TRILLIUM®

PSF - ISUP (FT/HA)

Software Test Sample

1094146 1.2

PSF - ISUP (FT/HA)

Software Test Sample
1094146 1.2

Trillium Digital Systems, Inc.
12100 Wilshire Blvd., Suite 1800
Los Angeles, CA 90025-7118
Phone: +1 (310) 442-9222
Fax: +1 (310) 442-1162
Web: http://www.trillium.com

PSF - ISUP (FT/HA)
Software Test Sample
1094146 1.2

Information furnished herein by Trillium Digital Systems, Inc., is believed to be accurate and reliable. However, Trillium assumes no liability for errors that may appear in this document, or for liability otherwise arising from the application or use of any such information or for any infringement of patents or other intellectual property rights owned by third parties which may result from such application or use. The products, their specifications, and the information appearing in this document are subject to change without notice.

To the extent this document contains information related to software products you have not licensed from Trillium, you may only apply or use such information to evaluate the future licensing of those products from Trillium. You should determine whether or not the information contained herein relates to products licensed by you from Trillium prior to any application or use.

Printed in U.S.A.

Preface

Objective

This document describes the software test sample for the PSF - ISUP (FT/HA) software (p/n 1000146) designed by Trillium Digital Systems, Inc.

Audience

Trillium assumes that the readers of this document are familiar with telecommunication protocols, specifically SS7 and Trillium's ISUP, PSIF - ISUP, and Fault-Tolerant/High Availability Core Products.

Document Organization

This document is organized into the following sections:

Section	Description
1 Introduction	Provides information about the software tests and their usefulness
2 Overview	Describes the objectives of the tests
3 Test Environment	Describes the protocol stack architecture for tests. A diagram illustrates the architecture and the files used.
4 File Descriptions	Provides a detailed description of acceptance test files, product files, system services files, configuration files, and test files
5 Sample Configuration	Describes the sample PSF configuration used for the tests
6 Test Description	Describes the testing sequence

Document Set

The suggested reading order of this document set is:

1. *Functional Specification*

   Contains the features and highlights that describe the protocol and system characteristics. It includes the memory characteristics and conformance details.

2. *Training Course*

   Offers a detailed overview of the features and interfaces of the software. It contains code samples, data flow diagrams, and a list of files.

3. *Service Definition*

   Describes the procedures and layer manager interface used to pass information between the software and other software elements. The Interface Primitives section describes the services of the software. The Interface Procedures section describes and illustrates the flow of primitives and messages across the interfaces.

   Note: *Information on porting the software is contained in the Service Definition.*

4. *Software Test Sample*

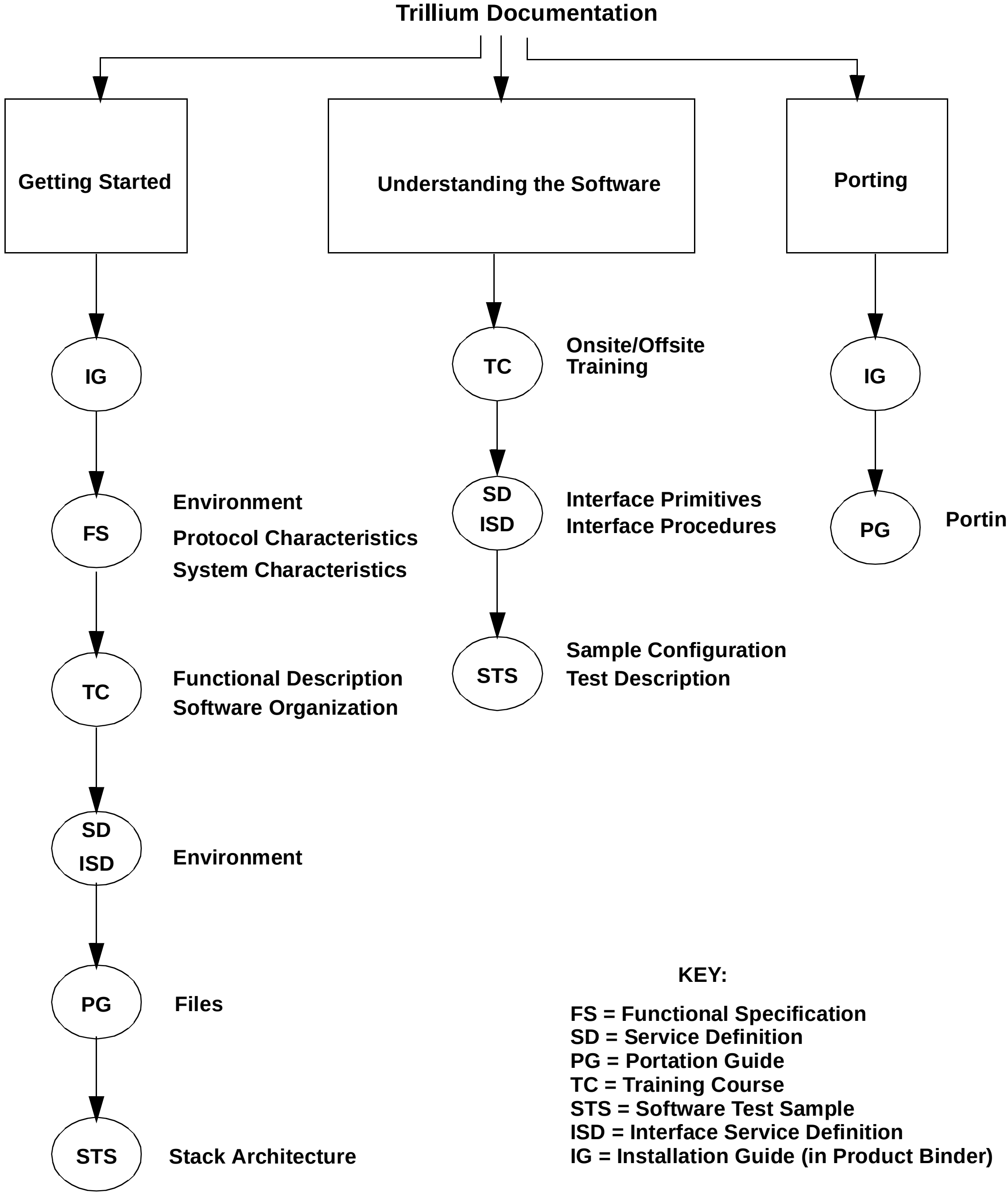
   Describes the sample files delivered with the product and the procedures to build a sample test. This test partially demonstrates the product initialization, configuration, and execution. It may contain data flow diagrams illustrating the correct operation of the software.

In addition to the above PSF documents, the following documents should also be read for a better understanding of the fault-tolerant system:

1. *Fault-Tolerant/High-Availability (FT/HA) Core Functional Specification*
2. *Fault-Tolerant/High-Availability (FT/HA) Core Service Definition*

Using Trillium Documentation

The figure below illustrates the various approaches the user can take when utilizing the software documentation. First time users should read the documents under the **Getting Started** column; important sections and subsections are listed to the right of each document. For users familiar with the documentation but who need to look up certain points concerning the use of the software, the **Understanding the Software** column is suggested. The **Porting** column is for those users who are familiar with Trillium software and related telecommunications protocols and who wish to install the software immediately onto their operating systems.



Notations

This table displays the notations used in this document:

Notation	Explanation	Examples
Arial	**Titles**	**1.1 Title**
Palatino	Body text	This is body text.
Bold	**Highlights information**	**Loose coupling, tight coupling**, **upper layer interface**
ALL CAPS	CONDITIONS, MESSAGES	AND, OR CONNECT ACK
Italics	*Document names, emphasis*	*PSF - ISUP (FT/HA) Software Test Sample* This adds *emphasis*.

`Courier New Bold`	**`Code`**
	`Filenames, pathnames`
		`PUBLIC S16 ZiMiLziCfgReq(pst, cfg)`
		`Pst          *pst;`
		`CmPFthaMngmt   *cfg;`

Release History

This table lists the history of changes in successive revisions to this document:

Version	Date	Initials	Description
1.2	December 31, 1999	sk	Changes for software release 1.2, including: • Addition of multiple point code support • Addition of NTT and Bellcore variants • FT/HA support for PSIF - ISUP
1.1	November 16, 1998	rs	• Initial release

Contents

Illustrations

1 INTRODUCTION

This document describes the software test sample for the PSF - ISUP portable software (p/n 1000146) designed by Trillium Digital Systems, Inc.

The software test sample is a collection of files, delivered with the released product, written and used by Trillium to configure and test the portable software. All the files in this test sample conform to Trillium Advanced Portability Architecture (TAPA). The test is hardware-independent.

The test files outline the interfaces with PSF - ISUP. In particular, they describe the management and system services interfaces for configuring and initializing the layer. As a result, the files are useful for porting the software.

To better understand these test files and the test organization, you should be familiar with Trillium's software architecture, which is fully described in the *PSF - ISUP (FT/HA) Functional Specification* and *PSF - ISUP (FT/HA) Service Definition.*

1.1 Terms and Abbreviations

The following abbreviations are used in this document:

Name	**Description**
ANSI	American National Standards Institute
CC	Call Control (ISUP application)
CFG	Configuration
CST	Controlled status change (of layer)
CSW	Controlled Switchover
ETSI	European Telecommunications Standards Institute
FT/HA	Fault-Tolerant/High-Availability
FTZ	Forschungs-und Technologiezentrum der Deutschen Telekom (Research and Technology Center of the German Telecom (PTT))
ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
ITU	International Telecommunications Union
LM	Layer Manager
MOS	Multiprocessor Operating System
MTP	Message Transfer Part
OOS	Out-Of-Service
PSF	Protocol Specific Function
PSIF	Protocol Specific Interface Function

Name	Description
RT	Run Time
SAP	Service Access Point
SCCP	Signalling Connection Control Part
SD	Shut Down
SS	System Services
SS7	Signalling System 7
TAPA	Trillium Advanced Portability Architecture
WS	Warmstart

2 OVERVIEW

The objectives of this sample test are as follows:

- To demonstrate the robustness of the PSF - ISUP software—that is, the absence of software problems—by running an extensive test suite designed to test the PSF - ISUP functionality
- To demonstrate conformance as indicated in the *PSF - ISUP Functional Specification*
- To demonstrate how PSF - ISUP can be configured and exercised
- To demonstrate the portation work required

The acceptance test does not measure the performance of the PSF - ISUP software. Since the software under test is portable, any performance issues will depend on the environment.

The files that are provided as part of the software test sample for PSF - ISUP have been used to run tests on the product as part of Trillium's in-house software testing. These sample files, which can be found on the CD ROM, are not officially released code and, therefore, are not supported. These files are distributed so that potential users can become familiar with the use of the portable software.

3 TEST ENVIRONMENT

The sample files are portable C files, as are all Trillium portable software deliverables. The sample files do not need special hardware to run. Once portation is done for PSF - ISUP, the sample files will run without further effort.

In order for the tests to run, the sample files should be linked with both the ISUP and PSF - ISUP software, as well as a system services provider. Within Trillium, the system services provider used is the Multiprocessing Operating System (MOS) product.

3.1 Testing Strategy

Figure 3-1 illustrates the testing strategy for the PSF - ISUP software:

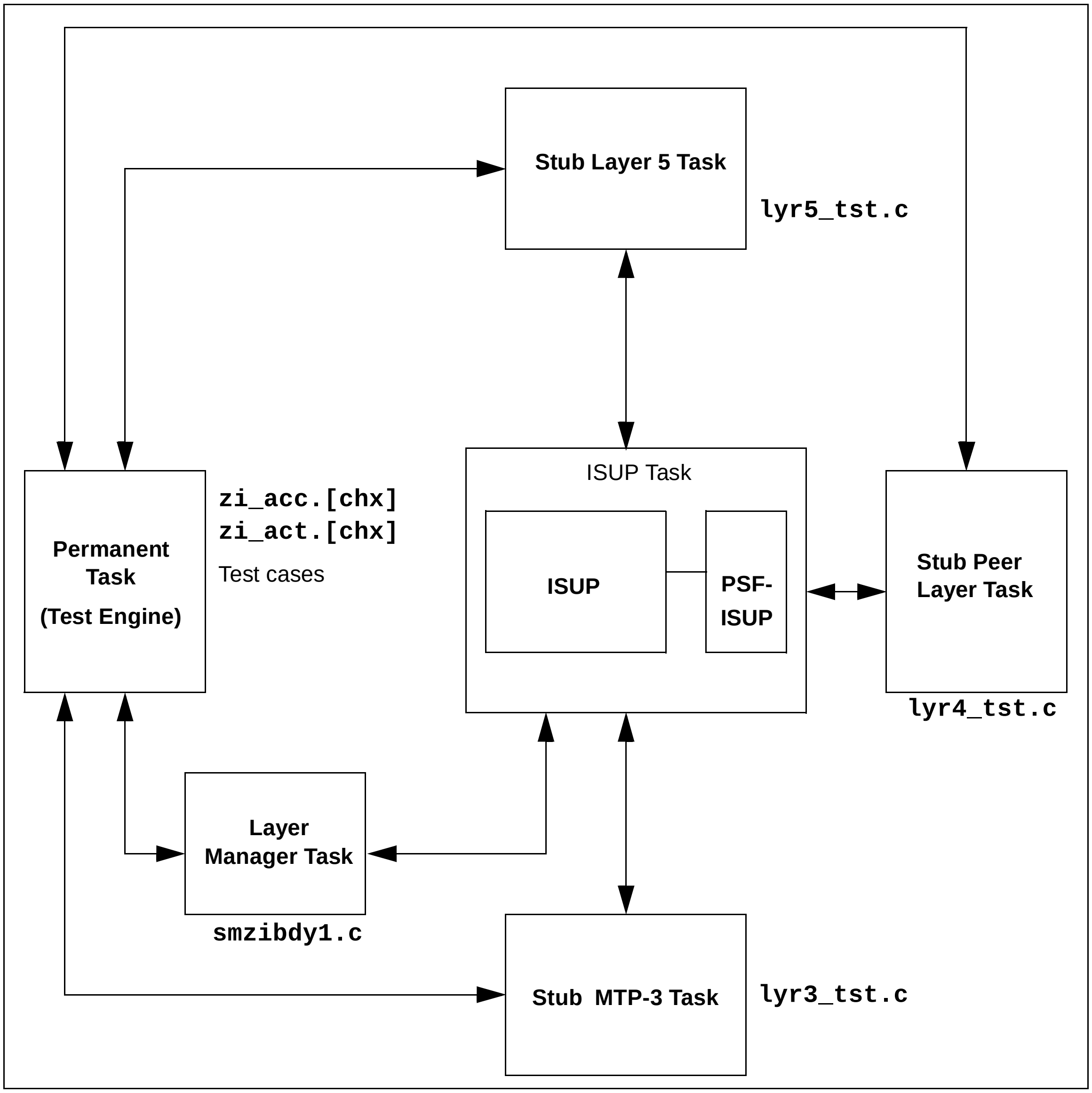


Figure 3-1: Testing strategy

Within Figure 3-1, ISUP is linked with a stub layer 5 task, which emulates the Call Control (CC), and a stub layer 3 task, which emulates the network layer (MTP Level 3). ISUP is also linked with a stub stack manager task. PSF - ISUP is linked with a stub peer layer task, which emulates the peer PSF - ISUP.

A permanent task is also registered with system services apart from ISUP, stub layer 5, stub layer 3, stub stack manager, and stub peer layer. The permanent task drives the test engine. The permanent task is a low-priority task; system services schedules the permanent task when no other task is to be scheduled. Only the permanent task injects events (for example, configuration request from the stack manager, data indication from layer 3, or data request from layer 5) into ISUP and PSF - ISUP through the proper stub.

