
**Information technology — Open Systems
Interconnection — Conformance test suite
for the Session protocol —**

Part 4: **STANDARD PREVIEW**

Test management protocol specification

*Technologies de l'information — Interconnexion de systèmes ouverts —
Suite de tests de conformité pour le protocole de session —
Partie 4: Spécification du protocole de gestion de tests*



Contents

	Page
Foreword	iii
Introduction.....	iv
1 Scope	1
2 Normative references	1
3 Definitions.....	1
3.1 Reference model definitions	1
3.2 Conformance testing methodology and framework definitions	1
3.3 Session test management protocol definitions	1
4 Symbols and abbreviations	2
5 Overview.....	2
6 Service assumed from the Session IUT	3
6.1 General	3
6.2 Use of parameters by service primitives	3
7 Definition of the SL-TMP	3
7.1 General procedures	3
7.2 Implicit reaction.....	3
7.3 Explicit reaction.....	3
7.3.1 Definition of SL-TMPDU sent by LT	3
7.3.2 General procedure for executing the reactions memorized in the FIFO queue.....	4
7.3.2.1 Use of this general procedure.....	4
7.3.2.2 Execution of this general procedure	4
7.3.3 Processing a valid explicit reaction.....	4
7.3.3.1 Processing an "Absorption of SP" reaction.....	5
7.3.3.2 Processing an "execute the reactions memorized in the FIFO queue" reaction.....	5
7.3.3.3 Processing a "Purge reactions memorized in the FIFO queue" reaction.....	6
7.3.3.4 Processing any other reaction	6
7.3.4 Building a reaction	6
7.3.4.1 Building parameters.....	7
7.3.4.2 Building User-Data.....	7
7.4 Service Violation	8
7.5 Protection against connection rejection by session.....	8
8 Coding rules	9
8.1 PDU sent by LT.....	9
8.1.1 Coding of REACTION parameter.....	10
8.1.2 Coding of RES-PAR-VAL parameter	11
8.2 Coding of the RP-SL-TMPDU	12
8.2.1 Coding of the RECEIVED-SP parameter.....	12
8.2.2 Coding of RECEIVED-PAR parameter	12

© ISO/IEC 1996

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 publisher.

ISO/IEC copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Foreword

ISO (The International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10168 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open systems interconnection, data management and open distributed processing*.

ISO/IEC 10168 consists of the following parts, under the general title *Information technology - Open Systems Interconnection - Conformance test suite for the Session protocol*:

- Part 1: *Test suite structure and test purposes*
- Part 4: *Test management protocol specification*

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 10168-4:1996](https://standards.iteh.ai/catalog/standards/sist/9497039d-b42a-4451-8e43-e89bcc6ebca1/iso-iec-10168-4-1996)

<https://standards.iteh.ai/catalog/standards/sist/9497039d-b42a-4451-8e43-e89bcc6ebca1/iso-iec-10168-4-1996>

Introduction

This part of ISO/IEC 10168, a multipart International Standard, specifies a Test Management Protocol together with an Upper Tester for the Coordinated test method as applied to ISO/IEC 8327-1, the Session protocol, versions 1 and 2, except for the symmetric synchronize and the data separation functional units.

A fundamental objective of the related standardized conformance test suites is to establish uniform conformance testing and unambiguous evaluation procedures for checking the ability of a Session protocol implementation to operate according to ISO/IEC 8327.

The standardization of these test suites should lead to comparability and wide acceptance of test results produced by different test laboratories, and therefore minimize repeated conformance testing of the same Session protocol implementation.

The conformance test suites based on this part of ISO/IEC 10168 are designed for use by :

- a) test laboratories which provide a conformance testing service for the Session protocol ;
- b) test realizers which provide a means of testing to be used by such test laboratories ;

The purpose of conformance testing is to increase the probability that different implementations are able to interwork, although conformance testing alone cannot give a guarantee of interworking. Conformance testing increases the confidence that each implementation conforms to the protocol specification by establishing that it has the required capabilities and that its behaviour conforms to the protocol specification in representative instances of communication.

