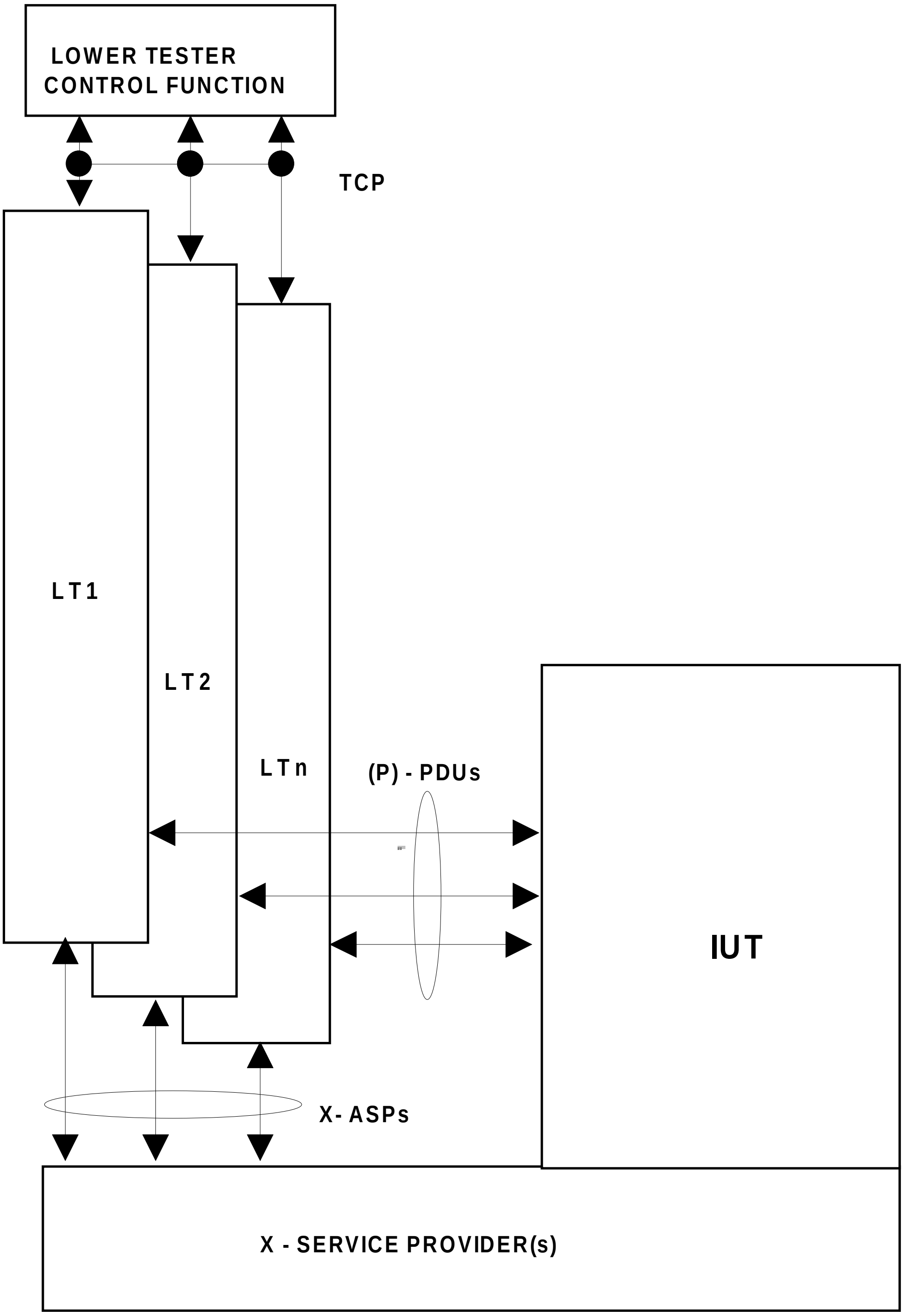


Figure 4: MPTM used for End-to-End testing

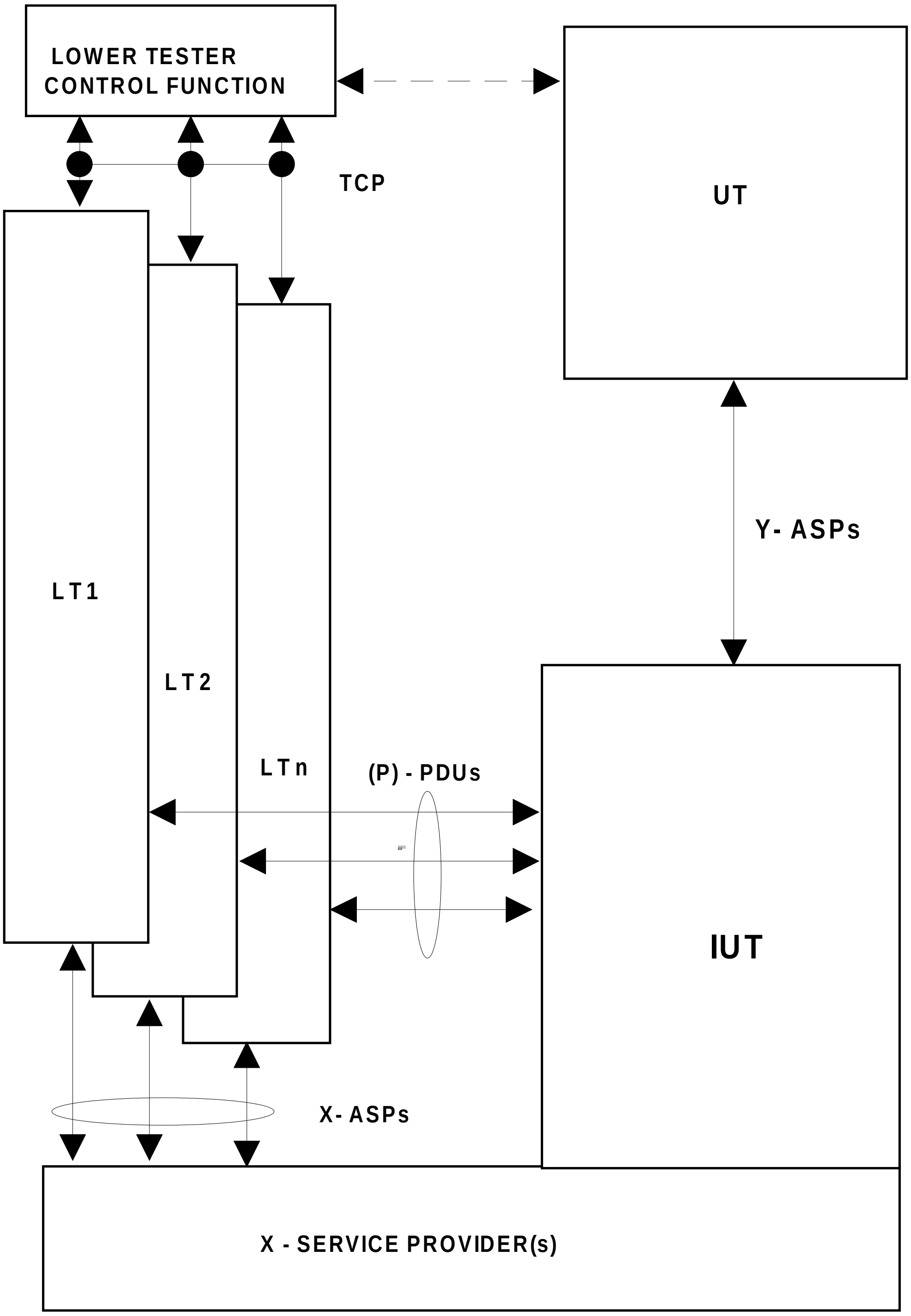


Figure 5: MPTM used for Node-to-Node testing

6.2.6 Test notation

The ATS designer should use a standardized notation defined in ISO/IEC 9646-3 [4] (TTCN). In particular MPTM is used with "Concurrent TTCN" as specified in ISO/IEC 9646-3/AM1 [5].

7 Test Co-ordination Procedure (TCP) style guide

This clause is a set of guidelines for the designer of a NIT ATS. Different kinds of TCPs are described. Examples of usage of TCPs within concurrent TTCN are shown. Concerning the TTCN problems see ETR 141 [8].

TCPs are introduced within the concurrent TTCN framework.

TCPs are written in the ATS and they allow the writer to achieve a complete description of test case realisation. The purpose of this ETR is to show the different kind of TCPs, with their description, that can be used to cover various requirements of tester co-ordination.

A good description of TCPs should produce:

- a better comprehension of the ATS concerning tester co-ordination;
- an Executable Test Suite (ETS) with co-ordination information between testers that is implementation independent.

TCPs are described with Co-ordination Messages (CMs) exchanged through Co-ordination Points (CPs) that are defined between two Test Components.

7.1 TCP types

Using concurrent TTCN in describing NIT ATSs, two different TCP types can be used:

- communication TCPs;
- synchronisation TCPs.

The difference between the two types depends on the presence or absence of a parameter within the CM. In this way it is possible to divide, semantically, the synchronization from the communication problem.

7.1.1 Communication TCPs

In this clause all the TCPs that are oriented to the exchange of a parameter between two Test Components are described. The CM contains the message identifier and the parameters.

An example of this CM is:

INFO(par1, par2,...,parN) This TCP can be used to specify within ATS the transfer of parameters between two Parallel Test Components (PTC) or between Master Test Component (MTC).