In the case of loose coupling, scheduled stub tasks call a function to give control to the test engine to process the received output. The test cases are the data for the test engine.

The PSF functionality involves state updates from active to standby. Some of the tests require both active and standby PSF - ISUP to be present; however, only a single instance of PSF - ISUP can run at one time in the test environment. The active and standby status is emulated by the same instance of ISUP and PSF - ISUP by changing their status (to active or standby) as required by various steps in the test. For example, when the active PSF - ISUP sends state update messages, the messages are queued by the stub peer-layer task. If the test requires the behavior of the standby to be tested, then the status of PSF - ISUP is changed from active to standby and the queued update messages are applied to the standby.

4 FILE DESCRIPTIONS

The **envopt.acc** delivered with the PSF - ISUP software contains the settings required to build the acceptance test. The **envopt.acc** file must be renamed to **envopt.h** in order to compile the software test sample. The **zi.mak** file compiles the source files required for PSF - ISUP testing.

A detailed description of these files and their functions can be found in the appropriate product *Service Definition*.

4.1 System Services Files

These files provide the system services interface. The files used depend on the operating system in use. If Trillium MOS is used, the following files are needed, along with the relevant header files. These files are needed for all test configurations and are independent of the particular configuration being run.

File Name	Description
ms_bdy1.c	MOS - Body - Part 1
ms_bdy2.c	MOS - Body - Part 2
ms_bdy5.c	MOS - Body - Part 5
ms_cfg.c	MOS - Configuration
ms_id.c	MOS - ID
ms_ex_ms.c	MOS - External interface
ms_ptmi.c	MOS - Portable management interface
cm_ss.c	Common system services functions

4.2 ISUP Files

The following files provide the ISUP functionality and are described in the *ISUP Portation Guide*:

File Name	Description
si_id.c	ISUP - ID
si_bdy1.c	ISUP - Body - Part 1
si_bdy2.c	ISUP - Body - Part 2
si_bdy3.c	ISUP - Body - Part 3
si_bdy4.c	ISUP - Body - Part 4
si_bdy5.c	ISUP - Body - Part 5
si_bdy6.c	ISUP - Body - Part 6

File Name	Description
`si_bdy7.c`	ISUP - Body - Part 7
`si_ptli.c`	ISUP - Portable lower interface
`si_ptui.c`	ISUP - Portable upper interface
`si_ptmi.c`	ISUP - Portable management interface
`si_ex_ms.c`	ISUP - External interface

4.3 PSF - ISUP Product Files

The following files provide the PSF - ISUP functionality and are described in the *PSF - ISUP (FT/HA) Service Definition*:

File Name	Description
`zi_id.c`	PSF - ISUP - ID
`zi_bdy1.c`	PSF - ISUP - Body - Part 1
`zi_bdy2.c`	PSF - ISUP - Body - Part 2
`zi_bdy3.c`	PSF - ISUP - Body - Part 3
`zi_bdy4.c`	PSF - ISUP - Body - Part 4
`zi_ptpi.c`	PSF - ISUP - Portable upper interface
`zi_ptmi.c`	PSF - ISUP - Portable management interface
`zi_ex_ms.c`	PSF - ISUP - External interface

4.4 Common Files

The following files provide the common functionality to all Trillium products:

File Name	Heading
`cm_hash.c`	Common hashing functions
`cm_lib.c`	Common utility functions
`cm_bdy5.c`	Common timer functions
`cm_gen.c`	Common packing / unpacking functions
`cm_pftha.c`	Common packing / unpacking functions for PSF - ISUP
`lsi.c`	Common packing / unpacking functions at the LSI interface
`sit.c`	Common packing / unpacking functions at the SIT interface

4.5 Stub Layer 5 Files

The following files provide the stub layer 5 functionality. These files are needed for all test configurations.

File Name	Description
`lyr5_tst.c`	Layer 5 - Stub
`l5_ptli.c`	Layer 5 - Portable lower interface
`l5_ex_ms.c`	Layer 5 - External interface

The **`layer5.c`** file contains all the incoming primitives to layer 5. All stubs call a common function, **`zixtopchk,`** when an event is received.

4.6 Stub Layer 3 Files

The following files provide the stub layer 3 functionality. These files are needed for all test configurations.

File Name	Description
`lyr3_tst.c`	Layer 3 - Stub
`l3_ptui.c`	Layer 3 - Portable upper interface
`l3_ex_ms.c`	Layer 3 - External interface

The **`layer3.c`** file contains all the incoming primitives to layer 3. All stubs in this file call a common function, **`zixtopchk`**, to give the test engine the control to check its fields.

4.7 Stub Peer Layer Files

The following file provides the stub functionality for the peer layer. This file is needed for all test configurations.

File Name	Description
`lyr4_tst.c`	Peer layer stub

4.8 Sample Layer Manager Files

The following files provide the layer manager functionality. These files passively terminate the layer manager primitives invoked by PSF - ISUP.

Common Layer Manager Files:

File Name	Description
`sm_bdy1.c`	Common layer manager - Body - Part 1
`sm_ex_ms.c`	Common layer manager - External interface

ISUP Layer Manager Files:

File Name	Description
`smsibdy1.c`	ISUP layer manager - Body - Part 1
`smsiptmi.c`	ISUP layer manager - Portable management interface
`smsiexms.c`	ISUP layer manager - External interface

PSF - ISUP Layer Manager Files:

File Name	Description
`smzibdy1.c`	PSF - ISUP layer manager - Body - Part 1
`smziptmi.c`	PSF - ISUP layer manager - Portable management interface
`smziexms.c`	PSF - ISUP layer manager - External interface

4.9 Test Files

The test files provide the core test and configuration functionality. They contain the main test driver script and a set of tests described in Section 6, "TEST DESCRIPTION."

The following table contains the configuration and test file names and their descriptions:

Name	Description
si_acc.h	Contains the ISUP-related defines for the main test driver
si_acc.x	Contains the ISUP-related **typedef**s, variables, and prototypes for the main test driver
si_acc.c	Contains the following important function: • **tst** - The entry point into the test driver from system services. This function registers initialization and activation tasks and initializes global variables.
zi_acc.h	Contains the PSF - ISUP-related defines for the main test driver
zi_acc.x	Contains the PSF - ISUP-related **typedef**s, variables, and prototypes for the main test driver
zi_acc.c	Provides the test engine. This file contains the following important functions: • **zixtperm** - The entry point function of the permanent task. It runs the tests for the specified group(s) on the command line • **tst** - The initialization function for the test engine and information related to the test cases
zi_act.c	Provides the data for the test engine. The test engine (**zi_acc.c**) views information in this file as its data to operate on.
zi_act.h	Contains defines used in the corresponding **.c** file. These defines are related to the test cases.
zi_act.x	Contains **typedef**s and external declaration for the test cases

5 SAMPLE CONFIGURATION

The following configuration is needed for ISUP and PSF - ISUP to run the tests. The configuration is found in the `si_acc.c` and `zi_acc.c` files for ISUP and PSF - ISUP, respectively. Each test requires ISUP to be configured before PSF - ISUP is configured.

Figure 5-1 illustrates the PSF - ISUP software environment:

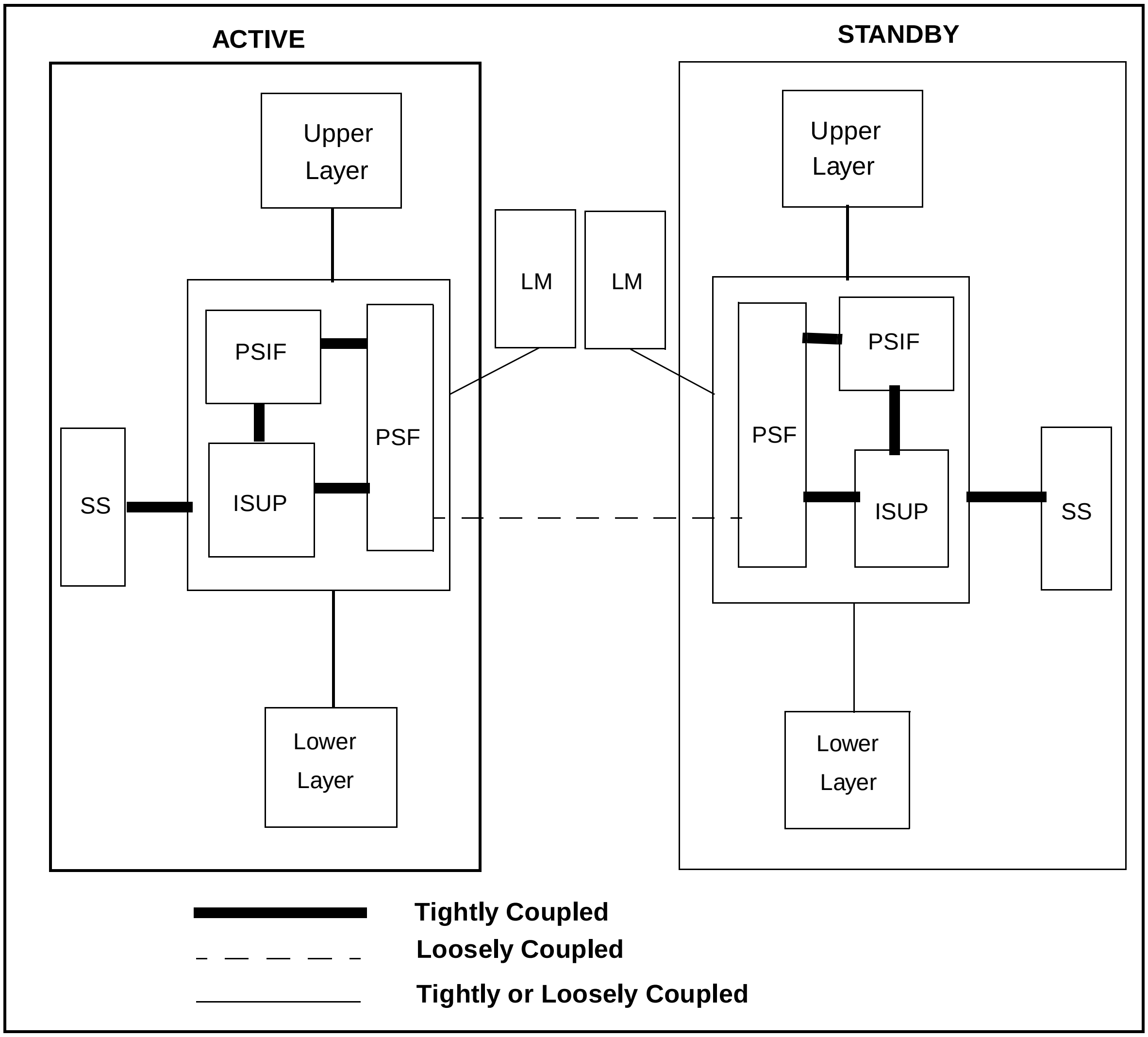


Figure 5-1: PSF - ISUP environment

5.1 ISUP Configuration

For information on the general configuration of ISUP, see the *ISUP Software Test Sample*.

5.2 PSF - ISUP Configuration

PSF - ISUP is configured after ISUP. The following sections describe the elements that must be configured in PSF - ISUP.

5.2.1 General Configuration

General configuration of PSF - ISUP is required before proceeding to any test requiring the peer SAP. Timer resolution, virtual processor ID, **mem**, and **post** structure of the stack manager are included in the general configuration.

5.2.2 Peer SAP Configuration

One peer SAP is configured for each test. Refer to the **cfgZnPeerSAP** function in **zi_acc.c**. Peer SAP configuration includes selector, maximum size of the update message, and the value of the update completion timer.

6 TEST DESCRIPTION

The test is activated when the system services invokes the **tst** function within the **zi_acc.c** file. The scope of one test case is one scheduling (or a timer expiry); multiple test cases are linked together to make a test scenario. For example, testing an incoming successful call is a test scenario consisting of multiple test cases (from configuration to releasing the call). One test case in this test scenario tests the behavior of the ISUP layer on receiving IAM from the peer.

Similarly, multiple test scenarios are categorized into test groups. A test group is a logical classification of the test scenarios. For example, all test scenarios related to configuration are categorized in a single group.

6.1 Test Preparation

The **zi.mak** sample file compiles ISUP and PSF - ISUP code to generate the **zi_acc** executable. To run **zi_acc**, the number of the group to be tested must be specified on the command line. The following group number options are available:

```
zi_acc [-g <GroupNumber/.> [-s <ScenarioNumber/.> [-a] ]]
[-t <Basetimeout>] [-e ] [-c 0] [-v <verbose> ]
```

Group Number:

The following groups are independently accessible:

Group Number	Description
0	All test groups
1	PSF - ISUP configuration-related
2	PSF - ISUP status LM interface
3	Different sequence of the control request
4	Different sequences for run-time update
5	Different sequence of the warmstart
6	Different sequence of the control switchover
7	Tests new interfaces and scenarios related to primitive loss

Scenario Number:

This is the index of the scenario in a test group.

Base Time-out:

This is a factor that is multiplied with all internal timers running for the test engine. Default value of the base time-out is 10.

Verbose Option:

Option	Description
1	Basic heading for test groups and test scenarios along with the result of the test scenario
2	Details of 1st and a description of the test case
3	Details of 2nd and a description of the input/output in a test case
4	Graphical representation of the test case in the form of a message sequence. This is the default option.

Examples:

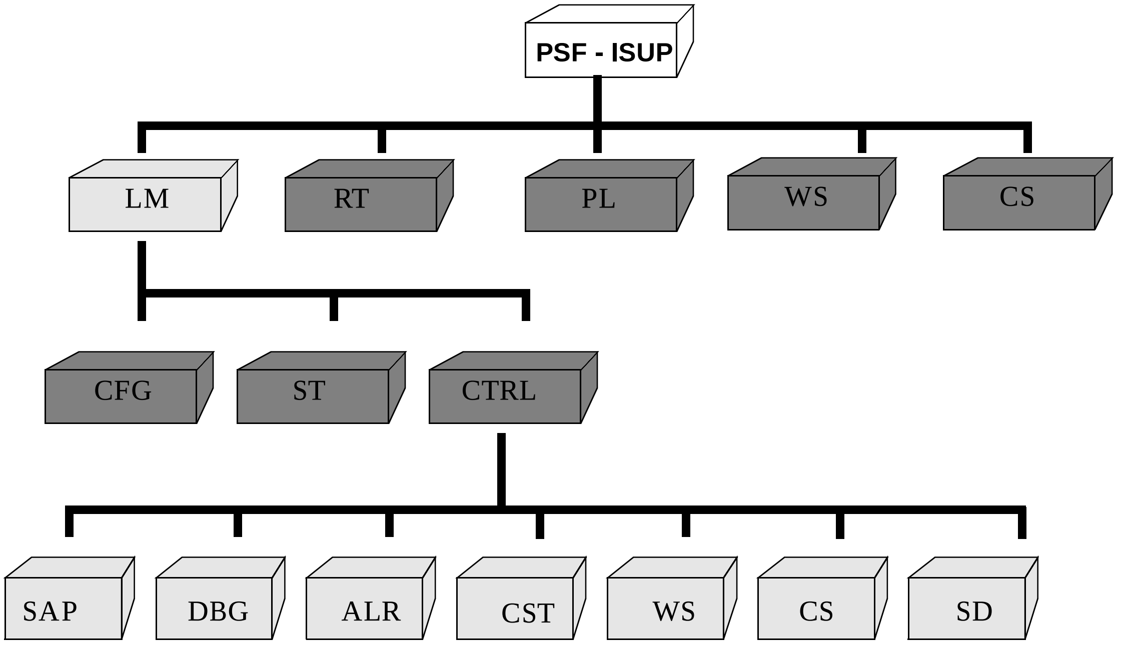
1. To run all test groups and test scenarios: **`zi_acc`**
2. To run test cases in the first group only: **`zi_acc -g 1`**
3. To run the first scenario in test group 4: **`zi_acc -g 4 -s 1`**
4. To run the first scenario in test group 4 with minimum delay: **`zi_acc -g 4 -s 1 -t 1`**
5. To get a listing of all test groups and test scenarios: **`zi_acc -g . -s`**
6. To get a listing of all test cases in a specific test group (in this example, test group 5): **`zi_acc -g 5 -s`**
7. To get a listing of all test cases in all scenarios in all test groups: **`zi_acc -g . -s . -a`**
8. To explore all possible options for running the PSF - ISUP tests: **`zi_acc -h`**
9. To run the given (or all) test group[s] (or a specific test scenario) in loosely coupled mode: **`zi_acc -c 0`**
10. To run the given (or all) test group[s] (or a specific test scenario) in a loop: **`zi_acc -e`**

The **`-c 0`** option configures the selector values for a loosely coupled interface. Proper compile options are required to run in loosely coupled mode. Tests will run in tightly coupled mode by default and will stop after completing the requested test groups (or test scenarios)

Figure 6-1 illustrates the test group hierarchy.