[ISO/IEC 10168-4:1996](https://standards.iteh.ai/catalog/standards/sist/9497039d-b42a-4451-8e43-e89bcc6ebca1/iso-iec-10168-4-1996)

<https://standards.iteh.ai/catalog/standards/sist/9497039d-b42a-4451-8e43-e89bcc6ebca1/iso-iec-10168-4-1996>

Information technology — Open Systems Interconnection — Conformance test suite for the Session protocol —

Part 4:

Test management protocol specification

1 Scope

This part of ISO/IEC 10168 defines :

- a) procedures for a single protocol for the transfer of data and control information from one test management entity (the Lower Tester) to a peer test management entity (the Upper Tester);
- b) the structure and encoding of the test management protocol data units used to operate the test management protocol.

These procedures are defined in term of :

- a) the service primitives sent and received by the Upper Tester ;
- b) the Session Layer - Test Management Protocol Data Unit sent and received by the Upper Tester.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10168. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10168 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 7498-1 : 1994, *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*.

ISO/IEC 7498-3 : 1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and addressing*.

ISO/IEC 8326 : 1996, *Information technology - Open Systems Interconnection - Session service definition*.

ISO/IEC 8327-1 : 1996, *Information technology - Open Systems Interconnection - Connection-oriented Session protocol: Protocol specification*.

ISO/IEC 8348 : 1993, *Information technology - Open Systems Interconnection - Network Service Definition*.

ISO/IEC 10731 : 1994, *Information technology - Open Systems Interconnection - Basic Reference Model - Conventions for the definition of OSI services*.

ISO/IEC 9646-1 : 1994, *Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts*.

ISO/IEC 9646-2 : 1994, *Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification*.

3 Definitions

3.1 Reference model definitions

For the purposes of this part of ISO/IEC 10168, the following definitions are taken from ISO/IEC 7498-1 :

- a) Service primitive,
- b) Session service,
- c) Session connection,
- d) Session service data unit,
- e) Protocol data unit.

3.2 Conformance testing methodology and framework definitions

For the purposes of this part of ISO/IEC 10168, the following definitions are taken from ISO/IEC 9646-1 :

- a) Upper tester,
- b) Lower tester,
- c) Test coordination procedures,
- d) Test management protocol,
- e) Coordinated test method,
- f) Implementation under test,
- g) System under test,
- h) Point of control and observation,
- i) Verdict.

3.3 Session test management protocol definitions

For the purposes of this part of ISO/IEC 10168, the following definitions are also used :

3.3.1 TLV : an encoding technique in which a parameter is encoded by a type, a length and a value.

3.3.2 Session layer test management protocol : the test management protocol used to test session entities.

3.3.3 Session layer test management protocol data unit : a protocol data unit of the above defined protocol.

3.3.4 Report test management protocol data unit: a protocol data unit sent by the upper tester to report to the lower tester the service primitive and associated received by the UT.

4 Symbols and abbreviations

The following abbreviations are used :

ADDRF	Additional reference information
CDADDP1	Called Session Address Part 1
CDADDP2	Called Session Address Part 2
CDADDP3	Called Session Address Part 3
CDSSREF	Called SS user reference
CGADDP1	Calling Session Address Part 1
CGADDP2	Calling Session Address Part 2
CGADDP3	Calling Session Address Part 3
CGSSREF	Calling SS user reference
COMREF	Common reference
CS	Coordinated Single Layer test method
ISERNUM	Initial Serial Number
TOKSET	Initial Token Assignment
LT	Lower Tester
NACTID	New Activity identifier
OACTID	Old Activity identifier
PCO	Point of Control and Observation
REASON	Reason
RESULT	Result
RP-SL-TMPDU	Report SL-TMPDU
RSTYPE	Resynchronization type
SERNUM	Serial number
SL-TMP	Session Layer Test Management Protocol
SL-TMPDU	Session Layer Test Management Protocol Data Unit
SP	Service Primitive
SQOS	Session quality of service
SYNCTYPE	Synchronization type
TLV	Type, Length, Value
TMP	Test Management Protocol
TOKITEM	Tokens (give/please)
USEREQ	Functional Units
UT	Upper Tester

- SL-TMPDU explicitly designates the service primitive(s) to be issued.
- The SL-TMPDU also indicates the parameters to be used in the service primitive.
- The SL-TMPDU also indicates what user data is to be included in the service primitive (if relevant). The user data forms an SL-TMPDU sent by the UT to the LT.