7.1.2 Synchronisation TCPs

In this clause all the TCPs that are oriented to the synchronisation of two or more Test Components are described. In this case the only information is the identifier of the CM.

Some examples of these CMs are:

STOP	This can be used by a MTC or a PTC to stop another PTC.
TOKEN	This can be used to transfer the test execution between two PTCs.

7.2 Naming convention for TCPs

Having a naming convention for a TCP means giving a common name for the CMs and for the CPs.

This can be obtained using only capital letters, separating name with "_" (underscore) when it is needed. Furthermore it is advisable to give names semantically in the scope (e.g. INFO indicates a CM that carries an information).

Concerning the conventions for the CP, it is recommended to define the name indicating:

"CP_" + "Test Component name" + "Test Component name" (e.g. CP_AB where A and B are the PTCs involved in the test.),

or,

"CP_" + "Test Component name" + "_" + "Test Component name" (e.g. CP_A1_B1 where A1 and B1 are the PTCs involved in the test.).

7.3 CM constraint definition

For the constraint definition see ETR 141 [8].

8 Example of concurrent TTCN for NIT

In this clause some simple recommendations to follow when writing a NIT ATS are described. They are not a set of rules, but only some advice coming from the recent experiences in the NIT.

8.1 Test component declaration

No particular advice for NIT ATS.

Table 1: Test component declaration

Test component declaration				
Component Name	**Component Role**	**Nr PCOs**	**Nr CPs**	**Comments**
M	MTC	0	0	Master Test Component
A	PTC	0	1	Parallel Test Component
B	PTC	0	1	Parallel Test Component
C	PTC	0	1	Parallel Test Component
D	PTC	0	1	Parallel Test Component
A1	PTC	0	1	Parallel Test Component
A2	PTC	0	1	Parallel Test Component
Detailed comments:				

8.2 Test component configuration declaration

No particular configuration seems to be critical for NIT. Anyway there are some typical configurations that are used. The following configuration has no CPs between the MTC and the PTCs and no PCOs on the MTC.