6.2 Test Case Organization

This section describes the test cases that test the PSF - ISUP functionality. In order to have efficient test coverage, the test cases are organized in various groups and subgroups, represented in hierarchal form in Figure 6-1:



□ : Test groups that can not be run individually

▨ : Test groups that can be run individually from the command line

LM	:	Layer Manager
RT	:	Real Time
PL	:	Primitive Loss
WS	:	Warmstart
CS	:	Controlled Switchover
CFG	:	Configuration
ST	:	Status
CTRL	:	Control request related test group
SAP	:	Control request related to SAPs
DBG	:	Debug classes
ALR	:	Alarms
CST	:	Status change
SD	:	Shutdown

Figure 6-1: Test group hierarchy

6.3 Assumptions

Following is a list of assumptions made for each test case description:

- Each step of a test procedure is independent; the test engine performs only one step at a time. After executing a step, the test should deschedule itself for a sufficient period of time before performing the next step in the sequence (if required). This ensures that all the messages resulting from the first step reach the destination entities and that the destination entities have operated on them before the next step is performed.
- Before starting the execution of a test, the ISUP protocol layer and the PSF - ISUP reset to the same initial state that existed before configuration.
- Testing strategy assumes that the permanent task (that is, the test engine) is scheduled after all other layers (ISUP and stub layers) have completed their processing. Running the test engine, therefore, has the lowest priority. Any violation in this assumption can result in incorrect results for the tests.
- Once scheduled, the test engine (or the permanent task) should not be pre-empted by any stub layer or by ISUP prior to giving input for the test case. In order to minimize the effects of pre-emption, timers are run between two test cases and after the clean-up action of the test scenarios.

6.4 Test Configurations

The ISUP tests use the configuration illustrated in Figure 6-2:

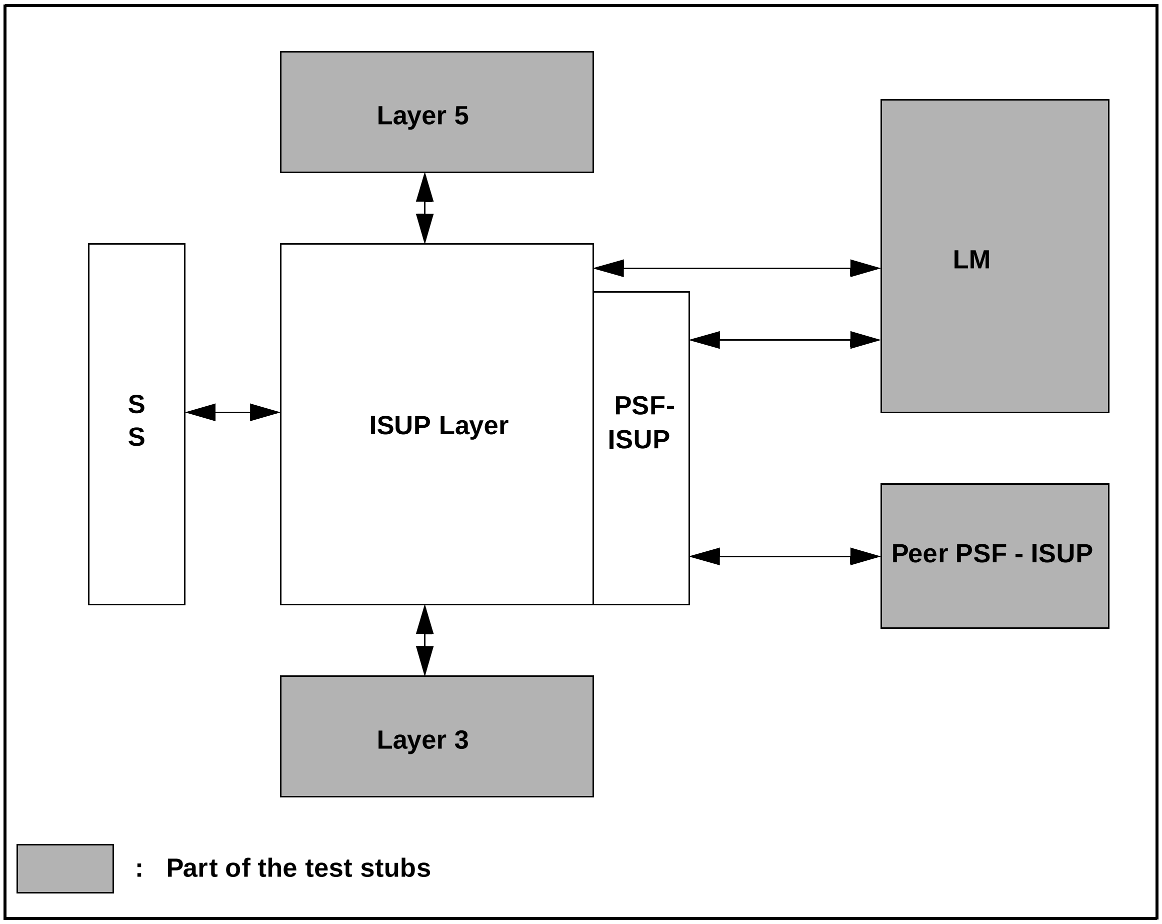


Figure 6-2: Test engine configuration

The test stubs simulate the lower layer, upper layer, layer manager interface, and peer PSF - ISUP around the ISUP layer under test. Figure 6-2 illustrates interaction of the test stub parts with the ISUP layer. The test stub around the ISUP + PSF - ISUP layer interacts with the test engine to verify the received event and to generate the next input.

This configuration is independent of the status of the layer—that is, the layer can be in-service, standby, or out-of-service. The peer PSF - ISUP part of the test engine handles update messages coming from the ISUP layer for run time, warmstart, or controlled switchover.

6.5 Test Case Descriptions

This section describes the test cases required to test the functionality of PSF - ISUP. The test cases are grouped according to function. Each group may or may not have further subgroups.

Test ID	Test Group	Description
1.1.1	Layer manager - configuration	Successful general configuration
1.1.2	Layer manager - configuration	Successful peer SAP configuration
1.1.3	Layer manager - configuration	Invalid configuration - PSF - ISUP configuration before ISUP configuration
1.2.1	Status inquiry	Successful status query for status of a circuit
1.2.2	Status inquiry	Successful status query for status of a circuit group
1.2.3	Status inquiry	Successful status query for status of an interface
1.2.4	Status inquiry	Invalid status query for circuit
1.2.5	Status inquiry	Successful status query for status of the layer
1.2.6	Status inquiry	Successful status query for status of the peer SAP
1.2.7	Status inquiry	Successful status query for the system ID
1.3.1.1	Control request	Control request to enable the lower SAP
1.3.1.2	Control request	Control request to enable the group of lower SAPs
1.3.1.3	Control request	Control request to disable the lower SAP
1.3.1.4	Control request	Control request to disable the group of lower SAPs
1.3.1.5	Control request	Control request to disable the upper SAP
1.3.1.6	Control request	Control request to disable the group of upper SAPs
1.3.1.7	Control request	Control request to standby to disable the upper SAP
1.3.1.8	Control request	Control request with invalid **elmntId**
1.3.1.9	Control request	Control request with invalid action
1.3.1.10	Control request	Control request to disable an upper SAP with invalid **sapId**
1.3.1.11	Control request	Control request to standby to disable an upper SAP
1.3.2.1	Debug	Debug print control enable
1.3.2.2	Debug	Debug print control disable
1.3.3.1	Alarm	Alarm generation control
1.3.3.2	Alarm	Alarm generation on getting erroneous update message
1.3.4.1	Status change	Active to active with peer disabled

Test ID	Test Group	Description
1.3.4.2	Status change	Active doing WS to active with peer disabled
1.3.4.3	Status change	Standby layer to active with peer disabled
1.3.4.4	Status change	OOS layer to active with peer disabled
1.3.4.5	Status change	Active doing WS to active with peer enabled
1.3.4.6	Status change	Standby layer to active with peer enabled
1.3.4.7	Status change	OOS layer to active with peer enabled
1.3.4.8	Status change	Active layer to standby
1.3.4.9	Status change	Active layer doing WS to standby
1.3.4.10	Status change	OOS layer to standby
1.3.4.11	Status change	Disable peer interface to active node
1.3.4.12	Status change	Disable peer interface to standby node
1.3.4.13	Status change	Disable peer interface to OOS node
1.3.4.14	Status change	Abort interface to active node
1.3.4.15	Status change	Abort interface to standby node
1.3.4.16	Status change	Abort interface to OOS node
1.3.4.17	Status change	Invalid status change
1.3.5.1	Warmstart	Warmstart peer request to active with peer enabled
1.3.5.2	Warmstart	Warmstart peer request to active with peer disabled
1.3.5.3	Warmstart	Warmstart peer request to active currently doing WS
1.3.5.4	Warmstart	Warmstart peer request to standby
1.3.5.5	Warmstart	Warmstart peer request to OOS
1.3.5.6	Warmstart	Disable peer SAP request while warmstart is going on
1.3.5.7	Warmstart	Warmstart abort
1.3.6.1	CSW	CSW request to active PSF - ISUP with peer enabled
1.3.6.2	CSW	CSW request to active PSF - ISUP with peer disabled
1.3.6.3	CSW	CSW request to active PSF - ISUP currently doing CSW
1.3.6.4	CSW	CSW request to active PSF - ISUP currently doing WS
1.3.6.5	CSW	CSW request to standby PSF - ISUP
1.3.6.6	CSW	CSW request to OOS PSF - ISUP
1.3.6.7	CSW	CSW abort
1.3.7.1	Shutdown	Shutdown request to active layer
1.3.7.2	Shutdown	Shutdown request to active layer doing WS

Test ID	Test Group	Description
1.3.7.3	Shutdown	Shutdown request to active layer doing CSW
1.3.7.4	Shutdown	Shutdown request to standby layer
1.3.7.5	Shutdown	Shutdown request to OOS layer
2.1	Run-time update	Run-time update of the upper SAP control block: bind request
2.2	Run-time update	Run-time update of the upper SAP control block: control request to disable the upper SAP
2.3	Run-time update	Run-time update: control request to disable the group of upper SAPs
2.4	Run-time update	Run-time update of the lower SAP control block: bind request
2.5	Run-time update	Run-time update of the lower SAP control block: control request to disable the lower SAP
2.6	Run-time update	Run-time update: control request to disable the group of lower SAPs
2.7	Run-time update	Run-time update of the circuit control block on circuit-related event (idle to remotely blocked)
2.8	Run-time update	Run-time update of the circuit control block on circuit-related event (remotely blocked to idle)
2.9	Run-time update	Run-time update of the circuit control block on circuit-related event (idle to locally blocked)
2.10	Run-time update	Run-time update of the circuit control block on circuit-related event (locally blocked to idle)
2.11	Run-time update	Run-time update of the circuit control block on circuit-related event (locally blocked to locally and remotely blocked)
2.12	Run-time update	Run-time update of the circuit control block on circuit group-related event
2.13	Run-time update	Run-time update of the circuit control block on circuit group-related event (remotely blocked to idle)
2.14	Run-time update	Run-time update of the circuit control block on circuit group-related event (idle to locally blocked)
2.15	Run-time update	Run-time update of the circuit control block on circuit group-related event (locally blocked to idle)
2.16	Run-time update	Run-time update of the circuit control block on circuit group-related event (locally blocked to locally and remotely blocked)

Test ID	Test Group	Description
2.17	Run-time update	Run-time update of the connection control block on getting connection-related events (idle to active)
2.18	Run-time update	Run-time update of the connection control block on getting connection-related events (active to idle)
2.19	Run-time update	Out-of-sequence run-time update message
2.20	Run-time update	Run-time update message with invalid table type
2.21	Run-time update	Handling of event from the upper interface by standby layer
2.22	Run-time update	Handling of event from the lower interface by standby layer
2.23	Run-time update	No run-time update if peer SAP is disabled
2.24	Run-time update	Standby fallbacks the information if required
3.1	Primitive loss/new primitive	ISUP sends bind confirm to the upper layer
3.2	Primitive loss/new primitive	Multiple bind requests are accepted by ISUP layer
3.3	Primitive loss/new primitive	ISUP repeats bind request to MTP Level 3 if bind confirmation is delayed
3.4	Primitive loss/new primitive	ISUP repeats bind request to MTP Level 3 if bind confirm is delayed, and after a certain number of retries sends an alarm to LM and stops trying
3.5	Primitive loss/new primitive	ISUP repeats **RelInd** to the upper layer if **RelRsp** is delayed or lost
3.6	Primitive loss/new primitive	ISUP repeats **RelInd** to the upper layer if **RelRsp** is delayed or lost and sends an alarm to LM after a certain number of retries
3.7	Primitive loss/new primitive	ISUP responds with status of the point code and congestion on receiving **ptCdStaReq** from CC
3.8	Primitive loss/new primitive	ISUP requests the status of the point code and congestion on getting **SiLiSntStaInd** from MTP Level 3 and does not expect confirmation on receiving the resume indication
3.9	Primitive loss/new primitive	ISUP requests the status of the point code and congestion on receiving **SiLiSntStaInd** from MTP Level 3 and on receiving confirmation with status as **SN_RESUME**
3.10	Primitive loss/new primitive	ISUP repeats **RelInd** to the upper layer if **RelRsp** is delayed or lost (call is released from the peer)
4.1	Warmstart	Warmstart update of the upper SAP control block

Test ID	Test Group	Description
4.2	Warmstart	Warmstart update of the lower SAP control block
4.3	Warmstart	Warmstart update of the circuit control block
4.4	Warmstart	Warmstart update of the connection control block
4.5	Warmstart	Abort warmstart
5.1	Control switchover	CS update of the circuit control block
5.2	Control switchover	CS update of the connection control block
5.3	Control switchover	CS update of the circuit group control block

6.5.1 Test Group 1 Layer Manager (LM)

The layer manager test group contains the test cases that test the PSF - ISUP functionality at the layer manager interface. For control and configuration requests, unless otherwise stated, a successful confirmation from the layer manager means the status field contains **LCM_PRIM_OK**. In case of failure, this field contains **LCM_PRIM_NOK**.

The subgroups contained in this test group are described in the following sections.

