

**Intelligent Network (IN);
Intelligent Network Application Protocol (INAP);
Capability Set 2 (CS2);
Part 3: Test Suite Structure and Test Purposes (TSS&TP)
specification for Service Switching Function (SSF);
Sub-part 1: Basic capability set of CS-1
including CS-2 complements**



Reference

DEN/SPS-03038-3-1

Keywords

CS1, CS2, IN, INAP, TSS&TP, SSF

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet

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

Important notice

This ETSI deliverable may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2000.
All rights reserved.

Contents

Intellectual Property Rights	5
Foreword	5
1 Scope	6
2 References	6
3 Definitions and abbreviations	6
3.1 Definitions	6
3.2 Abbreviations	7
4 Test Purpose generalities	7
4.1 Introduction	7
4.2 Grouping of test purposes per elementary procedures	7
4.3 Source of test purpose definitions	8
4.4 Method used for developing TPs	8
4.4.1 Use of MSCs generated by the SDL model of Core INAP CS-2	8
4.4.2 TCAP adapter primitives	8
4.4.3 Generation of corresponding Test Cases	8
4.5 Method used for TP description	9
4.5.1 Text and MSCs	9
4.5.2 Test categories	9
4.5.3 Test purpose naming convention	10
4.5.4 Preambles and their naming conventions	10
4.5.5 How to interpret the parameters and their values as used in the MSCs	11
5 Functional configurations under test	12
5.1 SSF basic functions	12
5.2 SSF additional functions	13
6 TSS and TPs for CS-1 and CS-2 basic capabilities	14
6.1 Preambles and postambles used	14
6.1.1 List of preambles and postambles for CS1	14
6.1.2 Preamble descriptions	15
6.1.2.1 O_OS preamble	15
6.1.2.2 O_S2P preamble	16
6.1.2.3 T_OS preamble	17
6.1.2.4 T_S2P preamble	18
6.1.3 Postamble descriptions	19
6.1.3.1 SigConA_Release postamble	19
6.1.3.2 SigConA_Release_thenB postamble	19
6.1.3.3 ReleaseCallA postamble	20
6.1.3.4 ReleaseICA postamble	20
6.1.3.5 ReleaseCallAB_cause_00 postamble	21
6.1.3.6 ReleaseCallAB_cause_0F postamble	22
6.1.3.7 SigConB_Release postamble	23
6.1.3.8 SigConB_Release_cause_0D postamble	23
6.1.3.9 SigConA_Release_thenB_cause10 postamble	24
6.1.3.10 ReleaseCallB postamble	25
6.1.3.11 ReleaseCallA2 postamble	26
6.2 Basic procedures	26
6.2.1 List of procedures	26
6.2.2 Definitions of the procedures	27
6.3 Structure of the test suite (TSS) for the basic capabilities	28
6.4 Test Purposes (TP) description	30
6.4.1 ServiceFiltering procedure	30
6.4.2 ActivityTest procedure	40
6.4.3 ApplyCharging procedure	44

6.4.4	CallGap procedure	57
6.4.5	CallInformation procedure	81
6.4.6	Cancel procedure	96
6.4.7	CollectInformation procedure	106
6.4.8	Connect procedure	111
6.4.9	Continue procedure	126
6.4.10	FurnishChargingInformation procedure	129
6.4.11	InitialDP procedure	133
6.4.12	InitiateCallAttempt procedure	142
6.4.13	ReleaseCall procedure	152
6.4.14	RequestReportBCSMEEvent procedure	157
6.4.15	SendChargingInformation procedure	222
6.4.16	RequestNotificationChargingEvent procedure	223
Annex A (informative):	Description of various functional configurations	230
Annex B (normative):	Parameter values used in MSCs for CORE INAP primitives	247
Annex C (normative):	Parameter values used in MSCs for TCAP primitives	249
Bibliography		250
History		251

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This European Standard (Telecommunications series) has been produced by ETSI Technical Committee Services and Protocol for Advanced Networks (SPAN), and is now submitted for the Voting phase of the ETSI standards Two-step Approval Procedure.

The present document is part 3, sub-part 1 of a multi-part EN covering the Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 2 (CS2), as identified below:

Part 1: "Protocol specification";

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

Part 3: "Test Suite Structure and Test Purposes (TSS&TP) specification for Service Switching Function (SSF)";

Sub-part 1: "Basic capability set of CS-1 including CS-2 complements";

Sub-part 2: "Call Party Handling (CPH)";

Sub-part 3: "Specialized Resource Functions (SRF)";

Part 4: "Abstract Test Suite (ATS) specification and Partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma for Service Switching Function (SSF)";

Part 5: "Distributed Functional Plane (DFP) [ITU-T Recommendation Q.1224 (1997), modified]".

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

1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS&TP) for testing of the Service Switching Function (SSF) and the Specialized Resource Function (SRF) of the Intelligent Network Application Protocol (INAP) of Intelligent Network (IN) Capability Set 2 (CS2) according to EN 301 140-1 [1].

The present document relates to the basic capability set, which covers the CS-1 operations, plus the CS-2 additions related to these operations, mainly due to the test of the CS-2 additional parameters or functionalities.

The present document is completed by other parts constituting the CS-2 Core INAP specifications.

In the present version of the TP description included in tables, references to specification requirements and references to PICS in the "condition for selection" are not included, except to mention when it is a CS-2 addition.

ISO/IEC 9646-1 [3] and ISO/IEC 9646-2 [4] are used as the basis for the test methodology.

2 References

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

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

- [1] ETSI EN 301 140-1 (V1.3): "Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 2 (CS2); Part 1: Protocol specification".
- [2] ETSI EN 301 140-4 (V1.1): "Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 2 (CS2); Part 4: Abstract Test Suite (ATS) specification and Partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma for Service Switching Function (SSF)".
- [3] ISO/IEC 9646-1: "Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 1: General concepts".
- [4] ISO/IEC 9646-2: "Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 2: Abstract test suite specification".
- [5] ETSI ETS 300 374-1: "Intelligent Network (IN); Intelligent Network Capability Set 1 (CS1); Core Intelligent Network Application Protocol (INAP); Part 1: Protocol specification".

3 Definitions and abbreviations

3.1 Definitions

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

- terms defined in EN 301 140-1 [1];
- terms defined in ISO/IEC 9646-1 [3] and in ISO/IEC 9646-2 [4].

In particular, the following terms defined in ISO/IEC 9646-1 [3] apply:

- Abstract Test Suite (ATS);
- Implementation Under Test (IUT);
- System Under Test (SUT);
- Protocol Implementation Conformance Statement (PICS).

3.2 Abbreviations

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

ATS	Abstract Test Suite
BI	Invalid Behaviour tests
BO	Inopportune Behaviour tests
BV	Valid Behaviour tests
CA	Capability tests
CS	Call Segment
CS	Capability Set
EDP-R	Event Detection Point - Request
FSM	Finite State Machine
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
IP	Intelligent Peripheral
iS	initiating SSF
iSSP	initiating SSP
IUT	Implementation Under Test
MSC	Message Sequence Chart
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
SCF	Service Control Function
SCP	Service Control Point
SDF	Service Data Function
SDL	Specification and Description Language
SRF	Specialized Resource Function
SSF	Service Switching Function
SSP	Service Switching Point
SUT	System Under Test
TCAP	Transaction Capabilities Application Part
TP	Test Purpose
TSS	Test Suite Structure

4 Test Purpose generalities

4.1 Introduction

A TP is defined for one or several conformance requirements to be tested. Each TP will result in a test case keeping the same name, specified in the ATS.

4.2 Grouping of test purposes per elementary procedures

The Test Purposes are grouped by elementary procedures. A procedure groups elementary INAP operations which it is possible to test together. For each elementary procedure, are defined: how to invoke it; and what are the possible return results and return error(s) at the INAP interface.

NOTE: Some have no results at all at this INAP interface. In these cases, and to have a "visible" result, the PCO will be at the signalling control interface.

4.3 Source of test purpose definitions

The test purposes are based on the requirement documented in EN 301 140-1 [1].

4.4 Method used for developing TPs

4.4.1 Use of MSCs generated by the SDL model of Core INAP CS-2

The SDL model of INAP CS-2 is specified with object oriented SDL (SDL'92) and specifies the behaviour of the SSF. The CS-2 specification inherits the CS-1 and specifies the whole of CS-1 and CS-2. The SDL specification is the normative specification of the INAP behaviour and is contained in annex A of EN 301 140-1 [1].

The SDL model specifies precisely and unambiguously the behaviour of and the interworking between the different functional entities of the SSF. The external interfaces of the SDL model are two signalling control interfaces (SigConA and SigConB) carrying abstract primitives, and the INAP interfaces to the SCF. Mappings are provided from SigConA and SigConB to DSS.1 and ISUP. The behaviour of the SDL model thus resembles an SSP, and can be used for service emulation and the development of test purposes and test cases. MSCs delivered by this SDL model are used in the TP definition and are provided in addition to the descriptive text.

The development of the test purposes (TP) is done in two steps:

- a) the descriptive text is created together with a rough MSC defined by hand. It illustrates the basic behaviour in MSC-like form which is expected from the IUT. The rough MSC does not contain all the constraints in detail. The description makes reference to a preamble and a postamble;
- b) a detailed MSC is developed by simulation:
 - 1) system level MSC for Autolink (the tool used to automatically generate the TTCN test cases based on the MSCs and the SDL model);
 - 2) MSC for documentation of the TPs.

The reason for developing the detailed MSC by simulation is that it can be done step by step while the SDL model prompts the developer for the correct options and parameters.

The MSCs identify the different entities (SSF, SCF, SigCon A and B) involved in a given configuration and shows the different components used for a test, in term of the IUT (representing the SSF for instance) and the testers (representing the SCF and the SigCon A, B or C).

4.4.2 TCAP adapter primitives

In addition to showing the INAP protocol, and in order to ease the implementation of the test suite, the MSCs show the TCAP adapter primitives such as TC begin, TC continue, TC invoke and TC end and show using standard abbreviations the INAP operations which are embedded in the TCAP primitive, together with the operation arguments.

4.4.3 Generation of corresponding Test Cases

Using Computer Aided Test Generation techniques, TTCN test cases can be automatically generated from the SDL model. It is also possible to verify manually developed test cases against the SDL model. The clear separation of CS-1 and CS-2 in the SDL model makes it possible to use it for both CS-1 and CS-2 test case development.

4.5 Method used for TP description

4.5.1 Text and MSCs

In general, a TP is described using text presented in a table followed by an MSC.

The table describing each TP is as follows:

Table 1a

(TP name, also corresponding test case name)	
Test Purpose:	
Requirement Ref:	
Selection Cond:	
Test preamble:	
Test description	
Pass criteria:	
Test postamble	

In addition to the TP name and a reference to the specification requirement, the table contains a short title of the test purpose, the condition to select and run this test case (expression in terms of PICS references), the name of the test preamble, information on the test body including for instance details on parameters which do not appear in the companion MSC, the pass criteria for a successful test and the name of the test postamble.

The MSC which follows the TP description describes the test body, as the preambles and postambles are mostly defined by a single line in the MSC.

4.5.2 Test categories

Capability tests (CA)

Capability testing provides a limited testing to ascertain the capabilities stated in the PICS can be observed.

Valid Behaviour tests (BV)

Predefined state transitions are considered as valid. The test purposes in the valid behaviour test sub group cover as far as reasonable the verification of the normal and exceptional procedures of the various Finite State Machines (FSMs), i.e. a valid behaviour test is a test where the message sequence and the message contents is considered as valid.

Invalid Behaviour tests (BI)

This test sub group is intended to verify that the IUT is able to react properly having received an invalid Protocol Data Unit (PDU). An invalid PDU is defined as a syntactically incorrect message.

Inopportune Behaviour tests (BO)

This test group is intended to verify that the IUT is able to react properly in the case an inopportune protocol event occurring. Such an event is syntactically correct but occurs when it is not expected, e.g. a correctly coded operation is received in a wrong state (the IUT may respond by sending error UnexpectedComponentSequence).

4.5.3 Test purpose naming convention

The identifier of the TP is built according to the scheme in table 1b.

Table 1b: TP identifier naming convention scheme

Identifier:	IN2_<i>_<sss>_<pp>_<cc>_<nn>		
IN2	indicates IN Capability Set 1 and 2 (CS-1 being in CS-2)		
<i>	=	interface:	A SSF-SCF interface B SSF-SRF interface C SCF-SCF interface
<sss>	=	common set	BASIC Basic set for CS-1 complemented for CS-2 CPH Call Party Handling from Capability Set 2 CTM Cordless Terminal Portability from Capability Set 2
<pp>	=	procedure name like	SF ServiceFiltering
<cc>	=	test category:	CA Capability tests BV Valid Behaviour tests BI Invalid Behaviour tests BO Inopportune Behaviour tests
<nn>	=	sequential number:	(01-99)
Example of test purpose and test case name: IN2_A_BASIC_SF_BV_02			

4.5.4 Preambles and their naming conventions

Preambles are used to bring the IUT from the initial state to the state where the test takes place. In the CS-2 scheme, the set of the preambles forms a tree, which means that in order to reach the state created by preamble P3, it is necessary to execute preamble P1 followed by preambles P2 then P3.

The naming convention used reflects the description of the connection view set by executing the preamble, in terms of nature of the legs per Call Segment (CS), starting from the stable legs then the ones on hold then the ones in transfer, with the indication of the number of legs, while the first letter indicates how this configuration was initiated.

The general form is:

a_[stableLegsParty or onHold (legs) or transfer(legs) for CallSegment 1]_[idem for CallSegment2]_[idem for CallSegment 3]

where:

a is letter:

- O for Originating (outgoing call for a user);
- T for Terminating (incoming call for a user);
- I for Initiate Call Attempt (initiated from the network).

The state names and their abbreviations used are:

Null

1 Party	1P
Originating Set-up	OS
Terminating Set-up	TS
Terminating 1 Party Set-up	T1P
Stable 1 Party	S1P
Stable 2 Party	S2P
Transfer (no. of passive legs)	TF(x)
On Hold (no. of passive legs)	OH(x)
Stable MultiParty (no. of passive legs)	S(x)P

The term "null" stands for "none" as in preamble O_NULL_S2P_OH3.

There can be two set of CSs with the same nature of legs present at the same time, as in the preamble name O_S2P_OH2_OH3.

4.5.5 How to interpret the parameters and their values as used in the MSCs

The MSCs show the exchanges of PDUs of the TCAP protocol, as well as the Core INAP protocol. PDUs of both protocols use parameters.

The list of the parameters for the Core INAP protocol is given in reference ETS 300 374-1 [5].

The list of parameters for the TCAP protocol is recalled here for each TCAP primitives. Note that only mandatory parameters are used.

TCAP primitives from SCF to TCAP:

```
TC_InvokeReq (InvokeID, DialogueID, Class, OperationCode, Timeout);
TC_BeginReq ( DialogueID, OriginatingAddress);
TC_ContinueReq ( DialogueID, OriginatingAddress);
TC_EndReq ( DialogueID, Termination);
TC_AbortReq ( DialogueID).
```

TCAP primitives from TCAP to SCF:

```
TC_InvokeInd (InvokeID, DialogueID, Class, OperationCode, LastComponent);
TC_BeginInd ( DialogueID, OriginatingAddress, ComponentPresent);
TC_ContinueInd ( DialogueID, OriginatingAddress, ComponentPresent);
TC_EndInd ( DialogueID, Termination, ComponentPresent);
TC_AbortInd ( DialogueID);
TC_ErrorInd (InvokeID, DialogueID, ErrorCode, LastComponent);
TC_ReturnResultInd (InvokeID, DialogueID, LastComponent, OperationCode, OperationArg);
```

TC_RejectInd (InvokeID, DialogueID).

The values of these parameters are either mandatory and imposed by the specifications, or they are informative only and chosen arbitrarily in ranges compatible with the specifications.

The list of the informative parameters, for which a value is to be assigned in particular for the execution of a test suite, is included in the PIXIT proforma. See reference EN 301 140-4 [2].

Annex B and Annex C of the present document contain a copy of the PIXIT proforma parameter tables of respectively the Core INAP and the TCAP protocols. These proforma tables are filled up and contain the parameter values used for the definition of the MSCs and TPs.

The preamble T_OS (and all preambles and test cases which use this preamble) contains reference to an ASP Mgt_SetTriggerTable. This does not exist in the protocol, but in the SDL model it identifies which Trigger Detection points need to be set before commencing the test case.

5 Functional configurations under test

5.1 SSF basic functions

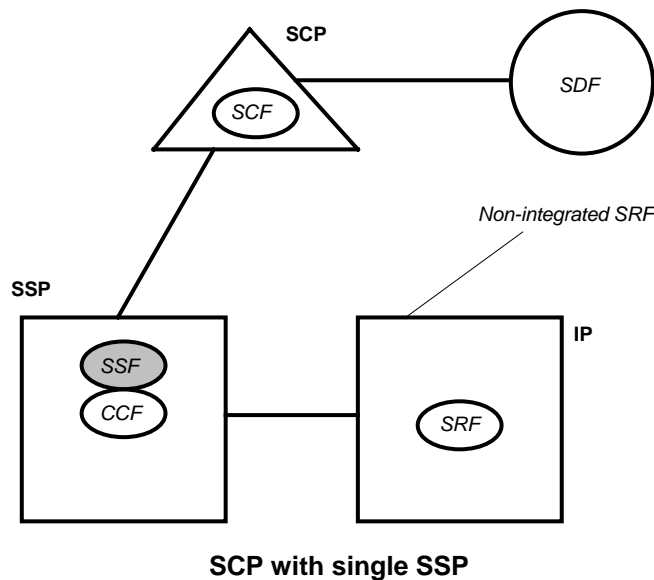


Figure 1: Configuration 1: IUT= SSF (non-integrated with SRF)

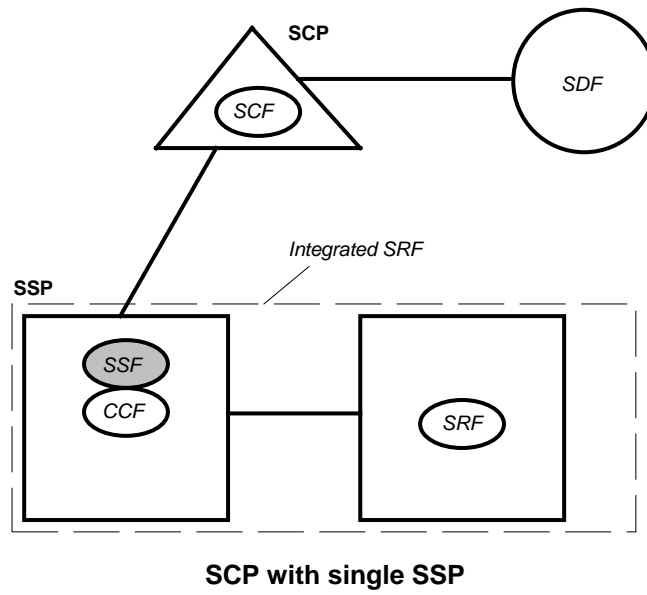


Figure 2: Configuration 2: IUT= SSF (integrated with SRF)

5.2 SSF additional functions

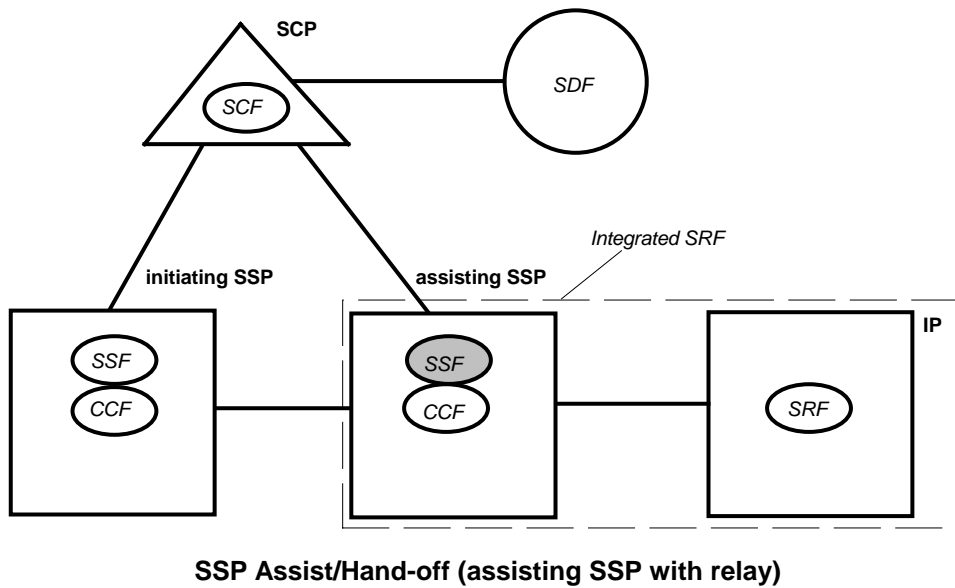


Figure 3: Configuration 3: IUT= SSF of assisting SSP (integrated SRF)

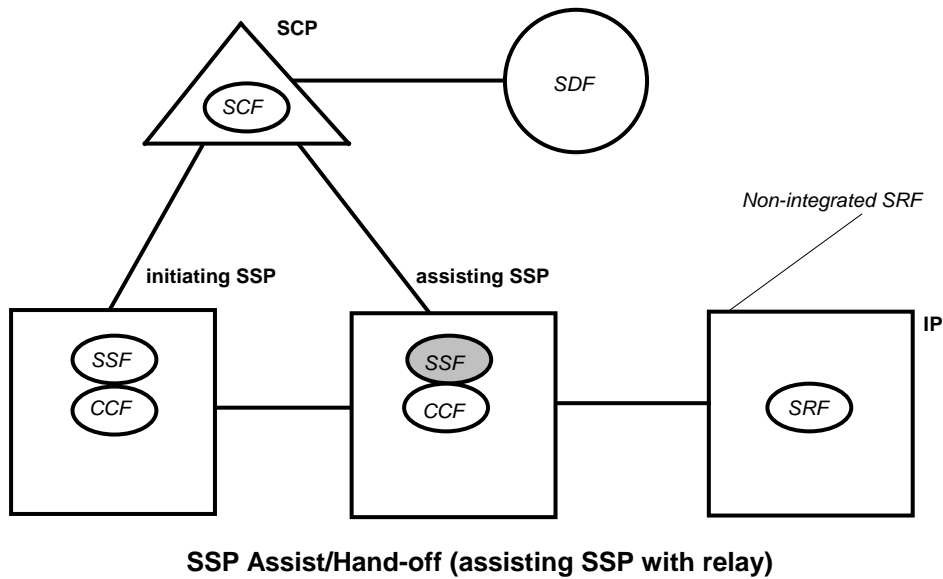


Figure 4: Configuration 4: IUT= SSF of assisting SSP (non integrated SRF)

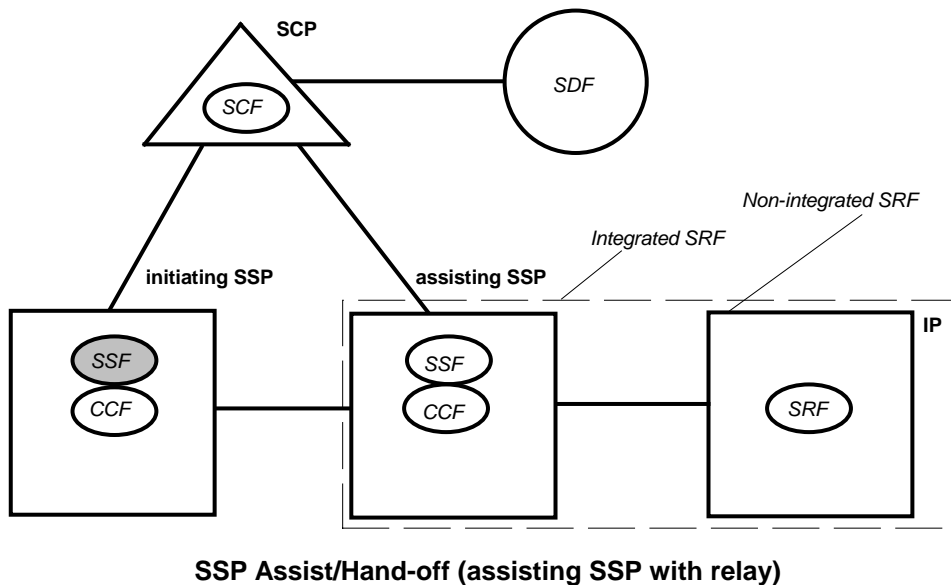


Figure 5: Configuration 5: IUT= SSF of initiating SSP

6 TSS and TPs for CS-1 and CS-2 basic capabilities

6.1 Preambles and postambles used

6.1.1 List of preambles and postambles for CS1

Here is a list of preambles used in the Basic CS-1 and CS-2 capabilities part:

O_OS;

O_S2P;

T_OS;

T_S2P.

Here is a list of postambles used in the Basic CS-1 and CS-2 capabilities part:

SigConA_Release;

SigConA_Release_thenB;

ReleaseCallA;

ReleaseICA;

ReleaseCallAB_cause_00;

ReleaseCallAB_cause_0F;

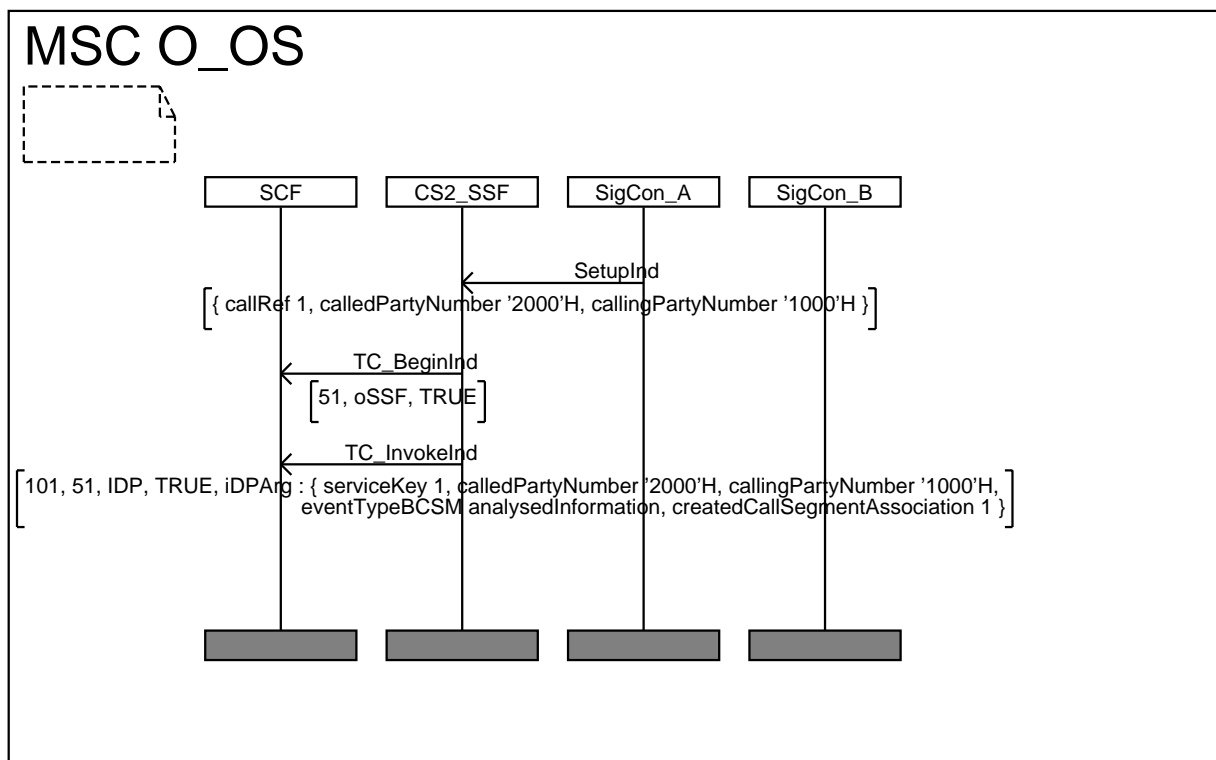
SigConB_Release.

More preambles and postambles are defined for the complete CS-2. See clause 7.

6.1.2 Preamble descriptions

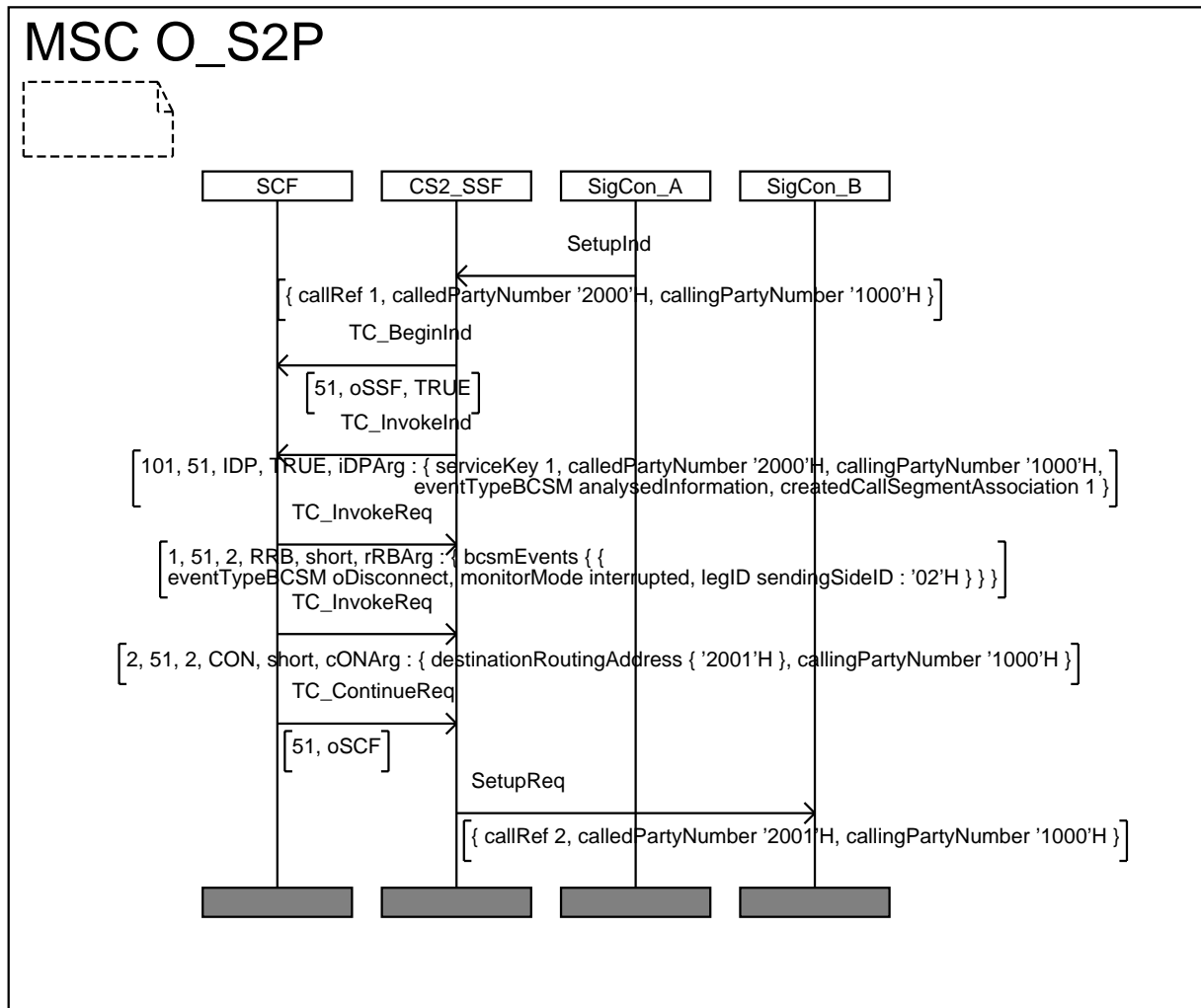
6.1.2.1 O_OS preamble

This preamble is used to bring the IUT from the idle or Null state to the 1 party state.

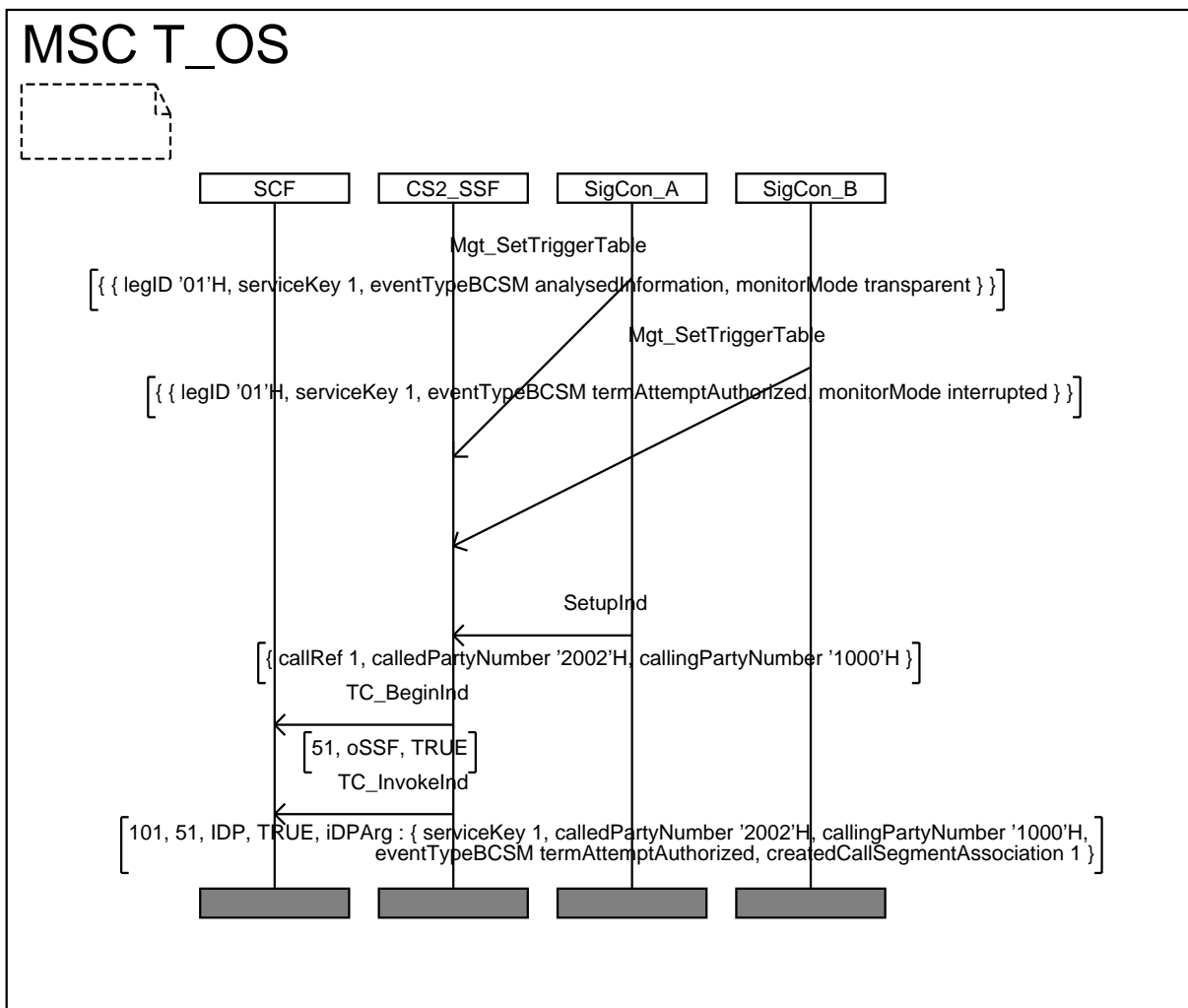


6.1.2.2 O_S2P preamble

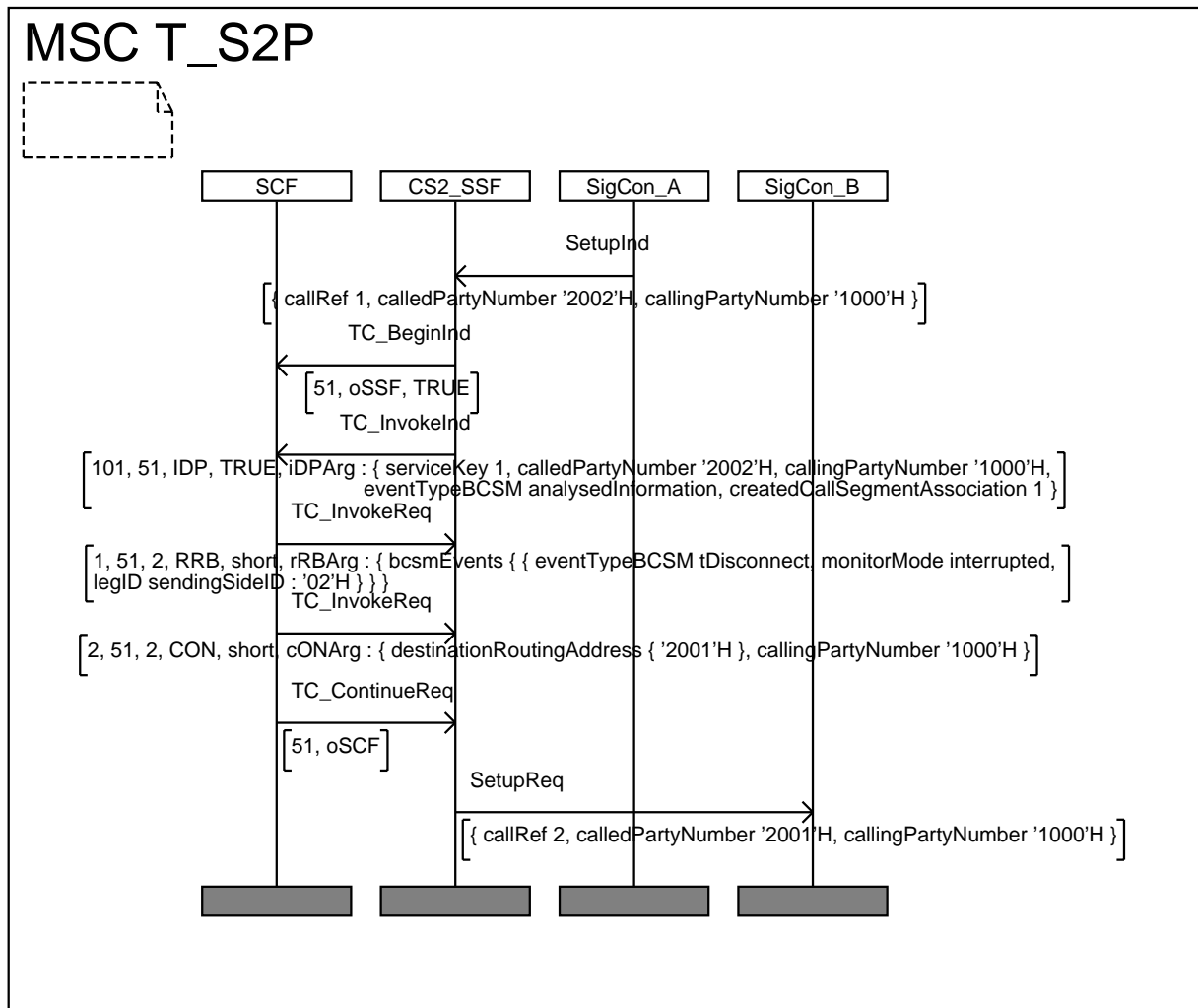
This preamble is used to bring the IUT from the idle or Null state to the 2 party state.



6.1.2.3 T_OS preamble



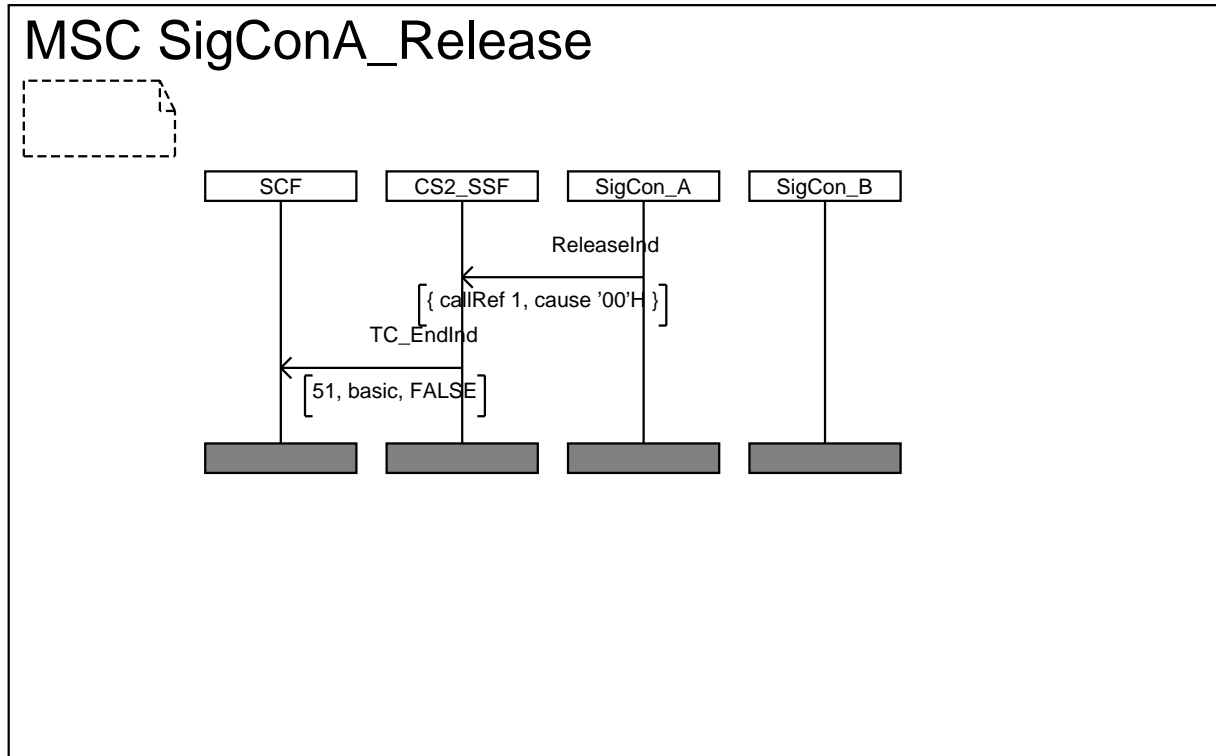
6.1.2.4 T_S2P preamble



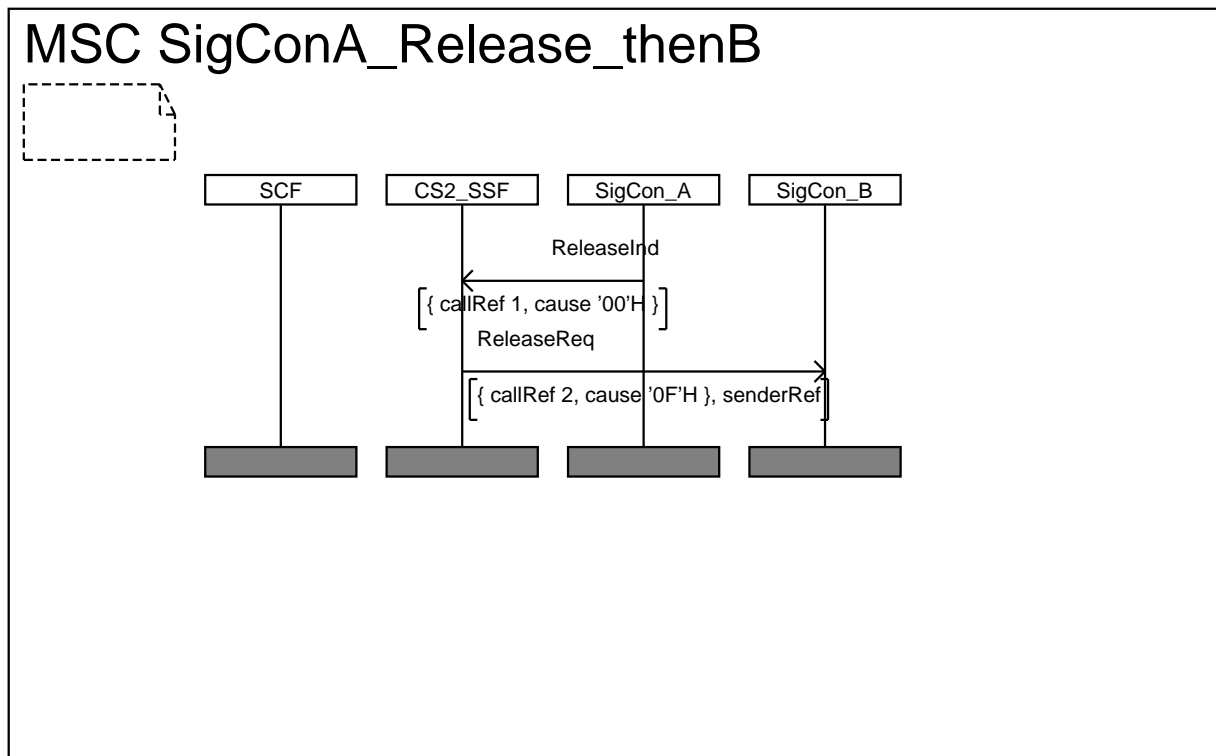
6.1.3 Postamble descriptions

Postambles are used to bring the IUT from the state where the test ends, back to the initial state.

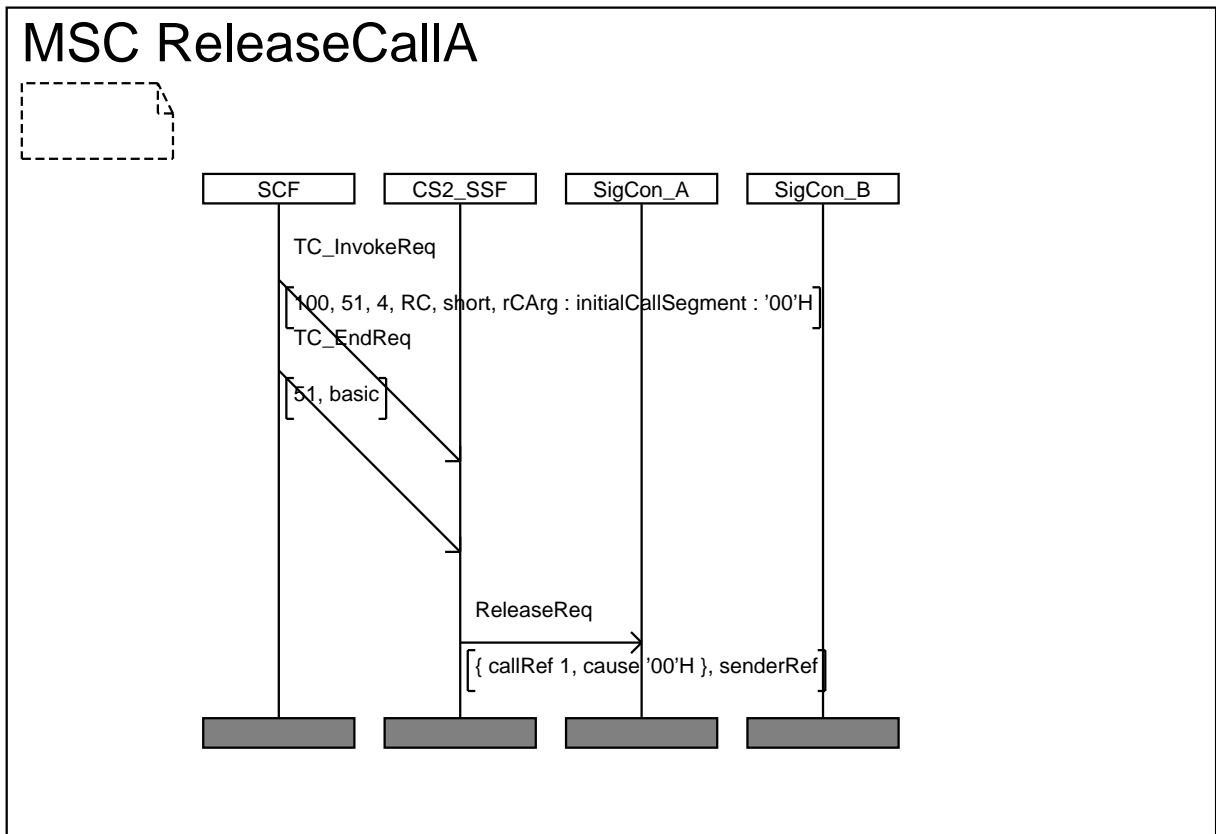
6.1.3.1 SigConA_Release postamble



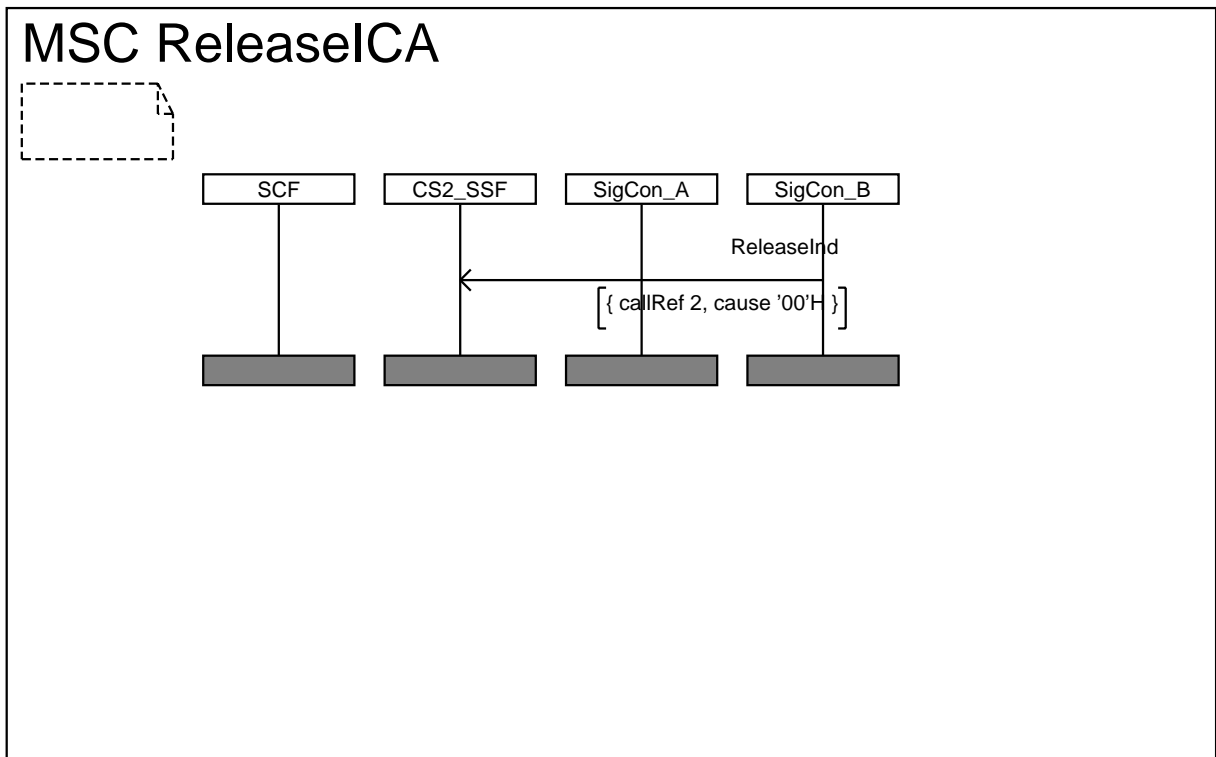
6.1.3.2 SigConA_Release_thenB postamble



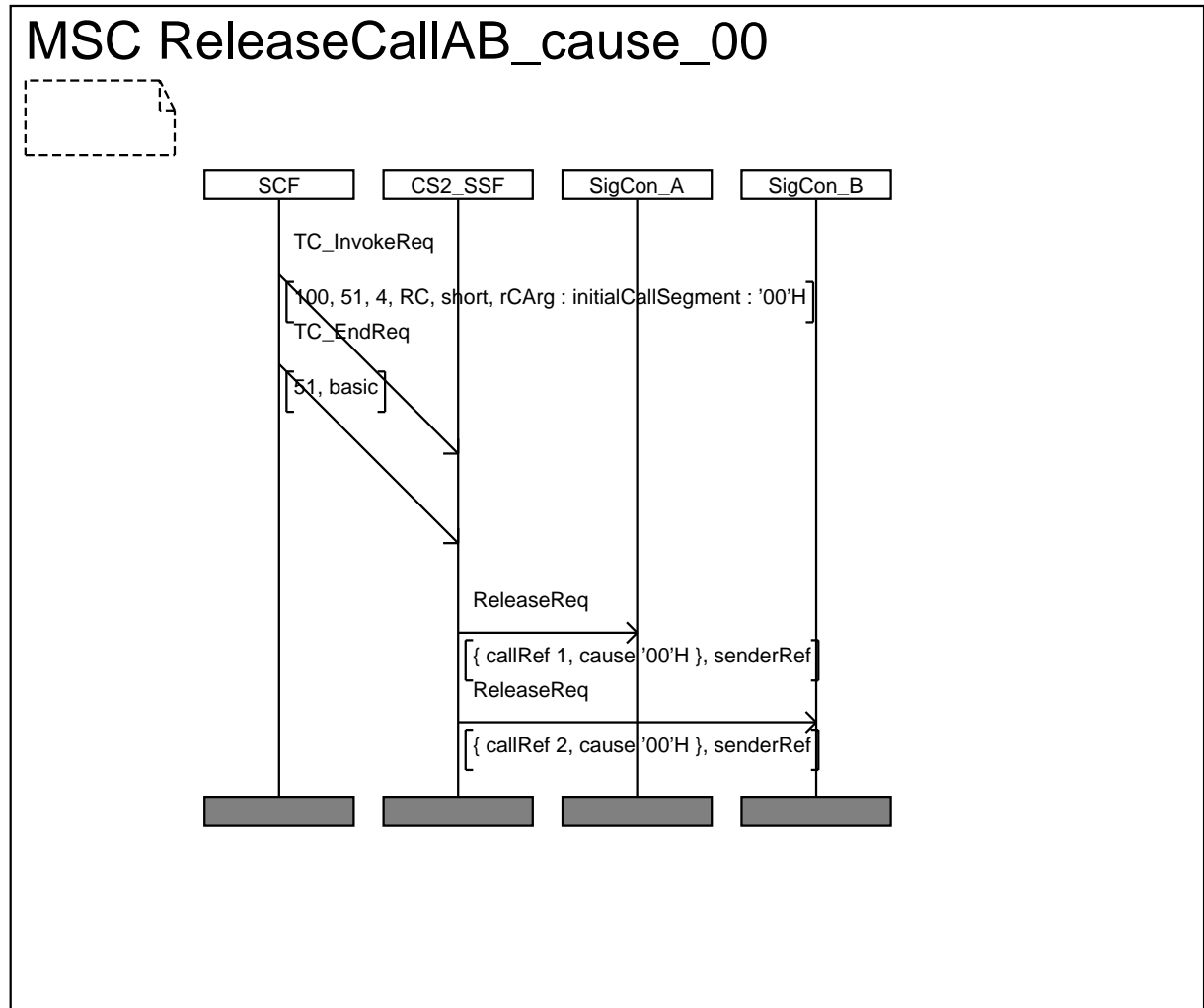
6.1.3.3 ReleaseCallA postamble



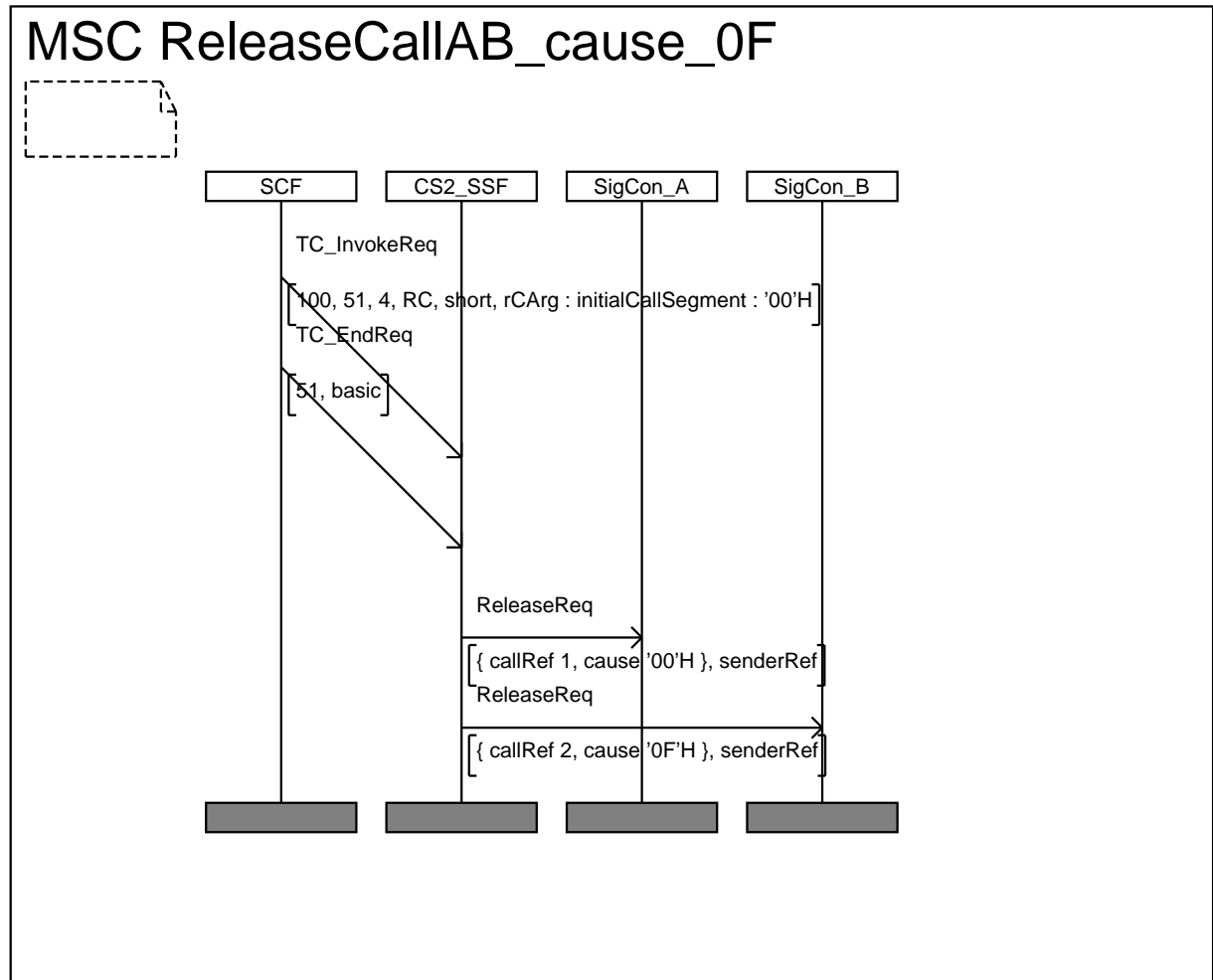
6.1.3.4 ReleaseICA postamble



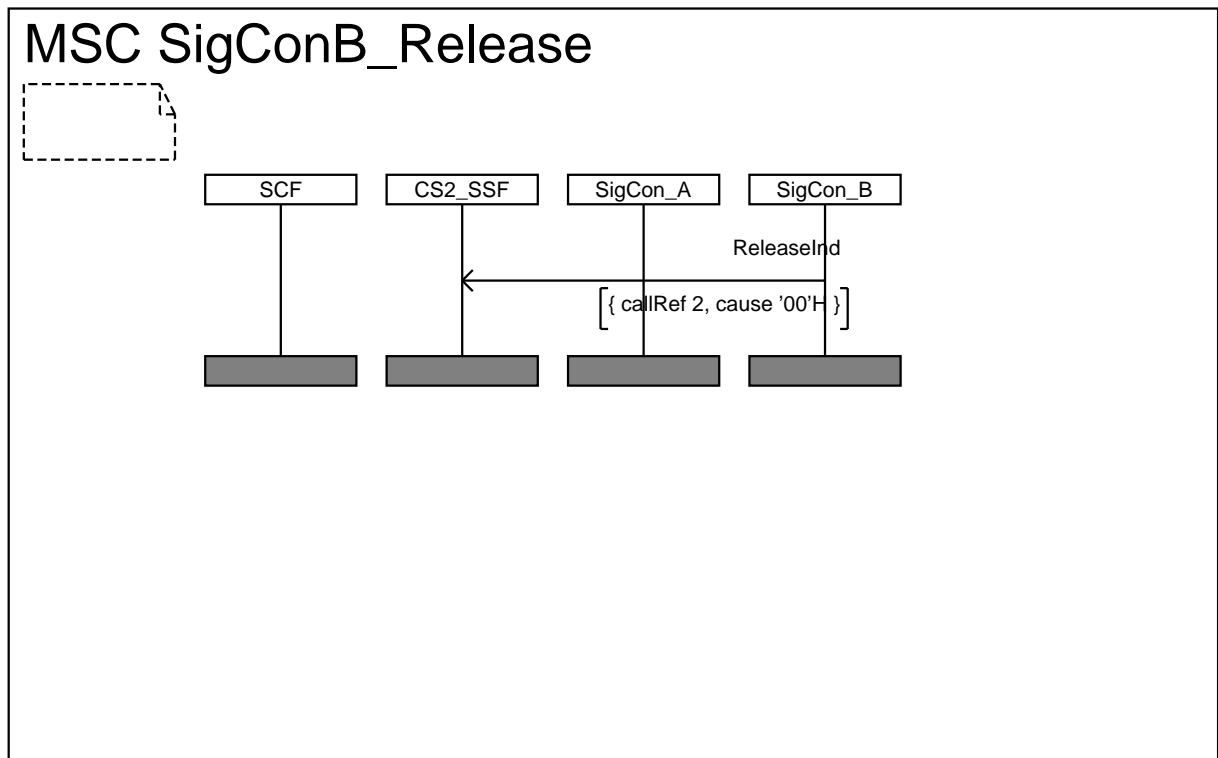
6.1.3.5 ReleaseCallAB_cause_00 postamble



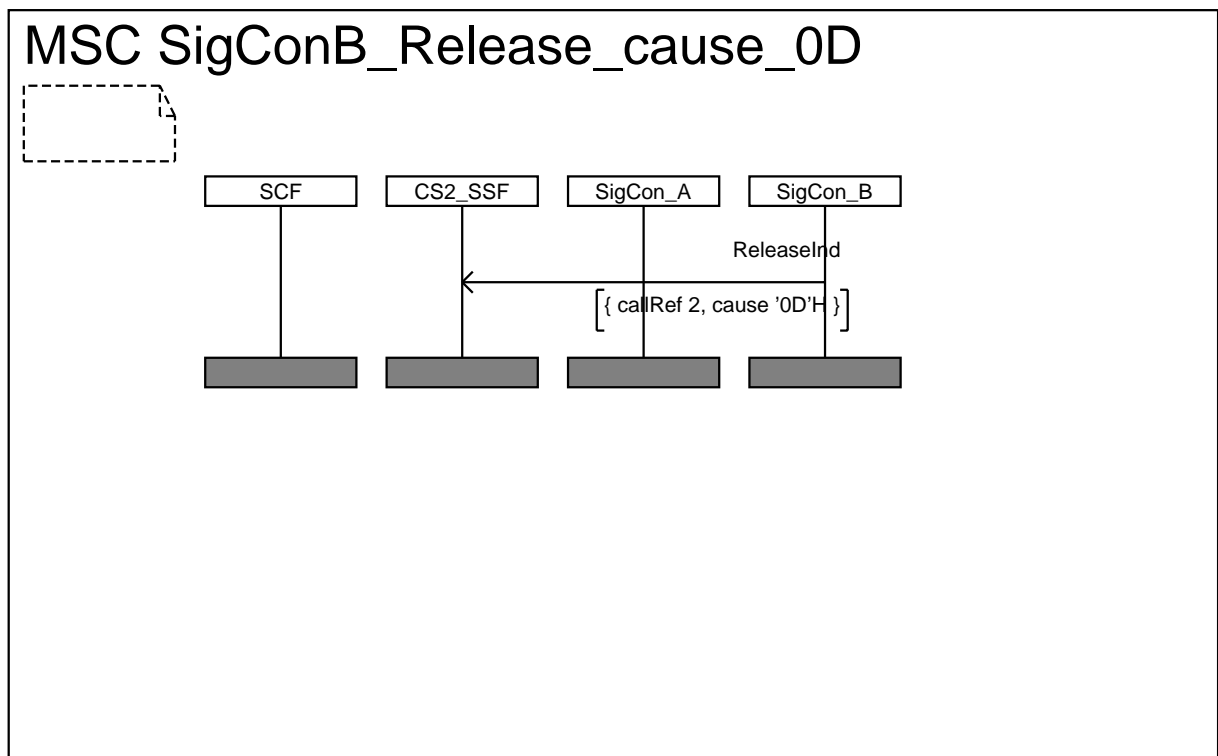
6.1.3.6 ReleaseCallAB_cause_0F postamble



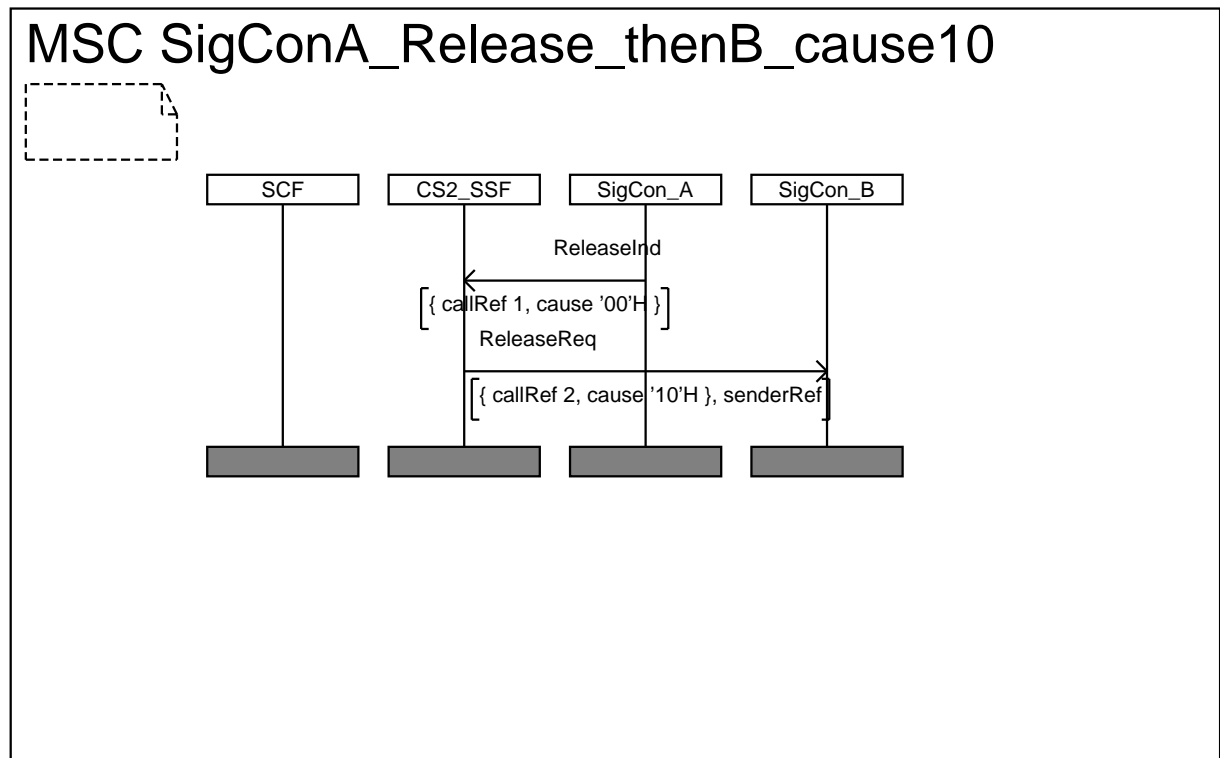
6.1.3.7 SigConB_Release postamble



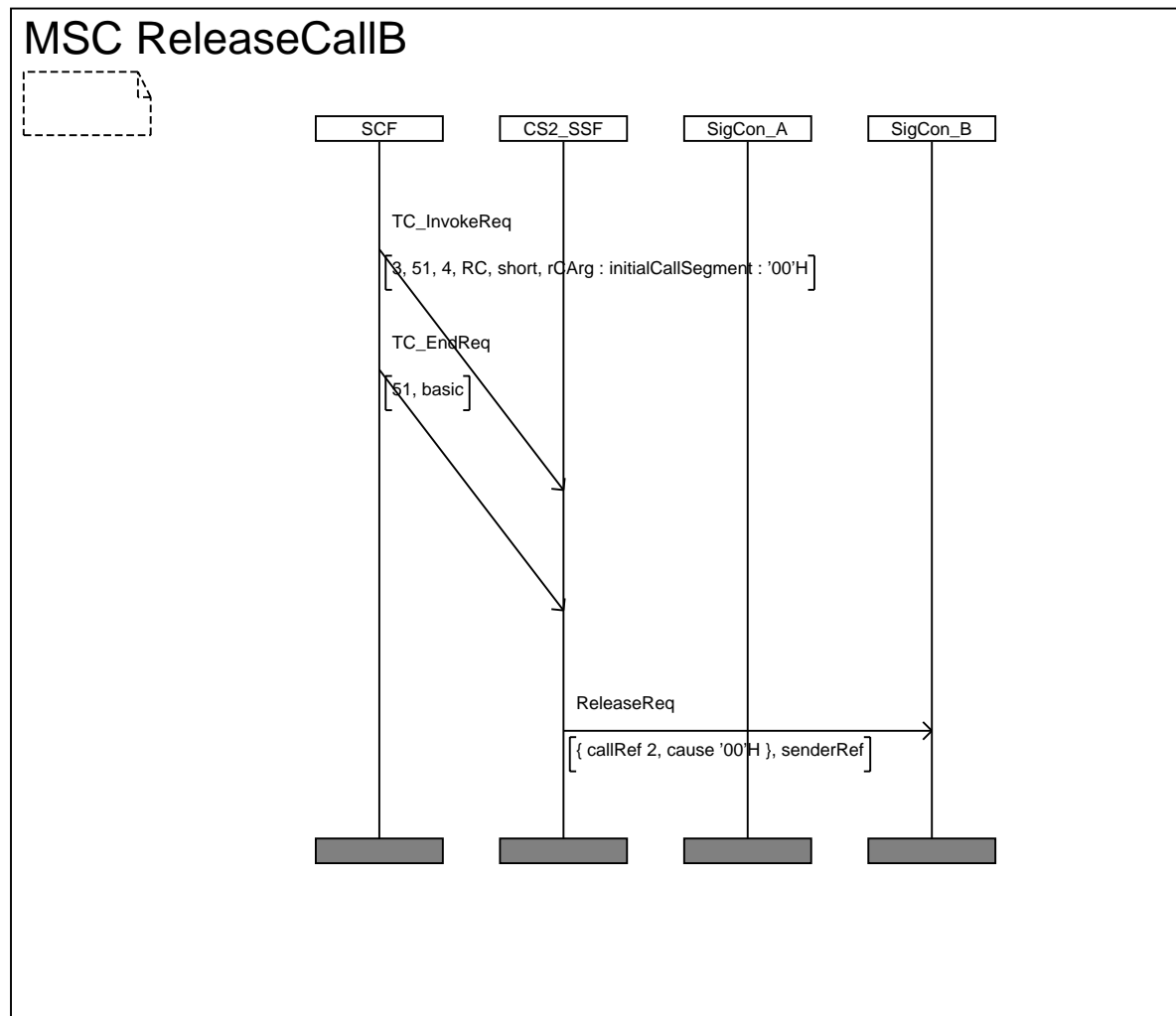
6.1.3.8 SigConB_Release_cause_0D postamble



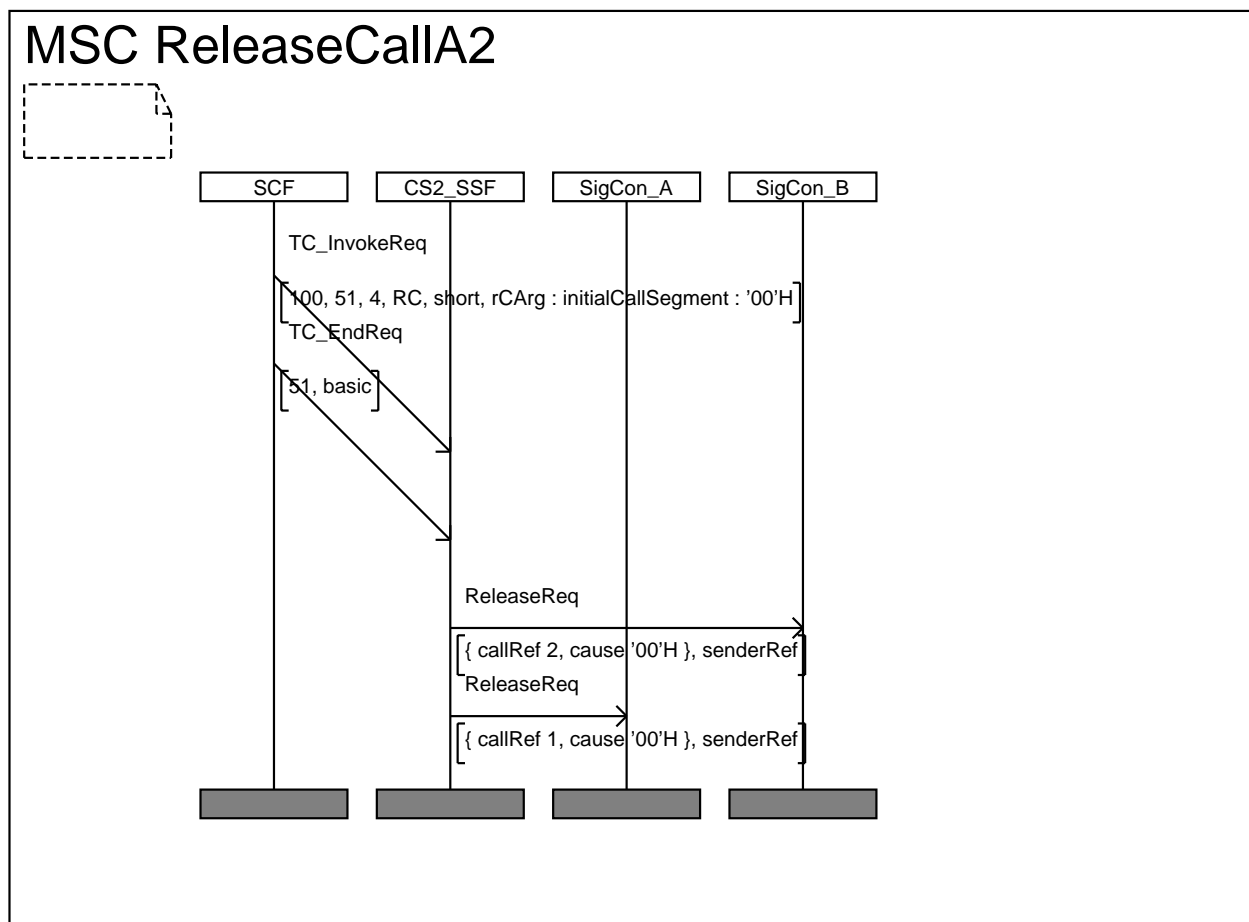
6.1.3.9 SigConA_Release_thenB_cause10 postamble



6.1.3.10 ReleaseCallB postamble



6.1.3.11 ReleaseCallA2 postamble



6.2 Basic procedures

6.2.1 List of procedures

The Test Purposes for Basic CS-1 and CS-2 functionalities are grouped according to the following procedures:

NOTE: The acronyms below are names given to a procedure (example: SF for Service Filtering), and may not be in line with the standardized ones used for invoking such a procedure (example: ASF for Activate Service Filtering). These acronyms are made of two letters only and are used when giving a name to a Test Purpose or a test case.