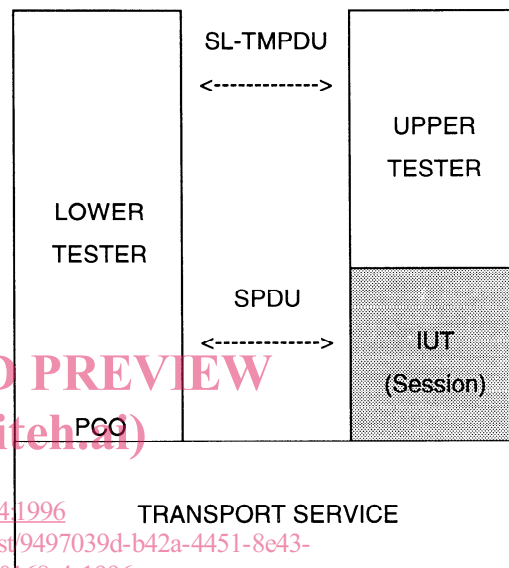


Figure 1 - Description of Test Management Protocol

Additionally the expected reaction may be delayed : the upper tester maintains one FIFO queue of delayed reactions associated with each existing connection and on receipt of a particular stimulus the UT shall issue in turn all service primitives held in the queue associated with the connection.

The UT shall be able to hold at least five reactions in each FIFO queue.

The TMP definition also defines the reaction expected from the UT on receipt of a service primitive which cannot contain user data and cannot therefore carry any SL-TMPDU (this is called implicit reaction).

NOTES

1 - The maximum length of the user data parameter of the EXPEDITED DATA SPDU is only 14 bytes. That means that not all types of SL-TMPDUs can be included in the user data parameter of the EXPEDITED DATA SPDU.

2 - Most of the primitives that "may not contain user data" in Session version 1, may contain user data in Session version 2. However the implicit reaction that was used with version 1 for such primitives is still valid.

6 Service assumed from the Session IUT

6.1 General

The service assumed is the service described in ISO/IEC 8326.

6.2 Use of parameters by service primitives

Since the description of the SL-TMP makes reference to parameters of service primitives, Table 1 summarizes which parameters are supplied in the TMP for use by the UT. Parameters which are not given in the table for a specific primitive are derived locally by the UT.

7 Definition of the SL-TMP

7.1 General procedures

- a) On receipt of a service primitive which does not contain user data the UT shall apply the procedures defined in 7.2 (implicit reaction).
- b) On receipt of a service primitive which contains user data the UT shall apply the procedures defined in 7.3 (explicit reaction).
- c) Additionally the UT shall follow the rules given in 7.4 and 7.5.

7.2 Implicit reaction

When receiving a session service primitive which does not contain user data, the UT shall apply one of the following :

- a) if the service primitive is an S-P-ABORT-Indication or an negative S-CONNECT-Confirm with a result different from "by user", destroy the FIFO queue associated with the connection; or
- b) if the service primitive is either an S-P-EXCEPTION-REPORT-Indication, S-TOKEN-PLEASE-Indication, an S-ACTIVITY-INTERRUPT-Confirm or S-ACTIVITY-DISCARD-Confirm, absorb the service primitive (i.e. do nothing) ; or
- c) if the service primitive is either a S-TOKEN-GIVE-Indication, S-CONTROL-GIVE-Indication, S-ACTIVITY-INTERRUPT-Indication or S-ACTIVITY-DISCARD-Indication, apply the procedure defined in 7.3.2 (general procedure for executing the reactions memorized in the FIFO queue).
- d) if the service primitive is none of the above and the connection is still open :
 - i) destroy the FIFO queue associated with the connection;
 - ii) issue an S-U-ABORT-Request service primitive without user data.

7.3 Explicit reaction

After receiving a service primitive containing user data the UT shall :

- a) check if the received user data represent a valid SL-TMPDU sent by the LT. This checking is made in accordance to:

- the definition of the SL-TMPDU (see 7.3.1),
- the coding of the SL-TMPDU (see 8.1).

- b) then, if the SL-TMPDU is valid apply the procedure defined in 7.3.3 (processing a valid explicit reaction), otherwise apply the procedure defined in 7.2 (implicit reaction).