8.2.1 Configuration 1

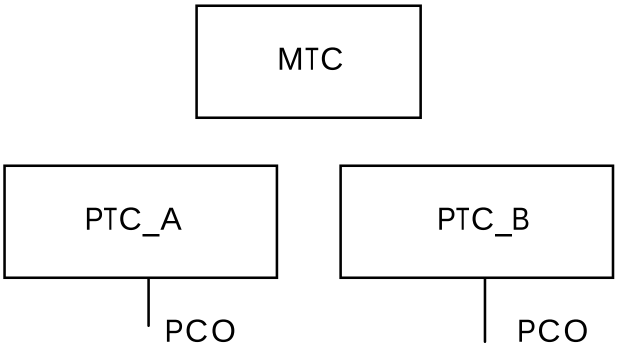


Figure 6: Config1

This configuration is used for an End-to-End test that checks an ISDN Basic Call. For this kind of test there is no need for co-ordination through CPs.

Table 2: Config1

Test component configuration declaration

Config Name : Config1

Comments : Configuration used for test without TCP

Components Used	PCOs Used	CPs Used	Comments
M	-	-	Master Test Component
A	LA	-	Parallel Test Component 1)
B	LB	-	Parallel Test Component 2)

Detailed comments:

1) Calling side;

2) Called side.

8.2.2 Configuration 2

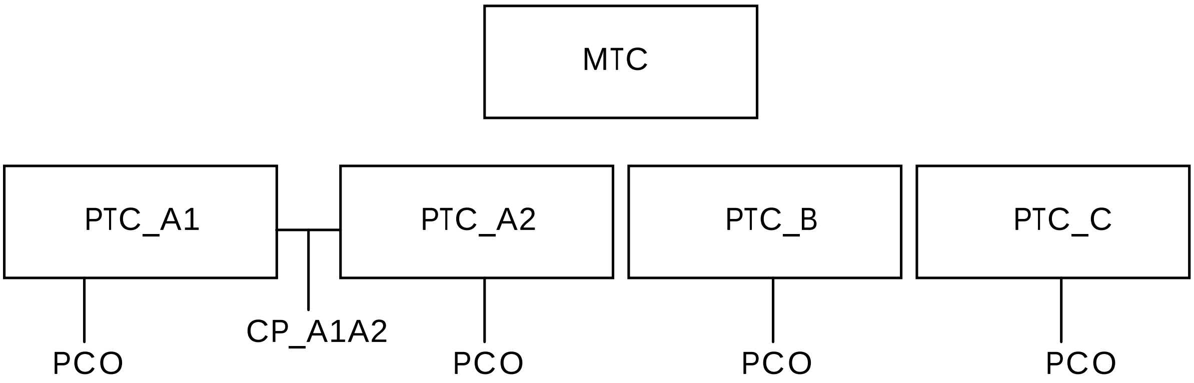


Figure 7: Config2

This configuration is used for an End-to-End test that checks the ISDN CONF supplementary service. The CP is used to pass the test control from PTC_A1 (which calls the B-side) to PTC_A2 (which calls the C-side). After that PTC_A2 returns the control to PTC_A1 which establishes the conference and gives the control to PTC_A2 which completes the test on the CONF supplementary service.

This is an example of TCPs co-ordination type. The choice of having two components on side A in order to manage independently the two calls with B and C is not strictly necessary, but allows for a more flexible behaviour and a closer relationship with the test implementation.

Table 3: Config2

Test component configuration declaration

Config Name : Config2

Comments : Configuration used for test with a simple co-ordination between two PTC

Components Used	PCOs Used	CPs Used	Comments
M	-	-	Master Test Component
A1	LA	-	Parallel Test Component 1)
A2	LA	-	Parallel Test Component 2)
B	LB	-	Parallel Test Component 3)
C	LC	-	Parallel Test Component 4)

Detailed comments:

1) A1 makes the first call for the establishment of the conference;

2) A2 makes the second call for the establishment of the conference;

3) First called side;

4) Second called side.

8.2.3 Configuration 3

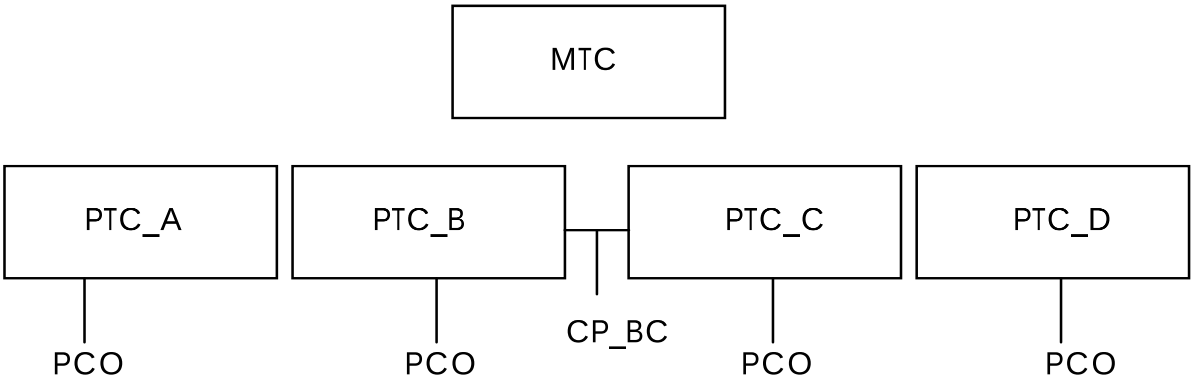


Figure 8: Config3

This configuration is used for an End-to-End test that performs two ISDN Basic Calls, that are made on the same Call Reference (CR), in the two directions. The purpose is to check that the same CR can be used in the same access by both the user and the network without provoking a rejection of the second call.

