INTERNATIONAL TELECOMMUNICATION UNION

CCITT

X.290

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

DATA COMMUNICATION NETWORKS

OPEN SYSTEMS INTERCONNECTION (OSI) PROTOCOL SPECIFICATIONS, CONFORMANCE TESTING

OSI CONFORMANCE TESTING METHODOLOGY AND FRAMEWORK FOR PROTOCOL RECOMMENDATIONS FOR CCITT APPLICATIONS – GENERAL CONCEPTS

Recommendation X.290

Geneva, 1992

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation X.290 was prepared by Study Group VII and was approved under the Resolution No. 2 procedure on the 17th of January 1992.

CCITT NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication Administration and a recognized private operating agency.

© ITU 1992

All rights reserved. Unless otherwise specified no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

Recommendation X.290

OSI CONFORMANCE TESTING METHODOLOGY AND FRAMEWORK FOR PROTOCOL RECOMMENDATIONS FOR CCITT APPLICATIONS – GENERAL CONCEPTS [1)]

(revised 1991)

The CCITT,

considering

(a) the Recommendation X.200 defines the Reference Model of Open Systems Interconnection (OSI) for CCITT Applications;

(b) that the objective of OSI will not be completely achieved until systems can be tested to determine whether they conform to the relevant OSI protocol Recommendations;

(c) that standardized test suites should be developed for each OSI protocol Recommendation as a means to:

– obtain wide acceptance and confidence in conformance test results produced by different testers,

– provide confidence in the interoperability of equipments which passed the standardized conformance tests;

(d) the need for defining Recommendation to specify the framework and general principles for conformance testing;

(e) the need for defining Recommendation for the specification of conformance test suites and testing of protocol implementations,

unanimously declares the view

that the general principles, definition of terms and concepts of OSI protocol conformance testing shall be in accordance with this Recommendation.

CONTENTS

0 *Introduction*

1 *Scope*

2 *References*

3 *Definitions*

3.1 Reference model definitions

3.2 Terms defined in other CCITT Recommendations, International Standards and Technical Reports

3.3 Conformance testing definitions

3.4 Basic terms

---

1) Recommendations X.290 and ISO 9646-1. Information technology – Open Systems Interconnection – Conformance Testing Methodology and Framework – Part I: General Concepts are technically aligned.

3.5 Types of testing

3.6 Terminology of test suites

3.7 Terminology of results

3.8 Terminology of test methods

4 *Abbreviations*

5 *The meaning of conformance in OSI*

5.1 Introduction

5.2 Conformance requirements

5.3 Static conformance requirements

5.4 Dynamic conformance requirements

5.5 Protocol Implementation Conformance Statement (PICS)

5.6 A Conforming system

5.7 Interworking and conformance

6 *Conformance and testing*

6.1 Objectives of conformance testing

6.2 Protocol Implementation Extra Information for Testing (PIXIT)

6.3 Conformance assessment process overview

6.4 Use of BITs and capability tests in the test campaign

6.5 Analysis of results

7 *Test methods*

7.1 Introduction

7.2 Classification of real open systems and IUTs for conformance testing

7.3 Abstract testing methodology

7.4 Abstract testing functions

7.5 Overview of Abstract Test Methods

7.6 Applicability of test methods to real open systems

7.7 Applicability of the test methods to OSI protocols and layers

8 *Test suites*

8.1 Structure

8.2 Abstract and executable test cases

9 *Relationships between parts, concepts and roles*

10 *Compliance*

Annex A – Options

Annex B – Applicability of the test methods to OSI protocols

Annex C – Index to Recommendations X.290, X.291, X.293 and X.294

0 Introduction

The objective of OSI will not be completely achieved until systems can be tested to determine whether they conform to the relevant protocol specification(s). The relevant ones can be OSI CCITT Recommendations or International Standards.

Standardized abstract test suites should be developed for each CCITT Recommendation or International Standard which specifies an OSI protocol, for use by suppliers or implementors in self-testing, by users of OSI products, by telecommunications Administrations and recognized private operating agencies, or by other third party testing organizations. This should lead to comparability and wide acceptance of test results produced by different test laboratories, and thereby minimize the need for repeated conformance testing of the same system.

The standardization of test suites requires international definition and acceptance of a common testing methodology, together with appropriate testing methods and procedures. It is the purpose of Recommendations X.290 to X.294 to define the methodology, to provide a framework for specifying conformance test suites, and to define the procedures to be followed during testing.

Conformance testing involves testing both the capabilities and behaviour of an implementation, and checking what is observed against both the conformance requirements in the relevant CCITT Recommendations or International Standards and what the implementor states the implementation's capabilities are.

Conformance testing does not include assessment of the performance nor the robustness or reliability of an implementation. It cannot give judgements on the physical realization of the abstract service primitives, how a system is implemented, how it provides any requested service, nor the environment of the protocol implementation. It cannot, except in an indirect way, prove anything about the logical design of the protocol itself.

The purpose of conformance testing is to increase the probability that different OSI implementations are able to interwork. However it should be borne in mind that the complexity of most protocols makes exhaustive testing impractical on both technical and economic grounds. Also, testing cannot guarantee conformance to a specification since it detects errors rather than their absence. Thus conformance to a test suite alone cannot guarantee interworking. What it does do is give confidence that an implementation has the required capabilities and that its behaviour conforms consistently in representative instances of communication.

It should be noted that the OSI basic reference model [CCITT X.200(1988) or ISO 7498: 1984] states (in § 4.3):

"Only the external behaviour of Open Systems is retained as the standard of behaviour of real Open Systems"

This means that although aspects of both internal and external behaviour are described in OSI CCITT Recommendations and International Standards, it is only the requirements on external behaviour that have to be met by real open systems. Although some of the methods defined in Recommendation X.291 do impose certain limitations on the implementor, for example that there be some means of realizing control and observation at one or more service access points, it should be noted that other methods defined herein do not impose such limitations.

However, in the case of partial OSI end-systems which provide OSI protocols up to a specific layer boundary, it is desirable to test not only the external behaviour of the implemented protocol entities, but also the ability of those entities to support correct external behaviour in higher layers.

Detailed investigation of relative benefits, efficiency and limitations of all methods is addressed in various of the Recommendations X.290 to X.294. However, any organization contemplating the use of test methods defined in Recommendation X.291 in a context such as certification should carefully consider the limitations on their applicability and the benefits of each.

Testing is voluntary as far as CCITT and ISO/IEC are concerned. Requirements for testing in procurement and other external contracts are not a matter for standardization.

This Recommendation is also published as ISO/IEC 9646-1: 1991.

1 Scope

1.1 Recommendations X.290 to X.294 specify a general methodology for testing the conformance of products to CCITT Recommendations or International Standards that specify OSI protocols which the products are claimed to implement. The methodology also applies to testing conformance to an CCITT Recommendation or International Standard that specifies a transfer syntax to the extent that can be determined by testing it in combination with a specific OSI protocol.

1.2 The contents of Recommendations X.290 to X.294 are also, in principle, applicable to conformance testing for ISDN two-party protocols.

1.3 Recommendations X.290 to X.294 are applicable to the different phases of the conformance testing process, these phases being characterized by three major activities. These activities are

a) the specification of abstract test suites for particular OSI protocols;

b) realization of the means of executing specific test suites;

c) the conformance assessment process carried out by a test laboratory for a specific client, culminating in the production of a Protocol Conformance Test Report, which gives the results in terms of the protocol specification and test suite used.

The X.290 Series of Recommendations is structured into five separate Recommendations, each of which, apart from this Recommendation, is applicable to just one of these activities.

This Recommendation is applicable to all three activities, providing tutorial introductory material, together with definitions of common terms and concepts.

Note – Recommendation X.291 deals with the requirements and guidance for the specification of abstract test suites, independent of test notation. Recommendation X.292 defines the recommended test notation. Recommendation X.293 deals with requirements and guidance for realization of the Means of Testing, and Recommendation X.294 deals with requirements and guidance for test laboratories and their clients for the conformance assessment process.

1.4 Recommendations X.290 to X.294 specify the requirements for and gives guidance on the procedures to be followed in OSI conformance testing.

1.5 Recommendations X.290 to X.294 include only such information as is necessary to meet the following objectives:

a) to achieve an adequate level of confidence in the tests as a guide to conformance;

b) to achieve comparability between the results of the corresponding tests on a particular OSI implementation applied in different places at different times;

c) to facilitate communication between the parties responsible for the activities described in § 1.2 above for parts 2 to 5.

1.6 This Recommendation includes tutorial introductory material which provides

a) an exposition of the meaning of conformance in the context of OSI;

b) a description of the major categories of conformance tests;

c) an introduction to the conformance assessment process;

d) an introduction to the Abstract Test Methods and their applicability;

e) an introduction to the concepts of test suite design.

In addition, this Recommendation describes the relationship between Recommendations X.291 to X.294 and the activities involved in conformance testing, and introduces the concept of compliance with respect to Recommendations X.291 to X.294.

1.7 The following are outside the scope of Recommendations X.290 to X.294:

a) certification, an administrative procedure which may follow conformance testing;

b) requirements for procurement and contracts;

c) testing by means of test methods which are specific to particular applications, protocols or systems;

d) testing of non-protocol conformance requirements;

e) test methods that involve more than two end-systems communicating together.

Note – Recommendations X.290 to X.294 do not apply fully to Physical layer protocols. Nevertheless, many of the concepts apply to all protocols.

2 References

Rec. X.200 (1988) – *Reference Model of Open Systems Interconnection for CCITT Applications.* (See also ISO 7498: 1984)

Rec. X.210 (1988) – *Open Systems Interconnection Layer Service Definition Conventions.* (See also ISO/TR 8509: 1987)

Rec. X.209 (1988) – *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).* (See also ISO 8825: 1990)

Rec. X.291 (1992) – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – Abstract Test Suite specification.* (See also ISO/IEC 9646-2: 1991)

Rec. X.292 – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – The tree and tabular combined notation.*[2)]

Rec. X.293 (1992) – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – Test Realization.* (See also ISO/IEC 9646-4: 1991)

Rec. X.294 (1992) – *OSI Conformance Testing Methodology and Framework for Protocol Recommendations for CCITT Applications – Requirements on test laboratories and clients for the Conformance Assessment Process*. (See also ISO/IEC 9646-5: 1991)

3 Definitions

3.1 *Reference model definitions*

This Recommendation is based upon the concepts developed in the Basic Reference Model for Open Systems Interconnection (CCITT X.200 / ISO 7498), and makes use of the following terms defined in that Recommendation:

a) (N)-entity

b) (N)-layer

c) (N)-protocol

d) (N)-protocol-data-unit

e) (N)-relay

---

2) To be published.

f) (N)-service

g) (N)-service-access-point

h) Application layer

i) Application-service-element

j) Data Link layer

k) Network layer

l) Physical layer

m) Presentation layer

n) Real open system

o) Real system

p) Session layer

q) Subnetwork

r) Transfer syntax

s) Transport layer

3.2 *Terms defined in other CCITT Recommendations, International Standards and Technical Reports*

This Recommendation uses the following terms defined in the OSI Service Conventions (CCITT X.210/ ISO/TR 8509):

a) service-user

b) service-provider

c) service primitive

This Recommendation uses the following term defined in the ASN.1 - Basic Encoding Rules (CCITT X.209/ ISO 8825):

– encoding

3.3 *Conformance testing definitions*

For the purposes of this Recommendation the definitions in §§ 3.4 to 3.8 apply.

3.4 *Basic terms*

3.4.1 **implementation under test [IUT]**

An implementation of one or more OSI protocols in an adjacent user/provider relationship, being that part of a real open system which is to be studied by testing.

3.4.2 **system under test [SUT]**

The real open system in which the IUT resides.

3.4.3 **dynamic conformance requirement**

One of the requirements which specifies what observable behaviour is permitted by the relevant OSI CCITT Recommendation(s) or International Standard(s) in instances of communication.

3.4.4 **static conformance requirement**

One of the requirements that specify the limitations on the combinations of implemented capabilities permitted in a real open system which is claimed to conform to that OSI CCITT Recommendation or International Standard.

3.4.5 **capability (of an implementation)**

A set of functions in the relevant protocol(s) which is supported by the implementation.

3.4.6 **protocol implementation conformance statement [PICS]**

A statement made by the supplier of an OSI implementation or system, stating which capabilities have been implemented, for a given OSI protocol.

3.4.7 **PICS proforma**

A document, in the form of a questionnaire, designed by the protocol specifier or conformance test suite specifier, which when completed for an OSI implementation or system becomes the PICS.

3.4.8 **protocol implementation extra information for testing [PIXIT]**

A statement made by a supplier or implementor of an IUT which contains or references all of the information (in addition to that given in the PICS) related to the IUT and its testing environment, which will enable the test laboratory to run an appropriate test suite against the IUT.

3.4.9 **PIXIT proforma**

A document, in the form of a questionnaire, provided by the test laboratory, which when completed during the preparation for testing becomes a PIXIT.

3.4.10 **conforming implementation**

An IUT which satisfies both static and dynamic conformance requirements, consistent with the capabilities stated in the PICS.

3.4.11 **system conformance statement [SCS]**

A document summarizing which OSI CCITT Recommendations or International Standards are implemented and to which ones conformance is claimed.

3.4.12 **client (of a test laboratory)**

The organization that submits a system or implementation for conformance testing.

3.4.13 **test laboratory**

An organization that carries out conformance testing. This can be a third party, a user organization, a telecommunications Administration or recognized private operating agency, or an identifiable part of a supplier organization.

3.5 *Types of testing*

3.5.1 **static conformance review**

A review of the extent to which the static conformance requirements are met by the IUT, accomplished by comparing the PICS with the static conformance requirements expressed in the relevant CCITT Recommendation(s) or International Standard(s).

3.5.2 **basic interconnection test [BIT]**

A test of an IUT which has limited scope to determine whether or not there is sufficient conformance to the relevant protocol(s) for interconnection to be possible, without trying to perform thorough testing.

3.5.3 **capability test**

A test to verify the existence of one or more claimed capabilities of an IUT.

Note – Capability testing involves checking all mandatory capabilities and those optional ones that are stated in the PICS as supported, but not checking those optional ones which are stated in the PICS as not supported by the IUT.

3.5.4 **behaviour test**

A test to determine the extent to which one or more dynamic conformance requirements are met by the IUT.

3.5.5 **conformance resolution test**

A non-standardized, possibly system-specific test to fulfil a test purpose for which a standardized abstract test case is not defined, in order to investigate the behaviour of an OSI protocol implementation with respect to one or more particular conformance requirements.

3.5.6 **conformance testing**

Testing the extent to which an IUT is a conforming implementation.

3.5.7 **conformance assessment process**

The complete process of accomplishing all conformance testing activities necessary to enable the conformance of an implementation or a system to one or more OSI CCITT Recommendations or International Standards to be assessed.

3.5.8 **test campaign**

The process of executing the Parameterized Executable Test Suite for a particular IUT and producing the conformance log.

3.5.9 **embedded testing**

Testing specified for a single-protocol within a multi-protocol IUT including the specification of the protocol activity above the one being tested, but without specifying control or observation at service boundaries within the multi-protocol IUT.

Note – This definition assumes that the protocols of the IUT are ordered in a continuous adjacent user/provider relationship.

3.6 *Terminology of test suites*

3.6.1 **(abstract) test method [ATM]**

The description of how an IUT is to be tested, given at an appropriate level of abstraction to make the description independent of any particular realization of a Means of Testing, but with enough detail to enable tests to be specified for this test method.

3.6.2 **abstract testing methodology**

An approach to describing and categorizing Abstract Test Methods.

3.6.3 **abstract test case**

A complete and independent specification of the actions required to achieve a specific test purpose (or a specified combination of test purposes), defined at the level of abstraction of a particular Abstract Test Method, starting in a stable testing state and ending in a stable testing state. This specification may involve one or more consecutive or concurrent connections.

Note 1 – The specification should be complete in the sense that it is sufficient to enable a test verdict to be assigned unambiguously to each potentially observable test outcome (i.e. sequence of test events).

Note 2 – The specification should be independent in the sense that it should be possible to execute the derived executable test case in isolation from other such test cases (i.e. the specification should always include the possibility of starting and finishing in the "idle" state).

3.6.4 **executable test case**

A realization of an abstract test case.

Note – In general the use of the word "test" in Recommendations X.290 to X.294 will imply its normal English meaning. Sometimes it may be used as an abbreviation for abstract test case or executable test case. The context should make the meaning clear.

3.6.5 **test purpose**

A prose description of a narrowly defined objective of testing, focusing on a single conformance requirement as specified in the appropriate OSI CCITT Recommendation or International Standard (e.g. verifying the support of a specific value of a specific parameter).

3.6.6 **test group objective**

A prose description of the common objective which the test purposes within a specific test group are designed to achieve.

3.6.7 **generic test case**

A specification of the actions required to achieve a specific test purpose, defined by a test body together with a description of the initial testing state in which the test body is to start.