SF	ServiceFiltering
AT	ActivityTest
AC	ApplyCharging
AR	AssistRequestInstructions
CG	CallGap
CF	CallInformation
CA	Cancel
CI	CollectInformation
CO	Connect
CR	ConnectToResource
CU	Continue (no specific Test Purpose)
DF	DisconnectForwardConnection
EC	EstablishTemporaryConnection
FC	FurnishChargingInformation
DP	InitialDP
IC	InitiateCallAttempt

PA	PlayAnnouncement (no specific Test Purpose)
PC	PromptAndCollectUserInformation
RC	ReleaseCall
RN	RequestNotificationChargingEvent
RR	RequestReportBCSMEvent
SC	SendChargingInformation

6.2.2 Definitions of the procedures

ServiceFiltering procedure (SF)

Invoke: ActivateServiceFiltering
 Return Result: ServiceFilteringResponse
 ActivateServiceFiltering
 Return Error: ActivateServiceFiltering

ActivityTest procedure (AT)

Invoke: ActivityTest
 Return Result: ActivityTest
 Return Error: None

ApplyCharging procedure (AC)

Invoke: ApplyCharging
 Return Result: ApplyChargingReport
 Return Error: ApplyCharging
 ApplyChargingReport

AssistRequestInstructions procedure (AR)

Invoke: AssistRequestInstructions
 Return Result: None
 Return Error: AssistRequestInstructions

CallGap procedure (CG)

Invoke: CallGap
 Return Result: None
 Return Error: None

CallInformation procedure (CF)

Invoke: CallInformationRequest
 Return Result: CallInformationReport
 Return Error: CallInformationRequest

Cancel procedure (CA)

Invoke: Cancel
 Return Result: PlayAnnouncement(Error)
 PromptAndCollectUserInformation(Error)
 Return Error: Cancel

CollectInformation procedure (CI)

Invoke: CollectInformation
 RequestReportBCSMEvent
 Return Result: EventReportBCSM
 Return Error: CollectInformation
 RequestReportBCSMEvent

Connect procedure (CO)

Invoke: Connect
 Return Result: None
 Return Error: Connect

ConnectToResource procedure (CR)

Invoke: ConnectToResource
 Return Result: None
 Return Error: ConnectToResource

Continue (CU)

Invoke: Continue
 Return Result: None
 Return Error: None

DisconnectForwardConnection procedure (DF)

Invoke: DisconnectForwardConnection
 Return Result: None
 Return Error: DisconnectForwardConnection

EstablishTemporaryConnection procedure (EC)

Invoke: EstablishTemporaryConnection
 Return Result: None
 Return Error: EstablishTemporaryConnection

FurnishChargingInformation procedure (FC)

Invoke: FurnishChargingInformation
 Return Result: None
 Return Error: FurnishChargingInformation

InitialDP procedure (DP)

Invoke: SetupInd (Signalling Control interface)
 Return Result: InitialDP
 Return Error: InitialDP

InitiateCallAttempt procedure (IC)

Invoke: InitialCallAttempt
 Return Result: None
 Return Error: InitiateCallAttempt

PlayAnnouncement procedure (PA)

Invoke: PlayAnnouncement
 Return Result: SpecializedResourceReport
 Return Error: PlayAnnouncement

PromptAndCollectUserInformation procedure (PC)

Invoke: PromptAndCollectUserInformation
 Return Result: PromptAndCollectUserInformation
 Return Error: PromptAndCollectUserInformation

ReleaseCall procedure (RC)

Invoke: ReleaseCall
 Return Result: None
 Return Error: None

RequestNotificationChargingEvent procedure (RN)

Invoke: RequestNotificationChargingEvent
 Return Result: EventNotificationCharging
 Return Error: RequestNotificationChargingEvent

RequestReportBCSMEvent procedure (RR)

Invoke: RequestReportBCSMEvent
 Return Result: EventReportBCSM
 Return Error: RequestReportBCSMEvent

SendChargingInformation procedure (SC)

Invoke: SendChargingInformation
 Return Result: None
 Return Error: SendChargingInformation

6.3 Structure of the test suite (TSS) for the basic capabilities

Table 1c shows the structure of the test suites for SSF functions and the number of Test Purposes produced.

Table 1c: Test suite structure of the SSF test

IUT	Interface	Protocol component	Procedure	Category and number
SSF	SSF-SCF	Basic subset	SF ServiceFiltering	CA 1
				BV 3
				BI 2
				BO 2
			AT ActivityTest	CA 1
				BV 2
				BI 1
				BO
			AC ApplyCharging	CA 1
				BV 4
				BI 2
				BO 1
			CG CallGap	CA 1
				BV 11
				BI 1
				BO
			CF CallInformation	CA 1
				BV 5
				BI 1
				BO 2
			CA Cancel	CA 1
				BV 3
				BI 1
				BO 1
			CI CollectInformation	CA 1
				BV
				BI 1
				BO 3
			CO Connect	CA 1
				BV 9
				BI 1
				BO 1
			CU Continue	CA 1
				BV 1
				BI
				BO
			FC Furnish Charging Information	CA 1
				BV 1
				BI
				BO
			DP InitialDP	CA 2
				BV 5
BI 2				
BO				
IC InitiateCall Attempt	CA 1			
	BV 2			
	BI			
	BO 2			
RC ReleaseCall	CA 1			
	BV 2			
	BI			
	BO 1			
RR RequestReport BCSMEvent	CA 1			
	BV 33			
	BI 2			
	BO 1			
SC SendCharging Information	CA			
	BV			
	BI			
	BO			

IUT	Interface	Protocol component	Procedure	Category and number
			RN	CA 1
			Request	BV 2
			Notification	BI 1
			ChargingEvent	BO 1

6.4 Test Purposes (TP) description

The objective is to test the INAP procedures at the Service Switching Point (SSP).

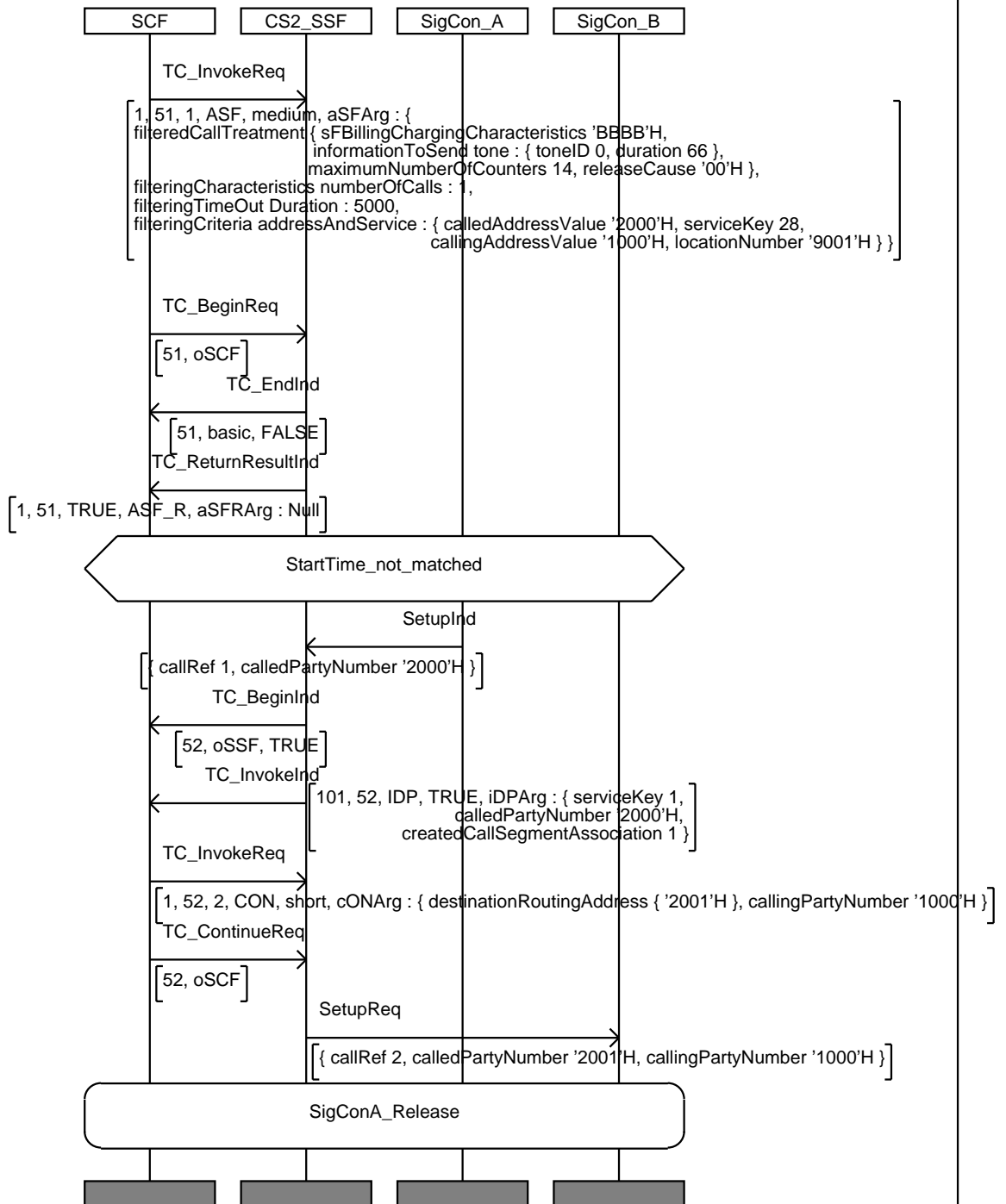
SigCon A and SigCon B are the signalling controls for users A and B, and the IUT is a SSF while the main lower tester is an SCF.

6.4.1 ServiceFiltering procedure

IN2_A_BASIC_SF_CA_01	
Purpose:	test ServiceFiltering procedure on duration
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF issues ActivateServiceFiltering invoke containing mandatory parameters only, with: <ul style="list-style-type: none"> - filteredCallTreatment including sFBillingChargingCharacteristics only, - filteringCharacteristics being interval, - filteringTimeOut being duration, - filteringCriteria being serviceKey, then a call is initiated after Characteristics being interval duration expires
Pass criteria	SSF accepts the call, then SSF issues ServiceFilteringResponse invoke with parameters <ul style="list-style-type: none"> - countersValue including 1 counterAndValue, - filteringCriteria being serviceKey
Postamble:	Release Call A.

IN2_A_BASIC_SF_BV_01	
Purpose:	test ServiceFiltering procedure on miscellaneous parameters
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF issues ActivateServiceFiltering invoke containing mandatory and optional parameters, with:</p> <ul style="list-style-type: none"> - filteredCallTreatment including: <ul style="list-style-type: none"> - sFBillingChargingCharacteristics, - informationToSend, - maximumNumberOfCounters, - filteringCharacteristics being numberOfCalls, - filteringTimeOut being stopTime, - filteringCriteria being addressAndService including: <ul style="list-style-type: none"> - calledAddressValue, - serviceKey, - callingAddressValue, - locationNumber, - startTime
Pass criteria	- Before startTime, SSF does not filter a call and passes it to SCF
Postamble:	SigConA_Release_thenB.

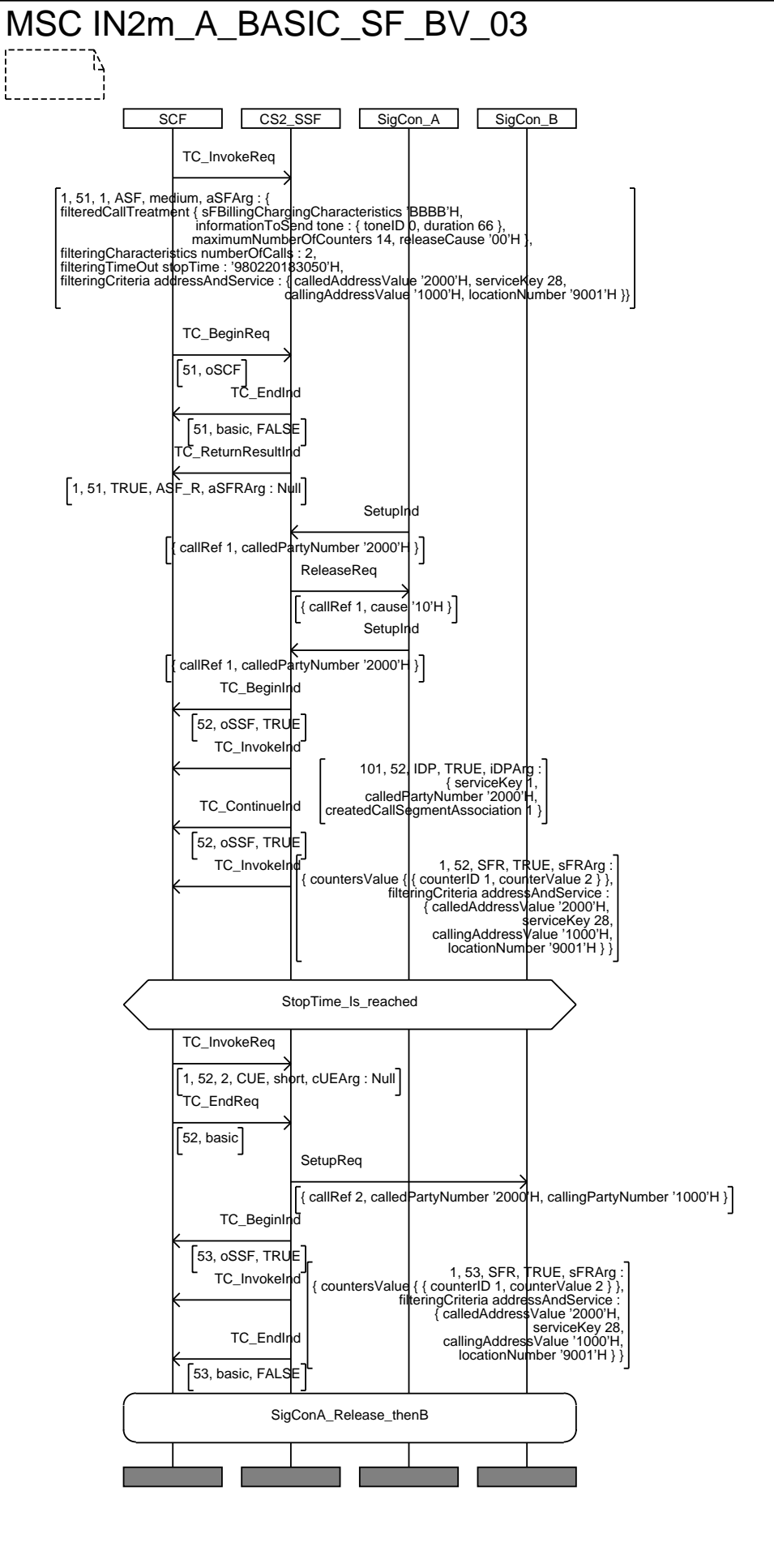
MSC IN2m_A_BASIC_SF_BV_01



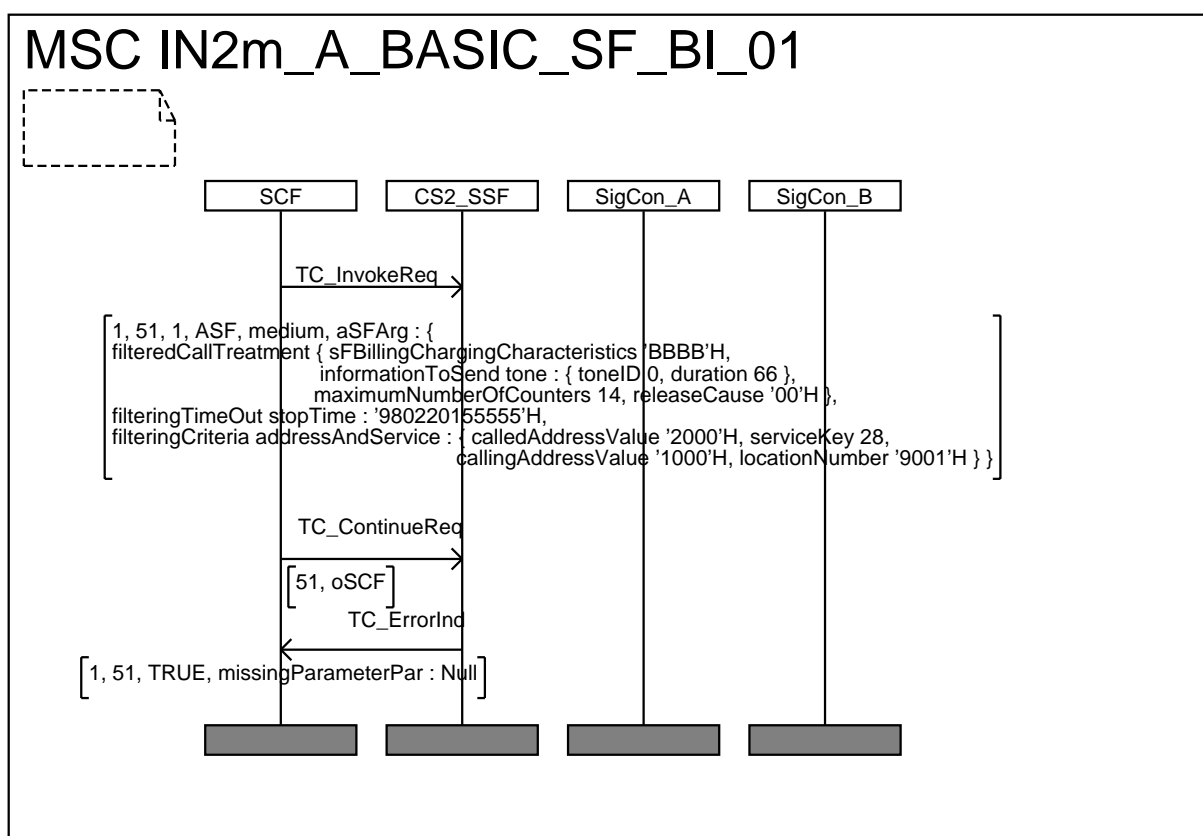
IN2_A_BASIC_SF_BV_02

This Test Purpose is not included.

IN2_A_BASIC_SF_BV_03	
Purpose:	test ServiceFiltering procedure on miscellaneous parameters
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF issues ActivateServiceFiltering invoke containing mandatory and optional parameters, with:</p> <ul style="list-style-type: none"> - filteredCallTreatment including: <ul style="list-style-type: none"> - sFBillingChargingCharacteristics; - informationToSend; - maximumNumberOfCounters. - filteringCharacteristics being numberOfCalls; - filteringTimeOut being stopTime; - filteringCriteria being addressAndService including: <ul style="list-style-type: none"> - calledAddressValue; - serviceKey; - callingAddressValue; - locationNumber. - startTime
Pass criteria	<ol style="list-style-type: none"> 1 SSF filters the first call 2 When Characteristics being numberOfCalls is reached, SSF passes a call to SCF then SSF issues ServiceFilteringResponse invoke with parameters: countersValue including counterValue, filteringCriteria being addressAndService 3 When stopTime is reached, SSF issues a final ServiceFilteringResponse with counterID remaining to previous value (no new call)
Postamble:	SigConA_Release_then_B.

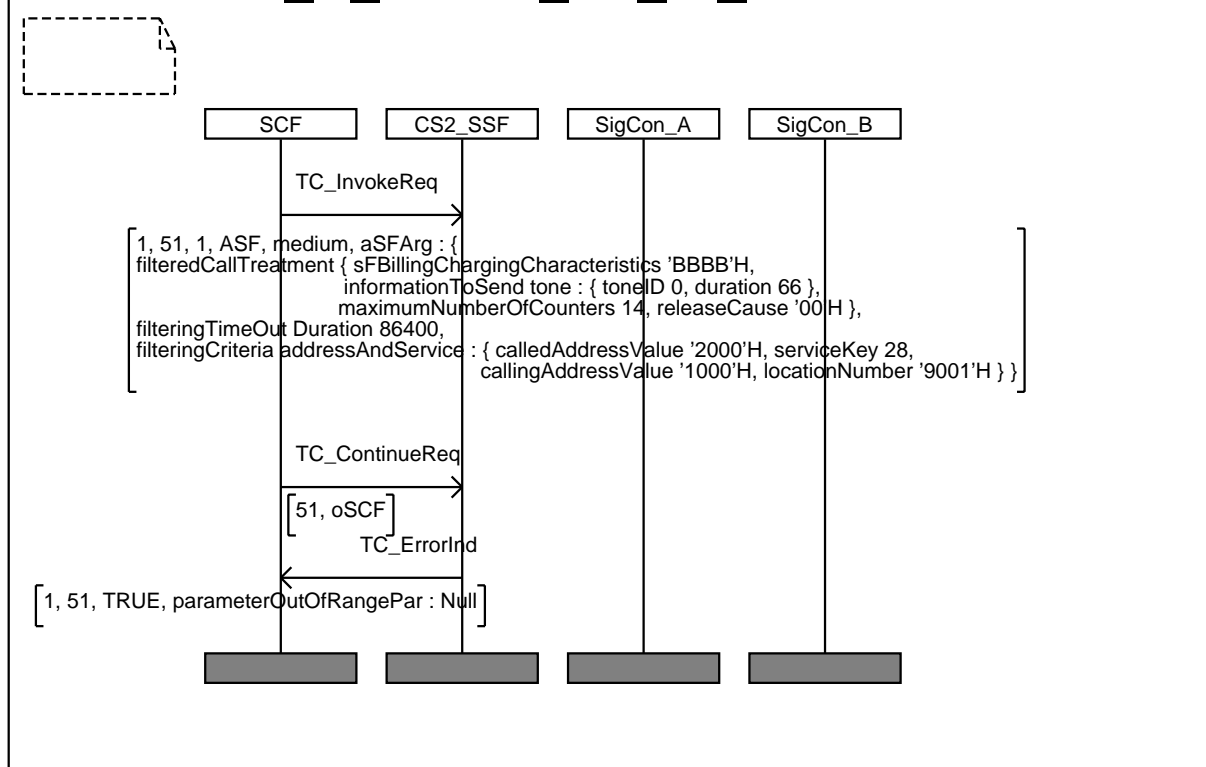


IN2_A_BASIC_SF_BI_01	
Purpose:	test ServiceFiltering procedure on missing parameters
Selection Cond.	
Preamble:	none
Test description	SCF issues ActivateServiceFiltering invoke with missing parameter - filteredCallTreatment
Pass criteria	- Check that SSF sends to SCF a ActivateServiceFiltering error with the indication of missing parameter
Postamble:	None.



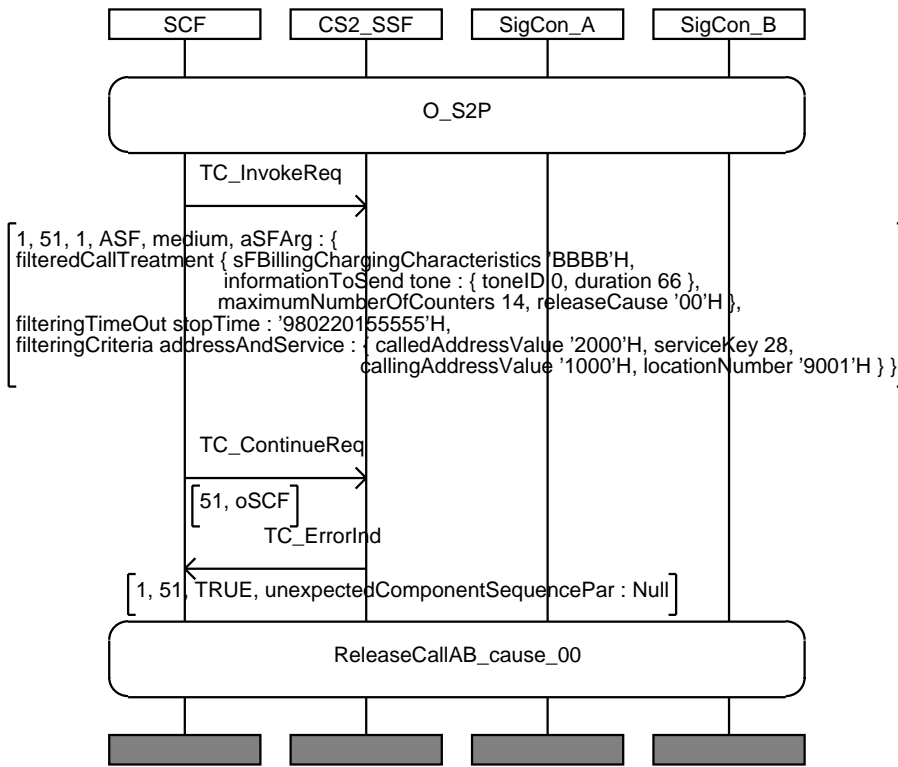
IN2_A_BASIC_SF_BI_02	
Purpose:	test ServiceFiltering procedure with parameter out of range
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF issues ActivateServiceFiltering invoke with parameter out of range - filteringTimeOut with duration > 86400
Pass criteria	- Check that SSF sends to SCF a ActivateServiceFiltering error with the indication of out of range parameter
Postamble:	None.

MSC IN2m_A_BASIC_SF_BI_02

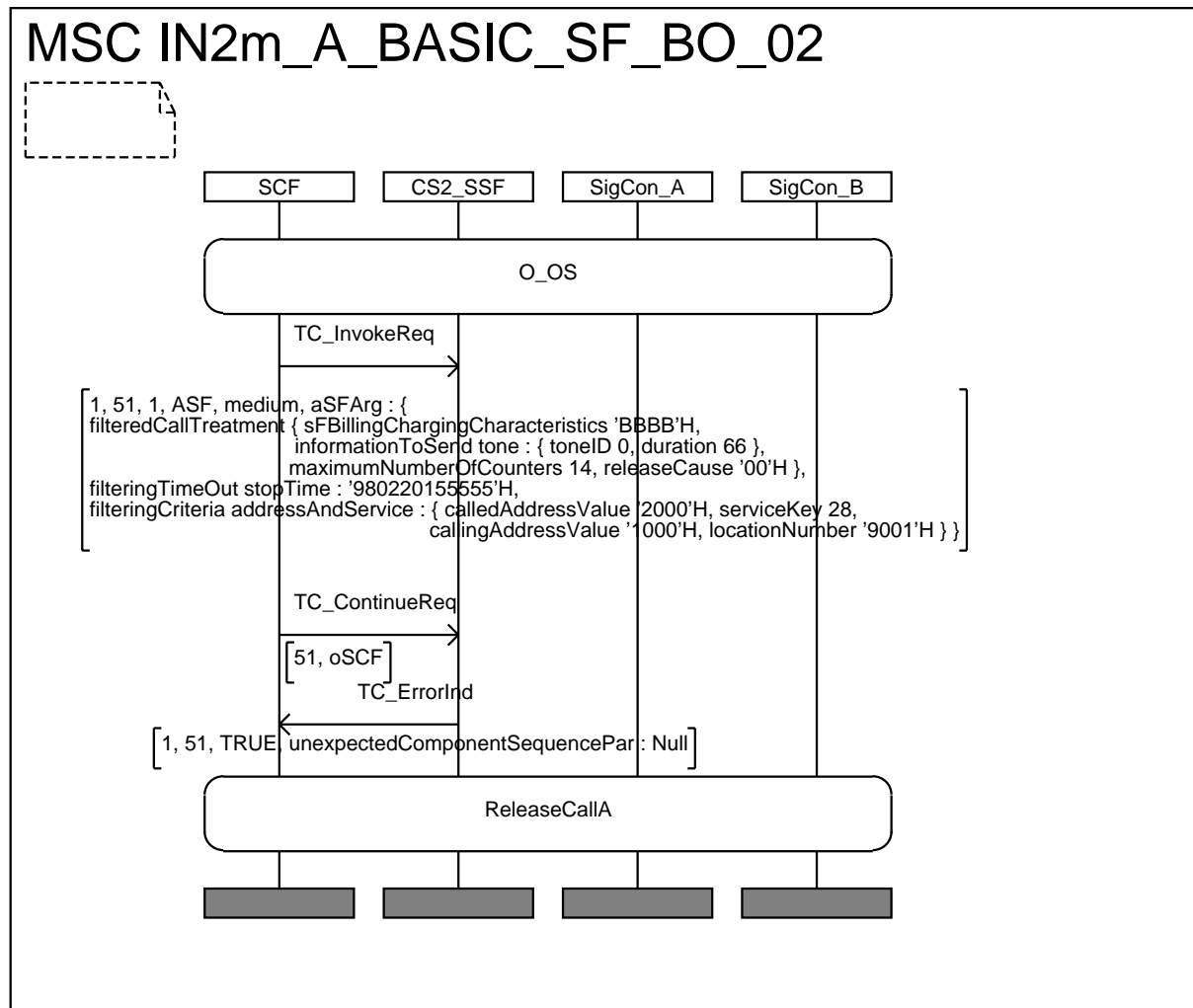


XXXX	IN2_A_BASIC_SF_BO_01
Purpose:	test ServiceFiltering procedure in Monitoring state
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	<p>SCF issues ActivateServiceFiltering invoke containing mandatory and optional parameters, with:</p> <ul style="list-style-type: none"> - filteredCallTreatment including: <ul style="list-style-type: none"> sFBillingChargingCharacteristics, - filteringTimeOut being duration, <ul style="list-style-type: none"> - filteringCriteria being addressAndService including: <ul style="list-style-type: none"> - calledAddressValue, - serviceKey, - callingAddressValue, - locationNumber - filteringCharacteristics being interval
Pass criteria	SSF issues ServiceFiltering error with unexpectedComponentSequence parameter
Postamble:	ReleaseCallAB_cause_00.

MSC IN2m_A_BASIC_SF_BO_01



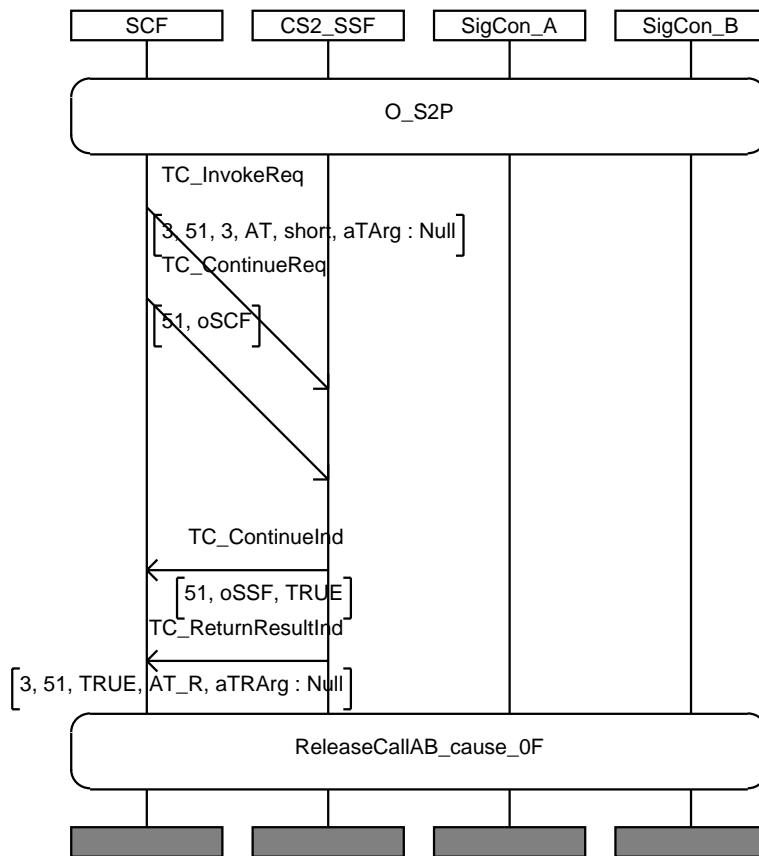
XXXX		IN2_A_BASIC_SF_BO_02	
Purpose:	test ServiceFiltering procedure in WaitForInstruction state		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF issues ActivateServiceFiltering invoke containing mandatory and optional parameters, with:</p> <ul style="list-style-type: none"> - filteredCallTreatment including: <ul style="list-style-type: none"> sFBillingChargingCharacteristics, - filteringTimeOut being duration, <ul style="list-style-type: none"> - filteringCriteria being addressAndService including: <ul style="list-style-type: none"> - calledAddressValue, - serviceKey, - callingAddressValue, - locationNumber - filteringCharacteristics being interval 		
Pass criteria	SSF issues ServiceFiltering error with unexpectedComponentSequence parameter		
Postamble:	ReleaseCallA.		



6.4.2 ActivityTest procedure

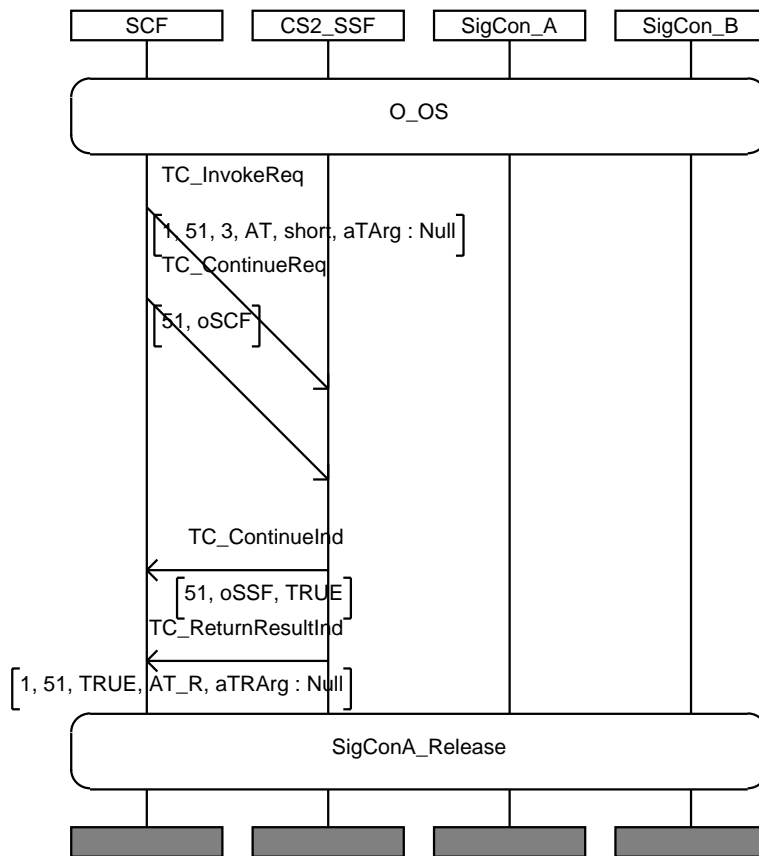
IN2_A_BASIC_AT_CA_01	
Purpose:	Test of ActivityTest in monitoring state
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	ActivityTest invoke sent by SCF to SSF, with TCAP DialogueId of dialogue identical to the one used in the preamble
Pass criteria	ActivityTest result sent by SSF to SCF related to the existing dialogue
Postamble:	ReleaseCallAB_cause_0F.

MSC IN2_A_BASIC_AT_CA_01



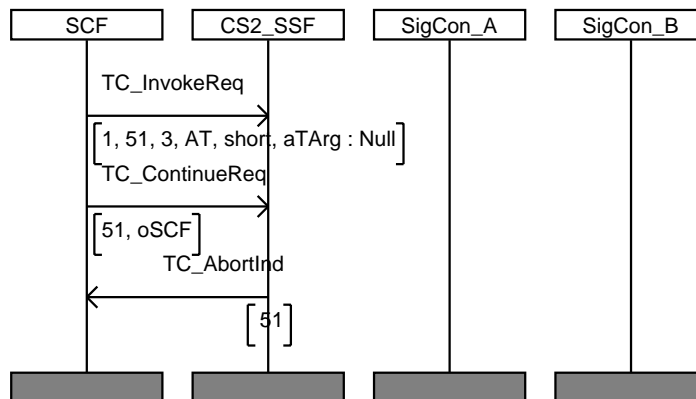
IN2_A_BASIC_AT_BV_01	
Purpose:	Test of ActivityTest in WaitForInstructions state
Requirement ref	
Preamble:	O_OS
Selection Cond.	
Test description	ActivityTest invoke sent by SCF to SSF with TCAP DialogueId of dialogue identical to the one used in the preamble
Pass criteria	ActivityTest result sent by SSF to SCF related to the existing dialogue
Postamble:	SigConA_Release.

MSC IN2_A_BASIC_AT_BV_01

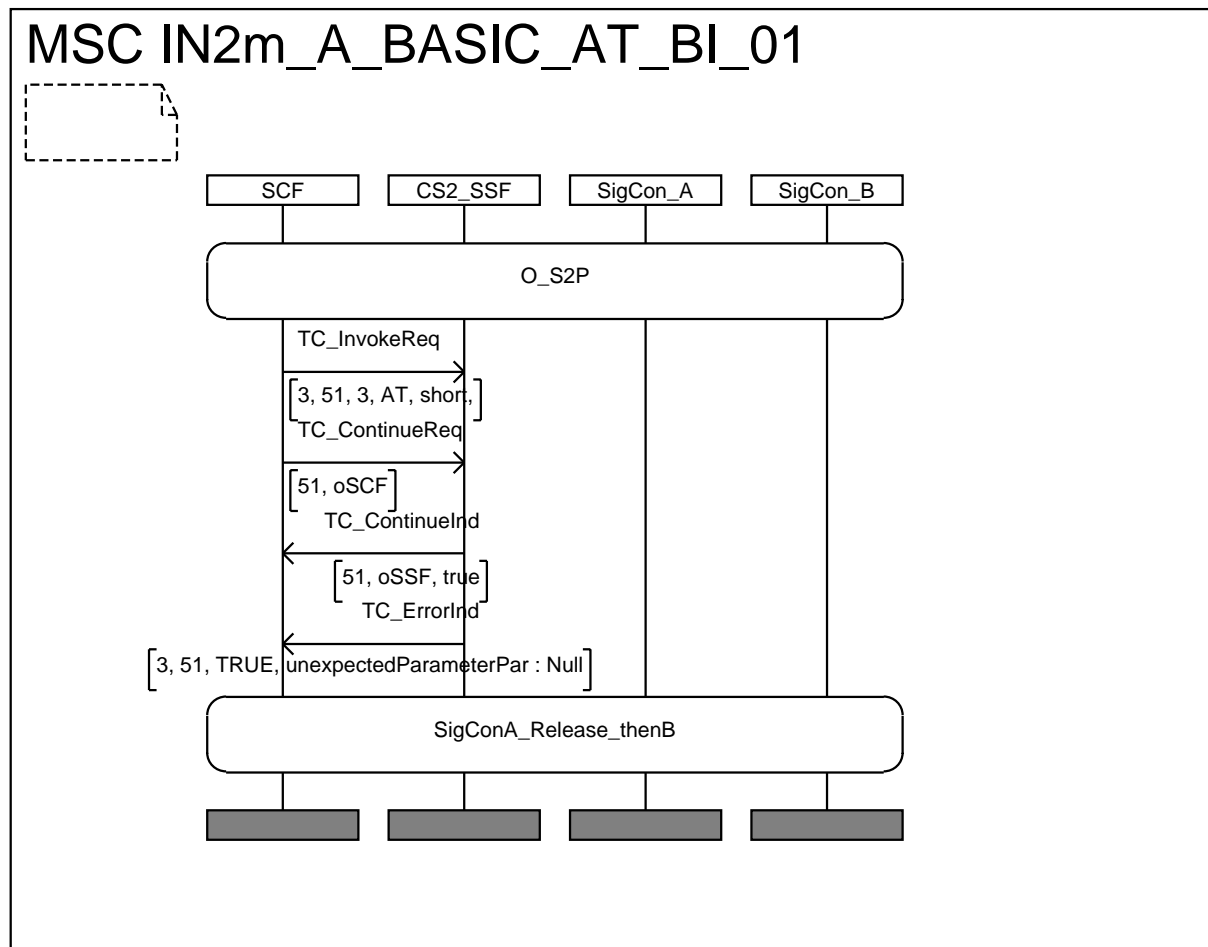


IN2_A_BASIC_AT_BV_02	
Purpose:	Test of ActivityTest in Idle state
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	ActivityTest invoke sent by SCF to SSF with TCAP DialogueId of dialogue which is not existing
Pass criteria	SSF rejects the invoke or aborts the dialogue (TCAP)
Postamble:	none

MSC IN2m_A_BASIC_AT_BV_02



IN2_A_BASIC_AT_BI_01	
Purpose:	Test of invalid ActivityTest invoke
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	ActivityTest invoke with argument sent by SCF to SSF during preamble
Pass criteria	SSF issues TC_error indicating unexpectedParameter
Postamble:	SigConA_release_then_B



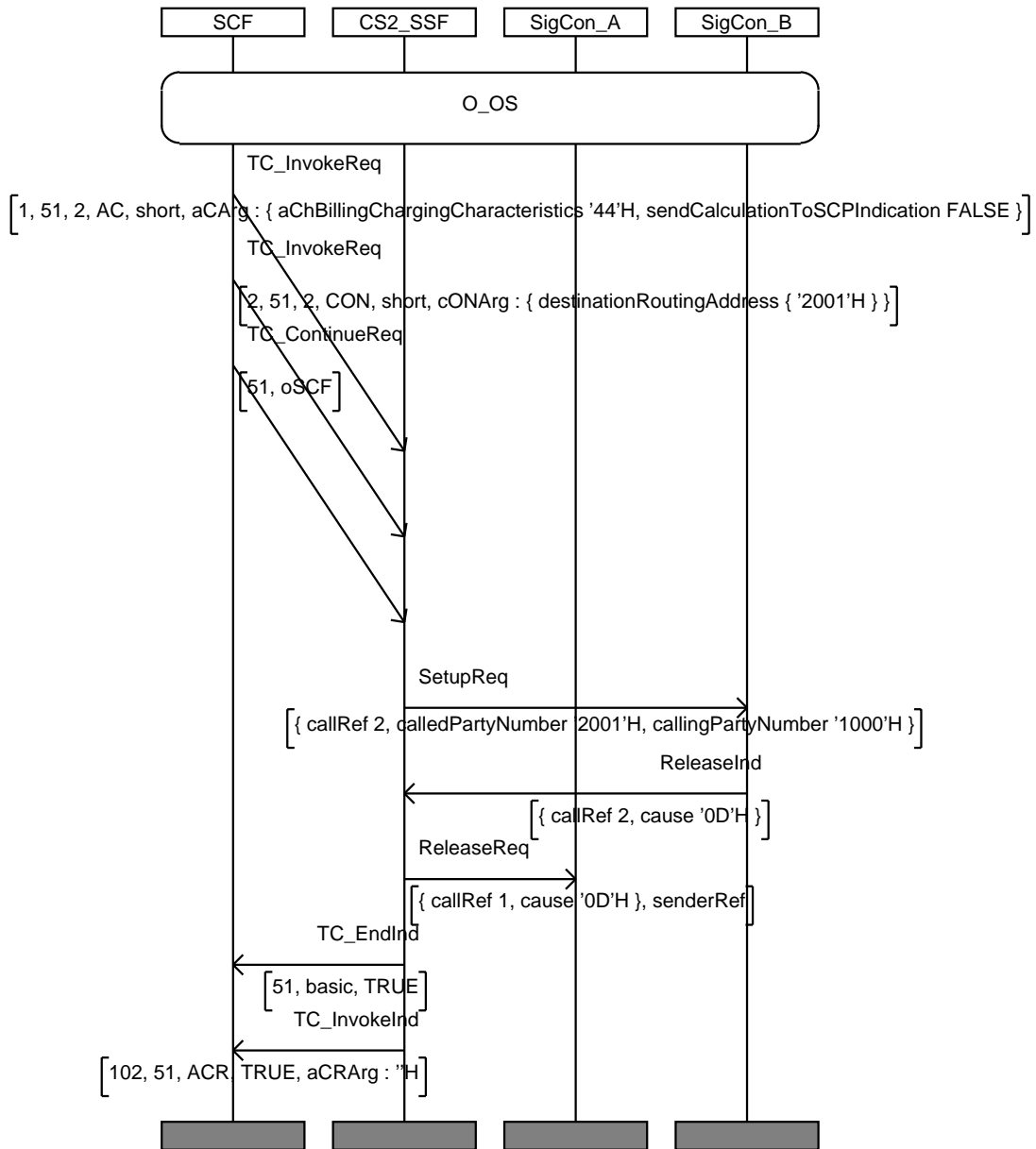
6.4.3 ApplyCharging procedure

Charging related aspects in IN are network operator specific. Therefore, it is not possible to define useful test purposes for charging procedures using a network operator independent approach. TP specification has to be done by network operators, using INAP procedures themselves. The TPs could be specified by combining ApplyCharging, FurnishChargingInformation and SendChargingInformation procedures.

Examples:

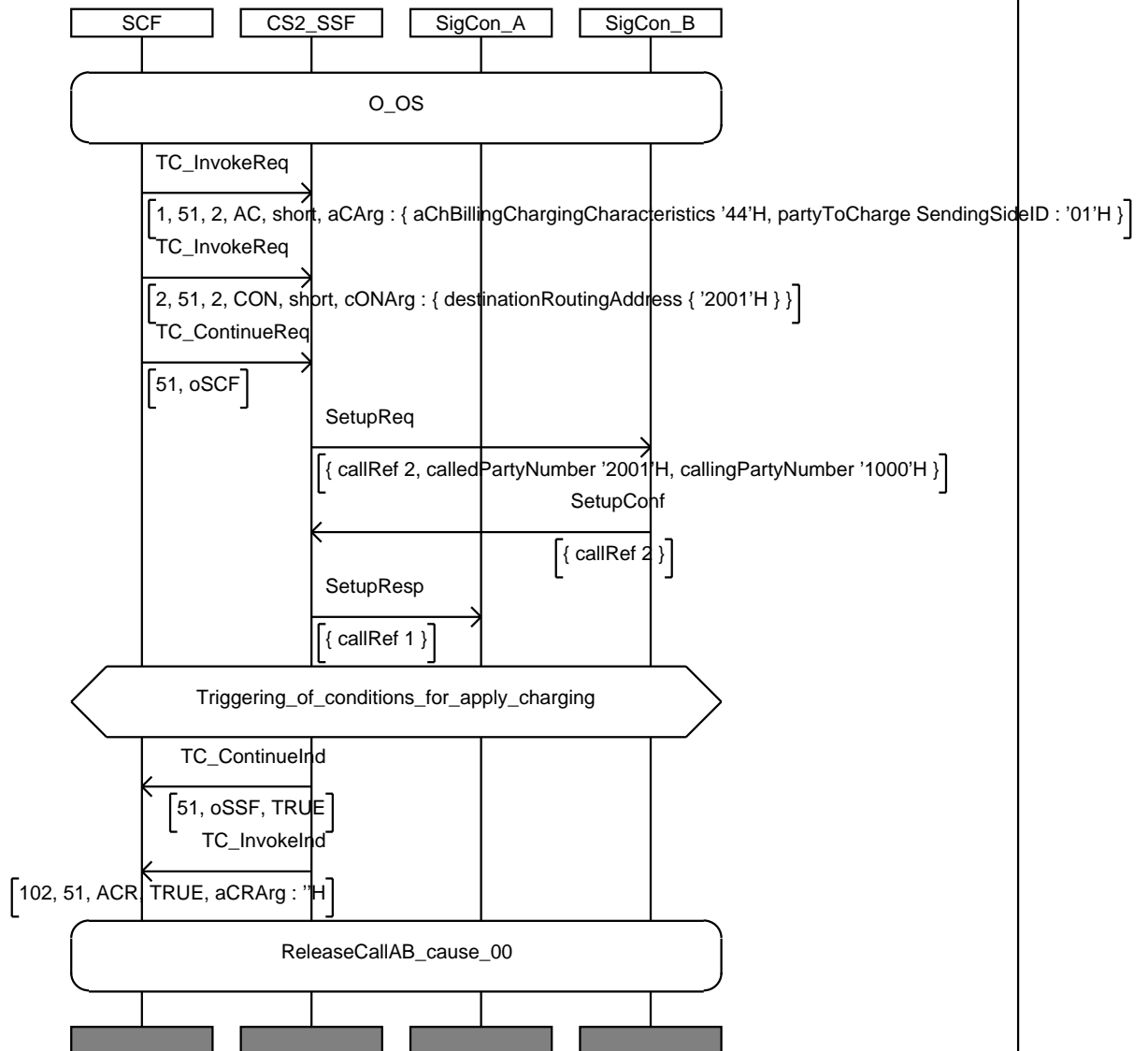
IN2 A BASIC AC CA 01	
Purpose:	Test of ApplyCharging base procedure
Requirement ref	
Preamble:	O_OS
Selection Cond.	
Test description	<p>ApplyCharging invoke sent by SCF to SSF, containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - aChBillingChargingCharacteristics, <p>followed by Connect invoke containing mandatory parameters only</p> <ul style="list-style-type: none"> - destinationRoutingAddress, <p>As a consequence, SSF sends a SetupReq to SigCon B</p>
Pass criteria	ApplyCharging report sent by SSF to SCF upon reception of a Release indication from SigCon B
Postamble:	none

MSC IN2_A_BASIC_AC_CA_01



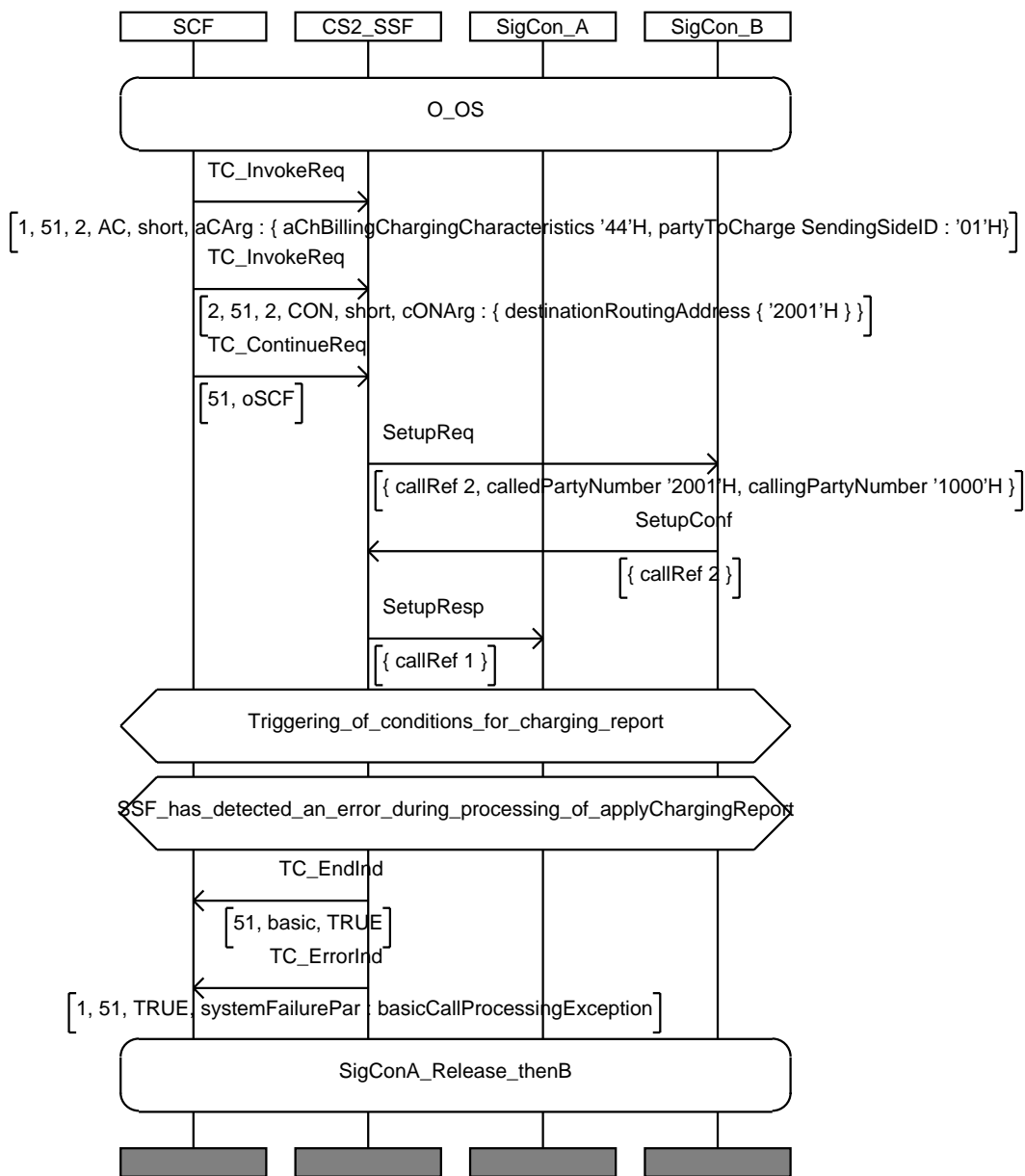
IN2_A_BASIC_AC_BV_01	
Purpose:	Test of ApplyCharging procedure with optional parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>ApplyCharging invoke sent by SCF to SSF, containing mandatory and optional parameters, with:</p> <ul style="list-style-type: none"> - aChBillingChargingCharacteristics, - partyToCharge being sendingSideID, <p>followed by Connect invoke containing mandatory parameters only</p> <p>As a consequence, SSF sends a SetupReq to SigCon B. SigCon B answers.</p>
Pass criteria	<p>upon detection of conditions for charging report, SSF sends to SCF an ApplyChargingReport invoke, with:</p> <ul style="list-style-type: none"> - callResult
Postamble:	ReleaseCallAB_cause_00

MSC IN2m_A_BASIC_AC_BV_01



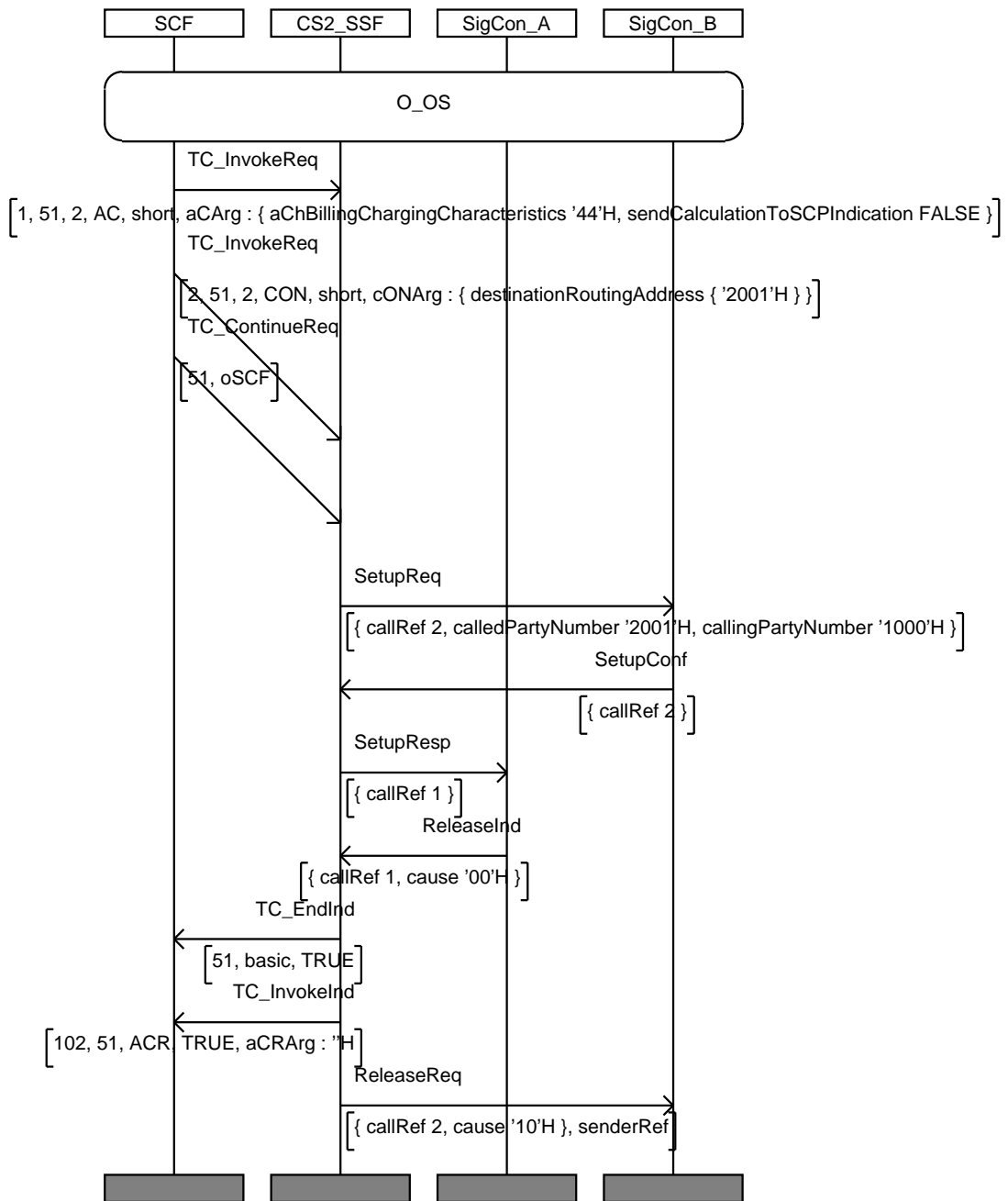
IN2_A_BASIC_AC_BV_02	
Purpose:	Test of ApplyCharging error procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>ApplyCharging invoke sent by SCF to SSF, containing mandatory and optional parameters, with:</p> <ul style="list-style-type: none"> - aChBillingChargingCharacteristics, - partyToCharge being sendingSideID, <p>followed by Connect invoke containing mandatory parameters only</p> <p>As a consequence, SSF sends a SetupReq to SigCon B. SigConB answers upon detection of conditions for charging report, SSF detects an error.</p>
Pass criteria	- SSF sends a TC_ErrorInd with systemFailurePar: basicCallProcessingException.
Postamble:	SigConA_Release_thenB

MSC IN2m_A_BASIC_AC_BV_02



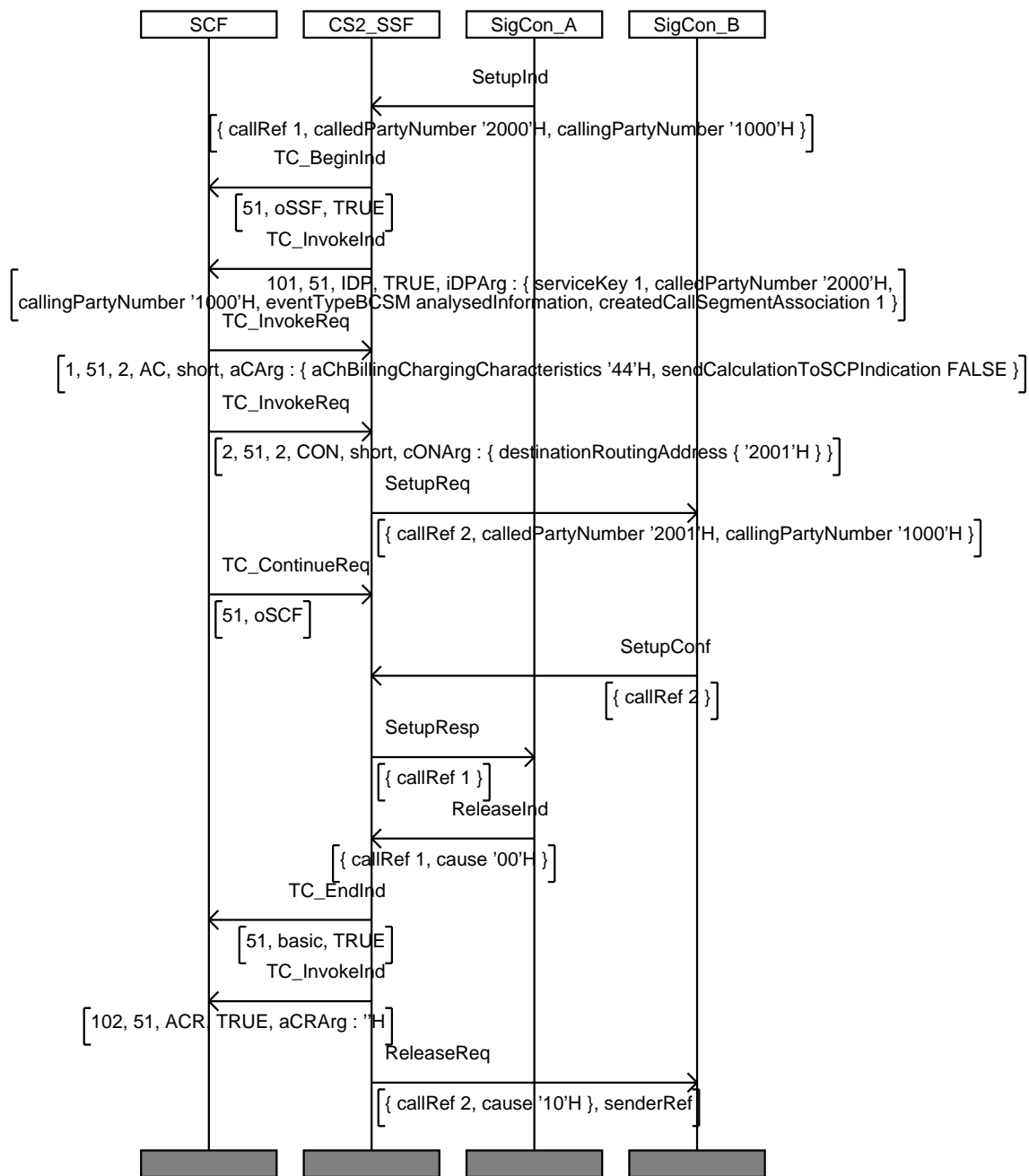
XXXX	IN2_A_BASIC_AC_BV_03
Purpose:	Test of ApplyCharging and Connect in same transaction
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>ApplyCharging invoke sent by SCF to SSF, containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - aChBillingChargingCharacteristics, <p>followed by Connect invoke containing mandatory parameters only</p> <ul style="list-style-type: none"> - destinationRoutingAddress, <p>As a consequence, SSF sends a SetupReq to SigCon B</p>
Pass criteria	ApplyCharging report sent by SSF to SCF upon reception of a Release indication from SigCon A
Postamble:	none

MSC IN2_A_BASIC_AC_BV_03



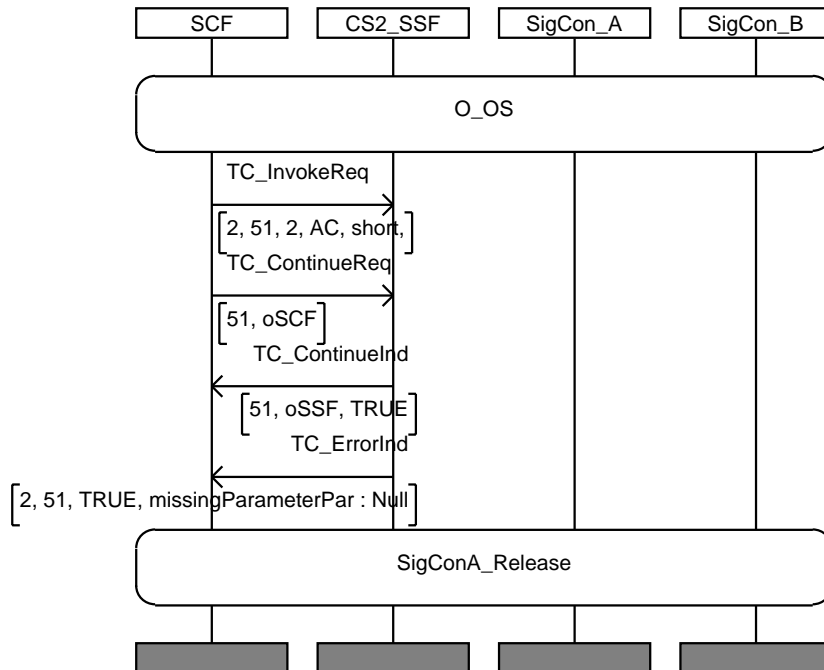
XXXX	IN2_A_BASIC_AC_BV_04
Purpose:	Test of ApplyCharging when Sigcon B answers
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>ApplyCharging invoke sent by SCF to SSF, containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - aChBillingChargingCharacteristics, <p>followed by Connect invoke containing mandatory parameters only</p> <ul style="list-style-type: none"> - destinationRoutingAddress, <p>As a consequence, SSF sends a SetupReq to SigCon B</p> <p>SigCon B sends SetupConf</p>
Pass criteria	ApplyCharging report sent by SSF to SCF upon reception of a Release indication from SigCon A
Postamble:	none

MSC IN2_A_BASIC_AC_BV_04



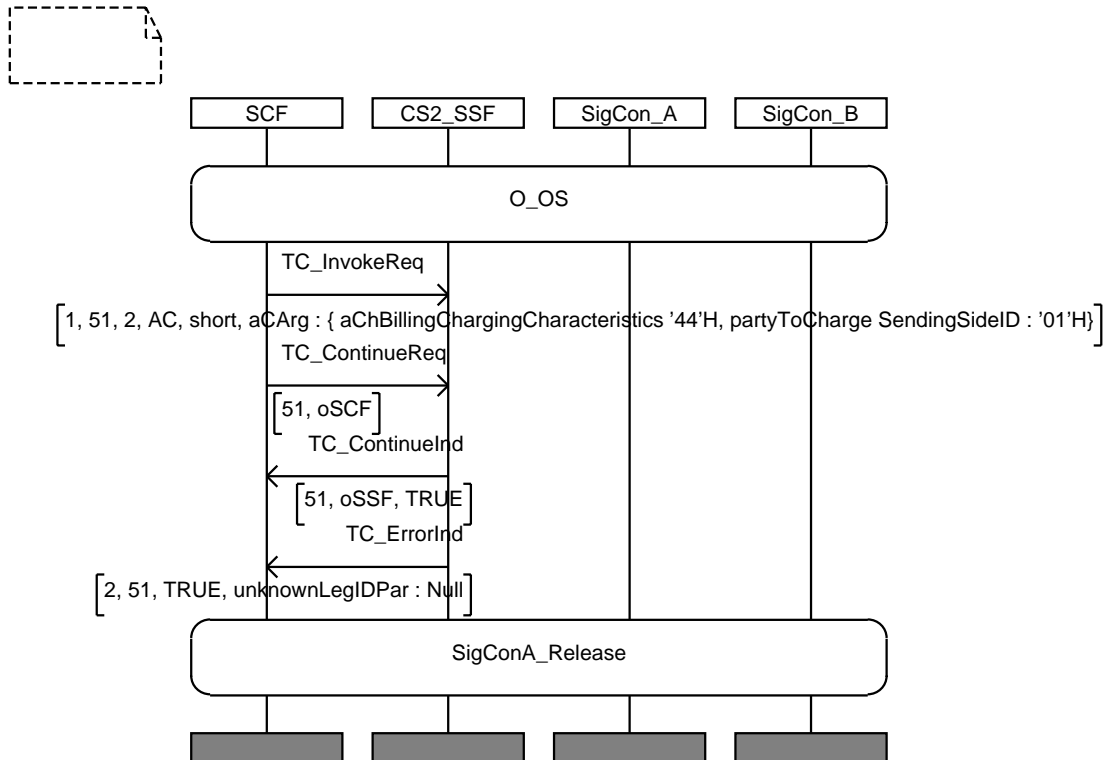
IN2_A_BASIC_AC_BI_01	
Purpose:	Test of ApplyCharging procedure with missing parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF ApplyCharging invoke without the parameter - aChBillingChargingCharacteristics,
Pass criteria	SSF sends to SCF TC_ErrorIndication containing missingParameter error
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_AC_BI_01



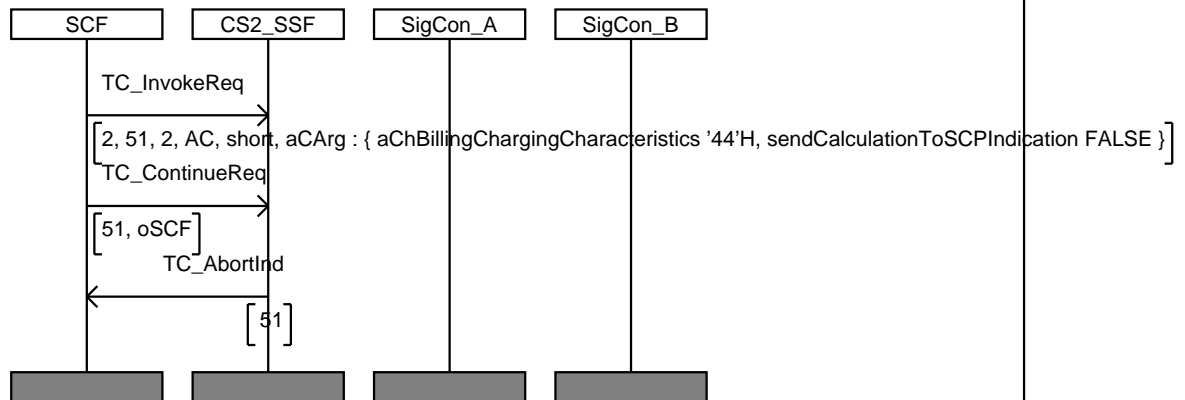
IN2_A_BASIC_AC_BI_02	
Purpose:	Test of ApplyCharging procedure error with unknown LegID
Requirement ref	
Selection Cond.	CS-2 only
Preamble:	O_OS
Test description	SCF sends to SSF ApplyCharging invoke with - partyToCharge being a not existing legid
Pass criteria	SSF sends to SCF TC_errorInd containing unknownLegID
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_AC_BI_02