6.5.1.1 Test Group 1.1 Configuration (CFG)

This subgroup tests the configuration interface of PSF - ISUP. It checks the responses when the system manager sends configuration requests to PSF - ISUP. The test cases in this group are described in detail below:

Test ID 1.1.1

Name:

Successful General Configuration

Purpose:

This test checks that when the layer manager sends a valid configuration request to PSF - ISUP, PSF - ISUP indicates success to the layer manager.

Pre-Test Conditions:

ISUP task created on an OOS node

Test Description:

Once the ISUP task is created, it is ready to accept configuration from the layer manager. The steps required to carry out the test are:

1. Configure ISUP
2. Send a general configuration request to PSF - ISUP with the following fields specified:
    - **`msgType`** as **`TCFG`**
    - **`hdr.entId`** with entity ID of ISUP
    - **`hdr.inst`** with instance of ISUP
    - **`hdr.elmnt.elmnt`** with **`STGEN`** for general configuration
    - **`hdr.transId`** with the number that should be received in confirmation of the configuration
    - **`hdr.response.selector`** to indicate the selector to be used for confirmation
    - **`hdr.response.route/prior`** with route / priority for confirmation of the configuration request
    - **`hdr.response.mem.pool/region`** with memory pool / region for the confirmation
    - **`cmPFthaGenCfg`**, with **`timerRes`**, **`vProcId`**, and **`mem`** filled in for configuration
    - **`smPst`** filled in the general configuration structure
3. Check that positive confirmation is received in the configuration confirm from PSF - ISUP with the same transaction ID as the one sent to PSF - ISUP in the configuration request

Test ID 1.1.2

Name:

Successful Peer SAP Configuration

Purpose:

This test checks that when the layer manager sends a valid configuration request for the peer SAP to PSF - ISUP, PSF - ISUP indicates success to the layer manager.

Pre-Test Conditions:

ISUP task created on an OOS node

Test Description:

Once the ISUP task is created, it is ready to accept configuration from the layer manager. The steps required to carry out the test are:

1. Perform general configuration for the ISUP layer
2. Send a general configuration request to PSF - ISUP with the following fields specified:
   - **`msgType`** as **`TCFG`**
   - **`hdr.entId`** with entity ID of ISUP
   - **`hdr.inst`** with instance of ISUP
   - **`hdr.elmnt.elmnt`** with **`STPEERSAP`** for configuration
   - **`hdr.transId`** with the number that should be received in confirmation of the configuration
   - **`hdr.response.selector`** to indicate the selector to be used for confirmation
   - **`hdr.response.route/prior`** with the route / priority for confirmation of the configuration request
   - **`hdr.response.mem.pool/region`** with the memory pool / region for the confirmation
   - **`cmPFthaSAPCfg`**, with **`region`**, **`pool`**, **`dstProcId`**, **`dstEnt`**, **`dstInst`**, **`prior`**, **`route`**, and **`selector`** filled
   - **`maxUpdMsgSize,`** with a value greater than the maximum size of a table to be packed by PSF - ISUP
   - **`tUpdCompAck`**, with timer value; enable it by setting the **`enb`** field
3. Check that positive confirmation is received in the configuration confirm from PSF - ISUP with the same transaction ID as the one sent to PSF - ISUP in the configuration request

Test ID 1.1.3

Name:

Invalid Configuration - PSF - ISUP Configuration before ISUP Configuration

Purpose:

This test checks that when the layer manager sends a configuration request to PSF - ISUP without first configuring the ISUP layer, PSF - ISUP indicates failure to the layer manager.

Pre-Test Conditions:

ISUP task created on an OOS node

Test Description:

Once the ISUP task is created, it is ready to accept configuration from the layer manager. The steps required to carry out the test are:

1. Send a configuration request to PSF - ISUP for peer SAP configuration
2. Check that PSF - ISUP sends a confirmation to the layer manager indicating failure with the reason field as **`LCM_REASON_GENCFG_NOT_DONE`**

6.5.1.2 Test Group 1.2 Solicited Status (ST)

This subgroup tests the configuration interface of PSF - ISUP. These tests check the responses of PSF - ISUP when the system manager sends status requests to PSF - ISUP. The test cases in this group are described in detail below:

Test ID 1.2.1

Name:

Successful Status Query for Status of a Circuit

Purpose:

This test checks that when the layer manager sends a valid status request to ISUP, PSF - ISUP answers with the required status information.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

PSF - ISUP handles the status request from the layer manager at all times. The steps required to carry out the test are:

1. Send a status request to the active PSF - ISUP with **`hdr.elmId.elmnt`** as **`STCIR`** and **`msgType`** as **`TSSTA`**. Fill the **`cir`** field associated with the concerned circuit.
2. Check that positive confirmation is received in the status confirm from PSF - ISUP
3. Validate the value returned in the status confirm with the circuit state
4. Repeat the test for the standby ISUP node

Test ID 1.2.2

Name:

Successful Status Query for Status of a Circuit Group

Purpose:

This test checks that when the layer manager sends a valid status request to ISUP, PSF - ISUP answers with the required status information.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

PSF - ISUP handles the status request from the layer manager at all times. The steps required to carry out the test are:

1. Send a status request to the active PSF - ISUP with **`hdr.elmId.elmnt`** as **`SI_STCIRGRP`** and **`msgType`** as **`TSSTA`**. Fill the **`cirgr`** field with information regarding the circuit and range associated with the concerned circuit group.
2. Check that positive confirmation is received in the status confirm from PSF - ISUP
3. Validate the values returned in the status confirm with the states of the circuit group
4. Repeat the test for the standby ISUP node

Test ID 1.2.3

Name:

Successful Status Query for Status of an Interface[1]

Purpose:

This test checks that when the layer manager sends a valid status request to ISUP, PSF - ISUP answers with the required status information.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

PSF - ISUP handles the status request from the layer manager at all times. The steps required to carry out the test are:

1. Send a status request to the active PSF - ISUP with **`hdr.elmId.elmnt`** as **`SISTINTF`** and **`msgType`** as **`TSSTA`**. Fill the **`intfId`** field associated with the concerned interface ID.
2. Check that positive confirmation is received in the status confirm from PSF - ISUP
3. Validate the value returned in the status confirm with the state of the interface
4. Repeat the test for the standby ISUP node

[1] DPC entity at the layer manager interface has been replaced by an *interface* entity, which is a combination of DPC, OPC, ISUP protocol variant type, and network type.

Test ID 1.2.4

Name:

Invalid Status Query for Circuit

Purpose:

This test checks that, if an invalid parameter is sent in the status request from the layer manager, ISUP returns a failure indication.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

PSF - ISUP validates the status request sent by the layer manager. If the status request contains invalid parameters, PSF - ISUP responds with a failure indication. The steps required to carry out the test are:

1. Send a status request to PSF - ISUP with **`hdr.elmId.elmnt`** as **`STICIR`**, but an invalid circuit in the **`cir`** field
2. Check that PSF - ISUP sends a confirmation to the layer manager indicating failure and that the reason for failure is **`LCM_REASON_HASHING_FAILED`**

Test ID 1.2.5

Name:

Successful Status Query for Status of the Layer

Purpose:

This test checks that when the layer manager sends a valid status request to PSF - ISUP, PSF - ISUP answers with the required status information.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

PSF - ISUP handles the status request from the layer manager at all times. The steps required to carry out the test are:

1. Send a configuration request to PSF - ISUP for general configuration with the following fields specified:
   - **`msgType`** as **`TSSTA`**
   - **`hdr.entId`** with entity ID of ISUP
   - **`hdr.inst`** with instance of ISUP
   - **`hdr.elmnt.elmnt`** with **`STGEN`**
   - **`hdr.transId`** with a number that should be received in confirmation of the configuration
   - **`hdr.response.selector`** to indicate the selector to be used for confirm
   - **`hdr.response.route/prior`** with the route/priority for confirmation of the configuration request
   - **`hdr.response.mem.pool/region`** with memory pool/region for the confirmation
   - **`cmPFthaSAPCfg`** with **`region`**, **`pool`**, **`dstProcId`**, **`dstEnt`**, **`dstInst`**, **`prior`**, **`route`**, and **`selector`** filled
2. Check that positive confirmation is received in the status confirm from PSF - ISUP with the same transaction ID. **`genSta`** should be ACTIVE in this case.

Test ID 1.2.6

Name:

Successful Status Query for Status of the Peer SAP

Purpose:

This test checks that when the layer manager sends a valid status request to PSF - ISUP, PSF - ISUP answers with the required status information.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

PSF - ISUP handles the status request from the layer manager at all times. The steps required to carry out the test are:

1. Send a configuration request to PSF - ISUP for general configuration with the following fields specified:
    - **`msgType`** as **`TSSTA`**
    - **`hdr.entId`** with entity ID of ISUP
    - **`hdr.inst`** with instance of ISUP
    - **`hdr.elmnt.elmnt`** with **`STPEERSAP`**
    - **`hdr.transId`** with the number that should be received in confirmation of the configuration
    - **`hdr.response.selector`** to indicate the selector to be used for confirm
    - **`hdr.response.route/prior`** with the route/priority for confirmation of the configuration request
    - **`hdr.response.mem.pool/region`** with memory pool/region for the confirmation
2. Check that positive confirmation is received in the status confirm from the PSF - ISUP with the same transaction ID. **`bndState`** should be **`CMPFTHA_BND`** and **`updState`** should be **`CMPFTHA_IDLE`**.

Test ID 1.2.7

Name:

Successful Status Query for the System ID

Purpose:

This test checks that when the layer manager sends a valid status request to PSF - ISUP, PSF - ISUP answers with the required status information.

Pre-Test Conditions:

ISUP layer is in OOS state.

Test Description:

PSF - ISUP handles the status request from the layer manager at all times. The steps required to carry out the test are:

1. Perform general configuration for the ISUP layer
2. Send a configuration request to PSF - ISUP for general configuration with the following fields specified:
   - **`msgType`** as **`TSSTA`**
   - **`hdr.entId`** with entity ID of ISUP
   - **`hdr.inst`** with instance of ISUP
   - **`hdr.elmnt.elmnt`** with **`STSID`**
   - **`hdr.transId`** with the number that should be received in confirmation of the configuration
   - **`hdr.response.selector`** to indicate the selector to be used for confirm
   - **`hdr.response.route/prior`** with the route/priority for confirmation of the configuration request
   - **`hdr.response.mem.pool/region`** with memory pool/region for the confirmation
3. Check that positive confirmation is received in the status confirm from PSF - ISUP with the same transaction ID. Layer returns with **`mVer`**, **`mRev`**, **`bVer`**, **`bRev`**, and **`partNumber`**.

6.5.1.3 Test Group 1.3 Control

This group tests the control interface of PSF - ISUP. This test group is divided into subgroups, depending upon the distinct functions performed by the control request. Warmstart and controlled switchover subgroups are included in the control group to test the warmstart and controlled switchover from the layer manager interface point-of-view only. Warmstart and controlled switchover are also included as subgroups inside PSF - ISUP (highest level of hierarchy), where they describe the test cases for testing their internal core functionality.

6.5.1.3.1 Test Group 1.3.1 Control Request Related to SAPs

This test group tests the control request related to SAPs for enabling and disabling one or more lower or upper SAPs.

Test ID 1.3.1.1

Name:

Control Request to Enable the Lower SAP

Purpose:

This test checks that when the layer manager sends a control request to enable the lower SAP, the lower SAP is successfully enabled.

Pre-Test Conditions:

General configuration of the ISUP layer is performed, and a lower and an upper SAP are configured.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STNSAP`**
   - **`elmId.sapID=sapid`**
   - **`action=ABND_ENA`**
2. ISUP enables and binds the concerned SAP
3. ISUP sends the control confirmation indicating success

Test ID 1.3.1.2

Name:

Control Request to Enable the Group of Lower SAPs

Purpose:

This test checks that when the layer manager sends a control request to enable the group of lower SAPs, the ISUP layer successfully enables the concerned SAPs.

Pre-Test Conditions:

General configuration of the ISUP layer is performed, and two lower SAPs are configured. If two different variants are not supported, there are two options for testing:

1. Configure two SAPs with the same variant
2. Test group-related test cases on a single SAP

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STRGRSINSAP`**
   - **`action=ABND_ENA`**
   - **`subAction=SAELMNT`**
   - **`cntrl.par.dstProcId=dstProcId`**
2. ISUP enables and binds the concerned group of SAPs
3. ISUP sends the control confirmation indicating success

Test ID 1.3.1.3

Name:

Control Request to Disable the Lower SAP

Purpose:

This test checks that when the layer manager sends a control request to disable the lower SAP, the lower SAP is successfully disabled.

Pre-Test Conditions:

General configuration of the ISUP layer is performed, and one upper and one lower SAP are configured.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STNSAP`**
   - **`elmId.sapID=sapid`**
   - **`action=AUBND_DIS`**
2. ISUP disables the concerned SAP
3. ISUP sends the control confirmation indicating success
4. Check that ISUP rejects the data indication for the disabled SAP

Test ID 1.3.1.4

Name:

Control Request to Disable the Group of Lower SAPs

Purpose:

This test checks that when the layer manager sends a control request to disable the group of lower SAPs, the ISUP layer successfully disables the concerned SAPs.

Pre-Test Conditions:

General configuration of the ISUP layer is performed, and one upper and two lower SAPs are enabled. If two different variants are not supported, there are two options for testing:

1. Configure two SAPs with the same variant
2. Test group-related test cases on a single SAP

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STRGRSINSAP`**
   - **`action=AUBND_DIS`**
   - **`subAction=SAELMNT`**
   - **`cntrl.par.dstProcId=dstProcId`**
2. ISUP enables and binds the concerned group of SAPs
3. ISUP sends the control confirmation indicating success

Test ID 1.3.1.5

Name:

Control Request to Disable the Upper SAP

Purpose:

This test checks that when the layer manager sends a control request to disable the upper SAP, the upper SAP is successfully disabled.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STISAP`**
   - **`elmId.sapID=sapid`**
   - **`action=AUBND_DIS`**
2. ISUP disables the concerned SAP
3. ISUP sends the control confirmation indicating success

Test ID 1.3.1.6

Name:

Control Request to Disable the Group of Upper SAPs

Purpose:

This test checks that when the layer manager sends a control request to disable the group of upper SAPs, the ISUP layer successfully disables the concerned SAPs.

Pre-Test Conditions:

General configuration of the ISUP layer is performed, and one lower and two upper SAPs are enabled. If two different variants are not supported, there are two options for testing:

1. Configure two SAPs with the same variant
2. Test group-related test cases on a single SAP

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STRGRSIUSAP`**
   - **`action=AUBND_DIS`**
   - **`subAction=SAELMNT`**
   - **`cntrl.par.dstProcId=dstProcId`**
2. ISUP enables and binds the concerned group of SAPs
3. ISUP sends the control confirmation indicating success

Test ID 1.3.1.7

Name:

Control Request to Standby to Disable the Upper SAP

Purpose:

This test checks that when the layer manager sends a control request to the standby PSF - ISUP to disable the upper SAP, the request is rejected.

Pre-Test Conditions:

General configuration of the ISUP layer is performed and standby exists. One upper and one lower SAP are configured.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
    - **`elmId.emlnt=STISAP`**
    - **`elmId.sapID=sapid`**
    - **`action=AUBND_DIS`**
2. ISUP disables the concerned SAP
3. ISUP sends the control confirmation indicating failure with the reason as **`LCM_REASON_INVALID_STATE`**

Test ID 1.3.1.8

Name:

Control Request with Invalid **elmntId**

Purpose:

This test checks that when the layer manager sends a control request with an invalid **elmntId**, the request is rejected.