3.6.8 **(test) preamble**

The sequences of test steps from the starting stable testing state of the test case up to the initial testing state from which the test body will start.

3.6.9 **test body**

The sequences of test steps that achieve the test purpose.

3.6.10 **(test) postamble**

The sequences of test steps from the end of the test body up to the finishing stable testing state(s) for the test case.

3.6.11 **test step**

A named subdivision of a test case, constructed from test events and/or other test steps.

3.6.12 **test event**

An indivisible unit of test specification at the level of abstraction of the specification (e.g. sending or receiving a single PDU).

3.6.13 **unidentified test event**

A test event which is used to provide for receipt of PDUs and/or ASPs without identifying them in the test case.

Note – In TTCN, the unidentified test event is the Otherwise statement.

3.6.14 **testing state**

A state encountered during testing, comprising the combination of the states of the SUT, the test system, the protocols for which control and observation is specified in the ATS, and, if relevant, the state of the underlying service.

3.6.15 **stable testing state**

A testing state which can be maintained, without prescribed Lower Tester behaviour, sufficiently long to span the gap between one test case and the next in a test campaign.

3.6.16 **idle testing state**

A stable testing state in which there is no established connection of the relevant protocol(s) and in which the state of the SUT is independent of any previously executed test cases.

3.6.17 **transient testing state**

Any testing state which is not a stable testing state.

Note – Transient testing states include those testing states that are in the middle of a logical exchange of PDUs (e.g. to establish a connection or perform negotiation), particularly when a request PDU has been sent (or received) and the corresponding response PDU has not been received (or sent).

3.6.18 **initial testing state**

The testing state in which a test body starts.

Note – This may be either a stable testing state or a transient state.

3.6.19 **(conformance) test suite**

The complete set of test cases, possibly combined into nested test groups, that is needed to perform dynamic conformance testing for one or more OSI protocols.

Note – It should cover both capability testing and behaviour testing. It may be qualified by the adjectives: abstract, generic or executable, as appropriate. Unless stated otherwise, an abstract test suite is meant.

3.6.20 **test case**

A generic, abstract or executable test case.

3.6.21 **test group**

A named set of related test cases.

3.6.22 **generic test suite**

A test suite composed of generic test cases.

3.6.23 **abstract test suite [ATS]**

A test suite composed of abstract test cases.

3.6.24 **executable test suite [ETS]**

A test suite composed of executable test cases.

3.6.25 **selected abstract test suite [SATS]**

The subset of an ATS selected using a specific PICS and PIXIT.

3.6.26 **selected executable test suite [SETS]**

The subset of an ETS selected using a specific PICS and PIXIT.

3.6.27 **parameterized abstract test case**

An abstract test case in which all appropriate parameters have been supplied with values in accordance with a specific PICS and PIXIT.

3.6.28 **parameterized executable test case**

An executable test case, in which all appropriate parameters have been supplied with values in accordance with a specific PICS and PIXIT, and corresponding to a parameterized abstract test case.

3.6.29 **parameterized abstract test suite [PATS]**

A SATS in which all test cases have been parameterized in accordance with the appropriate PICS and PIXIT.

3.6.30 **parameterized executable test suite [PETS]**

A SETS, in which all test cases have been parameterized in accordance with the appropriate PICS and PIXIT, and corresponding to a PATS.

3.6.31 **standardized abstract test suite**

An abstract test suite specified within an CCITT Recommendation or International Standard or, in the absence of such a CCITT Recommendation or International Standard, within a publicly available document which is in the process of being standardized within CCITT or ISO/IEC, and which has the highest standardization status currently available, having reached at least the committee draft, draft proposal or draft Recommendation status.

3.6.32 **conformance testing standard**

The CCITT Recommendation or International Standard or draft thereof that contains a standardized ATS.

3.7 *Terminology of results*

3.7.1 **repeatability (of results)**

Characteristic of a test case, such that repeated executions on the same IUT under the same conditions lead to the same test verdict, and by extension a characteristic of a test suite.

3.7.2 **comparability (of results)**

Characteristic of conformance assessment processes, such that their execution on the same IUT, in different test environments, leads to the same overall summary of conformance for the specified IUT.

3.7.3 **(observed) test outcome**

The sequence of test events, together with associated data and/or parameter values, which occurred during test execution of a specific parameterized executable test case.

3.7.4 **foreseen test outcome**

An observed test outcome identified in the abstract test case.

Note – A foreseen test outcome may include an unidentified test event.

3.7.5 **unforeseen test outcome**

An observed test outcome not identified in the abstract test case.

Note – An unforeseen test outcome can only lead to a test case error or an abnormal test case termination.

3.7.6 **(test) verdict**

A statement of "pass", "fail" or "inconclusive", specified in an abstract test case, concerning conformance of an IUT with respect to that test case when it is executed.

3.7.7 **system conformance test report [SCTR]**

A document, written at the end of the conformance assessment process, giving an overall summary of the conformance of the system or implementation to the set of protocols for which conformance testing was carried out.

3.7.8 **protocol conformance test report [PCTR]**

A document, written at the end of the conformance assessment process, giving the details of the testing carried out for a particular protocol. It lists all of the abstract test cases and identifies those for which corresponding executable test cases were run, together with the verdicts assigned to each test case executed.

3.7.9 **valid test event**

A test event which is allowed by the protocol specification, being both syntactically and semantically correct, and occurring when allowed to do so by the protocol specification.

3.7.10 **invalid test event**

A test event that violates at least one conformance requirement of the relevant protocol or transfer syntax specification.

Note – This term is not to be confused with the term "invalid event" as defined in ISO 7776.

3.7.11 **inopportune test event**

A test event which occurs when not allowed to do so by the protocol specification.

3.7.12 **syntactically invalid test event**

A test event which syntactically is not allowed by the protocol specification.

3.7.13 **semantically invalid test event**

A test event which is neither inopportune nor syntactically invalid, but which contains a semantic error with respect to the relevant protocol specification (e.g. a PDU containing a parameter value outside the negotiated range for that parameter).

3.7.14 **pass (verdict)**

A test verdict given when the observed test outcome gives evidence of conformance to the conformance requirement(s) on which the test purpose(s) of the test case is (are) focused, and when all the test events are valid with respect to the relevant CCITT Recommendation(s) or International Standard(s).

3.7.15 **fail (verdict)**

A test verdict given when the observed test outcome either demonstrates non-conformance with respect to (at least one of) the conformance requirement(s) on which the test purpose(s) of the test case is (are) focused, or contains at least one invalid test event, with respect to the relevant CCITT Recommendation(s) or International Standard(s).

3.7.16 **inconclusive (verdict)**

A test verdict given when the observed test outcome is such that neither a pass nor a fail verdict can be given.

3.7.17 **test case error**

The term used to describe the result of execution of a test case when an error is detected in the test case itself.

3.7.18 **abstract test case error**

A test case error resulting from an error in the abstract test case.

3.7.19 **executable test case error**

A test case error in the realization of an abstract test case.

3.7.20 **abnormal (test case) termination**

The term used to describe the result of execution of an abstract test case when it has been prematurely terminated by the test system.

3.7.21 **conformance log**

A human-readable record of information produced as a result of a test campaign, which is sufficient to record the observed test outcomes and verify the assignment of test results (including test verdicts).

3.8 *Terminology of test methods*

3.8.1 **point of control and observation (PCO)**

A point within a testing environment where the occurrence of test events is to be controlled and observed, as defined in an Abstract Test Method.

Note – A PCO is characterized by the set of ASPs and/or PDUs that can occur, according to the ATS, at that PCO.

3.8.2 **lower tester (LT)**

The representation in Recommendations X.290 to X.294 of the means of providing, during test execution, indirect control and observation of the lower service boundary of the IUT via the underlying service-provider.

Note – The underlying service-provider is immediately beneath the (lowest layer) protocol which is the focus of testing. It may use one or more OSI layers, or only the Physical medium.

3.8.3 **upper tester (UT)**

The representation in Recommendations X.290 to X.294 of the means of providing, during test execution, control and observation of the upper service boundary of the IUT, as defined by the chosen Abstract Test Method.

3.8.4 **abstract (N)-service-primitive [(N)-ASP]**

An implementation-independent description of an interaction between a service-user and a service-provider at an (N)-service boundary, as defined in an OSI service definition.

3.8.5 **test coordination procedures**

The rules for cooperation between the Lower and Upper Testers during testing.

3.8.6 **test management protocol [TMP]**

A protocol which is used in the test coordination procedures for a particular test suite.

3.8.7 **test system**

The real system which includes the realization of the Lower Tester.

Note – The same test system can be used as part of several Means of Testing.

3.8.8 **local test method**

An Abstract Test Method in which both the Lower and Upper Testers are located within the test system and there is a PCO at the upper service boundary of the IUT.

3.8.9 **distributed test method**

An Abstract Test Method in which the Upper Tester is within the SUT and there is a PCO at the upper service boundary of the IUT.

3.8.10 **coordinated test method**

An Abstract Test Method in which the Upper Tester is within the SUT and for which a standardized TMP is defined for the test coordination procedures, enabling the control and observation to be specified solely in terms of the Lower Tester activity, including the control and observation of Test Management PDUs.

3.8.11 **remote test method**

An Abstract Test Method in which the control and observation of test events is specified solely in terms of Lower Tester activity, and in which some requirements for test coordination procedures may be implied or informally expressed in the ATS, but in which no assumption is made regarding their feasibility or realization.

3.8.12 **means of testing (MOT) (IUTs)**

The combination of equipment and procedures that can perform the derivation, selection, parameterization and execution of test cases, in conformance with a reference standardized ATS, and can produce a conformance log.

3.8.13 **test realization**

The process of producing a Means of Testing IUTs.

3.8.14 **reference standardized (OSI) abstract test suite (ATS)**

The standardized ATS for which a Means of Testing is realized.

3.8.15 **test realizer**

An organization which takes responsibility for providing, in a form independent of the clients of a test laboratory and their IUTs, a Means of Testing IUTs in conformance with an ATS.

3.8.16 **comprehensive testing service**

A service, offered to clients by a test laboratory, to perform the conformance assessment process for one or more OSI protocol(s), with a choice of test methods sufficient to make the service applicable to all real open systems that claim to implement the specified protocols.

4 Abbreviations

For the purposes of Recommendation X.290 the following abbreviations apply.

ACSE	Association control service element
ASE	Application-service-element
ASN.1	Abstract syntax notation one
ASP	Abstract-service-primitive
ATM	Abstract test method
ATS	Abstract test suite
BIT	Basic interconnection test
DTE	Data terminal equipment
ETS	Executable test suite
ISDN	Integrated services digital network
IUT	Implementation under test
LT	Lower tester
MOT	Means of testing
OSI	Open systems interconnection
PATS	Parameterized abstract test suite
PCO	Point of control and observation
PCTR	Protocol conformance test report
PDU	Protocol data unit
PETS	Parameterized executable test suite
PICS	Protocol implementation conformance statement
PIXIT	Protocol implementation extra information for testing
SAP	Service-access-point
SATS	Selected abstract test suite
SCS	System conformance statement
SCTR	System conformance test report
SETS	Selected executable test suite
SUT	System under test
TMP	Test management protocol
TM-PDU	Test management PDU
TTCN	Tree and tabular combined notation
UT	Upper tester

5 The meaning of conformance in OSI

5.1 *Introduction*

In the context of OSI, a real system is said to exhibit conformance if it complies with the requirements of applicable OSI CCITT Recommendations or International Standards in its communication with other real systems.

Applicable OSI CCITT Recommendations or International Standards include those that specify OSI protocols, and those that specify OSI transfer syntaxes (insofar as they are implemented in conjunction with protocols).

OSI CCITT Recommendations and International Standards form a set of inter-related CCITT Recommendations and International Standards which together define behaviour of open systems in their communication. Conformance of a real system is, therefore, expressed at two levels, conformance to each individual CCITT Recommendation or International Standard, and conformance to the set.

Note – If the implementation is based on a predefined set of CCITT Recommendations or International Standards, often referred to as a functional standard or profile, the concept of conformance can be extended to specific requirements expressed in the functional standard or profile, as long as they do not conflict with the requirements of the base (protocol) CCITT Recommendations and International Standards.

5.2 *Conformance requirements*

5.2.1 The conformance requirements in an CCITT Recommendation or International Standard can be

a) *mandatory requirements:* these are to be observed in all cases;

b) *conditional requirements:* these are to be observed if the conditions set out in the specification apply;

c) *options:* these can be selected to suit the implementation, provided that any requirements applicable to the option are observed. More information on options is provided in Annex A.

For example, CCITT essential facilities (E) are mandatory requirements; additional facilities (A) can be either conditional or optional requirements.

Note – The CCITT terms “essential facilities” and “additional facilities” need to be considered in the context of the scope of the CCITT Recommendation concerned; for example, in many cases, essential facilities are mandatory for networks but not for DTEs.

5.2.2 Furthermore, conformance requirements in an CCITT Recommendation or International Standard can be stated

a) *positively:* they state what shall be done;

b) *negatively:* they state what shall not be done.

5.2.3 Finally, conformance requirements fall into two groups:

a) static conformance requirements;

b) dynamic conformance requirements.

These are discussed in §§ 5.3 and 5.4, respectively.

5.3 *Static conformance requirements*

Static conformance requirements are those that specify the limitations on the combinations of implemented capabilities permitted in a real system which is claimed to conform to the relevant OSI CCITT Recommendation or International Standard. They define the allowed minimum capabilities in order to facilitate interworking. They may be specified at a broad level, such as the grouping of PDUs into functional units or protocol classes, or at a detailed level, such as a range of values that have to be supported for specific parameters or timers.

Static conformance requirements in OSI CCITT Recommendations or International Standards can be of two varieties:

a) those which determine the capabilities to be included in the implementation of the particular protocol;

b) those which determine multi-layer dependencies, e.g. those which place limitations on the capabilities of the underlying layers of the system in which the protocol implementation resides. These are likely to be found in upper layer CCITT Recommendations or International Standards.

5.4 *Dynamic conformance requirements*

Dynamic conformance requirements are all those requirements which specify what observable behaviour is permitted by the relevant OSI CCITT Recommendation(s) or International Standard(s) in instances of communication. They form the bulk of each OSI protocol specification. They define the set of allowable behaviours of an implementation or real system. This set of allowable behaviours implicitly defines the maximum set of capabilities, related to the use of the OSI protocol, that a conforming implementation or real open system can have.

A system exhibits dynamic conformance in an instance of communication if its behaviour is a member of the set of all behaviours permitted by the relevant OSI protocol specification in a manner which is consistent with the static conformance requirements.

Dynamic conformance requirements are those that define the actual protocol: the use and format of its PDUs, state transitions, negotiation rules, etc. They are usually structured according to the major capabilities (e.g. functional units) that are the subject of the main static conformance requirements.

5.5 *Protocol Implementation Conformance Statement (PICS)*

To evaluate the conformance of a particular implementation, it is necessary to have a statement of the capabilities and options which have been implemented, for the relevant protocol, so that the implementation can be tested for conformance against relevant requirements, and against those requirements only. Such a statement is called a Protocol Implementation Conformance Statement (PICS).

In a PICS there should be a distinction between the following categories of information which it may contain:

a) information related to the mandatory, optional and conditional static conformance requirements of the protocol itself;

b) information related to the mandatory, optional and conditional static conformance requirements for multi-layer dependencies.

If a set of interrelated OSI protocols has been implemented in a system, a PICS is needed for each protocol. A System Conformance Statement will also be necessary, itemizing all protocols in the system for which a distinct PICS is provided.

5.6 *A Conforming system*

A conforming system or implementation is one which satisfies both static and dynamic conformance requirements, consistent with the capabilities stated in the PICS, for each protocol declared in the System Conformance Statement.

5.7 *Interworking and conformance*

5.7.1 The primary purpose of conformance testing is to increase the probability that different implementations are able to interwork.

5.7.2 While conformance is a necessary condition, it is not on its own a sufficient condition to guarantee interworking capability. Even if two implementations conform to the same OSI protocol specification, they may fail to interwork fully. Trial interworking is therefore recommended.

5.7.3 Successful interworking of two or more real open systems is more likely to be achieved if they all conform to the same subset of an OSI CCITT Recommendation or International Standard, or to the same selection of OSI CCITT Recommendations or International Standards, than if they do not.

In order to prepare two or more systems for successful interworking, it is recommended that a comparison be made of the System Conformance Statements and PICSs of these systems.

If the PICSs indicate that different subsets or versions of the OSI CCITT Recommendations or International Standards have been implemented, the nature of the differences and their implications for interworking need to be determined. This study should be undertaken both for the options in the protocols themselves, and for the combined use of the protocols in an OSI system.