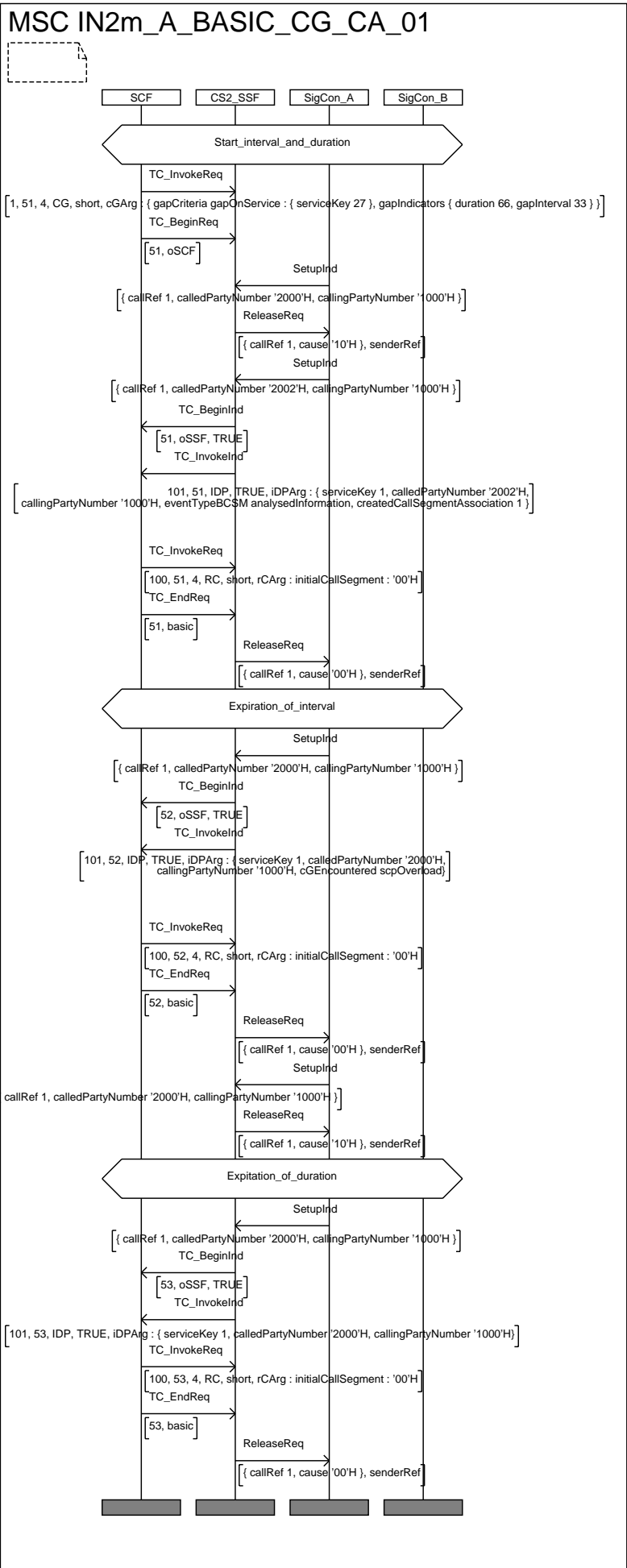
IN2_A_BASIC_AC_BO_01	
Purpose:	Test of ApplyCharging procedure in wrong state
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF ApplyCharging invoke from idle state
Pass criteria	SSF sends to SCF a TC-ABORT
Postamble:	none

MSC IN2m_A_BASIC_AC_BO_01

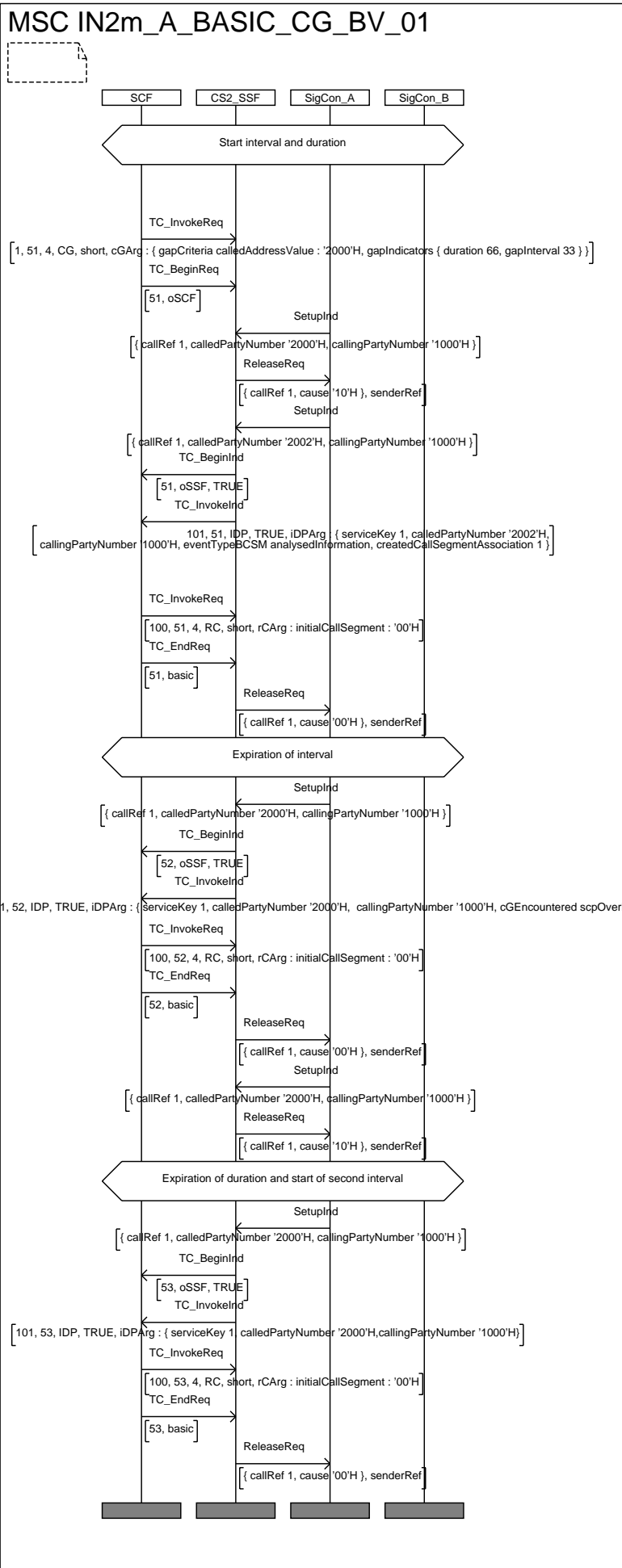


6.4.4 CallGap procedure

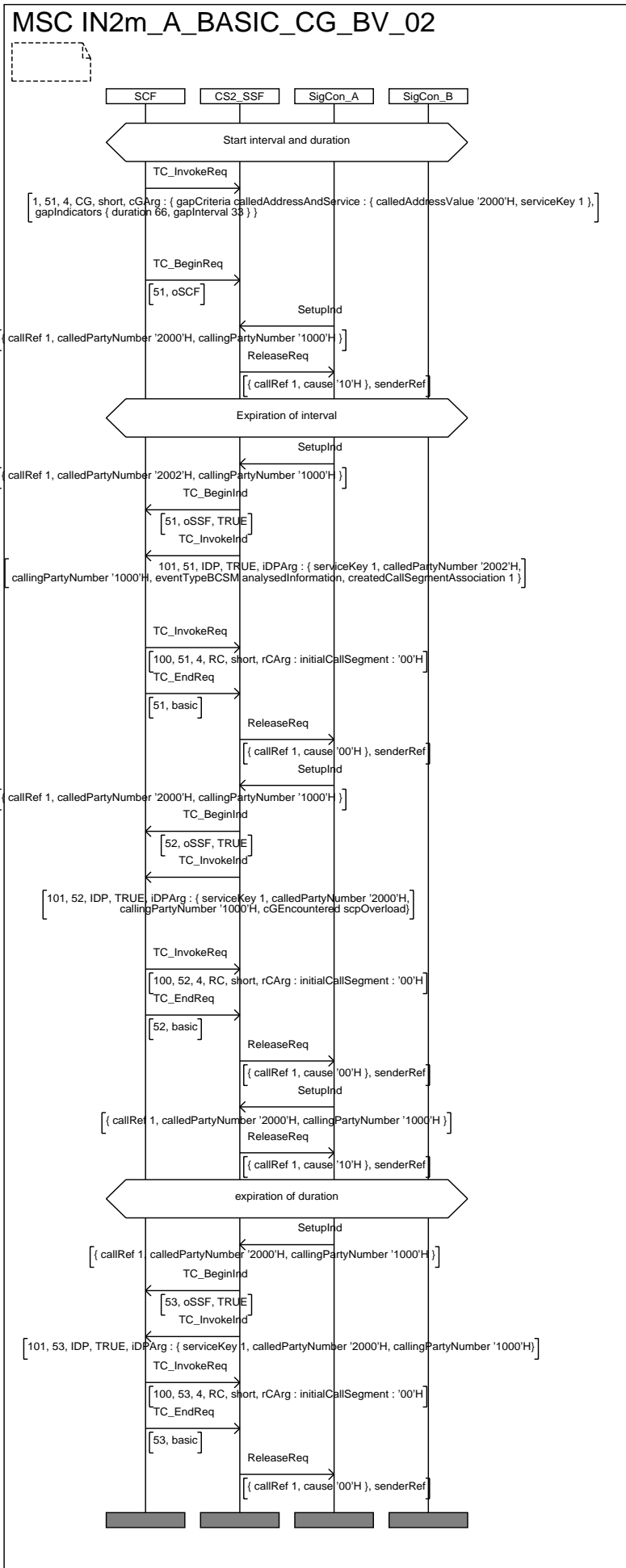
IN2_A_BASIC_CG_CA_01	
Purpose:	Test of CallGap base procedure and serviceKey parameter
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> gapOnService with any valid value for serviceKey, - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callGapping is active for the service key used in SetupInd - Check that SSF sends to SCF an InitialDP invoke as callgapping is NOT active when the service key in the SetupInd is different - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered"
Postamble:	none



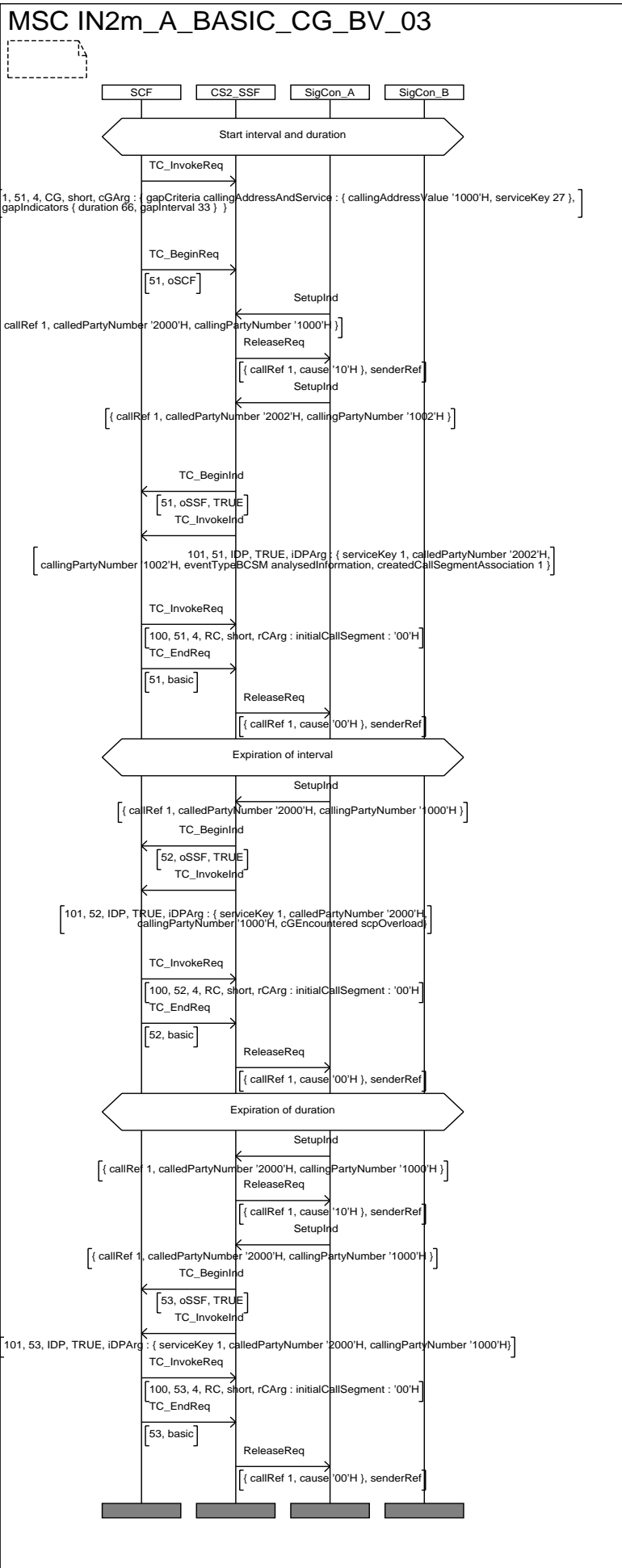
IN2_A_BASIC_CG_BV_01	
Purpose:	Test of CallGap procedure and calledAddressValue parameter
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF a CallGap invoke containing mandatory parameters only, with: <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> calledAddressValue with any valid value, - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callgapping is active for the called address used in SetupInd - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - Check that SSF releases a call coming within the second interval, as callgapping is still active for the called address used in SetupInd - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, - and without "cGEncountered"
Postamble:	none



IN2_A_BASIC_CG_BV_02	
Purpose:	Test of CallGap procedure and calledAddressAndService parameter
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - gapCriteria: calledAddressAndService with any valid value for serviceKey - gapIndicators duration being a duration value in seconds gapInterval being an interval value in seconds
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callgapping is active for the called address and service key used in SetupInd - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - Check that SSF releases a call coming within the second interval, as callgapping is still active for the called address and service key used in SetupInd - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, - and without "cGEncountered"
Postamble:	none

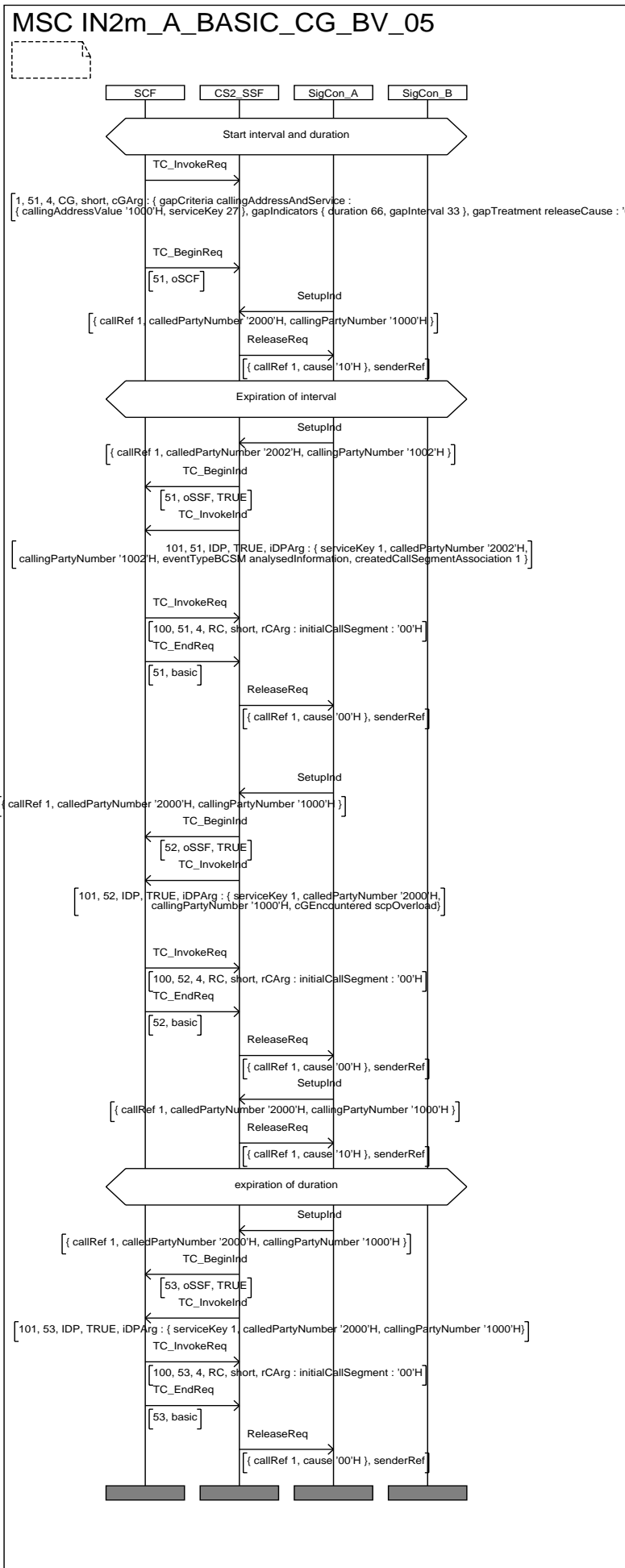


IN2_A_BASIC_CG_BV_03	
Purpose:	Test of CallGap procedure and optional parameter callingAddressAndService
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF a CallGap invoke containing mandatory parameters with: <ul style="list-style-type: none"> - gapCriteria: callingAddressAndService with any valid value, - gapIndicators duration being a duration value in seconds gapInterval being an interval value in seconds
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callGapping is active for the calling address and service key used in SetupInd - When a SetupInd comes with service key used in CallGap but not the calling address, check that SSF sends an InitialDP invoke containing all mandatory parameters - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - Check that SSF releases a call coming within the second interval, as callGapping is still active for the calling address and service key used in SetupInd - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered"
Postamble:	none

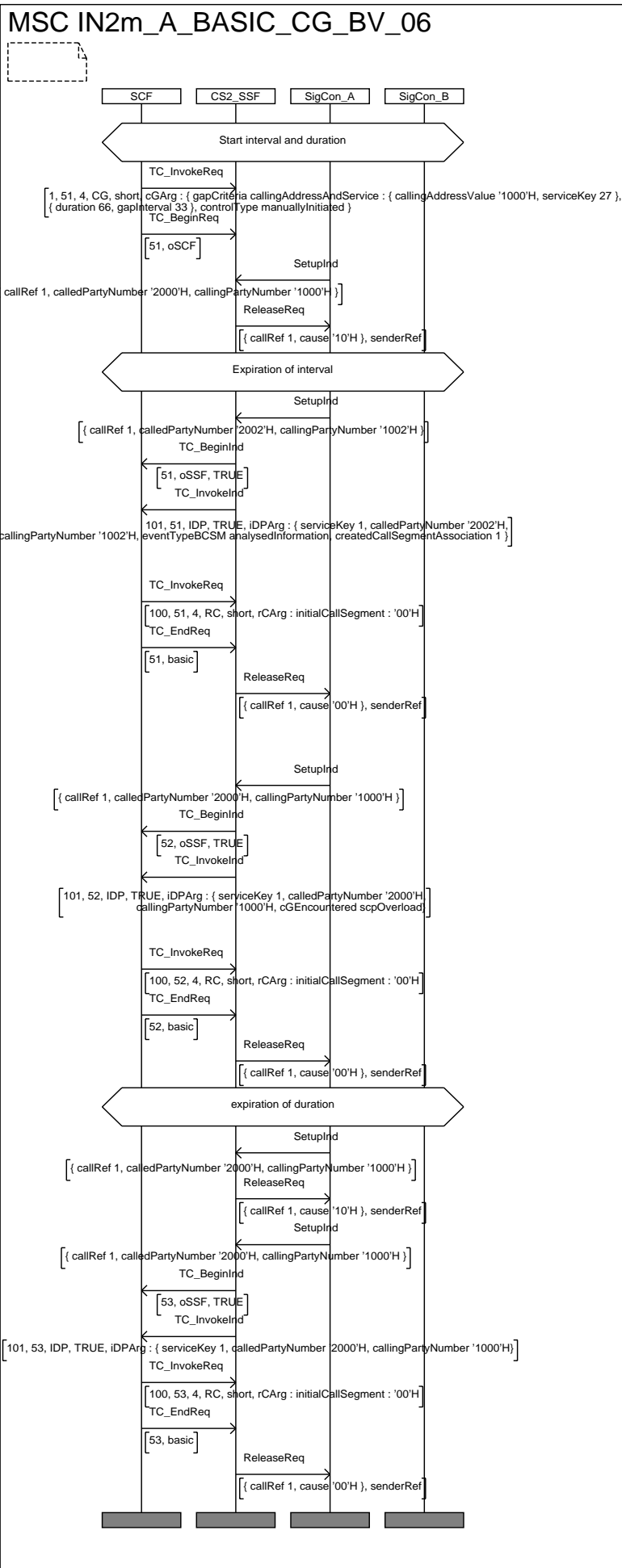


IN2_A_BASIC_CG_BV_04	
Purpose:	Test of CallGap procedure and optional parameters callingAddressAndService and gapTreatment via inbandInfo
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory and optional parameters with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> callingAddressAndService with any valid value - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds - gapTreatment <ul style="list-style-type: none"> informationToSend being inbandInfo
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callgapping is active for the calling address and service key used in SetupInd. The inband information has to be sent to the calling party - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - Check that SSF releases a call coming within the second interval, as callgapping is still active for the calling address and service key used in SetupInd. The inband information has to be sent to the calling party - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, <p>but without "cGEncountered"</p>
Postamble:	none

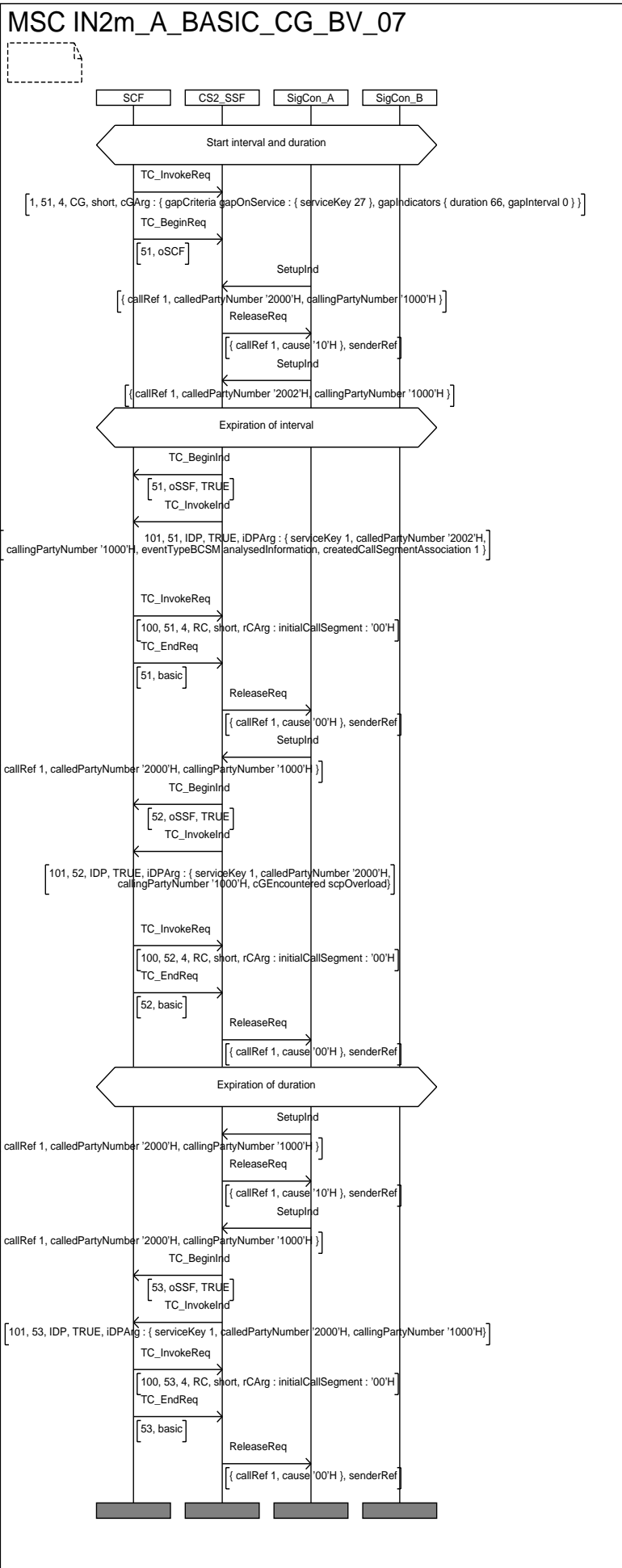
IN2_A_BASIC_CG_BV_05	
Purpose:	Test of CallGap procedure and optional parameters callingAddressAndService and gapTreatment with releaseCause
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory and optional parameters with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> callingAddressAndService - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds - gapTreatment <ul style="list-style-type: none"> releaseCause being any value except default value
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callgapping is active for the calling address and service key used in SetupInd. The cause value has to be sent to the calling party - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - Check that SSF releases a call coming within the second interval, as callgapping is still active for the calling address and service key used in SetupInd. The cause value has to be sent to the calling party - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, <p>but without "cGEncountered"</p>
Postamble:	none



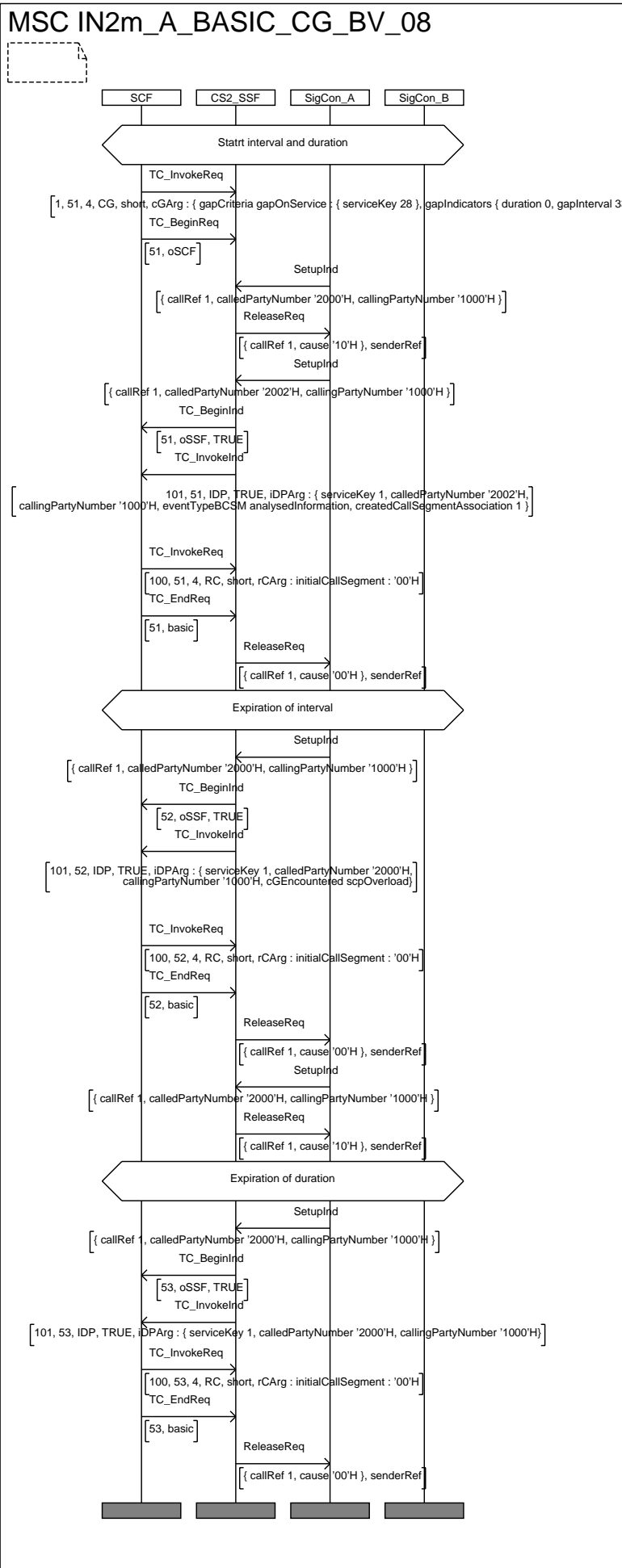
IN2_A_BASIC_CG_BV_06	
Purpose:	Test of CallGap procedure and optional parameters callingAddressAndService and controlType
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory and optional parameters with:</p> <ul style="list-style-type: none"> - gapCriteria: callingAddressAndService - gapIndicators duration being a duration value in seconds gapInterval being an interval value in seconds - controlType being manuallyInitiated
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callgapping is active for the calling address and service key used in SetupInd - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - Check that SSF releases a call coming within the second interval, as callGapping is still active for the calling address and service key used in SetupInd. - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered"
Postamble:	none



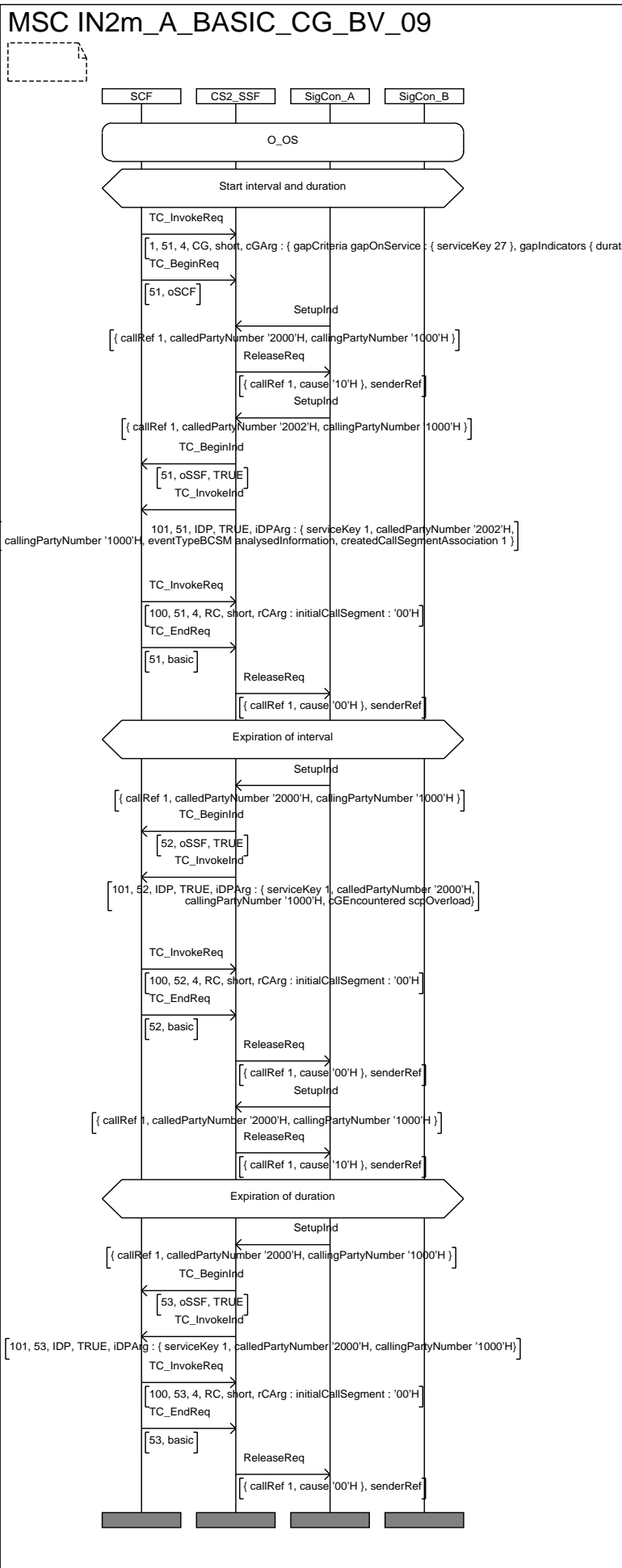
XXXX		IN2_A_BASIC_CG_BV_07	
Purpose:	Test of CallGap procedure and serviceKey parameter		
Requirement ref			
Selection Cond.			
Preamble:	none		
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> gapOnService with any valid value for serviceKey, - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds set to 0 		
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callGapping is active for the service key used in SetupInd - Check that SSF sends to SCF an InitialDP invoke as callgapping is NOT active when the service key in the SetupInd is different - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered" 		
Postamble:	none		



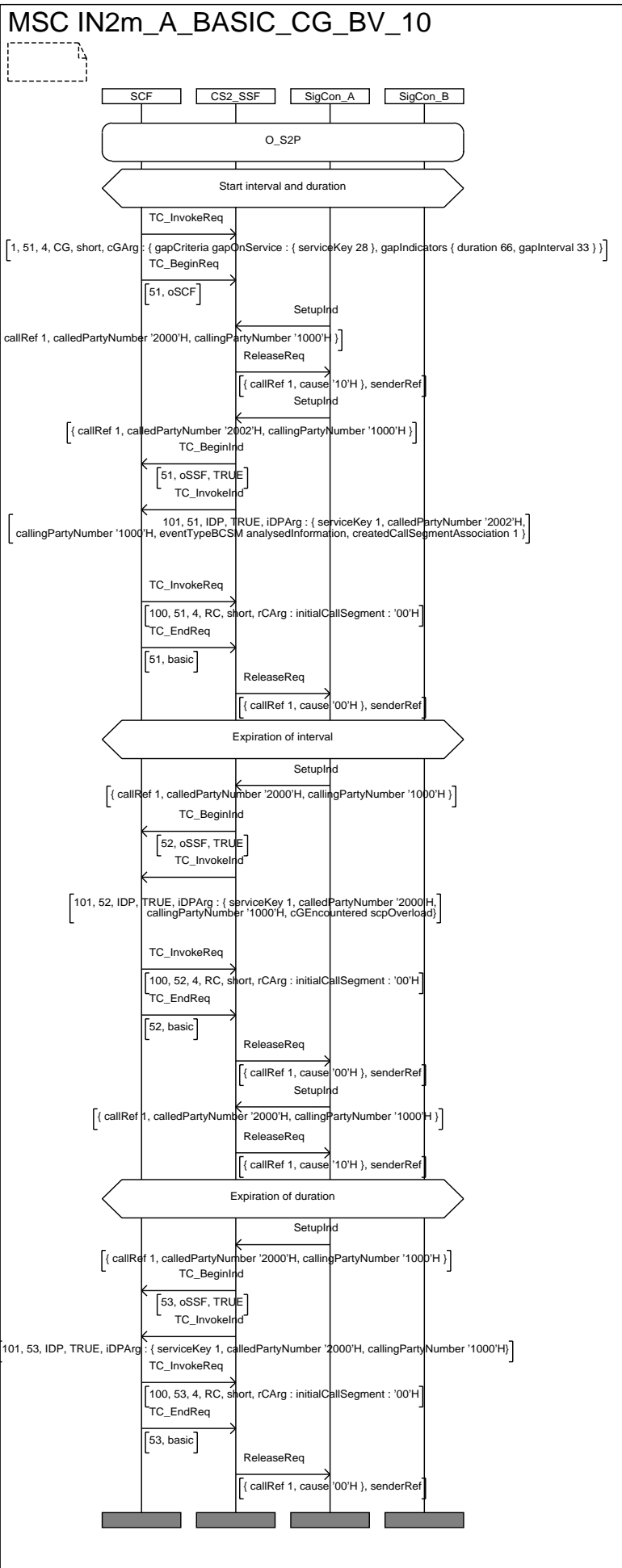
XXXX		IN2_A_BASIC_CG_BV_08	
Purpose:	Test of CallGap procedure and serviceKey parameter		
Requirement ref			
Selection Cond.			
Preamble:	none		
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> gapOnService with any valid value for serviceKey, - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds set to 0 gapInterval being an interval value in seconds 		
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callGapping is active for the service key used in SetupInd - Check that SSF sends to SCF an InitialDP invoke as callgapping is NOT active when the service key in the SetupInd is different - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered" 		
Postamble:	none		



XXXX		IN2_A_BASIC_CG_BV_09	
Purpose:	Test of CallGap procedure in WaitForInstruction state		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> gapOnService with any valid value for serviceKey, - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds 		
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callGapping is active for the service key used in SetupInd - Check that SSF sends to SCF an InitialDP invoke as callgapping is NOT active when the service key in the SetupInd is different - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered" 		
Postamble:	none		

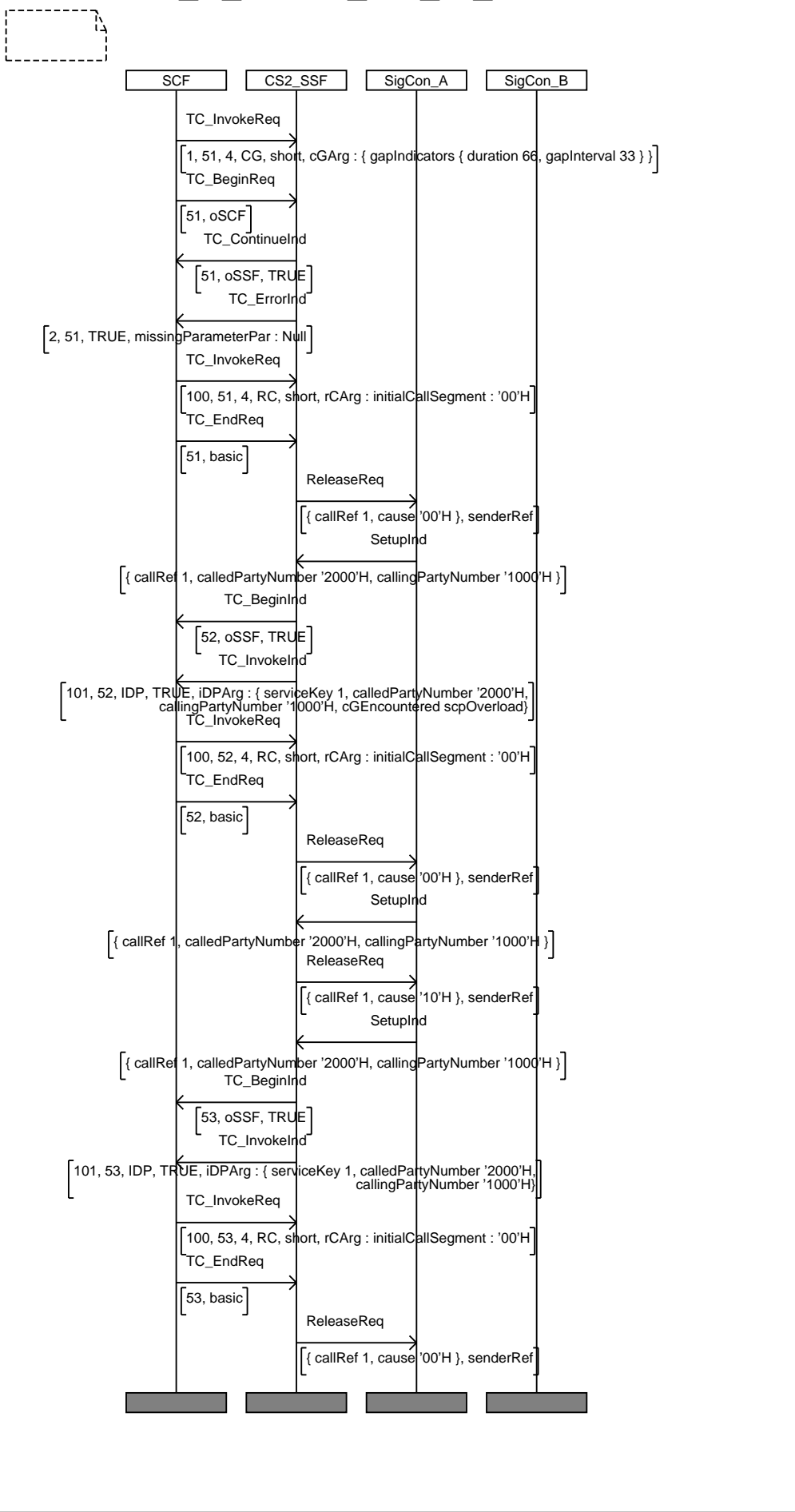


XXXX		IN2_A_BASIC_CG_BV_10
Purpose:	Test of CallGap procedure in Monitoring state	
Requirement ref		
Selection Cond.		
Preamble:	O_S2P	
Test description	<p>SCF sends to SSF a CallGap invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - gapCriteria: <ul style="list-style-type: none"> gapOnService with any valid value for serviceKey, - gapIndicators <ul style="list-style-type: none"> duration being a duration value in seconds gapInterval being an interval value in seconds 	
Pass criteria	<ul style="list-style-type: none"> - Check that SSF releases a call when callGapping is active for the service key used in SetupInd - Check that SSF sends to SCF an InitialDP invoke as callgapping is NOT active when the service key in the SetupInd is different - When a SetupInd comes after expiration of interval, check that SSF sends an InitialDP invoke containing all mandatory parameters and indicating call gapping encountered, with at least the parameters: <ul style="list-style-type: none"> - serviceKey, - cGEncountered - When a SetupInd comes after expiration of duration, check that SSF sends an InitialDP invoke containing all mandatory parameters without indicating call gapping encountered, with at least the parameter: <ul style="list-style-type: none"> - serviceKey, but without "cGEncountered" 	
Postamble:	none	



IN2_A_BASIC_CG_BI_01	
Purpose:	Test of CallGap procedure and missing parameter
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF a CallGap invoke with missing mandatory parameter gapCriteria
Pass criteria	Check that SSF rejects the invoke
Postamble:	none

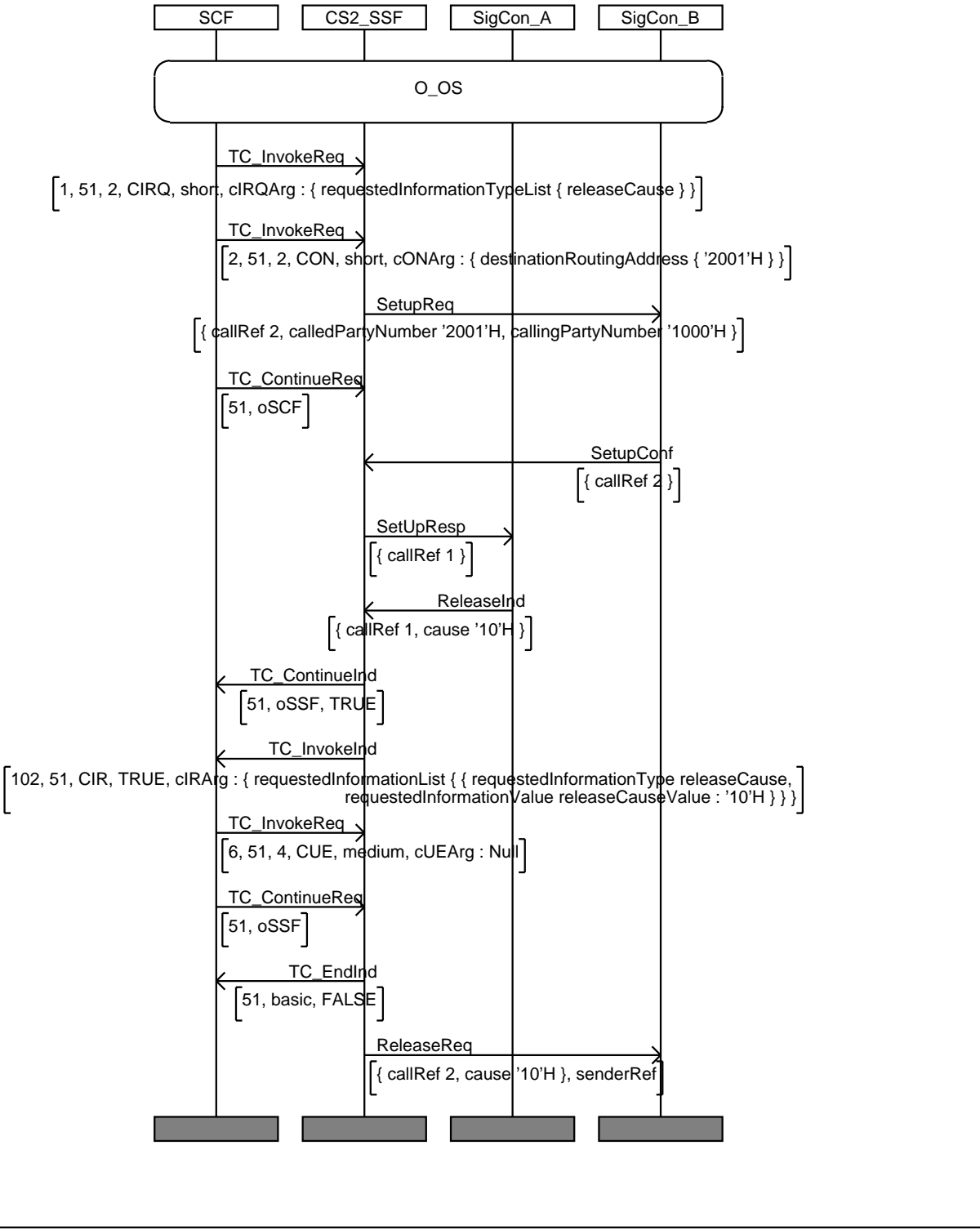
MSC IN2m_A_BASIC_CG_BI_01



6.4.5 CallInformation procedure

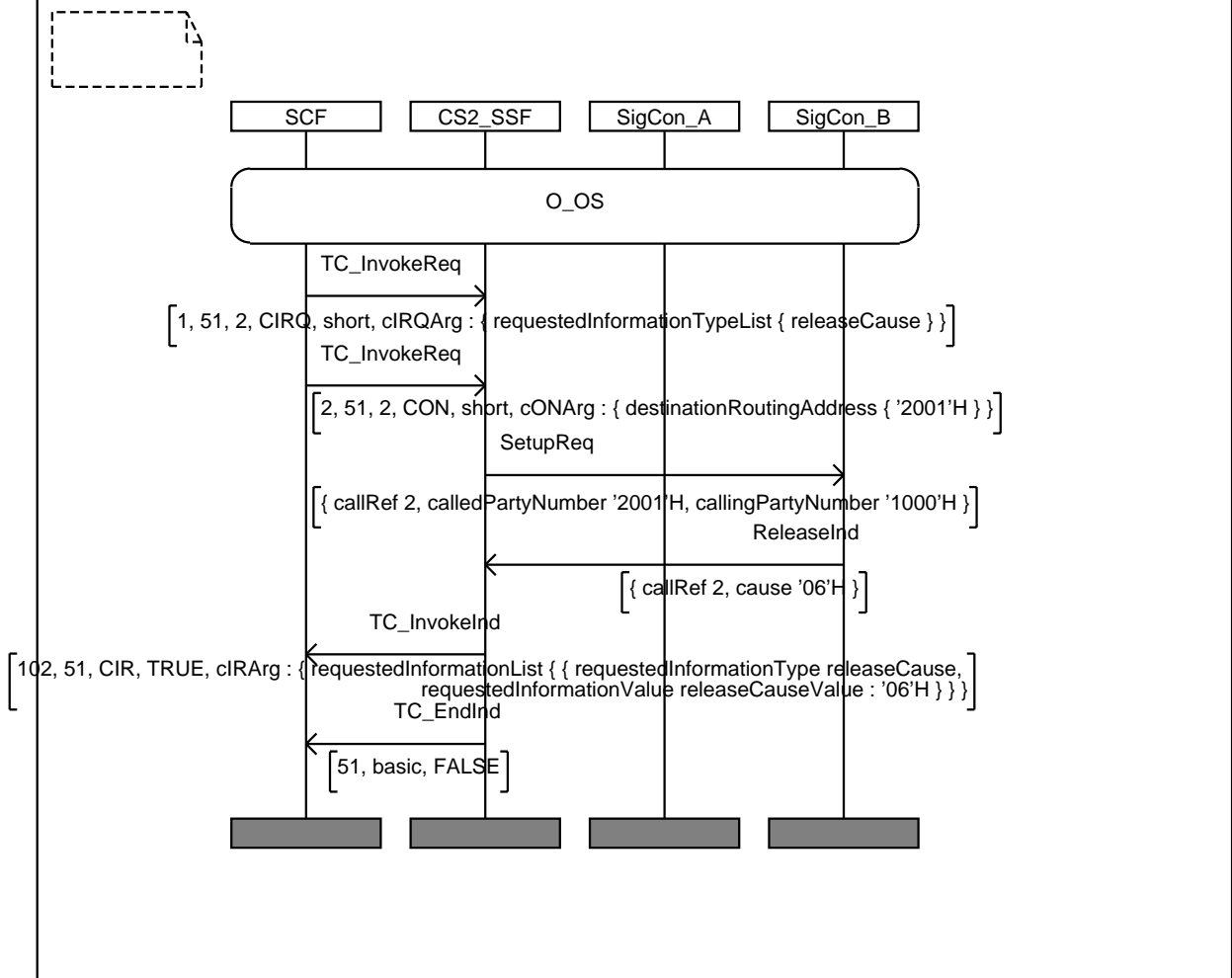
IN2_A_BASIC_CF_CA_01	
Purpose:	Test of CallInformation procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>SCF sends to SSF a CallInformationRequest invoke, containing mandatory parameters only and indicating a single information type, with at least the parameters:</p> <ul style="list-style-type: none"> - requestedInformationTypeList including: <ul style="list-style-type: none"> - requestedInformationType being releaseCause, <p>followed by a Connect to establish a Connection with SigConB When the connection is established, SigConA releases the call</p>
Pass criteria	<ul style="list-style-type: none"> - Check that upon detection of call release, SSF sends CallInformationReport with at least the parameters - requestedInformationList including: <ul style="list-style-type: none"> - requestedInformationType being releaseCause, - requestedInformationValue being releaseCauseValue used <p>then SSF becomes idle.</p>
Postamble:	none

MSC IN2m_A_BASIC_CF_CA_01



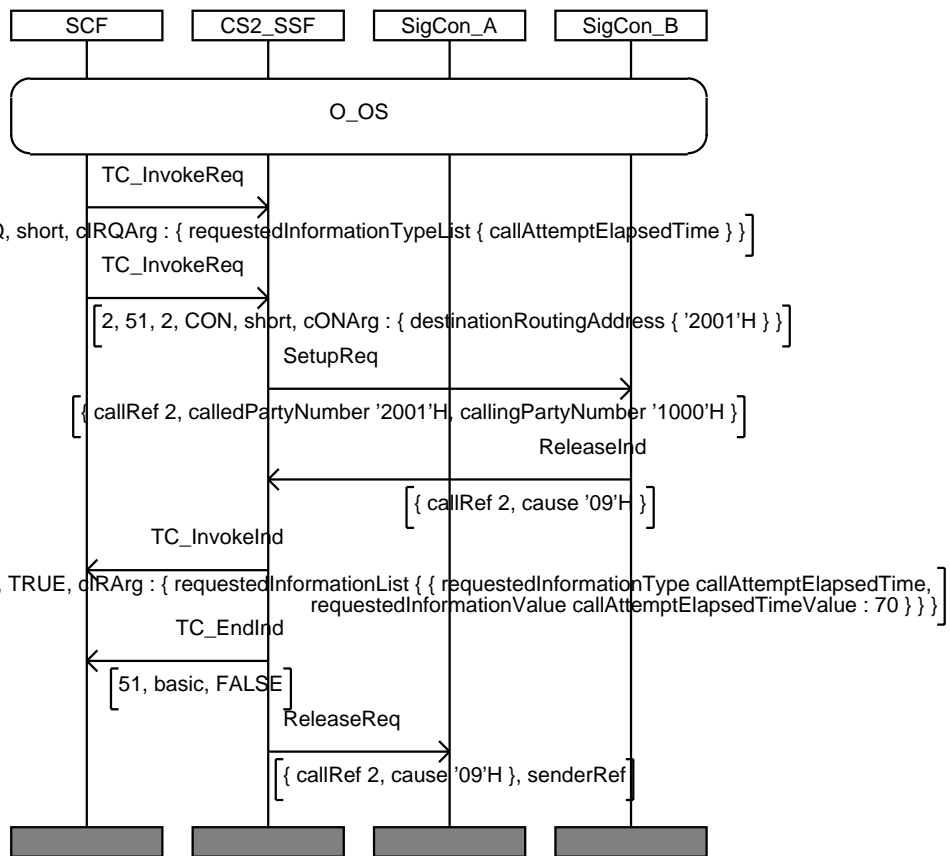
IN2_A_BASIC_CF_BV_01	
Purpose:	Test of CallInformation procedure and release parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS In addition, user B is declared busy
Test description	SCF sends to SSF a CallInformationRequest invoke, containing mandatory parameters only and indicating a single information type, with at least the parameters: <ul style="list-style-type: none"> - requestedInformationTypeList including: <ul style="list-style-type: none"> - requestedInformationType being releaseCause, followed by a Continue to establish a Connection with SigConB But the connection is not established, as B is busy
Pass criteria	- Check that upon detection of call release, SSF sends CallInformationReport with at least the parameters <ul style="list-style-type: none"> - requestedInformationList including: <ul style="list-style-type: none"> - requestedInformationType being releaseCause, - requestedInformationValue being releaseCauseValue used
Postamble:	none

MSC IN2m_A_BASIC_CF_BV_01



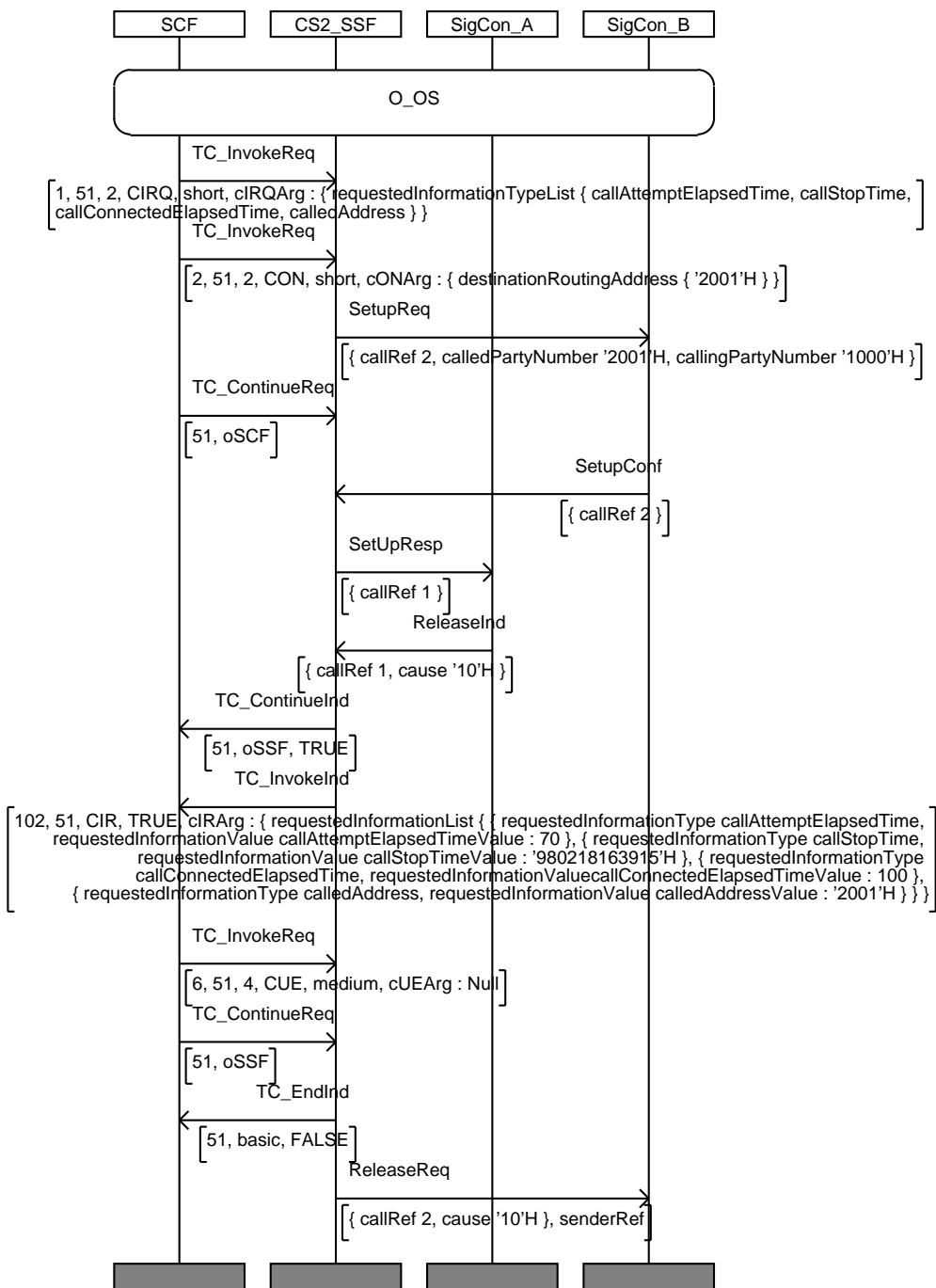
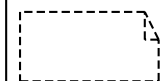
IN2_A_BASIC_CF_BV_02	
Purpose:	Test of CallInformation procedure and time parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS In addition, user B is declared not answering within timeout
Test description	SCF sends to SSF a CallInformationRequest invoke, containing mandatory parameters only and indicating a single information type, with at least the parameters: <ul style="list-style-type: none"> - requestedInformationTypeList including: <ul style="list-style-type: none"> - requestedInformationType (callAttemptElapsedTime), followed by a Continue to establish a Connection with SigConB But the connection is not established, as B does not answer
Pass criteria	- Check that upon detection of SSF timer expiration, SSF sends CallInformationReport with at least the parameters <ul style="list-style-type: none"> - requestedInformationList including: <ul style="list-style-type: none"> - requestedInformationType (callAttemptElapsedTime) - requestedInformationValue being callAttemptElapsedTimeValue,
Postamble:	none

MSC IN2m_A_BASIC_CF_BV_02



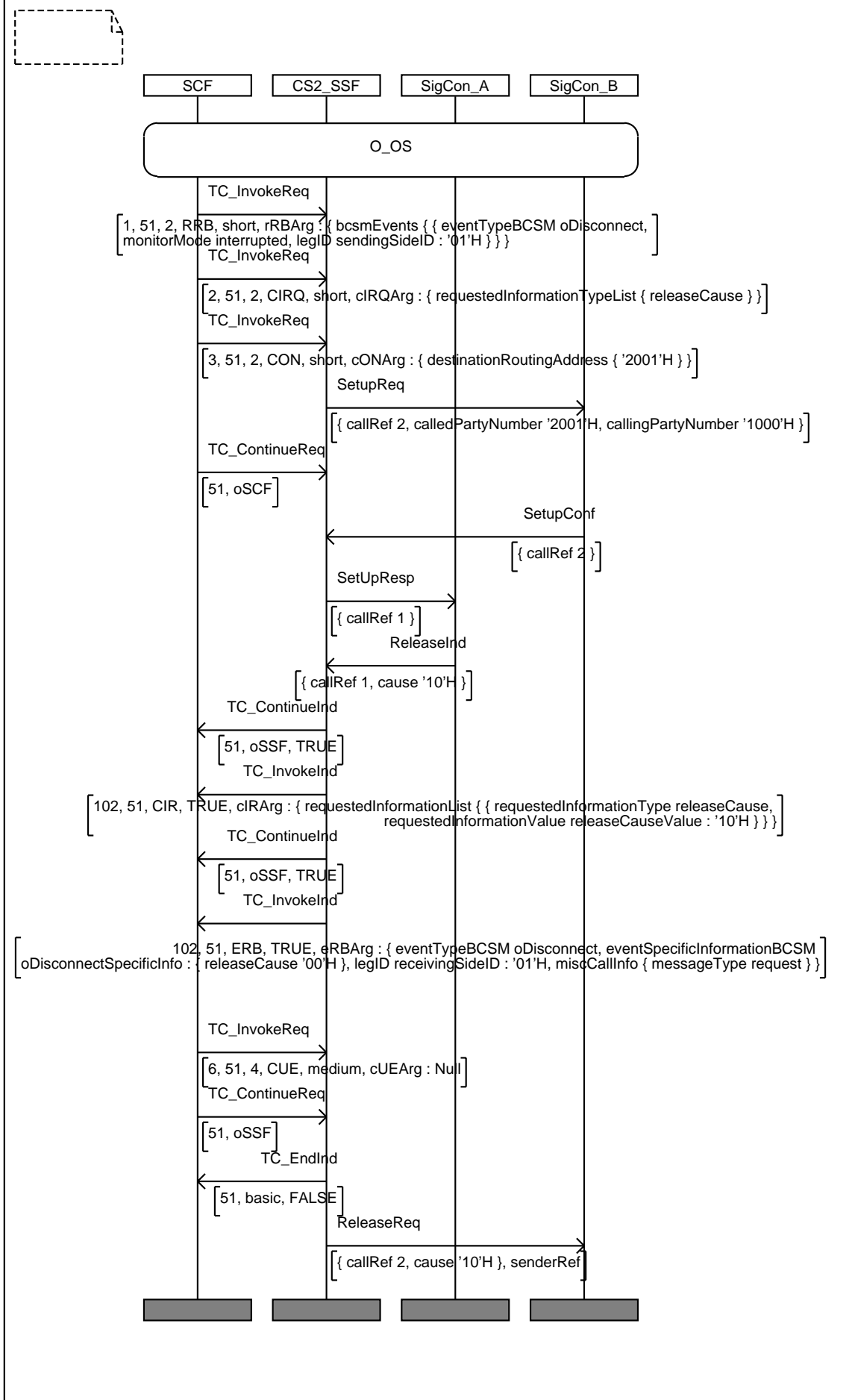
IN2_A_BASIC_CF_BV_03	
Purpose:	Test of CallInformation procedure and elapsed time parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>SCF sends to SSF a CallInformationRequest invoke, containing mandatory parameters and indicating a multiple information type, with at least the parameters:</p> <ul style="list-style-type: none"> - requestedInformationTypeList including: <ul style="list-style-type: none"> - requestedInformationType (callAttemptElapsedTime), also including: <ul style="list-style-type: none"> - requestedInformationType (callStopTime), also including: <ul style="list-style-type: none"> - requestedInformationType (callConnectedElapsedTime), and including: <ul style="list-style-type: none"> - requestedInformationType (calledAddress), <p>followed by a Continue to establish a Connection with SigConB</p>
Pass criteria	<ul style="list-style-type: none"> - Check that upon detection of a release from SigConA, SSF sends CallInformationReport to SCF and indicating a multiple information type, with at least the parameters: <ul style="list-style-type: none"> - requestedInformationList including: <ul style="list-style-type: none"> - requestedInformationType (callAttemptElapsedTime), - requestedInformationValue being callAttemptElapsedTimeValue, also including: <ul style="list-style-type: none"> - requestedInformationType (callStopTime), - requestedInformationValue being callStopTimeValue, also including: <ul style="list-style-type: none"> - requestedInformationType (callConnectedElapsedTime), - requestedInformationValue being callConnectedElapsedTime Value, and including: <ul style="list-style-type: none"> - requestedInformationType (calledAddress), - requestedInformationValue being calledAddressValue
Postamble:	none

MSC IN2m_A_BASIC_CF_BV_03



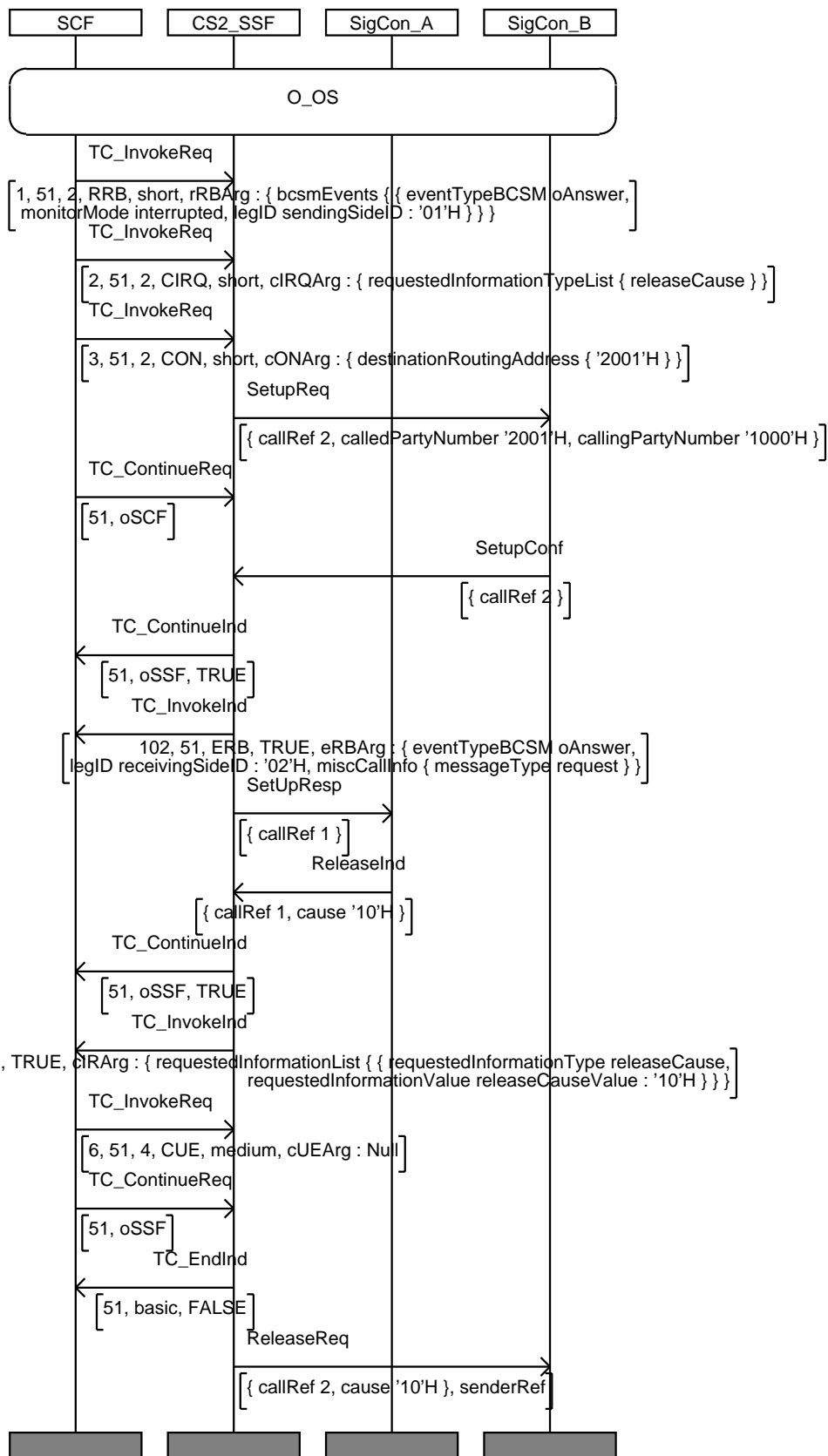
IN2_A_BASIC_CF_BV_04	
Purpose:	Test of CallInformation procedure combined with RequestReportBCSMEvent procedure.
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oDisconnect - monitoringMode=interrupted <p>then SCF sends a CallInformationRequest invoke, containing mandatory parameters only, with at least the parameters:</p> <ul style="list-style-type: none"> - requestedInformationTypeList including: - requestedInformationType being releaseCause, <p>followed by a Connect invoke and SSF establishes the call The call is answered (SigCon B sends SetupConf) SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)</p>
Pass criteria	<ul style="list-style-type: none"> - Check that upon detection of call release, SSF sends CallInformationReport with at least the parameters - requestedInformationList including: - requestedInformationType being releaseCause, - requestedInformationValue being releaseCauseValue used <p>then Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oDisconnect</p>
Postamble:	none

MSC IN2m_A_BASIC_CF_BV_04



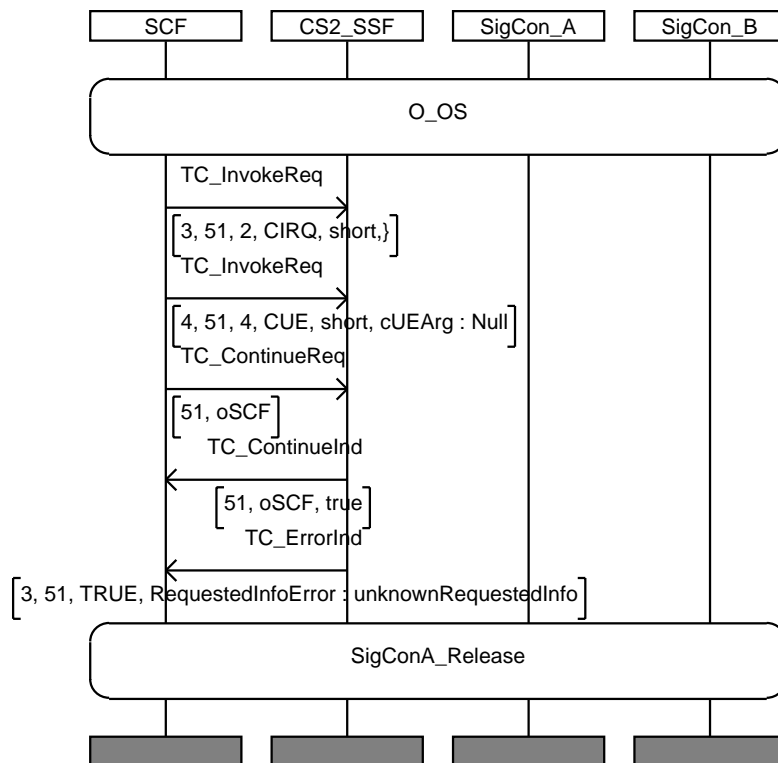
IN2_A_BASIC_CF_BV_05	
Purpose:	Test of CallInformation procedure combined with RequestReportBCSMEvent procedure.
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oAnswer - monitoringMode=interrupted <p>then SCF sends a CallInformationRequest invoke, containing mandatory parameters only, with at least the parameters:</p> <ul style="list-style-type: none"> - requestedInformationTypeList including: <ul style="list-style-type: none"> - requestedInformationType being releaseCause, <p>followed by a Connect invoke and SSF establishes the call The call is answered (SigCon B sends SetupConf) SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)</p>
Pass criteria	<ul style="list-style-type: none"> - Check that when SigConB is answering, SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oAnswer - Check that upon detection of call release from SigConA, SSF sends CallInformationReport with at least the parameters <ul style="list-style-type: none"> - requestedInformationList including: <ul style="list-style-type: none"> - requestedInformationType being releaseCause, - requestedInformationValue being releaseCauseValue used
Postamble:	none

MSC IN2m_A_BASIC_CF_BV_05



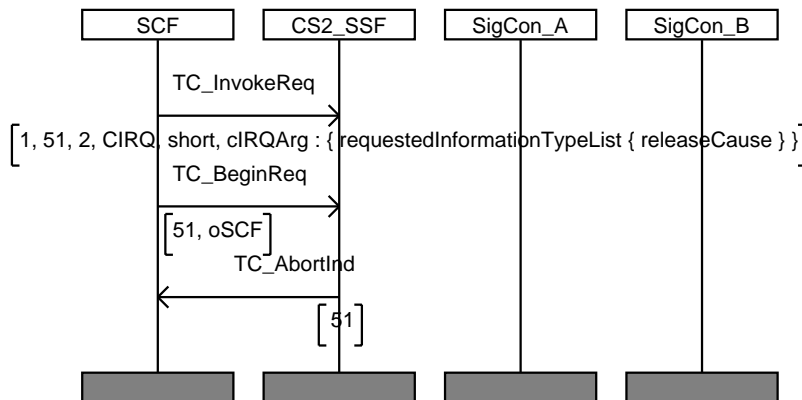
IN2_A_BASIC_CF_BI_01	
Purpose:	Test of CallInformation procedure with invalid parameters
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF a CallInformationRequest invoke, with <ul style="list-style-type: none"> - RequestedInformationTypeList, containing an invalid parameter followed by a Continue to establish a Connection with SigConB
Pass criteria	- Check that SSF sends back CallInformationRequest error to SCF indicating: requestedInfoError
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_CF_BI_01



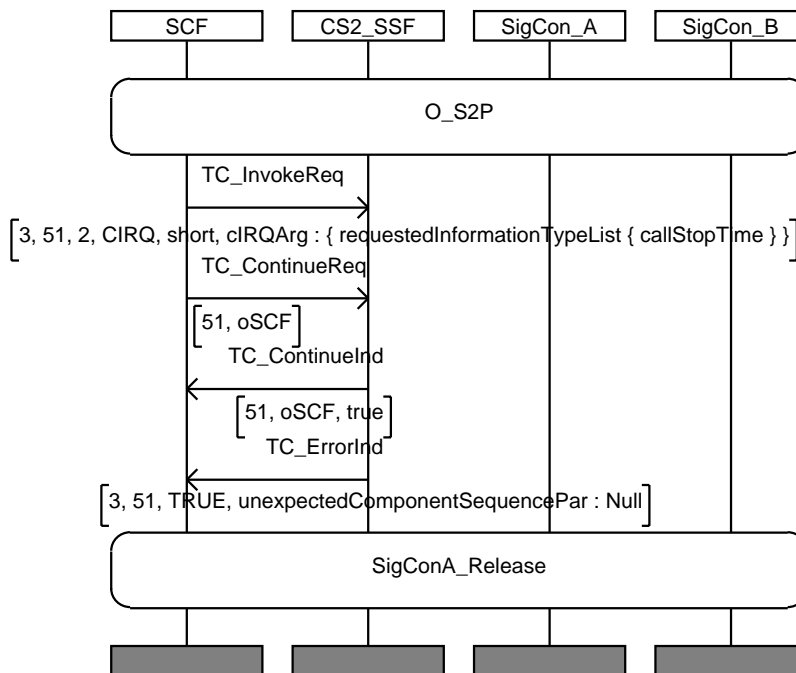
IN2_A_BASIC_CF_BO_01	
Purpose:	Test of CallInformation procedure in wrong state (idle)
Requirement ref	
Selection Cond.	
Preamble:	None
Test description	SCF sends CallInformationRequest invoke to SSF
Pass criteria	Check that SSF sends to SCF a TC_ABORT
Postamble:	None

MSC IN2m_A_BASIC_CF_BO_01



IN2_A_BASIC_CF_BO_02	
Purpose:	Test of CallInformation procedure in wrong state (monitoring)
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	SCF sends CallInformationRequest invoke to SSF
Pass criteria	Check that SSF sends to SCF a CallInformationRequest error with an indication of UnexpectedComponentSequence
Postamble:	SigConA_Release_thenB

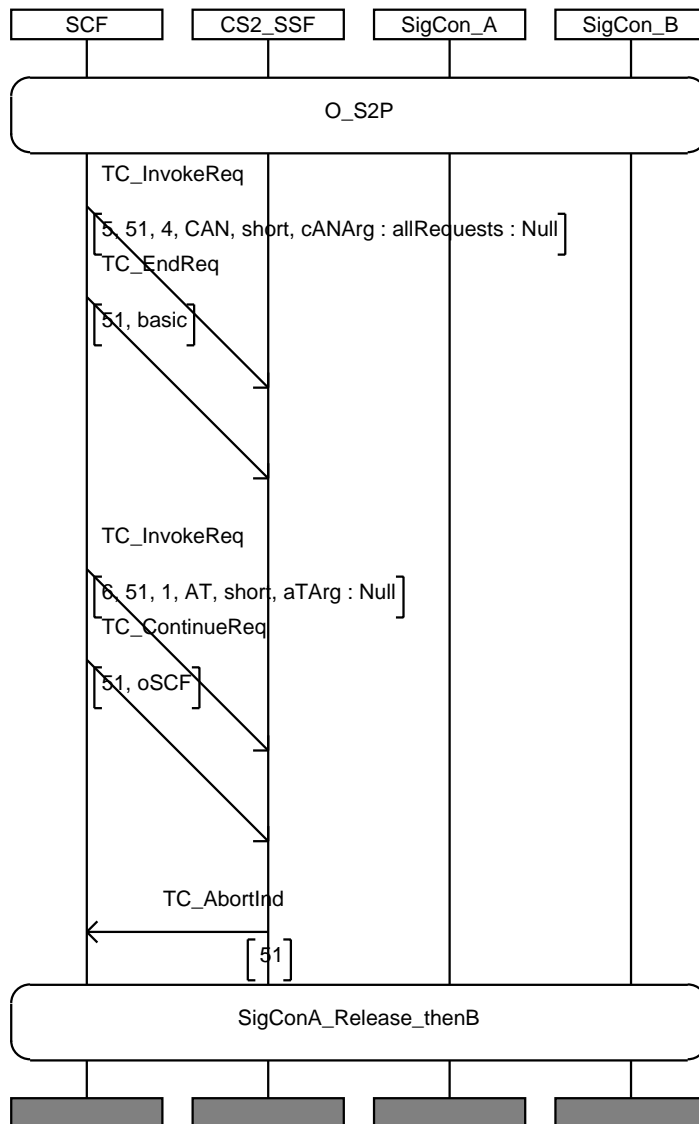
MSC IN2m_A_BASIC_CF_BO_02



6.4.6 Cancel procedure

IN2_A_BASIC_CA_CA_01	
Purpose:	Test of Cancel base procedure
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	Cancel invoke sent by SCF to SSF, containing allRequests
Pass criteria	- Check that SSF returns to idle state - To ensure that SSF is now in idle state, SCF sends ActivityTest invoke to SSF with DialogueId used in InitialDP. SSF rejects or aborts the invoke as dialogue is not used any more
Postamble:	SigConA_Release_thenB

MSC IN2_A_BASIC_CA_CA_01

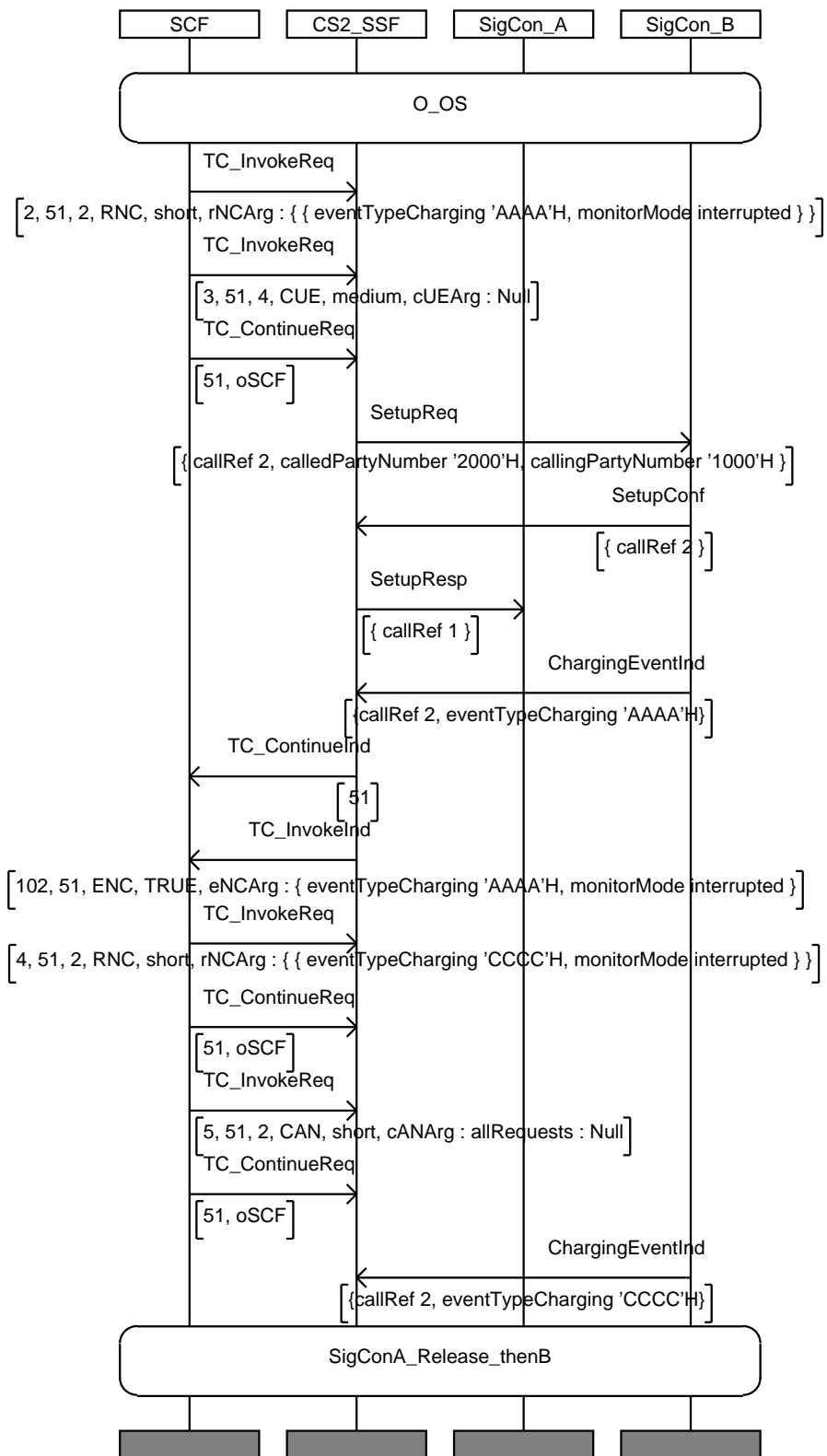


IN2_A_BASIC_CA_BV_01

This test purpose was dropped.