Pre-Test Conditions:

General configuration of the ISUP layer is performed and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following field specified:
   - Invalid **elmId.elmnt**
2. ISUP sends the control confirmation indicating failure with the reason as **LCM_REASON_INVALID_ELMNT**

Test ID 1.3.1.9

Name:

Control Request with Invalid Action

Purpose:

This test checks that when the layer manager sends a control request with an invalid action, the request is rejected.

Pre-Test Conditions:

General configuration of the ISUP layer is performed and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following field specified:
   - Invalid action
2. ISUP sends the control confirmation indicating failure with the reason as **`LCM_REASON_INVALID_ACTION`**

Test ID 1.3.1.10

Name:

Control Request to Disable an Upper SAP with an Invalid **`sapId`**

Purpose:

This test checks that when the layer manager sends a control request with an invalid **`sapId`** to disable a SAP, the request is rejected.

Pre-Test Conditions:

General configuration of the ISUP layer is performed and standby exists. One upper SAP is enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STISAP`**
   - **`elmId.sapID=sapid`** (invalid)
   - **`action=AUBND_DIS`**
2. ISUP sends the control confirmation indicating failure with the reason as **`LCM_REASON_INVALID_SAP`**

Test ID 1.3.1.11

Name:

Control Request to Standby to Disable an Upper SAP

Purpose:

This test checks that when the layer manager sends a control request to disable a SAP to the standby layer, the request is rejected.

Pre-Test Conditions:

General configuration of the ISUP layer is performed and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the standby layer with the following fields specified:
   - **`elmId.emlnt=STISAP`**
   - **`elmId.sapID=sapid`**
   - **`action=AUBND_DIS`**
2. ISUP sends a control confirmation indicating failure with the reason as **`LCM_REASON_INVALID_STATE`**

6.5.1.3.2 Test Group 1.3.2 Debug

This test group tests the debug printing functionality of PSF - ISUP.

Test ID 1.3.2.1

Name:

Debug Print Control Enable

Purpose:

This test checks the debug message printing function of PSF - ISUP by sending the control request to enable the debug prints in all classes.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP with the following fields specified:
   - **`elmId.elmnt`** as **`STGEN`**
   - **`AENA`** in the **`action`** field and **`SADBG`** in **`subaction`**
   - **`t.cntrl.ctlType.umDbg.dbgMask`** as **`0`**
2. Check that PSF - ISUP debugger prints are enabled by making a call and observing the debug print during update
3. Check that control confirmation indicating success is received and contains the same transaction ID as the one sent to PSF - ISUP in the control request

Test ID 1.3.2.2

Name:

Debug Print Control Disable

Purpose:

This test checks the debug message printing function of PSF - ISUP by sending the control request to disable the debug prints in all classes.

Pre-Test Conditions:

General configuration for the ISUP layer is performed.

Test Description:

The steps required to carry out the test are:

1. Send a control request to ISUP with the following fields specified:
   - **`elmId.elmnt`** as **`STGEN`**
   - **`ADISIMM`** in the **`action`** field and **`SADBG`** in **`subaction`**
   - **`t.cntrl.ctlType.umDbg.dbgMask`** as **`0`**
2. Check that PSF - ISUP debugger prints are disabled
3. Check that control confirmation indicating success is received and contains the same transaction ID as the one sent to PSF - ISUP in the control request

6.5.1.3.3 Test Group 1.3.3 Alarm

This test group tests the alarm-reporting function of PSF - ISUP. The test cases are described in detail below.

Test ID 1.3.3.1

Name:

Alarm Generation Control

Purpose:

This test checks that when alarms are enabled, PSF - ISUP sends a status indication to the layer manager whenever needed.

Pre-Test Conditions:

ISUP standby copy is available and configured.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the standby PSF - ISUP to enable the alarm generation
2. Check that a positive confirmation is received from PSF - ISUP in the control confirm with the same transaction ID as the one sent to PSF - ISUP in the control request
3. Send an update message with an out-of-order sequence number
4. Check that the standby ISUP generates an alarm with the event as **`CMSWFT_SEQERR`**
5. Send a control request to the standby PSF - ISUP to disable the alarm generation
6. Send an update message with an out-of-order sequence number
7. Check that the standby ISUP does not generate an alarm

Test ID 1.3.3.2

Name:

Alarm Generation on Getting Erroneous Update Message

Purpose:

This test checks that when alarms are enabled, PSF - ISUP sends a status indication to the layer manager whenever needed.

Pre-Test Conditions:

ISUP standby copy is available and configured.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the standby PSF - ISUP to enable alarm-generation
2. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID as the one sent to PSF - ISUP in the control request
3. Send an update message with an error, such as invalid table type
4. Check that standby ISUP generates an alarm with the event as **`CMSWFT_UPDMSG_ERR`**

6.5.1.3.4 Test Group 1.3.4 Change Status

The system manager may change the status of the protocol layer to active or standby for various fault-tolerant activities, such as forced switchover or controlled switchover. This group tests PSF - ISUP's responses when PSF - ISUP receives status change requests from the system manager. The internal actions taken by PSF - ISUP are tested in run-time, warmstart, and controlled switchover test groups. The test cases under this group are described in detail below.

Test ID 1.3.4.1

Name:

Active to Active with Peer Disabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer disabled.

Pre-Test Conditions:

ISUP is configured as active with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP on an active node to become active with peer disabled
2. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_UBND`**.
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID as the one sent to PSF - ISUP in the control request

Test ID 1.3.4.2

Name:

Active Doing WS to Active with Peer Disabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer disabled. The PSF - ISUP layer receives this primitive while warmstart is going on.

Pre-Test Conditions:

ISUP is configured as active with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request with **`action`** as **`AWARMSTART`** to the active layer
2. Send a control request to PSF - ISUP on an active node to become active with peer disabled
3. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_UBND`**.
4. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID as the one sent to PSF - ISUP in the control request, and that the warmstart update is stopped

Test ID 1.3.4.3

Name:

Standby Layer to Active with Peer Disabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer disabled. PSF - ISUP receives this request when it is standby.

Pre-Test Conditions:

ISUP is configured as standby with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to become active with peer disabled
2. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_UBND`**.
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID as the one sent to PSF - ISUP in the control request

Test ID 1.3.4.4

Name:

OOS Layer to Active with Peer Disabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer disabled. PSF - ISUP receives this request when it is OOS.

Pre-Test Conditions:

ISUP is in OOS state.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to become active with peer disabled
2. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_UBND`**.
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.5

Name:

Active Doing WS to Active with Peer Enabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer enabled. PSF - ISUP receives this primitive while warmstart is going on.

Pre-Test Conditions:

ISUP is configured as active with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request with **`action`** as **`AWARMSTART`** to the active layer
2. Send a control request to PSF - ISUP on an active node to become active with peer enabled
3. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_BND`**.
4. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request, and that warmstart update is stopped

Test ID 1.3.4.6

Name:

Standby Layer to Active with Peer Enabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer enabled. PSF - ISUP receives this request when it is standby.

Pre-Test Conditions:

ISUP is configured as standby with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to become active with peer enabled
2. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_BND`**.
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.7

Name:

OOS Layer to Active with Peer Enabled

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become active with peer enabled. PSF - ISUP receives this request when it is OOS.

Pre-Test Conditions:

ISUP is in OOS state.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to become active with peer enabled
2. Check that **`ziProtState`** is made active and sequence numbers are initialized. Check that **`tmrFlag`** is updated to enable the timer handling and that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_BND`**.
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.8

Name:

Active Layer to Standby

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become standby. PSF - ISUP receives this request when it is active.

Pre-Test Conditions:

ISUP is configured as an active layer.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP on an active node to become standby with peer enabled
2. Check that **`ziProtState`** is made standby and sequence numbers are initialized. Check that **`tmrFlag`** is updated to disable the timer handling and that all currently running timers are removed and component queues are cleared. Check that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_BND`**.
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.9

Name:

Active Layer Doing WS to Standby

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become standby. PSF - ISUP receives this request when it is doing warmstart.

Pre-Test Conditions:

ISUP is configured as an active state.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active node to initiate warmstart
2. Send a control request to PSF - ISUP on an active node to become standby with peer enabled
3. Check that **`ziProtState`** is made standby and sequence numbers are initialized. Check that **`tmrFlag`** is updated to disable the timer handling and that all currently running timers are removed and component queues are cleared. Check that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_BND`**.
4. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.10

Name:

OOS Layer to Standby

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to become standby. PSF - ISUP receives this request when it is OOS.

Pre-Test Conditions:

ISUP is in OOS state and general configuration is done.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to become standby with peer enabled
2. Check that **`ziProtState`** is made standby and that the peer SAP status is properly updated—that is **`ziPeerSapCb.state`** is **`CMPFTHA_BND`**
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.11

Name:

Disable Peer Interface to Active Node

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to disable the peer interface. The PSF - ISUP layer receives this request when it is active.

Pre-Test Conditions:

ISUP is configured as an active node.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to disable the peer
2. Check that the peer SAP status is properly updated—that is, **`ziPeerSapCb.state`** is **`CMPFTHA_UBND`**
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.12

Name:

Disable Peer Interface to Standby Node

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to disable the peer interface. PSF - ISUP layer receives this request when it is standby.

Pre-Test Conditions:

ISUP is configured as a standby node.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to disable the peer
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.13

Name:

Disable Peer Interface to OOS Node

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to disable the peer interface. PSF - ISUP receives this request when it is OOS.

Pre-Test Conditions:

ISUP is in OOS state and general configuration is done.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to disable the peer
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.14

Name:

Abort Control Request to Active Node

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to abort the update. PSF - ISUP receives this request when it is active.

Pre-Test Conditions:

ISUP is configured as an active node.

Test Description:

The steps required to carry out the test are as follows:

1. Send a control request to PSF - ISUP to abort
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.15

Name:

Abort Control Request to Standby Node

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to abort the update. PSF - ISUP receives this request when it is standby.

Pre-Test Conditions:

ISUP is configured as a standby node.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to abort
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.16

Name:

Abort Control Request to OOS Node

Purpose:

This test checks the functionality of PSF - ISUP when it receives a control request from the system manager to abort the update. PSF - ISUP receives this request when it is OOS.

Pre-Test Conditions:

ISUP is in OOS state and general configuration is done.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP to abort
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.4.17

Name:

Invalid Status Change

Purpose:

This is a negative test that tests the validation functionality of PSF - ISUP for a control request from the system manager. If a control request is sent to PSF - ISUP without first being configured, PSF - ISUP sends a failure indication to the system manager.

Pre-Test Conditions:

The ISUP layer is OOS and not configured.

Test Description:

The steps required to carry out the test are:

1. Send a control request to PSF - ISUP on an OOS node to become active
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP indicating that PSF - ISUP has not yet been configured

6.5.1.3.5 Test Group 1.3.5 Warmstart

The system manager sends a warmstart request to the active PSF - ISUP whenever an OOS node has to be made standby. This test group tests the PSF - ISUP responses to the warmstart control requests sent by the system manager. The test cases under this group are described in detail below:

Test ID 1.3.5.1

Name:

Warmstart Peer Request to Active with Peer Enabled

Purpose:

This test checks that when the active PSF - ISUP receives a control request to warmstart the peer, the active PSF - ISUP sends two confirmations back to the system manager—one indicating that the warmstart is continuing and another indicating that the warmstart has been completed successfully.

Pre-Test Conditions:

ISUP is configured as an active node with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP to warmstart the peer
2. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request. Check that the status in the control confirm indicates that the warmstart is continuing.
3. Check that another control confirm from the active PSF - ISUP is received indicating that warmstart is complete and that this confirm also contains the same transaction ID sent by the system manager in its control request

Test ID 1.3.5.2

Name:

Warmstart Peer Request to Active with Peer Disabled

Purpose:

This test checks that when the active PSF - ISUP with a disabled peer receives a control request to warmstart the peer, the active PSF - ISUP rejects the request.

Pre-Test Conditions:

ISUP is configured as an active node with peer disabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP to warmstart the peer
2. Check that a positive confirmation, indicating that WS is running, is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request.

Test ID 1.3.5.3

Name:

Warmstart Peer Request to Active Currently Doing WS

Purpose:

This test checks that when the active PSF - ISUP receives a control request to warmstart the peer, the active PSF - ISUP sends confirmation back to the system manager, indicating that the warmstart is continuing. This test also checks the behavior of the active PSF - ISUP when the control confirm from the PSF - ISUP to the system manager gets lost in response to the warmstart request.

Pre-Test Conditions:

ISUP is configured as an active node and WS is going on.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP to warmstart the peer
2. Send another control request to the active PSF - ISUP to warmstart the peer
3. Check that two positive confirmations are received in the control confirm from PSF - ISUP with the same corresponding transaction IDs that were sent to PSF - ISUP in the control requests. Check that the status in the control confirm indicates that the warmstart is continuing.
4. Check that another control confirm from the active PSF - ISUP is received indicating that warmstart is complete, and that this confirm also contains the same transaction ID sent by the system manager in its control request

Test ID 1.3.5.4

Name:

Warmstart Peer Request to Standby

Purpose:

This test checks that when the standby PSF - ISUP receives a control request to warmstart the peer, the standby PSF - ISUP rejects the request.

Pre-Test Conditions:

ISUP is configured as a standby node with peer disabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the standby PSF - ISUP to warmstart the peer
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.5.5

Name:

Warmstart Peer Request to OOS

Purpose:

This test checks that when the OOS PSF - ISUP receives a control request to warmstart the peer, the OOS PSF - ISUP rejects the request.

Pre-Test Conditions:

ISUP is in OOS state.

Test Description:

The steps required to carry out the test are:

- Send a control request to the OOS PSF - ISUP to warmstart the peer
- Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.5.6

Name:

Disable Peer SAP Request while WS is Going on

Purpose:

This test checks the PSF - ISUP functionality to handle the disabled peer SAP request while warmstart is going on.

Pre-Test Conditions:

ISUP is configured and peer SAP is enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP to warmstart the peer. Immediately send another control request to disable the peer SAP.
2. Check that a positive confirmation is received in the control confirm and that the status in the control confirm indicates that warmstart is continuing. This control confirm carries the transaction ID of the first control requests sent to PSF - ISUP.
3. Check that another control confirm is received indicating that the request can not be entertained. This control confirm should have the same transaction ID as the one sent in the control request to disable the peer SAP.

Test ID 1.3.5.7

Name:

Warmstart Abort

Purpose:

This test checks the PSF - ISUP functionality to abort the ongoing warmstart procedure.

Pre-Test Conditions:

ISUP is configured and the peer SAP is enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP to warmstart the peer. Immediately send another control request to abort the warmstart.
2. Check that a positive confirmation is received in the control confirm and that the status in the control confirm indicates that warmstart is continuing. This control confirm carries the transaction ID of the first control requests sent to PSF - ISUP.
3. Check that another control confirm is received with the same transaction ID as the one in the control request to abort warmstart
4. Check that no control confirm is sent by PSF - ISUP indicating that the warmstart is completed

6.5.1.3.6 Test Group 1.3.6 Controlled Switchover

The system manager sends a request to the active PSF - ISUP to synchronize the peer whenever the system manager wants to do controlled switchover. This test group checks the functionality of PSF - ISUP at the system manager interface for the synchronization request. The test cases in this group are described in detail below:

Test ID 1.3.6.1

Name:

CSW Request to Active PSF - ISUP with Peer Enabled

Purpose:

This test checks that when the active PSF - ISUP receives a control request for CSW, the active PSF - ISUP sends two confirmations to the system manager—one indicating that the CSW is continuing and one indicating that the CSW has been completed successfully.