7.3.1 Definition of SL-TMPDU sent by LT

There is only one SL-TMPDU sent by the LT and received by the UT. This SL-TMPDU has the following parameters:

REACTION : This parameter indicates the name of the service primitive(s) that the UT shall build.

DELAY : The value YES indicates that the service primitive(s) built according to REACTION is to be stored in the FIFO queue associated with the connection.

The value NO indicates that the SP(s) is to be passed immediately to the session entity.

RES-DAT-TYP : This parameter indicates how to build the user data of the SP designed by the REACTION parameter :

- NO means no user data,
- ECHO means echo the received user data (i.e. the received SL-TMPDU),
- REPORT means that a RP-SL-TMPDU shall be built by the UT, which contains the name and parameters of the received SP, and put as user data of the SP built according to REACTION.

RES-PAR-VAL : This parameter consists of a series of TLV encoded fields, each of which corresponds to a parameter of the SP to be built in accordance to the REACTION parameter.

DATA : Contains transparent user data, up to the end of the data field of the received SP.

Table 3 in 8.1.1 defines the possible values of the REACTION parameter.

The following additional rules apply to the received SL-TMPDU:

- a) If REACTION is "Absorption of SP" or "Purge reactions memorized in FIFO", all other parameters have no significance and may take any value.
- b) If REACTION is "Execute reactions memorized in FIFO" the DELAY parameter shall be set to NO and all other parameters have no significance and may take any value.
- c) If REACTION is "Build S-DATA-Request and S-TOKEN-GIVE-Request" or "Build S-TYPED-DATA-Request and S-TOKEN-GIVE-Request" or "Build S- EXPEDITED-Request and S-TOKEN-GIVE-Request", the RES-PAR-VAL parameter may take any value and the token(s) to be used when building the S-TOKEN-GIVE- Request are all available tokens.
- d) If REACTION is "Build S-U-ABORT-Request and S-CONNECT-Request" the RES-PAR-VAL applies only to the S-CONNECT-Request primitive and shall follow rule f) below for that SP.

- e) If REACTION is "Build S-CONNECT-Response and S-TOKEN-PLEASE-Request" the RES-PAR-VAL applies first to the S-CONNECT-Response primitive and then to the TOKEN-PLEASE-Request service primitive and shall follow rule f) below for each of them in turn. The RES-DAT-TYP parameter applies to the expected S-CONNECT-Response primitive, while the expected S-TOKEN-PLEASE-Request primitive will contain no user data.
- f) Otherwise the RES-PAR-VAL parameter shall contain one field, TLV encoded, for each parameter associated with the SP designated by REACTION and determined as follows :
- i) the name of the SP is used to determine the entry in table 1 to be used ;
 - ii) the entry gives the list of associated parameters and the following rules apply to this list:
 - all parameters given in the list shall always be present unless the parameter is conditional in ISO/IEC 8326. When the parameter is conditional in ISO/IEC 8326, it shall be omitted if its presence would cause the test responder to violate the service specification,
 - they shall always be in the same order as in the list,
 - for a given parameter the length shall be as specified by table 4,
 - RES-PAR-VAL shall not include any extra parameter, even with a null length.
- b) Otherwise the UT shall remove the SP at the head of the FIFO queue and pass the SP to the session service and :
- 1) if the service primitive was an S-CONNECT-Request, a new FIFO queue is created for the new connection, the remaining content of the old FIFO queue is moved to the new FIFO queue, and the UT shall continue to execute the procedure from 7.3.2.2 a) above for the new connection; or
 - 2) if the service primitive passed to the session service terminates the current connection (i.e. S-U-ABORT-Request or S-CONNECT-Response with negative RESULT parameter or S-RELEASE-Response and no S-RELEASE-Confirm is still awaited) then :
 - a) the UT shall test the head of the FIFO queue and,
 - b) if the head of the queue is an S-CONNECT-Request, then the UT shall remove the S-CONNECT-Request from the head of the queue and pass it to the session service, create a new FIFO queue associated with the new connection, move the content of the old queue to the new queue and destroy the old queue, and then continue to execute the procedure from 7.3.2.2 a) above; or
 - c) otherwise (the head does not content an S-CONNECT-Request primitive) the UT shall destroy the FIFO queue and terminate the procedure.
 - 3) if the service primitive passed to the session service does not belong to the categories defined in 1) and 2) above, the UT shall continue to execute the procedure in 7.3.2.2 a) above.