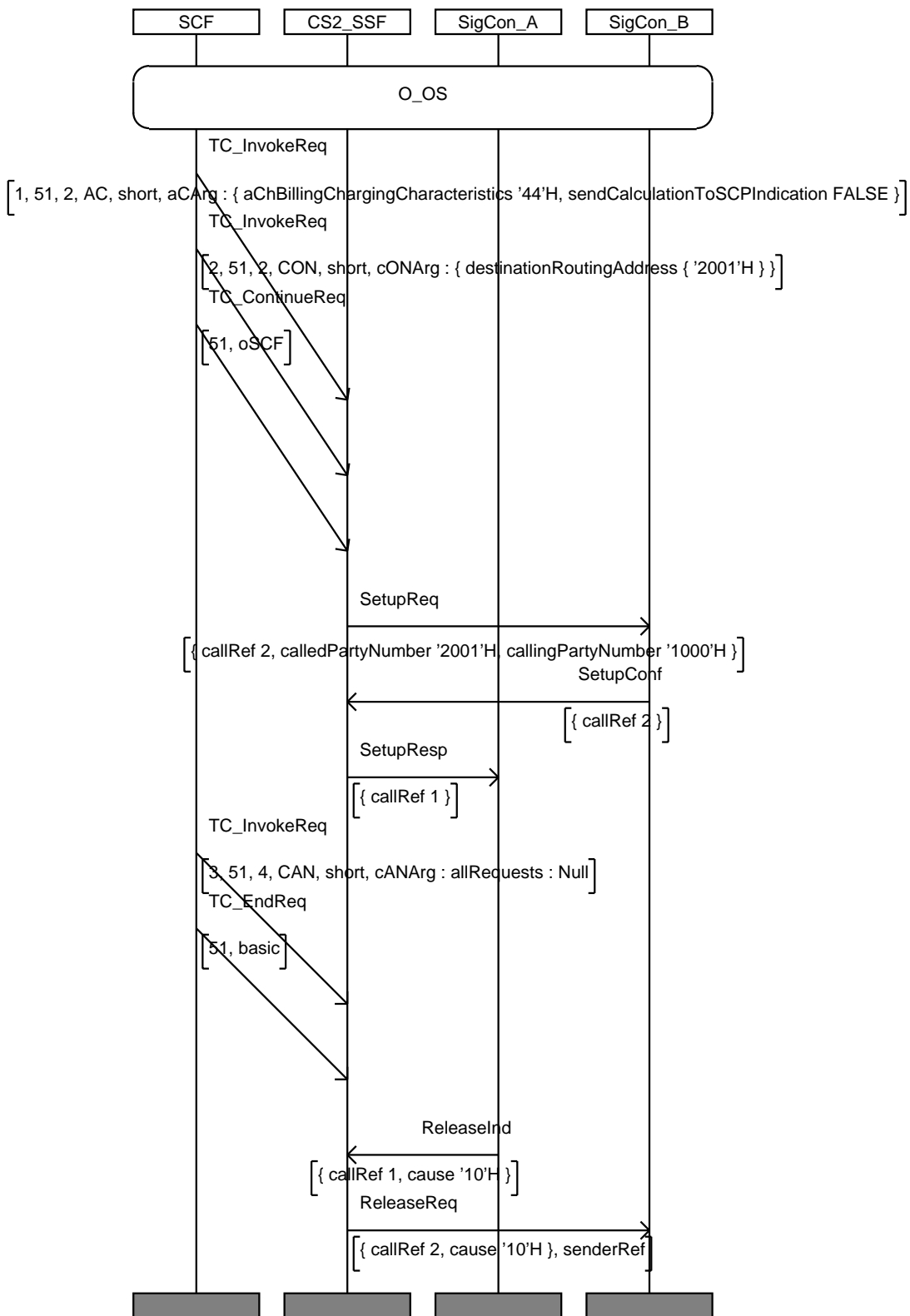
IN2_A_BASIC_CA_BV_02	
Purpose:	Test of Cancel procedure on RequestNotificationChargingEvent
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>SCF sends to SSF RequestNotificationChargingEvent invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - eventTypeCharging, - monitorMode (interrupted) <p>After triggering of charging event from SigConA, SSF sends to SCF an EventNotificationCharging invoke with the indication of eventTypeCharging</p> <p>SCF sends to SSF Continue invoke then a new RequestNotificationChargingEvent invoke containing mandatory parameters only, with:</p> <ul style="list-style-type: none"> - eventTypeCharging, - monitorMode (interrupted) <p>followed by a Cancel invoke containing allRequests</p>
Pass criteria	- Check that SSF cancels the request for an EventNotificationCharging and does not send it to SCF when the calling party (SigConA) triggers the charging event.
Postamble:	SigConA_Release_thenB

MSC IN2m_A_BASIC_CA_BV_02



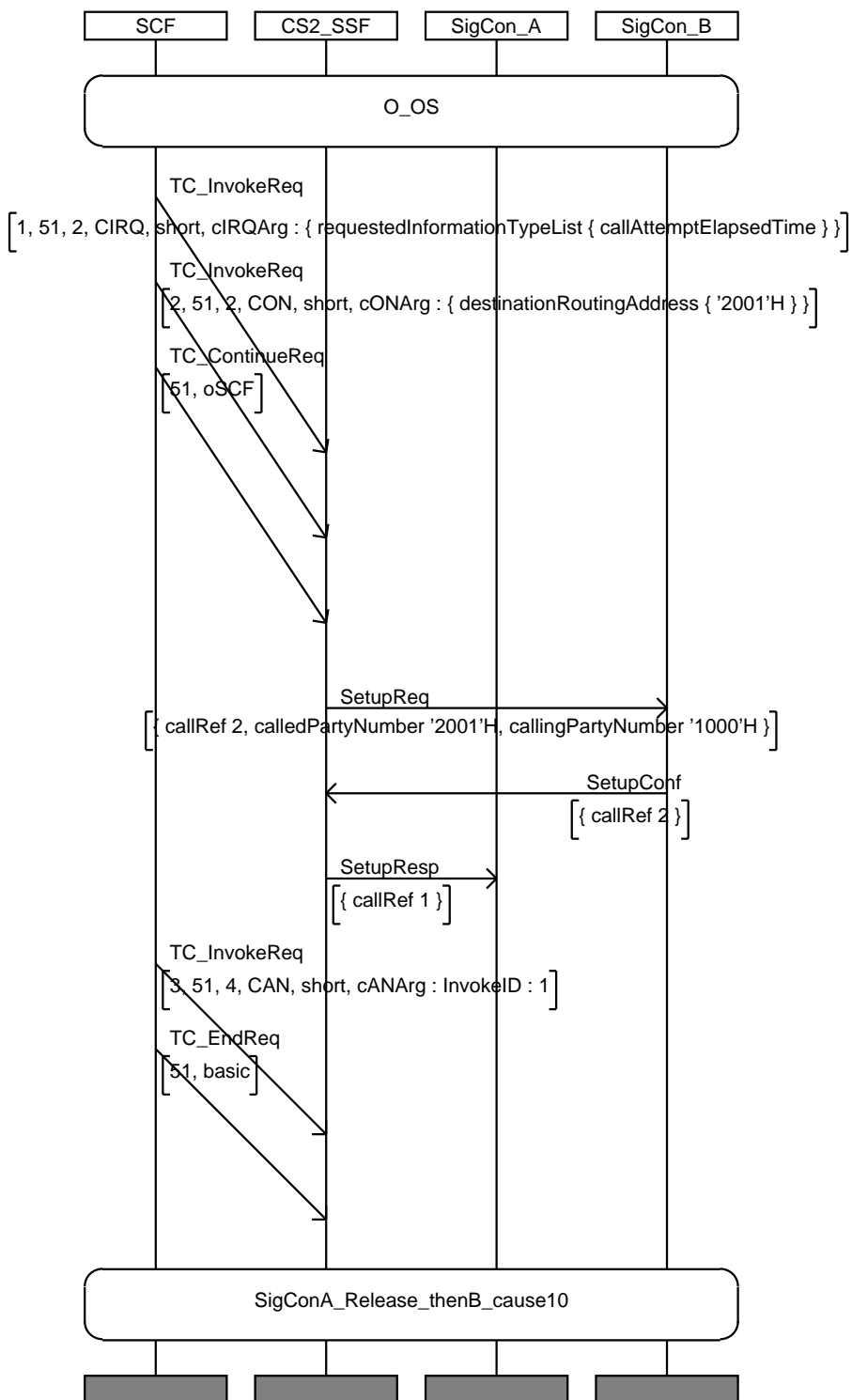
IN2_A_BASIC_CA_BV_03	
Purpose:	Test of Cancel procedure on ApplyCharging
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF ApplyCharging invoke containing mandatory parameter aChBillingChargingCharacteristics followed by a Connect to establish a Connection with SigConB When the connection is established, Cancel invoke is sent by SCF to SSF, containing allRequests
Pass criteria	- Check that SSF cancels the request for an ApplyChargingReport and does not send it to SCF when the calling party (SigConA) releases the call.
Postamble:	none

MSC IN2_A_BASIC_CA_BV_03



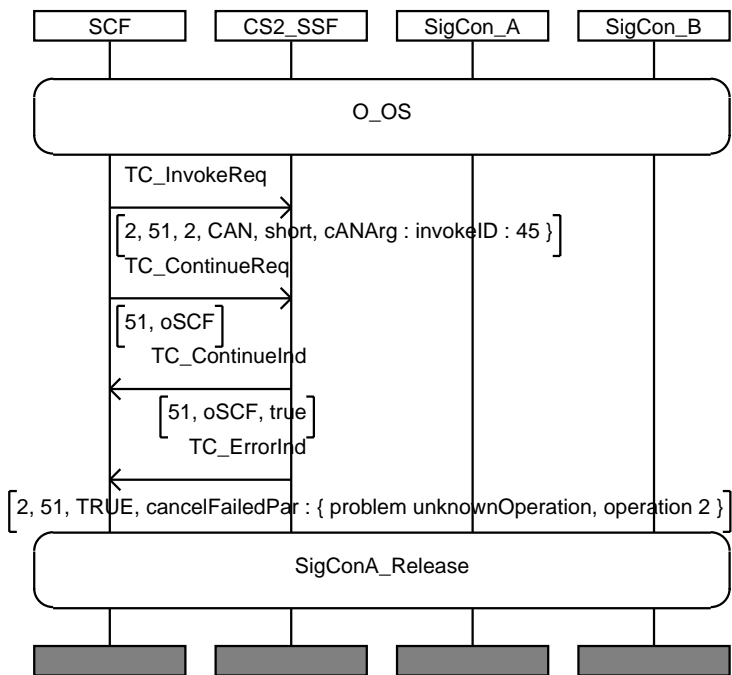
IN2_A_BASIC_CA_BV_04	
Purpose:	Test of Cancel procedure on CallInformationRequest
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF CallInformationRequest invoke containing mandatory parameters requestedInformationTypeList including: re requestedInformationType (callAttemptElapsedTime), followed by a Connect to establish a Connection with SigConB When the connection is established, Cancel invoke is sent by SCF to SSF, containing invokeID
Pass criteria	- Check that SSF cancels the request for an CallInformationReport and does not send it to SCF when the calling party (SigConA) releases the call.
Postamble:	SigConA_Release_thenB_cause10

MSC IN2_A_BASIC_CA_BV_04



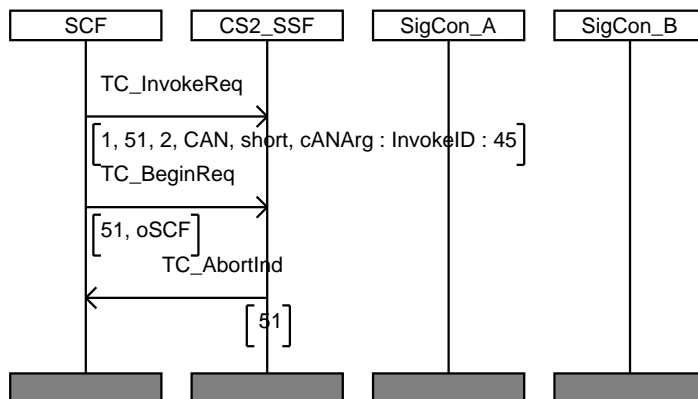
IN2_A_BASIC_CA_BI_01	
Purpose:	Test of Cancel error procedure with cancelFailed
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	Cancel invoke is sent by SCF to SSF, containing invokeID being not existing operation invoke Id
Pass criteria	- Check that SSF sends to SCF Cancel with error cancelFailed
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_CA_BI_01



IN2_A_BASIC_CA_BO_01	
Purpose:	Test of Cancel procedure in wrong (idle) state
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	Cancel invoke sent by SCF to SSF, containing invokeID
Pass criteria	- Check that SSFsends a TC-ABORT
Postamble:	none

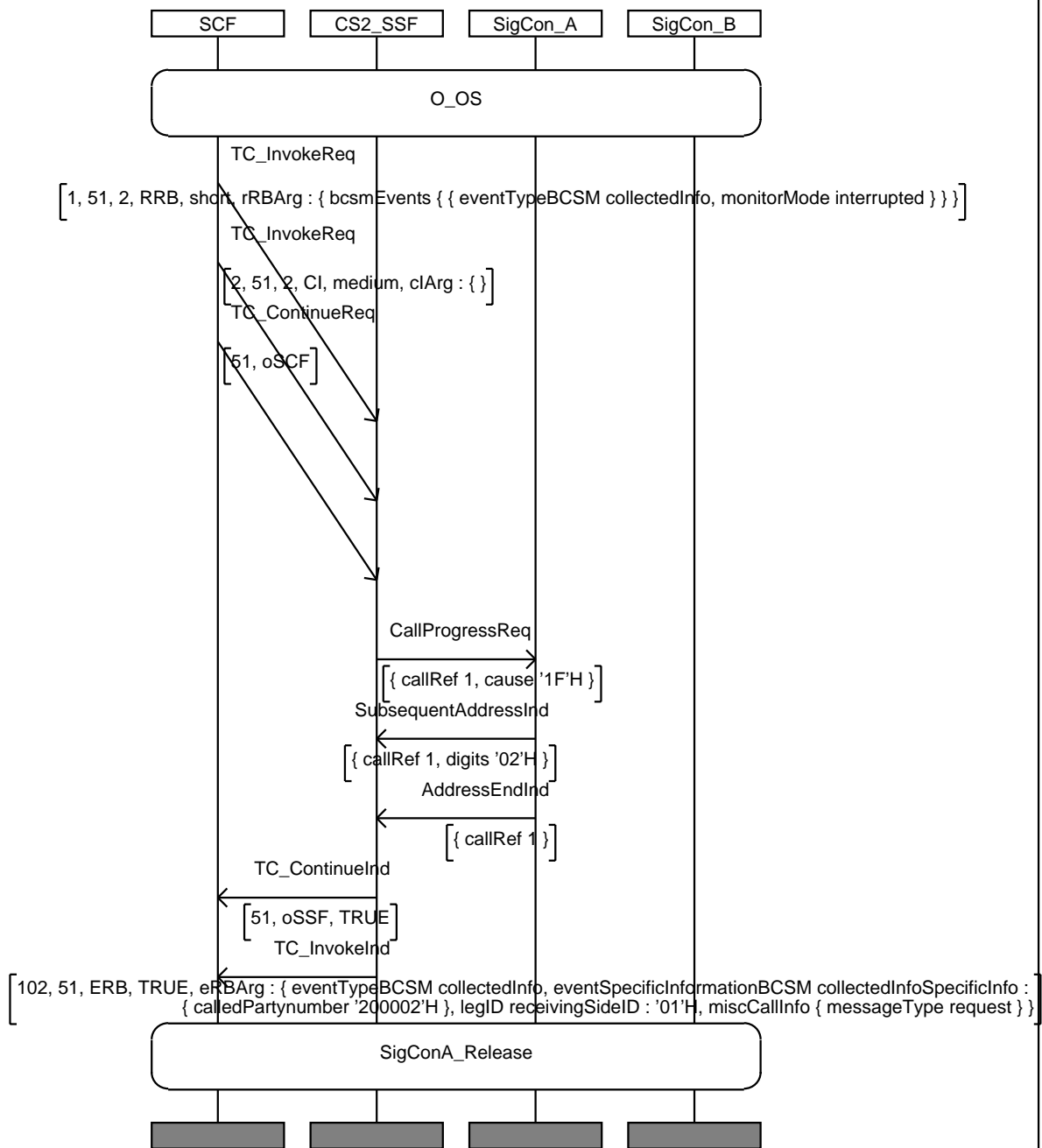
MSC IN2m_A_BASIC_CA_BO_01



6.4.7 CollectInformation procedure

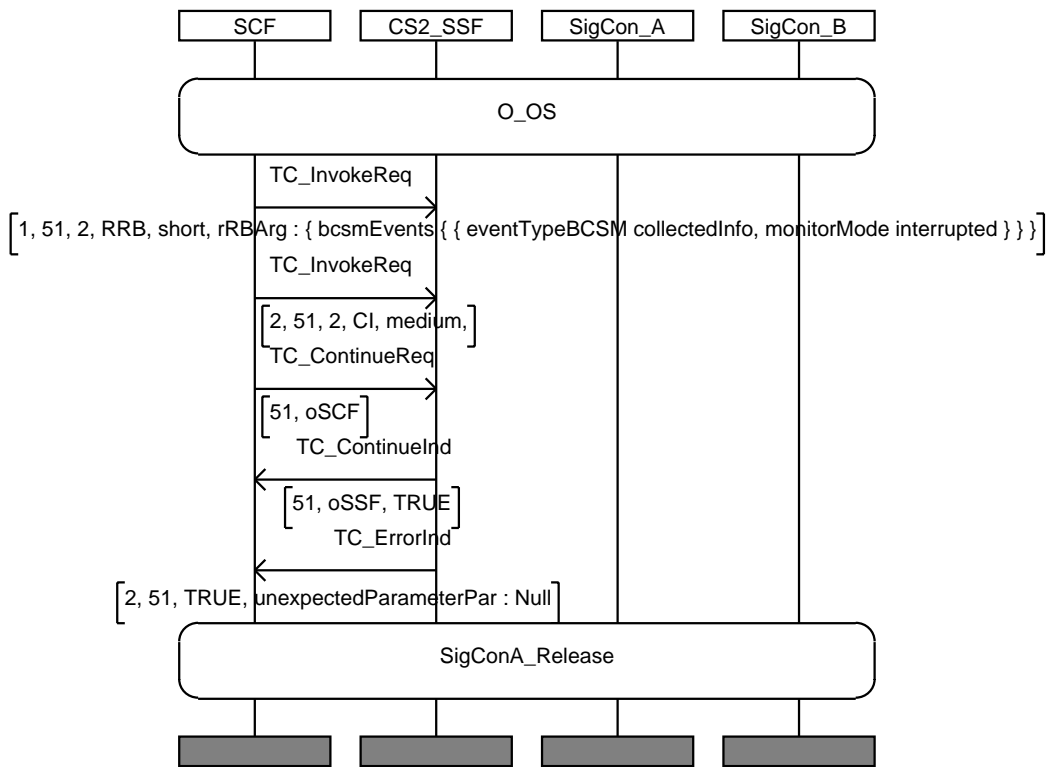
IN2_A_BASIC_CI_CA_01	
Purpose:	Test of CollectInformation base procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS Preamble contains an InitialDP without complete digits for CalledPartyNumber
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=collectedInfo - monitoringMode=interrupted followed by CollectInformation invoke then the calling party sends the remaining digits (after CallProgressReq is received and SubsequentAddressInd and AddressEndInd is sent)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=collectedInfo, together with the remaining called party digits
Postamble:	SigConA_Release

MSC IN2_A_BASIC_CI_CA_01



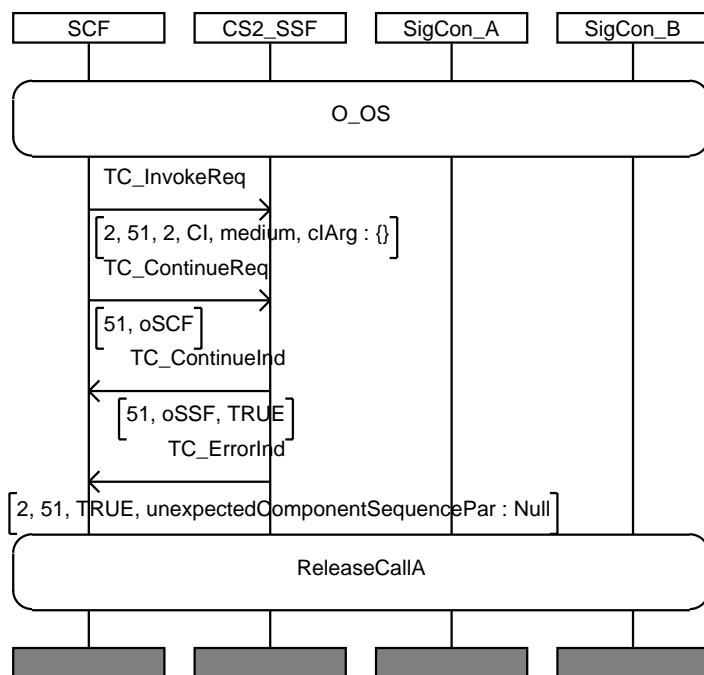
IN2_A_BASIC_CI_BI_01	
Purpose:	Test of CollectInformation procedure and unknown CallsegmentId
Requirement ref	
Selection Cond.	CS-2 only
Preamble:	O_OS
Test description	SCF - SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM= collectedInfo - monitoringMode=interrupted followed by CollectInformation invoke with - callSegmentID being a not existing call segment
Pass criteria	- Check that SSF sends to SCF a TC_ErrorInd with unexpectedParameter
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_CI_BI_01



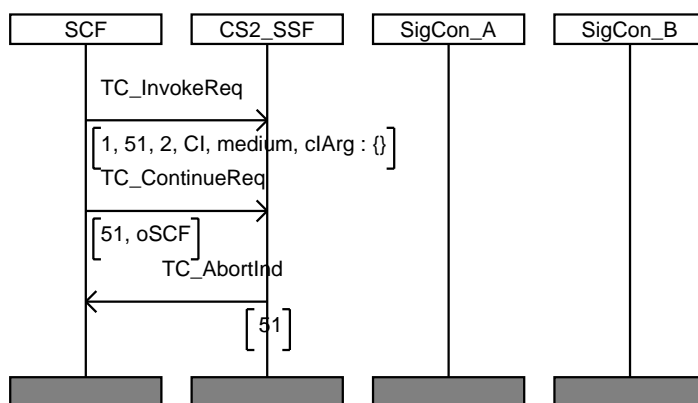
IN2_A_BASIC_CI_BO_01	
Purpose:	Test of CollectInformation procedure in wrong sequence
Requirement ref	
Selection Cond.	
Preamble:	O_OS Preamble contains an InitialDP without complete digits for CalledPartyNumber
Test description	SCF sends CollectInformation invoke to SSF without sending before any RequestReportBCSMEEvent invoke
Pass criteria	Check that SSF sends to SCF a CollectInformation error with an indication of UnexpectedComponentSequence
Postamble:	ReleaseCallA

MSC IN2m_A_BASIC_CI_BO_01

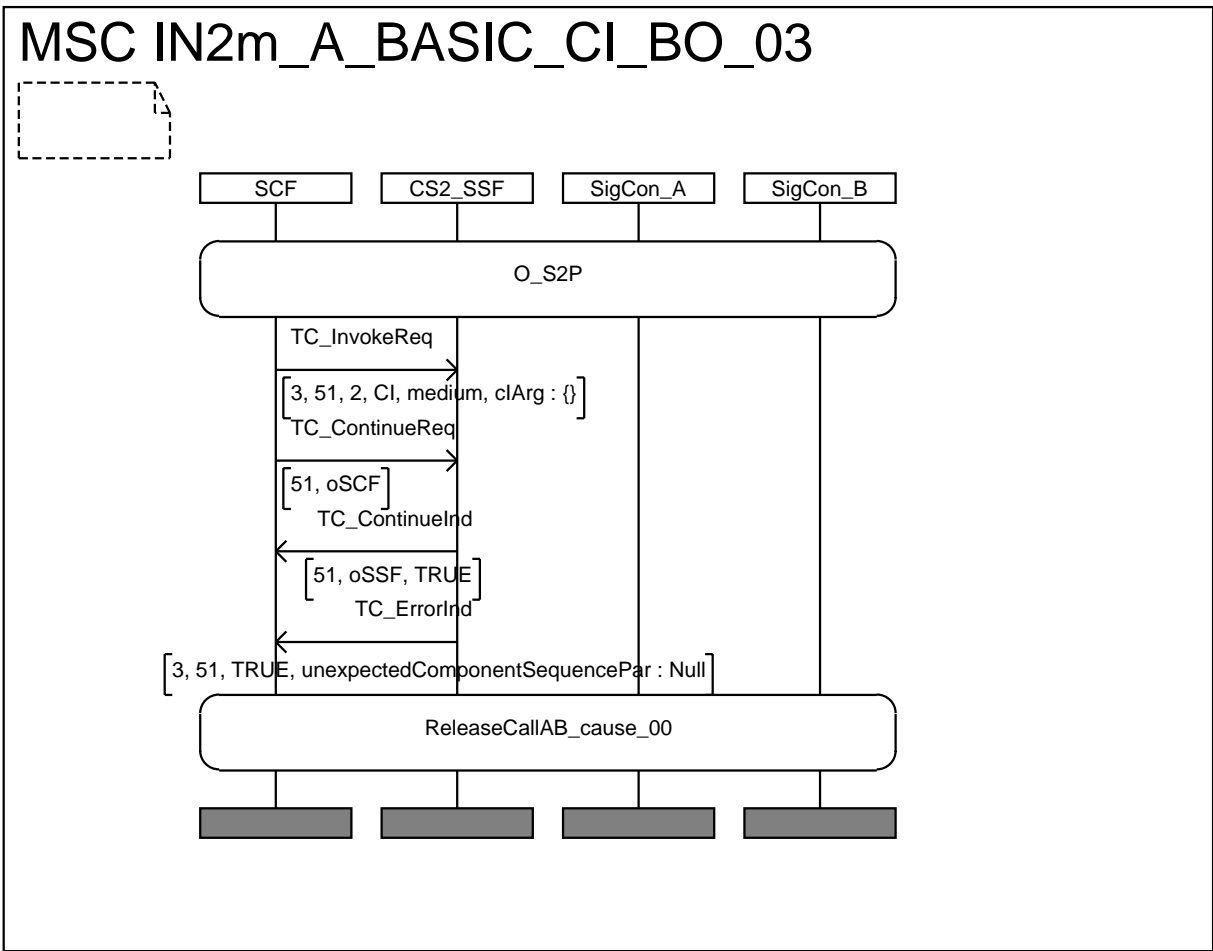


IN2_A_BASIC_CI_BO_02	
Purpose:	Test of CollectInformation procedure in wrong state (idle state)
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends CollectInformation invoke to SSF from idle state
Pass criteria	Check that SSF sends to SCF a TC-ABORT
Postamble:	None

MSC IN2m_A_BASIC_CI_BO_02



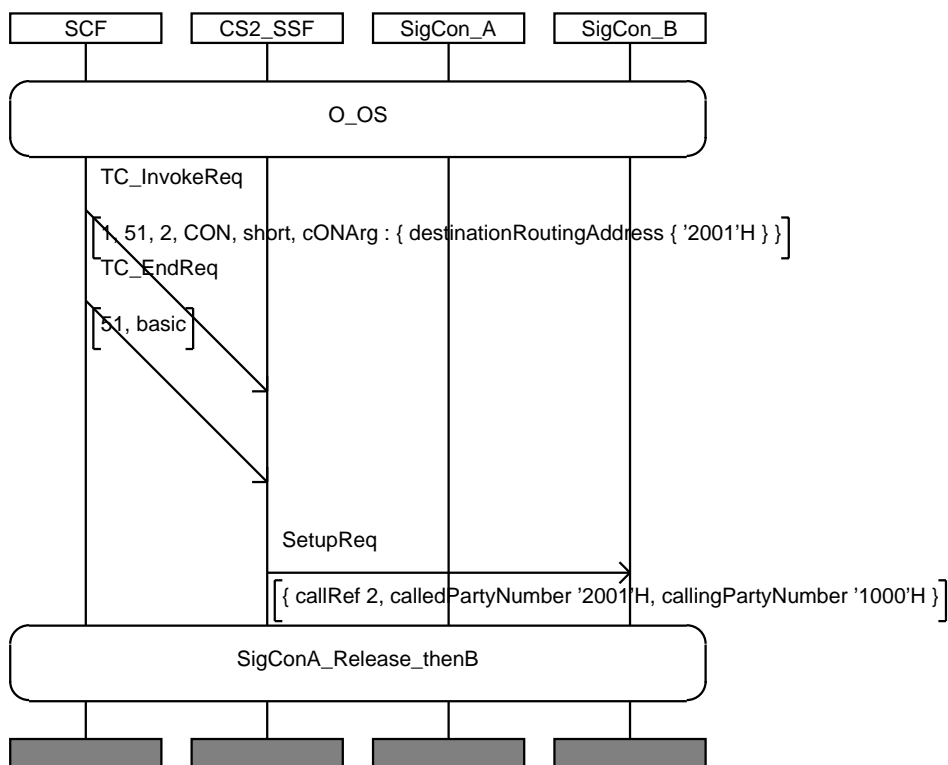
IN2_A_BASIC_CI_BO_03	
Purpose:	Test of CollectInformation procedure in wrong state (monitoring state)
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	SCF sends CollectInformation invoke to SSF from Monitoring state
Pass criteria	Check that SSF sends to SCF a CollectInformation error with an indication of UnexpectedComponentSequence
Postamble:	ReleaseCallAB_cause_00



6.4.8 Connect procedure

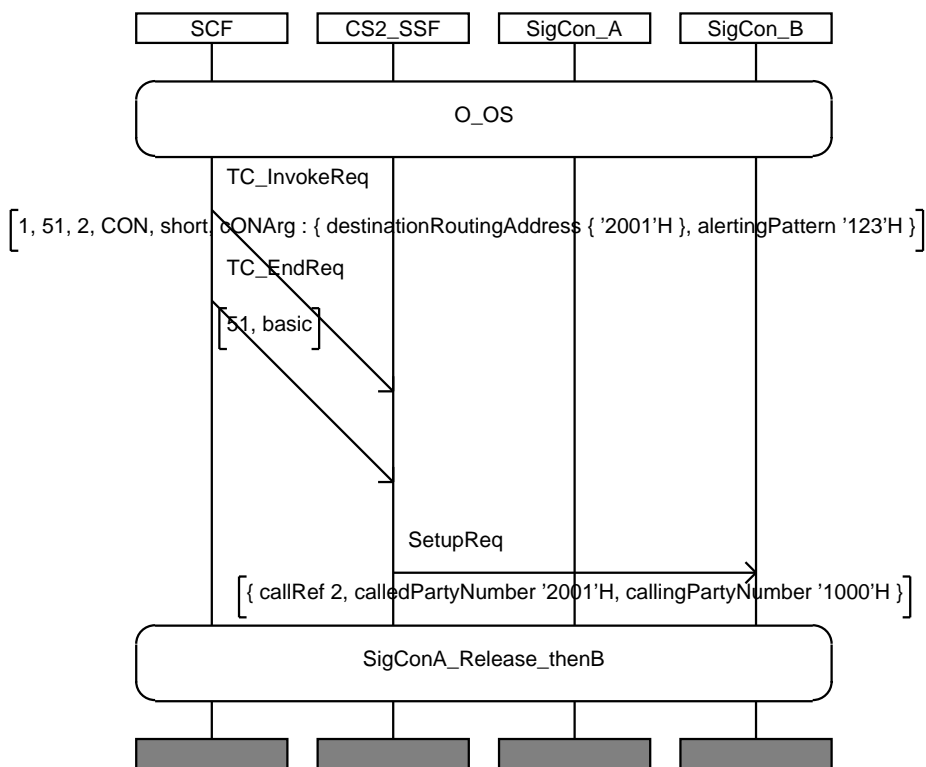
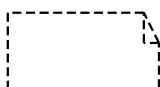
IN2_A_BASIC_CO_CA_01	
Purpose:	Test of Connect base procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Connect invoke with mandatory parameters only, with destinationRoutingAddress SSF sends a SetupRequest to B side
Pass criteria	Check that the relevant parameters are mapped from Connect into the Set-up request
Postamble:	SigConA_Release_thenB

MSC IN2_A_BASIC_CO_CA_01



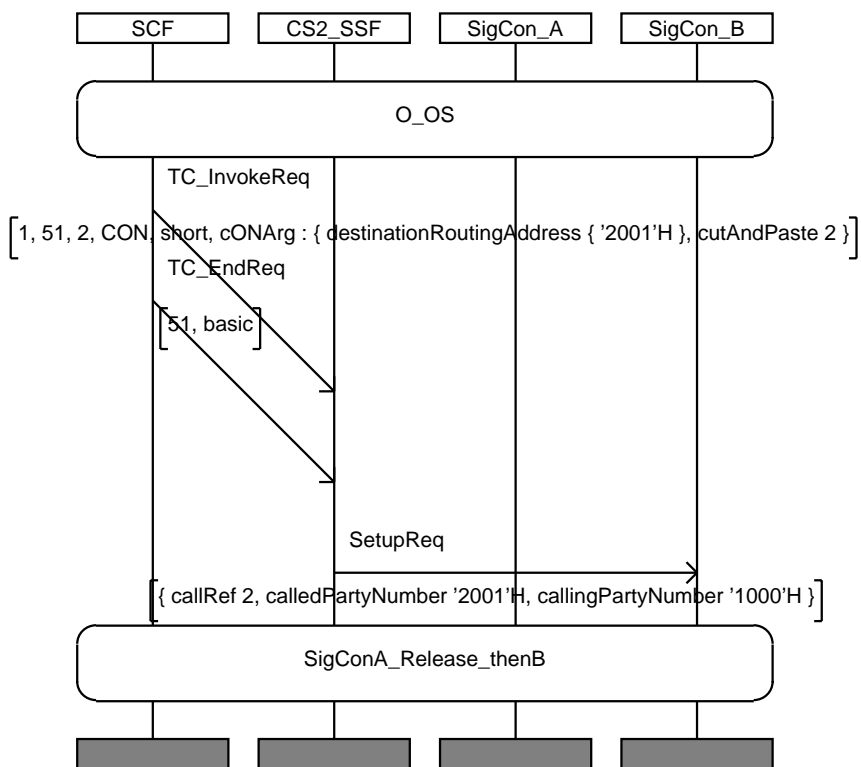
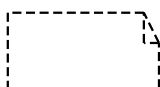
IN2_A_BASIC_CO_BV_01	
Purpose:	Test of Connect procedure with alertingPattern parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Connect invoke with mandatory and optional parameters, with destinationRoutingAddress alertingPattern SSF sends a SetupRequest to B side
Pass criteria	Check that the connect operation is not rejected
Postamble:	SigConA_Release_thenB

MSC IN2_A_BASIC_CO_BV_01

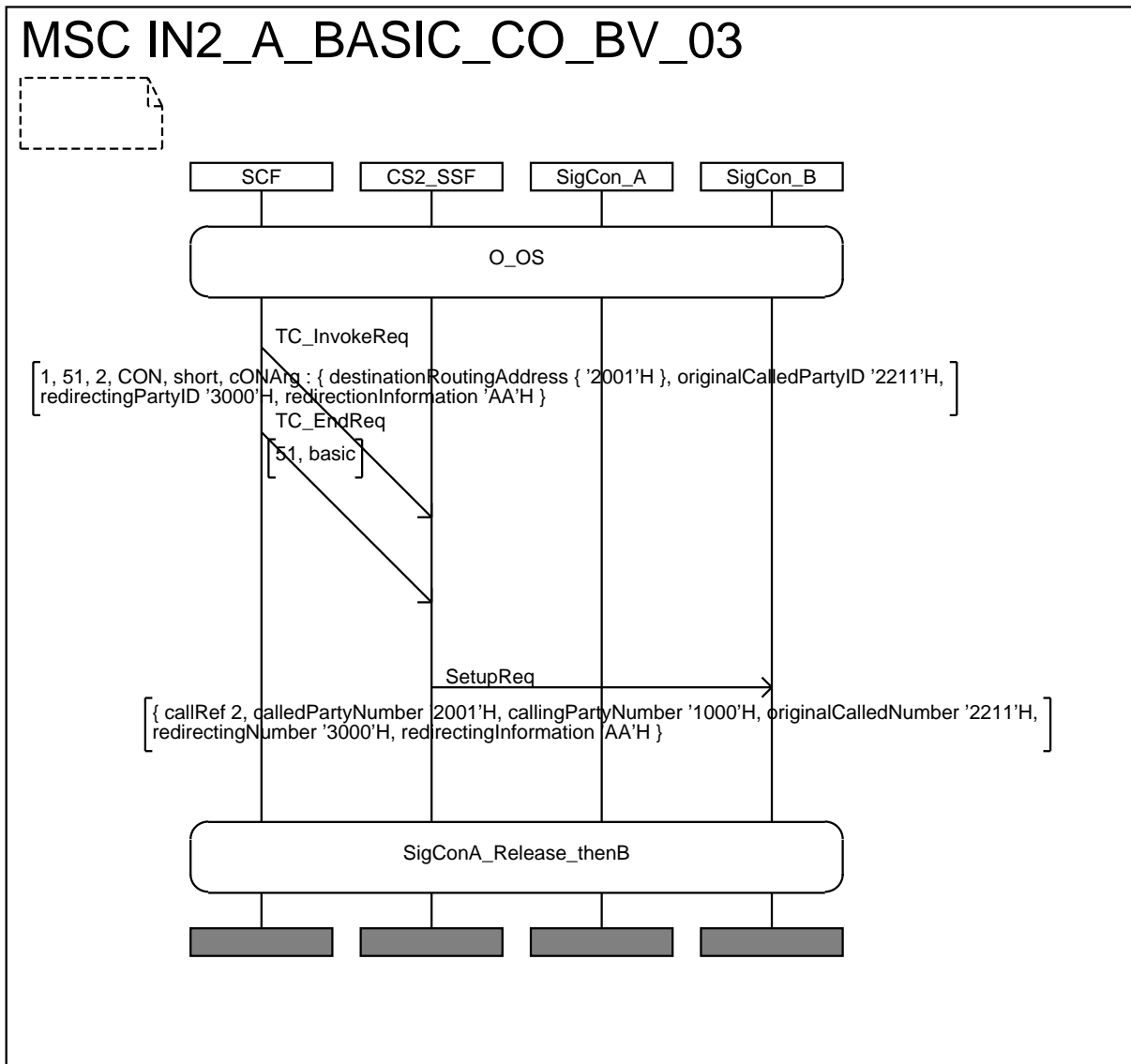


IN2_A_BASIC_CO_BV_02	
Purpose:	Test of Connect procedure with cutAndPaste parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Connect invoke with mandatory and optional parameters, with destinationRoutingAddress cutAndPaste SSF sends a SetupRequest to B side
Pass criteria	Check that the Connect operation is not rejected
Postamble:	SigConA_Release_thenB

MSC IN2_A_BASIC_CO_BV_02

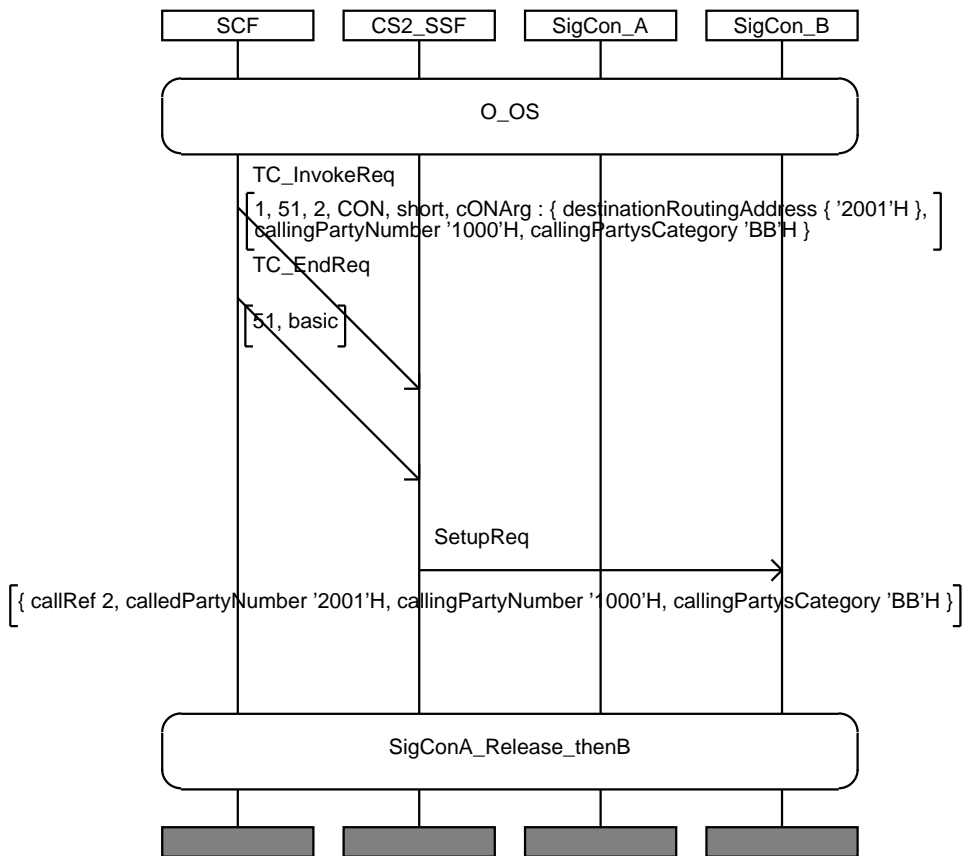


IN2_A_BASIC_CO_BV_03	
Purpose:	Test of Connect procedure with supplementary services parameters
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Connect invoke containing parameters related to supplementary services, with: <ul style="list-style-type: none"> - destinationRoutingAddress, - originalCalledPartyID, - redirectingPartyID, - redirectionInformation SSF sends a SetupRequest to B side
Pass criteria	Check that the above parameters are mapped from Connect into the Set-up request: <ul style="list-style-type: none"> - destinationRoutingAddress -----> calledPartyNumber - originalCalledPartyID-----> originalCalledNumber - redirectingPartyID-----> redirectingNumber - redirectionInformation-----> redirectionInformation
Postamble:	SigConA_Release_thenB



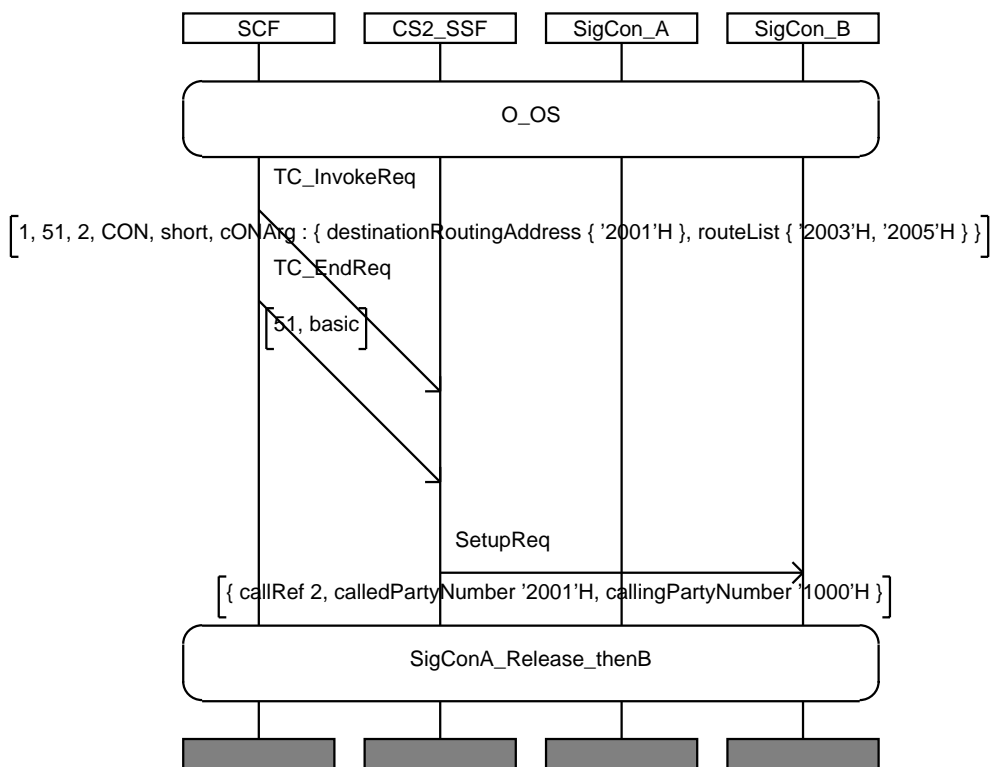
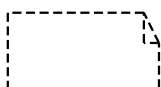
IN2_A_BASIC_CO_BV_04	
Purpose:	Test of Connect procedure with optional parameters related to the calling party
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Connect invoke containing mandatory and optional parameters - destinationRoutingAddress, - callingPartyNumber, - callingPartysCategory SSF sends a SetupRequest to B side
Pass criteria	Check that the above parameters are mapped from Connect into the Set-up request: - destinationRoutingAddress-----> calledPartyNumber - callingPartyNumber-----> callingPartyNumber - callingPartysCategory-----> callingPartysCategory
Postamble:	SigConA_Release_thenB

MSC IN2_A_BASIC_CO_BV_04



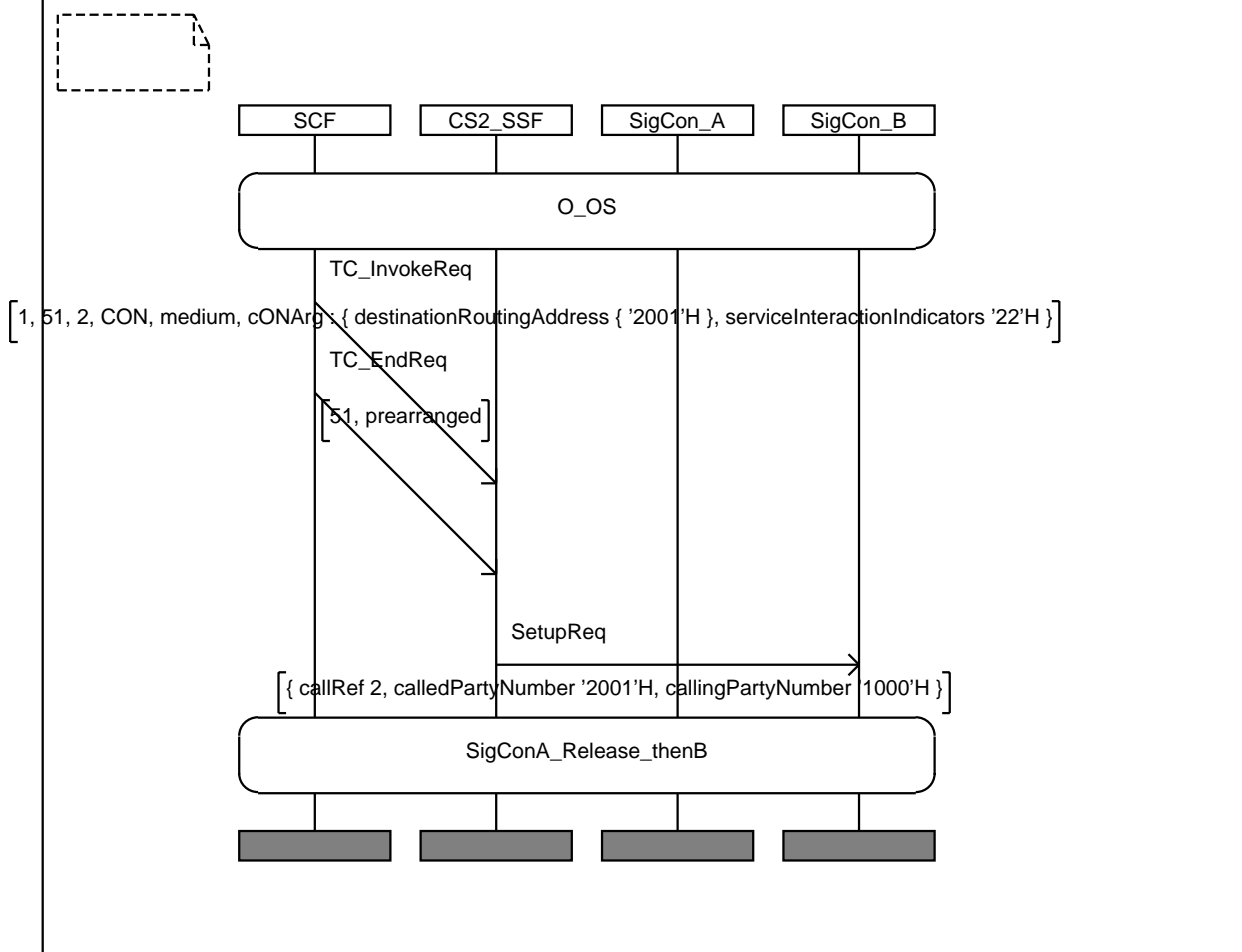
XXXX		IN2_A_BASIC_CO_BV_05	
Purpose:	Test of Connect procedure with optional parameter routeList		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF sends to SSF Connect invoke containing mandatory and optional parameters - destinationRoutingAddress, - routeList (with two different routes) SSF sends a SetupRequest to B side B Side sends a RelInd with release cause being routeSelectFailure		
Pass criteria	Check that if the first route fails the IUT sends a second SetUpReq to B side using the second route.		
Postamble:	SigConA_Release_thenB		

MSC IN2_A_BASIC_CO_BV_05



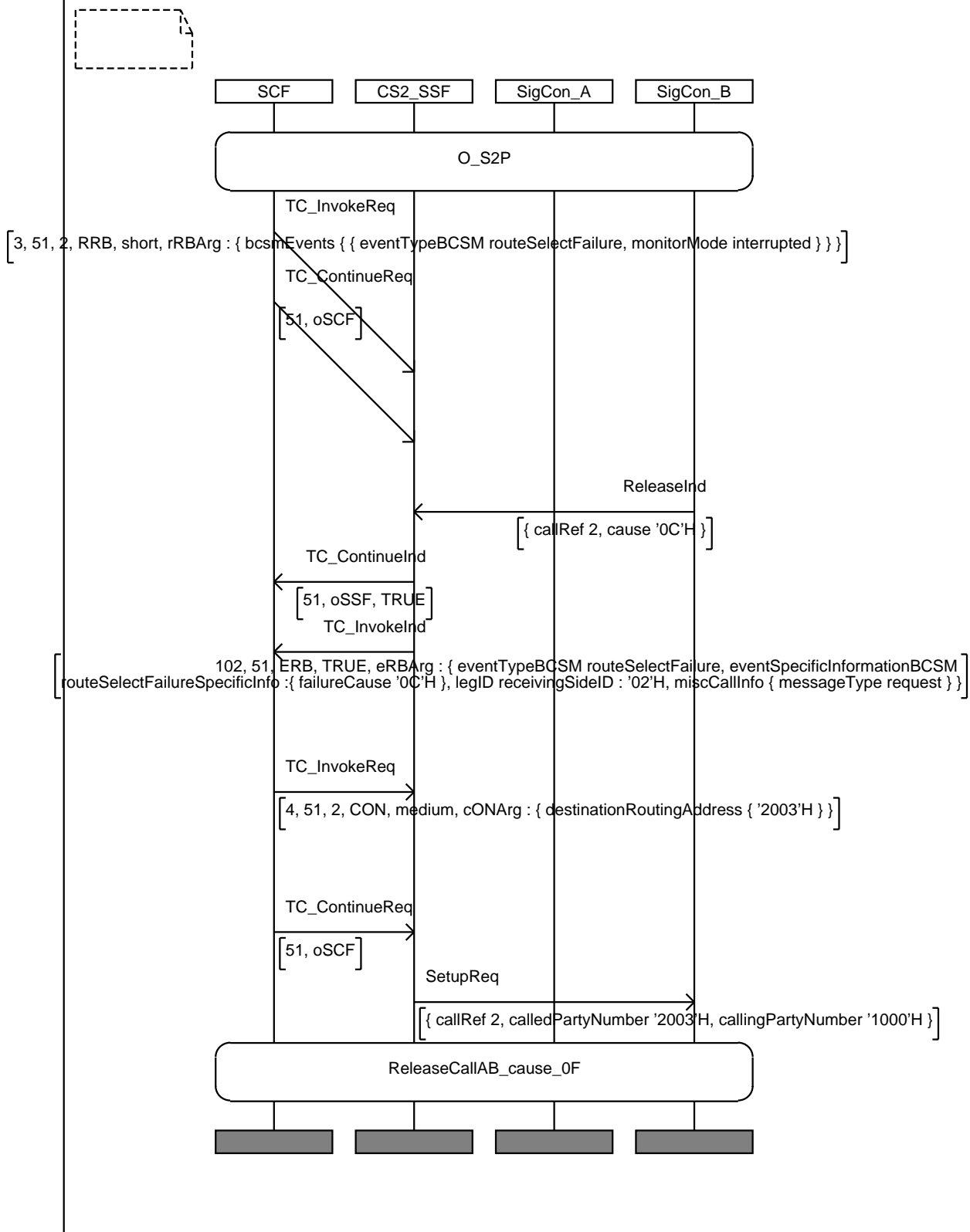
XXXX		IN2_A_BASIC_CO_BV_06	
Purpose:	Test of Connect procedure with optional parameter serviceInteractionIndicators		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF sends to SSF Connect invoke containing mandatory and optional parameters - destinationRoutingAddress, - serviceInteractionIndicators SSF sends a SetupRequest to B side		
Pass criteria	Check that the above parameters are mapped from Connect into the Set-up request: - destinationRoutingAddress-----> calledPartyNumber, - serviceInteractionIndicators-----> serviceIndicators		
Postamble:	SigConA_Release_thenB		

MSC IN2_A_BASIC_CO_BV_06



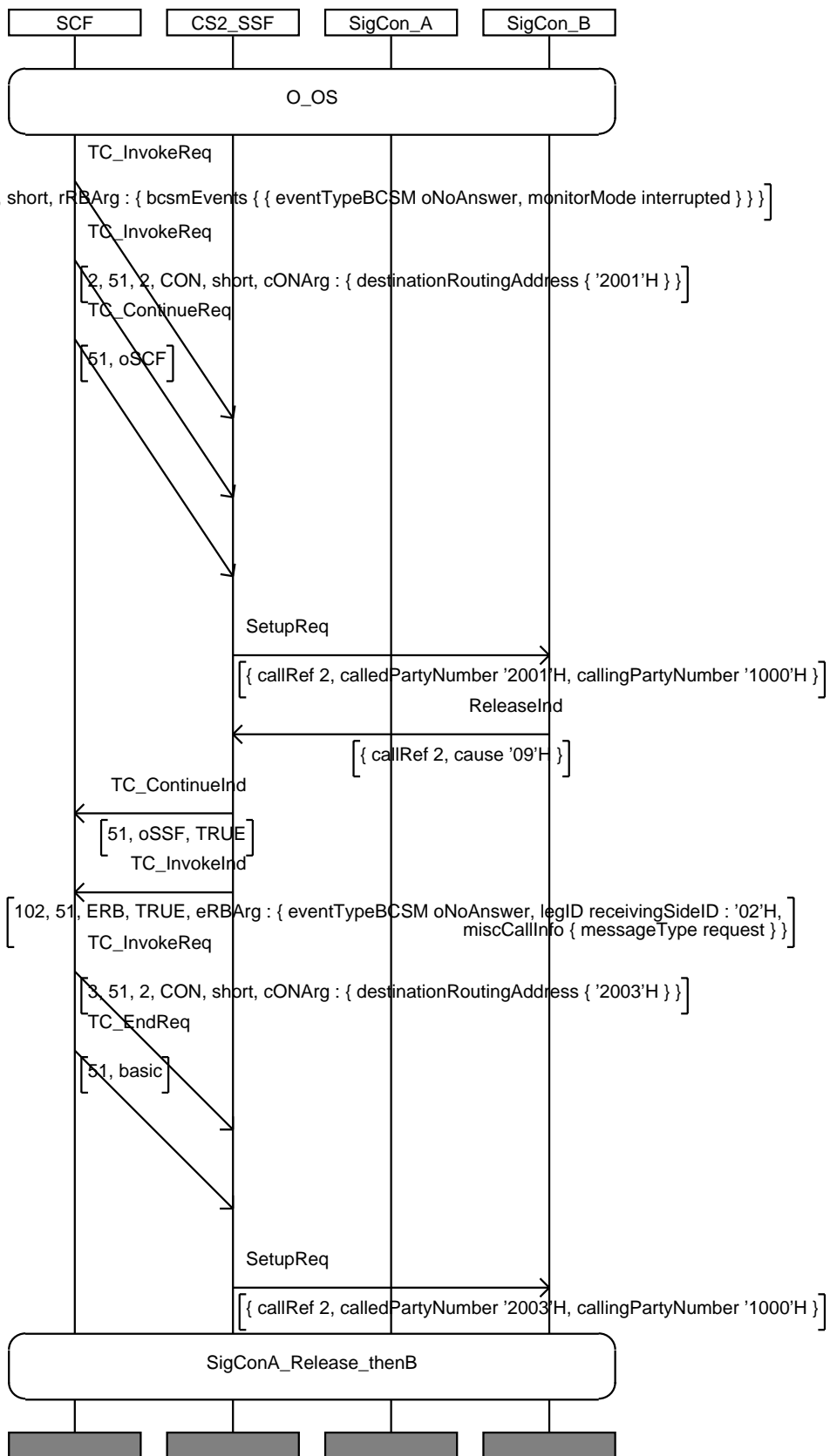
XXXX		IN2_A_BASIC_CO_BV_07	
Purpose:	Test of Connect procedure received on each allowed DP of O_BCSM		
Requirement ref			
Selection Cond.			
Preamble:	O_S2P		
Test description	SCF sends a RRB operation to arm routeSelectFailure event SigConB sends a ReleaseInd with release cause being routeSelectFailure SCF sends to SSF Connect invoke containing mandatory parameter - destinationRoutingAddress,		
Pass criteria	Check that SSF sends a SetupReq to B side		
Postamble:	ReleaseCallAB_cause_0F		

MSC IN2_A_BASIC_CO_BV_07



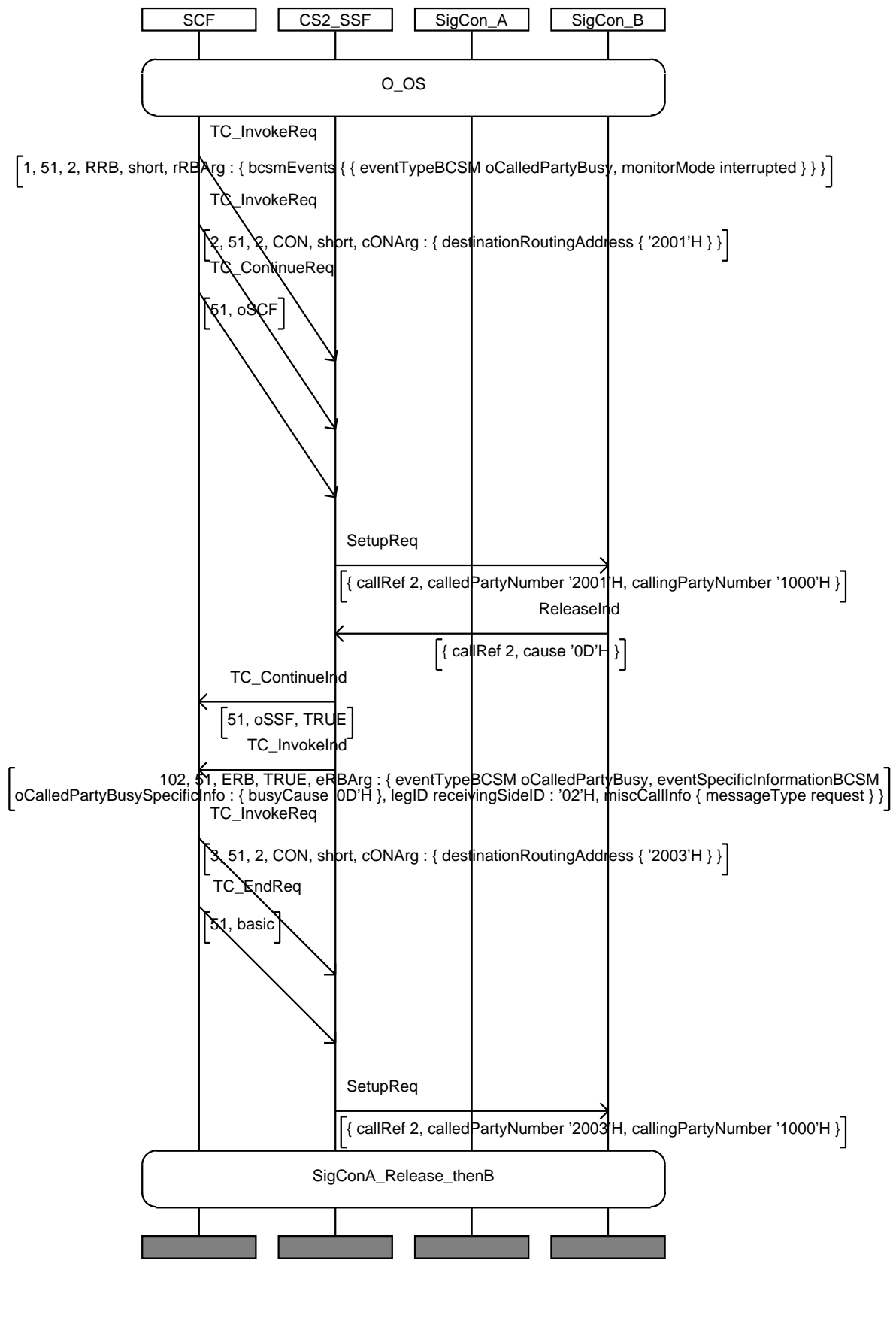
XXXX		IN2_A_BASIC_CO_BV_08	
Purpose:		Test of Connect procedure in response to oNoAnswer DP.	
Requirement ref			
Selection Cond.			
Preamble:		O_OS	
Test description		SCF sends a RRB operation to arm oNoAnswer event SCF sends a Connect operation with mandatory parameters SigConB sends a ReleaseInd with release cause being bPtyNoAnswer Scs SCF sends to SSF Connect invoke containing mandatory parameter - destinationRoutingAddress,	
Pass criteria		Check that SSF sends a SetupReq to B side	
Postamble:		SigConA_Release_thenB	