Pre-Test Conditions:

ISUP is configured as an active node with the peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP for CSW
2. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request. Check that the status in the control confirm indicates that the CSW is continuing.
3. Check that another control confirm from the active PSF - ISUP is received indicating that CSW is complete, and that this confirm also contains the same transaction ID that was sent by the system manager in its control request

Test ID 1.3.6.2

Name:

CSW Request to Active PSF - ISUP with Peer Disabled

Purpose:

This test checks that when the active PSF - ISUP with disabled peer receives a control request for CSW, the active PSF - ISUP rejects the request.

Pre-Test Conditions:

ISUP is configured as an active node with peer disabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP for CSW
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.6.3

Name:

CSW Request to Active PSF - ISUP Currently Doing CSW

Purpose:

This test checks that when the active PSF - ISUP receives a control request for CSW, the active PSF - ISUP sends confirmation back to the system manager, indicating that the CSW is continuing. This test also checks the behavior of the active PSF - ISUP when the control confirm from the PSF - ISUP to the system manager gets lost in response to the CSW request.

Pre-Test Conditions:

ISUP is configured as an active node.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP for CSW
2. Send another control request to the active PSF - ISUP for CSW
3. Check that two positive confirmations are received in the control confirm from PSF - ISUP with the same corresponding transaction IDs that were sent to PSF - ISUP in the control requests. Check that the status in the control confirm indicates that the CSW is continuing.
4. Check that another control confirm from the active PSF - ISUP is received indicating that CSW is complete, and that this confirm also contains the same transaction ID that was sent by the system manager in its control request for the first CSW

Test ID 1.3.6.4

Name:

CSW Request to Active PSF - ISUP Currently Doing WS

Purpose:

This test checks that when the active PSF - ISUP receives a control request for CSW, the active PSF - ISUP sends negative confirmation back to the system manager, indicating that the CSW can not be performed.

Pre-Test Conditions:

ISUP is configured as an active node and warmstart is going on.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP to warmstart the peer
2. Send another control request to the active PSF - ISUP for CSW
3. Check that a positive confirmation is received in the control confirm from PSF - ISUP with the same corresponding transaction IDs that were sent to PSF - ISUP in the control requests. Check that the status in the control confirm indicates that the warmstart is continuing.
4. Check that another control confirm from the active PSF - ISUP is received indicating that CSW can not be performed

Test ID 1.3.6.5

Name:

CSW Request to Standby PSF - ISUP

Purpose:

This test checks that when the standby PSF - ISUP receives a control request for CSW, the standby PSF - ISUP rejects the request.

Pre-Test Conditions:

ISUP is configured as a standby node with peer disabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the standby PSF - ISUP for CSW
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.6.6

Name:

CSW Request to OOS PSF - ISUP

Purpose:

This test checks that when the OOS PSF - ISUP receives a control request for CSW, the OOS PSF - ISUP rejects the request.

Pre-Test Conditions:

ISUP is in OOS state.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the OOS PSF - ISUP for CSW
2. Check that a negative confirmation is received in the control confirm from PSF - ISUP with the same transaction ID that was sent to PSF - ISUP in the control request

Test ID 1.3.6.7

Name:

CSW Abort

Purpose:

This test checks the PSF - ISUP functionality to abort the ongoing CSW procedure.

Pre-Test Conditions:

ISUP is configured and peer SAP is enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active PSF - ISUP for CSW. Immediately send another control request to abort the CSW
2. Check that a positive confirmation is received in the control confirm and that the status in the control confirm indicates that CSW is continuing. This control confirm carries the transaction ID of the first control requests sent to PSF - ISUP.
3. Check that another control confirm is received with the same transaction ID as the one sent in the control request to abort CSW
4. Check that no control confirm is sent by PSF - ISUP indicating that the CSW is completed

6.5.1.3.7 Test Group 1.3.7 ShutDown

The system manager sends a request to the active PSF - ISUP to shutdown—that is, to release all of its resources and initialize its data structure. The test cases in this groups are described in detail below:

Test ID 1.3.7.1

Name:

Shutdown Request to Active Layer

Purpose:

This test checks that when the active PSF - ISUP receives a control request for shutdown, the active PSF - ISUP sends confirmation back to the system manager after releasing all of its resources.

Pre-Test Conditions:

ISUP is configured as an active node with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active ISUP with **`elmnt`** as **`STGEN`** and **`action`** as **`ASHUTDOWN`**
2. Check that ISUP sends confirmation of the shutdown request, releases all of its resources, and transits to OOS state without any configuration

Test ID 1.3.7.2

Name:

Shutdown Request to Active Layer Doing WS

Purpose:

This test checks that when the active PSF - ISUP receives a control request for shutdown, the active PSF - ISUP sends confirmation back to the system manager after releasing all of its resources. This request is received when the active layer is doing warmstart.

Pre-Test Conditions:

ISUP is configured as an active node with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active layer for warmstart
2. Send a control request to the active ISUP with **`elmnt`** as **`STGEN`** and **`action`** as **`ASHUTDOWN`**
3. Check that a confirmation is received indicating that WS is going on
4. Check that ISUP sends confirmation of the shutdown request and releases all of its resources and transits to OOS state without any configuration. ISUP sends another confirmation indicating the WS failure.

Test ID 1.3.7.3

Name:

Shutdown Request to Active Layer Doing CSW

Purpose:

This test checks that when the active PSF - ISUP receives a control request for shutdown, the active PSF - ISUP sends confirmation back to the system manager after releasing all of its resources. This request is received when the active layer is doing CSW.

Pre-Test Conditions:

ISUP is configured as an active node with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active layer for CSW
2. Send a control request to the active ISUP with **`elmnt`** as **`STGEN`** and **`action`** as **`ASHUTDOWN`**
3. Check that confirmation is received indicating that CSW is going on
4. Check that ISUP sends confirmation of the shutdown request and releases all of its resources and transits to OOS state without any configuration. ISUP sends another confirmation to indicate the CSW failure.

Test ID 1.3.7.4

Name:

Shutdown Request to Standby Layer

Purpose:

This test checks that when the standby PSF - ISUP receives a control request for shutdown, the standby PSF - ISUP sends confirmation back to the system manager after releasing all of its resources.

Pre-Test Conditions:

ISUP is configured as a standby node with peer enabled.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active ISUP with **`elmnt`** as **`STGEN`** and **`action`** as **`ASHUTDOWN`**
2. Check that ISUP sends confirmation of the shutdown request, releases all of its resources, and transits to OOS state without any configuration

Test ID 1.3.7.5

Name:

Shutdown Request to OOS Layer

Purpose:

This test checks that when the OOS PSF - ISUP receives a control request for shutdown, the OOS PSF - ISUP sends confirmation back to the system manager after releasing all of its resources.

Pre-Test Conditions:

ISUP is in OOS state and general configuration is done.

Test Description:

The steps required to carry out the test are:

1. Send a control request to the active ISUP with **`elmnt`** as **`STGEN`** and **`action`** as **`ASHUTDOWN`**
2. Check that ISUP sends confirmation of the shutdown request, releases all of its resources, and transits to OOS state without any configuration

6.5.2 Test Group 2 Run Time (RT)

The run-time test group contains the test cases to test the PSF - ISUP functionality to update the standby at run time. The active ISUP modifies the stable states of its control blocks during the handling of various run-time protocol events. The active PSF - ISUP updates the state changes to the standby ISUP.

Some of the test cases check that the active PSF - ISUP generates state update messages when ISUP handles different protocol events.

Test ID 2.1

Name:

Run-Time Update of the Upper SAP Control Block: Bind Request

Purpose:

This test checks that when ISUP receives a **BndReq** from the upper layer, ISUP sends the update primitive to update the upper SAP control block.

Pre-Test Conditions:

General configuration of the ISUP layer is done, and standby is available.

Test Description:

The steps required to carry out the test are:

1. Bind the ISUP layer by sending **BndReq** from the upper layer
2. Check that ISUP marks the state of the concerned SAP bound and passes the primitive to the lower layer
3. Respond with a positive **BndCfm** from the lower layer
4. Check that ISUP sends **BndCfm** to the upper layer with status as OK
5. Check that ISUP sends the update primitive with **spId**, **suInd**, **state**, **dstProcId**, **dstEnt**, and **dstInst** to update the standby

Test ID 2.2

Name:

Run-Time Update of the Upper SAP Control Block: Control Request to Disable the Upper SAP

Purpose:

This test checks that when the layer manager sends a control request to disable the upper SAP, the upper SAP is successfully disabled and standby ISUP is updated.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STISAP`**
   - **`elmId.sapID=sapid`**
   - **`action=AUBND_DIS`**
2. ISUP disables the concerned SAP
3. ISUP sends the control confirmation indicating success
4. Check that the upper SAP control block in the standby ISUP is also updated, particularly the **`suId`**, **`spId`**, **`state`**, **`dstProcId`**, **`dstEnt`**, and **`dstInst`** fields

Test ID 2.3

Name:

Run-Time Update: Control Request to Disable the Group of Upper SAPs

Purpose:

This test checks that when the layer manager sends a control request to disable the group of upper SAPs, the ISUP layer successfully disables the concerned SAPs and updates the standby ISUP layer. If two different variants are not supported, there are two options for testing:

1. Configure two SAPs with the same variant
2. Test group-related test cases on a single SAP

Pre-Test Conditions:

The general configuration of the ISUP layer is performed.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STRGRSIUSAP`**
   - **`action=AUBND_DIS`**
   - **`subAction=SAELMNT`**
   - **`cntrl.par.dstProcId=dstProcId`**
2. ISUP disables the concerned group of SAPs
3. ISUP sends the control confirmation indicating success
4. ISUP sends the update message to update the standby with **`spId`**, **`suId`**, **`state`**, **`pst.dstProcId`**, **`dstEnt`**, and **`dstInst`** for every associated SAP

Test ID 2.4

Name:

Run-Time Update of the Lower SAP Control Block: Bind Request

Purpose:

This test checks that when ISUP receives a **`BndReq`** from the upper layer, ISUP sends an update primitive to update the upper SAP control block and sends **`BndReq`** to the lower layers. Once confirmation is received, it sends the update message to standby to update **`suId`** and **`state`**.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby is available.

Test Description:

The steps required to carry out the test are:

1. Bind the ISUP layer by sending **`BndReq`** from the upper layer
2. Respond with a positive **`BndCfm`** from the lower layer
3. Check that ISUP has sent the update primitive to update the lower SAP control block on standby with **`suId`** and **`state`**

Test ID 2.5

Name:

Run-Time Update of the Lower SAP Control Block: Control Request to Disable the Lower SAP

Purpose:

This test checks that when the layer manager sends a control request to disable the lower SAP, the lower SAP is successfully disabled and standby ISUP is updated.

Pre-Test Conditions:

General configuration of the ISUP layer is performed.

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STNSAP`**
   - **`elmId.sapID=sapid`**
   - **`action=AUBND_DIS`**
2. ISUP disables the concerned SAP
3. Check that the lower SAP control block in the standby ISUP is also updated, particularly the state of the SAP

Test ID 2.6

Name:

Run-Time Update: Control Request to Disable the Group of Lower SAPs

Purpose:

This test checks that when the layer manager sends a control request to disable the group of lower SAPs, the ISUP layer successfully disables the concerned SAPs and updates the standby ISUP layer.

Pre-Test Conditions:

The general configuration of the ISUP layer is performed. If two different variants are not supported, there are two options for testing:

1. Configure two SAPs with the same variant
2. Test group-related test cases on a single SAP

Test Description:

The steps required to carry out the test are:

1. Send a control request with the following fields specified:
   - **`elmId.emlnt=STRGRSINSAP`**
   - **`action=AUBND_DIS`**
   - **`subAction=SAELMNT`**
   - **`cntrl.par.dstProcId=dstProcId`**
2. ISUP disables the concerned group of SAPs
3. ISUP sends the control confirmation indicating success
4. ISUP sends the update message to update the standby with **`suId`** and **`state`**

Test ID 2.7

Name:

Run-Time Update of the Circuit Control Block on Circuit-Related Event (Idle to Remotely Blocked)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, BLO) to the active ISUP layer on a valid circuit. Respond to the message (for example, with a block response).
2. Check that the following fields in the circuit control block are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 2.8

Name:

Run-Time Update of the Circuit Control Block on Circuit-Related Event (Remotely Blocked to Idle)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists. A valid circuit is remotely blocked.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, UBL) to the active ISUP layer on a valid, remotely-blocked circuit. Acknowledge the message (for example, with an unblock response).
2. Check that the following fields in the circuit control block are updated on the standby:
    - **`key`**
    - **`transStat`**
    - **`calProcStat`**
    - **`cirCtl`**
    - **`noRspFlgToUp`**
    - **`swtch`**
    - **`resFlag`**
    - **`tmrCnt`**
    - **`nonSS7Con`**

Test ID 2.9

Name:

Run-Time Update of the Circuit Control Block on Circuit-Related Event (Idle to Locally Blocked)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, **`BloReq`**) to the active ISUP layer on a valid idle circuit. Acknowledge the message (for example, with BLA).
2. Check that the following fields in the circuit control block are updated on the standby:
    - **`key`**
    - **`transStat`**
    - **`calProcStat`**
    - **`cirCtl`**
    - **`noRspFlgToUp`**
    - **`swtch`**
    - **`resFlag`**
    - **`tmrCnt`**
    - **`nonSS7Con`**

Test ID 2.10

Name:

Run-Time Update of the Circuit Control Block on Circuit-Related Event (Locally Blocked to Idle)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists. A valid circuit is locally blocked.

Test Description:

The steps required to carry out the test are:

1. Send a circuit related message (for example, **UnBloReq**) to the active ISUP layer on a valid, locally-blocked circuit. Acknowledge the message (for example, with UBA).
2. Check that the following fields in the circuit control block are updated on the standby:
   - **key**
   - **transStat**
   - **calProcStat**
   - **cirCtl**
   - **noRspFlgToUp**
   - **swtch**
   - **resFlag**
   - **tmrCnt**
   - **nonSS7Con**

Test ID 2.11

Name:

Run-Time Update of the Circuit Control Block on Circuit-Related Event (Locally Blocked to Locally and Remotely Blocked)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists. A valid circuit is locally blocked.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, BLO) to the active ISUP layer on a valid, locally-blocked circuit. Acknowledge the message (for example, with BLO response).
2. Check that the following fields in the circuit control block are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 2.12

Name:

Run-Time Update of the Circuit Control Block on Circuit Group-Related Event

Purpose:

This test verifies that the circuit control blocks for associated circuits in standby are properly updated on receiving a circuit group-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a circuit group-related message (for example, CGB) to the active ISUP layer on a valid circuit with a given range (for example, 2). Respond to the message (for example, with CGB response).
2. Check that the following fields in the circuit control blocks related to the associated circuits are updated on the standby:
   - `key`
   - `transStat`
   - `calProcStat`
   - `cirCtl`
   - `noRspFlgToUp`
   - `swtch`
   - `resFlag`
   - `tmrCnt`
   - `nonSS7Con`

Test ID 2.13

Name:

Run-Time Update of the Circuit Control Block on Circuit Group-Related Event (Remotely Blocked to Idle)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit group-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists. A valid circuit is remotely blocked.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, CGU) to the active ISUP layer on a remotely blocked circuit. Respond with a circuit group unblock response to acknowledge it.
2. Check that the following fields in the circuit control blocks (for all effected circuits) are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 2.14

Name:

Run-Time Update of the Circuit Control Block on Circuit Group-Related Event (Idle to Locally Blocked)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit group-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, CGB request) to the active ISUP layer on valid idle circuits. Acknowledge the message (for example, with CGBA).
2. Check that the following fields in the circuit control block are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 2.15

Name:

Run-Time Update of the Circuit Control Block on Circuit Group-Related Event (Locally Blocked to Idle)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit group-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists. A valid circuit is locally blocked.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, CGU request) to the active ISUP layer on valid, locally-blocked circuits. Acknowledge the message (for example, with CGUA).
2. Check that the following fields in the circuit control block are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 2.16

Name:

Run-Time Update of the Circuit Control Block on Circuit Group-Related Event (Locally Blocked to Locally and Remotely Blocked)

Purpose:

This test verifies that the circuit control block in standby is properly updated on receiving a circuit group-related message.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists. A valid circuit is locally blocked.