5.7.4 Further information to assist interworking between two systems can be obtained by comparing other relevant information, including test reports and PIXIT (see 6.2). The comparison can focus on

a) additional mechanisms claimed to work around known ambiguities or deficiencies not yet corrected in the CCITT Recommendations or International Standards or in other real open systems with which interworking is desired, e.g. the solution of multi-layer problems;

b) selection of optional capabilities which are not constrained by the static conformance requirements of the CCITT Recommendations or International Standards (i.e. where the implementor has a free choice, e.g. the provision of an inactivity timer of a specific duration).

Note – The comparison can be made between two individual systems, between two or more types of product, or, for the PICS comparison only, between two or more specifications for procurement, permissions to connect, etc.

5.7.5 Recommendation X.294 (1992), Annex A, specifies a System Conformance Test Report proforma which includes in § 1.5 an appropriate warning, stating the limits of conformance testing with respect to interworking.

6 Conformance and testing

6.1 *Objectives of conformance testing*

6.1.1 *Introduction*

Conformance testing as discussed in Recommendations X.290 to X.294 is focused on testing for conformance to CCITT Recommendations or International Standards which specify OSI protocols. However, it also applies to testing for conformance to CCITT Recommendations or International Standards which specify OSI transfer syntaxes, to the extent that this can be carried out by testing the transfer syntax in combination with an OSI protocol.

In principle, the objective of conformance testing is to establish whether the implementation being tested conforms to the specification in the relevant CCITT Recommendation or International Standard. Practical limitations make it impossible to be exhaustive, and economic considerations may restrict testing still further.

Therefore, Recommendations X.290 to X.294 distinguish four types of testing, according to the extent to which they provide an indication of conformance:

a) basic interconnection tests, which provide *prima facie* evidence that an IUT conforms;

b) capability tests, which check that the observable capabilities of the IUT are in accordance with the static conformance requirements and the capabilities claimed in the PICS;

c) behaviour tests, which endeavour to provide testing which is as comprehensive as possible over the full range of dynamic conformance requirements specified by the CCITT Recommendation or International Standard, within the capabilities of the IUT;

d) conformance resolution tests, which probe in depth the conformance of an IUT to particular requirements, to provide a definite yes/no answer and diagnostic information in relation to specific conformance issues; such tests are not standardized.

Note – As a by-product of conformance testing, errors and deficiencies in protocol specifications may be identified.

6.1.2 *Basic Interconnection Tests (BITs)*

6.1.2.1 BITs provide limited testing of an IUT in relation to the main features in a protocol and/or transfer syntax specification, to establish that there is sufficient conformance for interconnection to be possible, without trying to perform thorough testing.

6.1.2.2 BITs are appropriate

a) for detecting severe cases of non-conformance;

b) as a preliminary step to decide whether or not to run further capability and behaviour tests;

c) for checking addressing and other matters concerned with the test environment;

d) for use by users of implementations, to determine whether the implementations appear to be usable for communication with other conforming implementations, e.g. as a preliminary to data interchange.

6.1.2.3 BITs are inappropriate

a) when taken on their own, as a basis for claims of conformance by the supplier of an implementation;

b) as a means to determine causes for communications failure.

6.1.2.4 Standardized BITs are always drawn from the set of capability and behaviour tests comprising a standardized ATS. A list of which tests are appropriate may optionally be included. To provide additional tests in the standardized ATS for this purpose would contravene the requirements stated in Recommendation X.291 (1992).

6.1.3 *Capability tests*

6.1.3.1 Capability tests provide limited testing of each of the static conformance requirements in a protocol and/or transfer syntax specification, to ascertain which capabilities stated in the PICS can be observed and to check that those observable capabilities are valid with respect to the static conformance requirements.

6.1.3.2 Capability tests are appropriate

a) to check that the capabilities of the IUT are consistent with the static conformance requirements;

b) to check as far as possible the consistency of the PICS with the IUT;

c) when taken together with behaviour tests, as a basis for claims of conformance.

6.1.3.3 Capability tests are inappropriate

a) on their own, as a basis for claims of conformance by the supplier of an implementation;

b) for testing in detail the behaviour associated with each capability which has been implemented or has not been implemented;

c) for resolution of problems experienced during live usage or where other tests indicate possible non-conformance even though the capability tests have been satisfied.

6.1.3.4 Capability tests are standardized within a standardized ATS, and may be executed either as a separate test group or together with the behaviour tests.

6.1.4 *Behaviour tests*

6.1.4.1 Behaviour tests test an implementation as thoroughly as is practical, over the full range of dynamic conformance requirements specified in a protocol and/or transfer syntax specification.

6.1.4.2 Behaviour tests are appropriate, when taken together with capability tests, as a basis for the conformance assessment process.

6.1.4.3 Behaviour tests are inappropriate for resolution of problems experienced during live usage or where other tests indicate possible non-conformance even though the behaviour tests have been satisfied.

6.1.4.4 Behaviour tests are standardized as the major part of a standardized ATS.

6.1.4.5 Behaviour tests include tests for valid behaviour by the IUT in response to valid and invalid protocol behaviour by the Lower Tester.

6.1.5 *Conformance resolution tests*

6.1.5.1 Conformance resolution tests are non-standardized, possibly system-specific, tests to fulfil test purposes for which standardized abstract test cases are not defined. They may be used to complement the standardized tests used in the conformance assessment process, in order to investigate the behaviour of an IUT with respect to particular conformance requirements.

6.1.5.2 The test architecture and test method will normally be chosen specifically for the requirements to be tested, and need not be ones that are generally useful for other requirements. They may even be ones that are regarded as being unacceptable for standardized ATSs, e.g. involving implementation-specific methods using, say, the diagnostic and debugging facilities of the specific operating system.

6.1.5.3 In particular, conformance resolution tests may include SUT-specific means of controlling the occurrence of internal events and states (e.g. internally generated reset or the "busy" state) in order to test aspects of the protocol which are untestable using a standardized ATS.

6.1.5.4 *Conformance resolution tests are appropriate*

a) to fulfil test purposes that are identified in the relevant standardized test purposes, but for which no test cases can be included in the standardized ATS because of limitations of the chosen test method or the general untestability of the conformance requirement;

b) for providing a yes/no answer in a strictly confined and previously identified situation (e.g. during implementation development, to check whether a particular feature has been correctly implemented, or during operational use, to investigate the cause of problems).

c) to investigate the problems encountered in the execution of a standardized ATS.

6.1.5.5 Conformance resolution tests are inappropriate as a basis for judging whether or not an implementation conforms overall.

6.1.5.6 Conformance resolution tests are not standardized.

6.2 *Protocol Implementation Extra Information for Testing (PIXIT)*

In order to test a protocol implementation, the test laboratory will require information relating to the IUT and its testing environment in addition to that provided by the PICS. This "Protocol Implementation extra Information for Testing" (PIXIT) will be provided by the client submitting the implementation for testing, as a result of completing the PIXIT proforma supplied by the test laboratory.

The PIXIT may contain the following:

a) information about the SUT which is needed by the test laboratory in order to be able to run the appropriate test suite against that SUT (e.g. addressing information, information related to the realization of the Upper Tester within the SUT, etc.);

b) information which adds precision to the information given in the PICS (e.g. if the PICS states that a range of values is supported for a particular parameter or timer, the PIXIT can give a specific value which is to be used in all relevant test cases other than those that are designed to test different values for the same parameter or timer);

c) information to help determine which capabilities stated in the PICS as being supported, are testable and which are untestable;

d) other administrative information (e.g. IUT identification information, reference to the related PICS, etc.).

The PIXIT should not conflict with the related PICS.

There is one PIXIT for each ATS to be run against an IUT. The ATS specifier, test realizer and test laboratory all contribute to the development of the PIXIT proforma.

6.3 *Conformance assessment process overview*

6.3.1 The conformance assessment process is the complete process of accomplishing all conformance testing activities necessary to enable the conformance of an implementation or system to one or more OSI CCITT Recommendations or International Standards to be assessed.

The conformance assessment process involves three phases:

a) preparation for testing;

b) test operations;

c) test report production.

6.3.2 *The preparation for testing phase involves*

a) production of the System Conformance Statement, PICS and PIXIT;

b) choice of Abstract Test Method and ATS, based on those documents;

c) preparation of the SUT and Means of Testing.

6.3.3 *The test operations phase involves*

a) a static conformance review, conducted by analysing the PICS with respect to the relevant static conformance requirements;

b) test selection and parameterization based on the PICS and PIXIT;

c) one or more “test campaigns”.

A test campaign is the process of executing the Parameterized Executable Test Suite (PETS), produced as a result of the test selection and parameterization steps, and recording the observed sequences of test events and any other relevant information in a conformance log.

A test campaign involves the use of a configuration of equipment allowing protocol exchanges to take place between the SUT and a test system, such exchanges being controlled by the test system. A test campaign includes the following three types of test:

– BITs (optional);

– capability tests; and

– behaviour tests.

Note – These tests need not be performed in separate test groups.

The test operations phase culminates in the analysis of results and this leads on to the test report production phase. These activities are discussed in § 6.5.

6.3.4 *The overview given above of the conformance assessment process is illustrated in Figure 1/X.290.*

The requirements to be met by the test laboratory and its client during the conformance assessment process are specified in Recommendation X.294.

6.4 *Use of BITs and capability tests in the test campaign*

6.4.1 If the conformance testing standard identifies a list of tests to be used as BITs, the test laboratory can use them as a preliminary filter in the test campaign. This shall be decided by agreement between the test laboratory and client prior to the start of the test campaign.

6.4.2 It is optional whether the capability tests are executed as a separate test group or as part of the behaviour tests. This should be decided by agreement between the test laboratory and client prior to the start of the test campaign.

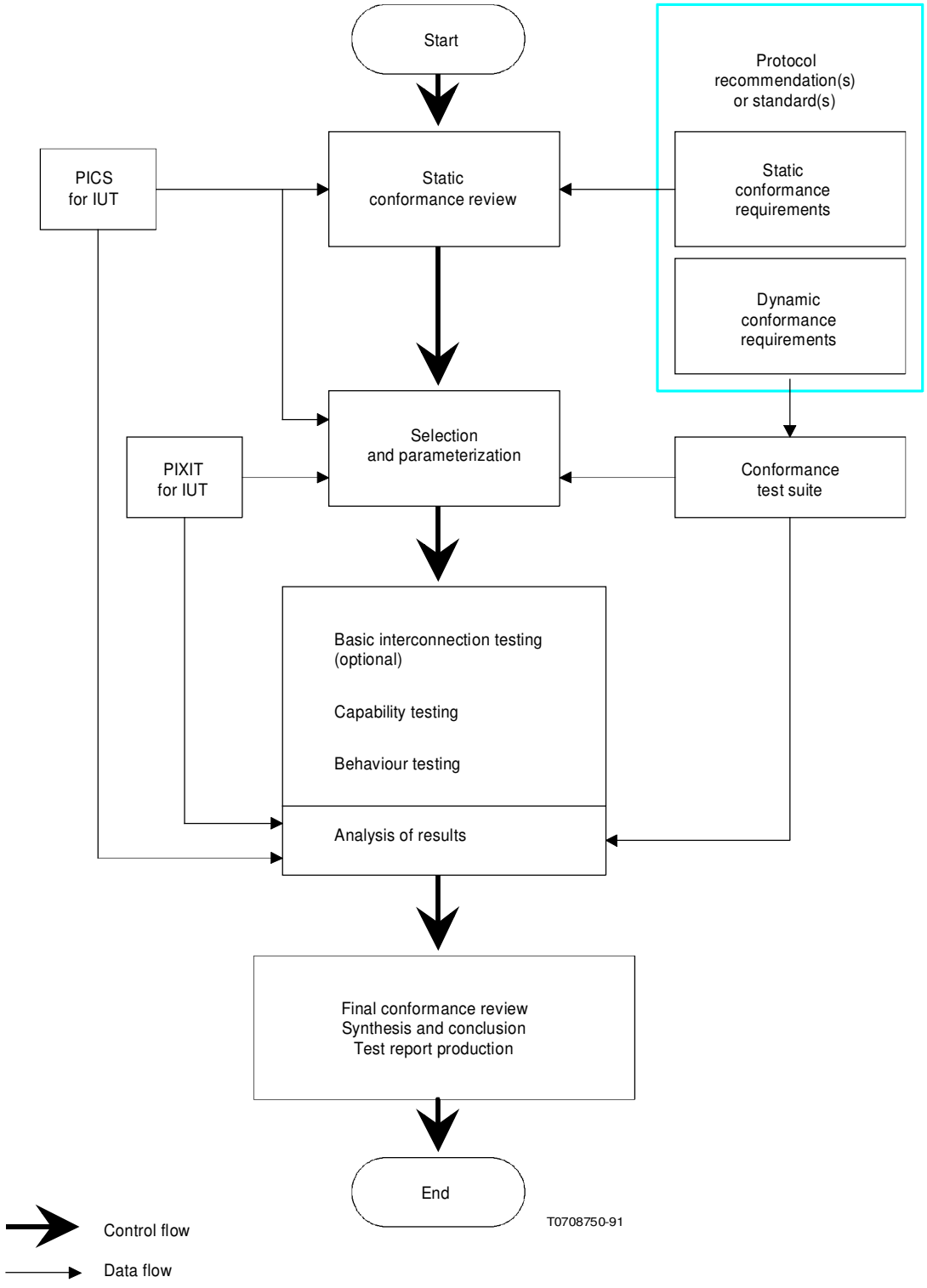

FIGURE 1/X.290

Conformance assessment process overview

6.5 *Analysis of results*

6.5.1 *Test outcomes and test verdicts*

6.5.1.1 The test outcome is the series of events which occurred during execution of a test case; it includes all input to and output from the IUT at the points of control and observation.

6.5.1.2 A foreseen test outcome is one which has been defined by the abstract test case; i.e the events which occurred during execution of the test case matched a sequence of test events defined in the abstract test case. A foreseen test outcome may include unidentified test events, and always results in the assignment of a test verdict to the test case.

6.5.1.3 The test verdict will be pass, fail or inconclusive:

a) pass means that the observed test outcome gives evidence of conformance to the conformance requirement(s) on which the test purpose(s) of the test case is (are) focused, and is valid with respect to the relevant CCITT Recommendation(s) or International Standard(s);

b) fail means that the observed test outcome either demonstrates non-conformance with respect to (at least one of) the conformance requirement(s) on which the test purpose(s) of the test case is (are) focused, or contains at least one invalid test event, with respect to the relevant CCITT Recommendation(s) or International Standard(s);

c) inconclusive means that the observed test outcome is such that neither a pass nor a fail verdict can be given.

6.5.1.4 An unforeseen test outcome is one which has not been identified by the abstract test case; i.e. the events which occurred during execution of the test case did not match any sequence of test events defined in the abstract test case. An unforeseen test outcome always results in the recording of a test case error or an abnormal test case termination for the test case.

6.5.1.5 A test case error is recorded if an error is detected in either the abstract test case itself (i.e an abstract test case error) or in its realization (i.e an executable test case error).

6.5.1.6 An abnormal test case termination is recorded if the execution of the test case is prematurely terminated by the test system for reasons other than test case error.

6.5.1.7 The results of executing the relevant individual test cases will be recorded in an overall summary for the IUT.

6.5.2 *Conformance test reports*

The results of conformance testing will be documented in a set of conformance test reports. These reports will be of two types: a System Conformance Test Report (SCTR), and a Protocol Conformance Test Report (PCTR).

The SCTR, which will always be provided, gives a summary of the conformance status of the SUT, including a summary of the verdicts assigned during the conformance assessment process. A proforma for the SCTR can be found in Recommendation X.294.

The PCTR, one of which will be issued for each protocol tested in the SUT, documents all of the results of the test cases giving a reference to the conformance log(s) which contain(s) the observed test outcomes. The PCTR also gives reference to all necessary documents relating to the conduct of the conformance assessment process for that protocol.

A proforma for the PCTR can be found in Recommendation X.294. The order in which test cases are to be presented in the PCTR is specified in the conformance testing standard.

6.5.3 *Repeatability of results*

In order to achieve the objective of credible conformance testing, it is clear that the result of executing a test case on an IUT should be the same whenever it is performed. Experience shows that it may not be possible to execute a complete conformance test suite and observe test outcomes which are identical to those obtained on another occasion.

Nevertheless, at the test case level, it is very important that every effort be made by the test specifiers and test laboratories to minimize the possibility that a test case produce different test outcomes on different occasions.

6.5.4 *Comparability of results*

The standardization of all of the procedures concerned with conformance testing should result in comparable test reports being accorded to the IUT, whether the testing is done by the supplier, a user or a third party test laboratory. There are a large number of factors to be considered to achieve this, of which some of the more important are

a) careful design and unambiguous specification of the test cases to give flexibility where appropriate, but showing which requirements have to be met, and how the verdicts are to be assigned;

b) careful specification of the Means of Testing which should be used to run the test suite; this specification should give flexibility where appropriate, but should meet the requirements of the test suite, including all test coordination procedures (if any);

c) careful specification of the procedures to be followed by test laboratories regarding the repetition of a test case before recording a verdict for that test case;

d) a proforma for a conformance test report;

e) careful specification of the procedures necessary when reviewing results and preparing test reports.