MSC IN2_A_BASIC_CO_BV_08



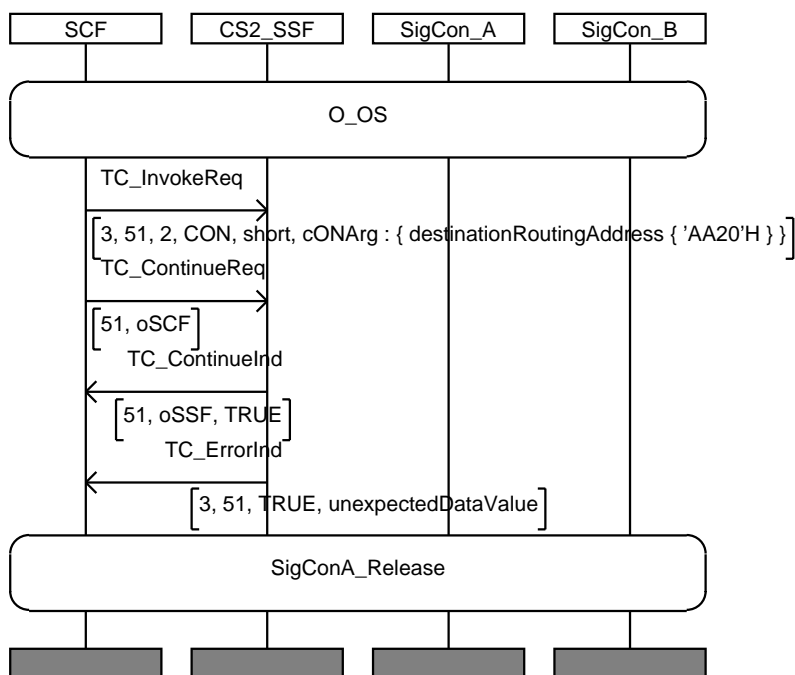
XXXX		IN2_A_BASIC_CO_BV_09	
Purpose:	Test of Connect procedure in response to oCalledPartyBusy DP.		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF SCF sends a RRB operation to arm oCalledPartyBusy event SC SCF sends a Connect operation with mandatory parameters SigConB sends a ReleaseInd with release cause being bPtyBusy_UDUB SCF sends to SSF Connect invoke containing mandatory parameter - destinationRoutingAddress.		
Pass criteria	Check that SSF sends a SetupReq to B side		
Postamble:	SigConA_Release_thenB		

MSC IN2_A_BASIC_CO_BV_09



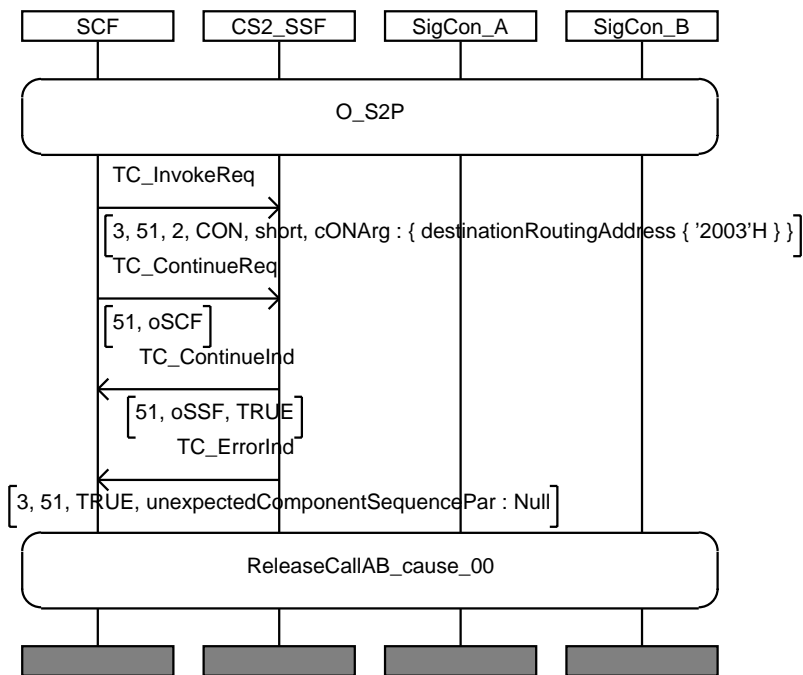
IN2_A_BASIC_CO_BI_01	
Purpose:	Test of Connect procedure with invalid parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Connect invoke containing mandatory parameters with an invalid value in - destinationRoutingAddress,
Pass criteria	Check that SSF sends back Connect error, with error parameter UnexpectedDataValue
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_CO_BI_01



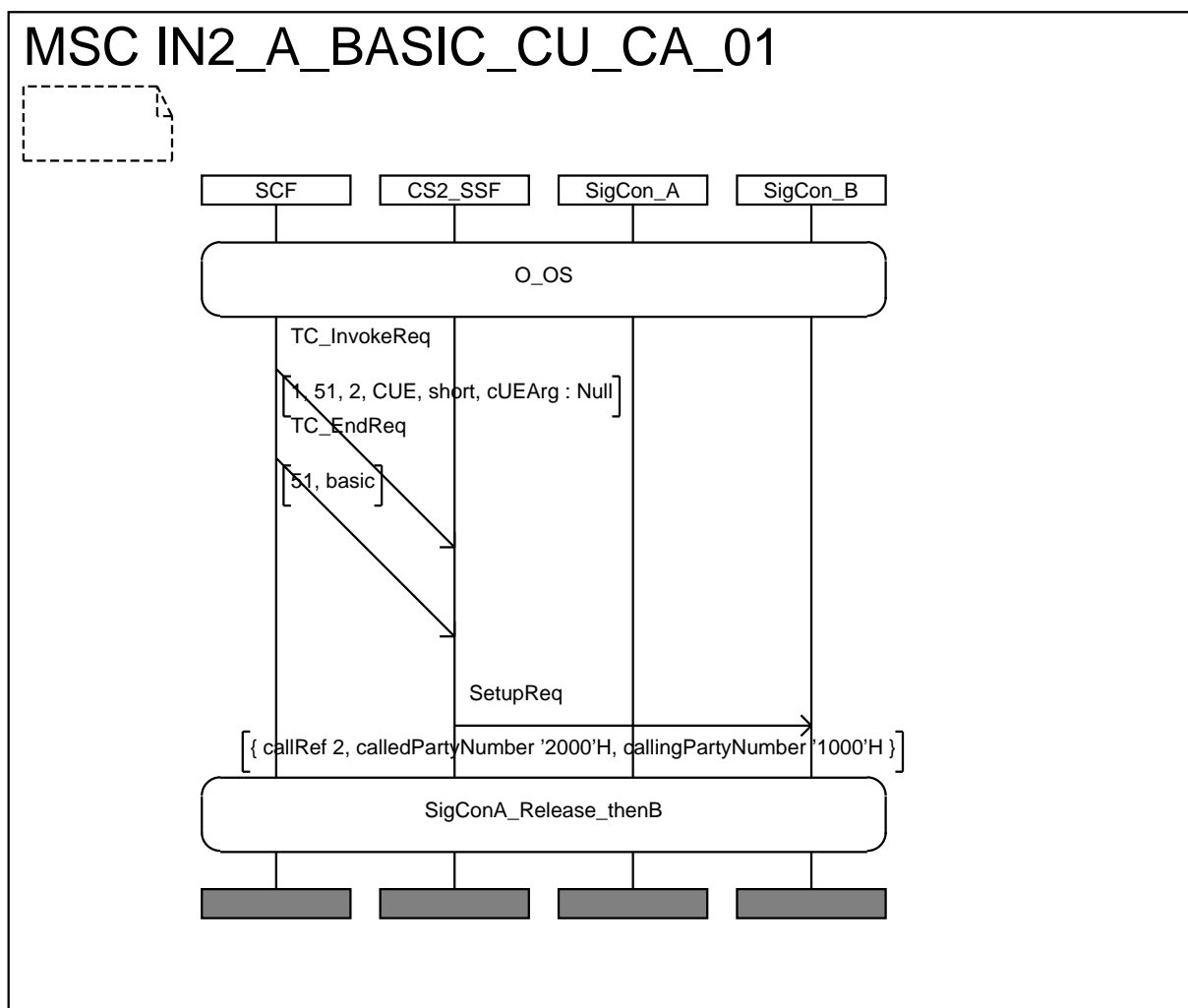
IN2_A_BASIC_CO_BO_01	
Purpose:	Test of Connect procedure initiated from wrong state
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	SCF sends to SSF Connect invoke containing mandatory parameters, while in monitoring state
Pass criteria	Check that SSF sends back Connect error, with error parameter UnexpectedComponentSequence
Postamble:	ReleaseCallAB_cause_00

MSC IN2m_A_BASIC_CO_BO_01

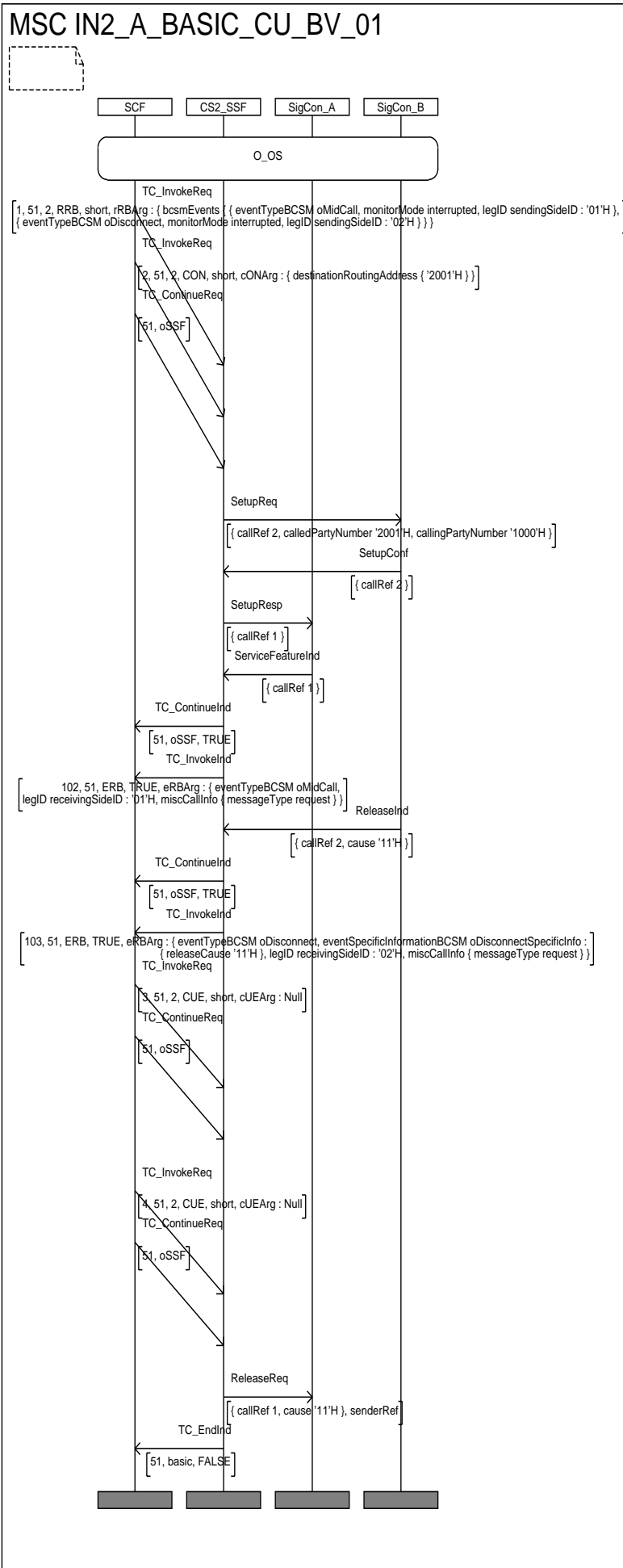


6.4.9 Continue procedure

IN2_A_BASIC_CU_CA_01	
Purpose:	Test of Continue procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF Continue invoke
Pass criteria	Check that SSF continues call processing, i.e. SetupReq is detected at SigConB
Postamble:	SigConA_Release_thenB



IN2_A_BASIC_CU_BV_01	
Purpose:	Test of Continue procedure when 2 outstanding EDP-Rs are reported
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<ol style="list-style-type: none"> 1. SCF sends to SSF RequestReportBCSMEEvent invoke with parameters (oMidCall, interrupted, leg1) and (oDisconnect, interrupted, leg2) 2. SCF sends to SSF Connect invoke with a called party number to establish a call to the B-party. 3. After SetupReq is detected at SigConB, the call is answered by SetupConf at SigConB 4. A party issues ServiceFeatureInd at SigConA 5. SSF sends to SCF EventReportBCSMEEvent invoke with parameter (oMidCall, leg1) 6. B party issues ReleaseInd at SigConB 7. SSF sends to SCF EventReportBCSMEEvent invoke with parameter (oDisconnect, leg2) 8. SCF sends to SSF Continue invoke (note that the release should <u>not</u> be continued) 9. SCF sends to SSF a 2nd Continue invoke
Pass criteria	Check that SSF sends ReleaseReq to SigConA
Postamble:	none



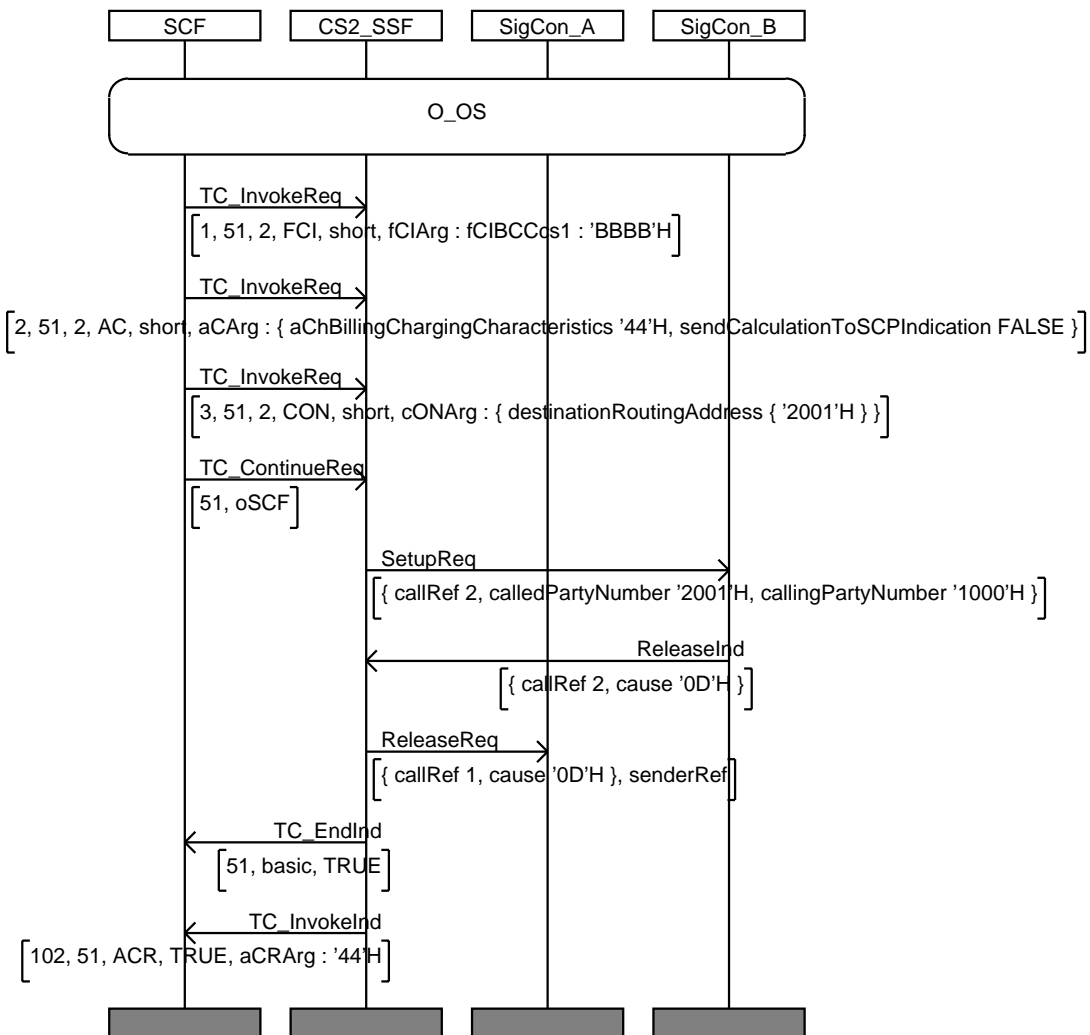
6.4.10 FurnishChargingInformation procedure

Charging related aspects in IN are network operator specific. Therefore, it is not possible to define useful test purposes for charging procedures using a network operator independant approach. TP specification has to be done by network operators, using INAP procedures themselves. TP could be specified by combining ApplyCharging, FurnishChargingInformation and SendChargingInformation procedures.

Following some examples:

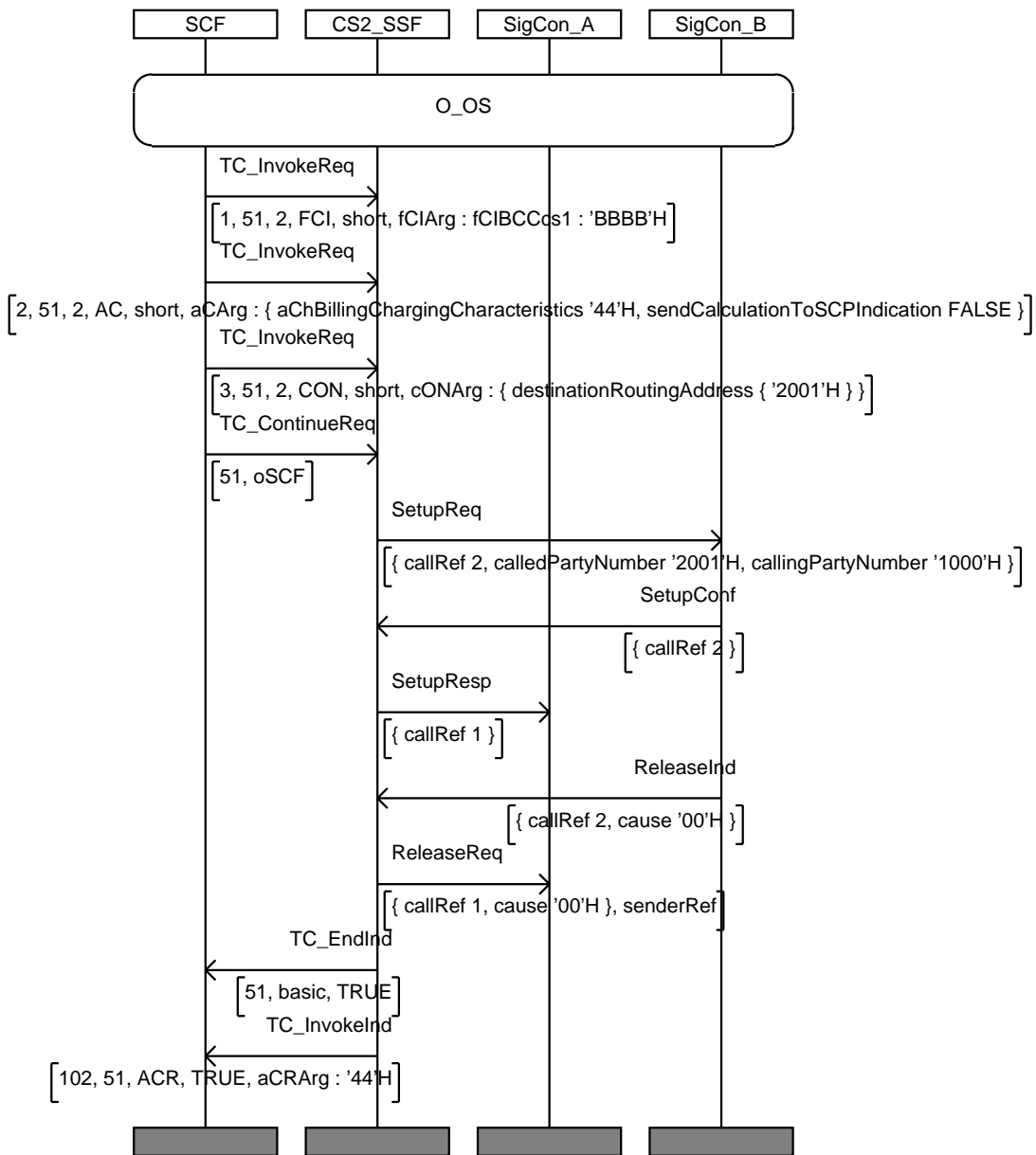
IN2_A_BASIC_FC_CA_01	
Purpose:	Test of FurnishChargingInformation procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF FurnishChargingInformation invoke containing parameter - FCIBillingChargingCharacteristics (with completeChargingrecord) followed by ApplyCharging with mandatory parameters Then a call is established and remains for a given time to obtain a charging record
Pass criteria	Check that upon release of the call, SSF sends to SCF an ApplyChargingReport invoke and that the call is released
Postamble:	none

MSC IN2m_A_BASIC_FC_CA_01



IN2_A_BASIC_FC_BV_01	
Purpose:	Test of FurnishChargingInformation procedure
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	<p>SCF sends to SSF FurnishChargingInformation invoke containing parameter</p> <p><input type="checkbox"/> FCIBillingChargingCharacteristics (with completeChargingrecord)</p> <p>followed by ApplyCharging with mandatory parameters</p> <p>Then a call is established and remains for a given time to obtain a charging record</p>
Pass criteria	Check that upon release of the call, SSF sends to SCF an ApplyChargingReport invoke and that the call is released
Postamble:	none

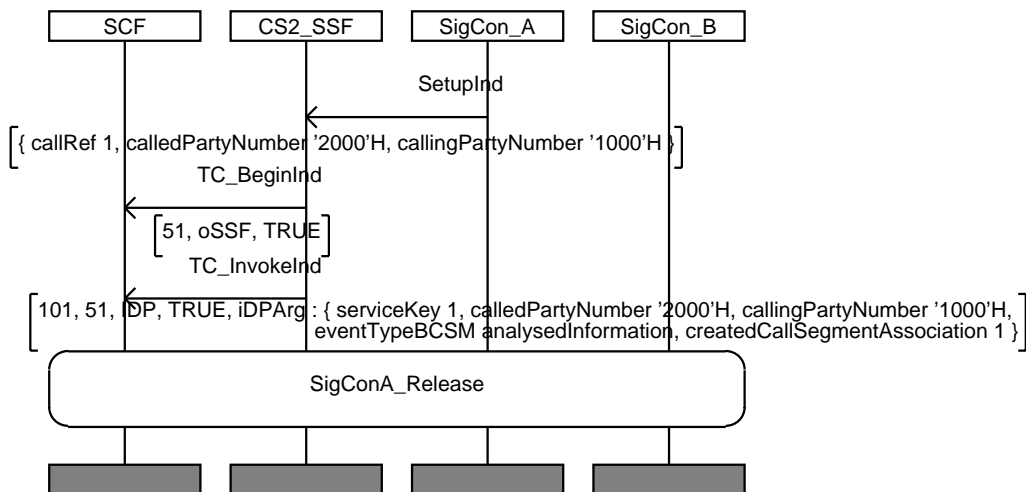
MSC IN2m_A_BASIC_FC_BV_01



6.4.11 InitialDP procedure

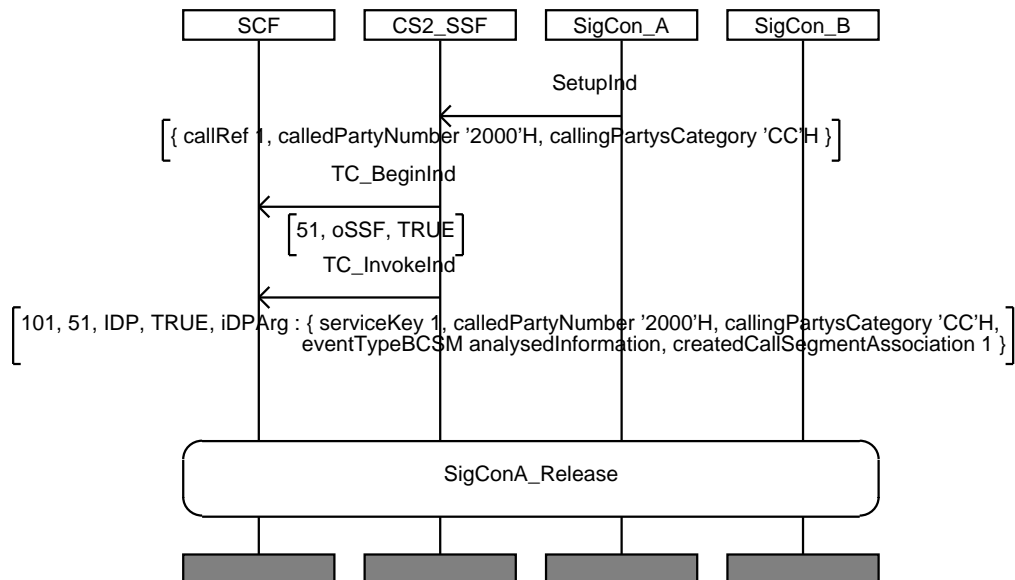
IN2_A_BASIC_DP_CA_01	
Purpose:	Test of InitialDP procedure and its parameter calledPartyNumber
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SigConA sends to SSF a SetupInd containing at least the parameter: - calledPartyNumber
Pass criteria	Check that SSF sends to SCF an InitialDP invoke containing the parameter related to the called party: - calledPartyNumber
Postamble:	SigConA_Release

MSC IN2_A_BASIC_DP_CA_01



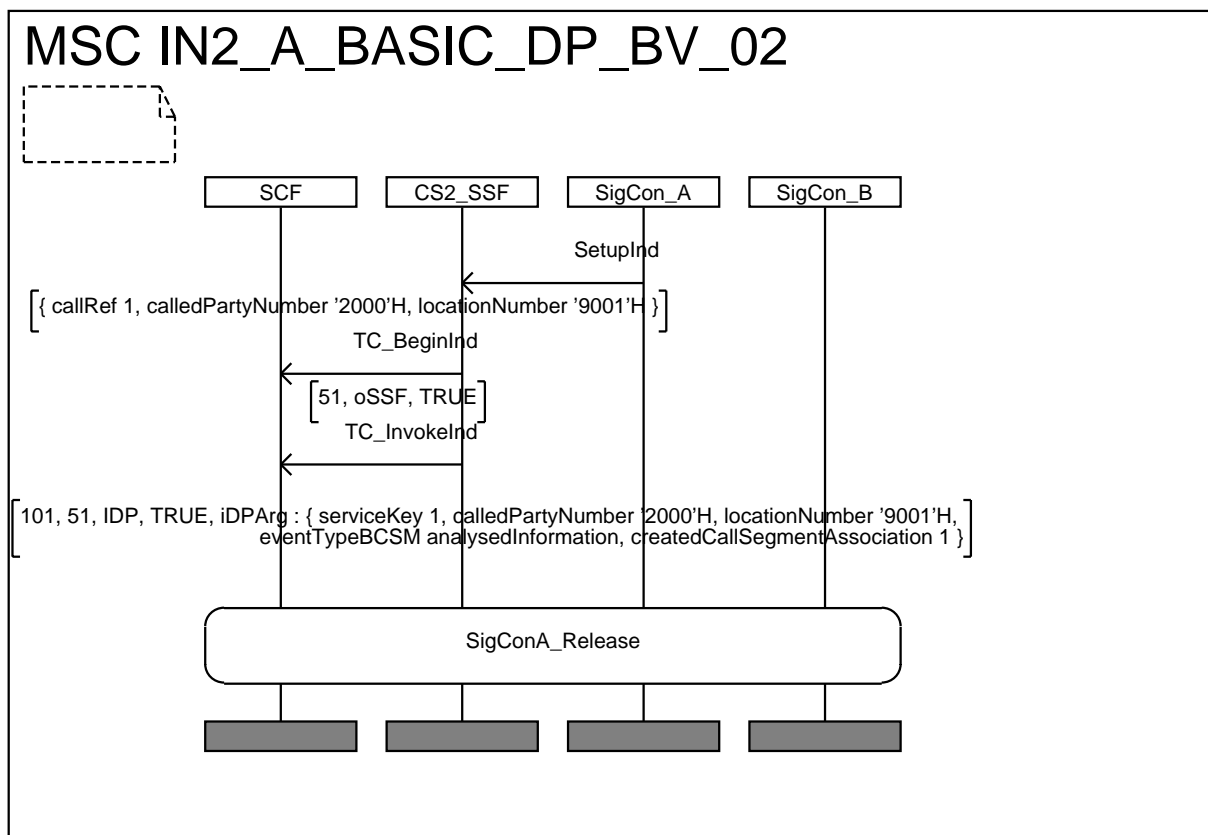
IN2_A_BASIC_DP_BV_01	
Purpose:	Test of InitialDP procedure and its parameter callingPartysCategory
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SigConA sends to SSF a SetupInd containing at least the parameter: - callingPartysCategory
Pass criteria	Check that SSF sends to SCF an InitialDP invoke containing the parameter related to the calling party category: - callingPartysCategory
Postamble:	SigConA_Release

MSC IN2_A_BASIC_DP_BV_01



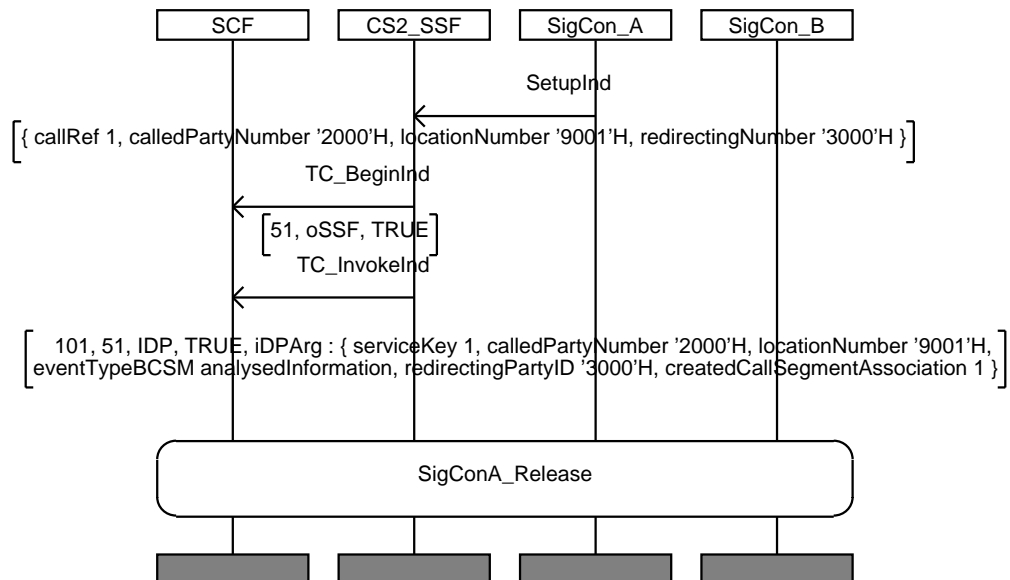
IN2_A_BASIC_DP_BV_02	
Purpose:	Test of InitialDP procedure and its parameter locationNumber
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SigConA sends to SSF a SetupInd containing at least the parameter: - locationNumber
Pass criteria	Check that SSF sends to SCF an InitialDP invoke containing the parameter related to the location information: - locationNumber
Postamble:	SigConA_Release

MSC IN2_A_BASIC_DP_BV_02

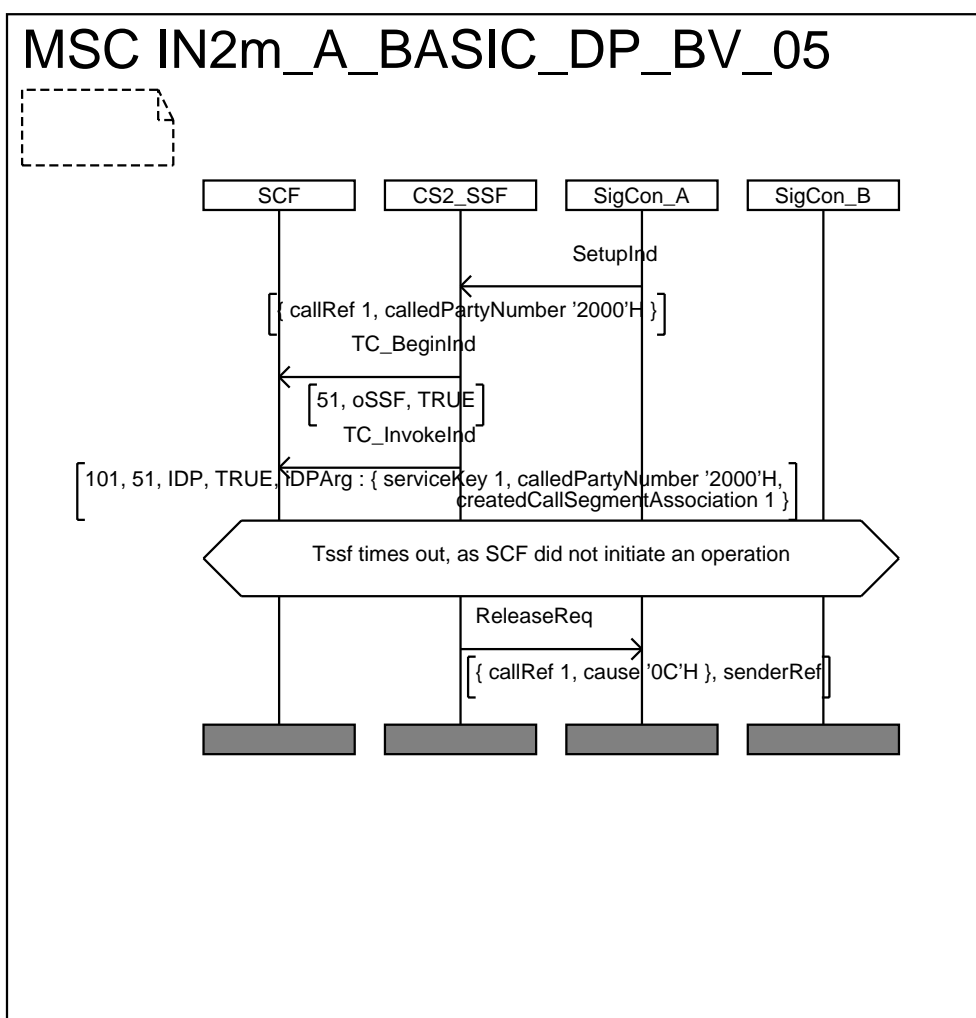


IN2_A_BASIC_DP_BV_04	
Purpose:	Test of InitialDP procedure and its parameter redirectingPartyID
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SigConA sends to SSF a SetupInd containing at least the parameter: - redirectingPartyID
Pass criteria	Check that SSF sends to SCF an InitialDP invoke containing the parameter related to redirecting party number: - redirectingPartyID
Postamble:	SigConA_Release

MSC IN2_A_BASIC_DP_BV_04

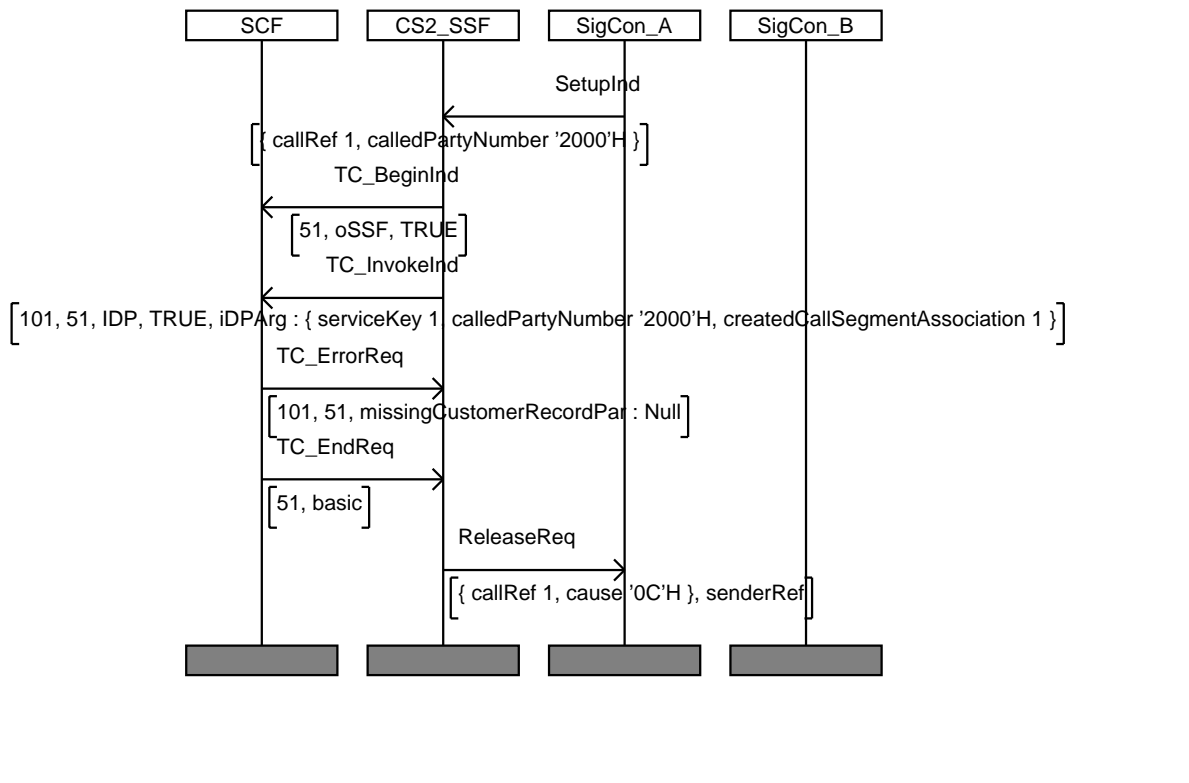


XXXX	IN2_A_BASIC_DP_BV_05
Purpose:	Test of InitialDP procedure with timer expiration
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<ul style="list-style-type: none"> - SigConA sends a SetupInd containing at least the mandatory parameters - SSF sends to SCF an InitialDP invoke - SCF does not send to SSF an operation, so timer Tssf expires
Pass criteria	SSF sends release to SigConA to free all resources involved
Postamble:	



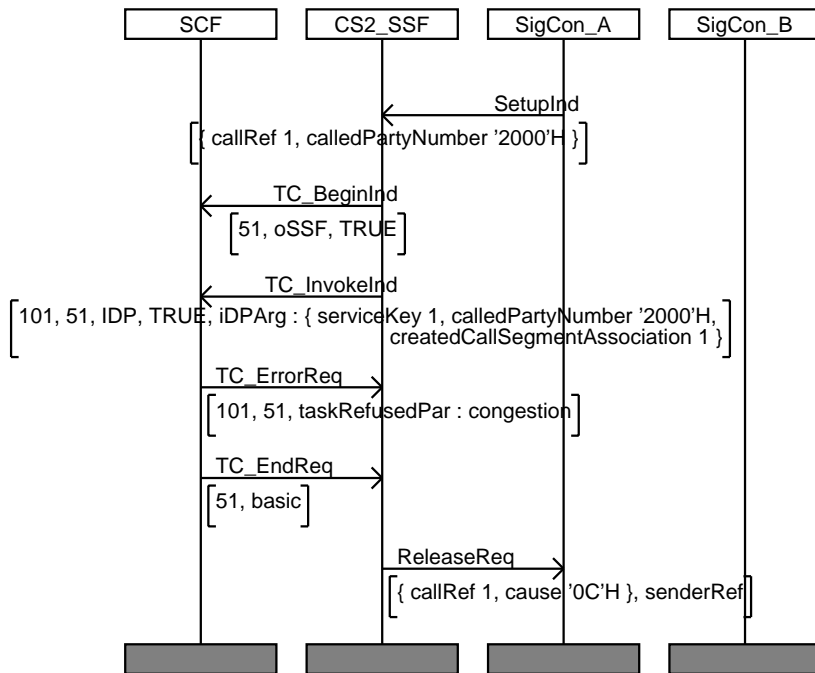
IN2_A_BASIC_DP_BI_01	
Purpose:	Test of InitialDP procedure with error
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<ul style="list-style-type: none"> - SigConA sends a SetupInd containing at least the mandatory parameters - SSF sends to SCF an InitialDP invoke - SCF sends to SSF an InitialDP error containing parameter: missingCustomerRecord
Pass criteria	SSF goes to the idle state.
Postamble:	none

MSC IN2m_A_BASIC_DP_BI_01



IN2_A_BASIC_DP_BI_02	
Purpose:	Test of InitialDP procedure with error
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	<ul style="list-style-type: none"> - SigConA sends a SetupInd containing at least the mandatory parameters - SSF sends to SCF an InitialDP invoke - SCF sends to SSF an InitialDP error containing parameter: taskRefused
Pass criteria	SSF goes to the idle state.
Postamble:	none

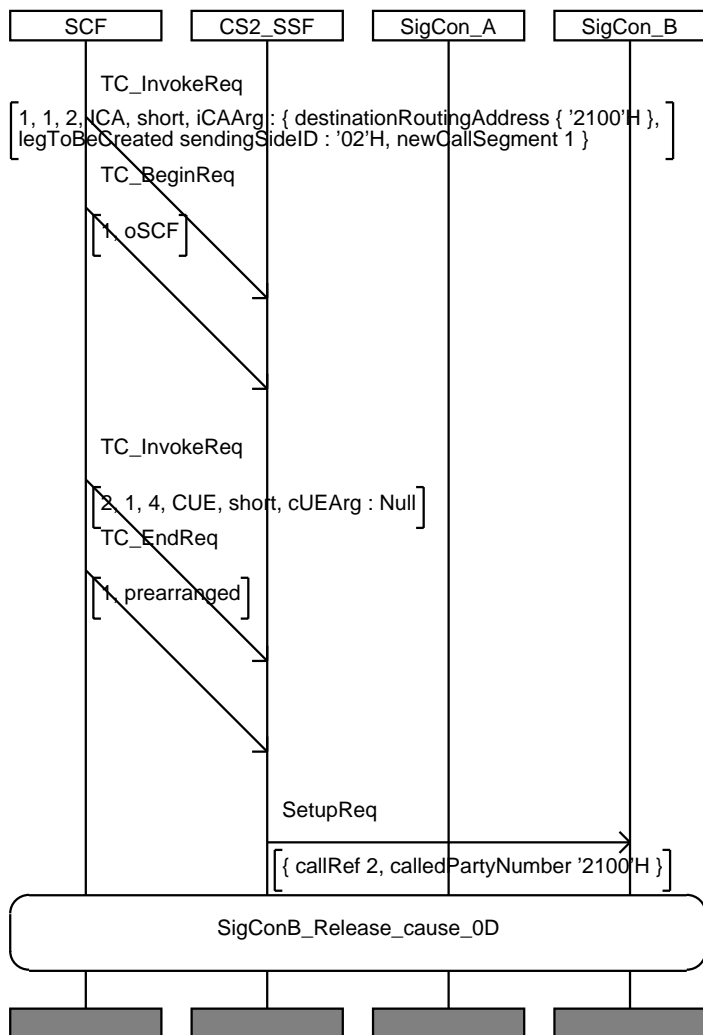
MSC IN2m_A_BASIC_DP_BI_02



6.4.12 InitiateCallAttempt procedure

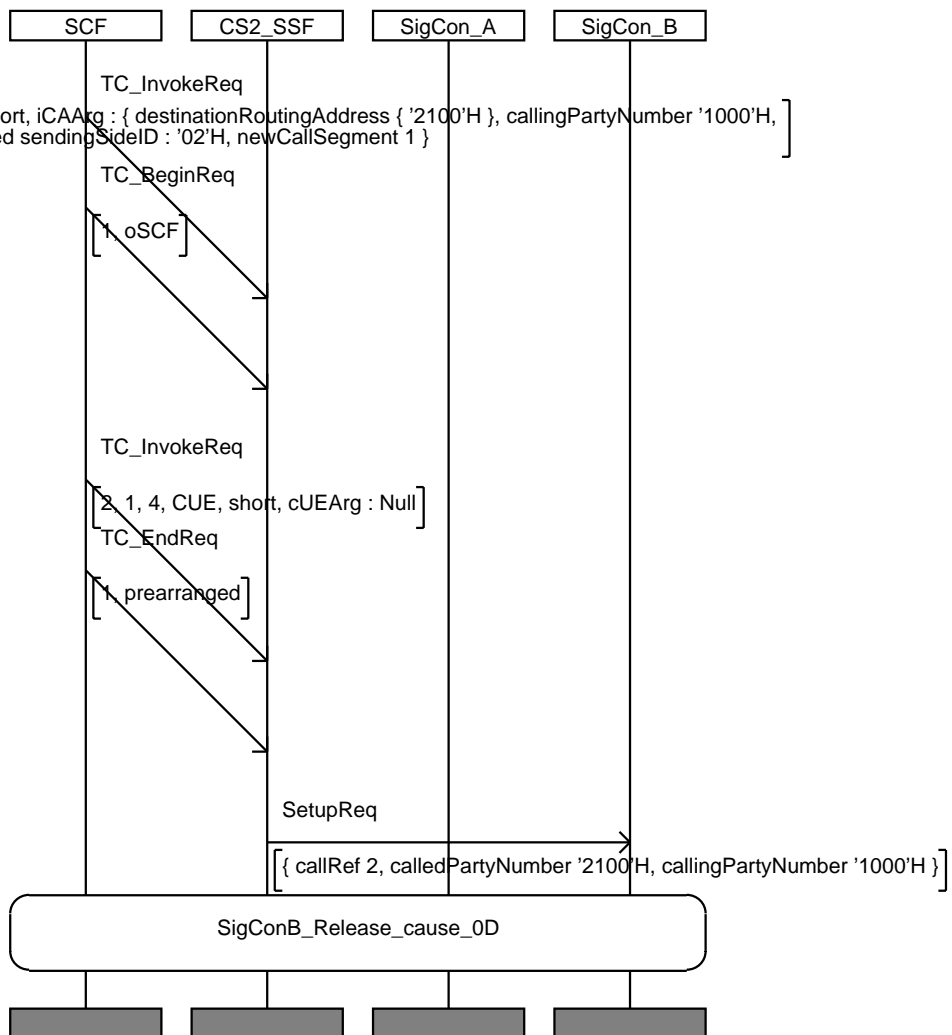
IN2_A_BASIC_IC_CA_01	
Purpose:	Test of InitiateCallAttempt base procedure
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF an InitiateCallAttempt with mandatory parameters: destinationRoutingAddress followed by a Continue invoke
Pass criteria	Check that SSF sends a SetupReq to the proper SigCon according to the InitiateCallAttempt
Postamble:	SigConB_Release

MSC IN2_A_BASIC_IC_CA_01



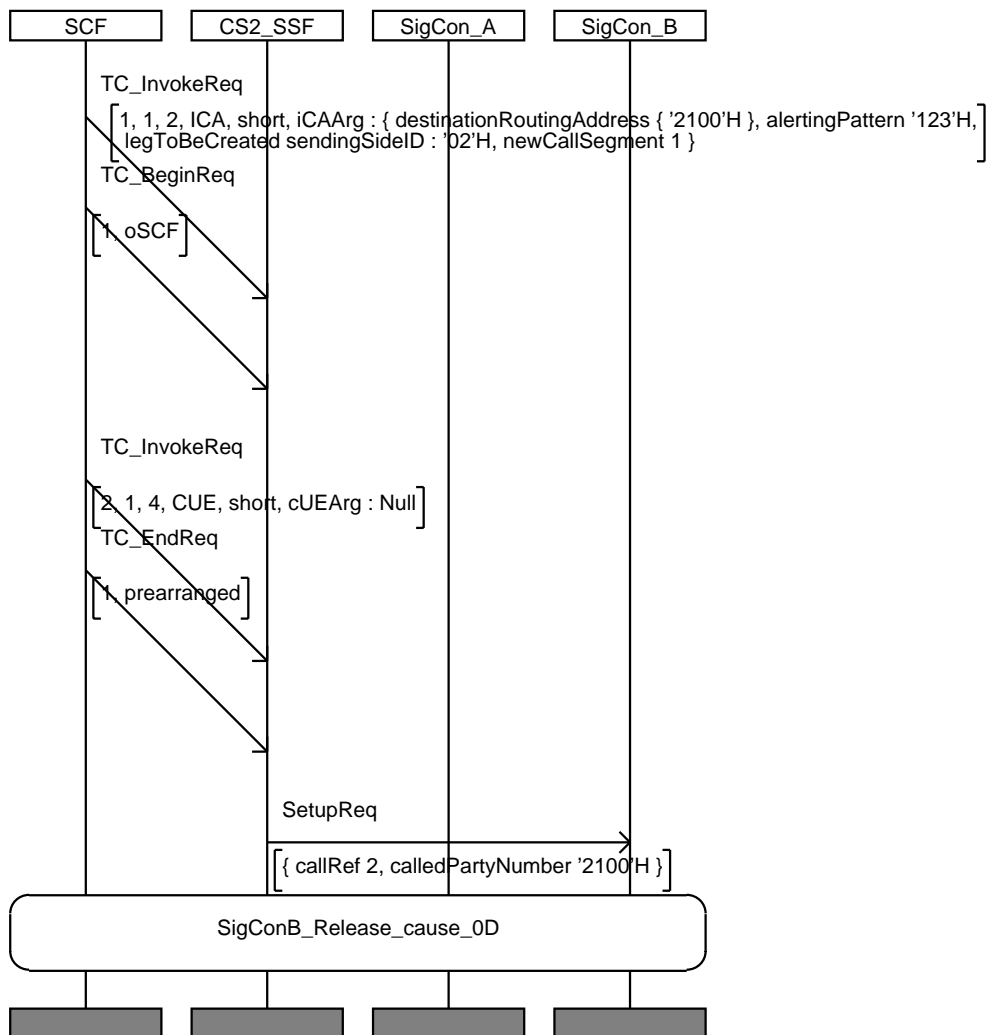
IN2_A_BASIC_IC_BV_01	
Purpose:	Test of InitiateCallAttempt procedure with parameter
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF an InitiateCallAttempt with mandatory and optional parameters destinationRoutingAddress callingPartyNumber followed by a Continue invoke
Pass criteria	Check that SSF sends a SetupReq to the proper SigCon according to the InitiateCallAttempt
Postamble:	SigConB_Release

MSC IN2_A_BASIC_IC_BV_01



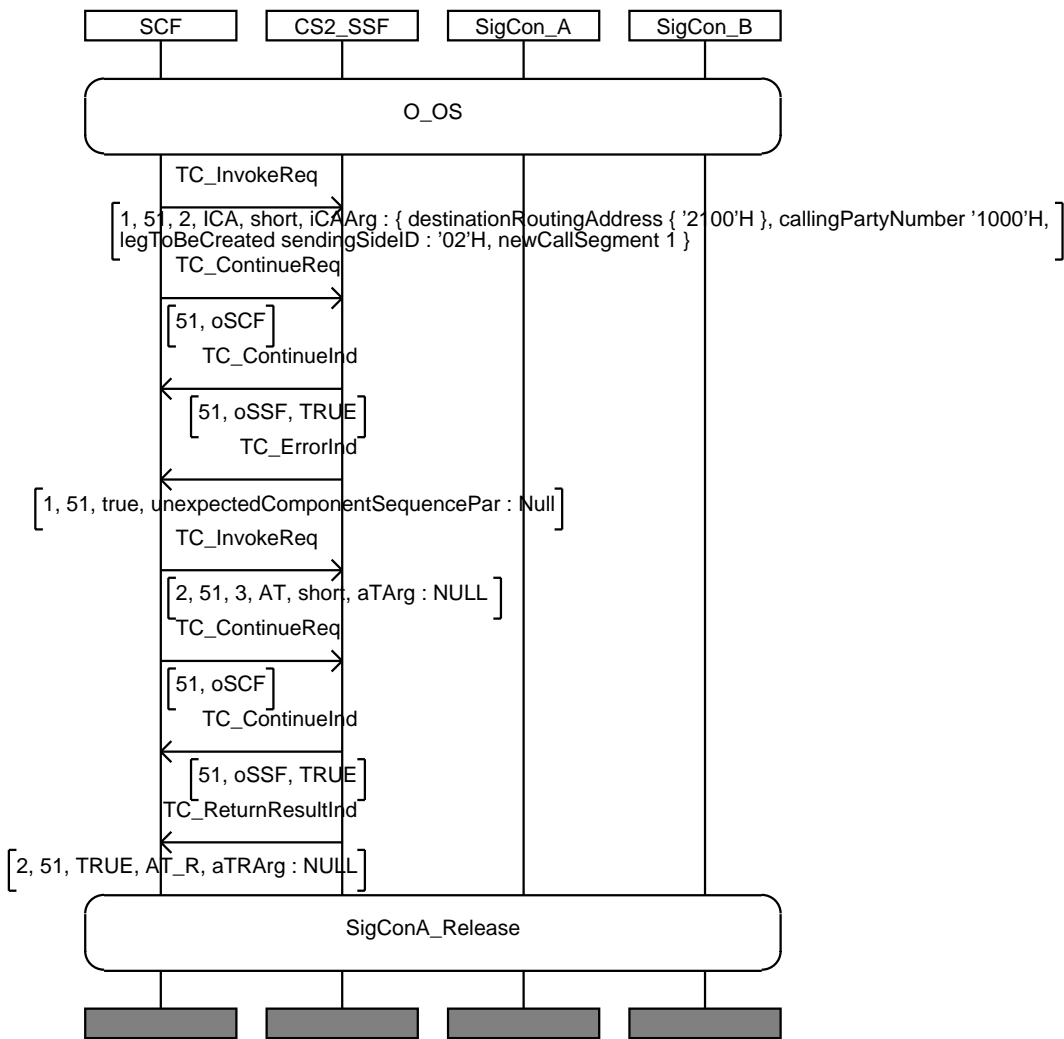
IN2_A_BASIC_IC_BV_02	
Purpose:	Test of InitiateCallAttempt procedure with parameter
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF an InitiateCallAttempt with mandatory and optional parameters destinationRoutingAddress alertingPattern followed by a Continue invoke
Pass criteria	Check that SSF sends a SetupReq to the proper SigCon according to the InitiateCallAttempt and check the special tone indicated in alertingPattern
Postamble:	SigConB_Release

MSC IN2_A_BASIC_IC_BV_02



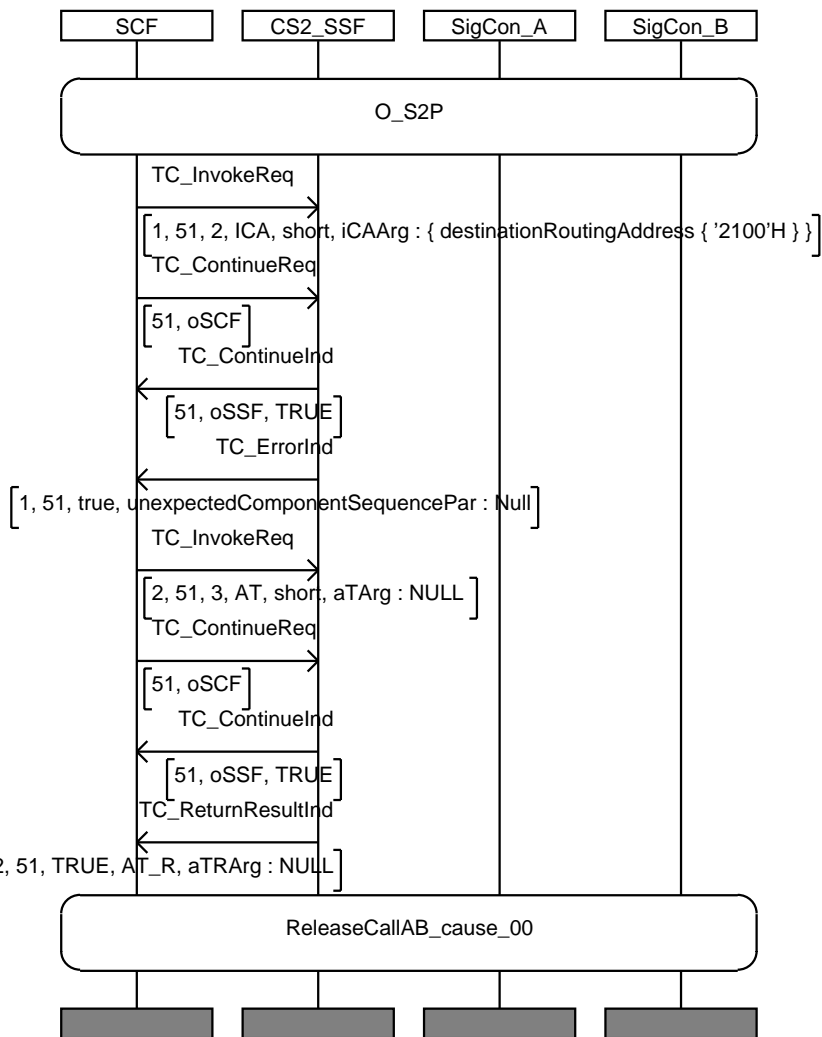
IN2_A_BASIC_IC_BO_01	
Purpose:	Test of InitiateCallAttempt procedure in wrong state (WaitForInstructions)
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF an InitiateCallAttempt in a wrong state (WFI)
Pass criteria	Check that SSF returns InitiateCallAttempt error with parameter UnexpectedComponentSequence and remains in the same state. To check SSF stays in WaitingForInstructions state, - SCF sends to SSF an ActivityTest invoke with DialogID used in previous InitiateCallAttempt - SSF answers with ActivityTest result because DialogID is still active
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_IC_BO_01



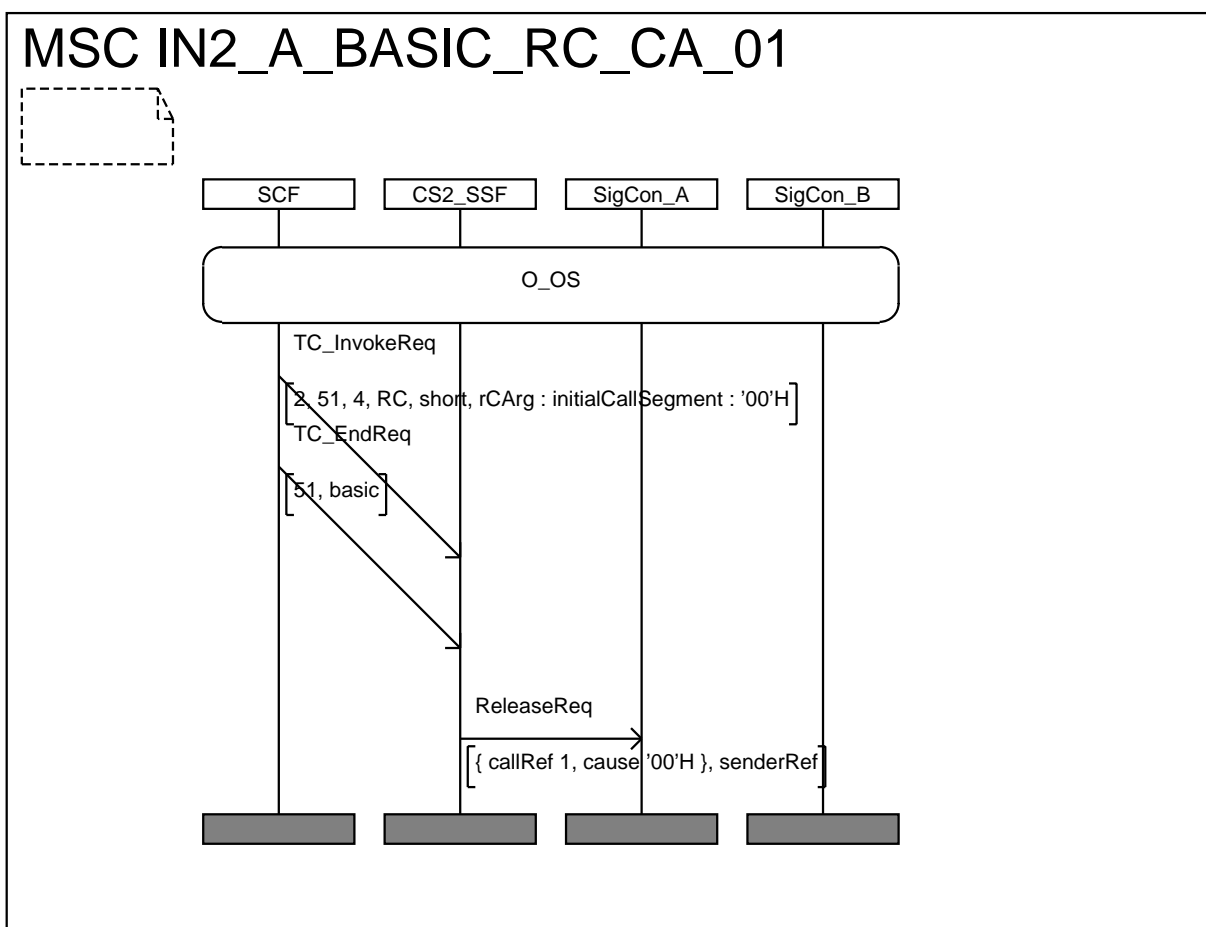
XXXX		IN2_A_BASIC_IC_BO_02	
Purpose:	Test of InitiateCallAttempt procedure in wrong state (monitoring)		
Requirement ref			
Selection Cond.			
Preamble:	O_S2P		
Test description	SCF sends to SSF an InitiateCallAttempt in a wrong state (Monitoring)		
Pass criteria	<p>Check that SSF returns InitiateCallAttempt error with parameter UnexpectedComponentSequence and remains in the same state.</p> <p>To check SSF stays in monitoring state,</p> <ul style="list-style-type: none"> - SCF sends to SSF an ActivityTest invoke with DialogID used in previous InitiateCallAttempt - SSF answers with ActivityTest result because DialogID is still active 		
Postamble:	ReleaseCallAB_cause_00		

MSC IN2m_A_BASIC_IC_BO_02



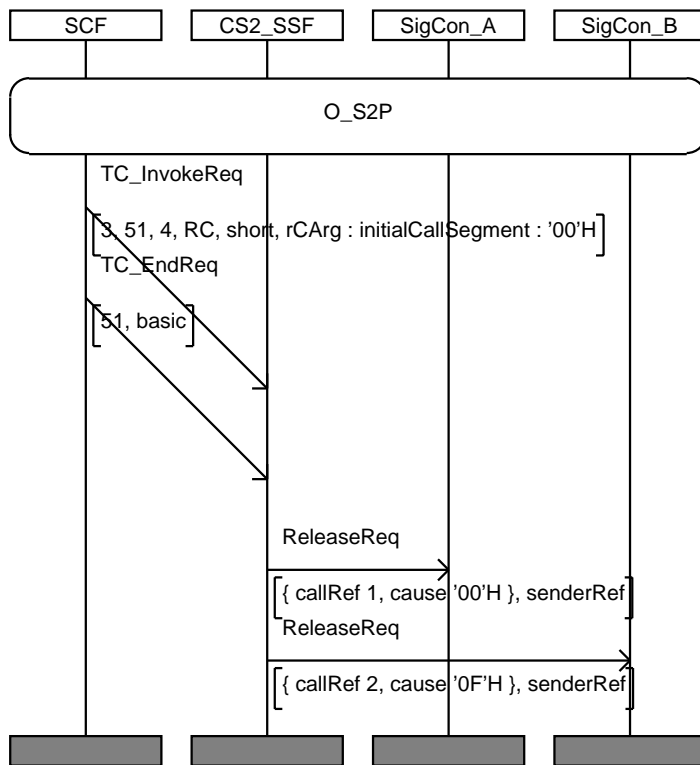
6.4.13 ReleaseCall procedure

IN2_A_BASIC_RC_CA_01	
Purpose:	Test of ReleaseCall base procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF ReleaseCall invoke, with: - initialCallSegment (cause)
Pass criteria	Check that SSF releases the call (ReleaseReq received by SigConA)
Postamble:	none



IN2_A_BASIC_RC_BV_01	
Purpose:	Test of ReleaseCall procedure with two parties
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	SCF sends to SSF ReleaseCall invoke, with: - initialCallSegment (cause)
Pass criteria	Check that SSF releases the call (ReleaseReq received by SigConA and SigConB)
Postamble:	none

MSC IN2_A_BASIC_RC_BV_01



IN2_A_BASIC_RC_BV_02

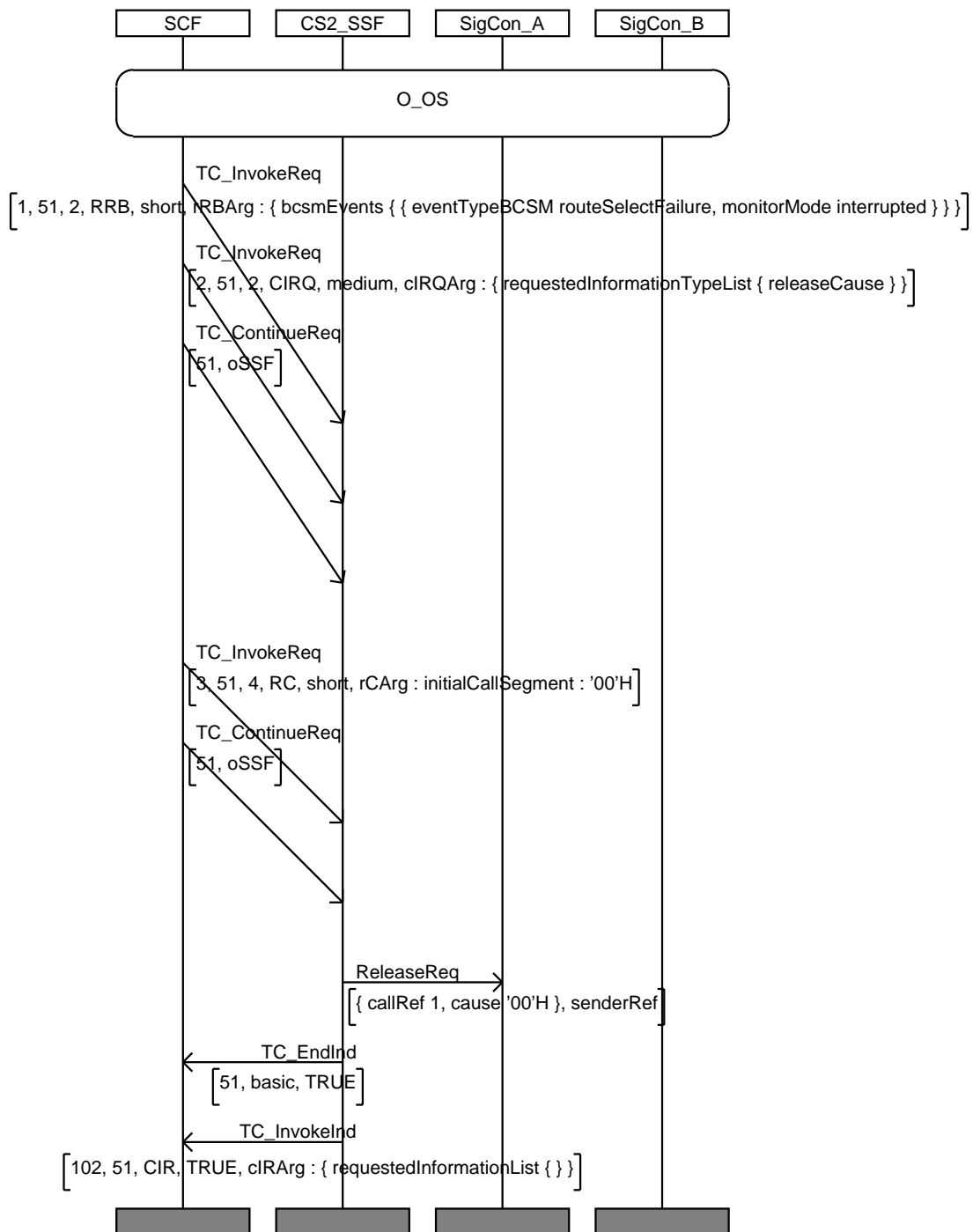
This TP was dropped.

IN2_A_BASIC_RC_BV_03

This TP was dropped.