7.3.2 General procedure for executing the reactions memorized in the FIFO queue

7.3.2.1 Use of this general procedure

This general procedure is used in three cases :

- a) An S-TOKEN-GIVE-Indication, S-CONTROL-GIVE-Indication, S-ACTIVITY-INTERRUPT-Indication or S-ACTIVITY-DISCARD-Indication is received (see 7.2 c) ; or
- b) An "Execute reactions memorized in FIFO queue" reaction is received, contained in a service primitive which does not terminate the connection (see 7.3.3.2 b) ; or
- c) An "Execute reactions memorized in FIFO queue" reaction is received, contained in a service primitive which terminates the connection (see 7.3.3.2 c).

NOTE - In cases a) and b) the procedure is entered from the beginning, while in case c) the procedure is entered from 7.3.2.2.b)2a) (since the current session connection does not exist any more).

7.3.2.2 Execution of this general procedure

When executing this procedure :

- a) If the FIFO queue associated with the connection is empty the UT shall terminate the procedure.

7.3.3 Processing a valid explicit reaction

When processing this reaction the UT shall do one of the following :

- a) if the REACTION parameter of the received SL-TMPDU is "Absorption of SP" the UT shall follow the procedure defined in 7.3.3.1 below; or
- b) if the REACTION parameter of the received SL-TMPDU is "Execute reactions memorized in FIFO" the UT shall follow the procedure defined in 7.3.3.2 below ; or
- c) if the REACTION parameter of the received SL-TMPDU is "Purge reactions memorized in FIFO" the UT shall follow the procedure defined in 7.3.3.3 below ; or
- d) otherwise the UT shall follow the procedure defined in 7.3.3.4 below.

SERVICE PRIMITIVE NAME	PARAMETERS SUPPLIED IN THE TMP
S-CONNECT-Request	CGSSREF COMREF ADDRREF ISERNUM TOKSET USEREQ CGADDP1 CGADDP2 CGADDP3 CDADDP1 CDADDP2 CDADDP3 SQOS
S-CONNECT-Response	CDSSREF COMREF ADDRREF ISERNUM TOKSET USEREQ CDADDP1 CDADDP2 CDADDP3 SQOS RESULT
S-RELEASE-Request S-RELEASE-Response S-U-ABORT-Request S-DATA-Request	RESULT
S-TOKEN-GIVE-Request	TOKITEM
S-TOKEN-PLEASE-Request-Request	TOKITEM
S-EXPEDITED-DATA-Request	
S-TYPED-DATA-Request	
S-MINOR-Request S-MINOR-Response	SYNCTYPE SERNUM
S-MAJOR-Request S-MAJOR-Response	
S-RESYNC-Request S-RESYNC-Response	TOKSET RSTYPE SERNUM TOKSET SERNUM
S-CAPABILITY-DATA-Request S-CAPABILITY-DATA-Response	
S-U-EXCEPTION-REPORT-Request	REASON
S-CONTROL-GIVE-Request S-ACTIVITY-START-Request S-ACTIVITY-RESUME-Request	NACTID CDSSREF CGSSREF COMREF ADDRREF OACTID SERNUM NACTID
S-ACTIVITY-END-Request S-ACTIVITY-END-Response S-ACTIVITY-INTERRUPT-Request S-ACTIVITY-INTERRUPT-Response S-ACTIVITY-DISCARD-Request S-ACTIVITY-DISCARD-Response	REASON REASON

Table 1 - Use of Parameters by primitives

7.3.3.1 Processing an "Absorption of SP" reaction

When processing this reaction for a given received service primitive on a given connection, the UT shall do one of the following :

- if the received SP terminates the current connection (i.e. S-CONNECT-Confirm negative, S-U-ABORT-Indication, S-P-ABORT-Indication or S-RELEASE-Confirm and no collision of release), the UT shall destroy the FIFO queue ; or
- if the received SP is an S-CONNECT-Indication, the UT creates an empty FIFO queue for this new connection ; or
- if the received SP does not belong to one of the categories defined in a) or b) above, the UT simply absorbs the SP (i.e. do nothing).