6.5.5 *Auditability of results*

It may be necessary to review the observed test outcomes from the execution of a conformance test suite in order to make sure that all procedures have been correctly followed. Whether or not analysis of results is carried out in a manual or automatic mode, it is essential that all inputs, outputs, and other test events be logged, for each test case being run. It is the responsibility of the test laboratory to retain sufficient information to be able to produce a conformance log for each test campaign, for future reference.

7 Test methods

7.1 *Introduction*

Real systems that contain OSI protocol implementations come in a wide variety of configurations and vary in the ways in which their behaviour can be controlled and observed during testing. A range of test methods is, therefore, defined, to correspond to the range of possibilities for control and observation of the System under Test (SUT).

This section first characterizes the features of the SUT which are to be taken into consideration, next defines the possible test methods in abstract terms, and finally provides guidance on their applicability to real systems.

7.2 *Classification of real open systems and IUTs for conformance testing*

7.2.1 *Classification of SUTs*

7.2.1.1 There is a relationship between the test methods and the configurations of the real open systems to be tested. The appropriate test methods vary according to:

a) the main function of the system (end-system or relay-system);

b) which layers use OSI protocols;

c) whether the alternative of non-OSI protocols is also available.

7.2.1.2 The following configurations of systems have been identified for the purposes of conformance testing, as illustrated in Figures 2/X.290 to 4/X.290. Configurations 1 to 3 are the basic configurations of SUTs:

a) *Configuration 1: 7-layer open system (end-system)*

These systems use OSI standardized protocols in all seven layers.

b) *Configuration 2: Partial (N)-open system (end-system)*

These systems use OSI standardized protocols in layers 1 to N.

c) *Configuration 3: Open relay-systems*

These use OSI protocols in layers 1 to 3 (Network relay-systems) or 1 to 7 (Application relay-systems).

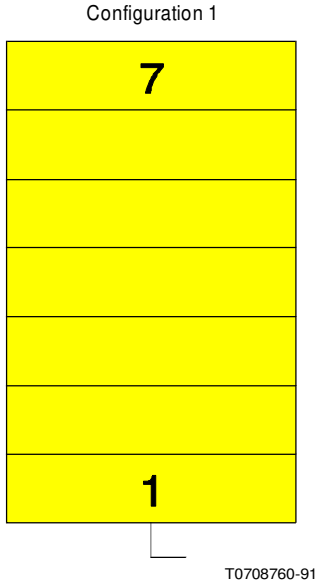

FIGURE 2/X.290

7-layer open system

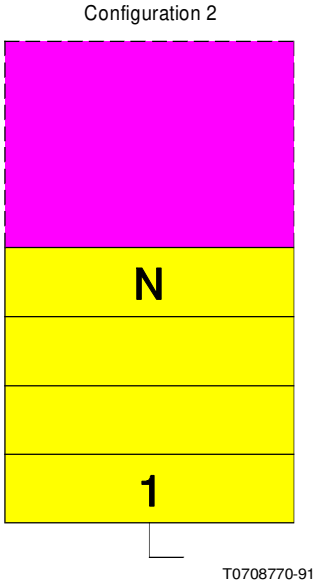

FIGURE 3/X.290

Partial (N)-open system

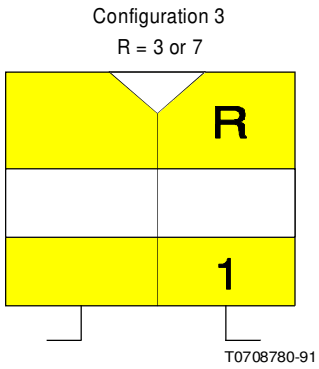

FIGURE 4/X.290

Open-relay system

7.2.1.3 Other configurations can be derived from the basic configurations.

An SUT can be a combination of basic configurations 1 and 2, allowing the alternative of using OSI and non-OSI protocols above layer N (see Figure 5/X.290).

7.2.2 *Identification of the IUT*

An IUT is that part of a real open system which is to be studied by conformance testing. It should be an implementation of one or more related OSI protocols in the same layer or in adjacent layers.

IUTs can be defined for configurations 1 and 2 of SUTs as single-protocol IUTs (one single protocol of the SUT is to be tested), or as multi-protocol IUTs (a set of any number of protocol implementations in the SUT to be tested in combination).

An IUT defined in an open relay-system will include at least the layer which provides the relay function.

When OSI and non-OSI protocols exist in a system, the IUT(s) are defined for the OSI mode(s) of operation. Testing non-OSI protocols is outside the scope of Recommendations X.290 to X.294.

The part of the SUT to be considered as the IUT is subject to the agreement of the test laboratory and the client.

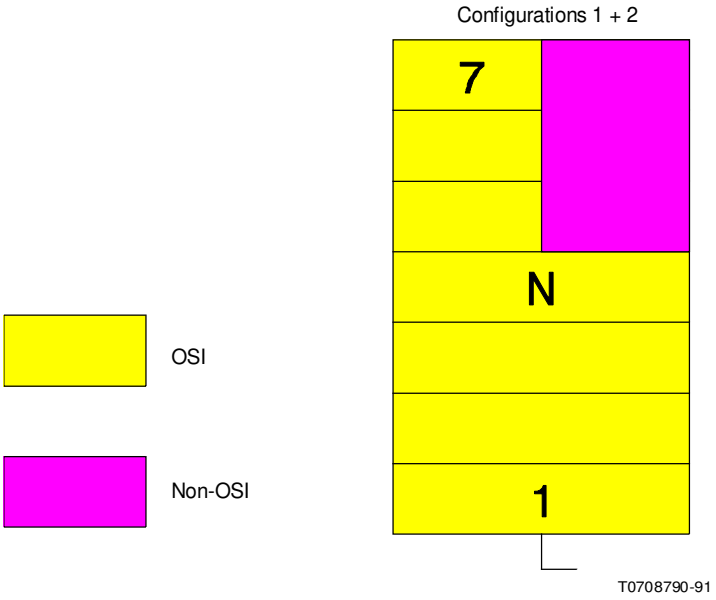

FIGURE 5/X.290

Combination of 7-layer and partial (N)-open systems

7.3 *Abstract testing methodology*

7.3.1 *General*

Test methods need to refer to an abstract testing methodology, based upon the OSI reference model. Considering end-systems (7-layer or partial (N)-open systems) and single protocol IUTs within these systems, Abstract Test Methods are described in terms of what outputs from the IUT are observed and what inputs to it can be controlled. More specifically, an Abstract Test Method is described by identifying the points closest to the IUT at which control and observation are to be exercised.

The CCITT Recommendations and International Standards which specify OSI protocols define allowed behaviour of a protocol entity (i.e. the dynamic conformance requirements) in terms of the PDUs and the ASPs both above and below that entity. Thus the behaviour of an (N)-entity is defined in terms of the (N)-ASPs and (N – 1)-ASPs (the latter including the (N)-PDUs).

If an IUT comprises more than one protocol entity, the required behaviour can be defined in terms of the ASPs above and below the IUT, including the PDUs of the protocols in the IUT.

The starting point for developing Abstract Test Methods is the conceptual testing architecture, illustrated in Figure 6/X.290. It is a "black-box" active testing architecture, based on the definition of behaviour required of the IUT.

7.3.2 *Points of Control and Observation (PCOs)*

7.3.2.1 The action of the conceptual tester shown in Figure 6/X.290 involves two sets of interactions: one above and one below the IUT. These can, in theory, be observed and controlled from several different points.

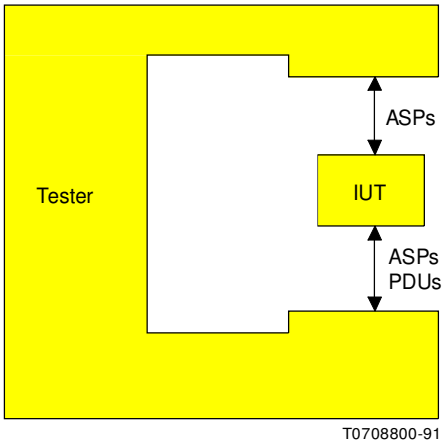

FIGURE 6/X.290

Conceptual testing architecture

7.3.2.2 Each possible PCO is identified by three factors:

a) the service boundary within the OSI model at which the test events are controlled and observed;

b) the set of test events (ASPs or PDUs) that are controlled and observed at this point;

c) whether they are controlled and observed within the SUT or in the test system.

7.3.2.3 Possible PCOs are illustrated in Figure 7/X.290. It can be seen from these Figures that there is theoretically a multiplicity of possible PCOs, which offer different degrees of control and observation of IUT behaviour. Recommendations X.290 to X.294 make a selection from this set of possible PCOs, and defines for use in standardized ATSs a limited number of Abstract Test Methods using these PCOs.

7.3.2.4 If control and observation is specified in terms of ASPs, it will include control and observation of the PDUs carried by those ASPs; but if it is specified solely in terms of PDUs (at layer N) then the underlying ASPs are not considered to be controlled or observed.

7.3.2.5 The PCOs can be modelled as two queues:

a) one output queue for control of test events to be sent towards the IUT; and

b) one input queue for the observation of test events received from the IUT.

7.3.2.6 The ASP activity below the IUT can be observed and controlled by the peer activity in a test system via an underlying service-provider. It is assumed that the underlying service offered is sufficiently reliable for control and observation to take place remotely.

It is possible that the ASP activity above the IUT might not be controllable nor observable, in which case this activity is said to be hidden.

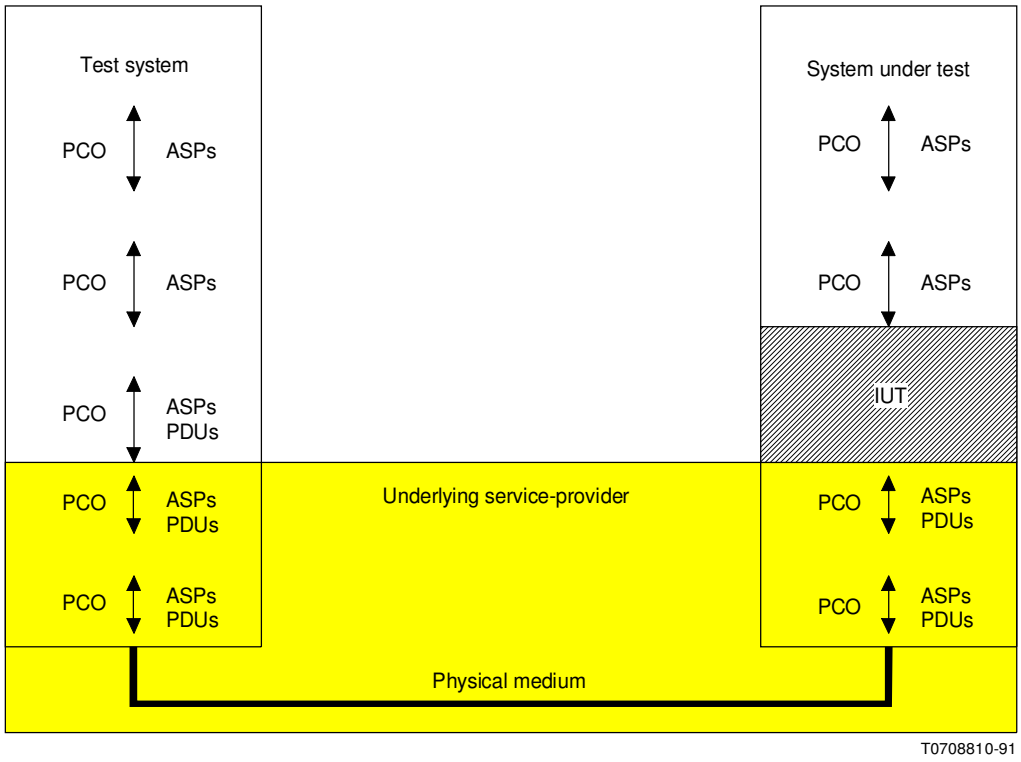

FIGURE 7/X.290

Possible PCOs for testing

7.3.2.7 SUTs are not required to provide access to service boundaries. However, the possible provision of such access and the possible positions of such boundaries with respect to the protocol of the IUT are factors to be taken into consideration in the definition of the Abstract Test Methods, which may take advantage of this access to define the ATSs in terms of the corresponding ASPs. It does not matter whether the accessible boundaries are accessed via Service-Access-Points (SAPs) or via some other PCOs.

7.4 *Abstract testing functions*

The definition of an Abstract Test Method makes use of two abstract testing functions, named the Lower Tester and the Upper Tester.

The Lower Tester is the representation in Recommendations X.290 to X.294 of the means of providing, during test execution, indirect control and observation of the lower service boundary of the IUT via the underlying service-provider. The underlying service-provider is, as illustrated in Figure 7/X.290, beneath the (lowest layer) protocol which is the focus of testing. It may use one or more OSI layers, or the Physical medium only.

The Upper Tester is the representation in Recommendations X.290 to X.294 of the means of providing, during test execution, control and observation of the upper service boundary of the IUT, as defined by the chosen Abstract Test Method. There is a need for cooperation between the Upper Tester and the Lower Tester; the rules for such cooperation are called the test coordination procedures.

The ATSs for different Abstract Test Methods will vary in the way that they specify requirements on the test coordination procedures. In some cases, it is possible to define a Test Management Protocol to provide the coordination between the Upper and Lower Testers. In other cases, it is not possible to specify what mechanisms might be used for the test coordination procedures; it is possible only to describe the requirements to be met by them.

7.5 *Overview of Abstract Test Methods*

7.5.1 *End-system IUTs*

For the IUTs defined within end-system SUTs (configurations 1 and 2 in Figures 2/X.290 and 3/X.290) four categories of Abstract Test Methods are defined: two that use a PCO between the Upper Tester and the IUT (the Local and Distributed test methods) and two that use only the single PCO beneath the Lower Tester (the Coordinated and Remote test methods).

All test methods use control and observation of ASPs below the IUT and PDUs exchanged with the IUT, by means of a Lower Tester separated from the SUT, possibly together with control and observation of ASPs above the IUT.

7.5.2 *The Local and Distributed test methods*

In both the Local and Distributed test methods, there are two PCOs: one beneath the Lower Tester and the other at the upper service boundary of the IUT.

In the Local test method, the Upper Tester is located within the test system; whereas in the Distributed test method, the Upper Tester is located within the SUT.

The Local test method requires the upper service boundary of the IUT to be a standardized hardware interface; the Distributed test method requires it to be either a human user interface or a standardized programming language interface. In both methods access to this interface is required for testing purposes.

In the Local test method, the test coordination procedures are realized entirely within the test system. In both methods the requirements for the test coordination procedures are specified, but the procedures themselves are not.

These test methods are illustrated in diagrams a) and b) of Figure 8/X.290.

7.5.3 *The Coordinated and Remote test methods*

In both the Coordinated and Remote test methods, there is only one PCO: beneath the Lower Tester. Neither method requires access to the upper service boundary of the IUT.

In the Coordinated test method, the test coordination procedures are realized by means of standardized Test Management Protocols (TMPs). The Upper Tester is an implementation of the relevant TMP. This method is illustrated in diagram c) of Figure 8/X.290.

In the Remote test method, some requirements for test coordination procedures may be implied or informally expressed in the ATS, but no assumption is made regarding their feasibility or realization. There is no Upper Tester as such, but some Upper Tester functions may be performed by the SUT. This method is illustrated in diagram d) of Figure 8/X.290, the dotted lines indicating that only the desired effects of the test coordination procedures are described in the ATS.

7.5.4 *Variants of end-system test methods*

Each category of test methods has an embedded variant which can be used for multi-protocol IUTs.

All Abstract Test Methods for end-systems are fully specified in Recommendation X.291 (1992), § 8, including embedded variants where applicable.

7.5.5 *Relay-system IUTs*

For open relay-systems, two test methods are defined, loop-back and transverse. These are fully specified in Recommendation X.291 (1992), § 8.

FIGURE 8/X.290

Overview of abstract test methods

7.6 *Applicability of test methods to real open systems*

The architecture and stage of development of a real open system determines the applicability of test methods to it.

Local test methods are applicable only to testing SUTs that have two hardware interfaces (e.g. transceivers).

Distributed test methods are applicable only to testing IUTs that have an upper interface accessible either to a human user or to a software Upper Tester with a standardized programming language interface.

Coordinated test methods apply where it is possible to implement a standardized TMP in an Upper Tester in the SUT, above the IUT.

Remote test methods apply when it is possible to make use of some functions of the SUT to control the IUT during testing, instead of using a specific Upper Tester.

Single-layer test methods are the most appropriate ones for testing the majority of the protocol conformance requirements.

Embedded test method variants permit the application of single-layer testing to all protocols of a multi-protocol IUT.