Test Description:

The steps required to carry out the test are:

1. Send a circuit-related message (for example, CGB) to the active ISUP layer on valid, locally-blocked circuits. Acknowledge the message (for example, with CGB response).
2. Check that the following fields in the circuit control block are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 2.17

Name:

Run-Time Update of the Connection Control Block on Receiving Connection-Related Events (Idle to Active)

Purpose:

This test verifies that the connection control blocks in standby are properly updated on receiving a connection-related event.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. On an idle circuit, send **ConReq** from layer 5, followed by CON from layer 3
2. Check that the following fields in the connection control block are updated on the standby:
   - **key**
   - **incC** or **outC**
   - **suInstId**
   - **lnkSel**
   - **dstAdr**
   - **srcAdr**
   - **chrgNum**
   - **charge**
   - **exchCalRef**
   - **end2end**
   - **useSCCP**
   - **SCCPUsed**
   - **resDir**
   - **xchgType**
   - **evntType**
   - **callTRef**

3. Check that the following fields in the circuit connection control block are updated on the standby:
   - **`toBeRelsd`**
   - **`chgRcvd`**
   - **`conPrcs`**
   - **`dCAllRef`**
   - **`phyDpc`**
   - **`cllModProc`**
   - **`conState`**
   - **`cirId`**
   - **`relResp`**
   - **`suspDir`**
   - **`eventType`**

Test ID 2.18

Name:

Run-Time Update of the Connection Control Block on Receiving Connection-Related Events (Active to Idle)

Purpose:

This test verifies that the connection control blocks in standby are properly updated on receiving a connection-related event.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. On a circuit with a connection in the active state, send REL followed by **`RelRsp`** from the upper layer
2. Check that the corresponding connection control block is deleted from the standby

Test ID 2.19

Name:

Out of Sequence Run-Time Update Message

Purpose:

This test verifies that the standby rejects an update message with an unexpected sequence number and generates an alarm on the layer manager.

Pre-Test Conditions:

The ISUP layer is configured as standby with peer SAP enabled.

Test Description:

The steps required to carry out the test are:

1. Send an update message to the standby PSF - ISUP with a random sequence number
2. Check that the standby PSF - ISUP generates an alarm with event **`LCM_EVENT_SEQERR`** on the management interface

Test ID 2.20

Name:

Run-Time Update Message with Invalid Table Type

Purpose:

This test verifies that the standby PSF - ISUP rejects an update message with an unexpected table type parameter and generates an alarm on the layer manager.

Pre-Test Conditions:

ISUP is configured as standby with peer SAP enabled.

Test Description:

The steps required to carry out the test are:

1. Send an update message to the standby PSF - ISUP with an unexpected table type
2. Check that standby generates an alarm with event **`LCM_EVENT_UPDMSG_ERR`** on the management interface

Test ID 2.21

Name:

Handling of Event from the Upper Interface by Standby Layer

Purpose:

This test checks the standby PSF - ISUP functionality to reject any event from the upper layer. The router is responsible for disabling any event to the standby copy of the layer.

Pre-Test Conditions:

ISUP layer is configured as standby and alarms are enabled.

Test Description:

The steps required to carry out the test are:

1. Send a connection request to the standby ISUP layer
2. The ISUP layer dumps the request from the upper layer
3. The ISUP layer sends an alarm to LM with event as **`LCM_EVENT_UI_INV_EVT`**

Test ID 2.22

Name:

Handling of Event from the Lower Interface by Standby Layer

Purpose:

This test checks the standby PSF - ISUP functionality to reject any event from the lower layer. The router is responsible for disabling any event to the standby copy of the layer.

Pre-Test Conditions:

ISUP is configured as standby.

Test Description:

The steps required to carry out the test are:

1. Send a data indication to the standby ISUP layer
2. The ISUP layer dumps the request from the lower layer
3. The ISUP layer sends an alarm to LM with event as **`LCM_EVENT_LI_INV_EVT`**

Test ID 2.23

Name:

No Run-Time Update if Peer SAP is Disabled

Purpose:

This test checks that the active PSF - ISUP with peer SAP disabled does not send update messages for the run-time update of the information.

Pre-Test Conditions:

General configuration for the ISUP layer is done and peer SAP is disabled.

Test Description:

The steps required to carry out the test are:

1. Send **`ConReq`** to the active ISUP layer and respond to IAM with CON
2. Check that no run-time update messages are generated towards the peer layer

Test ID 2.24

Name:

Standby Fallbacks the Information if Required

Purpose:

This test checks that the standby PSF - ISUP does not copy the information in an update message if it is transient information. This can occur if one event triggers the update of more than one data structure, and some of the data structures are not in a stable state.

Pre-Test Conditions:

ISUP is configured as an active layer and peer SAP is enabled.

Test Description:

The steps required to carry out the test are:

1. On an idle circuit, send **`ConReq`** from layer 5, followed by a block request to block the same circuit
2. Respond to the generated IAM with CON. This results in an update of the connection control block and circuit control block.
3. Check that, if standby ISUP receives this update message, the **`mntce`** state in the circuit control block is not updated, because this information is transient in nature

6.5.3 Test Group 3 New Primitives/Recovery from Primitive Loss

This group of test cases focuses on the recovery mechanism in the ISUP layer to handle the loss of the primitives from both sides.

Test ID 3.1

Name:

ISUP Sends Bind Confirmation to the Upper Layer

Purpose:

This test checks that when ISUP receives **`BndReq`** from the upper layer, ISUP responds with the confirmation.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Bind the ISUP layer by sending **`BndReq`** from the upper layer
2. Check that ISUP marks the state of the concerned SAP as bound and passes the primitive to the lower layer
3. Respond with a positive **`BndCfm`** from the lower layer
4. Check that ISUP sends **`BndCfm`** to the upper layer with status as OK

Test ID 3.2

Name:

Multiple Bind Requests are Accepted by the ISUP Layer

Purpose:

This test checks that when the upper layer sends more than one bind request, ISUP does not reject the subsequent requests. If the bind request is lost, the upper layer will resend the request. This test checks the handling of subsequent bind request to the ISUP layer.

Pre-Test Conditions:

ISUP task created on an OOS node

Test Description:

Once the ISUP task is created, it is ready to accept configuration from the layer manager. The steps required to carry out the test are:

1. Perform general configuration of ISUP
2. Send bind request from the upper layer
3. ISUP binds the upper SAP. The corresponding lower SAP is also bound.
4. Send another bind request from the upper layer
5. ISUP does not send this request to the lower layer, but sends bind confirm to the upper layer

Test ID 3.3

Name:

ISUP Repeats Bind Request to MTP Level 3 if Bind Confirmation is Delayed

Purpose:

This test checks that ISUP can recover from the loss of the bind request to the lower layer by guarding the confirmation of the bind request. ISUP resends the bind request once the timer has expired.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Bind the ISUP layer by sending **BndReq** from the upper layer
2. ISUP marks the state of the concerned upper SAP as bound and passes the primitive to the lower layer at the associated SAP
3. Do not respond to the bind request until another bind request is received
4. Check that the corresponding SAPs are properly bound

Test ID 3.4

Name:

ISUP Repeats Bind Request to MTP Level 3 if Bind Confirmation is Delayed and, after a Certain Number of Retries, it Sends Alarm to LM and Stops Retrying

Purpose:

This test checks that ISUP, after trying for a bind confirm for a certain number of times, sends an alarm to the layer manager.

Pre-Test Conditions:

General configuration of the ISUP layer is done and alarms are enabled.

Test Description:

The steps required to carry out the test are:

1. Bind the ISUP layer by sending **`BndReq`** from the upper layer
2. ISUP marks the state of the concerned upper SAP as bound and passes the primitive to the lower layer at the associated SAP in the form of **`SiLiSntBndReq`**. ISUP marks the status of the SAP as **`SI_SAP_WAIT_BNDCFM`**.
3. Do not respond to the bind request
4. Check that ISUP retries a certain number of times and then sends an alarm to LM with **`eventType`** as **`LCM_EVENT_BND_FAIL`**

Test ID 3.5

Name:

ISUP Repeats **`RelInd`** to the Upper Layer if **`RelRsp`** is Delayed or Lost

Purpose:

This test checks that ISUP can recover from the loss of the **`RelInd`** primitive to the upper layer. If the upper layer does not respond with **`RelRsp`**, ISUP sends **`RelInd`** again. This test aims at the single expiry of the timer guarding **`RelRsp`**.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Send **`ConReq`** to ISUP and do not respond with IAM
2. ISUP sends REL to the peer and **`RelInd`** to the upper layer
3. Do not respond with **`RelRsp`** until another **`RelInd`** is received
4. Check that, after sending **`RelRsp`**, no more **`RelInd`** are received

Test ID 3.6

Name:

ISUP Repeats **`RelInd`** to the Upper Layer if **`RelRsp`** is Delayed or Lost and Sends Alarm to LM after a Certain Number of Retries

Purpose:

This test checks that, if ISUP can not recover from the loss of the **`RelInd`** primitive, it retries. After a certain number of attempts, an alarm is sent to the layer manager and the circuit is blocked.

Pre-Test Conditions:

General configuration of the ISUP layer is done and alarms are enabled.

Test Description:

The steps required to carry out the test are:

1. Send **`ConReq`** to ISUP and do not respond with IAM
2. ISUP sends REL to the peer and **`RelInd`** to the upper layer
3. Do not respond with **`RelRsp`** from the upper layer
4. Check that ISUP sends multiple **`RelInd`** periodically and that, after a certain number of retries, an alarm is sent to the layer manager and status indication to call control that the corresponding circuit is blocked

Test ID 3.7

Name:

ISUP Responds with Status of the Point Code (Interface[1] ID) and Congestion on Receiving **ptCdStaReq** from CC

Purpose:

This test checks that ISUP provides CC with the correct status of the point code (interface ID) and its congestion level.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Send **PtCdeStaReq** from the upper layer, requesting status and congestion level of a valid point code (interface ID) (which is resumed with congestion level **0**)
2. Check that ISUP sends the status request to the lower layer and waits for the confirmation
3. Send **PtCdeStaCfm** with status as **SIT_STA_RESUME** and congestion level as **SIT_STA_CONG0**
4. Check that ISUP sends correct status and congestion level of the corresponding point code (interface ID); that is, that ISUP sends status as **SIT_STA_RESUME** and congestion level as **SIT_STA_CONG0**

[1] DPC entity at the layer manager interface has been replaced by *Interface* entity, which is a combination of DPC, OPC, ISUP protocol variant type, and network type.

Test ID 3.8

Name:

ISUP Requests the Status of the Point Code and Congestion on Receiving **SiLiSntStaInd** from MTP Level 3 and Does Not Expect Confirmation on Receiving the Resume Indication

Purpose:

This test checks that when ISUP receives **SiLiSntStaInd** from MTP Level 3 with status as **SN_PAUSE** or **SN_CONG**, ISUP sends **SiLiSntStaReq** to MTP Level 3 to request the current status of the point code (interface ID). The confirmation to this request is guarded by a timer. The request is resent when the timer expires.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Send **SiLiSntStaInd** with the status of the point code (interface ID) as **SN_PAUSE**
2. Check that ISUP requests the status of the associated point code (interface ID) by sending **SiLiSntStaReq**
3. Do not respond with confirmation until **StaReq** is resent (after timer expiry)
4. Respond with **SiLiSntStaInd** with the status of the same point code (interface ID) as **SN_RESUME**
5. Check that the timer is stopped and that **SiLiSntStaReq** is not sent again

Test ID 3.9

Name:

ISUP Requests the Status of the Point Code and Congestion on Receiving **SiLiSntStaInd** from MTP Level 3 and on Receiving Confirmation with Status as **SN_RESUME**

Purpose:

This test checks that when ISUP receives **SiLiSntStaInd** from MTP Level 3 with status as **SN_PAUSE** or **SN_CONG,** ISUP sends **SiLiSntStaReq** to MTP Level 3 to request the current status of the point code (interface ID) . The confirmation to this request is guarded by a timer. The request is resent when the timer expires.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Send **SiLiSntStaInd** with the status of the point code (interface ID) as **SN_PAUSE**
2. Check that ISUP requests the status of the associated point code (interface ID) by sending **SiLiSntStaReq**
3. Do not respond with confirmation until **StaReq** is resent (after timer expiry)
4. Respond with **SiLiSntStaCfm** with the status of the same point code (interface ID) as **SN_RESUME**

Test ID 3.10

Name:

ISUP Repeats **`RelInd`** to the Upper Layer if **`RelRsp`** is Delayed or Lost (Call is Released from the Peer)

Purpose:

This test checks whether ISUP can recover from the loss of the **`RelInd`** primitive to the upper layer. If the upper layer does not respond to the indication with **`RelRsp`**, ISUP resends the indication when it receives another REL from the peer.

Pre-Test Conditions:

General configuration of the ISUP layer is done.

Test Description:

The steps required to carry out the test are:

1. Send **`ConReq`** to ISUP and respond with CON
2. Send REL from the peer; ISUP sends **`RelInd`** to the upper layer
3. Do not respond with **`RelRsp`**
4. Send REL from the peer to ISUP
5. Check that another **`RelInd`** is received

6.5.4 Test Group 4 Warmstart (WS)

The Warmstart test group contains the test cases to test the PSF - ISUP functionality to update the peer when the peer has to be made standby from out-of-service. This group also tests the run-time state update during warmstart as warmstart is completed over multiple scheduling. The failures encountered during the warmstart procedure are also tested. The test cases under this group are described in detail below:

Test ID 4.1

Name:

Warmstart Update of the Upper SAP Control Block

Purpose:

This test checks that when ISUP receives a warmstart request, ISUP updates the upper SAP control block properly.

Pre-Test Conditions:

General configuration of the ISUP layer is done and warmstart request is given to the layer.

Test Description:

The steps required to carry out the test are:

1. Check that ISUP has sent the update primitive to update the upper SAP control block in standby with **`spId`**, **`suInd`**, **`state`**, **`dstProcId`**, **`dstEnt`**, and **`dstInst`**

Test ID 4.2

Name:

Warmstart Update of the Lower SAP Control Block

Purpose:

This test checks that when ISUP receives a warmstart request, ISUP updates the lower control block properly.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby is available.

Test Description:

The steps required to carry out the test are:

1. Check that ISUP has sent the update primitive to update the lower SAP control block on standby with **`suId`** and **`state`**

Test ID 4.3

Name:

Warmstart Update of the Circuit Control Block

Purpose:

This test checks that the circuit control block in standby is properly updated on receiving a request for warmstart.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. Check that the following fields in the circuit control block are updated on the standby:
   - **`key`**
   - **`transStat`**
   - **`calProcStat`**
   - **`cirCtl`**
   - **`noRspFlgToUp`**
   - **`swtch`**
   - **`resFlag`**
   - **`tmrCnt`**
   - **`nonSS7Con`**

Test ID 4.4

Name:

Warmstart Update of the Connection Control Block

Purpose:

This test checks that the connection control blocks in standby are properly updated on receiving a request for warmstart.

Pre-Test Conditions:

General configuration of the ISUP layer is done, standby exists, and a connection exists.