7.3.3.2 Processing an "execute the reactions memorized in the FIFO queue" reaction

When processing this reaction for a given received service primitive on a given connection, the UT shall do one of the following :

- if the received SP is an S-CONNECT Indication, the UT creates an empty FIFO queue for this new connection ; or
- if the received service primitive terminates the current connection (i.e. S-CONNECT-Response negative, S-U-ABORT-Indication, S-P-ABORT-Indication or S-RELEASE-Confirm and no collision of release), the UT shall execute the procedure defined in 7.3.2.2 b)2)a); or
- if the received SP does not belong to one of the categories defined in a) and b) above, the UT shall execute the procedure defined in 7.3.2 (general procedure for executing the reactions memorized in the FIFO queue).

7.3.3.3 Processing a "Purge reactions memorized in the FIFO queue" reaction

When processing this reaction for a given received service primitive on a given connection, the UT shall do one of the following :

- a) if the received service primitive terminates the current connection (i.e. S-CONNECT-Confirm negative, S-U-ABORT-Indication, S-P-ABORT-Indication or S-RELEASE-Confirm and no collision of release), the UT shall destroy the FIFO queue associated with the connection (i.e. remove the content of the queue and the queue itself) ; or
- b) otherwise, the UT shall remove the content of the queue (the memorized reactions) and leave the queue empty.

7.3.3.4 Processing any other reaction

7.3.3.4.1 Delayed reaction

When processing a delayed reaction for a given received service primitive on a given connection, the UT shall do one of the following :

- a) if the received service primitive terminates the current connection (i.e. S-CONNECT-Confirm negative, S-U-ABORT-Indication, S-P-ABORT-Indication or S-RELEASE-Confirm and no collision of release), the UT shall destroy the FIFO queue ; or
- b) if the received SP is an S-CONNECT-Indication, the UT shall first create a FIFO queue associated with this new connection and then build a session service primitive according to the procedure described in 7.3.4, and store it into the FIFO queue ; or
- c) otherwise (the received SP does not belong to one of the categories defined in a) and b) above, the UT shall build a session service primitive according to the procedure described in 7.3.4, and store it into the FIFO queue.

The UT shall be capable of storing at least five memorized reactions in a FIFO queue.

NOTE - If the REACTION parameter indicates to build two service primitives, then both of them are stored (in the order specified by the REACTION parameter). If storing a SP in the FIFO queue would lead to the size of the queue exceeding five, the UT may apply the procedure defined in 7.2 (implicit reaction).

7.3.3.4.2 Immediate reaction

When processing an immediate reaction for a given received service primitive on a given connection, the UT shall do one of the following :

- a) if the received SP terminates the current connection the UT shall :
 - 1) if the REACTION is not "Build an S-CONNECT-Request" the FIFO queue is destroyed ; or
 - 2) otherwise (REACTION is "Build an S-CONNECT-Request") the S-CONNECT-Request SP is built according to 7.3.4, a new

FIFO queue is created for this connection, the content of the queue associated with the current connection is moved to newly created FIFO queue and the old queue is destroyed ;

- b) if the received SP is an S-CONNECT-Indication, the UT shall create an empty FIFO queue associated with that connection and then apply the procedure defined in 7.3.3.4.2 c) below; or
- c) otherwise (the received SP does not belong to one of the categories defined in a) and b), the UT shall :
 - 1) if the REACTION is such that it will terminate the current connection, the responding service primitive is built according to 7.3.4 and passed to the session service and the FIFO queue is destroyed ; or
 - 2) if the REACTION is "Build an S-U-ABORT-Request and then an S-CONNECT-Request", an S-U-ABORT-Request (no user data) is issued, the FIFO queue is destroyed, an S-CONNECT-Request primitive is built according to the procedure defined in 7.3.4 and passed to the session service and an empty FIFO queue is created and associated with the new connection ; or
 - 3) if the REACTION is "Build an S-CONNECT-Request", the S-CONNECT-Request SP is built according to 7.3.4 and passed to the session service, a new FIFO queue is created and associated with the new connection, and the content of the old FIFO is moved to the new FIFO assigned to the new connection ; or
 - 4) if the REACTION is neither a "Build an S-U-ABORT-Request and then an S-CONNECT-Request", nor a "Build an S-CONNECT-Request" nor a primitive which would terminate the current connection a responding session service primitive is built according to 7.3.4 and passed to the session service.