For 7-layer open systems, the preferred test methods are the appropriate single-layer embedded test methods, used incrementally, with the following PCOs:

a) the upper interface of the Application layer as provided by the 7-layer open system, when applicable;

b) successively, each SAP (or corresponding PCO if there is no SAP as such) below the protocol which is the focus of the testing, as controlled and observed in the Lower Tester, starting from the lowest protocol of the IUT and working upwards.

7.7 *Applicability of the test methods to OSI protocols and layers*

Annex B provides guidance on the applicability of test methods to particular protocols and layers.

8 Test suites

8.1 *Structure*

Test suites have a hierarchical structure (see Figure 9/X.290) in which an important level is the test case. Each test case normally has a single test purpose, such as that of verifying that the IUT has a certain required capability (e.g. the ability to support certain packet sizes) or exhibit a certain required behaviour (e.g. behave as required when a particular event occurs in a particular state).

Within a test suite, nested test groups are used to provide a logical ordering of the test cases. Test groups may be nested to an arbitrary depth. They may be used to aid planning, development, understanding or execution of the test suite.

Associated with each test group may be a test group objective. When such objectives exist, the complete test group objective for a given test group is formed by the concatenation of the given test group objective with those of any higher level test groups containing the given test group. Likewise, complete test purposes are formed by concatenating the appropriate complete test group objective with the individual test purposes.

Test cases may be modularized by using named subdivisions called test steps.

For practical reasons, common test steps may be grouped together into test step libraries (analogous to subroutine or procedure libraries in programming languages). Test step libraries may be structured into nested sets of test steps, to any depth of nesting. Test step libraries may be associated with the whole test suite or with a particular test group.

Test events are indivisible units of specification within a test step (e.g. the transfer of a single PDU or ASP to or from the IUT). All test steps are specified as an ordering of test events or other (smaller) test steps. All test steps are, therefore, equivalent to an ordering of test events.

8.2 *Abstract and executable test cases*

8.2.1 An abstract test case is derived from a test purpose (or a combination of test purposes, as defined by the test suite specifier) and the relevant OSI CCITT Recommendations or International Standards. It

a) specifies all sequences of foreseen test events necessary in order to achieve the test purpose; these test events comprise the test body;

b) specifies, if the initial testing state required by the test body is not the desired starting stable state of the test case, at least one sequence of test events to put the IUT into the initial testing state for the test body; these test events comprise the test preamble;

c) specifies, if the test body can end without the IUT being returned to the desired stable testing state, at least one sequence of test events to return the IUT to the desired stable testing state; these test events comprise the test postamble;

d) uses a single Abstract Test Method in the specification of all sequences of test events;

e) uses a standardized test notation for the specification of all sequences of test events;

f) may be constructed from test steps, each of which is a set of sequences of test events;

g) specifies the verdict to be assigned to each possible sequence of test events comprising a complete path through the test case.

8.2.2 The test preamble and postamble may be specified in different ways depending on the degree of control and observation provided by the test method used, or on the variety of different possible stable testing states from which the derived abstract test case can start and in which it can end. These abstract test cases are simply different ways of achieving the same test purpose.

8.2.3 An executable test case is derived from an abstract test case, and is in a form which allows it to be executed by the test system, in combination with the SUT.

8.2.4 The terms abstract and executable are used to describe test suites which comprise abstract and executable test cases respectively.

9 Relationships between Recommendations in the X.290-Series, concepts and roles

Figure 10/X.290 is a pictorial representation of the relationship between the various Recommendations in the X.290-Series and the processes of producing abstract and executable test suites and test reports.

Recommendation X.291 concerns the production of testable protocol specifications and conformance testing standards. Recommendation X.292 concerns a standardized test notation for ATS specification. Recommendation X.293 concerns the realization of the Means of Testing. Recommendation X.294 concerns the roles of a test laboratory and its client in the conformance assessment process, which culminates in test report production. Recommendation X.290 provides general concepts and definitions which underpin the other parts.

10 Compliance

In Recommendations X.290 to X.294, compliance refers to meeting the requirements specified in one or more of the parts. This word is used in an attempt to eliminate confusion between compliance to one or more of Recommendations X.290 to X.294 and conformance of a protocol implementation to CCITT Recommendations or International Standards which specify protocols.

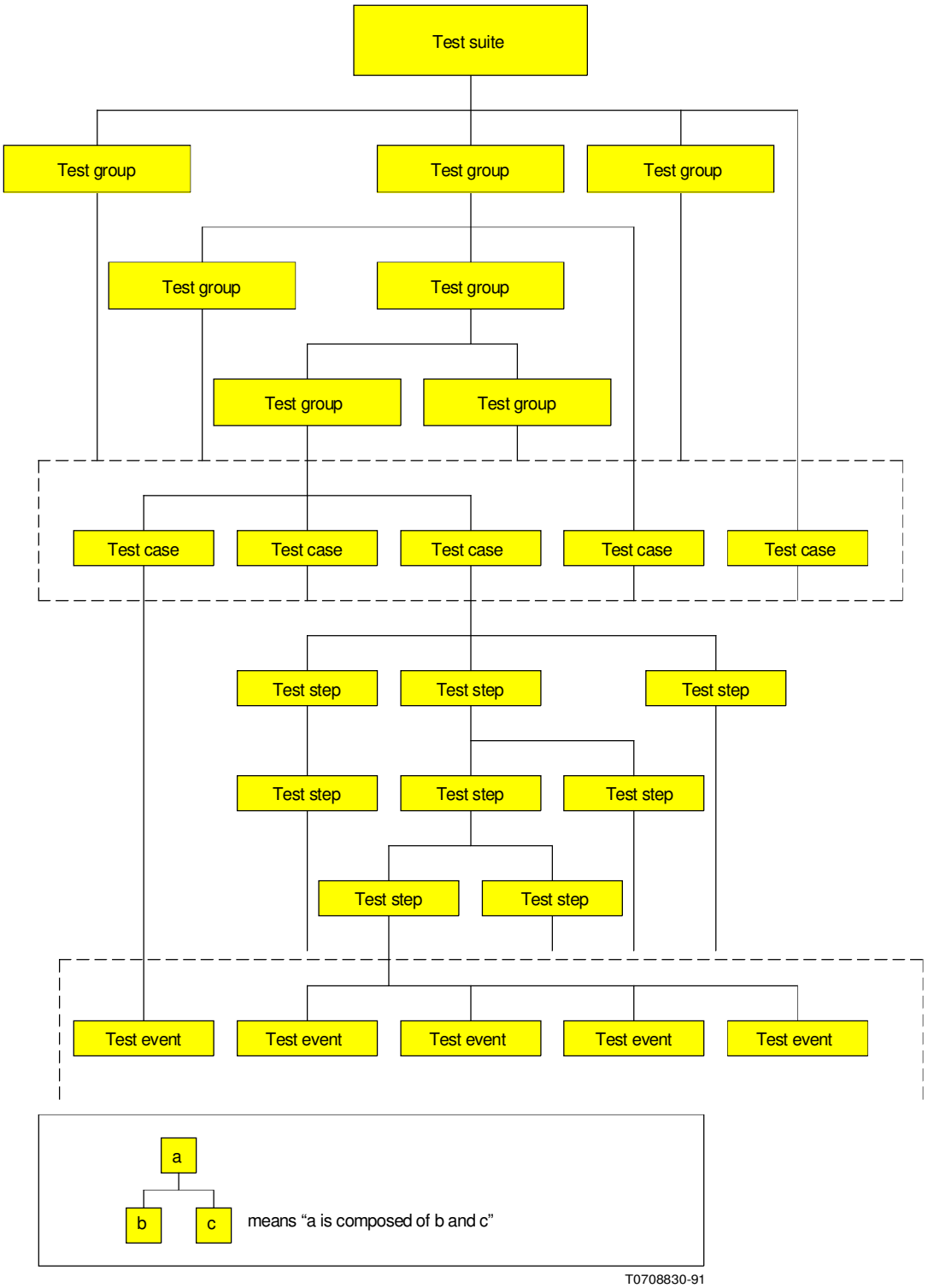

FIGURE 9/X.290

Test suite structure

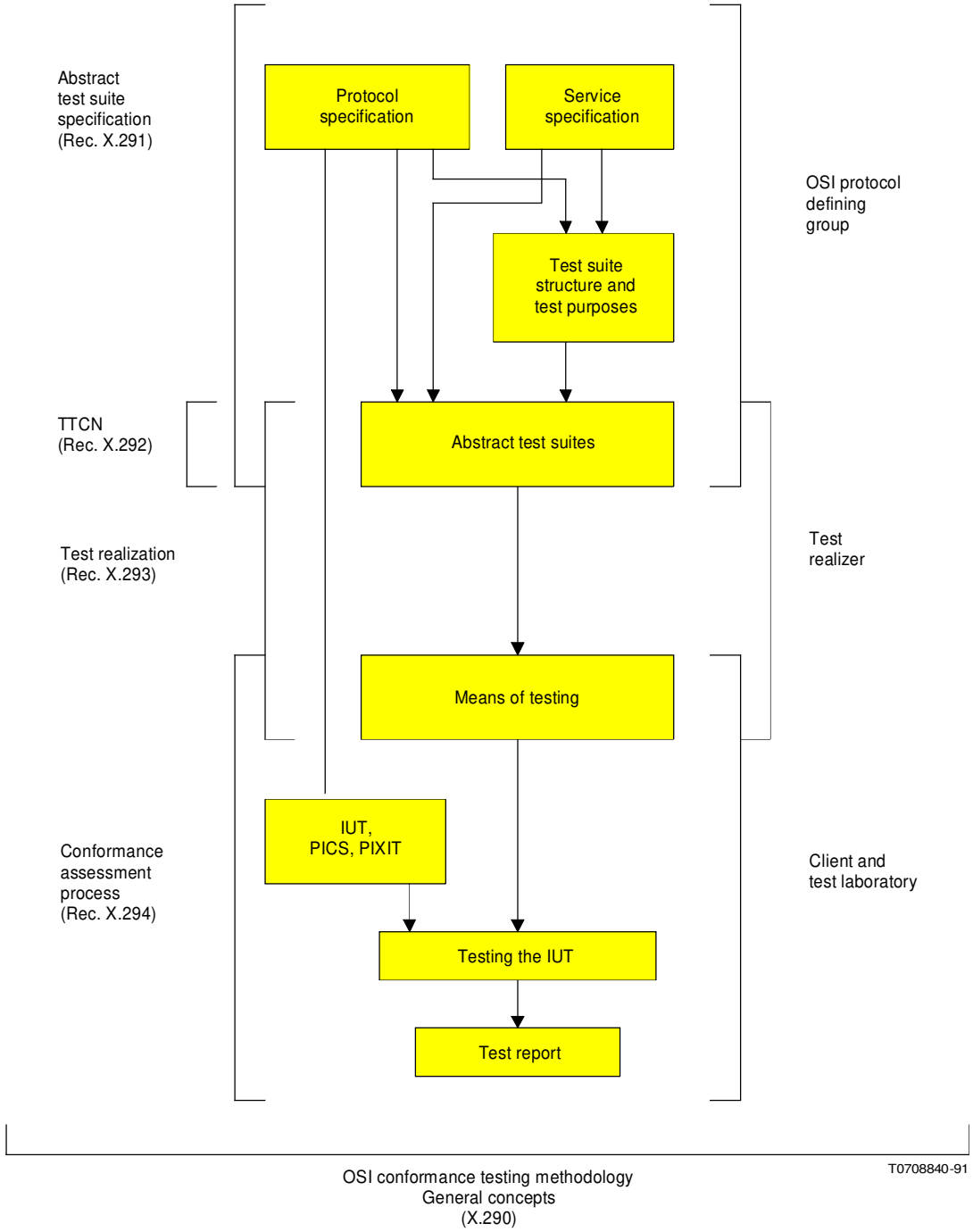

FIGURE 10/X.290

Relationships between Recommendations, concepts and activities

Recommendations X.291 to X.294 contain compliance requirements, appropriate to the various activities addressed. These, if met, will achieve the objectives of conformance testing, as described in the Introduction of Recommendation X.290.

This Recommendation contains no compliance requirements.

ANNEX A

(This annex does not form an integral part of this Recommendation)

Options

A.1 Options are those items in an CCITT Recommendation or International Standard for which the implementor may make a choice regarding the item to suit the implementation.

A.2 Such a choice is not truly free. There are requirements which specify the conditions under which the option applies and the limitations of the choice.

Conversely, there may be mandatory or conditional requirements, or prohibitions, in an CCITT Recommendation or International Standard which are dependent on the choice made or on a combination of the choices already made.

A.3 The following are examples of options and associated requirements; the list is not exhaustive:

a) "*Boolean" options:* the option is "do or do not do"; the requirement is "if do, then do as specified".

b) *Mutually exclusive options:* the requirement is to do just one of n actions, the option is which one of them to do. These options could also be considered to be alternative mandatory features.

c) *Selectable options:* the option is to do any m out of n actions, with a requirement to do at least one action ($1 \leq m \leq n$ and $n \geq 2$).

A.4 Options may apply to anything within the scope of an CCITT Recommendation or International Standard (e.g. static or dynamic aspects, use or provision of a service, actions to be taken, presence/absence or form of parameters, etc.).

A.5 In a wider context, the choice may be determined by conditions which lie outside the scope of the CCITT Recommendation or International Standard (e.g. other CCITT Recommendations or International Standards which apply to the implementation, the protocols used in the (N – 1) and (N + 1) layers, the intended application, conditions of procurement, target price for the implementation, etc.). However, these have no bearing on conformance to the CCITT Recommendation or International Standard in which the option appears.

ANNEX B

(This annex does not form an integral part of this Recommendation)

Applicability of the test methods to OSI protocols

B.1 *The Physical layer*

In the Physical layer, test events include the act of measuring some characteristic of or generating a Physical signal (e.g. an electrical or optical signal). Nevertheless, Recommendations X.290 to X.294 do not fully address the requirements of the Physical layer (e.g. no standardized test notation is provided for the Physical layer).

For the Physical layer functions of Physical components, such as Modems and Transceivers, the Local test method is directly applicable.

The Remote and Coordinated embedded test methods are likely to be the most practical for local area networks (LAN).

In some cases for local area networks, sufficient control and observation above the IUT can be provided by the normal activity of a Data Link protocol. In such cases, the Data Link protocol implementation in the SUT provides the Upper Tester functionality and employs the Data Link protocol for test coordination. This is an example of the Remote Embedded test method. If, however, there are no protocols used above the Data Link protocol, then this can be considered to be an example of the Coordinated test method.

B.2 *Data Link and Media Access Control protocols*

For testing Data Link protocols, the following points should be considered:

a) the Local single-layer test method is applicable only if the IUT has a standardized hardware upper interface;

b) the test methods are applicable only if a Lower Tester can be realized with control over Physical Service primitives (or perhaps more realistically Physical and Data Link PDUs). This may be difficult for some types of subnetwork.

For Media Access Control protocol testing:

c) sufficient control and observation above the IUT may be provided by the normal activity of the Logical Link Control protocol. In such cases, the Logical Link Control protocol implementation in the SUT provides the Upper Tester functionality and employs the Logical Link Control protocol for test coordination. This is an example of the Remote embedded test method. If, however, there are no protocols used above the Logical Link Control protocol, then this can be considered to be an example of the Coordinated test method.

If single-layer testing of a Data Link protocol is not possible, single-layer embedded test methods should be considered.

B.3 *Network protocols*

For Network protocols, the test methods to be used are dependent upon whether the IUT is an end-system or an open relay system.

It should be recognized that with some subnetwork technologies there are more than three protocols required to provide the Network Service. Each of these protocols may be tested separately or in any combination of adjacent protocols.

Considering the layer as a whole, both Network and Data Link ASPs are controllable and observable. Thus, for end-systems, all four Single-layer (non-embedded) test methods are applicable, but since the Data Link Service is not end-to-end, the Lower Tester has to be connected to the SUT over a single link.

Both the Loop-back and Transverse test methods are applicable to testing Network relay systems.

B.4 *Transport protocol*

The Coordinated, Remote and Distributed embedded test methods are applicable to Transport protocol conformance testing.

B.5 *Session protocol*

The Coordinated, Remote and Distributed embedded test methods are applicable to Session protocol conformance testing.

For a large group of systems it will be appropriate to test the Session protocol in combination with the Presentation protocol and appropriate ASEs. Testing of the Session protocol should, therefore, be done in one of the two following ways:

a) as a single-layer implementation, in order to test the provision of a general purpose Session Service capable of supporting several different ASEs; the Coordinated Single-layer test method is likely to be appropriate;

b) in combination with Presentation protocol and ASEs, in order to test it in a specific Application context; the Remote or Distributed Single-layer Embedded test methods are likely to be appropriate.

B.6 *Presentation and Application protocols*

B.6.1 *General comments*

The Presentation protocol and protocols for ASEs in a specific Application Context are interrelated to a large extent. Invalid Application PDUs will (e.g. in the case of syntactic errors) have to be detected by the Presentation layer, and, in the case of semantic errors, by the relevant ASE. Real systems may choose to combine these functions.