In this example PTC_A calls PTC_B, which uses CP_BC to give the information of the CR value to PTC_C, which then uses that value to make a call to PTC_D. PTC_B and PTC_C simulate two equipment connected to the same user interface.

Table 4: Config3

Test component configuration declaration			
Config Name	**:**Config3		
Comments	**:**Configuration used for the test with the TCP that carry a parameter		
Components Used	**PCOs Used**	**CPs Used**	**Comments**
M	-	-	Master Test Component
A	LA	-	Parallel Test Component 1)
B	LB	CP_BC	Parallel Test Component 2)
C	LC	CP_BC	Parallel Test Component 3)
D	LD	-	Parallel Test Component 4)
Detailed comments: 1) Calling PTC in the forward direction; 2) Called PTC in the forward direction; 3) Calling PTC in the backward direction; 4) Called PTC in the backward direction.			

8.2.4 Configuration 4

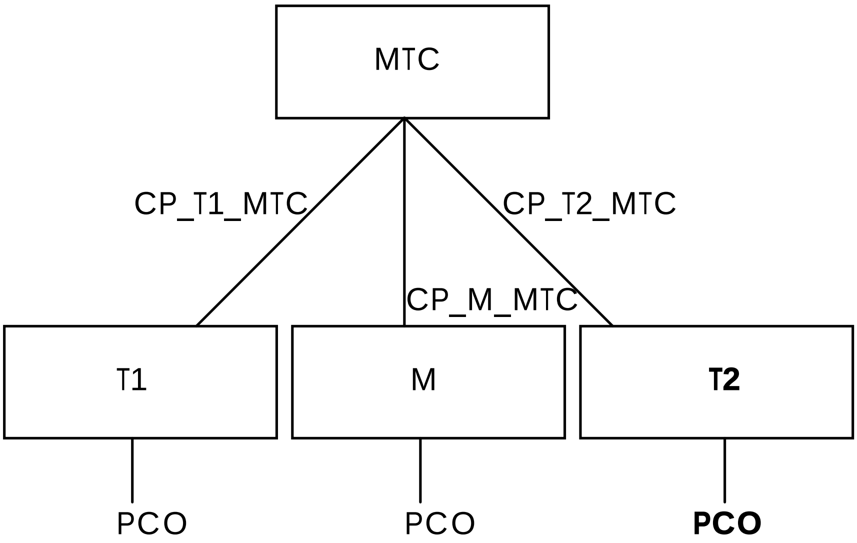


Figure 9: Config4

This configuration is used for a Node-to-Node test that has the purpose to check blocking, reset and unblocking of the circuit between the switching systems for the ISUP protocol. In this test CMs are strongly used. In fact, the test is divided into five steps, and each step starts with the MTC, which sends a CM to PTCs involved in the test step. This CM belongs to "synchronisation type" because it carries a token, which indicates whether the PTC has to be activated or not. A Message Sequence Chart (MSC) which explains the message exchange is given in figure 10.

Table 5: Config4

Test component configuration declaration			
Config Name	**:**Config4		
Comments	**:**Configuration used for the test with the synchronisation TCP (strongly used).		
Components Used	**PCOs Used**	**CPs Used**	**Comments**
MTC	-	CP_T1_MTC, CP_T2_MTC, CP_M_MTC	Master Test Component
T1	LT1A	CP_T1_MTC	Parallel Test Component 1)
T2	LT2B	CP_T2_MTC	Parallel Test Component 2)
M	LAB, LBA	CP_M_MTC	Parallel Test Component 3)
Detailed comments: 1) Tester 1; 2) Tester 2; 3) Monitor.			

8.3 Co-ordination point declaration

The only advice in this case is that the name of the co-ordination point contains the names of the two test components.

Table 6

CP Declaration	
CP Name	**Comments**
CP_BC	CP used in Config 3
CP_A1A2	CP used in Config 2
CP_T1_MTC	CP used in Config 4
CP_T2_MTC	CP used in Config 4
CP_M_MTC	CP used in Config 4
Detailed comments:	

8.4 TTCN CM type definition

In the following two tables (table 7 and 8) the two types of TCPs are represented. The synchronisation type is represented by CM type TOKEN and the communication type is represented by CM type INFO.

Table 7: TTCN CM type definitions

TTCN CM type definition		
CM Name	: TOKEN	
Comments	: Co-ordination Message used only for co-ordination	
Parameter Name	**Parameter Type**	**Comments**
CM_identification	INTEGER	Co-ordination Message identification
Detailed comments:		

Table 8: TTCN CM type definitions

TTCN CM type definition		
CM Name	: INFO	
Comments	: Co-ordination Message used to communicate a parameter	
Parameter Name	**Parameter Type**	**Comments**
CM_identification	INTEGER	Co-ordination Message identification
PAR1	INTEGER	Parameter Value
PAR2	INTEGER	Parameter Value
PAR3	INTEGER	Parameter Value
Detailed comments:		

8.5 ASN.1 CM type definition

In the following two tables (table 9 and 10) the two TCP types are represented in ASN.1 description.

Table 9: ASN.1 CM type definition

ASN.1 CM type definition	
Type Name	**:** TOKEN
Comments	**:** CM type definition for TCP Synchronisation type
Type Definition	

SEQUENCE {
cm_identification INTEGER
}	

Detailed comments:	

Table 10: ASN.1 CM type definition

ASN.1 CM type definition	
Type Name	**:** INFO
Comments	**:** CM type definition for TCP Communication type
Type Definition	

SEQUENCE {
cm_identification INTEGER,
par1 INTEGER,
par2 INTEGER,
par3 INTEGER
}	

Detailed comments:	

8.6 TTCN CM constraint declaration

A generic CM Constraint can be described by a CM_identifier for the CM TOKEN type. For the CM INFO type the parameters exchanged between test components are to be included as well.