7.3.4 Building a reaction

The service primitive to be built is the one designated by the REACTION field of the received SL-TMPDU, according to table 3.

Additionally the following applies :

- a) in the case of multiple reactions the two primitives are to be built consecutively,
- b) the rules for building the parameters of the built SP are given in 7.3.4.1,
- c) the rules for building the user data of the built SP are given in 7.3.4.2.

7.3.4.1 Building parameters

The parameters are built according to the fields received in the RES-PAR-VAL parameter. Each field contains the value to be used for each of the parameters of the service primitive as defined by table 1.

In the case where two SPs are to be built the following rules apply :

- a) S-CONNECT-Response followed by S-PLEASE-TOKEN-Request : the RES-PAR-VAL parameter contains first the values to be used for the S-CONNECT-Response followed by the value to be used for the S-TOKEN-PLEASE-Request (TOKITEM).
- b) S-DATA-Request, S-TYPED-DATA-Request or S-EXPEDITED-Request followed by S-GIVE-TOKEN-Request : the RES-PAR-VAL parameter is not used since the first service primitive has no parameter (only user data) and the TOKITEM to be used for the second one is "all available tokens".
- c) S-U-ABORT-Request followed by S-CONNECT-Request : the ABORT is to be built without parameters and user data, and the parameters described in RES-PAR-VAL apply to the parameters of the S-CONNECT-Request.

7.3.4.2 Building User-Data

When the response service primitive is built the following rules apply:

- a) If the built SP cannot contain user data (this is not possible when version 2 is in use, but still possible when version 1 is in use for the connection), the UT ignores the RES-DAT-TYP parameter in the received SL-TMPDU.
NOTE - It is the responsibility of the abstract test specifier to ensure that the LT does not send a SL-TMPDU asking for an echo of user data on these service primitives, which do not have the user data parameter in session version 1.
- b) If the built SP may contain user data and the RES-DAT-TYP parameter specifies NO, no user data is included in the built SP.
- c) If the built SP may contain user data and the RES-DAT-TYP parameter specifies ECHO, user data is built according to the procedure described in 7.3.4.2.1 and passed as the user data field of the built SP.
- d) If the built SP may contain user data and the RES-DAT-TYP parameter specifies REPORT, user data is built according to the procedure described in 7.3.4.2.2 and passed as the user data of the built SP.

7.3.4.2.1 Echo

The user data is built according to the following procedure :

- a) if the first three octets of the DATA parameter of the received SL-TMPDU contain the characters "REP" (uppercase or lowercase, ASCII encoded) and the remaining characters in the range "0" to "9", then this shall be taken as a decimal number N and the user data put into the built SP shall be the octet 99 (hexadecimal value) repeated N times.
- b) otherwise, the user data shall contain the user data as received in the received SL-TMPDU. All the user data field of the received SL-TMPDU is

echoed including the header of the received SL-TMPDU and up to and including its DATA parameter.

- c) in order to maintain compatibility with existing UTs which do not recognize the "REP" command and always apply b) above, a LT which makes use of "REP" shall always be able of also accepting the receipt of a simple echo.

7.3.4.2.2 Reporting

The user data shall contain a RP-SL-TMPDU, encoded according to rules given in 8.2 and whose parameters are built as follows :

RECEIVED-SP : Contains a code identifying the received SP, according to table 2.

RECEIVED-PAR : This parameter contains a list of TLV encoded values, each of which represent a parameter of the received service primitive. The list which is to be included shall be as specified by table 1, using the entry designating the Service Primitive. All parameters listed in the entry shall be present and in the same order. No extra parameter can be included, even with a null length. For each parameter, the length shall comply with the constraint given by table 4. If a parameter is not given in the local implementation of the SP but is listed in table 1, the UT shall in any case include it in the RP-SL-TMPDU (i.e. the UT shall be able to generate a valid value for this parameter). Similarly, if the local implementation of the SP includes additional parameters, these parameters shall not be reported.