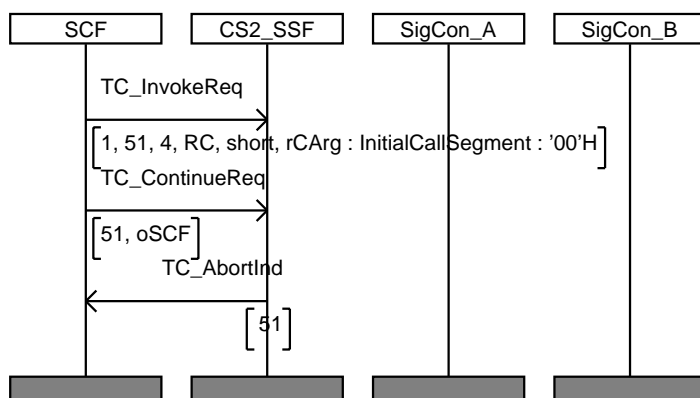
IN2_A_BASIC_RC_BV_04	
Purpose:	Test of ReleaseCall procedure in combination with CallInformation and RequestReportBCSMEvent procedures
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestReportBCSMEvent with eventTypeBCSM=routeSelectFailure followed by a CallInformationRequest invoke, with at least the parameters: <ul style="list-style-type: none"> - requestedInformationTypeList including: <ul style="list-style-type: none"> - requestedInformationType (releaseCause), Then SCF releases the call using ReleaseCall invoke with: <ul style="list-style-type: none"> - initialCallSegment (cause)
Pass criteria	<ul style="list-style-type: none"> - Check that upon detection of call release, SSF sends CallInformationReport with at least the parameters <ul style="list-style-type: none"> - requestedInformationList including: <ul style="list-style-type: none"> - requestedInformationType (releaseCause), - requestedInformationValue being releaseCauseValue used and check that no EventReportBCSM is sent Check that SigConA receives a ReleaseReq
Postamble:	none

MSC IN2_A_BASIC_RC_BV_04



IN2_A_BASIC_RC_BO_01	
Purpose:	Test of ReleaseCall procedure from wrong state (idle)
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF ReleaseCall invoke, with: - initialCallSegment (cause)
Pass criteria	Check that SSF rejects the invoke
Postamble:	none

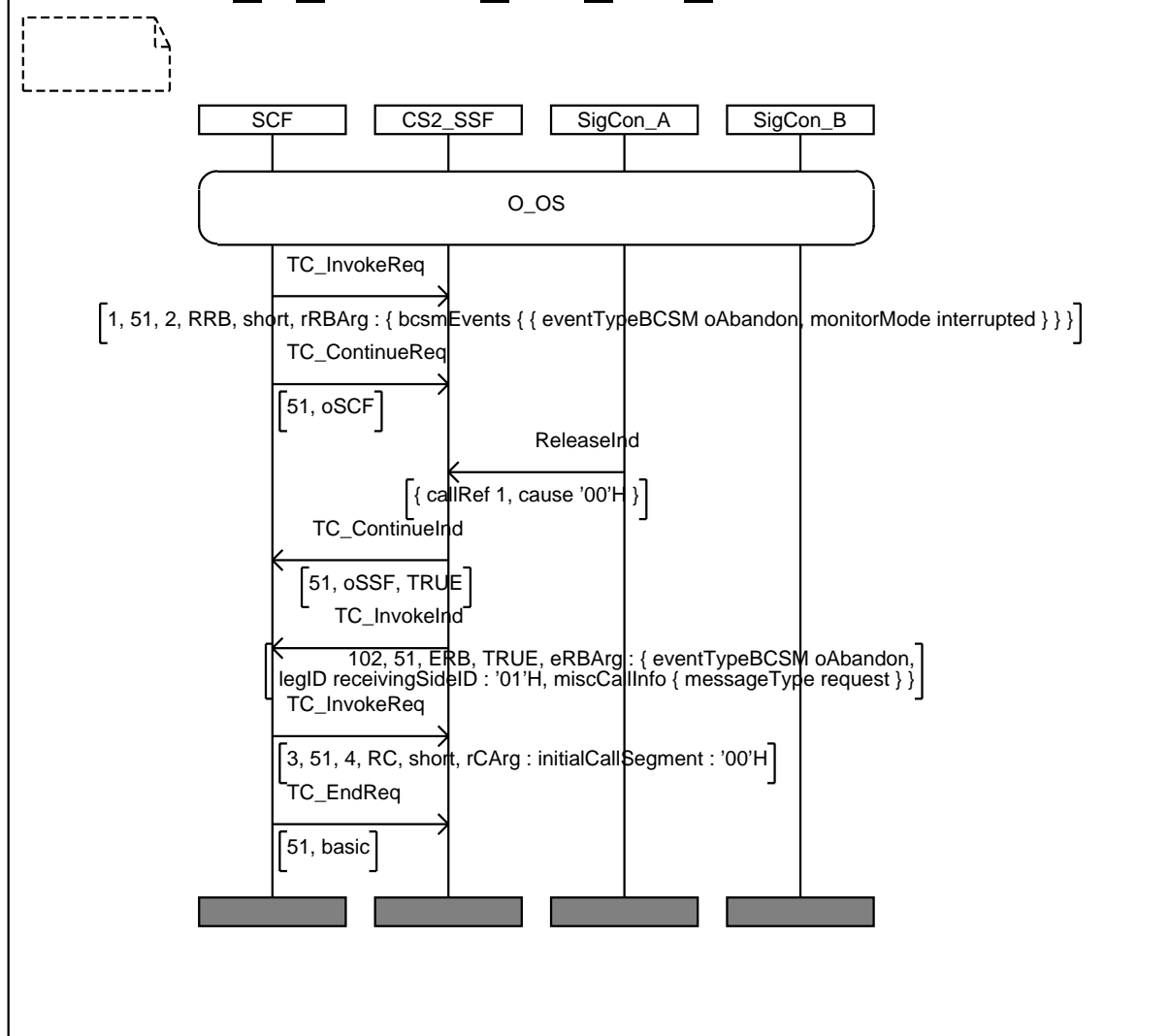
MSC IN2m_A_BASIC_RC_BO_01



6.4.14 RequestReportBCSMEvent procedure

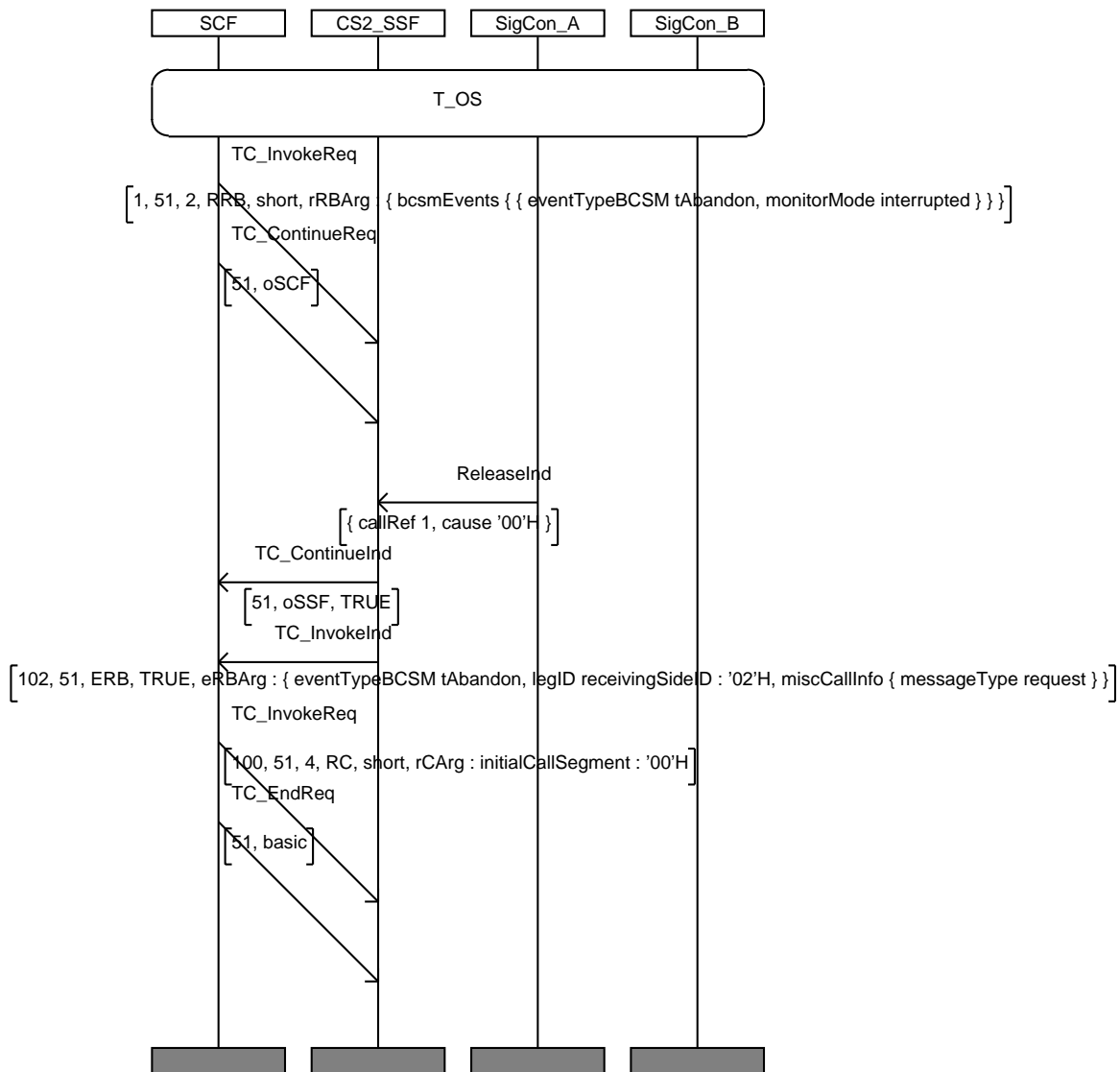
IN2_A_BASIC_RR_CA_01	
Purpose:	Test of RequestReportBCSMEvent base procedure
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=oAbandon - monitoringMode=interrupted then the calling party abandons the call before the call is answered (SigCon A to send ReleaseInd)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=oAbandon
Postamble:	none

MSC IN2_A_BASIC_RR_CA_01



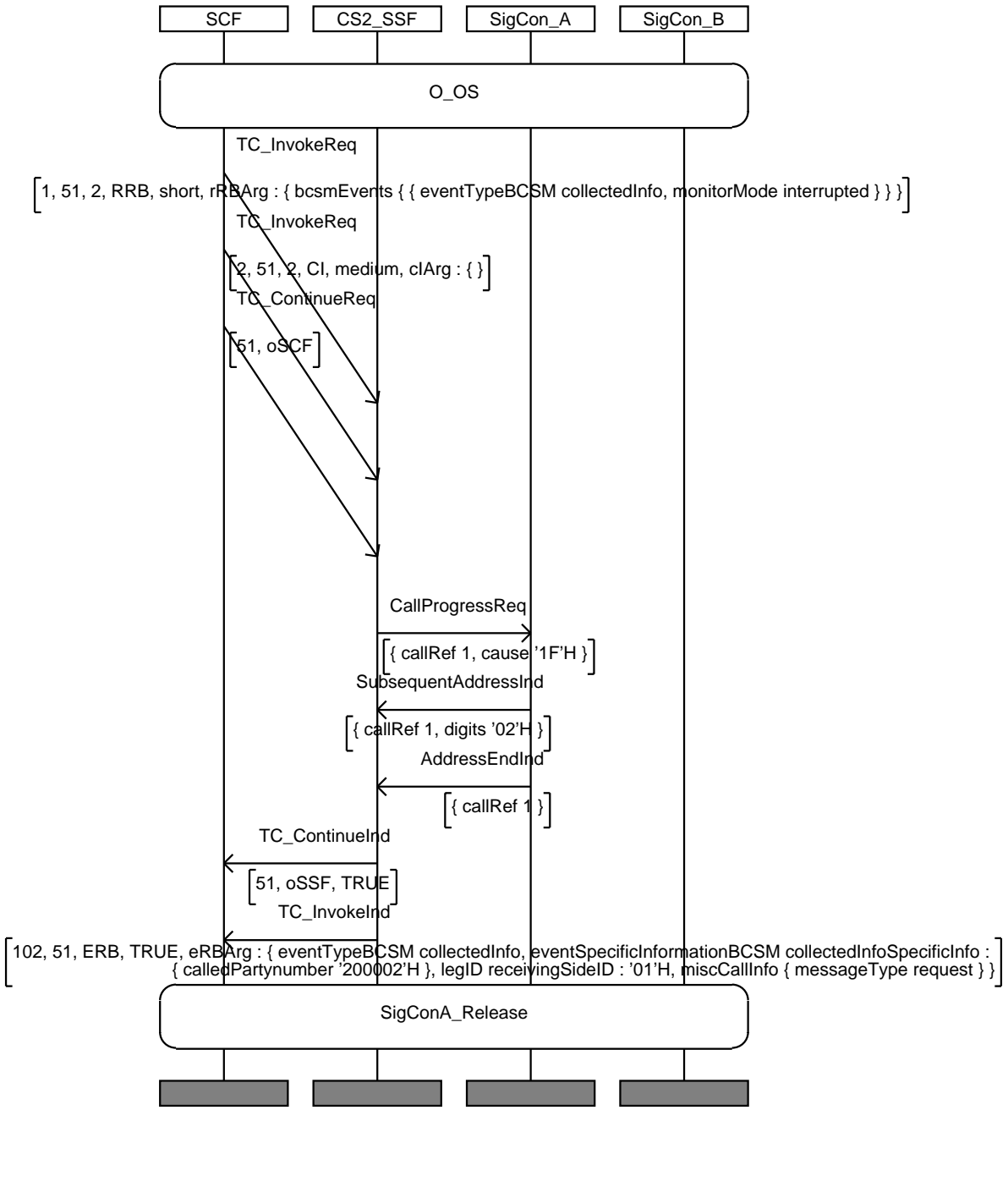
IN2_A_BASIC_RR_BV_01	
Purpose:	Test of RequestReportBCSMEvent procedure and tAbandon indication
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=tAbandon - monitoringMode=interrupted then the calling party abandons the call before the call is answered (SigCon A to send ReleaseInd)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=tAbandon
Postamble:	none

MSC IN2_A_BASIC_RR_BV_01



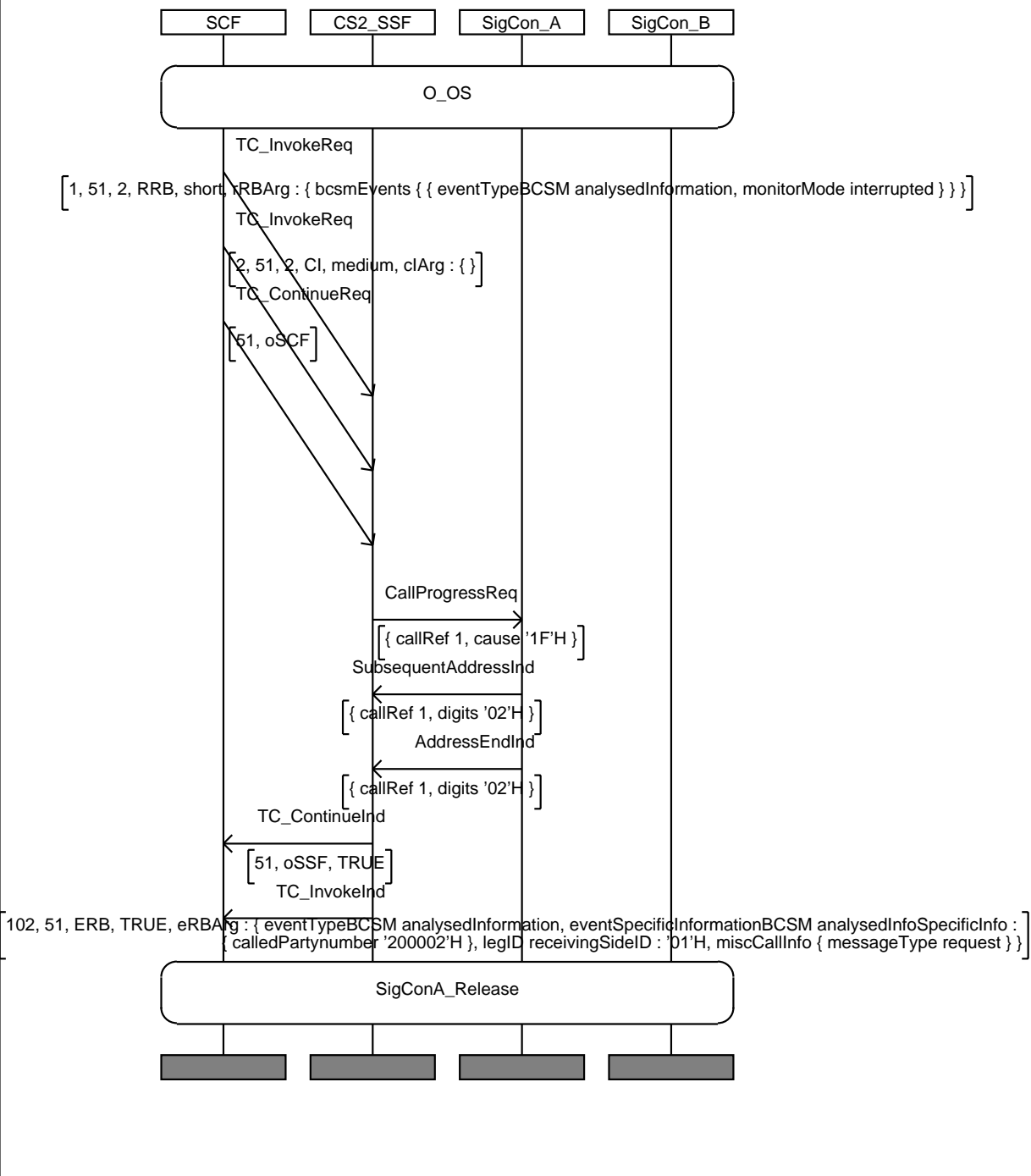
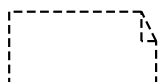
IN2_A_BASIC_RR_BV_02 (see also IN2_A_BASIC_CI_CA_01)	
Purpose:	Test of RequestReportBCSMEvent procedure and collectedInfo indication
Requirement ref	
Selection Cond.	
Preamble:	O_OS Preamble contains an InitialDP without complete digits for CalledPartyNumber
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM=collectedInfo - monitoringMode=interrupted - SCF sends a CollectInformation operation then the calling party sends the remaining digits (using CallProgressInd)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=collectedInfo
Postamble:	SigConA_Release

MSC IN2_A_BASIC_RR_BV_02



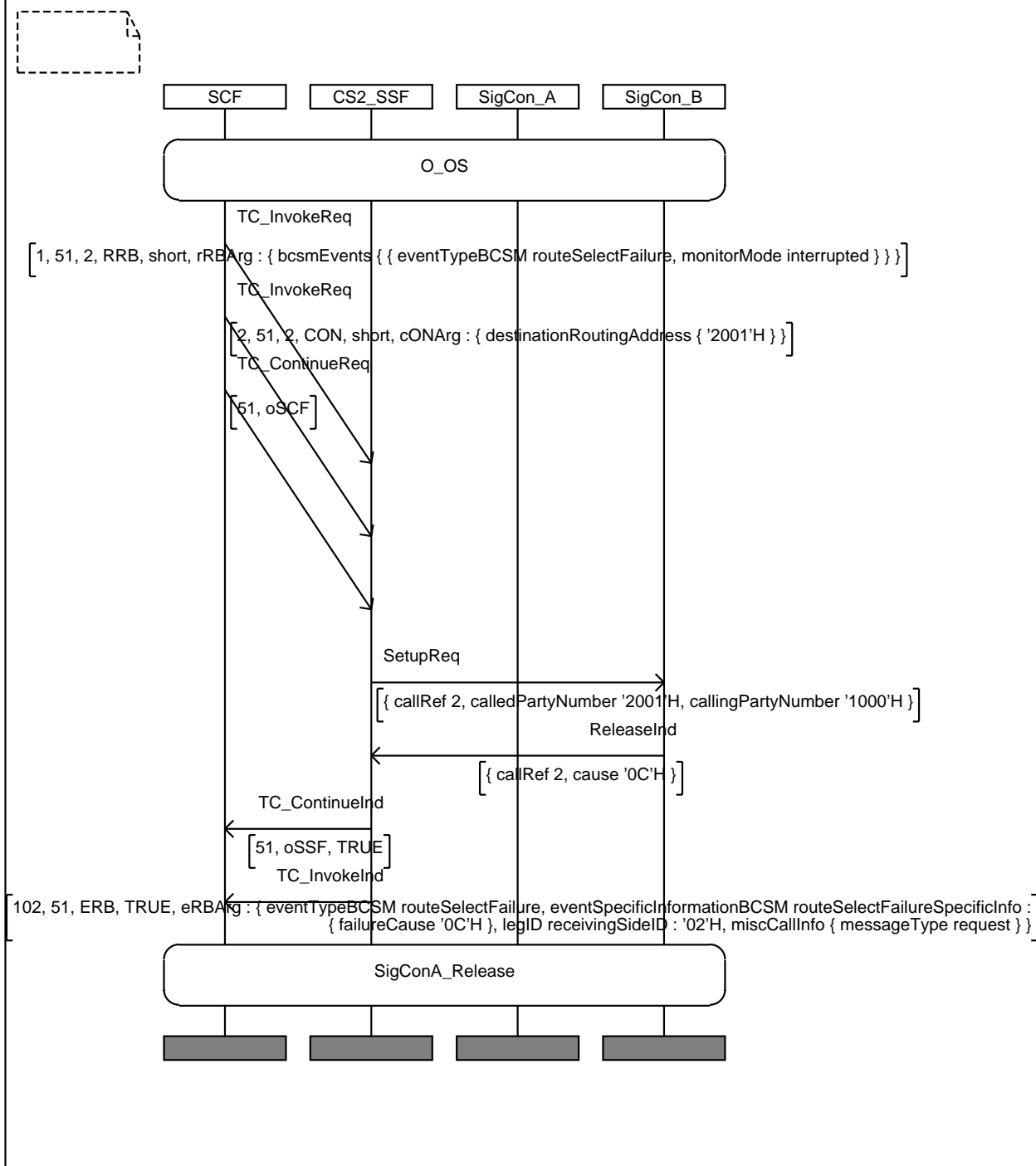
IN2_A_BASIC_RR_BV_03	
Purpose:	Test of RequestReportBCSMEvent procedure and analysedInfo indication
Requirement ref	
Selection Cond.	
Preamble:	O_OS Preamble contains an InitialDP without complete digits for CalledPartyNumber
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=analysedInfo - monitoringMode=interrupted then the calling party sends the remaining digits (after CallProgressReq is received and SubsequentAddressInd and AddressEndInd is sent)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=analysedInfo
Postamble:	SigConA_Release

MSC IN2_A_BASIC_RR_BV_03



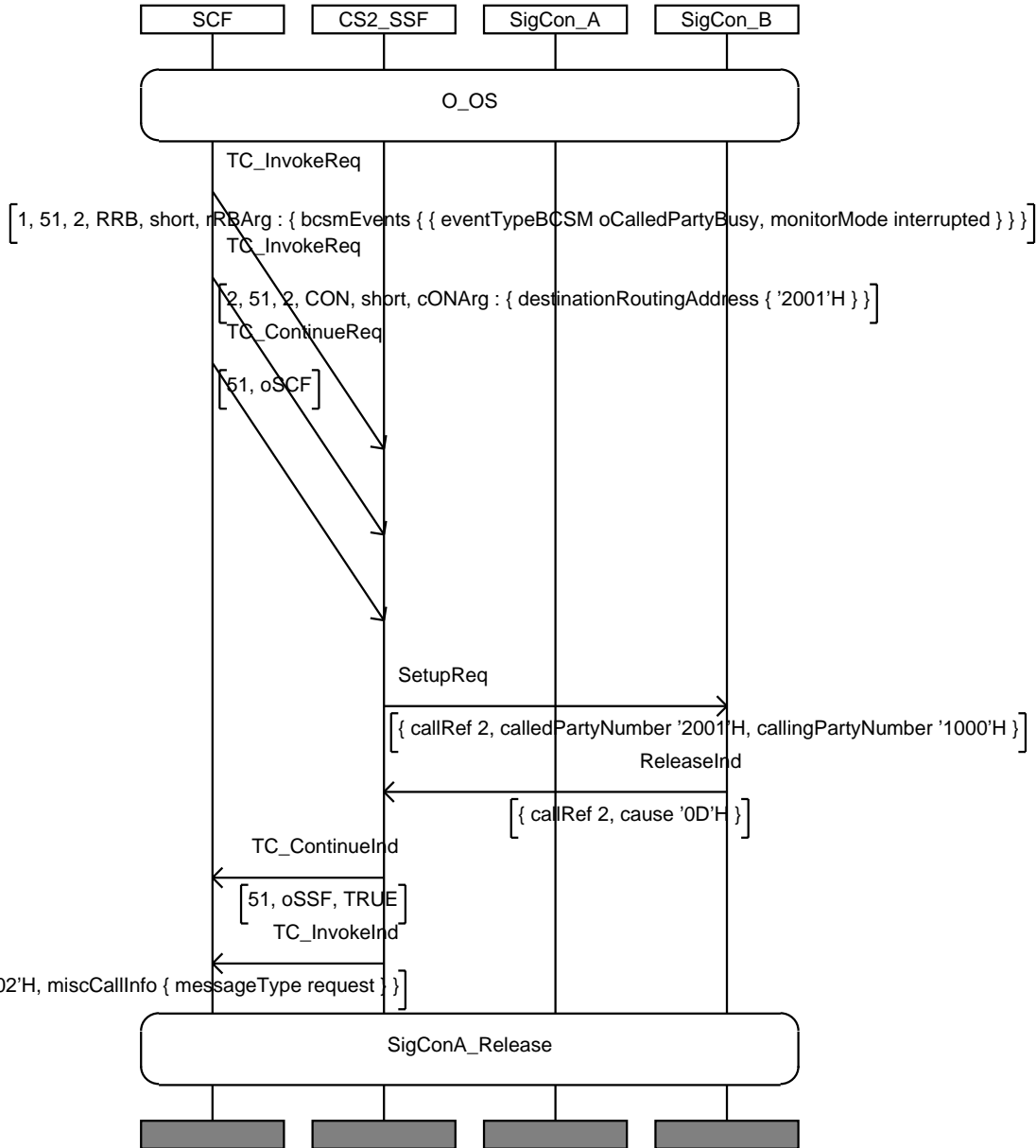
IN2_A_BASIC_RR_BV_04	
Purpose:	Test of RequestReportBCSMEvent procedure and routeSelectFailure indication
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM=routeSelectFailure - monitoringMode=interrupted followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd) because of error: routeFailure2 ("oc"H)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= routeSelectFailure
Postamble:	SigConA_Release

MSC IN2_A_BASIC_RR_BV_04



IN2_A_BASIC_RR_BV_05	
Purpose:	Test of RequestReportBCSMEvent procedure and oCalledPartyBusy indication.
Requirement ref	
Selection Cond.	
Preamble:	O_OS In addition, user B is declared busy
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=oCalledPartyBusy - monitoringMode=interrupted followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd) with bPtyBusy_UDUB
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oCalledPartyBusy
Postamble:	SigConA_Release

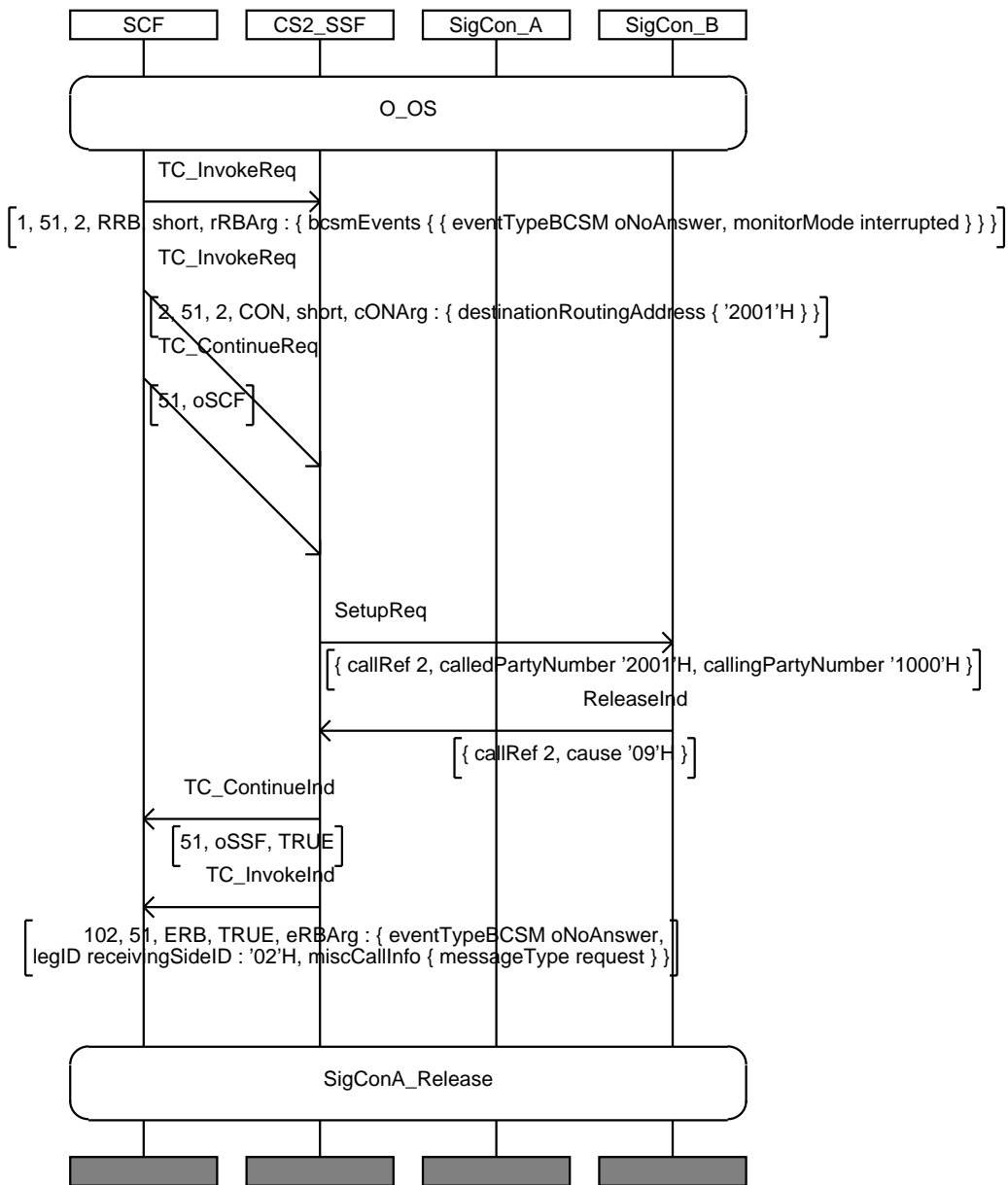
MSC IN2_A_BASIC_RR_BV_05



legID receivingSideD : '02'H, miscCallInfo { messageType request }

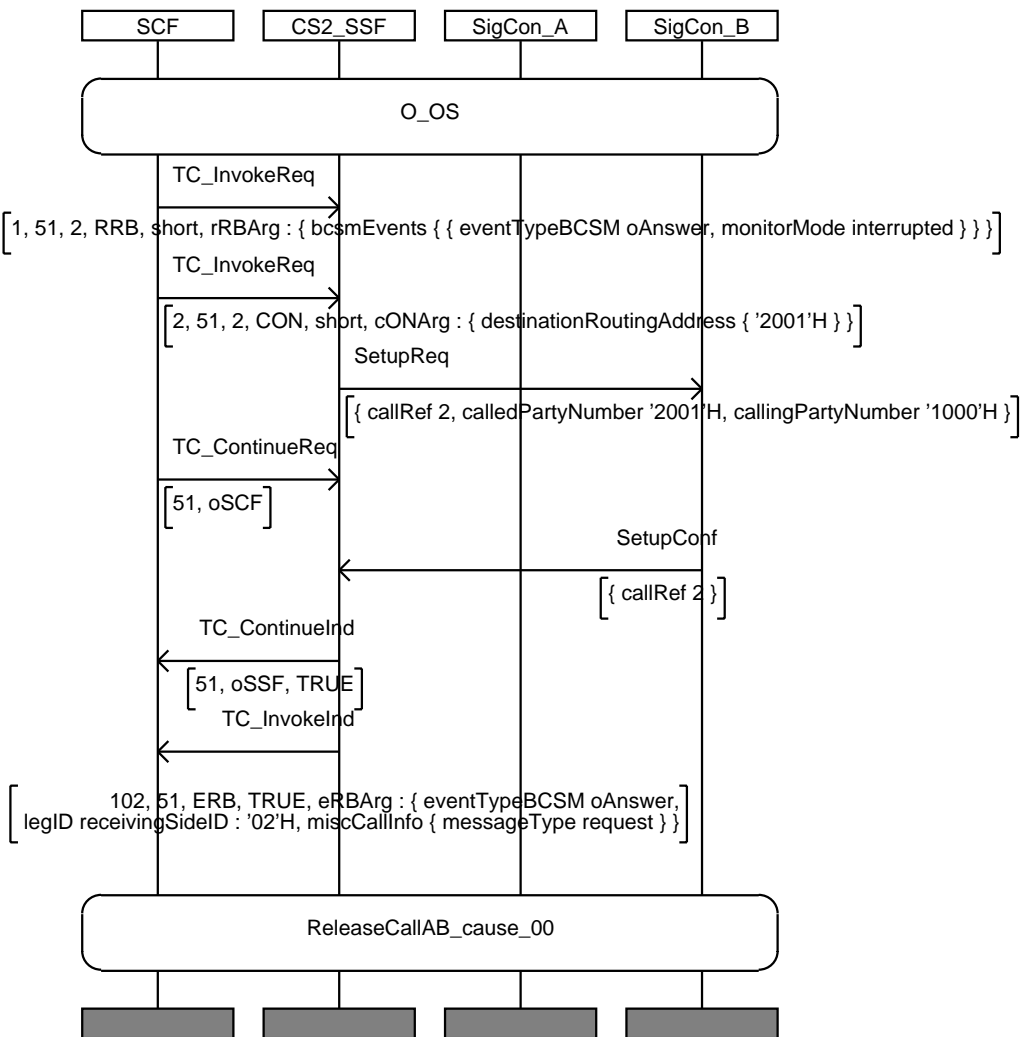
IN2_A_BASIC_RR_BV_06	
Purpose:	Test of RequestReportBCSMEvent procedure and oNoAnswer indication.
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM=oNoAnswer - monitoringMode=interrupted followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd) because user B does not answer
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oNoAnswer
Postamble:	SigConA_Release

MSC IN2_A_BASIC_RR_BV_06



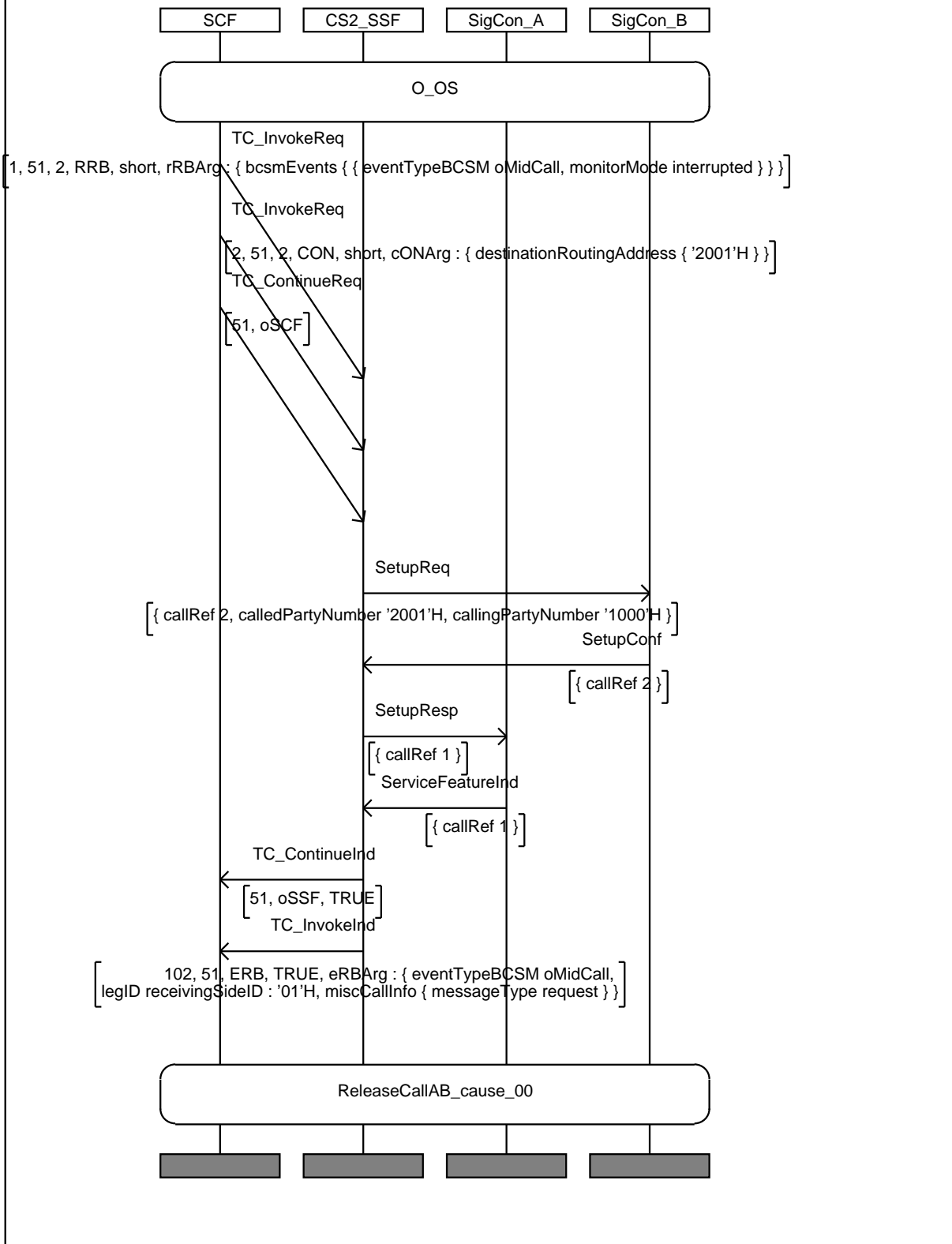
IN2_A_BASIC_RR_BV_07	
Purpose:	Test of RequestReportBCSMEvent procedure and oAnswer indication.
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=oAnswer - monitoringMode=interrupted followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B answers the call (SetupConf from SigConB to SSF)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oAnswer
Postamble:	ReleaseCallAB_cause_00

MSC IN2_A_BASIC_RR_BV_07



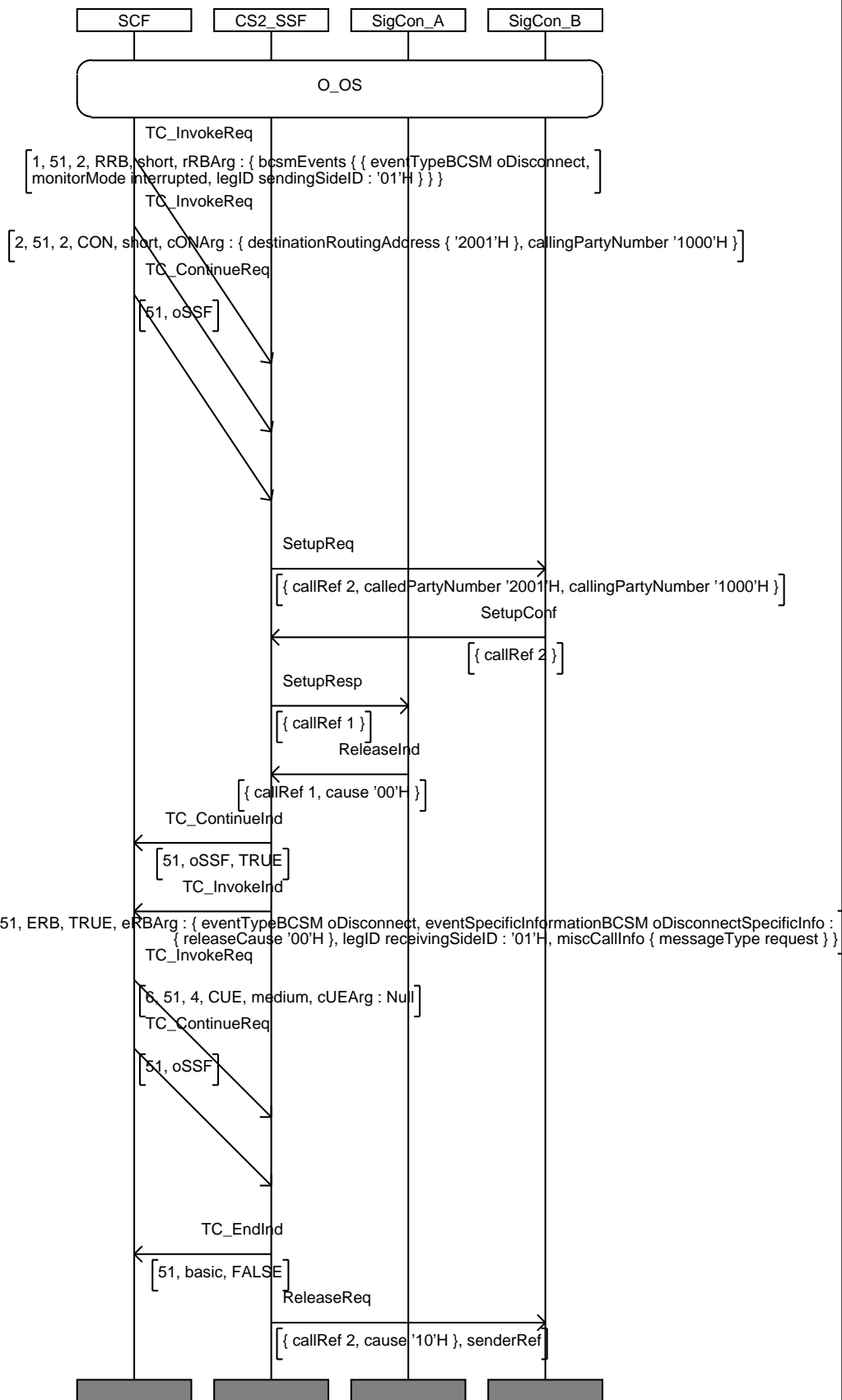
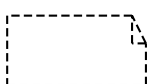
IN2_A_BASIC_RR_BV_08	
Purpose:	Test of RequestReportBCSMEvent procedure and oMidCall indication.
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oMidCall - monitoringMode=interrupted <p>followed by a Connect invoke</p> <p>Then SSF sends a SetupReq to SigCon B. SetupConf from SigConB is received by SSF which issues SetupResp to SigConA.</p> <p>SigConA calling party initiates a service (ServiceFeatureInd sent to SSF) and oMidCall DP is reached</p>
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oMidCall
Postamble:	ReleaseCallAB_cause_00

MSC IN2_A_BASIC_RR_BV_08



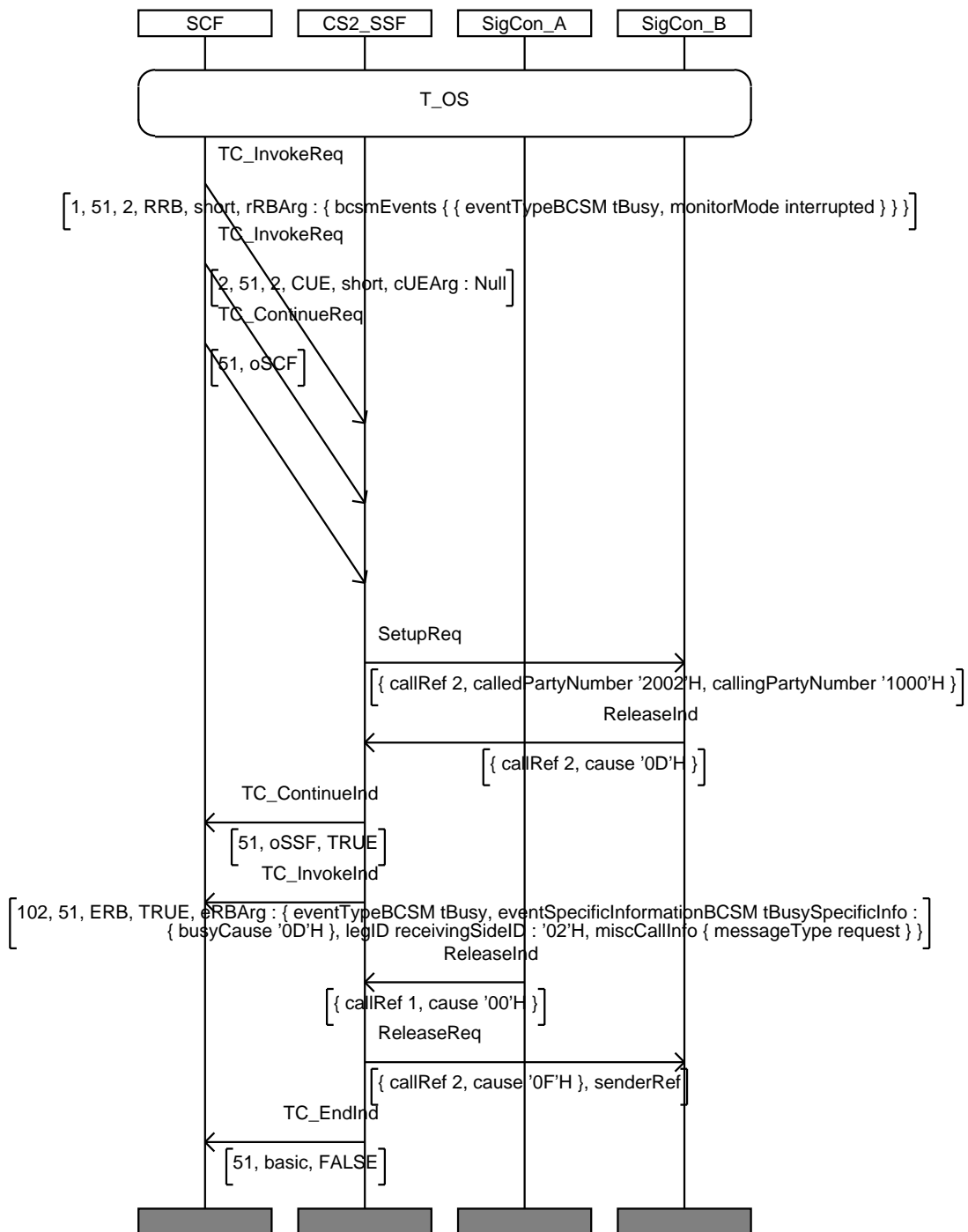
IN2_A_BASIC_RR_BV_09	
Purpose:	Test of RequestReportBCSMEvent procedure and oDisconnect indication.
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM= oDisconnect - monitoringMode=interrupted followed by a Connect invoke Then SSF establishes the call (a SetupReq to SigCon B. SetupConf from SigConB to SSF, then SetupResp to SigConA) SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oDisconnect SCF sends a Continue operation, check that the B side receives a RelReq
Postamble:	none

MSC IN2_A_BASIC_RR_BV_09



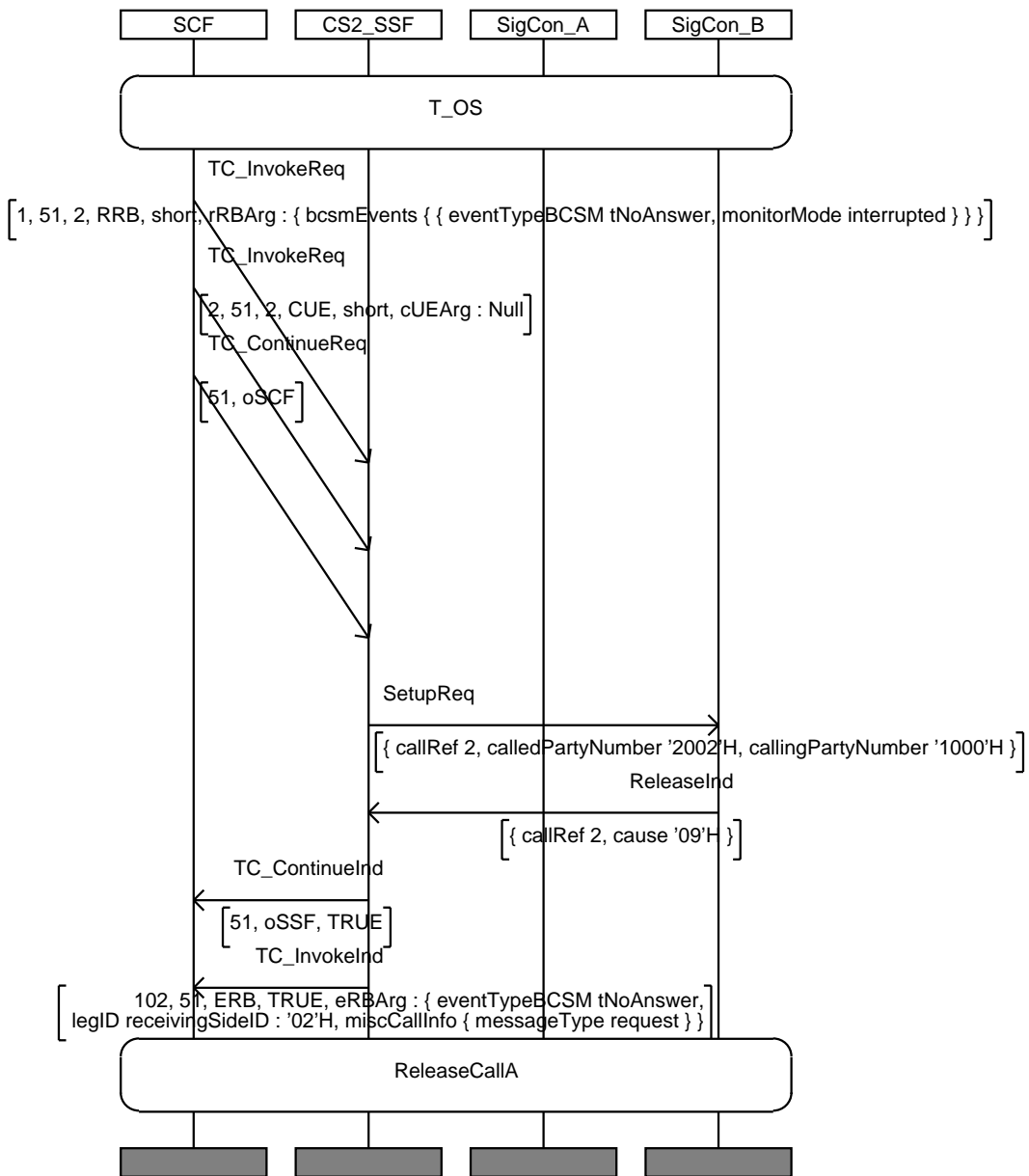
IN2_A_BASIC_RR_BV_10	
Purpose:	Test of RequestReportBCSMEvent procedure and tBusy indication.
Requirement ref	
Selection Cond.	
Preamble:	T_OS In addition, user B is declared busy
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM=tBusy - monitoringMode=interrupted followed by a Continue invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd sent) with bPtyBusy_UDUB
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tBusy
Postamble:	none

MSC IN2_A_BASIC_RR_BV_10



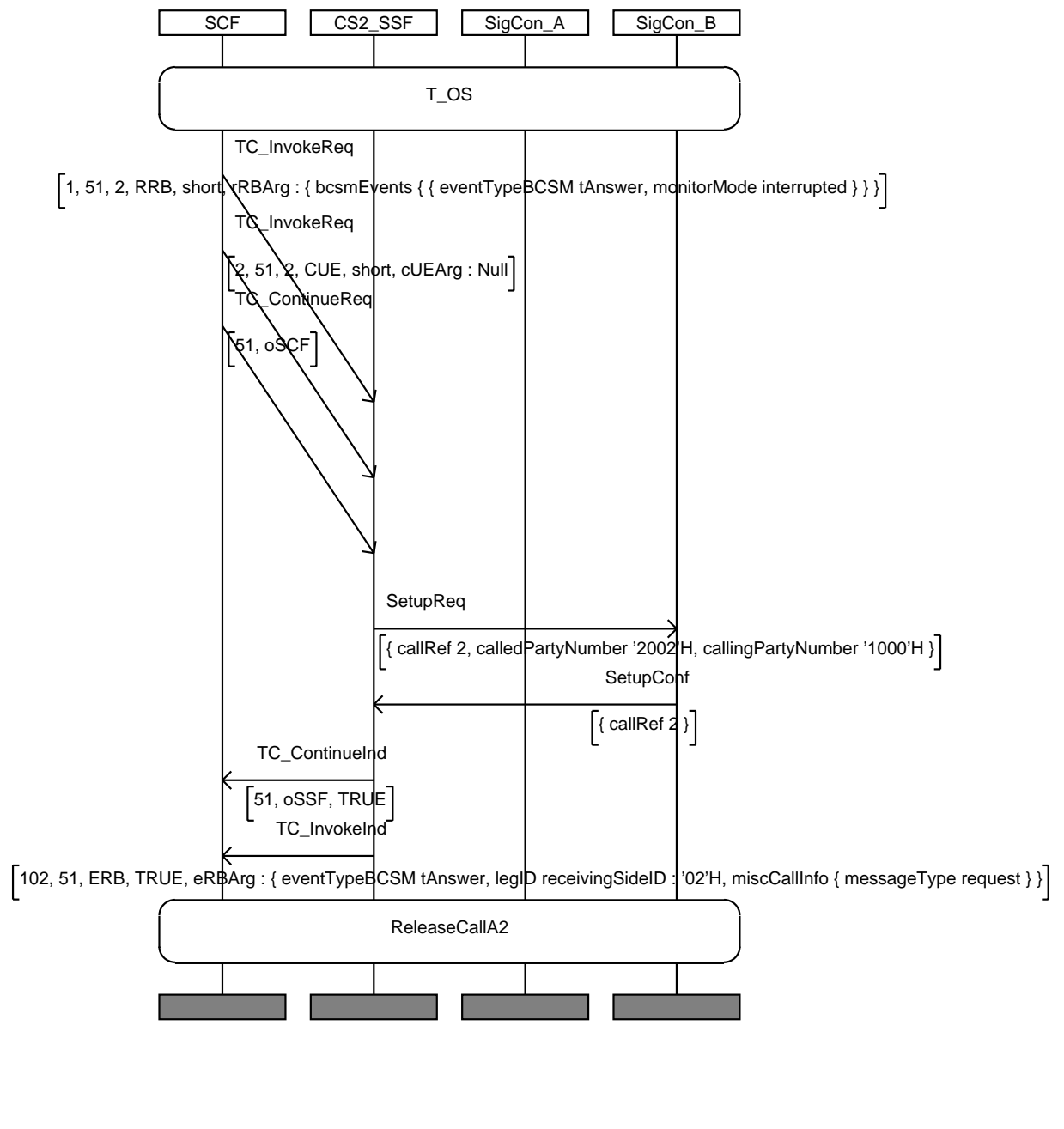
IN2_A_BASIC_RR_BV_11	
Purpose:	Test of RequestReportBCSMEvent procedure and tNoAnswer indication.
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=tNoAnswer - monitoringMode=interrupted followed by a Continue invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd sent) because user B does not answer
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tNoAnswer
Postamble:	ReleaseCallA

MSC IN2_A_BASIC_RR_BV_11



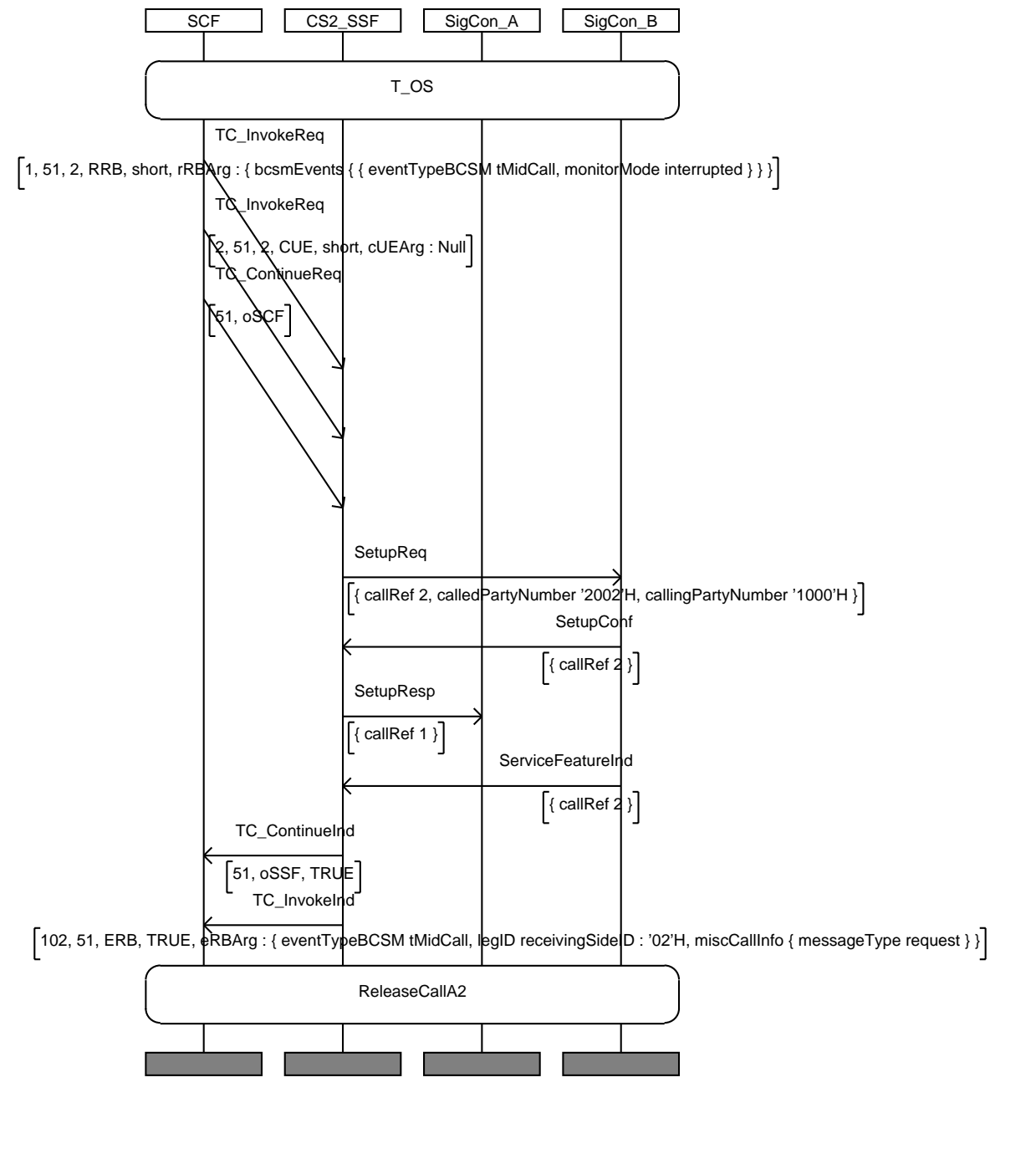
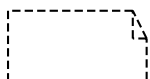
IN2_A_BASIC_RR_BV_12	
Purpose:	Test of RequestReportBCSMEvent procedure and tAnswer indication.
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=tAnswer - monitoringMode=interrupted followed by a Continue invoke Then SSF sends a SetupReq to SigCon B SigCon B answers the call (SetupConf from SigConB to SSF)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tAnswer
Postamble:	ReleaseCallA

MSC IN2_A_BASIC_RR_BV_12



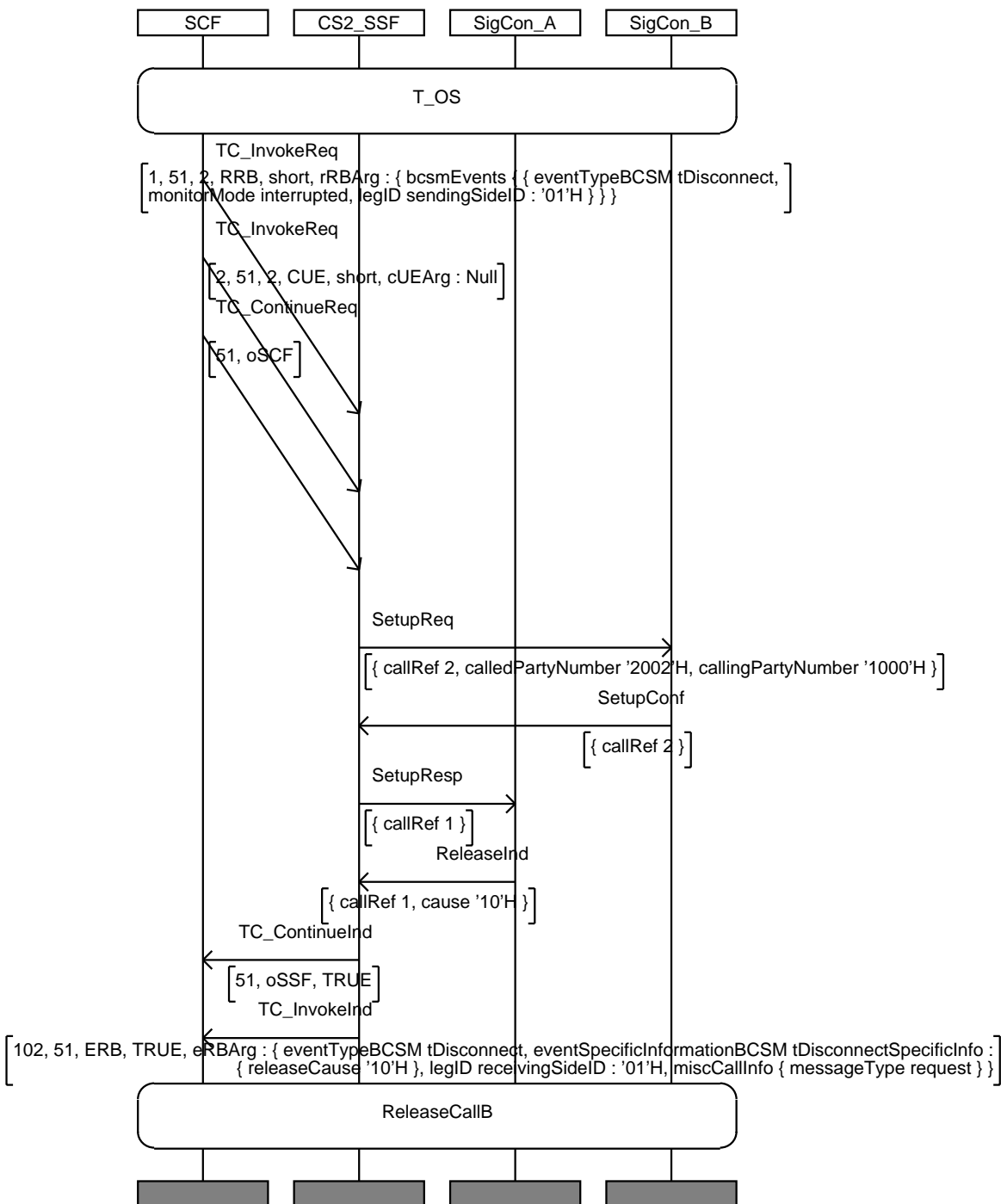
IN2_A_BASIC_RR_BV_13	
Purpose:	Test of RequestReportBCSMEvent procedure and tMidCall indication.
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= tMidCall - monitoringMode=interrupted <p>followed by a Continue invoke</p> <p>Then SSF sends a SetupReq to SigCon B. SetupConf from SigConB is received by SSF which issues SetupResp to SigConA.</p> <p>SigConB called party initiates a service (ServiceFeatureInd sent to SSF) and tMidCall DP is reached</p>
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tMidCall
Postamble:	ReleaseCallA

MSC IN2_A_BASIC_RR_BV_13



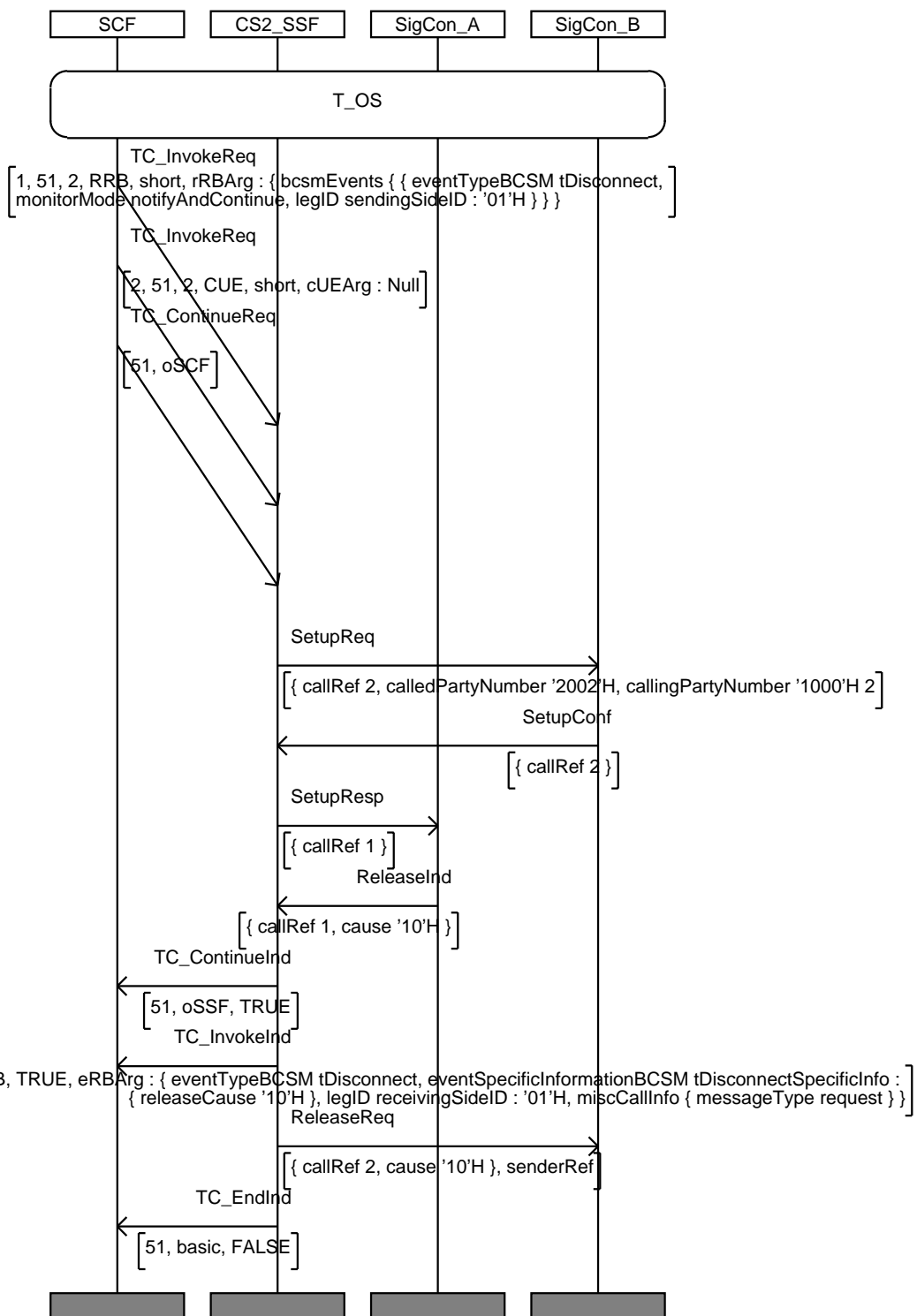
IN2_A_BASIC_RR_BV_14	
Purpose:	Test of RequestReportBCSMEvent procedure and tDisconnect indication.
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM= tDisconnect - monitoringMode=interrupted followed by a Continue invoke Then SSF establishes the call (a SetupReq to SigCon B. SetupConf from SigConB to SSF which sends SetupResp to SigConA) SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tDisconnect
Postamble:	ReleaseCallB

MSC IN2_A_BASIC_RR_BV_14



IN2_A_BASIC_RR_BV_15	
Purpose:	Test of RequestReportBCSMEvent procedure and tDisconnect indication.
Requirement ref	
Selection Cond.	
Preamble:	T_OS
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM= tDisconnect - monitoringMode=notifyAndContinue followed by a Continue invoke The IUT establishes the call, sends a SetUpReq to B side Then SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)
Pass criteria	- Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tDisconnect - Check that SigConB is receiving a ReleaseReq to continue clearing the call
Postamble:	none

MSC IN2_A_BASIC_RR_BV_15



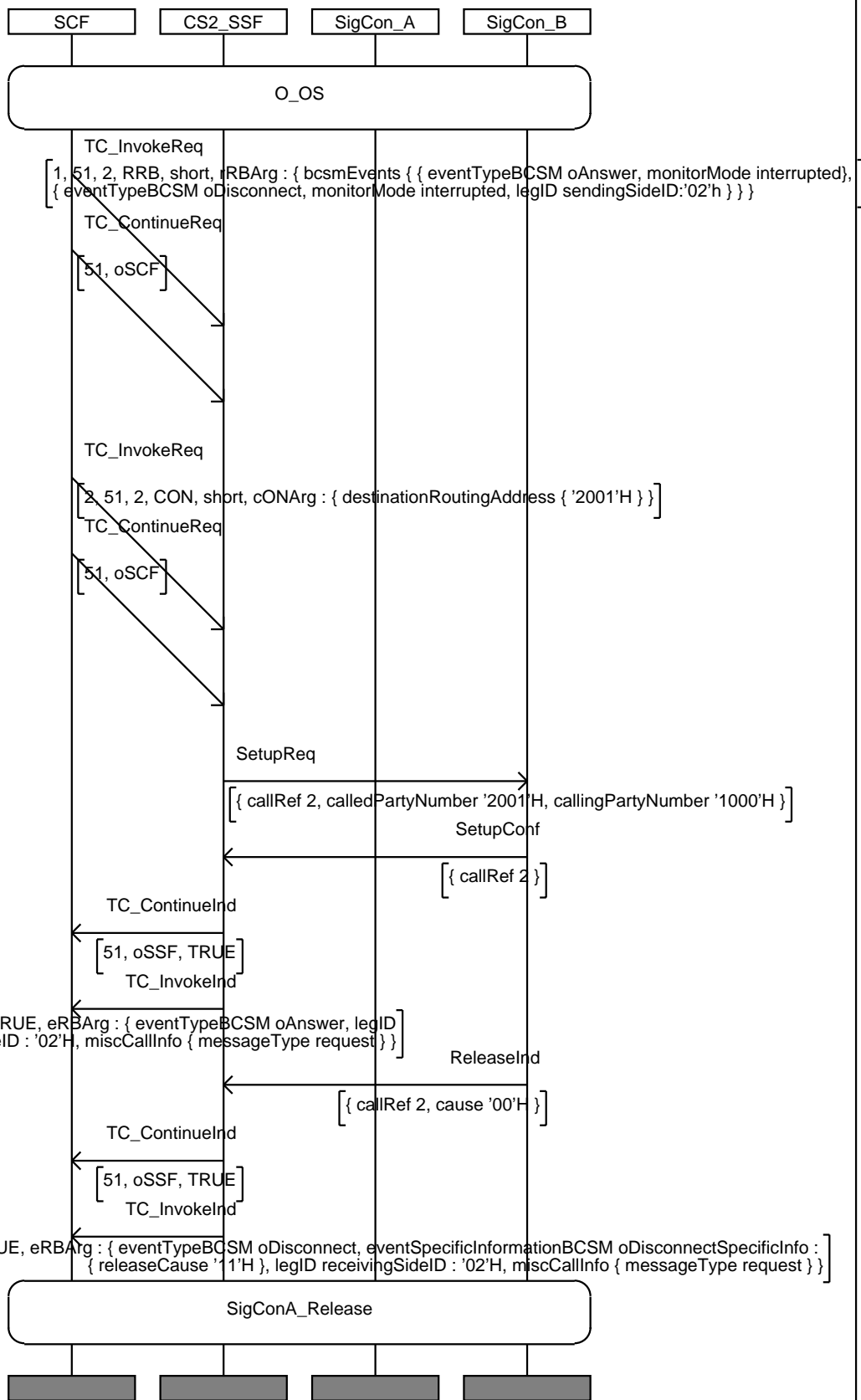
IN2_A_BASIC_RR_BV_16

This TP has been deleted.