It is therefore, in general, not feasible to test Presentation and Application protocols separate from each other.

B.6.2 *Presentation*

The service primitives are potentially observable and controllable to the same extent as for lower layers. Thus, all four Single-layer (non-embedded) test methods are theoretically applicable. However, the testing of Presentation protocol in isolation from an ASE is of limited value, because it could only test the protocol machine, leaving untested the more interesting aspect of the Presentation layer, namely the mapping between abstract and transfer syntaxes. Therefore, the testing of Presentation protocol embedded under Association Control and other ASEs in a specific Application Context is preferred. Thus, the relevant applicable test methods are the Remote and Distributed Single-layer Embedded ones.

B.6.3 *Application*

Conformance tests can be specified abstractly in terms of service primitives, whether or not there is any notion of a SAP associated with them. Thus, provided that there is some mapping between ASE primitives and effects which can be observed and/or controlled, tests can be specified in terms of those ASE primitives. The observation and control of the service primitives may be indirect because of the nature of the mapping onto corresponding effects, but as long as that mapping is possible then tests specified in these terms can be run.

It is accepted that, in some circumstances, CCITT Recommendations or International Standards for Applications, defining Application Contexts, may specify non-protocol conformance requirements which have to be achieved as a result of protocol exchanges. However, these requirements should be kept quite distinct from the normal protocol conformance requirements, possibly even in separate CCITT Recommendations or International Standards. Testing of non-protocol conformance requirements will in general require application-specific test methods, and so fall outside the scope of Recommendations X.290 to X.294.

When testing specific ASEs in an Application Context that includes ACSE, the PCO below the Lower Tester will be characterized by the set of possible ASPs that can occur at it. These will include both ACSE and Presentation ASPs.

B.6.4 *Transfer syntaxes*

Transfer syntaxes (e.g. ASN.1 or X.209) are rather different from the OSI protocol specifications with respect to conformance. In general, there would not be conformance testing of the encoding rules of a transfer syntax independent of the ASE using those rules. In any case, the transfer syntax encoding rules will be tested with the Presentation protocol, and the test methods appropriate to that protocol will be used.

B.7 *Connectionless protocols*

Since each test method described in Recommendations X.290 and X.291 is defined in terms of observation and control of ASPs and PDUs, and not in terms of connections, they are all applicable to the testing of connectionless protocols, taking into account the restrictions applying to each layer.

ANNEX C

(This annex does not form an integral part of this Recommendation)

Index of Recommendations X.290, X.291, X.293 and X.294

C.1 *Introduction*

This annex presents an alphabetical index of terms and acronyms with references to their uses in Recommendations X.290, X.291, X.293 and X.294. The references are in terms of clause, figure and table numbers, grouped by number, with each group prefixed by a number in square brackets. The significance of each reference is indicated as follows:

a) definitions of the terms and acronyms are in bold;

b) requirements clauses concerning the term or acronym are underlined;

c) major uses of the term or acronym are in *italics*;

d) major requirements clauses are in *underlined italics*;

e) other uses are in normal font.

Note – In the index, [1] corresponds to Recommendation X.290, [2] corresponds to Recommendation X.291, [4] corresponds to Recommendation X.293, [5] corresponds to Recommendation X.294.

C.2 *Index*

(N)-entity:
[1] **3** in **1**

(N)-layer:
[1] **3** in **1**

(N)-Protocole:
[1] **3** in **1**

(N)-relay:
[1] **3** in **1**

(N)-service:
[1] **3** in **1**

Abnormal test case termination:
[1] 3.7.5, **3.7.20**, 6.5.1.4, 6.5.1.6
[5] 7.6.2.3

Abstract service primitive:
[1] **3.8.4**, 7.3, Figure 6, Figure 7, Figure 8
[2] 1.2, 12.1, 12.3, Figure 1, Figure 2, Figure 3, Figure 4, 12.4, Figure 5, Figure 6, Figure 7, Figure 8, 12.5, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, 13.1
[4] 6.4, A.1.2.1
[5] 6.2.1.2.1, 6.2, 1.3, C.2(5)

Abstract test case error:
[1] **3.7.18**
[5] 7.6.2.3

Abstract test case:
[1] **3.6.3**, *8.2*
[2] 8.2, 10.3.2, 10.3.3, 10.4, 10.5, 11, 12, 12.5.1, *13.2,* 15, *D.3,* D.4
[4] 5.3, 6.1, 6.3.3, 6.3.4, 6.4, 6.6, A.1.2.1, A.4.3
[5] 7.3.1, 7.6.2.3, 8.3.1, B.2(6), C.2(5), C.2(7.2.4)

Abstract test method:
[1] **3.6.1**, 6.3.2, *7.5*, Figure 8, *Annex B*
[2] 1.2, 10.5, *12,* 13.2.3
[4] 6.2.1, 6.2.5, A.1.2.1, A.1.3, A.2.2
[5] Figure 1, Figure 2, 6.2.1.1, 6.3.2.3, A.2(1.7), A.2(2.n), B.2(1.3), C.1, C.2(2)

Abstract test suite specification:
[2] 5.2, *8, 13, 15*
[4] 5.1, 5.5, 6.2.1, A.4.5

Abstract test suite specifier:
[2] 8, 9.1, 9.3, 10.1.3, 10.2, *10.3,* 10.4, 12.7.2, 12.7.3, 13.2, 13.4, *15,* 16, A.1.2, Annex C, D.4.2

Abstract test suite:
[1] 3.6.23, 8.2, 9, Figure 10
[2] 5.3, 6.1, 8, 9, 10, 10.1.2, 11, 12, 13, 15, 16
[5] 1, Figure 1, Figure 2, 7.6.2.3, A.2(1.7), C.1, C.2(2)

Abstract testing methodology:
[1] **3.6.2**, *7.3*

Accreditation:
[5] 1, 6.3.1.2

Analysis of results:
[1] Figure 1, *6.5*
[2] A.1.3

Application context:
[1] B.5, B.6.1, B.6.2, B.6.3

Application layer:
[1] **3.1**, *B.6*

Application service element:
[1] **3.1**, B.5, B.6

ASE:
[1] **4**, B.5, B.6

ASN.1:
[1] 2, 3.2, **4**, B.6.4
[2] 2, A.8.6

ASP:
[1] **3.8.4, 4**, 7.3, Figure 6, Figure 7, Figure 8
[2] 1.2, 12.1, 12.3, Figure 1, Figure 2, Figure 3, Figure 4, 12.4, Figure 5, Figure 6, Figure 7, Figure 8, 12.5, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, 13.1
[4] **4**, 6.4, A.1.2.1
[5] **4.1**, 6.2.1.2.1, 6.2.1.3, C.2(5)

ATM:
[1] **3.6.1**, **4**, 6.3.2, *7.5*, Figure 8, *Annex B*
[2] 1.2, 10.5, *12*, 13.2.3
[4] 6.2.1, 6.2.5, A.1.2.1, A.1.3, A.2.2
[5] **4.1**, Figure 1, Figure 2, 6.2.1.1, 6.3.2.3, A.2(1.7), A.2(2.n), B.2(1.3), C.1, C.2(2)

ATS conformance clause:
[2] *13.3*

ATS:
[1] **3.6.23**, **4**, *8.2*, Figure 10
[2] 5.3, 6.1, *8*, 9, *10*, *10.1.2*, 11, *12*, *13*, *15*, *16*
[4] **4**
[5] 1, **4.1**, Figure 1, Figure 2, 7.6.2.3, A.2(1.7), C.1, C.2(2)

Auditability of results:
[1] *6.5.5*

Basic interconnection tests:
[1] **3.5.2**, 6.1.1, *6.1.2*, Figure 1, 6.4
[2] 10.2.3, 13.1, 15.2
[4] 6.2.3, A.4.5
[5] 5.3, Figure 3, 7.3.1, 7.3.2, 7.4.1.2, *7.6.2.2*, *7.6.3.2*, 8.3.1

Basic reference model:
[1] 2, 3.1, 7.3.1
[2] 2
[5] 2

Behaviour tests:
[1] **3.5.4**, 6.1.1, *6.1.4*, Figure 1
[2] 10.1.3, 10.2, 13.1, 15.2
[4] 6.2.3, A.4.5
[5] 5.3, 7.3.1, 7.6.2.2

BIT:
[1] **3.5.2**, **4**, 6.1.1, *6.1.2*, Figure 1, 6.4
[2] 10.2.3, 13.1, 15.2
[4] **4**, 6.2.3, A.4.5
[5] **4.1**, 5.3, Figure 3, 7.3.1, 7.3.2, 7.4.1.2, *7.6.2.2*, *7.6.3.2*, 8.3.1

Capability of an implementation:
[1] **3.4.5**, *6.1.3*
[2] A.2.1, *A.8.3*

Capability tests:
[1] **3.5.3**, *6.1.3*, Figure 1, 6.4
[2] 10.1.3, 10.2, 10.4, 15.2
[4] A.4.5
[5] 5.3, Figure 3, 7.3.1, 7.4.1.2, 7.6.2.2

Certification:
[1] 1.7
[5] 1

Client (of a test laboratory):
[1] **3.4.12**, 6.3.4, 9, Figure 10
[2] A.3.7, A.5
[4] 6.5, 6.6, A.4.1, *A.4.4*
[5] 1, 3.6, 3.7, 5.2, Figure 2, *6.2.1*, 6.3.1.3, 6.3.2.2, 6.4.1, *6.4.2.3*, *6.4.3.3*, 6.4.4.1, 6.4.4.3, 6.4.5.3, 7.2.1.2, 7.2.2, 7.3.2, 7.4.2, 7.6.2.2, 7.6.2.3, 7.6.2.4, *7.6.3*, 7.7.2, 7.7.3, 8.2.1, 8.2.2, 8.2.3, 8.3.2, 9.2, A.1, A.2(1.3), A.2(1.6), A.2(1.8), B.2(1.4), B.2(1.5), C.1, C.2(1), C.2(4), C.2(5), C.2(6), C.2(7.2.2)

Client and test laboratory:
[5] 3.5, 5.3, 6.1, *6.3*, *6.3.1.1*, 6.3.2.3, 6.3.3, 6.4.1, 7.2.3, *7.5.2*, 7.7.1, 7.7.4, 8.3.3, Annex D, E.2

Client checklist:
[5] **3.7**, Figure 2, 6.2.1.3.2, 6.2.1.5.2, *6.3.1*, 6.3.2.1

Client test manager:
[5] **3.1**, 7.6.4.1, C.2(4)

Comparability of results:
[1] 1.5, **3.7.2**, *6.5.4*
[5] 1

Compliance:
[1] 1.6, *10*
[2] *5*
[4] 6.2.1, 6.2.2, *7*
[5] 1, 6.3.1.2, 6.3.1.3, *9*

Comprehensive testing service:
[1] **3.8.16**
[2] *12.7.2*
[4] 6.2.5
[5] 6.3.1.2, 6.3.2.2, 9.1

Conformance assessment process:
[1] 1.3, **3.5.7**, 6.1.4.2, 6.3, Figure 1, 6.4, 6.5.2, 9, Figure 10
[2] 15.2, A.1.3
[4] 5.3, 6.4
[5] 1, 3.5, 3.6, *5*, Figure 1, 6.1, 6.3.1.2, 6.4.3.2, 7.2.1.2, 7.7, 8.1.1, 8.2.1, 8.3.1, A.1

Conformance clause:
[2] *6.3*, *A.3.6*

Conformance log:
[1] **3.7.21**, 6.3.3, 6.5.2, 6.5.5
[4] 1, 5.2, 5.3, 6.1, 6.2.6, *6.4*, 6.6, 7, *A.3*, Figure A.2, A.4.5
[5] 1, 7.6.1, 7.6.2.1, 7.6.2.1, 7.6.2.3, *7.6.2.4*, *7.7.2*, 7.7.3, 8.3.1, 8.3.2, B.2(1.3), E.2

Conformance resolution tests:
[1] **3.5.5**, 6.1.1, *6.1.5*
[2] 11, 13.2.3

Conformance test suite:
[1] **3.6.19**
[2] 1.1, 5.2

Conformance testing standard:
[1] **3.6.32**, 6.5.2, 9
[2] 1.1, 1.2, *8*, 8.3, 10.1.1, 11, 12.7.2, 14
[4] 5.1, 6.3.2
[5] 1, 5.2, 6.3.1.2, 6.3.2.1, 6.3.3, 6.4.5.1, 9.1

Conformance testing:
[1] **3.5.6**
[2] 5, Annex B
[5] 1, 3.1, 3.4, 6.3.1.2, 7.7.4, 8.2.1, 8.3.1, 9.1, A.2(1.5), B.1, Annex E

Conforming implementation:
[1] **3.4.10**
[2] C.4

Coordinated embedded test method:
[1] B.1
[2] *12.5.4*, *Figure 11*, 14

Coordinated test method:
[1] **3.8.10**, *7.5.3*, Figure 8, 7.6, Annex B. B.1, B.2, B.4, B.5
[2] *Figure 3*, *12.3.4*, *12.4.4*, *Figure 7*, 12.5.4, 14
[4] 6.2.1
[5] *6.2.1.4*, *6.4.5*, 7.5.2

Copyright:
[2] *A.4*
[5] Annex A, Annex B, Annex C

Coverage:
[2] 8.2, 10.1.3, 10.2.1, 10.3.1, 10.3.5, *10.4*, 13.1

CS:
[2] **4**, *12.4.4*, *Figure 7*, 12.5.4, 14

CSE:
[2] **4**, *12.5.4*, *Figure 11*, 14

Data link layer:
[1] **3.1**, *B.2*

Defect report:
[2] 13.2.2, 13.4, 16
[4] 6.6

Derivation:
[4] 5.2, 5.3, 5.4, 6.1, 6.2.4, *6.3*, 7, *A.2*, Figure A.1

Diagnostic trace:
[5] 1, 7.7.2, 8.1.1

Distributed embedded test method:
[2] *12.5.3*, *Figure 10*, 12.7.2

Distributed test method:
[1] **3.8.9**, *7.5.2*, Figure 8, 7.6, Annex B, B.4, B.5
[2] 12.3.3, *Figure 2*, *12.4.3*, *Figure 6*, D.2
[4] 6.2.1
[5] *6.2.1.3*

DS:
[2] **4**, 12.3.6, *12.4.3*, *Figure 6*, D.2

DSE:
[2] **4**, 12.3.6, *12.5.3*, *Figure 10*, 12.5.4, 12.7.2

DTE:
[1] **4**, 5.2.1

Dynamic conformance requirements:
[1] **3.4.3**, 5.2.3, *5.4*, 5.6, 6.1.4.1
[2] 1.2, 6.2.1, 6.3, A.8.3, B.4.2

Dynamic conformance summary:
[5] *B.2(4)*

Embedded test methods:
[1] 7.5.4, 7.6, Annex B, B.2
[2] *12.5*

Embedded testing:
[1] **3.5.9**
[2] 9, *12.3.6*, *12.5*
[5] 6.3.2.2, 6.4.3.2

Encoding:
[1] **3.2**, *B.6.4*
[2] 10.2.1, 10.2.2, 10.3.1, 10.4, A.8.6, B.5.1, B.5.8, B.5.10
[5] 7.4.1.1

End-systems:
[1] 7.2.1.1, *7.5.1*, 7.5.4
[2] 1.3, *12.3*

ETS:
[1] **3.6.24**, 3.6.26, **4**
[2] A.1.3
[4] **4**, 5.3, 5.5, *6.3*, A.2.1, Figure A.1, *A.2.2*, *A.2.3*, A.4.5
[5] 7.6.1, C.2(7.2)

Executable test case error:
[1] **3.7.19**
[5] 7.6.2.3

Executable test case:
[1] 3.6.3, **3.6.4**, 8.2.3
[4] 5.3, 6.1, 6.2.2, *6.3*, A.1.2.1, *A.2*, A.3, A.4.3
[5] 8.3.1

Executable test suite:
[1] **3.6.24**, 3.6.26, 9
[2] A.1.3
[4] 5.3, 5.5, *6.3*, A.2.1, Figure A.1, *A.2.2*, *A.2.3*, A.4.5
[5] 7.6.1, C.2(7.2)

Fail verdict:
[1] 3.7.6, **3.7.15**, *6.5.1.3*
[2] 13.2.7
[5] 7.6.2.3, 7.6.3.3, 7.6.4.2, 8.3.1, 8.3.1, A.2(2.n), B.2(2), B.2(4)