Table 11: TTCN CM constraint declaration

TTCN CM constraint declaration		
Constraint Name	: TOK1	
CM Type	: TOKEN	
Derivation Path	:	
Comments	: This Co-ordination Message allows synchronization of two Parallel Test Component	
Field Name	**Field Value**	**Comments**
CM_identification	1	first Co-ordination Message
Detailed comments:		

Table 12: TTCN CM constraint declaration

TTCN CM constraint declaration		
Constraint Name	: INFO1(CR_VAL: INTEGER)	
CM Type	: INFO	
Derivation Path	:	
Comments	: This Co-ordination Message allows synchronization of two Parallel Test Component carrying a parameter	
Field Name	**Field Value**	**Comments**
CM_identification	2	second Co-ordination Message
PAR1	CR_VAL	value of the parameter
PAR2	-	parameter not used
PAR3	-	parameter not used
Detailed comments:		

8.7 ASN.1 CM constraint declaration

An example of ASN.1 constraint declaration is shown below.

Table 13: ASN.1 CM constraint declaration

ASN.1 CM constraint declaration	
Constraint Name	: TOK1
CM Type	: TOKEN
Derivation Path	**:**
Comments	: This Co-ordination Message allows synchronize of two Parallel Test Component
Constraint Value	

SEQUENCE {
cm_identification 1
}	

Detailed comments:	

Table 14: ASN.1 CM constraint declaration

ASN.1 CM constraint declaration	
Constraint Name	: INFO1(CR_VAL: INTEGER)
CM Type	: INFO
Derivation Path	**:**
Comments	: This Co-ordination Message allows synchronize of two Parallel Test Component carrying a parameter
Constraint Value	

SEQUENCE {
cm_identification 1,
par1 CR_VAL ,
par2 | ABSENT,
par3 | ABSENT
}	

Detailed comments:	

8.8 Test Case dynamic behaviour

In the following test cases the only peculiarity is the order of the CREATE statement. They should be used starting from the last PTC that has to be run depending on practical consideration. For example, it may be necessary to run the PTCs starting from the last called side, finishing with the first calling side. This allows that when the first calling PTC is run, the other PTCs are already waiting for a message.

The result is passed implicitly, and the CMs are exchanged through CPs.

Table 15: Test Case dynamic behaviour

Test Case dynamic behaviour					
Test Case Name	: 110101				
Group	:				
Purpose	: En-block sending: Ensure that call establishment using en-block sending is performed correctly.				
Default	: CLEAR_DOWN_FAIL				
Configuration	**:** config_1				
Comments	:				
Nr.	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		CREATE (B, B_110101)			
2		CREATE (A, A_110101)			
3		START T1			
4		?DONE (A,B)		R	
5		?TIMEOUT T1		F	
Detailed comments:					

The test case above refers to an ISDN Basic Call. As shown, the PTC called side is run before the PTC calling side.

Table 16: Test Case dynamic behaviour

Test Case dynamic behaviour					
Test Case Name	: 210901				
Group	:				
Purpose	: Establish/Isolate/Reattach/Drop/Disconnect a conference: Ensure that the remote parties are notified of the conference call progress				
Default	: CLEAR_DOWN_FAIL				
Configuration	**:** config_2				
Comments	:				
Nr.	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		CREATE (C, C_210901)			
2		CREATE (B, B_210901)			
3		CREATE (A2, A2_210901)			
4		CREATE (A1, A1_210901)			
5		START T1			
6		?DONE (A1, A2, B, C)		R	
7		?TIMEOUT T1		F	
Detailed comments:					

The test case above refers to ISDN CONF supplementary service.

Table 17: Test Case dynamic behaviour

Test Case dynamic behaviour					
Test Case Name	: 110102				
Group	:				
Purpose	: En-block sending: Ensure that the same Call Reference can be used to make a call in the opposite direction.				
Default	: CLEAR_DOWN_FAIL				
Configuration	**:** config_3				
Comments	:				
Nr.	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		CREATE (D, D_110102)			
2		CREATE (C, C_110102)			
3		CREATE (B, B_110102)			
4		CREATE (A, A_110102)			
5		START T1			
6		?DONE (A, B, C, D)		R	
7		TIMEOUT T1		F	
Detailed comments:					

The test case above refers to the usage of the same CR, in two opposite ISDN calls.

In the following figure, the message exchange, which happens during a Node-to-Node test, is shown. This test deals with blocking, unblocking, and reset of a circuit for the ISUP protocol. This test is divided into five different steps. Each step is started in the PTC, receiving from the MTC a CM (synchronisation type). To minimize the flow of the messages, this CM is sent only when a real action is expected from the PTC. This is determined on the basis of a boolean parameter.

The same result could be achieved by always sending a communication type CM containing the boolean parameter.