IN2_A_BASIC_RR_BV_17

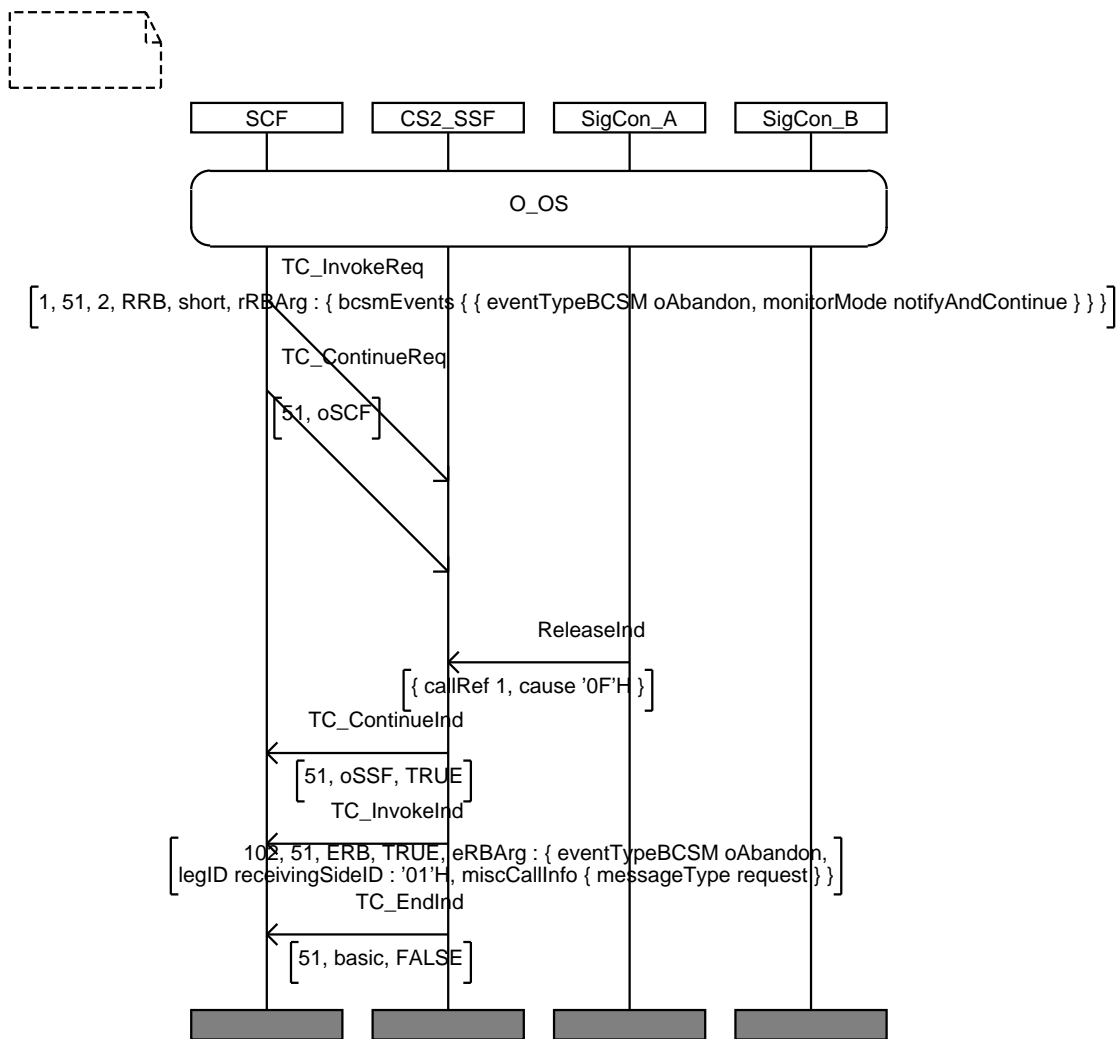
IN2_A_BASIC_RR_BV_17	
Purpose:	Test of RequestReportBCSMEvent procedure and oTriggers
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF - SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM= oAnswer - monitoringMode=interrupted - eventTypeBCSM= oDisconnect - monitoringMode=interrupted followed by Continue invoke - SSF calls SigConB (SetupReq answered with SetupConf)
Pass criteria	- Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oAnswer - When SigConB is releasing the call (ReleaseInd sent), check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oDisconnect
Postamble:	SigConA_Release

MSC IN2_A_BASIC_RR_BV_17

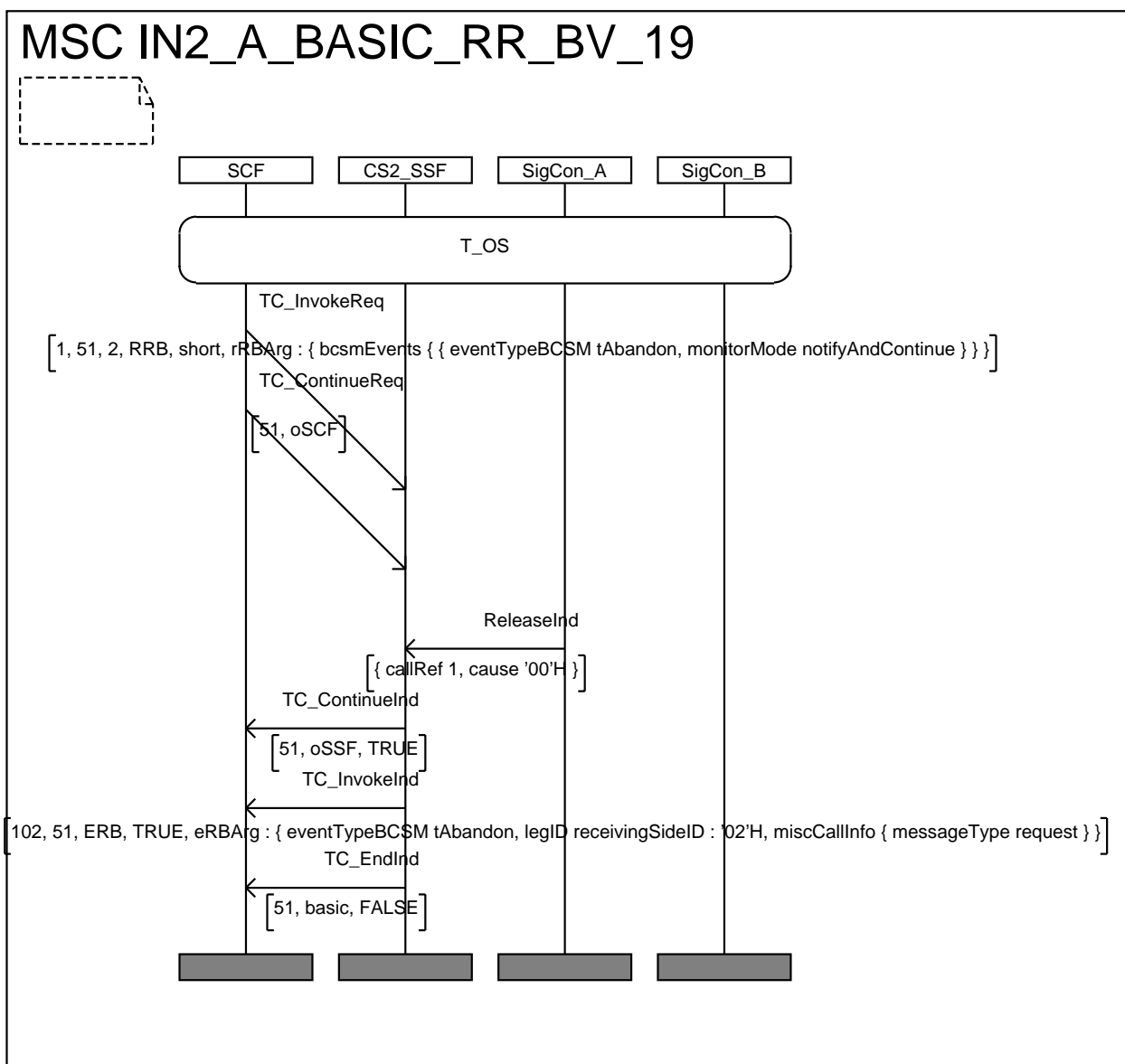


XXXX		IN2_A_BASIC_RR_BV_18	
Purpose:	Test of RequestReportBCSMEvent procedure and oAbandon		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=oAbandon - monitoringMode=notifyAndContinue then the calling party abandons the call before the call is answered (SigCon A to send ReleaseInd)		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=oAbandon		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_18

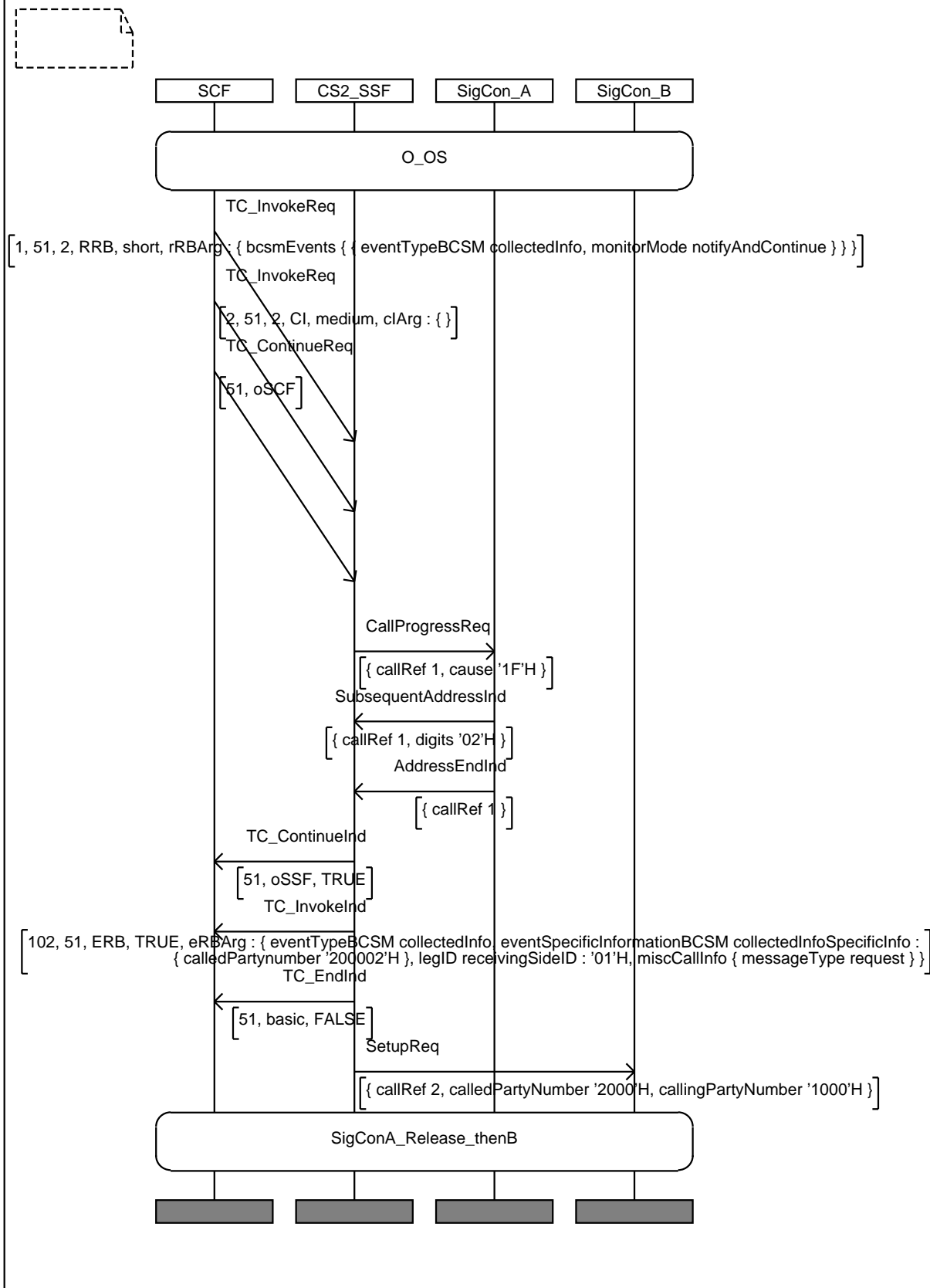


XXXX		IN2_A_BASIC_RR_BV_19	
Purpose:	Test of RequestReportBCSMEvent procedure and tAbandon		
Requirement ref			
Selection Cond.			
Preamble:	T_OS		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=tAbandon - monitoringMode=notifyAndContinue then the calling party abandons the call before the call is answered (SigCon A to send ReleaseInd)		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=tAbandon		
Postamble:	none		



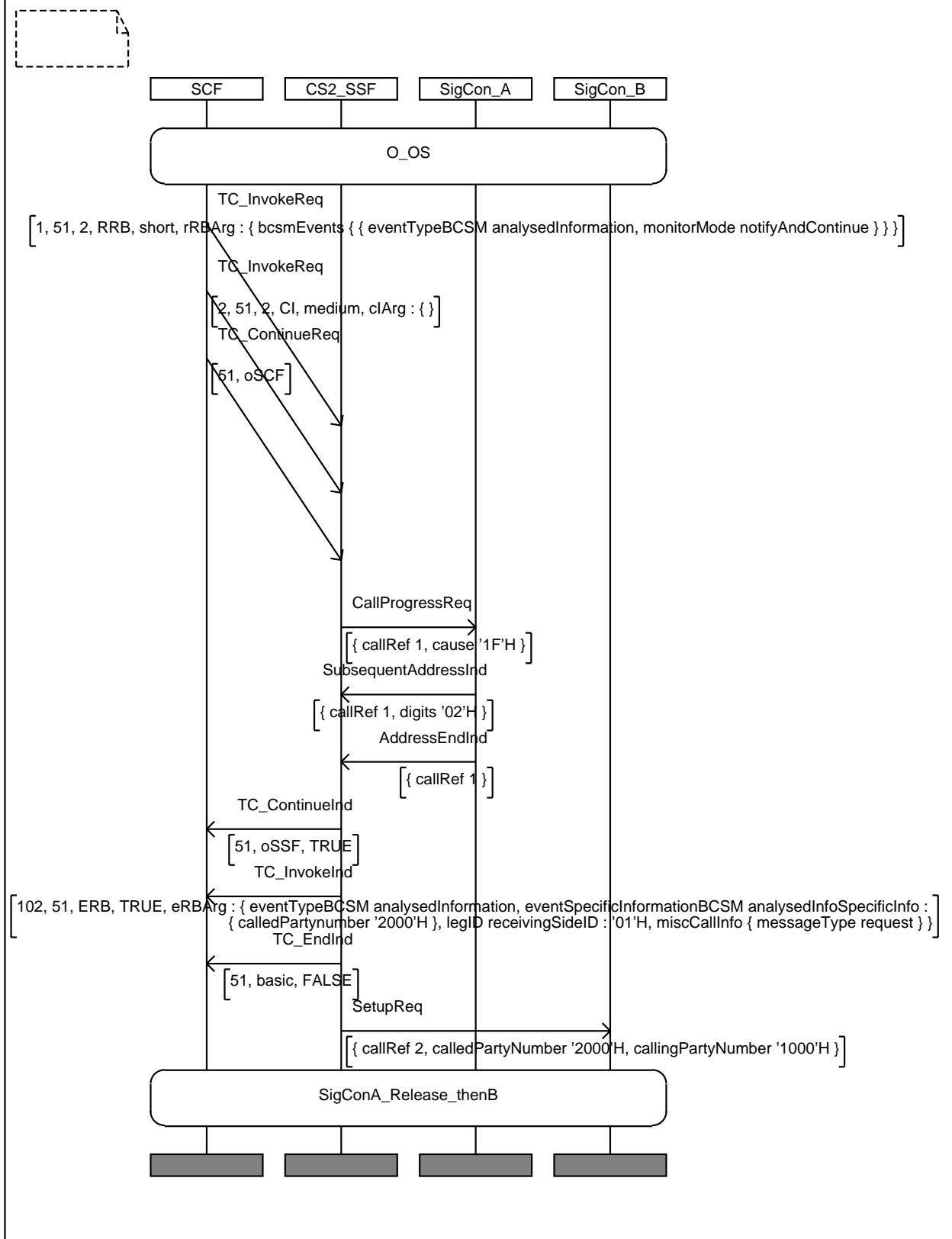
XXXX		IN2_A_BASIC_RR_BV_20	
Purpose:		Test of RequestReportBCSMEvent procedure and collectedInfo indication	
Requirement ref			
Selection Cond.			
Preamble:		O_OS Preamble contains an InitialDP without complete digits for CalledPartyNumber	
Test description		SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=collectedInfo - monitoringMode= notifyAndContinue then the calling party sends the remaining digits (using CallProgressInd)	
Pass criteria		Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=collectedInfo	
Postamble:		SigConA_Release_thenB	

MSC IN2_A_BASIC_RR_BV_20

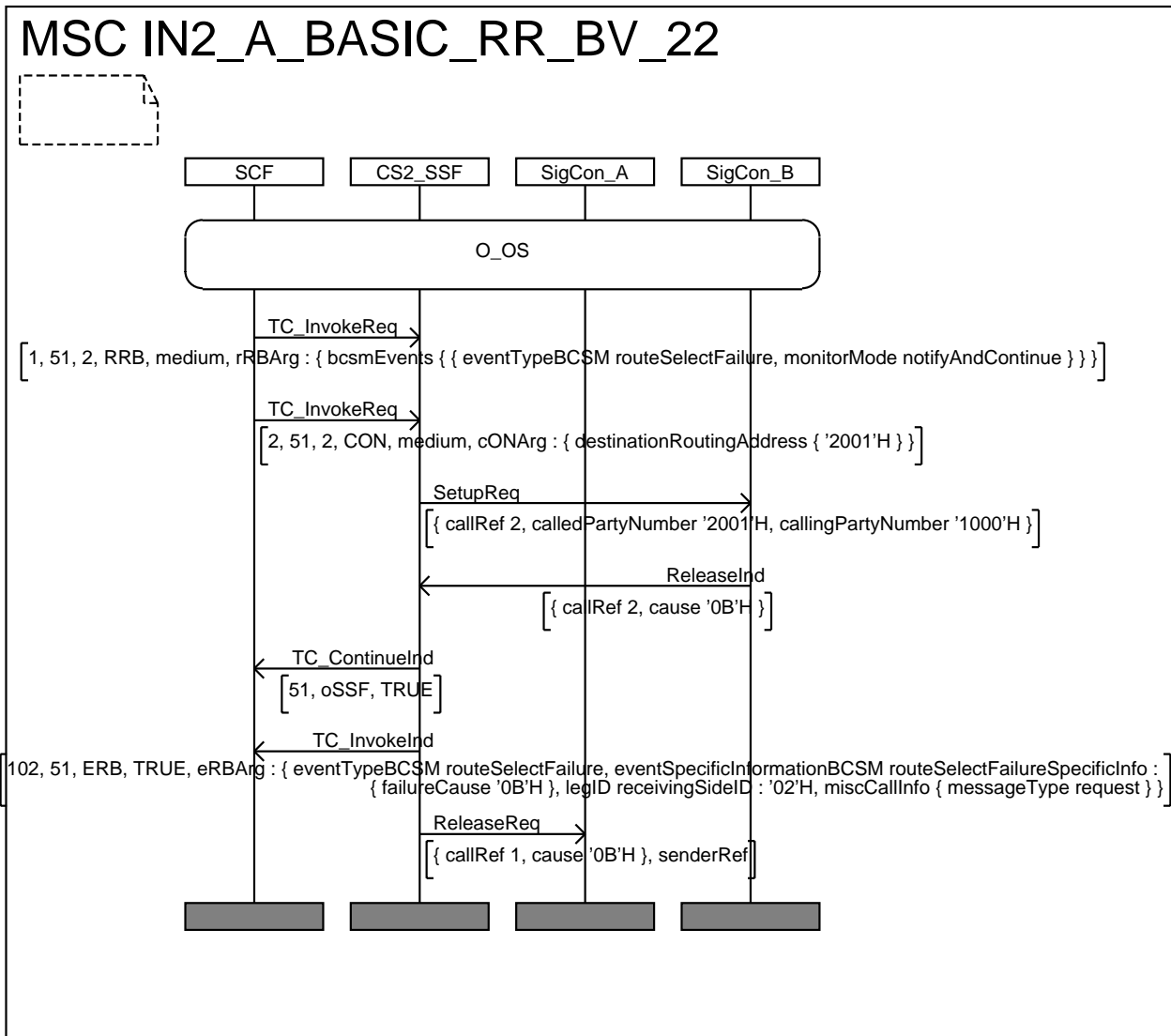


XXXX		IN2_A_BASIC_RR_BV_21	
Purpose:		Test of RequestReportBCSMEvent procedure and analysedInfo indication	
Requirement ref			
Selection Cond.			
Preamble:	O_OS	Preamble contains an InitialDP without complete digits for CalledPartyNumber	
Test description		<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> <input type="checkbox"/> eventTypeBCSM=analysedInfo monitoringMode= notifyAndContinue <input type="checkbox"/> followed by CollectInformation operation then the calling party sends the remaining digits (after CallProgressReq is received and SubsequentAddressInd and AddressEndInd is sent) 	
Pass criteria		Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM=analysedInfo	
Postamble:		SigConA_Release_thenB	

MSC IN2_A_BASIC_RR_BV_21

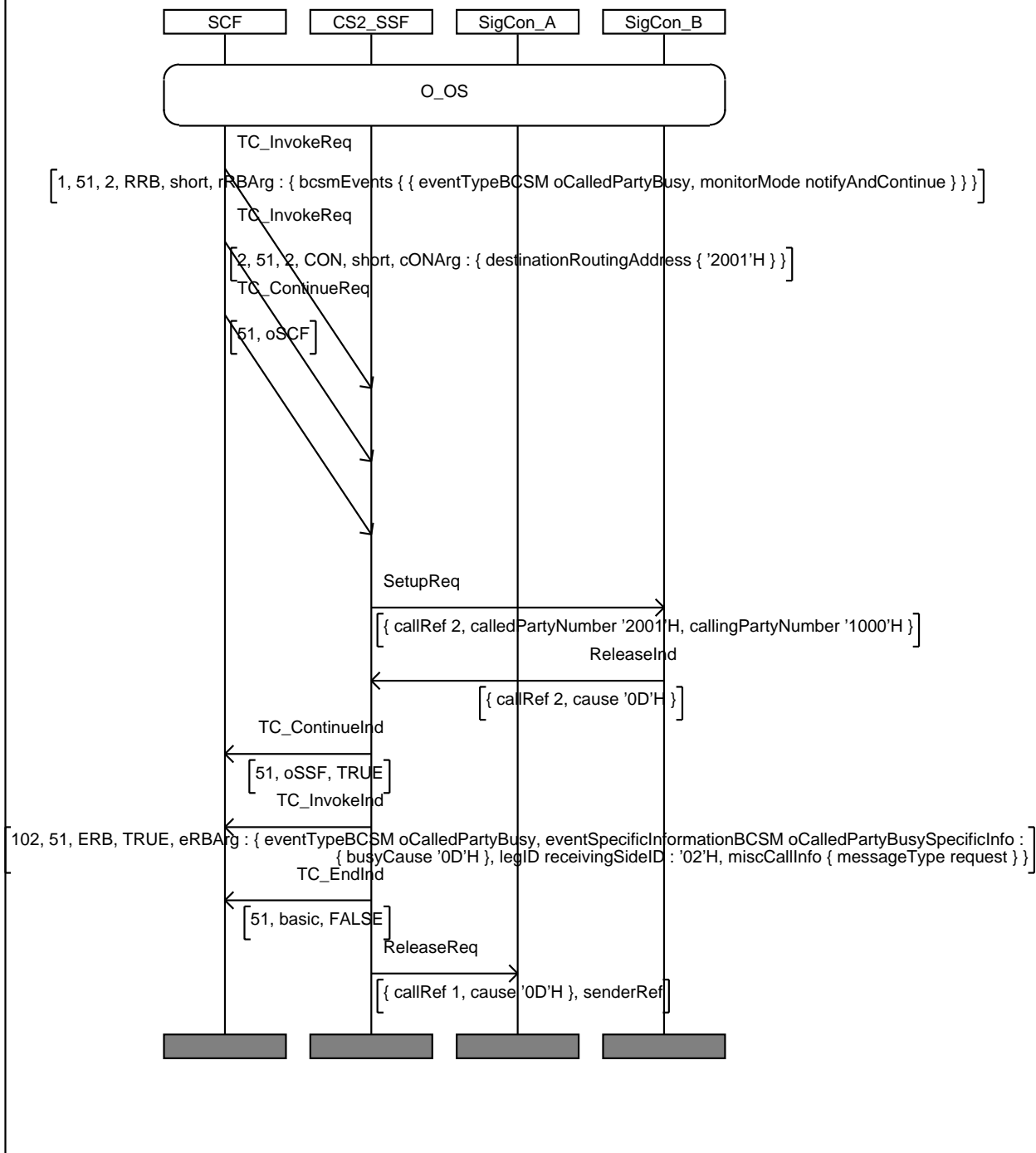


XXXX		IN2_A_BASIC_RR_BV_22	
Purpose:	Test of RequestReportBCSMEvent procedure and routeSelectFailure indication		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=routeSelectFailure - monitoringMode= notifyAndContinue followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd) with cause routeFailure2		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= routeSelectFailure		
Postamble:	none		



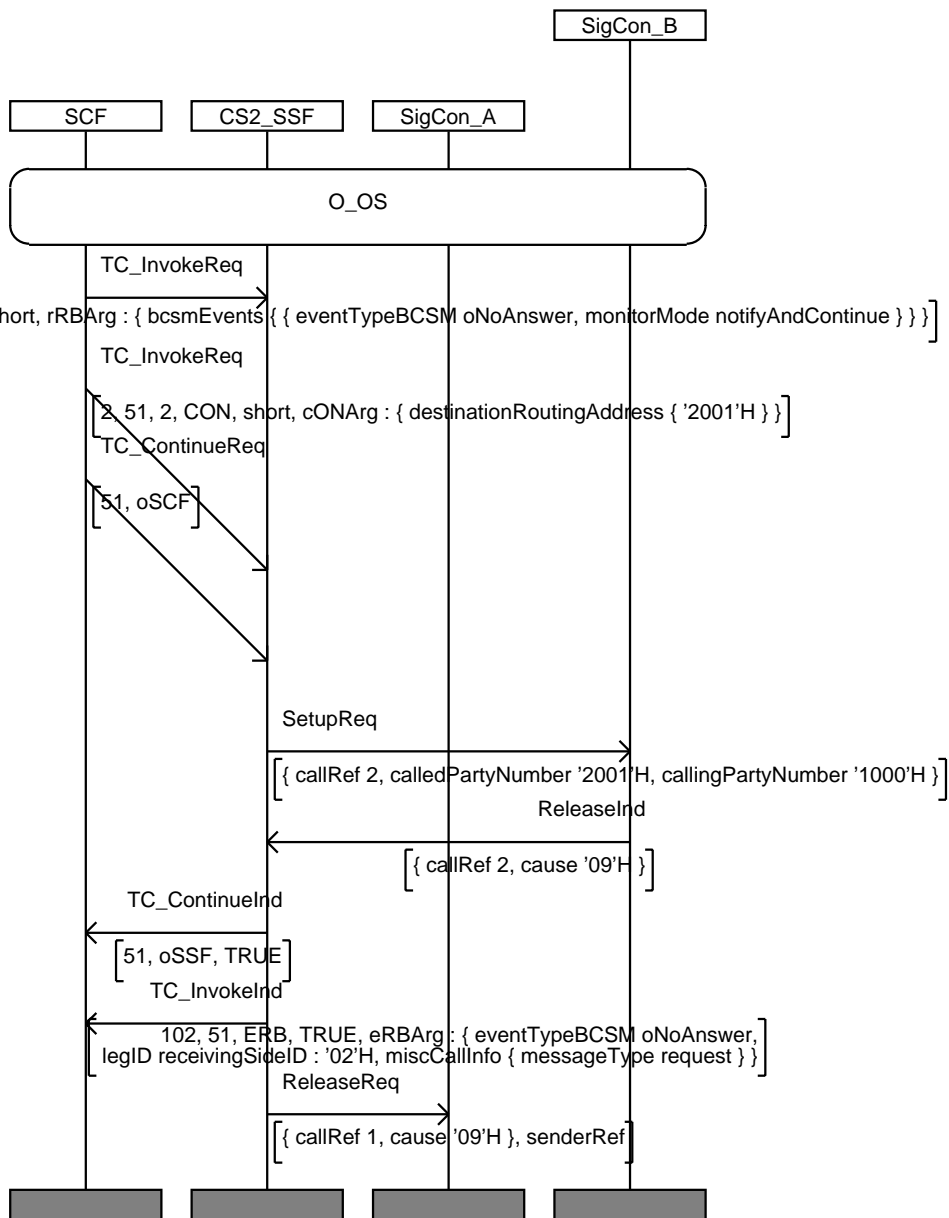
XXXX		IN2_A_BASIC_RR_BV_23	
Purpose:	Test of RequestReportBCSMEvent procedure and oCalledPartyBusy indication.		
Requirement ref			
Selection Cond.			
Preamble:	O_OS In addition, user B is declared busy		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=oCalledPartyBusy - monitoringMode= notifyAndContinue followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd) with cause bPtyBusy_UDUB		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oCalledPartyBusy		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_23



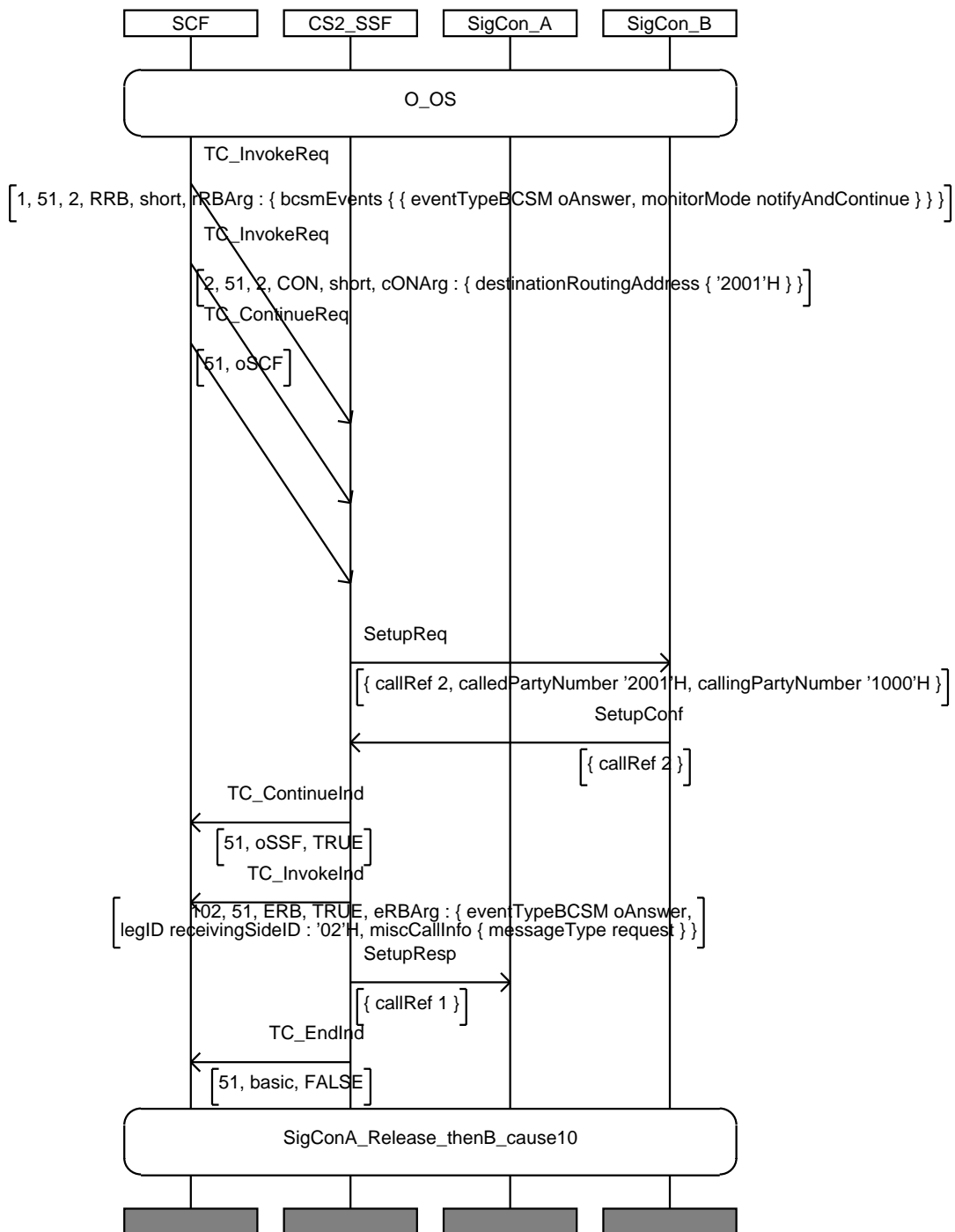
XXXX		IN2_A_BASIC_RR_BV_24	
Purpose:	Test of RequestReportBCSMEvent procedure and oNoAnswer indication.		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM=oNoAnswer - monitoringMode= notifyAndContinue <p>followed by a Connect invoke</p> <p>Then SSF sends a SetupReq to SigCon B</p> <p>SigCon B releases the call (ReleaseInd) because user B does not answer</p>		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oNoAnswer		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_24



XXXX		IN2_A_BASIC_RR_BV_25	
Purpose:	Test of RequestReportBCSMEvent procedure and oAnswer indication.		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM=oAnswer - monitoringMode= notifyAndContinue followed by a Connect invoke Then SSF sends a SetupReq to SigCon B SigCon B answers the call (SetupConf from SigConB to SSF)		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oAnswer		
Postamble:	SigConA_Release-thenB_cause10		

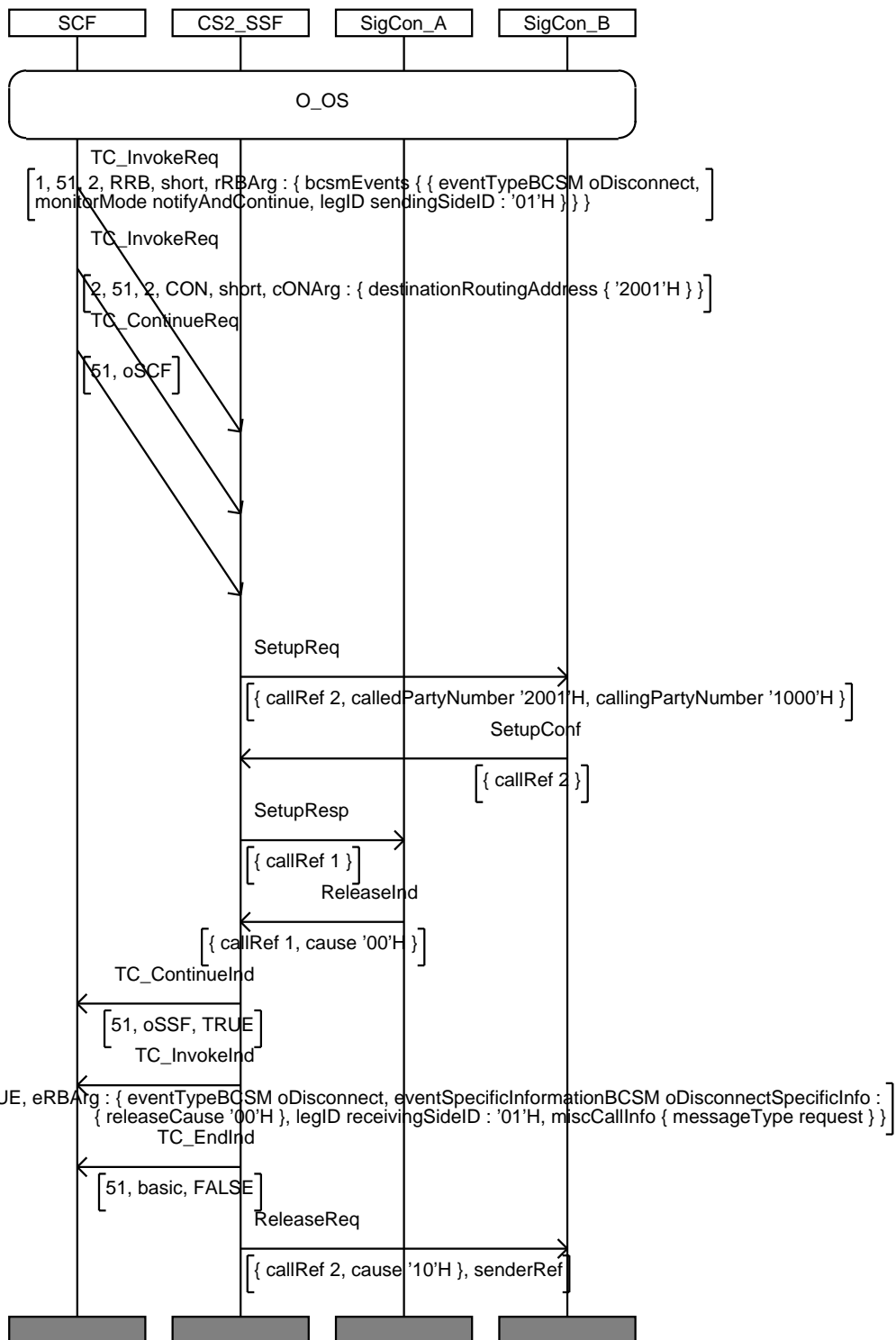
MSC IN2_A_BASIC_RR_BV_25



XXXX		IN2_A_BASIC_RR_BV_26	
Purpose:	Test of RequestReportBCSMEvent procedure and oMidCall indication.		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oMidCall - monitoringMode= notifyAndContinue <p>followed by a Connect invoke</p> <p>Then SSF sends a SetupReq to SigCon B. SetupConf from SigConB is received by SSF which issues SetupResp to SigConA.</p> <p>SigConA calling party initiates a service (ServiceFeatureInd sent to SSF) and oMidCall DP is reached</p>		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oMidCall		
Postamble:	SigConA_Release_thenB_cause10		

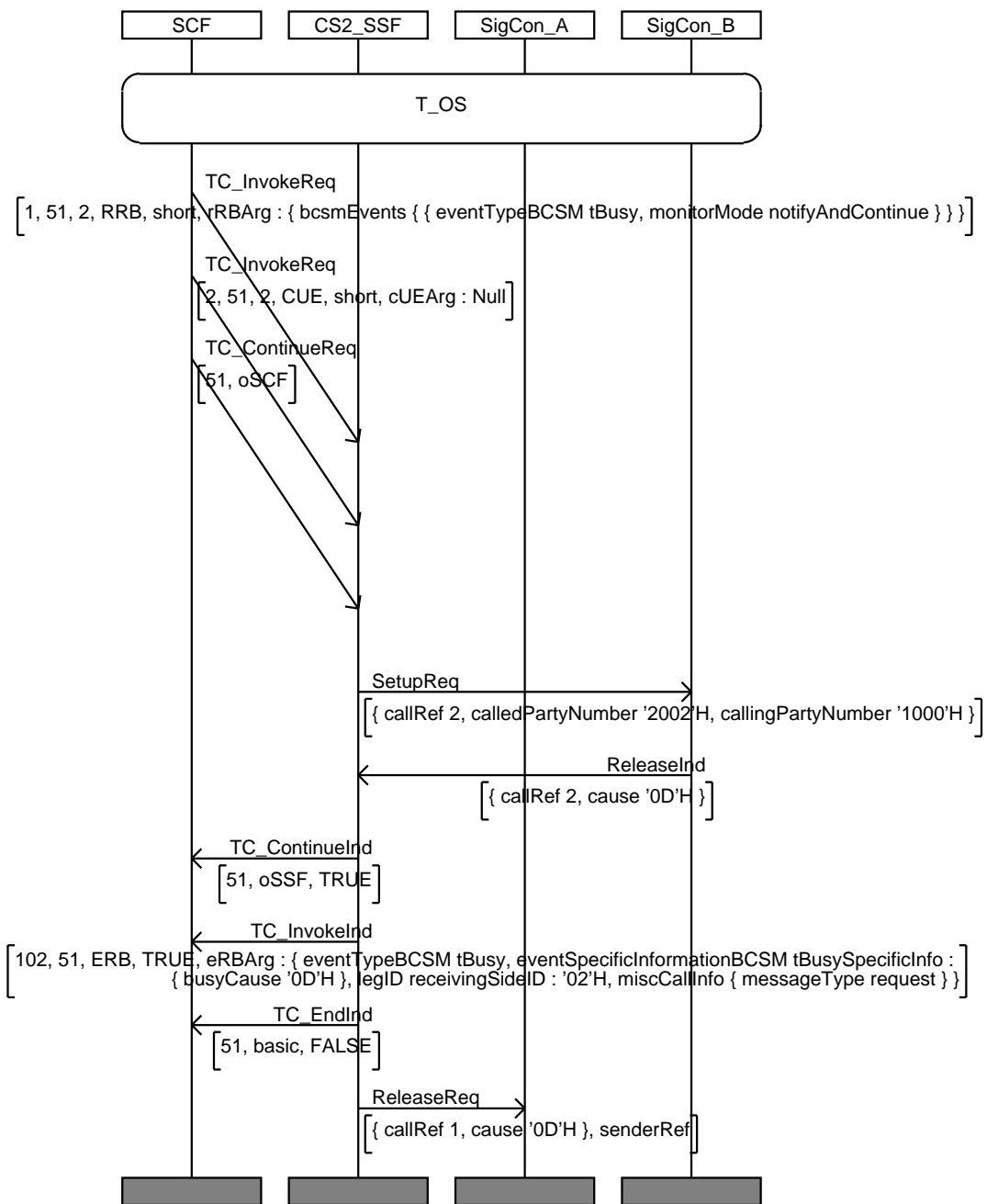
XXXX		IN2_A_BASIC_RR_BV_27	
Purpose:	Test of RequestReportBCSMEvent procedure and oDisconnect indication.		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oDisconnect - monitoringMode= notifyAndContinue - legID=sendingSideID: "01"H <p>followed by a Connect invoke</p> <p>Then SSF establishes the call (a SetupReq to SigCon B. SetupConf from SigConB to SSF, then SetupResp to SigConB)</p> <p>SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)</p>		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= oDisconnect		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_27



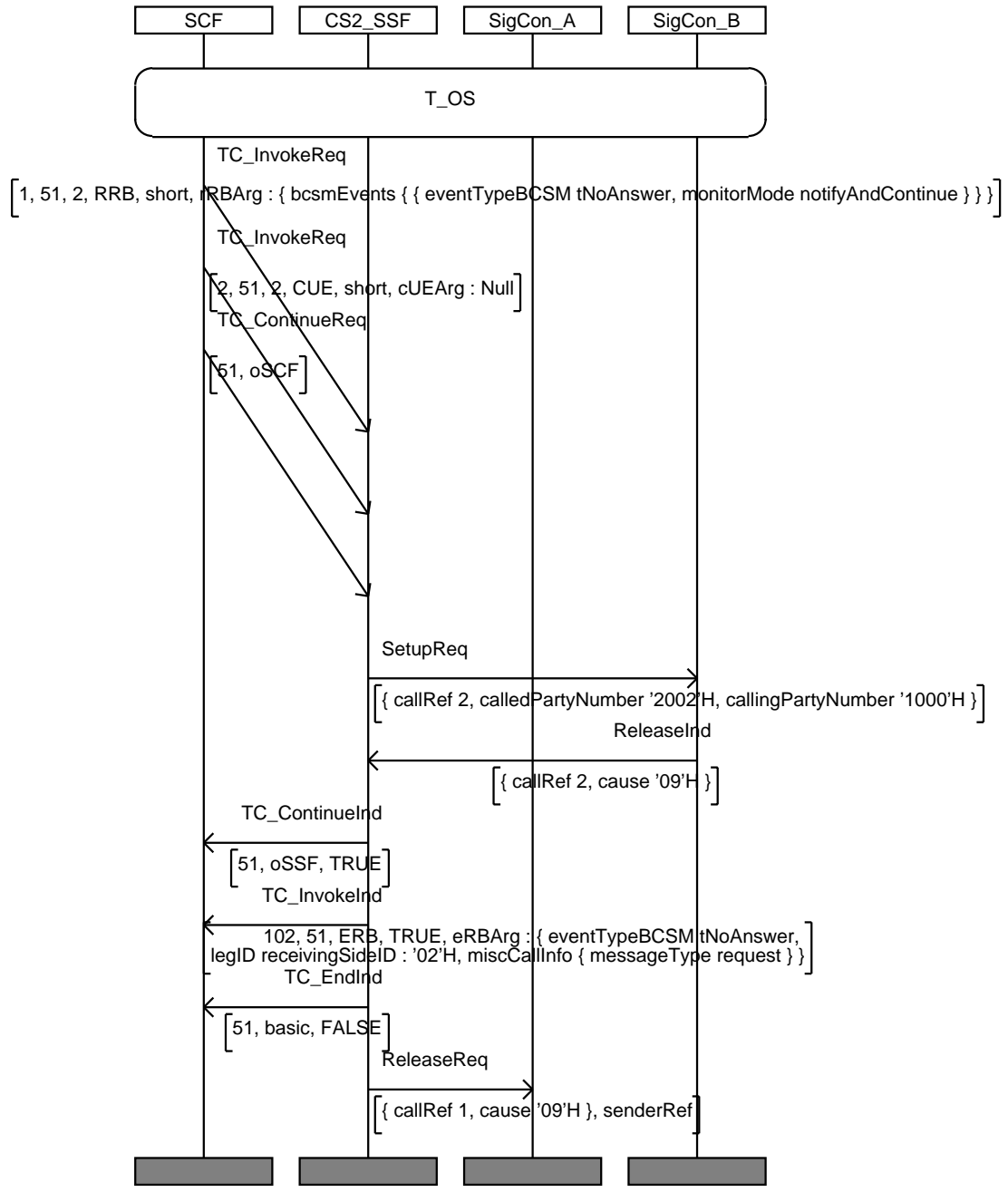
XXXX		IN2_A_BASIC_RR_BV_28	
Purpose:	Test of RequestReportBCSMEvent procedure and tBusy indication.		
Requirement ref			
Selection Cond.			
Preamble:	T_OS In addition, user B is declared busy		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=tBusy - monitoringMode= notifyAndContinue followed by a Continue invoke Then SSF sends a SetupReq to SigCon B SigCon B releases the call (ReleaseInd sent) because user B is busy (UDUB="0D"H)		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tBusy		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_28



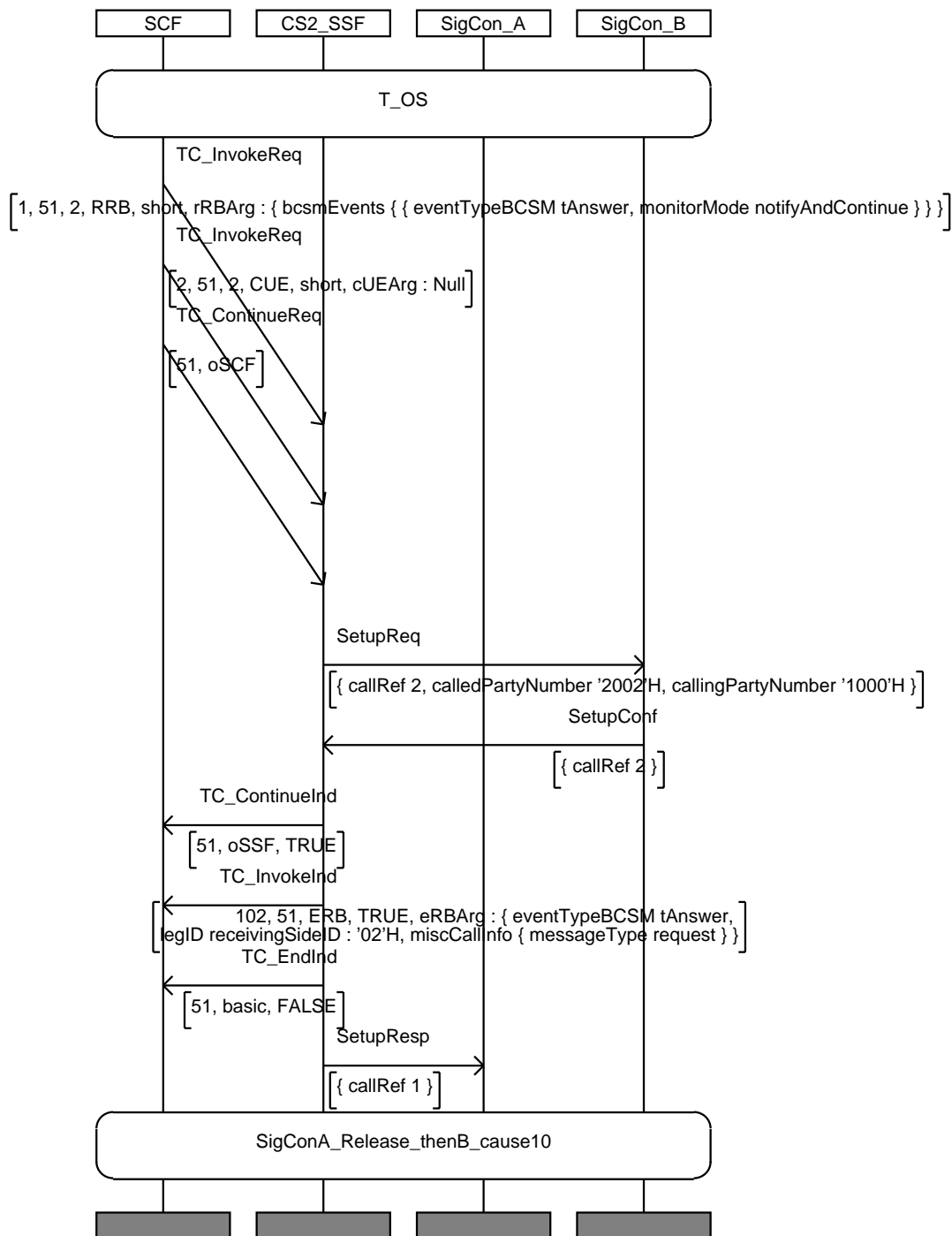
XXXX		IN2_A_BASIC_RR_BV_29	
Purpose:	Test of RequestReportBCSMEvent procedure and tNoAnswer indication.		
Requirement ref			
Selection Cond.			
Preamble:	T_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM=tNoAnswer - monitoringMode= notifyAndContinue <p>followed by a Continue invoke</p> <p>Then SSF sends a SetupReq to SigCon B</p> <p>SigCon B releases the call (ReleaseInd sent) because user B does not answer</p>		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tNoAnswer		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_29



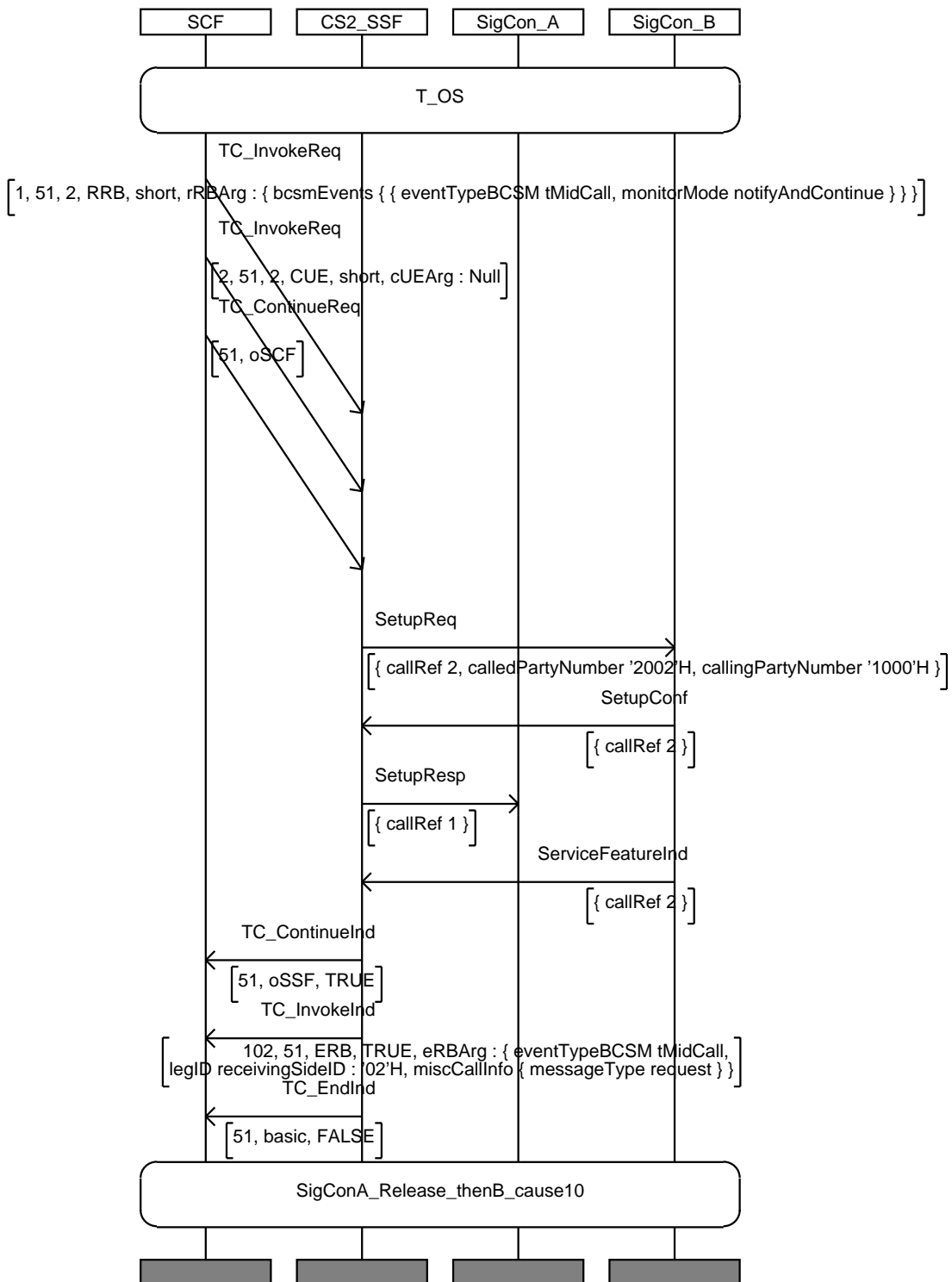
XXXX		IN2_A_BASIC_RR_BV_30	
Purpose:	Test of RequestReportBCSMEvent procedure and tAnswer indication.		
Requirement ref			
Selection Cond.			
Preamble:	T_OS		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM=tAnswer - monitoringMode= notifyAndContinue followed by a Continue invoke Then SSF sends a SetupReq to SigCon B SigCon B answers the call (SetupConf from SigConB to SSF)		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tAnswer		
Postamble:	SigConA_Release_thenB_cause10		

MSC IN2_A_BASIC_RR_BV_30

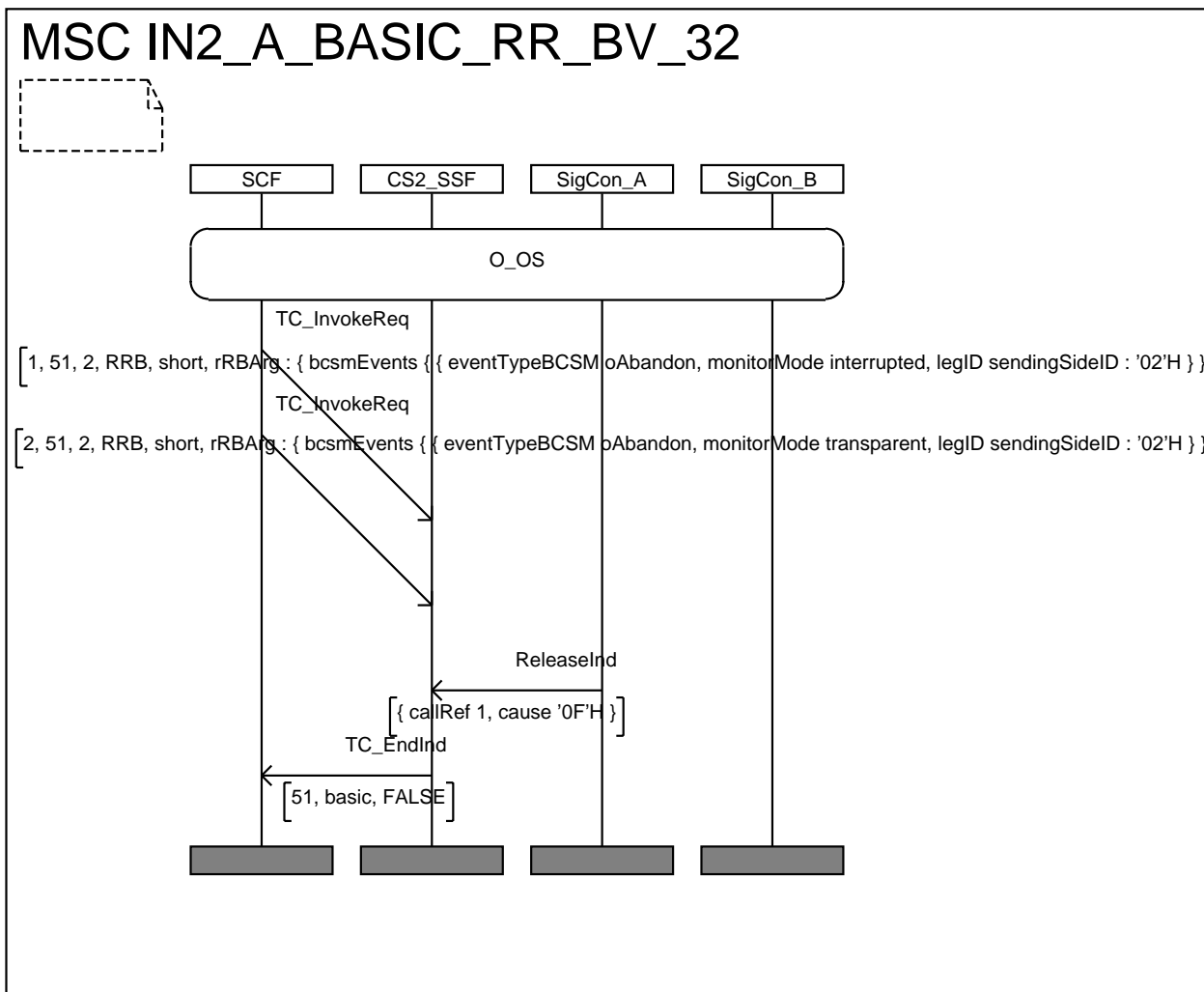


XXXX		IN2_A_BASIC_RR_BV_31	
Purpose:	Test of RequestReportBCSMEvent procedure and tMidCall indication.		
Requirement ref			
Selection Cond.			
Preamble:	T_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= tMidCall - monitoringMode= notifyAndContinue <p>followed by a Continue invoke</p> <p>Then SSF sends a SetupReq to SigCon B. SetupConf from SigConB is received by SSF which issues SetupResp to SigConA.</p> <p>SigConB called party initiates a service (ServiceFeatureInd sent to SSF) and tMidCall DP is reached</p>		
Pass criteria	Check that SSF sends to SCF an EventReportBCSM with the indication of eventTypeBCSM= tMidCall		
Postamble:	SigConA_Release_thenB_cause10		

MSC IN2_A_BASIC_RR_BV_31

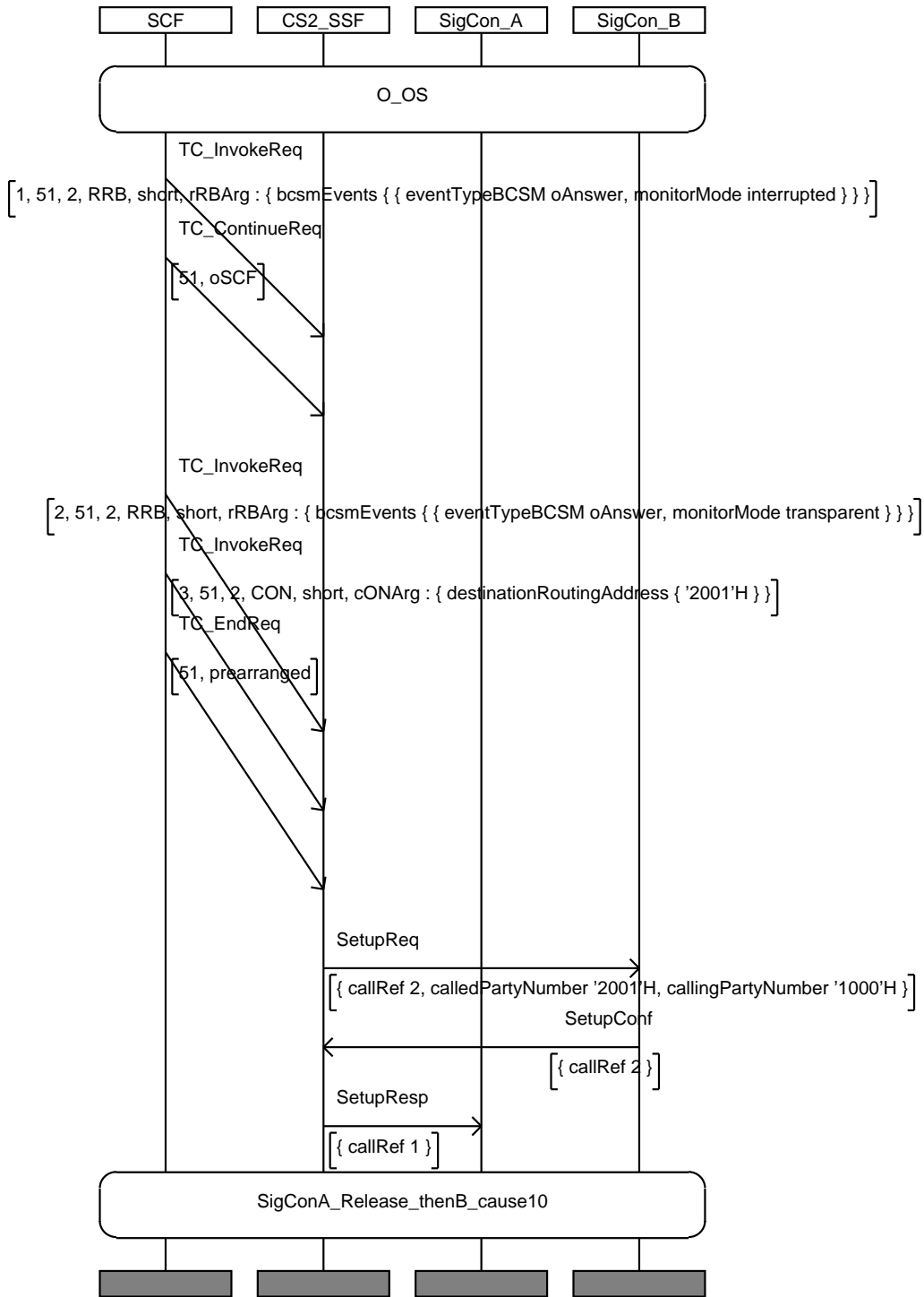


XXXX		IN2_A_BASIC_RR_BV_32	
Purpose:	Test of RequestReportBCSMEvent procedure and oAbandon - transparent		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - - eventTypeBCSM=oAbandon - monitoringMode=interrupted then SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM=oAbandon - monitoringMode=transparent then the calling party abandons the call before the call is answered (SigCon A to send ReleaseInd)		
Pass criteria	Check that SSF does not send to SCF an EventReportBCSM		
Postamble:	none		



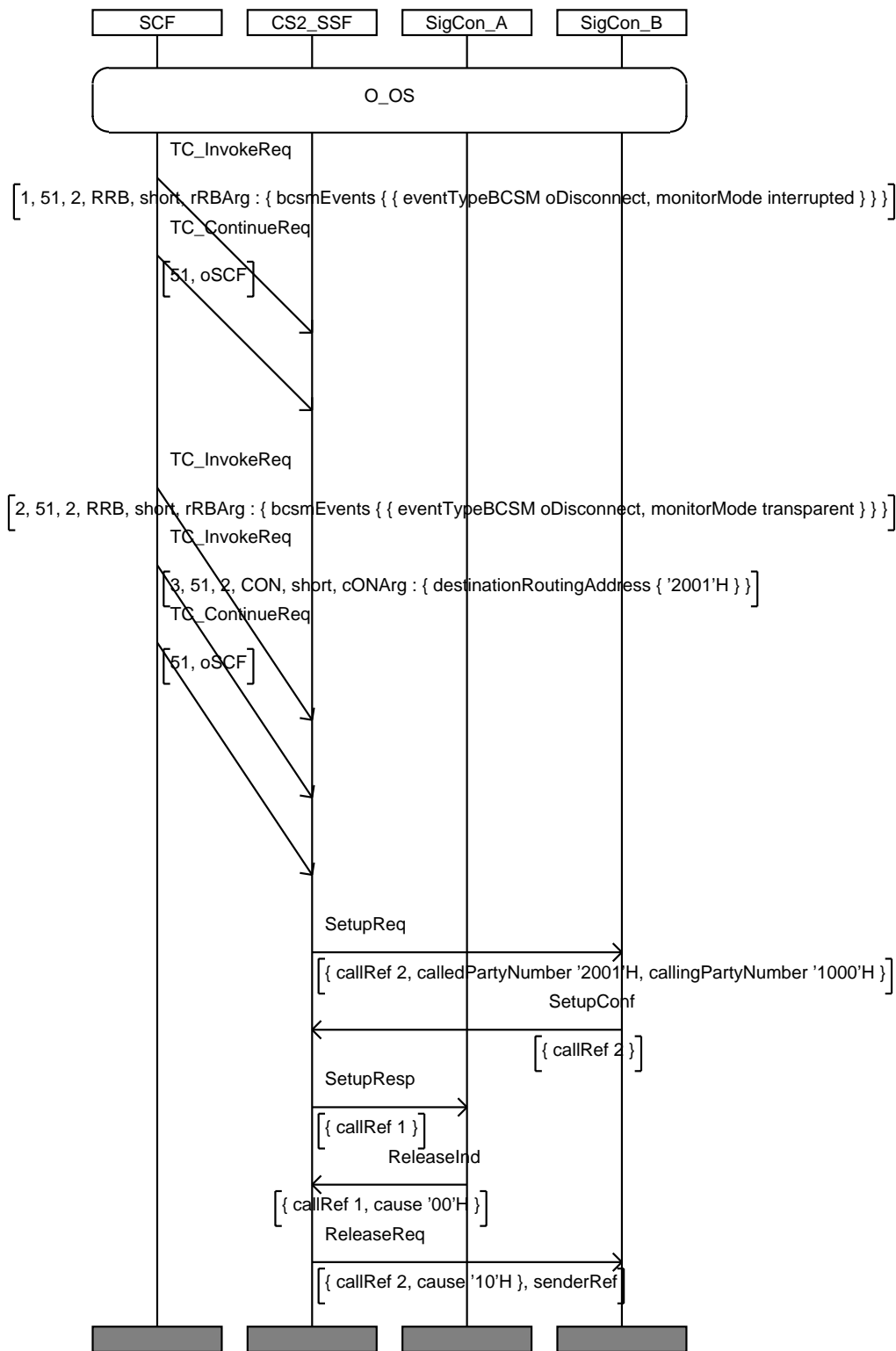
XXXX		IN2_A_BASIC_RR_BV_33	
Purpose:	Test of RequestReportBCSMEvent procedure and oAnswer - transparent		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - - eventTypeBCSM=oAnswer - monitoringMode=interrupted <p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM=oAnswer - monitoringMode=transparent <p>followed by a Connect invoke then SSF sends a SetupReq to SigCon B SigCon B answers the call (SetupConf from SigCon B to SSF)</p>		
Pass criteria	Check that SSF does not send to SCF an EventReportBCSM		
Postamble:	SigConA_Release_thenB		

MSC IN2_A_BASIC_RR_BV_33



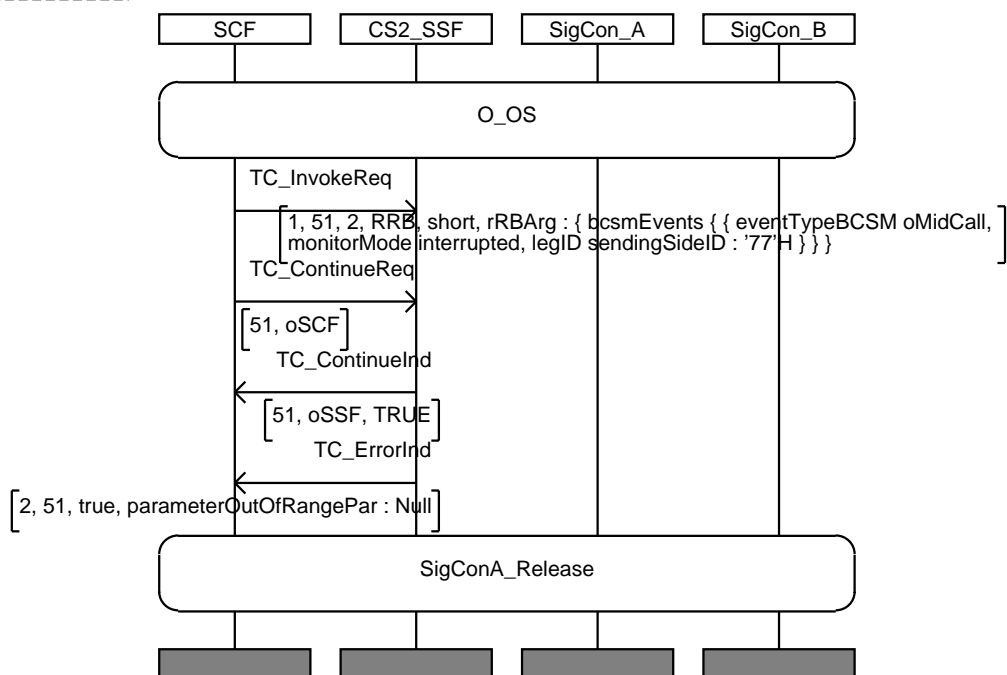
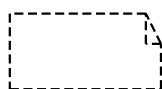
XXXX		IN2_A_BASIC_RR_BV_34	
Purpose:	Test of RequestReportBCSMEvent procedure and oDisconnect indication - transparent		
Requirement ref			
Selection Cond.			
Preamble:	O_OS		
Test description	<p>SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oDisconnect - monitoringMode= interrupted <p>then SCF sends to SSF RequestReportBCSMEvent invoke containing parameters</p> <ul style="list-style-type: none"> - eventTypeBCSM= oDisconnect - monitoringMode= transparent <p>followed by a Connect invoke</p> <p>Then SSF establishes the call (a SetupReq to SigCon B. SetupConf from SigConB to SSF, then SetupResp to SigConB)</p> <p>SigCon A (calling party) clears the call after it is answered (ReleaseInd sent)</p>		
Pass criteria	Check that SSF does not send to SCF an EventReportBCSM		
Postamble:	none		

MSC IN2_A_BASIC_RR_BV_34



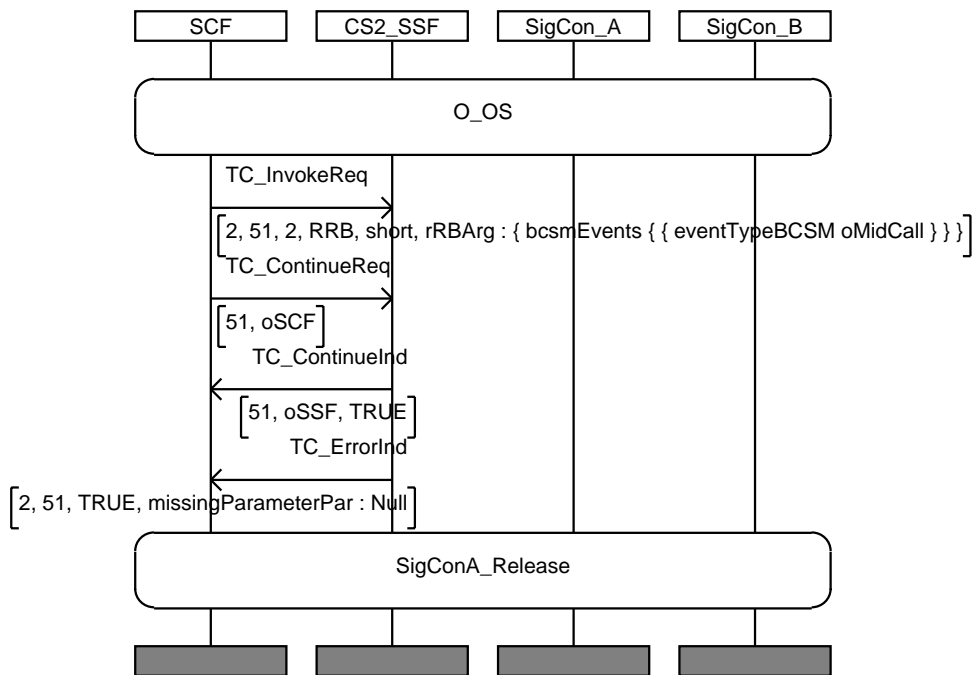
IN2_A_BASIC_RR_BI_01	
Purpose:	Test of RequestReportBCSMEvent procedure and out of range parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF - SCF sends to SSF RequestReportBCSMEvent invoke containing parameters <ul style="list-style-type: none"> - eventTypeBCSM= oMidCall - monitoringMode=interrupted - legID=invalid value
Pass criteria	- Check that SSF sends to SCF a RequestReportBCSMEvent error with the indication of out of range parameter - When call Set-up is established, check that SSF is not sending to SCF any EventReportBCSM
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_RR_BI_01

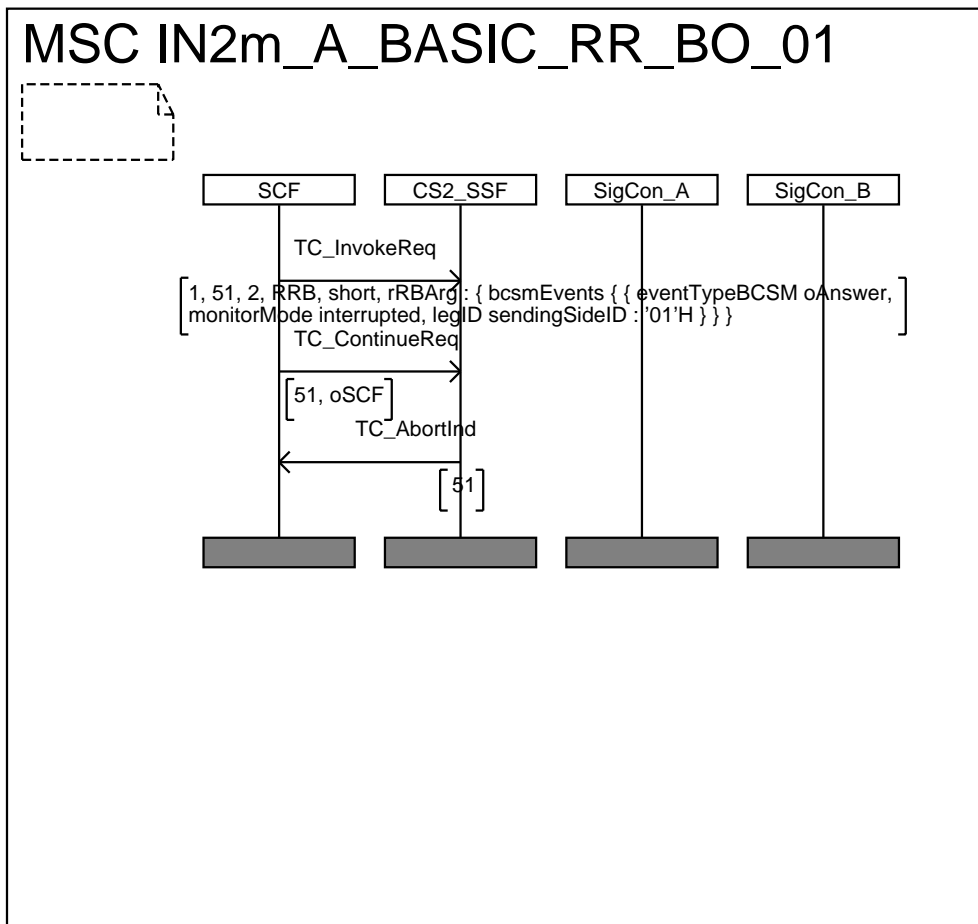


IN2_A_BASIC_RR_BI_02	
Purpose:	Test of RequestReportBCSMEvent procedure and missing parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF - SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM= oMidCall - monitoringMode=none
Pass criteria	- Check that SSF rejects the RequestReportBCSMEvent
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_RR_BI_02



IN2_A_BASIC_RR_BO_01	
Purpose:	Test of RequestReportBCSMEvent procedure in wrong state
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF - SCF sends to SSF RequestReportBCSMEvent invoke containing parameters - eventTypeBCSM= oAnswer - monitoringMode=interrupted
Pass criteria	- Check that SSF sends a TC-ABORT
Postamble:	none



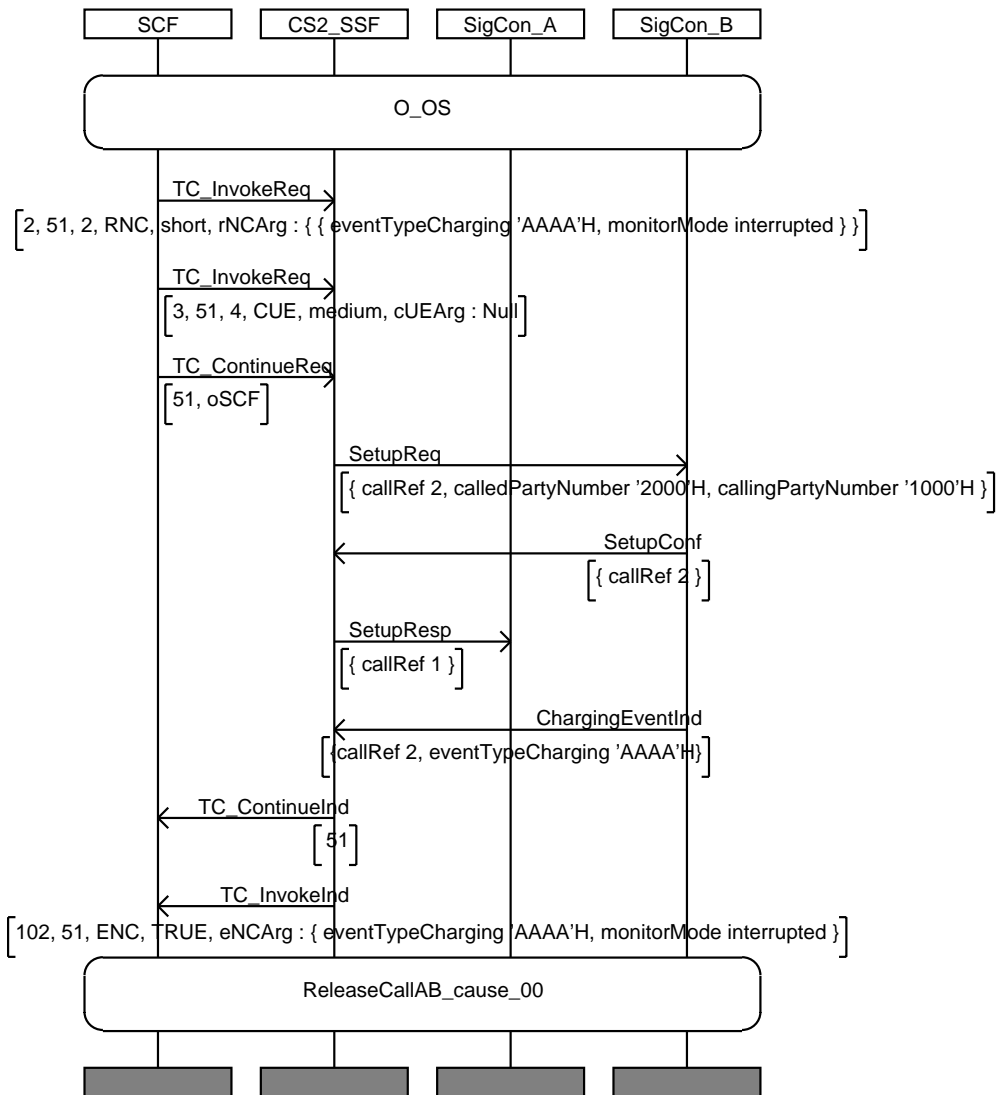
6.4.15 SendChargingInformation procedure

Charging related aspects in IN are network operator specific. Therefore, it is not possible to define useful test purposes for charging procedures using a network operator independant approach. TP specification has to be done by network operators, using INAP procedures themselves. SendChargingInformation TP could be specified in combination with ApplyCharging and FurnishChargingInformation procedures.