FDT:
[2] 1.3, **4**, 10.3.1, 13.4, B.3.1, *B.7*

Finite states:
[2] B.6

Foreseen test outcome:
[1] **3.7.4**, *6.5.1.2*

Formal description techniques:
[2] 1.3, 10.3.1, 13.4, B.3.1, *B.7*

Formal methods in conformance testing:
[2] 10.4

Functional standard:
[1] 5.1

Generic test case:
[1] **3.6.7**
[2] 8.2, 13.2.3, *Annex D*

Generic test suite:
[1] **3.6.22**
[2] 8.3, 10.5, *11*

Idle testing state:
[1] **3.6.16**
[2] 13.2.3, *13.2.5*

Implementation under test:
[1] **3.4.1**, *7.2.2*, 7.3.1, Figure 6, Figure 7, Figure 8, Figure 10
[2] 10.2.1, 10.2.2, 12.3, Figure 1, Figure 2, Figure 3, Figure 4, 12.4, Figure 5, Figure 6, Figure 7, Figure 8, 12.5, Figure 9, Figure 10, Figure 11, Figure 12, 12.7.3
[4] 5.3, 6.2.2, 6.3.3, 6.3.4, 6.3.5, 6.4, 6.5, A.1.2.1, A.1.2.2, A.1.3, *A.1.4*, A.2.1, A.4.2, A.4.3
[5] 1.3.3, Figura 1, 5.4, 6.2.1.3.1, 6.2.1.4.1, 6.3.1.1, 6.3.1.3, 6.3.2.1, 6.3.2.2, 6.3.2.3, 6.4.1, 6.4.2.3, 6.4.3.2, 7.2.1.1, 7.3.1, 7.6.1, 7.6.3.1, 7.6.3.3, 7.6.4.1, 7.6.4.2, 8.2.1, 8.3.1, 9.1, A.2(1.5), A.2(1.7) , A.2(2), A.2(2.n), *B.2*, C.1, C.2(5), C.2(6), *C.2(7.2)*, Annex D

Inconclusive verdict:
[1] 3.7.6, **3.7.16**, *6.5.1.3*
[2] 13.2.7
[5] 7.6.2.3, 7.6.4.2, A.2(2.n)

Incremental testing:
[1] 7.6
[5] 6.3.2.2, 7.1

Initial testing state:
[1] **3.6.18**
[2] *13.2.5*, D.2, D.4.2

Inopportune test event:
[1] **3.7.11**
[2] 10.2.1

Interworking:
[1] 5.3, *5.7*
[5] 8.2.1, A.2(1.5)

Invalid test event:
[1] **3.7.10**, 3.7.15, 6.1.4.5, 6.5.1.3, B.6.1
[2] 10.2.1, 10.2.2, 10.3.2, 10.4, B.5.3, B.5.4, B.5.6, B.5.7, B.6.4
[4] A.1.2.2

ISDN:
[1] 1.2, **4**

IUT:
[1] **3.4.1**, **4**, *7.2.2*, 7.3.1, Figure 6, Figure 7, Figure 8, Figure 10
[2] 10.2.1, 10.2.2, 12.3, Figure 1, Figure 2, Figure 3, Figure 4, 12.4, Figure 5, Figure 6, Figure 7, Figure 8, 12.5, Figure 9, Figure 10, Figure 11, Figure 12, 12.7.3
[4] **4**, 5.3, 6.2.2, 6.3.3, 6.3.4, 6.3.5, 6.4, 6.5, A.1.2.1, A.1.2.2, A.1.3, *A.1.4*, A.2.1, A.4.2, A.4.3
[5] 1, 3.3, **4.1**, Figure 1, 5.4, 6.2.1.3.1, 6.2.1.4.1, 6.3.1.1, 6.3.1.3, 6.3.2.1, 6.3.2.2, 6.3.2.3, 6.4.1, 6.4.2.3, 6.4.3.2, 7.2.1.1, 7.3.1, 7.6.1, 7.6.3.1, 7.6.3.3, 7.6.4.1, 7.6.4.2, 8.2.1, 8.3.1, 9.1, A.2(1.5), A.2(1.7), A.2(2), A.2(2.n), *B.2*, C.1, C.2(5), C.2(6), *C.2(7.2)*, Annex D

Local area networks:
[1] B.1

Local embedded test method:
[2] *12.5.2*, *Figure 9*

Local test method:
[1] **3.8.8**, *7.5.2*, 7.6, Annex B, B.1, B.2
[2] *12.3.2*, *Figure 1*, *12.4.2*, *Figure 5*, 12.7.2
[5] *6.2.1.2*, 7.5.2

Loop-back test method:
[1] 7.5.5, B.3
[2] 2.6.1, *12.6.2*, *Figure 13*

Lower tester:
[1] **3.8.2**, 3.8.7, 6.1.4.5, *7.4*
[2] 8.2, *12*, *12.2.1*, 14
[4] 6.2.1, 6.4, A.1.1, *A.1.2*, *A.1.4*, A.4.2, *A.4.3*, A.4.4, A.4.5
[5] 6.3.1.2, 7.6.4.1, C.2(3), C.2(7.2.1), Annex D

LS:
[2] **4**, *12.4.2*, *Figure 5*, 12.4.3

LSE:
[2] **4**, *12.5.2*, *Figure 9*

LT:
[1] **3.8.2**, 3.8.7, **4**, 6.1.4.5, *7.4*
[2] 8.2, *12*, *12.2.1*, 14
[4] 6.2.1, 6.4, A.1.1, *A.1.2*, *A.1.4*, A.4.2, *A.4.3*, A.4.4, A.4.5
[5] 6.3.1.2, 7.6.4.1, C.2(3), C.2(7.2.1), Annex D

Mandatory capability:
[2] 10.2.1, *A.2.1*, A.2.3, A.7, A.9.1, A.9.2.1
[5] 7.2.1.2, 7.3.1

Mandatory requirements:
[2] B.4.2, B.4.5

Means of testing IUTs:
[1] 3.6.1, **3.8.12**, 3.8.13, 3.8.14, 6.3.2, 6.5.4, 9, Figure 10
[2] 15.1
[4] 1, 5.1, *5.2*, 5.3, *5.5*, 6.1, *6.2*, 6.6, 7, *A.1*, A.2.3, A.4.1, A.4.2, *A.4.3*
[5] 1, Figure 1, Figure 2, 7.4.1.1, 7.6.2.1, A.2(2.n), B.2(1.3), C.1, C.2(3), C.2(6)

Media access control protocols:
[1] B.2

MOT:
[1] 3.6.1, **3.8.12**, 3.8.13, 3.8.14, **4**, 6.3.2, 6.5.4, 9, Figure 10
[2] 15.1
[4] 1, **4**, 5.1, *5.2*, 5.3, *5.5*, 6.1, *6.2*, 6.6, 7, *A.1*, A.2.3, A.4.1, A.4.2, *A.4.3*
[5] 1, **4.1**, Figure 2, Figure 3, 7.4.1.1, 7.6.2.1, A.2(2.n), B.2(1.3), C.1, C.2(3), C.2(6)

Multi-layer dependencies:
[1] 5.3, 5.5
[2] 6.3.2, 7.3, A.8.9

Multi-protocol IUT:
[1] 7.5.4, 7.6
[2] *12.5*
[5] 6.3.2.2, 7.1, C.2(8)

NE:
[5] **3.5**, **4.2**, 5.2, 5.3, 6.1, 6.3.2.3, 7.2.3, 7.6.2.2, 7.6.3.2, 7.6.4.1, *7.7*, 8.2.1

Negociated exit:
[5] **3.5**, 5.2, 5.3, 6.1, 6.3.2.3, 7.2.3, 7.6.2.2, 7.6.3.2, 7.6.4.1, *7.7*, 8.2.1

Negotiation:
[2] *A.8.7*

Network layer:
[1] **3.1**, *B.3*

Non-OSI protocols:
[1] 7.2

Options:
[1] 5.2.1, 5.5, *Annex A*
[2] 7.3, 8.2, A.1, A.2, A.3, A.9.1, *B.2.7*, *B.4*

Parameterization:
[1] 3.8.12, 6.3.3, Figure 1
[2] 8.2
[4] 5.2, 5.3, 6.2.4, 6.3.5, 7, A.2.1, Figure A.1, A.4.3, A.4.5
[5] Figure 1, 5.3, Figure 3, *7.4*, 7.5.1, 9.1

Parameterized abstract test case:
[1] **3.6.27**
[4] A.3

Parameterized abstract test suite:
[1] **3.6.29**
[4] 5.4, Figure A.1

Parameterized executable test case:
[1] **3.6.28**, 3.7.3

Parameterized executable test suite:
[1] 3.5.8, **3.6.30**, 6.3.3
[4] 5.4, 5.5, *6.3*, 6.4, Figure A.1, A.2.3, A.3, Figure A.2, A.4.5
[5] 5.3, Figure 3, 7.4.1.1, 7.4.1.2, 7.6.2.1

Parameters:
[1] 3.6.27, 3.6.28
[2] 7.3, 10.2.1, *10.3*, 10.4, 15.1, A.1.1, A.8.1, A.8.4, *A.8.6*, B.5.9, B.5.10
[4] 6.3.3
[5] 6.4.3.2, *7.4*, B.2(1.3), C.2(6), C.2(7.2), *C.2(7.2.2)*, Annex D

Partial open system:
[1] Figure 3, Figure 5, *7.2.1.2*, 7.3.1

Partial PIXIT proforma:
[2] 10.5
[4] 6.3.3, *6.5*, A.2.2

Pass verdict:
[1] 3.7.6, **3.7.14**, *6.5.1.3*
[2] 13.2.7
[5] 7.6.2.3, A.2(2.n)

PATS:
[1] **3.6.29**, **4**
[4] **4**, 5.4, Figure A.1

PCO:
[1] **3.8.1**, **4**. 6.5.1.1, *7.3.2*, Figure 7, Figure 8, 7.6
[2] *12*, Figure 13, Figure 14
[4] **4**, 6.4
[5] **4.1**, 6.2.1.3.1

PCTR proforma:
[5] *Annex B*

PCTR:
[1] **3.7.8**, **4**, *6.5.2*
[2] 15
[5] **4.1**, Figure 1, 5.4, 6.4.4.1, 7.6.2.3, 7.7.2, 8.1.1, *8.3*, A.2(2.n), *Annex B*, C.2(6)

PDU:
[1] **4**, 7.3.1, Figure 6, Figure 7, 7.3.2.4, 7.5.1, Figure 8
[2] 6.2.3, 6.3.4, 7.3, 10.2.1, *10.3*, 10.4, *12*, Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, A.2.3, A.8.3, *A.8.5*, *A.8.6*, *B.5*
[4] **4**, A.1.2.1, A.1.2.2
[5] 7.6.2.3, C.2(7.2.4)

PETS:
[1] 3.5.8, **3.6.30**, **4**, 6.3.3
[4] **4**, 5.4, 5.5, *6.3*, 6.4, Figure A.1, A.2.3, A.3, Figure A.2, A.4.5
[5] **4.1**, 5.3, Figure 3, 7.4.1.1, 7.4.1.2, 7.6.2.1

Physical layer:
[1] 1.7, **3.1**, *B.1*
[2] 1.3, 12.2.1

PICS proforma:
[1] **3.4.7**
[2] 6.3.3, *7*, 8.2, 9, 15.1, *Annex A*, Figure A.1
[4] A.2.2
[5] Figure 2, *6.4.2*, 7.2.1.2

PICS:
[1] **3.4.6**, 3.5.1, 3.6.25, 3.6.26, 3.6.27, 3.6.28, 3.6.29, 3.6.30, **4**, *5.5*, 5.6, 5.7.3, 6.1.1, 6.3, Figure 1, Figure 10
[2] 6.3.3, 8.2, 10.4, 15.1, A.1.3, A.3.2, A.5

[4] **4**, 5.3, 5.5, 6.2.2, 6.3.3, Figure A.1, Figure A.2, A.4.3, A.4.5
[5] 1, **4.1**, Figure 1, 5.3, 6.1, 6.3.1.2, 6.4.1, *6.4.2*, 6.4.3.2, 6.4.4.1, 7.1, Figure 3, *7.2.1*, 7.3,1, 7.4.1.1, 8.3.1, B.2(1.2), B.2(2), B.2(3), B.2(5), B.2(6), C.2(5), C.2(6), C.2(7.1), C.2(7.2), Annex D

PIXIT proforma:
[1] **3.4.9**
[2] 15.1
[4] 6.1, 6.3.4, *6.5*
[5] 5.2, Figure 2, 6.3.1.2, *6.4.3*, *Annex C*

PIXIT:
[1] **3.4.8**, 3.6.25, 3.6.26, 3.6.27, 3.6.28, 3.6.29, 3.6.30, **4**, 5.7.4, *6.2*, 6.3, Figure 1, Figure 10
[2] 8.2, 12.3.5
[4] **4**, 5.3, 5.5, 6.2.2, 6.2.5, 6.5, Figure A.1, Figure A.2, A.4.3, A.4.5
[5] **4.1**, Figure 1, 5.2, 5.3, 6.1, Figure 2, 6.2.1.5.1, 6.2.1.5.2, 6.3.1.2, 6.4.1, *6.4.3*, 7.1, Figure 3, 7.2.1.2, 7.3.1, 7.4.1.1, A.2(2.n), B.2(1.3), B.2(6), C.2(6)

Point of control and observation:
[1] **3.8.1**, 6.5.1, 1, *7.3.2*, Figure 7, Figure 8, 7.6
[2] *12*
[4] 6.4
[5] 6.2.1.3.1

Predicate:
[2] A.2.1, A.8.6, A.9.1, Table A.2, *A.9.2.6*, A.9.2.7

Presentation layer:
[1] **3.1**, *B.6*

Protocol conformance test report:
[1] 1.3, **3.7.8**, *6.5.2*
[2] 15
[5] Figure 1, 5.4, 6.4.4.1, 7.6.2.3, 7.7.2, 8.1.1, *8.3*, A.2(2.n), *Annex B*, C.2(6)

Protocol data unit:
[1] **3.1**, 7.3.1, Figure 6, Figure 7, 7.3.2.4, 7.5.1, Figure 8
[2] 6.2.3, 6.3.4, 7.3, 10.2.1, *10.3*, 10.4, *12*, Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, A.2.3, A.8.3, *A.8.5*, *A.8.6*, *B.5*
[4] A.1.2.1, A.1.2.2
[5] 7.6.2.3, C.2(7.2.4)

Protocol implementation conformance statement:
[1] **3.4.6**, 3.5.1, 3.6.25, 3.6.26, 3.6.27, 3.6.28, 3.6.29, 3.6.30, *5.5*, 5.6, 5.7.3, 6.1.1, 6.3, Figure 1, Figure 10
[2] 6.3.3, 8.2, 10.4, 15.1, A.1.3, A.3.2, A.5
[4] 5.3, 5.5, 6.2.2, 6.3.3, Figure A.1, Figure A.2, A.4.3, A.4.5
[5] 1, Figure 1, 5.3, 6.1, 6.3.1.2, 6.4.1, *6.4.2*, 6.4.3.2, 6.4.4.1, 7.1, Figure 3, *7.2.1*, 7.3.1, 7.4.1.1, 8.3.1, B.2(1.2), B.2(2), B.2(3), B2(5), B.2(6) C.2(5), C.2(6), C.2(7.1), C.2(7.2), Annex D

Real open system:
[1] **3.1**, 3.4.1, 3.4.2, 3.4.4, 3.8.16, 5.4, *5.7. 7.2*, *7.6*
[2] 12.7.3
[5] 8.2.1, A.2(1.5)

Real system:
[1] **3.1**, 3.8.7, 5.1, 5.3, 5.4, 7.1, B.6.1
[2] 12.7.3
[4] A.1.2.1

Reference standardized ATS:
[1] 3.8, 12, **3.8.14**
[4] 1, 5.1, 5.2, 5.3, 6.2.1, 6.2.2, 6.2.3, 6.2.5, 6.3.2, 6.3.3, *6.3.4*, 6.4, 6.5, 6.6, A.1.2.1, A.1.3, A.2.1, A.2.2, A.2.3, A.4.3, A.4.5
[5] 6.3.2.1, 6.4.3.2, 7.3.1, 7.4.1.1, 7.6.4.1, 8.2.1

Relay system:
[1] Figure 4, Figure 5, *7.2.1.2*, 7.5.5
[2] *12.6*, *Figure 13*, *Figure 14*, 12.7.3

Remote embedded test method:
[1] B.1, B.2

Remote test method:
[1] **3.8.11**, *7.5.3*, Figure 8, 7.6, Annex B, B.2, B.4, B.5
[2] Figure 4, *12.3.5*, *12.4.5*, *Figure 8*, *12.5.5*, *Figure 12*, 12.7.2, D.2
[4] A.1.3, A.4.3
[5] *6.2.1.5*, C.2(5), C.2(7.2.4)

Repeatibility of results:
[1] **3.7.1**, *6.5.3*

Re-running test cases:
[5] 7.6.2.3, 7.6.3.3, 7.6.4.2

RS:
[2] **4**, *12.4.5*, *Figure 8*, 12.5.5, D.2

RSE:
[2] **4**, *12.5.5*, *Figure 12*

SAP:
[1] **4**, 7.3.2.7, 7.6
[5] **4.1**, C.2(3), C.2(7.2.1)

SATS:
[1] **3.6.25**, 3.6.29, **4**
[4] **4**, 5.4, 6.3.4, Figure A.1
[5] **4.1**, 7.3.1, 7.4.1.2, 7.6.2.3