Test Description:

The steps required to carry out the test are:

1. Send a control request from the LM to initiate warmstart
2. Check that the following fields in the connection control block are updated on the standby:
    - **key**
    - **incC** or **outC**
    - **suInstId**
    - **lnkSel**
    - **dstAdr**
    - **srcAdr**
    - **chrgNum**
    - **charge**
    - **exchCalRef**
    - **end2end**
    - **useSCCP**
    - **SCCPUsed**
    - **resDir**
    - **xchgType**
    - **evntType**
    - **callTRef**

3. Check that the following fields in the circuit connection control block are updated on the standby:

   - **`toBeRelsd`**
   - **`chgRcvd`**
   - **`conPrcs`**
   - **`dCAllRef`**
   - **`phyDpc`**
   - **`cllModProc`**
   - **`conState`**
   - **`cirId`**
   - **`relResp`**
   - **`suspDir`**
   - **`eventType`**

Test ID 4.5

Name:

Abort Warmstart

Purpose:

This test checks that when the active PSF - ISUP is waiting for warmstart confirmation from the standby, and the system manager sends an abort to the active PSF - ISUP, then the active PSF - ISUP stops the timer to wait for the confirmation from the standby.

Pre-Test Conditions:

ISUP is configured as active layer and peer SAP is enabled.

Test Description:

The steps required to carry out the test are:

1. Send a warmstart request to the active PSF - ISUP
2. Delete the inbound data confirm from the standby PSF - ISUP and send a control request to the active PSF - ISUP to abort the warmstart
3. Check that the active PSF - ISUP sends a confirmation to the system manager for the abort request
4. Wait for a period of time longer than the time for which the active waits for confirmation from the standby
5. Check that no confirmation is sent by the active PSF - ISUP to the system manager for the warmstart request

6.5.5 Test Group 5 Controlled Switchover (CSW)

The Controlled Switchover test group contains the test cases to test the PSF - ISUP functionality to update the standby when the standby has to be made active. The failures encountered during the synchronization procedure are also tested by this group. The test cases under this group are described in detail below:

Test ID 5.1

Name:

CSW Update of the Circuit Control Block

Purpose:

This test checks that the circuit control block in standby is properly updated on receiving a request for controlled switchover.

Pre-Test Conditions:

General configuration of the ISUP layer is done and standby exists.

Test Description:

The steps required to carry out the test are:

1. Check that the following fields in the circuit control block are updated on the standby:
    - **`key`**
    - **`transStat`**
    - **`calProcStat`**
    - **`cirCtl`**
    - **`noRspFlgToUp`**
    - **`swtch`**
    - **`resFlag`**
    - **`tmrCnt`**
    - **`nonSS7Con`**
    - Timers associated with the circuit control block

Test ID 5.2

Name:

CSW Update of the Connection Control Block

Purpose:

This test checks that the connection control blocks in standby are properly updated on receiving a request for controlled switchover.

Pre-Test Conditions:

General configuration of the ISUP layer is done, standby exists, and a connection exists.

Test Description:

The steps required to carry out the test are:

1. Check that the following fields in the connection control block are updated on the standby:
    - **`key`**
    - **`incC`** or **`outC`**
    - **`suInstId`**
    - **`lnkSel`**
    - **`dstAdr`**
    - **`srcAdr`**
    - **`chrgNum`**
    - **`charge`**
    - **`exchCalRef`**
    - **`end2end`**
    - **`useSCCP`**
    - **`SCCPUsed`**
    - **`resDir`**
    - **`xchgType`**
    - **`evntType`**
    - **`callTRef`**
    - Timers associated with the connection control block

2. Check that the following fields in the connection control block are updated on the standby:
   - **`toBeRelsd`**
   - **`chgRcvd`**
   - **`conPrcs`**
   - **`dCAllRef`**
   - **`phyDpc`**
   - **`cllModProc`**
   - **`conState`**
   - **`cirId`**
   - **`relResp`**
   - **`suspDir`**
   - **`eventType`**
   - Release message buffer and **`msgToSegm`**

Test ID 5.3

Name:

CSW Update of the Circuit Group Control Block

Purpose:

This test verifies that the circuit group control blocks in standby are properly updated on receiving a request for controlled switchover.

Pre-Test Conditions:

General configuration of the ISUP layer is done, standby exists, and a circuit group control block exists.

Test Description:

The steps required to carry out the test are:

1. Check that the following fields in the circuit group control block are updated on the standby:
   - **`rangStat`**
   - **`cgsmti`**
   - **`state`**
   - **`cirState`**
   - **`tmrCnt`**
   - **`numResInd`**
   - **`querRange`**
   - **`quesrPrcs`**
   - **`timers`**

SS7 Glossary

A

Adjacent Signalling Points: Two signalling points directly connected through signalling links.

Authentication Center (AC): A database that stores security codes embedded into the memory of cellular phones. This code, along with the particular serial number of a given phone, prevents the use of unauthorized cellular devices within a particular network. See **Equipment Identity Register (EIR)**.

B

Base Station Subsystem (BSS): Within a cellular communications network, antenna sites (also known as cell sites) are made up of a **Base Transceiver Station (BTS)** and a **Base Station Controller (BSC).** The BSS is the pairing of the BTS and BSC. The BTS communicates with cellular phones within a given network, connecting the caller with the cell. The BSC is the interface between the BTS and any switching facilities that may be needed by the caller. See **Mobile Switching Center (MSC)**.

C

Combined Link Set: A collection of link sets that perform load sharing.

Common Channel Signalling: A signalling technique in which the signalling information is sent across the network separately from the voice and data that it is related to.

Consultative Committee International Telegraph and Telephone (CCITT): An international organization that developed communication standards such as Recommendation X.25. Replaced by the United Nation's ITU-T.

Cyclic Redundancy Check (CRC): A mathematical algorithm that derives a numerical value based on the bits in a block of data prior to transmission. If the receiving layer finds any discrepancies between the bit value of the received data packet and the accompanying frame check sequence field, then a transmission error is assumed.

D

Data Communications Equipment (DCE): Devices that handle routing and switching functions for a given network. See also **Data Terminal Equipment (DTE)**.

Destination Point Code (DPC): A node ID that identifies the destination point of a message in a signalling network. See **Originating Point Code (OPC)**.

Data Terminal Equipment (DTE): A device, such as a PC or main frame computer, that is attached to a network and is either the point of origin or the destination point of data.

E

End-to-End Signalling: Signalling that is transmitted directly between network endpoints.

Equipment Identity Register (EIR): A database that stores the serial numbers of each cellular telephone in use within a particular coverage area. In conjunction with the AC, the EIR prevents unauthorized use of cellular phones within a given geographic area.

F

Flow Control: A function that regulates the transmission of messages between adjacent protocol layers.

G

Global Title: An address which does not contain all the information necessary to route it to a specific point within the network. An example of a global title are digits dialed by the customer; in order for them to be correctly routed across an SS7 stack the SCCP translation function is needed.

Group Special Mobile (GSM): The European cellular network. Due to the use of SS7, the GSM is a more reliable network than its North American counterparts, which are still primarily analog in nature.

H

Home Location Register (HLR): A database which stores information about all cellular subscribers within a service provider's home service area. See **Visitor Location Register (VLR)**.

I

Institute of Electrical and Electronics Engineers (IEEE): Professional organization that defines network standards in a number of communication fields.

Integrated Service Digital Network (ISDN): Introduced in 1984, ISDN uses the existing infrastructure of the Public Switched Telephone Network (PSTN) to provide digital communication services for user-to-network interfaces.

Intelligent Networks Application Part (INAP): Protocol which separates switching from services in the SS7 network. Databases can be directly accessed in INAP, rather than through a Service Control Point (SCP).

International Standards Organization (ISO): Based in Geneva, Switzerland, the ISO establishes voluntary telecommunication technology standards among its ninety member countries.

ISDN User Part (ISUP): The protocol used to set up, manage, and release circuits used for voice and data transmission in the PSTN. ISUP uses out of band signalling, in which separate paths carry signalling and voice transmissions.

International Telecommunications Union - Telecommunications Standardization Sector (ITU - TSS, or more commonly ITU-T): Formerly the CCITT, this international body defines and implements recommendations and standards pertaining to the development of global telecommunications.

L

Line Information Database (LIDB): A database containing information relating to customer services, such as whether a customer subscribes to call waiting, caller identification, conference calls, and call forwarding. Also used for verification purposes with calling card services.

Load Distribution Function (LDF): Distributes traffic loads among various instances of the portable layer software (SCCP, TCAP, MTP3) based upon configuration parameters.

Local Exchange (LE): The primary switching node that provides access to the PSTN. In a basic telephony network, when a customer lifts the receiver they are connected to a local exchange, also known as a **Service Switching Point (SSP)**. The SSP provides the numbers of both the calling and called parties to a router, the **Signal Transfer Point (STP)**, so that the call can be completed across the network.

M

Media Access Control (MAC): Data link layer protocols that control traffic and data flow in multi-access channels. MACs are important in keeping LANs congestion free.

Message Transfer Part (MTP): Levels 1 through 3 of the SS7 protocol stack. Provides the upper levels with node-to-node transmission, message sequencing, and error detection/correction.

Mobile Application Part (MAP): A layer in the SS7 stack that runs on top of TCAP to query HLRs and VLRs within wireless networks. There are two standards for the MAP protocol, IS-41 and GSM. IS-41 is the ANSI standard and is used primarily in North America, while GSM is the ITU standard and is used in Europe, the U.S., and Asia.

Mobile Switching Center (MSC): Entity which receives signal strength reports from cell sites (antennas) regarding the particular strength or weakness of a cellular phone connection. Based upon these reports, the MSC determines the cell site that will then handle the given call. The MSC directly communicates with the BSC through digital facilities in the 64 Kbps range.

Multiprocessor Operating System (MOS): Trillium portable C source code designed to operate as an operating system on any embedded system or as a guest operating system under DOS, UNIX, or Solaris.

N

Network Service Part: The combination of Message Transfer Part (MTP) and Signalling Connection Control Part (SCCP).

O

Operations Administration and Maintenance (OAM): Non-data cells that provide the network with basic data management and performance diagnosis functions in order to prevent a catastrophic network or system failure.

Origination Point Code (OPC): A node ID that identifies the originating point of a message in a signalling network. See **Destination Point Code (DPC)**.

Open Systems Interconnection (OSI): An architectural system, developed by the ISO, for the interconnection of multiple data communication systems. The seven standardized layers of this model, with their associated layer managers, are: application, presentation, session, transport, network, data link, and physical. Each layer builds upon the services provided by the layers beneath it.

P

Peer Entities: Communicating entities residing in the same layer but within different nodes.

Protocol Data Unit (PDU): A unit of data used to exchange information between peer protocols communicating across a network, typically in the form of a packet with headers and/or trailers.

Protocol Specific Function (PSF): As stand-alone software, PSF provides fault tolerance functionality to portable protocol layers. In conjunction with Load Distribution Function (LDF), PSF also provides high availability functionality to the portable layer.

Protocol Specific Interface Function (PSIF): PSIF provides a generic interface from the underlying protocol layer to Trillium's Interworking Call Control (ICC) software. PSIF - ISUP understands the interface as implemented in the underlying protocol layer and maps it to a uniform interface required by ICC.

Public Switched Telephone Network (PSTN): Basic telephony system, through which calls are established and torn down between the user and a local exchange.

Q

QoS (Quality of Service): Quality of Service parameters center upon the working or contractual relationship between the service user and the service provider. These parameters, which are negotiated in advance, deal with the speed of the required service, the duration of the service, the rate of delivery, as well as desired or needed network characteristics, such as acceptable delay variations and errors in transmission.

S

Service Access Point (SAP): Information flow between the layers of a network is via Service Access Points (SAPs). The standardized interface of primitives and SAPs allows layers to be defined independently of each other. As long as the requirements of the layer interface are met, modifications may be made to the peer-to-peer protocol of one layer without affecting any upper or lower layer protocols.

Service Control Point (SCP): The interface between the STP and the database needed by the service user. The SCP is normally a computer connected to a mainframe computer that actually stores the needed information.

Service Switching Point (SSP): See **Local Exchange.**

Signal Transfer Point (STP): See **Local Exchange**

Signalling Connection and Control Protocol (SCCP): With TCAP, SCCP is part of layer 4 in the OSI protocol stack. In an SS7 network, SCCP provides end-to-end routing through the network of STPs. Unlike the layer below it (MTP 3), SCCP knows the entire route of the call. Connection Oriented SCCP is circuit-switched, while Connectionless SCCP is packet-switched.

Signalling Link: Abstract representation of the communications channel or connection between two logical nodes, including physical links and **Virtual Path Connections (VPCs)**.

Signalling System 7 (SS7): An ITU communication standard that first appeared in 1983, SS7 enables wireless and wireline call setup, call management, and call teardown over a digital signalling network. In addition to these basic functions, SS7 enables local number portability, the sharing of databases and connections within the PSTN, toll-free (800) and toll (900) wireline services, call forwarding, caller identification, and three-way calling.

Subsystem Number: Unique, fixed address given to a database, such as an HLR or VLR, used to route queries from SSPs through the network to the database itself.

T

Transaction Capabilities Application Part (TCAP): Layer 4 SS7 protocol that allows for remote database access (such as 800 or 900 numbers) from disparate networks using end-to-end switching.

Telephone User Part (TUP): The European equivalent to ISUP, except that TUP only supports analog circuits rather than digital circuits or data transmission.

V

Virtual Path Connection (VPC): A unidirectional concatenation of VPLs. Resources taken up in establishing individual VCCs are reduced or reserved by setting aside a given capacity within VPCs.

Visitor Location Register (VLR): Accesses the HLR through the SS7 network and stores information about subscribers who are outside of their home service area. See **Home Location Register (HLR)**.

Virtual Path Identifier (VPI): A distinct numerical marker, created within an 8-bit field in the ATM cell header, that identifies a particular VP for use by the cell.

Virtual Path Link (VPL): The unidirectional transmission of ATM cells within the life of a given VPI value—that is, from the assigning of the VPI at point A to its removal at point B.

References

Refer to the following documents for more information:

Fault-Tolerant/High-Availability (FT/HA) Core Functional Specification, Trillium Digital Systems, Inc. (p / n 1091133).

Fault-Tolerant/High-Availability (FT/HA) Core Service Definition, Trillium Digital Systems, Inc. (p / n 1092133).

ISDN User Part Functional Specification, Trillium Digital Systems, Inc. (p / n 1091029).

ISDN User Part Portation Guide, Trillium Digital Systems, Inc. (p / n 1093029).

ISDN User Part Service Definition, Trillium Digital Systems, Inc. (p / n 1092029).

ISDN User Part Software Test Sample, Trillium Digital Systems, Inc. (p / n 1094029).

ISDN User Part Training Course, Trillium Digital Systems, Inc. (p / n 1095029).

PSF - ISUP (FT/HA) Functional Specification, Trillium Digital Systems, Inc. (p / n 1091146).

PSF - ISUP (FT/HA) Service Definition, Trillium Digital Systems, Inc. (p / n 1092146).

PSF - ISUP (FT/HA) Training Course, Trillium Digital Systems, Inc. (p / n 1095146).

PSIF-ISUP Functional Specification, Trillium Digital Systems, Inc. (p / n 1091141).

PSIF-ISUP Service Definition, Trillium Digital Systems, Inc. (p / n 1092141).

PSIF - ISUP Training Course, Trillium Digital Systems, Inc. (p / n 1095141).

SIT Interface Service Definition, Trillium Digital Systems, Inc. (p / n 1100010)

System Services Interface Service Definition, Trillium Digital Systems, Inc. (p / n 1111011).

Index