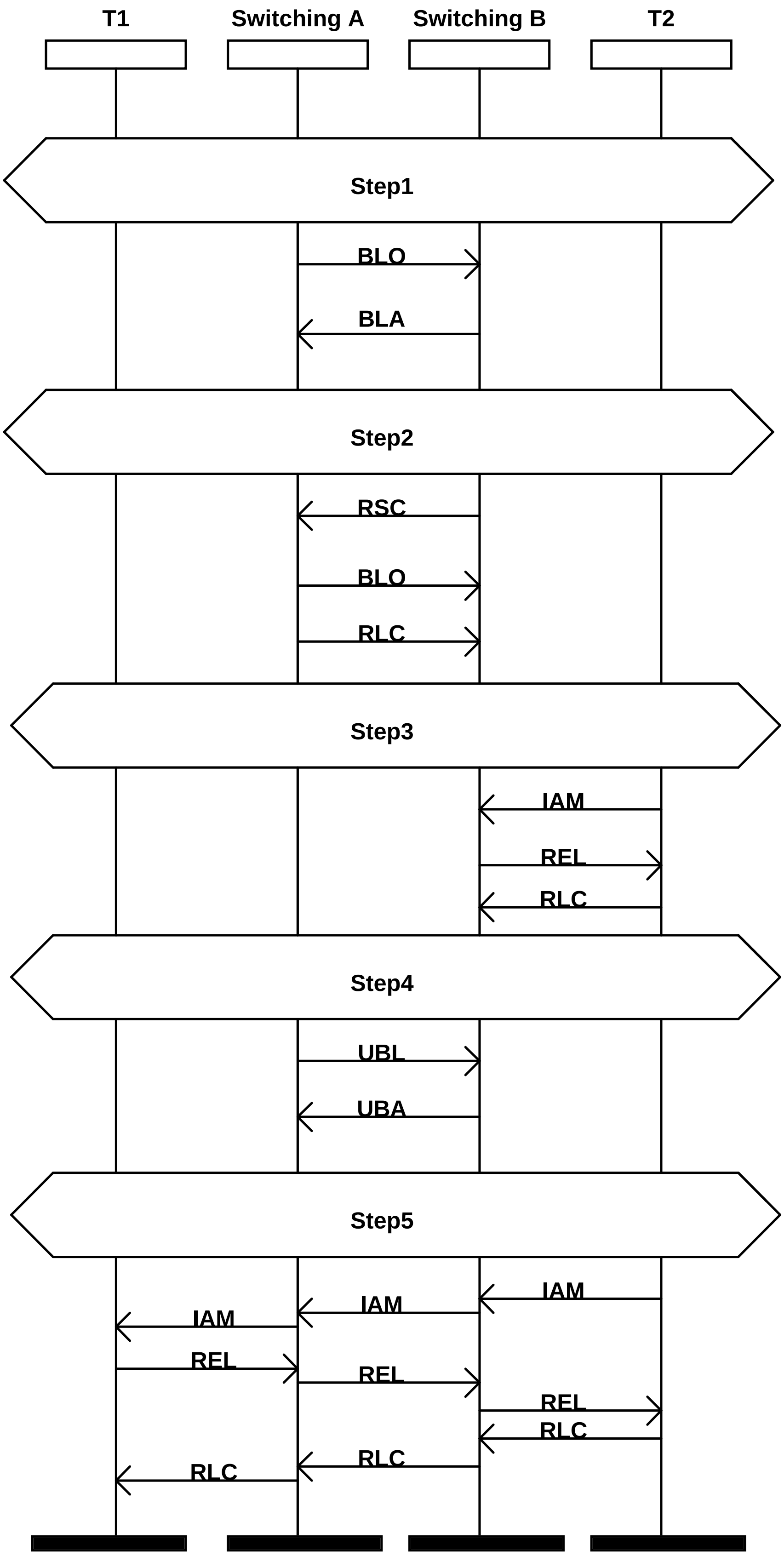


Figure 10: MSC of test case NTNB10203

Table 18: Test Case dynamic behaviour

Test Case dynamic behaviour

Test Case Name	:	NTNB10203
Group	:	
Purpose	:	Check that transit exchange B is able to generate RSC messages for remotely blocked circuits, and that transit exchange A is able to react to these messages
Default	:	
Configuration	**:**	config_4
Comments	:	

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		CREATE (M, M_TREE)			
2		CREATE (T2, T2_TREE)			
3		CREATE (T1, T1_TREE)			
4		+StartOneStep(FALSE, TRUE, FALSE)			Start Step 1
5		<IUT!BLO_REQ>	BLO_AB		
6		+FinishOneStep(FALSE, TRUE, FALSE)			Finish Step 1
7		+StartOneStep(FALSE, TRUE, FALSE)			Start Step 2
8		<IUT!RSC_REQ>	RSC_BA		
9		+FinishOneStep(FALSE, TRUE, FALSE)			Finish Step 2
10		+OneStep(FALSE, FALSE, TRUE)			Step 3
11		+StartOneStep(FALSE, TRUE, FALSE)			Start Step 4
12		<IUT!UBL_REQ>	UBL_AB		
13		+FinishOneStep(FALSE, TRUE, FALSE)			Finish Step 4
14		+OneStep(TRUE, TRUE, TRUE)			Step 5
15		START TWAIT1			
16		?DONE (M, T1, T2)		R	
17		?TIMEOUT TWAIT1		F	
		T1_TREE			
18		+EmptyStep			Step1
19		+EmptyStep			Step2
20		+EmptyStep			Step3
21		+EmptyStep			Step4
22		+AT1_CLUC_1			Step5
		T2_TREE			
23		+EmptyStep			Step1
24		+EmptyStep			Step2
25		+T2B_CLBC_1			Step3
26		+EmptyStep			Step4
27		+T2B_CLUC_1			Step5
		M_TREE			
28		+AB_BLOCK_1 (BLO_AB, BLA_BA)			Step1
29		+BA_RESET_1 (RSC_BA, RLC_AB. BLO_AB)			Step2
30		+EmptyStep			Step3
31		+AB_UNBLOCK_1 (UBL_AB, UBA_BA)			Step4
32		+BA_CLUC_1			Step5

Detailed comments:

In the test case above the exchange of CMs is made in StartOneStep, which contains three boolean values, which correspond to the presence, or not, of message exchange in the MSC for the current step.