SCS proforma:
[5] Figure 2

SCS:
[1] **3.4.11**, **4**, 5.5, 6.3.2
[2] A.5
[5] 1, **4.1**, Figure 1, 5.2, 6.1, 6.3.1.2, 6.4.1, 6.4.3.2, *6.4.4*, 7.2.1.2, A.2(1.4), C.2(5), E.2

SCTR proforma:
[1] 5.7.5
[5] *8.2*, *Annex A*

SCTR:
[1] **3.7.7**, **4**, *6.5.2*
[5] **4.1**, Figure 1, 5.4, 6.3.2.3, 6.4.4.1, 7.7.2, 8.1.1, *8.2*, *Annex A*

Selected abstract test suite:
[1] **3.6.25**, 3.6.29
[4] 5.4, 6.3.4, Figure A.1
[5] 7.3.1, 7.4.1.2, 7.6.2.3

Selected executable test suite:
[1] **3.6.26**, 3.6.30
[4] 5.4, 5.5, 6.3.1, Figure A.1

Selection:
[1] 3.8.12, 6.3.3, Figure 1
[2] 8.2, 10.5, A.1.3
[4] 5.2, 5.3, 6.2.4, 6.3.4, 6.3.5, 7, A.2.1, Figure A.1, A.4.3, A.4.5
[5] Figure 1, 5.3, *6.3.2*, Figure 3, *7.3*, 7.4.1.1, 9.1, B.2(6)

Semantically invalid test event:
[1] **3.7.13**
[2] 10.2.1, 10.2.2

Service access point:
[1] **3.1**, 7.3.2.7, 7.6
[5] C.2(3), C.2(7.2.1)

Service primitive:
[1] **3.2**

Service-provider:
[1] **3.2**

Service-user:
[1] **3.2**

Session layer:
[1] **3.1**, *B.5*

SETS:
[1] **3.6.26**, 3.6.30, **4**
[4] **4**, 5.4, 5.5, 6.3.1, Figure A.1

Single-layer test methods:
[2] *12.3.6*, *12.4*, 12.5.1
[5] 6.3.2.2, 6.4.3.2

Single-layer test suite:
[2] 10.2.2

Single-protocol IUT:
[2] 12.2.1, 12.3.6, *12.4*
[5] 7.1

Stable testing state:
[1] **3.6.15**
[2] 13.2.3, *13.2.5*

Standardized abstract test suite:
[1] **3.6.31**, 3.6.32, 3.8.12, 3.8.14, 6.1.2.4, 6.1.3.4, 6.1.4.4, 6.1.5, 7.3.2.3
[2] 5.2, 10.3.6, 12.3.4, 12.7.2, 13.1, 13.2.2, 13.2.4, 13.3, 13.4, *15*
[4] 1, 6.2.1, 6.2.5, 6.3.3
[5] 6.2.1.5.1, 6.3.2.1, 6.4.3.2, 7.1, 7.3.1, 7.3.2, 7.5.1, 7.5.2, 7.6.2.3, 8.3.1, 9.1

Static conformance requirements:
[1] **3.4.4**, 3.5.1, 5.2.3, *5.3*, 5.4, 5.7.4, 6.1.1, 6.1.3.1, 6.3.3, Figure 1
[2] 6.2.1, 6.3, 7.3, *9*, 10.2.1, A.2.4, A.8.3, A.9.1, B.4.5, Annex C
[5] B.2(2), B.2(5)

Static conformance review:
[1] **3.5.1**, 6.3.3, Figure 1
[2] A.1.3, A.2.5, A.8.10
[5] Figure 1, 5.3, Figure 3, 7.2, 8.3.1

Static conformance summary:
[5] *B.2(3)*

Status:
[2] *A.2.1*, A.2.5, A.6, A.8.3, A.8.5, A.8.6, A.9.1, A.9.2.1, Table A.1, Table A.2, Table A.3, A.9.2.7

Subnetwork:
[1] **3.1**
[2] Figure 13, Figure 14

Support:
[2] A.6, A.8.5, A.9.2.2

SUT operator:
[5] **3.4**

SUT:
[1] **3.4.2**, 4, 6.3.2, 6.5.2, *7.2.1*, Figure 7, 7.5.1
[2] *12*, Figure 5, Figure 6, Figure 7, Figure 8, Figure 10, Figure 11, Figure 12, 13.2.3
[4] **4**, A.1.2.1, A.1.3, A.4.4
[5] 3.4, *4.1*, 5.2, 6.1, Figure 2, *6.2*, 6.3.1.2, 6.3.1.3, 6.3.2.2, 6.4.1, 6.4.4.1, 6.4.4.2, 7.5.1, 7.5.2, 7.6.3.1, 8.2.1, 9.1, 9.2, *A.2(1.4)*, A.2(1.5), A.2(1.7), A.2(2.n), C.2(3), *C.2(5)*, C.2(6), C.2(7.2.1), C.2(7.2.4), Annex D

Syntactically invalid test event:
[1] **3.7.12**
[2] 10.2.1, 10.2.2

System conformance statement:
[1] **3.4.11**, 5.5, 6.3.2
[2] A.5
[5] 1, Figure 1, 5.2, 6.1, 6.3.1.2, 6.4.1, 6.4.3.2, *6.4.4*, 7.2.1.2, A.2(1.4), C.2(5), E.2

System conformance test report:
[1] **3.7.7**, 5.7.5, *6.5.2*
[5] Figure 1, 5.4, 6.3.2.3, 6.4.4.1, 7.7.2, 8.1.1, *8.2*, *Annex A*

System under test:
[1] **3.4.2**, 6.3.2, 6.5.2, *7.2.1*, Figure 7, 7.5.1
[2] *12*, Figure 5, Figure 6, Figure 7, Figure 8, Figure 10, Figure 11, Figure 12, 13.2.3
[4] A.1.2.1, A.1.3, A.4.4
[5] 3.4, 5.2, 6.1, Figure 2, *6.2*, 6.3.1.2, 6.3.1.3, 6.3.2.2, 6.4.1, 6.4.4.1, 6.4.4.2, 7.5.1, 7.5.2, 7.6.3.1, 8.2.1, 9.1, 9.2, *A.2(1.4)*, A.2(1.5), A.2(1.7), A2(2.n), C.2(3), *C.2(5)*, C.2(6), C.2(7.2.1), C.2(7.2.4), Annex D

Technical Corrigenda:
[2] 13.1

Test body:
[1] **3.6.9**, 8.2.1
[2] 11, *13.2*, D.2, D.4

Test campaign:
[1] **3.5.8**, 3.7.21, *6.3.3*
[4] 6.4, Figure A.2
[5] Figure 1, 5.3, 6.1, 6.3.1.2, Figure 3, 7.3.1, 7.3.2, 7.5.1, 7.5.2, *7.6*, 7.6.2.1, *7.7*, 8.2.1, 8.3.1, 9.1, B.2(4), *B.2(6)*

Test case error:
[1] 3.7.5, **3.7.17**, 3.7.18, 3.7.19, *6.5.1*
[5] 7.6.2.3, 8.3.1

Test case:
[1] **3.6.20**, Figure 9

Test coordination procedures:
[1] **3.8.5**, 3.8.6, 3.8.10, 3.8.11, *7.4*, 7.5.2, 7.5.3, Figure 8
[2] 8.2, 12.3, Figure 2, Figure 4, Figure 6, Figure 8, *12.8*, 13.1, 15.1
[4] 6.2.1, 6.4, 6.6, A.1.1
[5] 6.2.11, 6.2.1.3.2, 6.2.1.5.1, 6.2.1.5.2, 6.3.1.2, 6.3.1.3, Figure 3, *7.5*

Test event:
[1] 3.6.11, **3.6.12**, 3.6.13, 3.7.3, 3.7.9, 3.7.10, 3.7.11, 3.7.12, 3.7.13, 6.3.3, 6.5.1, 7.3.2.2, *8.1*, Figure 9, 8.2
[2] 10.2, 10.4
[4] 6.3.3, 6.4, A.1.2.1, A.3, A.4.5
[5] 6.2.1.5.2, 8.3.1

Test group objective:
[1] **3.6.6**, *8.1*
[2] 8.2, 10.3

Test group:
[1] 3.6.6, **3.6.21**, *8.1*, Figure 9
[2] 8.2, *10.2*, 10.3
[4] 6.3.3

Test laboratory and client:
[5] 3.5, 5.3, 6.1, *6.3*, *6.3.1.1*, 6.3.2.3, 6.3.3, 6.4.1, 7.2.3, *7.5.2*, 7.7.1, 7.7.4, 8.3.3, Annex D, E.2

Test laboratory checklist:
[5] **3.6**, Figure 2

Test laboratory manager:
[5] **3.2**, A.2(1.1), B.2(1.1)

Test laboratory:
[1] 1.3, 3.4.12, **3.4.13**, 3.8.16, 6.5.4, 6.5.5, 9, Figure 10
[2] 10.4, 13.1, 13.2.8, 15.1, A.5
[4] 6.2.5, 6.3.5, 6.5, 6.6, A.2.1, A.4.1, A.4.4, *A.4.5*
[5] *1*, 3.2, 3.3, 3.6, 3.7, 5.2, *6*, Figure 2, 6.3.1.2, 6.3.2.1, 6.4.2.2, *6.4.3.2*, 6.4.4.2, 6.4.5.2, *7.2.1*, *7.3.1*, *7.4.1*, *7.6.2*, 7.6.2.1, 7.7.2, 8.1.1, *8.2.1*, *8.3.1*, *9.1*, A.1, A.2(1.2), A.2(1.6), A.2(1.8), B.1, B.2(1.4), B.2(1.5)

Test management protocol:
[1] **3.8.6**, 3.8.10, 7.4, *7.5.3*, 7.6
[2] 8.3, *12.3.4*, 12.5.4, 13.1, *14*, 15.1
[4] 6.2.1, *A.1.4*, *A.4.4*
[5] 6.2.1.4.1

Test method:
[1] *3.8*, *7.5*, 7.6, *7.6*

Test notation:
[1] 8.2.1, 9
[2] 5.2, 5.3, 8.2, 13.1, 13.2
[4] 6.2.2, A.4.3

Test operations:
[1] *6.3*
[4] 6.6, A.2.1
[5] 5.1, Figure 1, 5.2, *5.3*, 5.4, 6.1, 6.3.1.2, *7*, 9.1

Test operator:
[5] **3.3**, 6.2.1.3.2, 6.2.1.5.2

Test outcome:
[1] **3.7.3**, 3.7.4, 3.7.5, 3.7.14, 3.7.15, 3.7.16, 3.7.21, *6.5*
[4] 6.4

Test postamble:
[1] **3.6.10**, *8.2.1*, 8.2.2
[2] *13.2*, D.3

Test preamble:
[1] **3.6.8**, *8.2.1*, 8.2.2
[2] *13.2*, D.3

Test preparation:
[4] A.2.1, A.4.5
[5] 3.6, 3.7, Figure 1, *6.3.1.2*, 6.3.1.3, *6.4.1*

Test purpose:
[1] 3.5.5, 3.6.3, **3.6.5**, 3.6.7, 6.1.5, *8.1*, Figure 9, *8.2*, Figure 10
[2] 8.2, 10.1.3, *10.3*, 10.4, 10.5, 11, 13.2.3
[4] 6.3.3

Test realization:
[1] **3.8.13**, Figure 10
[4] 7, A.2.1

Test realizer:
[1] **3.8.15**, 6.2, Figure 10
[2] 10.4, 13.1, 13.2.8, 13.3, 15.1
[4] 6.2.1, *6.2.5*, 6.3.4, *6.5*, *6.6*, *A.2.2*, *A2.3*, *A.4.4*, *A.4.5*

Test repetition:
[1] 6.5.4

Test report:
[1] 5.7.4, 6.3, Figure 1, *6.5.2*, 9, Figure 10
[4] 6.4
[5] 1, 5.1, 5.3, *5.4*, 6.3.1.2, 7.6.2.3, 7.7.2, *8*, 9.1, A.2(1.6), B.2(1.4)

Test step:
[1] 3.6.8, 3.6.9, 3.6.10, **3.6.11**, *8.1*, Figure 9
[2] 8.2, 10.1.3, 13.1, 13.2.6
[4] A.4.5

Test suite specifier:
[2] *10.2*

Test suite structure:
[1] Figure 10
[2] 8.3, *10*, 13.1, A.1.2

Test suite:
[1] 1.6, *3.6*, **3.6.19**, 3.8.14, *8*, Figure 9

Test system:
[1] 3.6.14, 3.7.20, **3.8.7**, 3.8.8, 6.3.3, 6.5.1.6, 7.3.2.2, Figure 7, 7.3.2.6, 7.5.2, Figure 8, 8.2.3
[2] *12*, Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14
[4] 1, 5.3, 6.2.1, 6.3.3, A.1.1, A.1.2.1, A.4.1, *A.4.2*
[5] 6.2.1.2.1, 8.3.1

Test verdict:
[1] 3.7.1, **3.7.6**, 3.7.14, 3.7.15, 3.7.16, 3.7.21, *6.5.1*, 8.2.1
[2] 13.2.3
[4] 6.3.3, 6.4, A.1.2.1
[5] *7.6.2.3*, 7.6.3.3, 7.7.1, 8.3.1, *B.2(6)*

Testing state:
[1] **3.6.14**, 3.6.15, 3.6.16, 3.6.17, 3.6.18

Timers:
[1] 5.3
[2] 10.2.1, 10.2.2, 10.4
[5] 7.4.1.1, B.2(1.3), C.2(6), C.2(7.2), *C.2(7.2.3)*

TMP:
[1] **3.8.6**, 3.8.10, **4**, 7.4, *7.5.3*, 7.6
[2] 8.3, *12.3.4*, 12.5.4, 13.1, *14*, 15.1
[4] 6.2.1, *A.1.4*, *A.4.4*
[5] **4.1**, 6.2.1.4.1

TM-PDU:
[1] **4**
[2] *12.3.4*, 12.4.4, 12.5.4
[4] A.1.2.1

Transfer syntax:
[1] 1.1, **3.1**, 3.7.10, 6.1.1, B.6.2, *B.6.4*
[2] 6.1, 9.1, 13.1, A.8.6, A.9.1
[4] 6.4
[5] 6.4.2.1, 6.4.2.3, 6.4.4.1

Transient testing state:
[1] **3.6.17**
[2] 13.2.5

Transport layer:
[1] **3.1**, *B.4*

Transverse test method:
[1] 7.5.5, B.3
[2] 12.6.1, *12.6.3*, *Figure 14*

Tree and tabular combined notation:
[1] Figure 10
[2] 5.3, *13.2*, A.9.2.6, *Annex D*
[4] 6.4, A.4.3
[5] 2, C.2(7.2.4)

TSS&TP compliance clause:
[2] *10.5*

TSS&TP:
[2] **4**, *10*

TTCN:
[1] **4**, Figure 10
[2] **5**.3, *13.2*, A.9.2.6, *Annex D*
[4] **4**, 6.4, A.4.3
[5] 2, C.2(7.2.4)

Unforeseen test outcome:
[1] **3.7.5**, *6.5.1.4*

Unidentified test event:
[1] **3.6.13**
[2] 13.2.7
[5] 7.6.2.3

Upper tester:
[1] **3.8.3**, *7.4*, 7.5.2, 7.5.3, Figure 8, 7.6
[2] 8.2, 14
[4] 1, 6.2.1, 6.4, 6.6, A.1.1, *A.1.3*, *A.1.4*, A.4.2, A.4.3, *A.4.4*, A.4.5
[5] 6.2.1.4.1, 6.2.1.4.2, 6.3.1.2, 7.5.2, 7.6.4.1, C.2(5)

UT:
[1] **3.8.3**, 4, *7.4*, 7.5.2, 7.5.3, Figure 8, 7.6
[2] 8.2, 14
[4] 1, 6.2.1, 6.4, 6.6, A.1.1, *A.1.3*, *A.1.4*, A.4.2, A.4.3, *A.4.4*, A.4.5
[5] 6.2.1.4.1, 6.2.1.4.2, 6.3.1.2, 7.5.2, 7.6.4.1, C.2(5)

Valid test event:
[1] **3.7.9**, 3.7.14, 6.1.4.5
[2] 10.2.1, 10.2.2, 10.3.2, 10.4, 13.2.7, B.5.7
[5] 7.6.2.3

Verdict:

[1] 3.7.1, **3.7.6**, 3.7.14, 3.7.15, 3.7.16, 3.7.21, *6.5.1*, 8.2.1
[2] 13.2.3
[4] 6.3.3, 6.4, A.1.2.1
[5] *7.6.2.3*, 7.6.3.3, 7.7.1, 8.3.1, *B.2(6)*

YL:

[2] **4**, 12.6.1, *12.6.2*, *Figure 13*

YT:

[2] **4**, 12,6.1, *12.6.3*, *Figure 14*