8.9 Test Step dynamic behaviour

The peculiarity in these test step is sending and receiving of CMs.

In the example the test steps are divided into three groups:

1) No TCP, with no exchange of CMs;
2) Synchronisation, with exchange of CMs belonging to TOKEN type;
3) Communication, with exchange of CMs belonging to INFO type.

Table 19: Test Step dynamic behaviour (no TCP)

Test Step dynamic behaviour		
Test Step Name	:	A_110101
Group	:	NoTCP/
Objective	:	En-block sending: Ensure that call establishment using en-block sending is performed correctly.
Default	:	CLEAR_DOWN_FAIL
Comments	:	Origination side description

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='0'B)			
2		L!SETUP START T303	SU_SPE_S		
3		+OUTGOING_CALL_ALERT			
4		+CHECK_BCHANNEL		(P)	
5		L?DISCONNECT START T305	DI_S		
6		+CLEAR_DOWN			

Detailed comments:

Table 20: Test Step dynamic behaviour (no TCP)

Test Step dynamic behaviour		
Test Step Name	:	B_110101
Group	:	NoTCP/
Objective	:	En-block sending: Ensure that call establishment using en-block sending is performed correctly.
Default	:	CLEAR_DOWN_FAIL
Comments	:	Destination side description

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='1'B) START TWAIT			
2		L?SETUP (CR=SETUP.CREF.CRV) CANCEL TWAIT	SU_SPE_R		Get call reference
3		L!ALERTING	AL_S		
4		L!CONNECT	CN_S		
5		+CHECK_BCHANNEL		(P)	
6		+CLEAR_DOWN			
7		?TIMEOUT TWAIT		F	No call delivered

Detailed comments:

Table 21: Test Step dynamic behaviour (Synchronization)

Test Step dynamic behaviour		
Test Step Name	:	A1_210901
Group	:	Synchronization/
Objective	:	Establish/Isolate/Reattach/Drop/Disconnect a conference: Ensure that the remote parties are notified of the conference call progress
Default	:	CLEAR_DOWN_FAIL
Comments	:	Origination side description, first call

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='0'B)			
2		L!SETUP START T303	SU_SPE_S		Call B
3		+OUTGOING_CALL_ALERT			
4		L!HOLD START THOLD	HD_S		
5		L?HOLD_ACK CANCEL THOLD	HDA_R		
6		CP_A1A2!TOKEN	TOK1		Send token to A2
7		CP_A1A2?TOKEN	TOK1		Receive token from A2
8		+ADD_CONF			
		ADD_CONF			
9		L!FACILITY	FA_ADD_CONF_INV_S		
10		L?FACILITY (PARTY_ID:=FACILITY.FAC.PARAMETER)	FA_ADD_CONF_RR_R		
11		L?DISCONNECT	DI_R		
12		L!RELEASE	RE_S		
13		L?RELEASE_COMP	RE_S		
14		CP_A1A2!TOKEN	TOK1		Send token to A2

Detailed comments: The calling party must subscribe to the CONF service.

Table 22: Test Step dynamic behaviour (Synchronization)

Test Step dynamic behaviour

Test Step Name : A2_210901
Group : Synchronization/
Objective : Establish/Isolate/Reattach/Drop/Disconnect a conference:
Ensure that the remote parties are notified of the conference call progress
Default : CLEAR_DOWN_FAIL
Comments : Origination side description, second call

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		CP_A1A2?TOKEN	TOK1		Receive token from A1
2		(FL:='0'B)			
3		L!SETUP START T303	SU_SPE_SEC_NR_S		Call C
4		+OUTGOING_CALL_ALERT			
5		L!FACILITY	FA_CONF_INV_S		
6		L!FACILITY	FA_CONF_RR_R		Conference established
7		CP_A1A2!TOKEN	TOK1		Send token to A1
8		CP_A1A2?TOKEN	TOK1		Receive token from A1
9		L!RETRIEVE START TRETR	RT_S		
10		L?RETRIEVE_ACK CANCEL TRETR			
11		+CONF CONF			
12		+CHECK_BCHANNEL		(P)	
13		L!FACILITY	FA_ISO_CONF_INV_S		Isolate B
14		L?FACILITY	FA_ISO_CONF_RR_R		
15		L!FACILITY	FA_REATT_CONF_INV_S		Reattach B
16		L?FACILITY	FA_REATT_CONF_RR_R		
17		L!FACILITY	FA_DROP_CONF_INV_S		Disconnect B
18		L?FACILITY	FA_DROP_CONF_RR_R		The call to B is cleared
19		L!DISCONNECT START T305	DI_S		Clear conf
20		+CLEAR_DOWN			

Detailed comments: The calling party must subscribe to the CONF service.

Table 23: Test Step dynamic behaviour (Synchronization)

Test Step dynamic behaviour

Test Step Name : B_210901
Group : Synchronization/
Objective : Establish/Isolate/Reattach/Drop/Disconnect a conference:
Ensure that the remote parties are notified of the conference call progress
Default : CLEAR_DOWN_FAIL
Comments : Destination side description 1

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='1'B) START TWAIT			
2		L?SETUP (CR:=SETUP.CREF.CRV) CANCEL TWAIT	SU_SPE_R		Get call reference
3		L!ALERTING	AL_S		
4		L!CONNECT START T313	CN_S		
5		L?CONNECT_ACK CANCEL T313	CA_R		
6		L?NOTIFY	NO_CONF_R		Conference established
7		+CHECK_BCHANNEL		(P)	Check 3way conversation
8		L?NOTIFY	NO_ISO_R		
9		L?NOTIFY	NO_REATT_R		
10		L?DISCONNECT	DI_R		
11		L!RELEASE	RE_S		
12		L?RELEASE_COMP	RC_R		
13		?TIMEOUT TWAIT		F	No call delivered

Detailed comments: The calling party must subscribe to the CONF service.

Table 24: Test Step dynamic behaviour (Synchronization)

Test Step dynamic behaviour					
Test Step Name	: C_210901				
Group	: Synchronization/				
Objective	: Establish/Isolate/Reattach/Drop/Disconnect a conference: Ensure that the remote parties are notified of the conference call progress				
Default	: CLEAR_DOWN_FAIL				
Comments	: Destination side description 2				
Nr.	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		(FL:='1'B) START TWAIT			
2		L?SETUP (CR:=SETUP.CREF.CRV) CANCEL TWAIT	SU_SPE_R		Get call reference
3		L!ALERTING	AL_S		
4		L!CONNECT START T313	CN_S		
5		L?CONNECT_ACK CANCEL T313	CA_R		
6		L?NOTIFY	NO_CONF_R		Conference established
7		L?NOTIFY	NO_OTH_ADD_R		
8		+CHECK_BCHANNEL		(P)	Check 3way conversation
9		L?NOTIFY	NO_OTH_ISO_R		
10		L?NOTIFY	NO_OTH_REATT_R		
11		L?NOTIFY	NO_OTH_DISC_R		
12		L?DISCONNECT	DI_R		
13		L!RELEASE	RE_S		
14		L?RELEASE_COMP	RC_R		
15		?TIMEOUT TWAIT		F	No call delivered
Detailed comments:	The calling party must subscribe to the CONF service.				

Table 25: Test Step dynamic behaviour (Synchronization)

Test Step dynamic behaviour					
Test Step Name	: AT1_CLUC_1				
Group	: Synchronization/				
Objective	: To check that the circuit is idle				
Default	: T1_Default_1				
Comments	:				
Nr.	**Label**	**Behaviour Description**	**Constraints Ref**	**Verdict**	**Comments**
1		T1_MTC?TOKEN	TOK1		
2		START TWAIT			
3		LT1A?IAM	IAM_AT1		
4		LT1A!REL	REL_NU_T1A		
5		LT1A?RLC	RLC_AT1	P	
6		T1_MTC!TOKEN	TOK1		
Detailed comments:					

Table 26: Test Step dynamic behaviour (Communication)

Test Step dynamic behaviour

Test Step Name : A_110102
Group : Communication/
Objective : En-block sending:
Ensure that the same Call Reference can be used to make a call in opposite direction.
Default : CLEAR_DOWN_FAIL
Comments : Origination side description, first call, forward direction.

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='0'B, CR_VAL:=CR)			
2		L!SETUP START T303	SU_SPE_S(FL, CR_VAL)		
3		+OUTGOING_CALL_ALERT			
4		+CHECK_BCHANNEL		(P)	
5		L?DISCONNECT START T305	DI_S (FL, CR_VAL)		
6		+CLEAR_DOWN			

Detailed comments:

Table 27: Test Step dynamic behaviour (Communication)

Test Step dynamic behaviour

Test Step Name : B_110102
Group : Communication/
Objective : En-block sending:
Ensure that the same Call Reference can be used to make a call in opposite direction.
Default : CLEAR_DOWN_FAIL
Comments : Destination side description, first call, forward direction.

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='1'B) START TWAIT			
2		L?SETUP (CR=SETUP.CREF.CRV) CANCEL TWAIT	SU_SPE_R		Get call reference
3		CP_BC!INFO	INFO1(CR_VAL)		
4		L!ALERTING	AL_S		
5		L!CONNECT	CN_S		
6		L!CONNECT_ACK CANCELT313	CA_R		
7		+CHECK_BCHANNEL		(P)	
8		+CLEAR_DOWN			
7		?TIMEOUT TWAIT		F	No call delivered

Detailed comments:

Table 28: Test Step dynamic behaviour (Communication)

Test Step dynamic behaviour

Test Step Name : C_110102
Group : Communication/
Objective : En-block sending:
Ensure that the same Call Reference can be used to make a call in opposite direction.
Default : CLEAR_DOWN_FAIL
Comments : Origination side description, second call, backward direction.

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		CP_BC?INFO	INFO1(CR)		
2		(FL:='0'B, CR_VAL:=INFO.CR)			
3		L!SETUP START T303	SU_SPE_S(FL, CR_VAL)		
4		+OUTGOING_CALL_ALERT			
5		+CHECK_BCHANNEL		(P)	
6		L?DISCONNECT START T305	DI_S		
7		+CLEAR_DOWN			

Detailed comments:

Table 29: Test Step dynamic behaviour (Communication)

Test Step dynamic behaviour

Test Step Name : D_110102

Group : Communication/

Objective : En-block sending:
Ensure that the same Call Reference can be used to make a call in opposite direction.

Default : CLEAR_DOWN_FAIL

Comments : Destination side description, first call, backwards direction.

Nr.	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		(FL:='1'B) START TWAIT			
2		L?SETUP (CR=SETUP.CREF.CRV) CANCEL TWAIT	SU_SPE_R		Get call reference
3		L!ALERTING	AL_S		
4		L!CONNECT	CN_S		
5		L!CONNECT_ACK CANCELT313	CA_R		
6		+CHECK_BCHANNEL		(P)	
7		+CLEAR_DOWN			
8		?TIMEOUT TWAIT		F	No call delivered

Detailed comments:

History

Document history	
October 1995	First Edition
February 1996	Converted into Adobe Acrobat Portable Document Format (PDF)

ISBN 2-7437-0226-5
Dépôt légal : Octobre 1995