6.4.16 RequestNotificationChargingEvent procedure

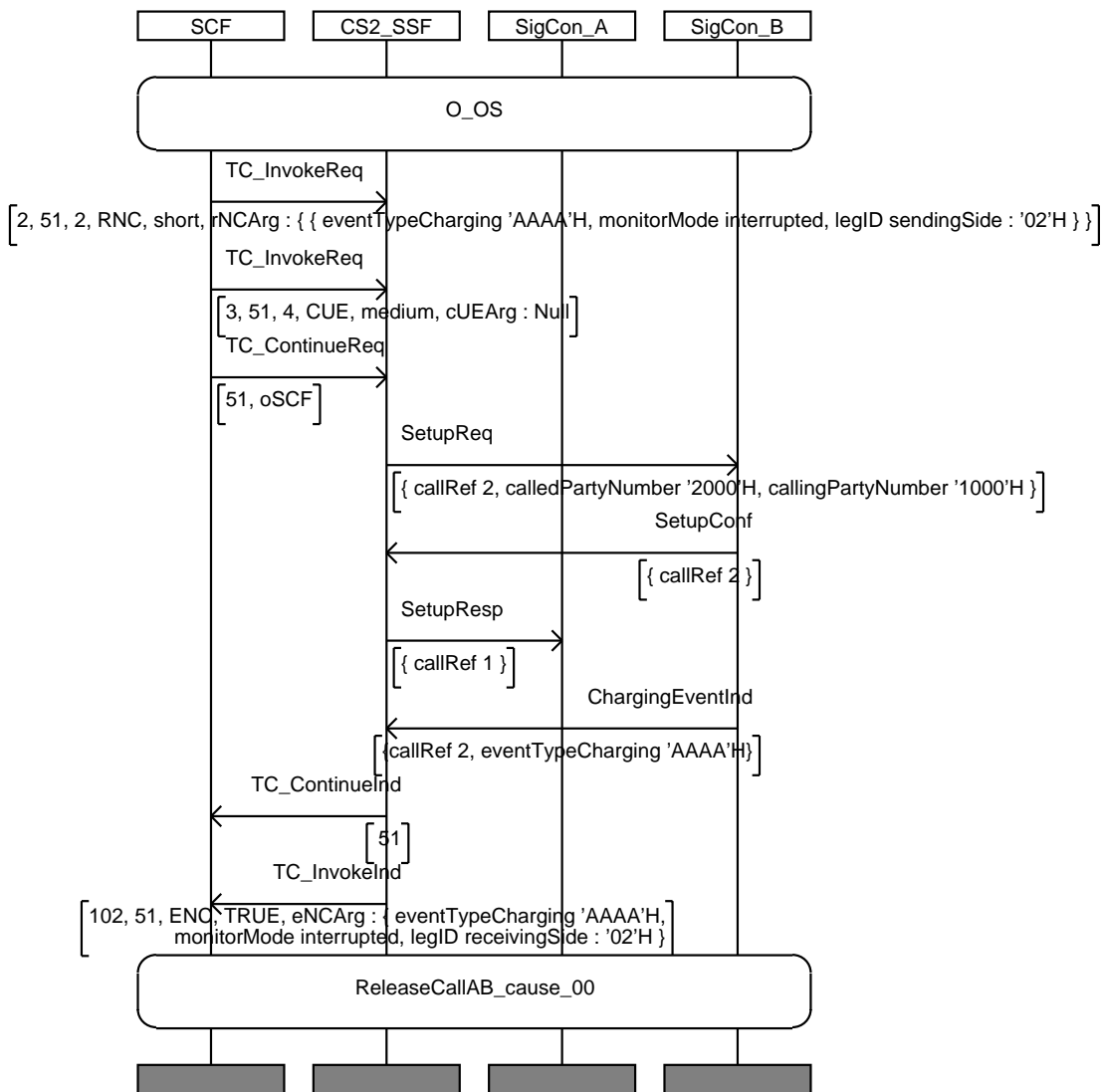
IN2_A_BASIC_RN_CA_01	
Purpose:	Test of RequestNotificationChargingEvent base procedure
Requirement ref	
Preamble:	O_OS
Selection Cond.	
Test description	SCF sends to SSF RequestNotificationChargingEvent invoke containing mandatory parameters only, with: - ChargingEvent eventTypeCharging, monitorMode (interrupted)
Pass criteria	After triggering of charging event from SigConA, check that SSF sends to SCF an EventNotificationCharging with the indication of eventTypeCharging
Postamble:	ReleaseCallAB_cause_00

MSC IN2m_A_BASIC_RN_CA_01



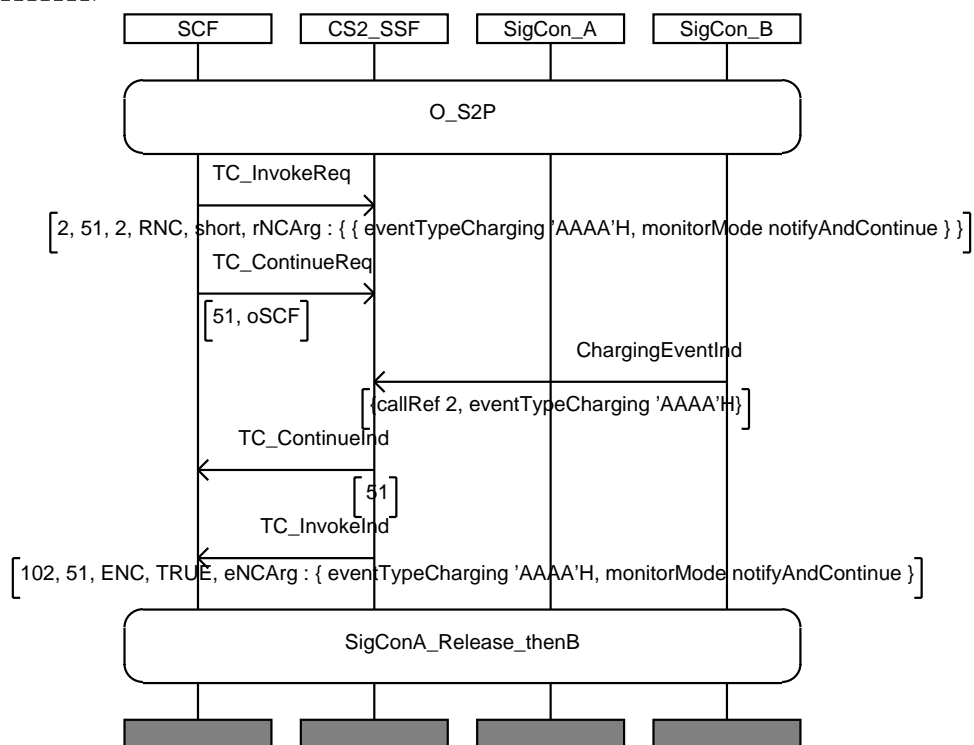
IN2_A_BASIC_RN_BV_01	
Purpose:	Test of RequestNotificationChargingEvent procedure with legID parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestNotificationChargingEvent invoke containing mandatory parameters only, with: - ChargingEvent eventTypeCharging, monitorMode (interrupted) legID being sendingSideID
Pass criteria	After triggering of charging event from SigConA, check that SSF sends to SCF an EventNotificationCharging with the indication of eventTypeCharging and legID being sendingSideID
Postamble:	ReleaseCallAB_cause_00

MSC IN2m_A_BASIC_RN_BV_01



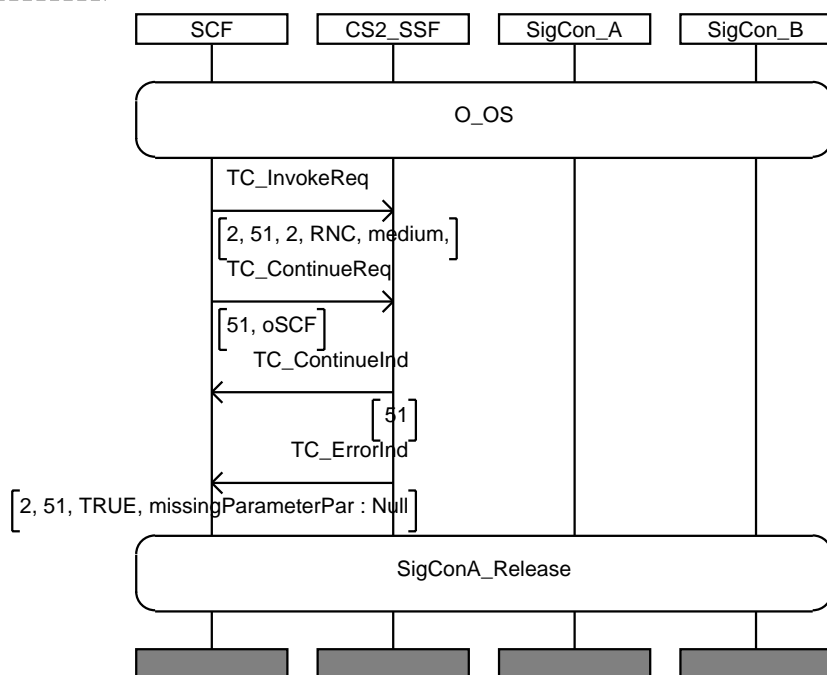
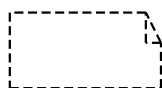
IN2_A_BASIC_RN_BV_02	
Purpose:	Test of RequestNotificationChargingEvent procedure with legID parameter
Requirement ref	
Selection Cond.	
Preamble:	O_S2P
Test description	SCF sends to SSF RequestNotificationChargingEvent invoke containing mandatory parameters only, with: - ChargingEvent eventTypeCharging, monitorMode (notifyAndContinue) legID being receivingSideID
Pass criteria	After triggering of charging event from SigConB, check that SSF sends to SCF an EventNotificationCharging with the indication of eventTypeCharging and legID being receivingSideID
Postamble:	SigConA_Release_thenB

MSC IN2m_A_BASIC_RN_BV_02



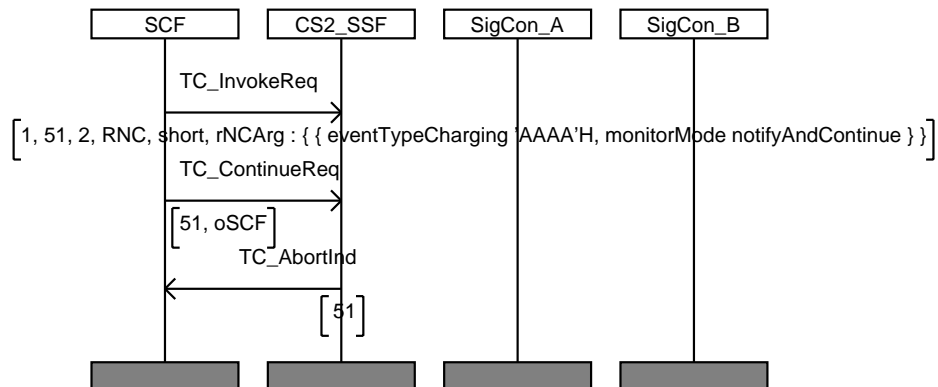
IN2_A_BASIC_RN_BI_01	
Purpose:	Test of RequestNotificationChargingEvent procedure with missing parameter
Requirement ref	
Selection Cond.	
Preamble:	O_OS
Test description	SCF sends to SSF RequestNotificationChargingEvent invoke without mandatory parameters
Pass criteria	check that SSF sends to SCF an RequestNotificationChargingEvent error with the indication of missingParameter
Postamble:	SigConA_Release

MSC IN2m_A_BASIC_RN_BI_01



IN2_A_BASIC_RN_BO_01	
Purpose:	Test of RequestNotificationChargingEvent procedure from wrong state
Requirement ref	
Selection Cond.	
Preamble:	none
Test description	SCF sends to SSF RequestNotificationChargingEvent invoke containing mandatory parameters only, with: - ChargingEvent eventTypeCharging, monitorMode (interrupted)
Pass criteria	Check that SSF sends to SCF a TC-ABORT
Postamble:	none

MSC IN2m_A_BASIC_RN_BO_02



Annex A (informative): Description of various functional configurations

In these various configurations, the shaded area represents the implementation under test (IUT).

Functional Configuration 1: Example for SCP with single SSP Non-Integrated or Integrated SRF

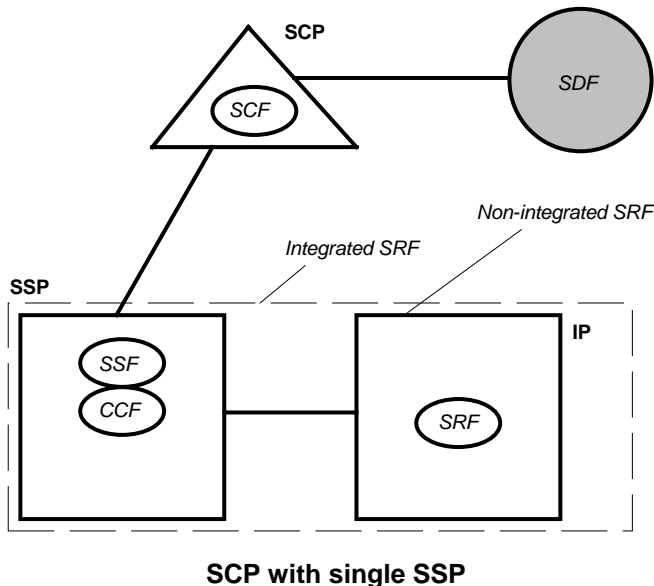


Figure A.1: Configuration 1_1: IUT= SDF

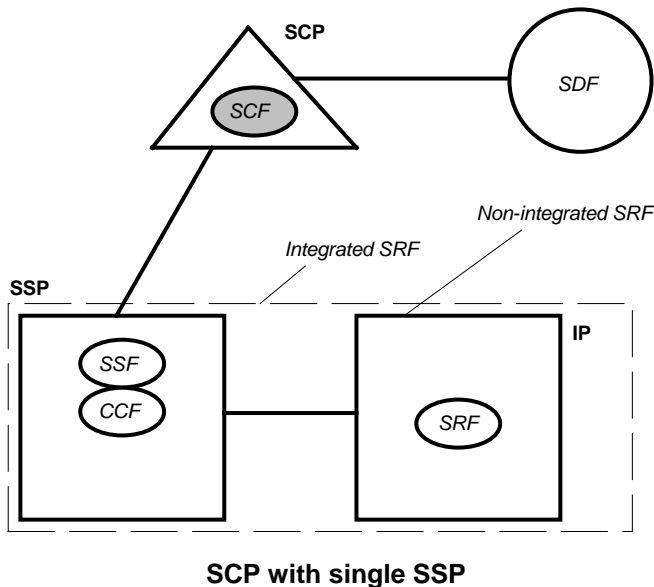


Figure A.2: Configuration 1_2: IUT= SCF

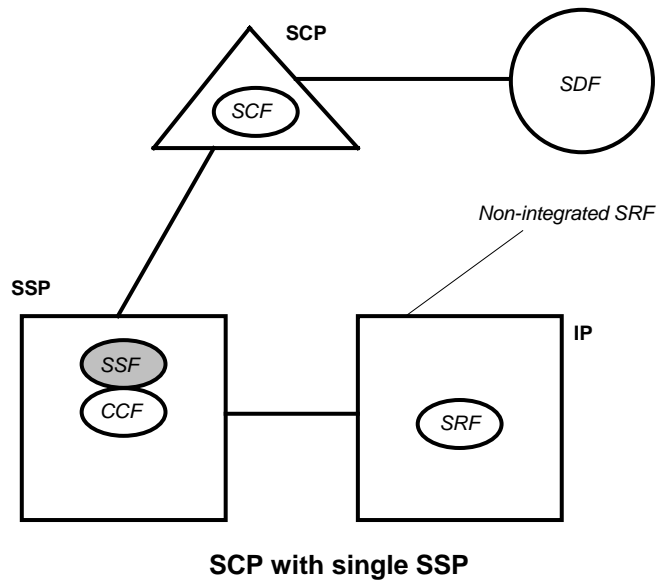


Figure A.3: Configuration 1_3: IUT= SSF (non integrated with SRF)

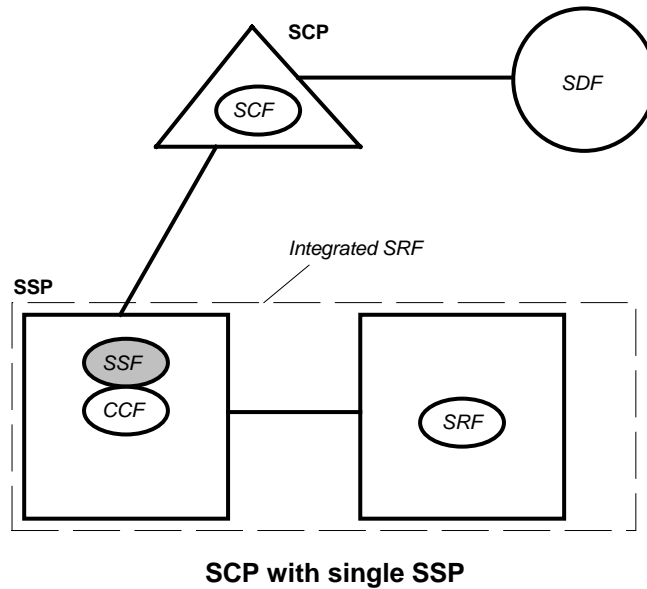


Figure A.4: Configuration 1_4: IUT= SSF (integrated with SRF)

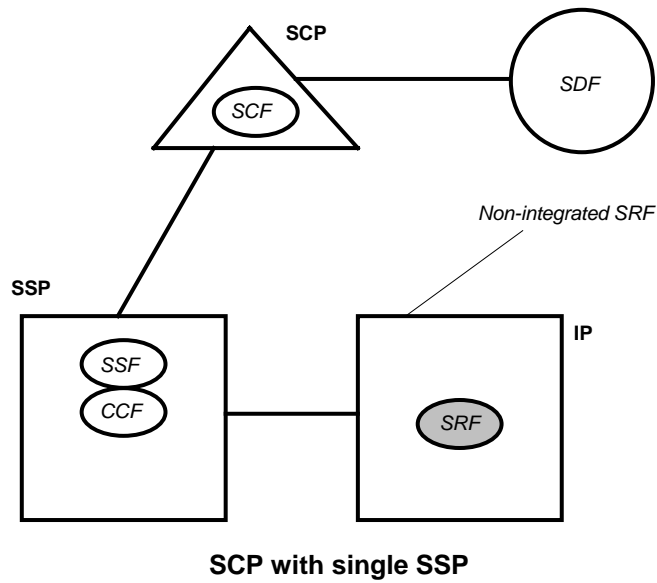


Figure A.5: Configuration 1_5: IUT= SRF (non integrated with SSF)

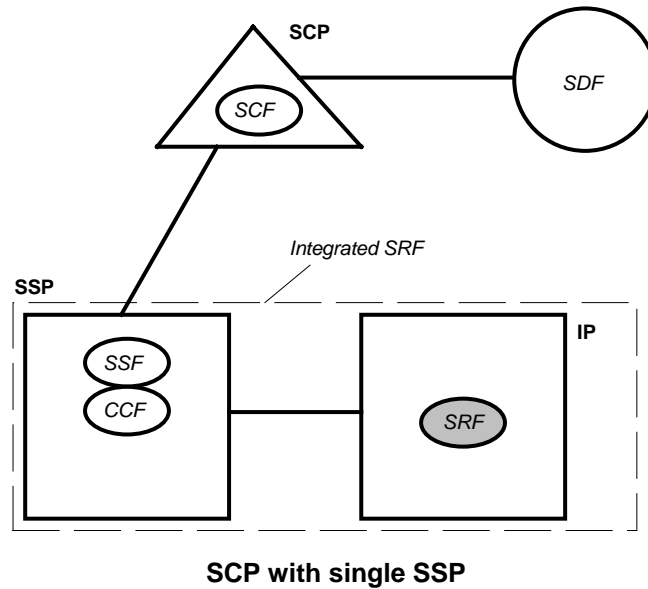
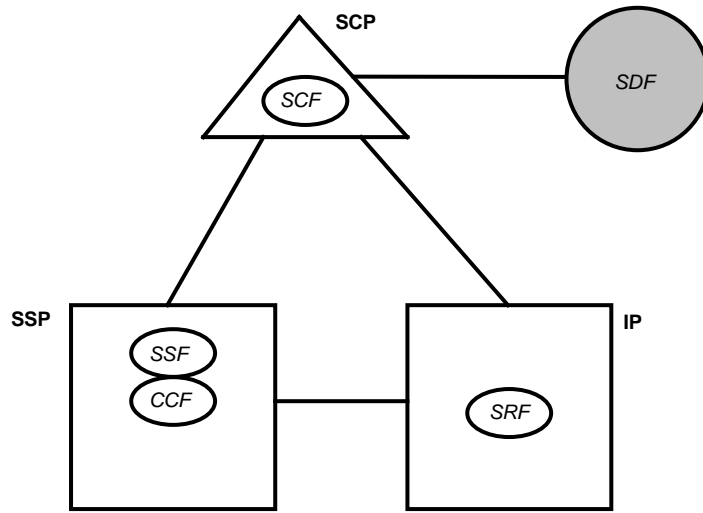


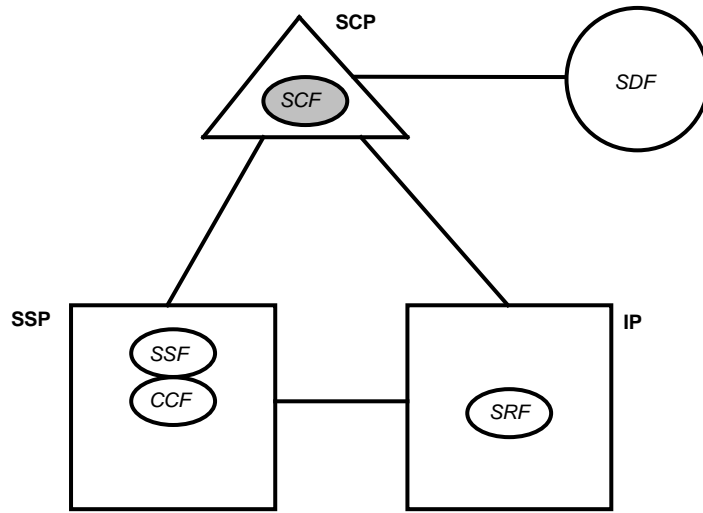
Figure A.6: Configuration 1_6: IUT= SRF (integrated with SSF)

Functional Configuration 2: Example for direct path SCP-IP



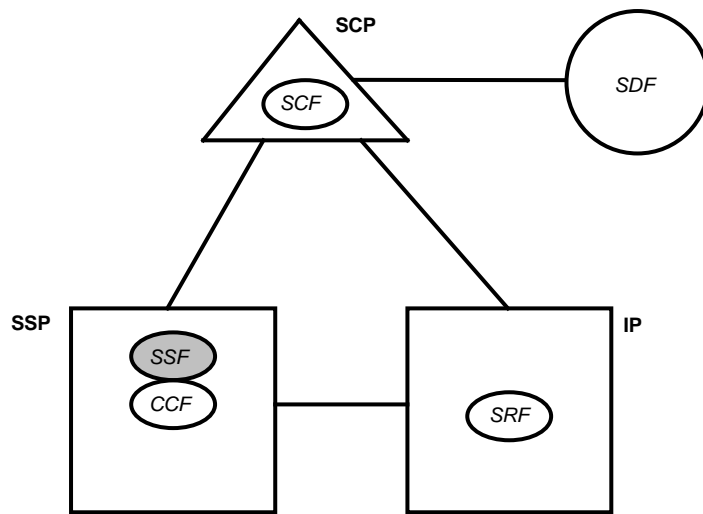
Direct path SCP - IP

Figure A.7: Configuration 2_1: IUT = SDF



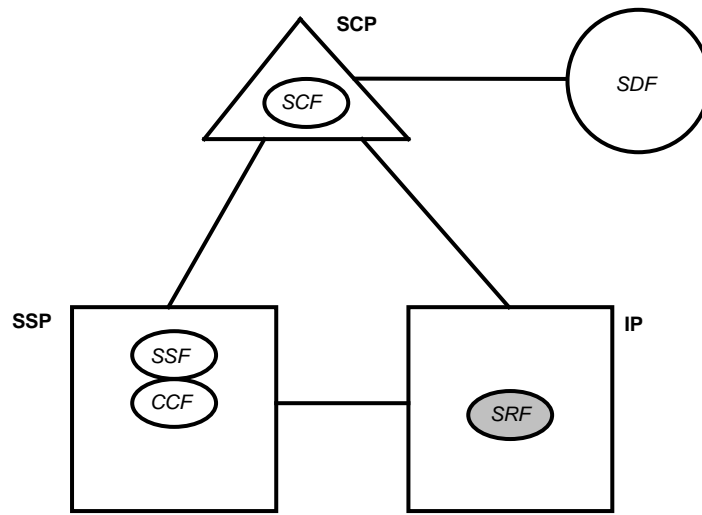
Direct path SCP - IP

Figure A.8: Configuration 2_2: IUT = SCF



Direct path SCP - IP

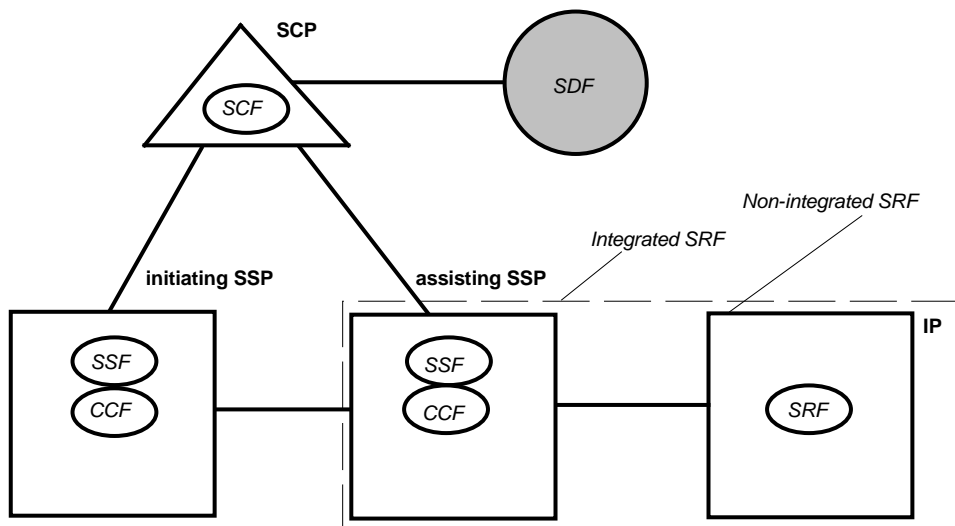
Figure A.9: Configuration 2_3: IUT = SSF



Direct path SCP - IP

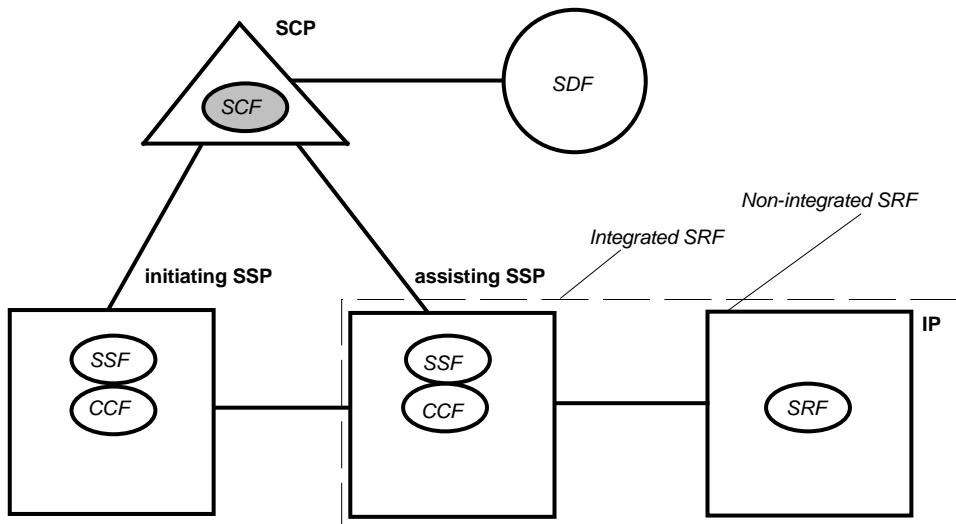
Figure A.10: Configuration 2_4: IUT = SRF

Functional Configuration 3: Example for SSP Assist/Hand-off (assisting SSP with relay)



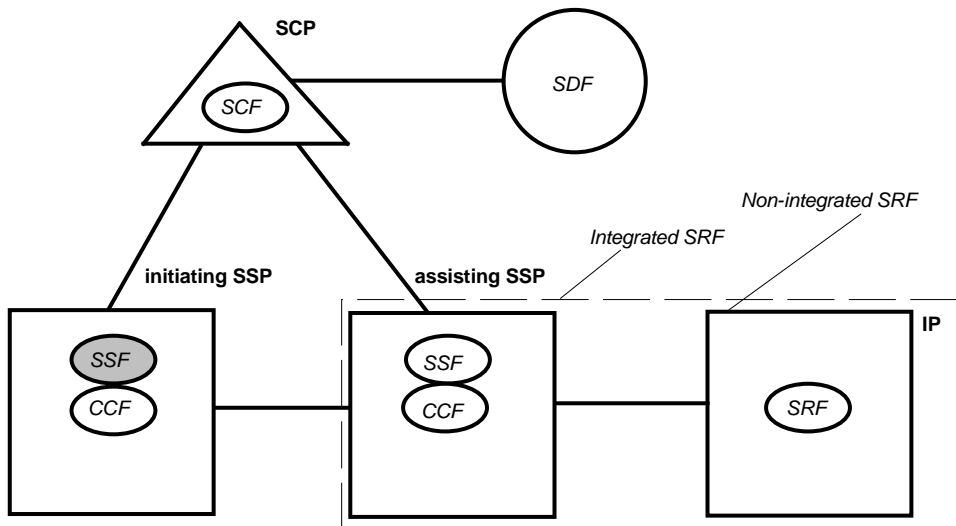
SSP Assist/Hand-off (assisting SSP with relay)

Figure A.11: Configuration 3_1: IUT= SDF



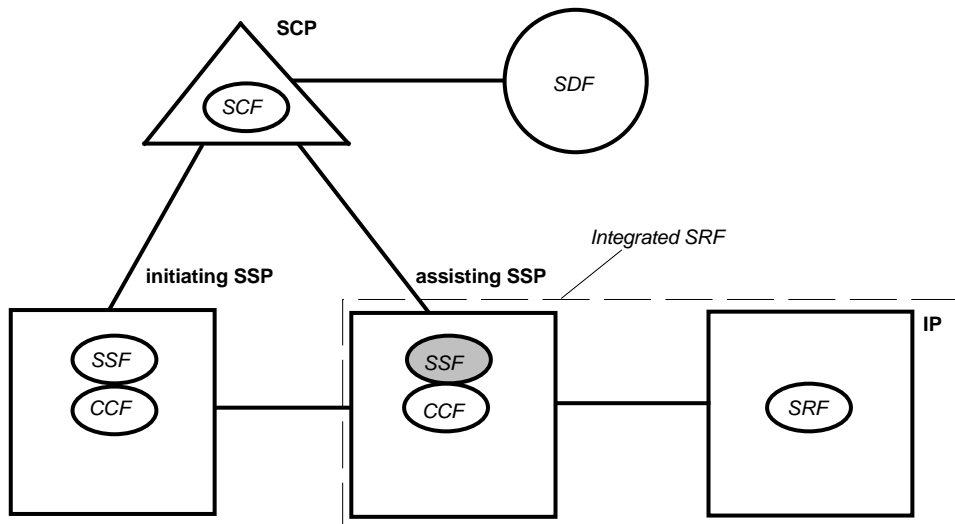
SSP Assist/Hand-off (assisting SSP with relay)

Figure A.12: Configuration 3_2: IUT= SCF



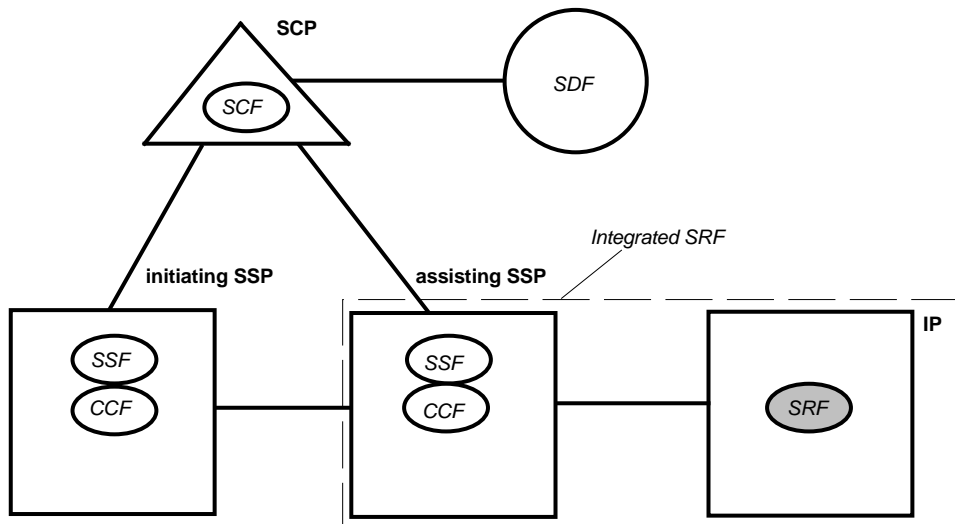
SSP Assist/Hand-off (assisting SSP with relay)

Figure A.13: Configuration 3_3: IUT= SSF of initiating SSP



SSP Assist/Hand-off (assisting SSP with relay)

Figure A.14: Configuration 3_4: IUT= SSF of assisting SSP(integrated SRF)



SSP Assist/Hand-off (assisting SSP with relay)

Figure A.15: Configuration 3_5: IUT= SRF (integrated with assisting SSF)

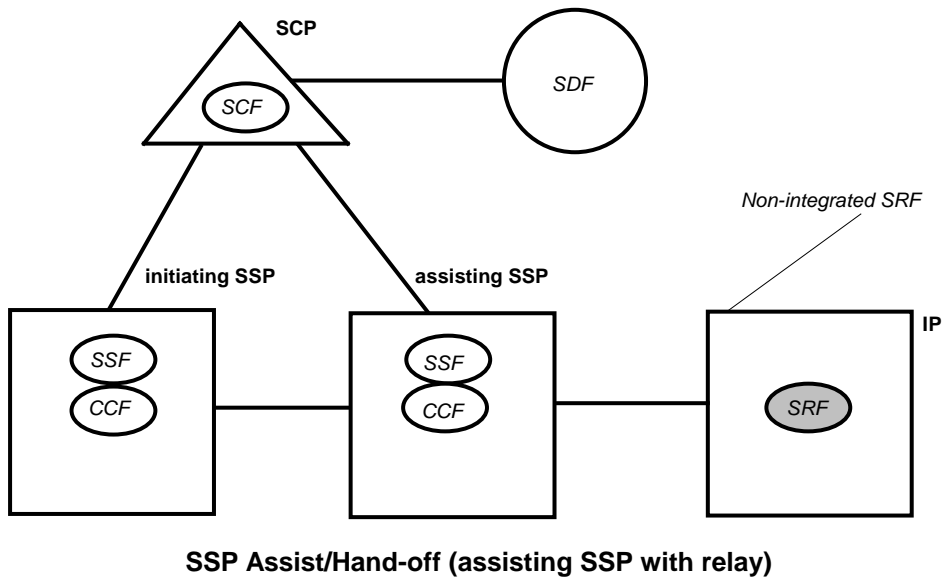


Figure A.16: Configuration 3_6: IUT= SRF (non integrated with assisting SSF)

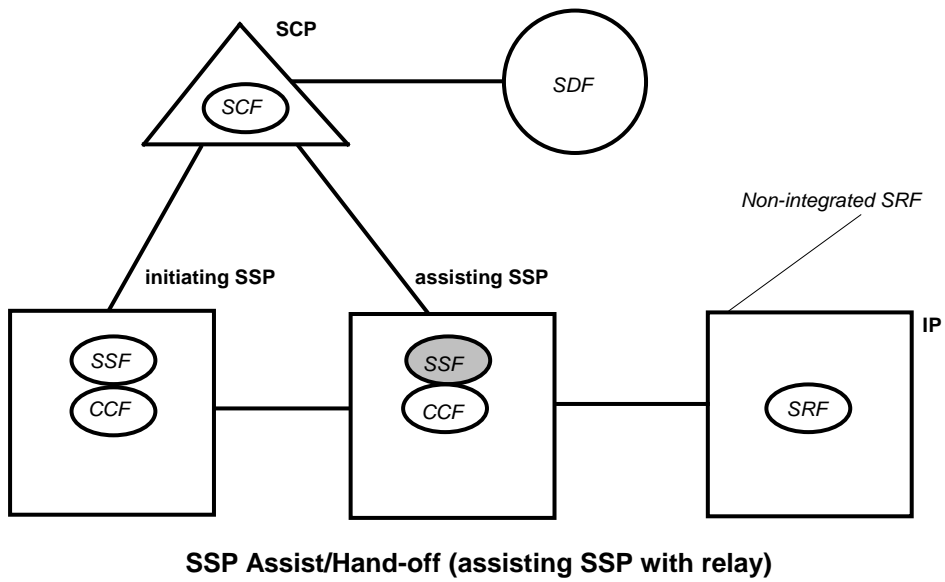
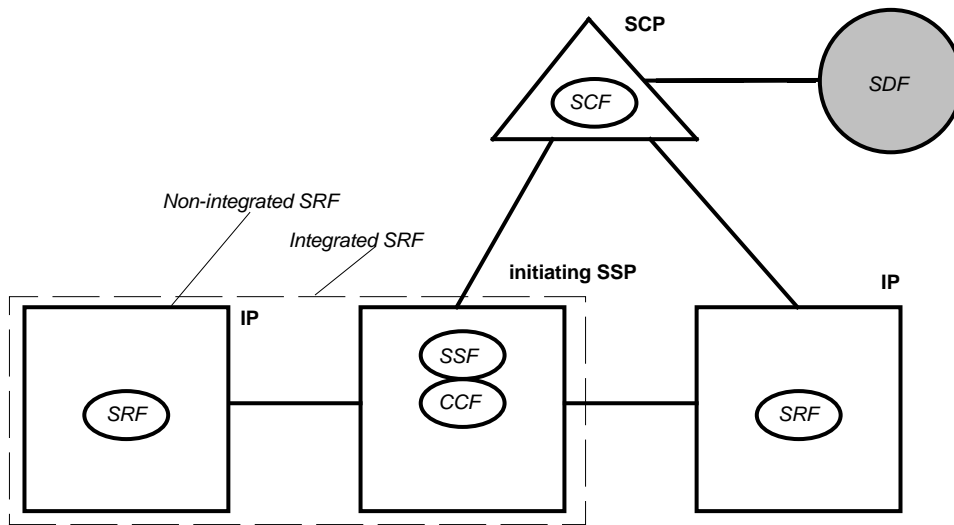


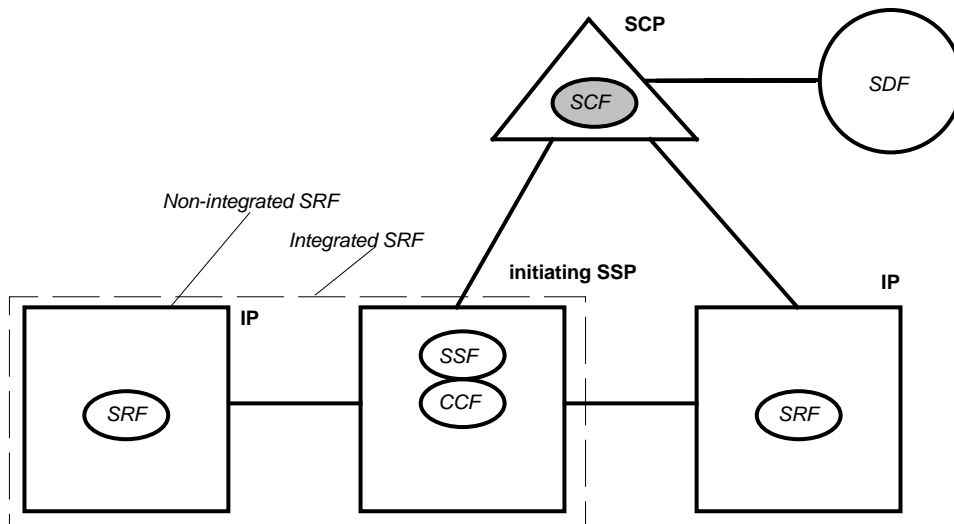
Figure A.17: Configuration 3_7: IUT= SSF of assisting SSP (non integrated SRF)

Functional Configuration 4: Example for SSP Assist/Hand-off (initiating SSP with relay)



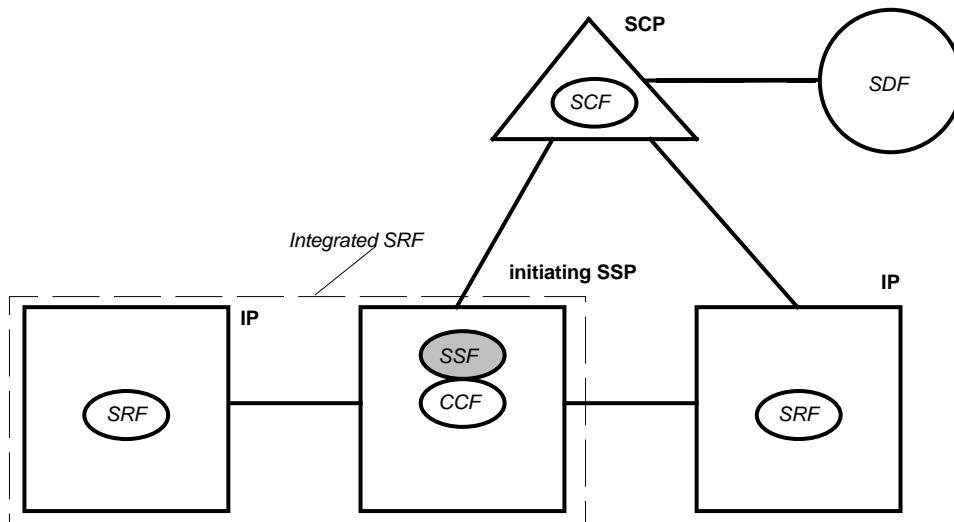
SSP Assist/Hand-off (initiating SSP with relay)

Figure A.18: Configuration 4_1: IUT= SDF



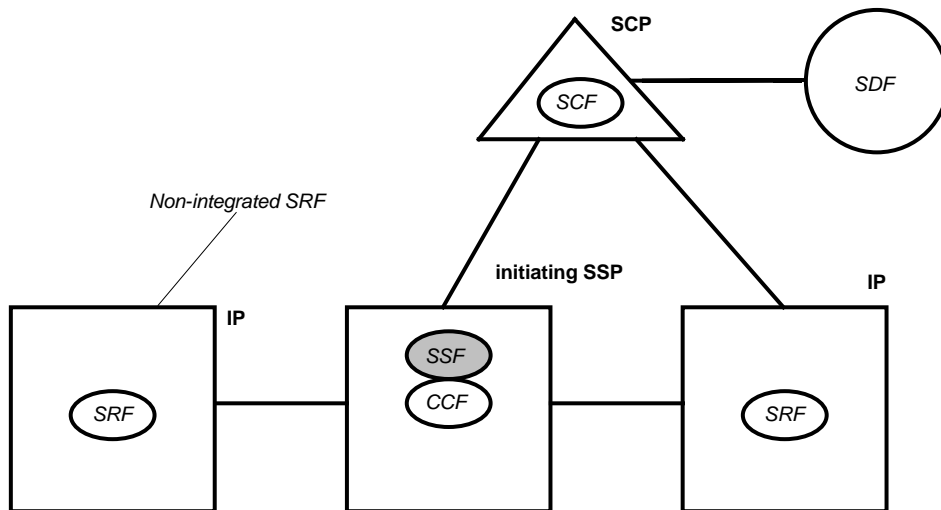
SSP Assist/Hand-off (initiating SSP with relay)

Figure A.19: Configuration 4_2: IUT= SCF



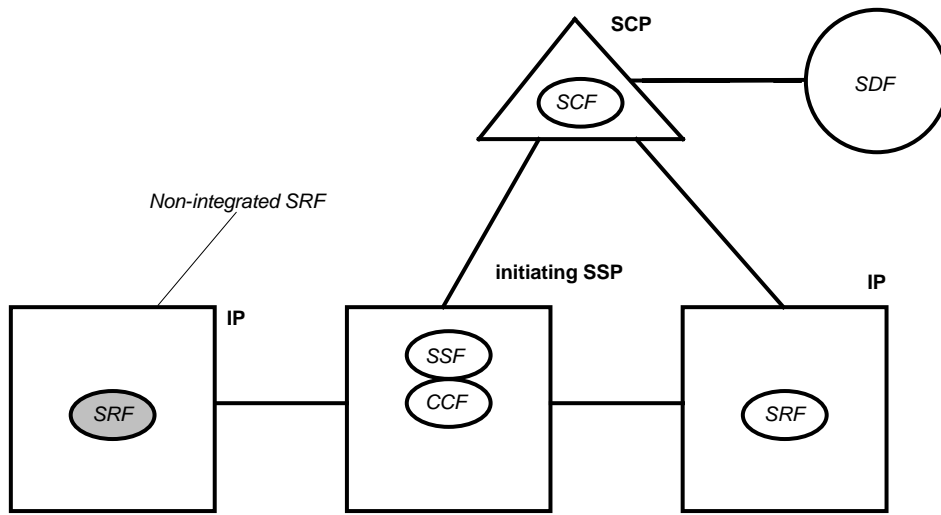
SSP Assist/Hand-off (initiating SSP with relay)

Figure A.20: Configuration 4_3: IUT= SSF of initiating SSP (integrated SRF)



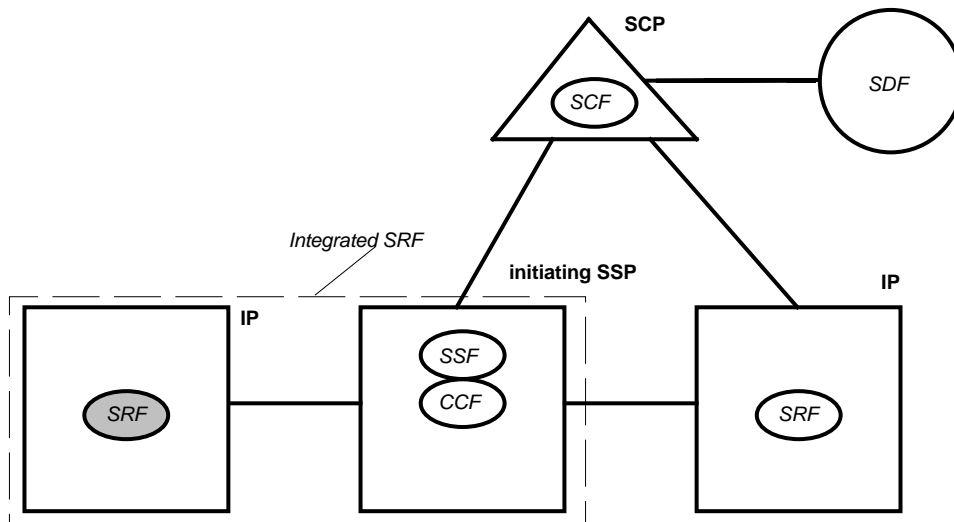
SSP Assist/Hand-off (initiating SSP with relay)

Figure A.21: Configuration 4_4: IUT= SSF of initiating SSP (non integrated SRF)



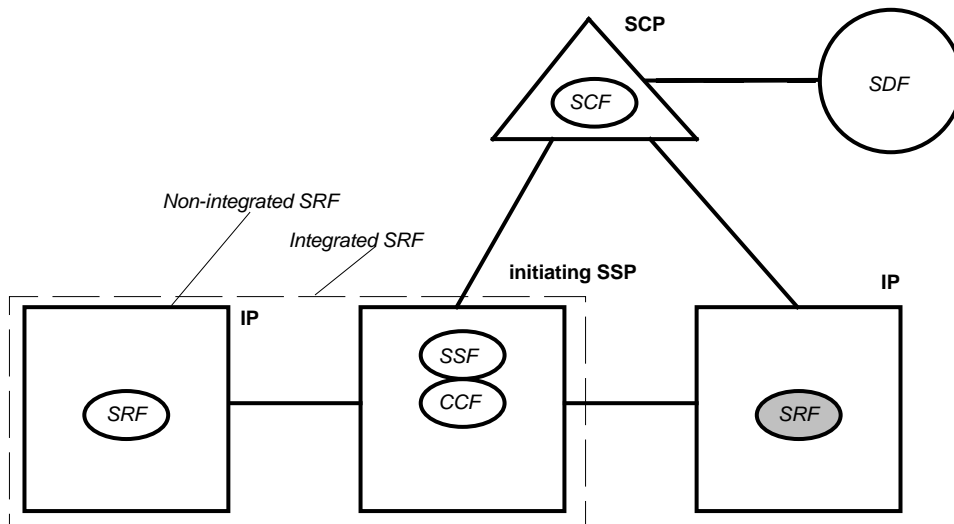
SSP Assist/Hand-off (initiating SSP with relay)

Figure A.22: Configuration 4_5: IUT= SRF of initiating SSP (non integrated SRF)



SSP Assist/Hand-off (initiating SSP with relay)

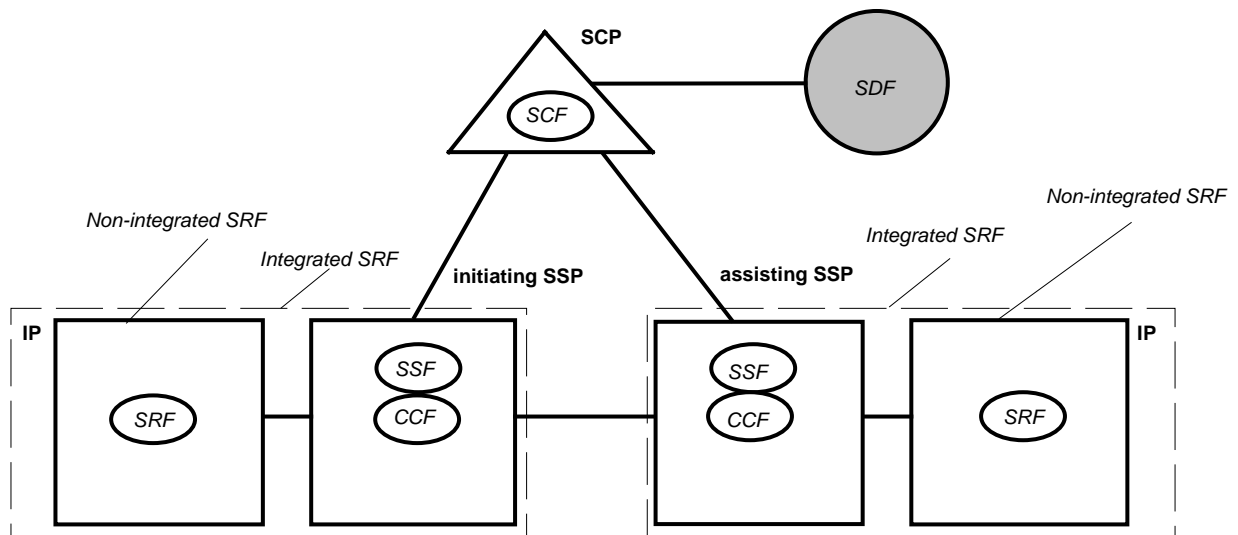
Figure A.23: Configuration 4_6: IUT= SRF of initiating SSP (integrated SRF)



SSP Assist/Hand-off (initiating SSP with relay)

Figure A.24: Configuration 4_7: IUT= SRF as IP

Functional Configuration: Example for SSP Assist/Hand-off (initiating and assisting SSP with relay)



SSP Assist/Hand-off (initiating and assisting SSP with relay)

Figure A.25: Configuration 5_1: IUT= SDF

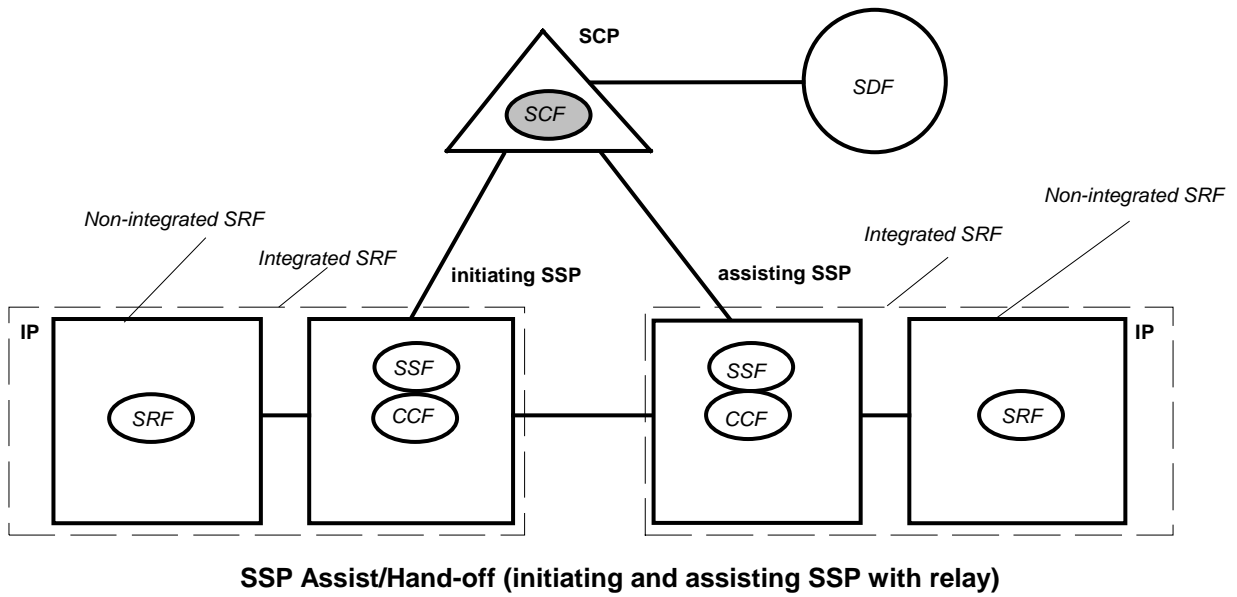


Figure A.26: Configuration 5_2: IUT= SCF

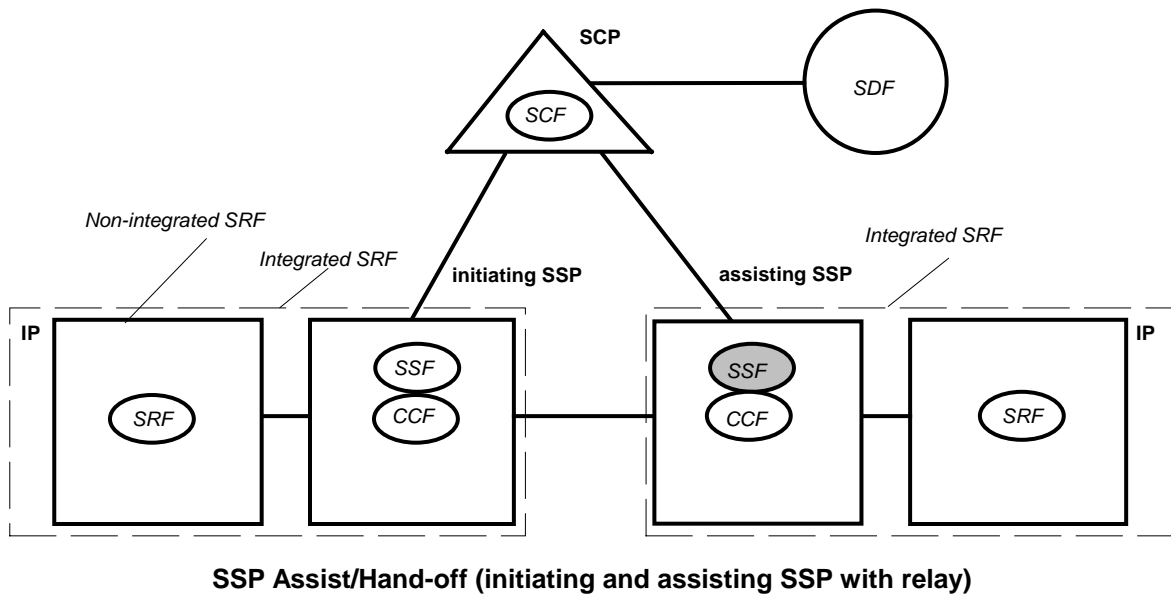


Figure A.27: Configuration 5_3: IUT= SSF of assisting SSP(with integrated SRF)

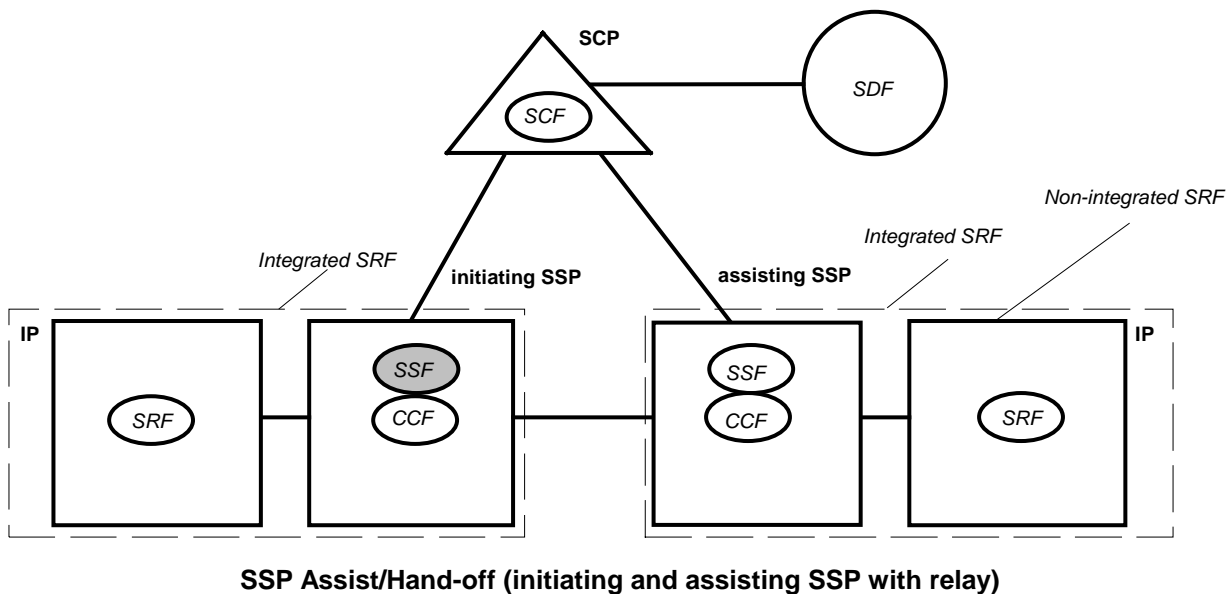


Figure A.28: Configuration 5_4: IUT= SSF of initiating SSP(with integrated SRF)

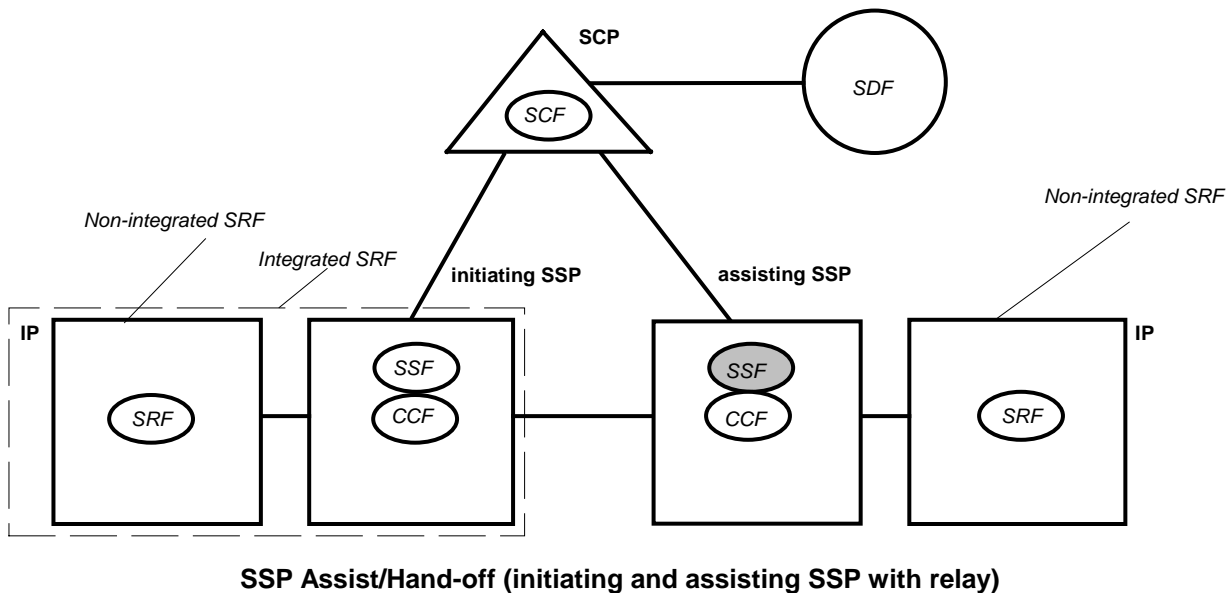
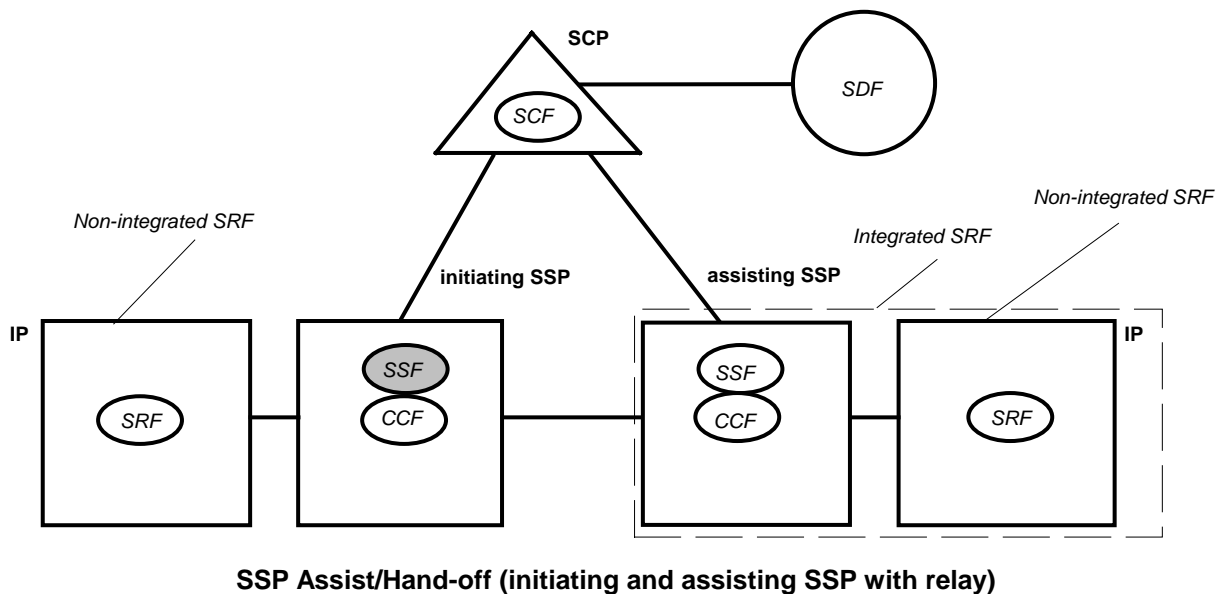
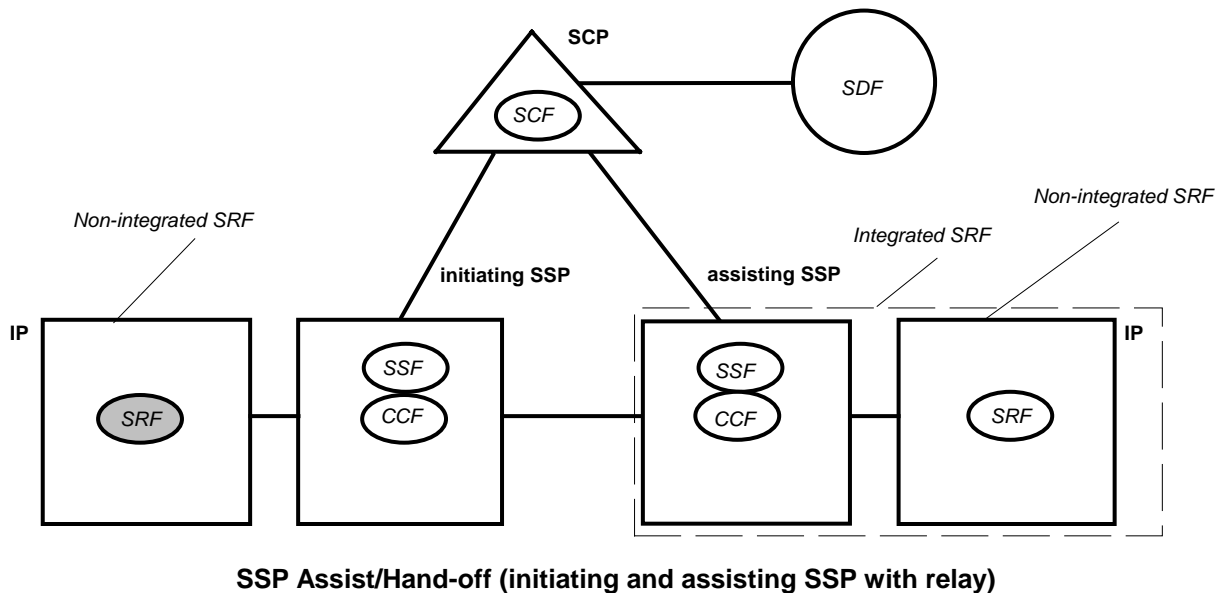


Figure A.29: Configuration 5_5: IUT= SSF of assisting SSP(with non integrated SRF)



SSP Assist/Hand-off (initiating and assisting SSP with relay)

Figure A.30: Configuration 5_6: IUT= SSF of initiating SSP(with non integrated SRF)



SSP Assist/Hand-off (initiating and assisting SSP with relay)

Figure A.31: Configuration 5_7: IUT= SRF of initiating SSP(non integrated SRF)

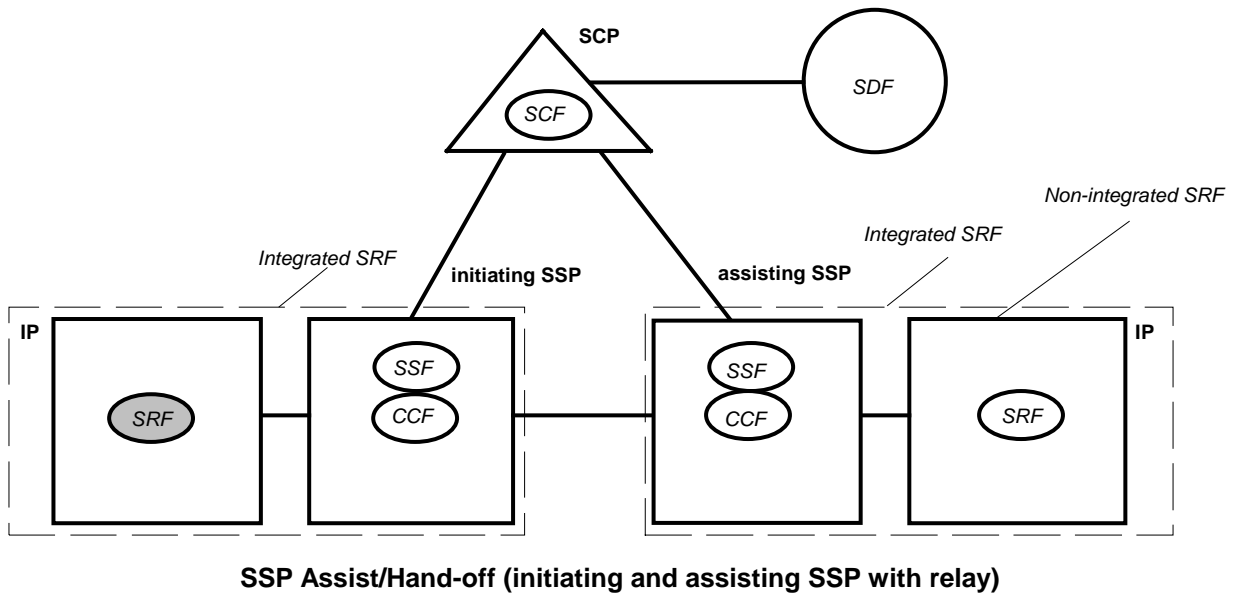


Figure A.32: Configuration 5_8: IUT= SRF of initiating SSP(integrated SRF)

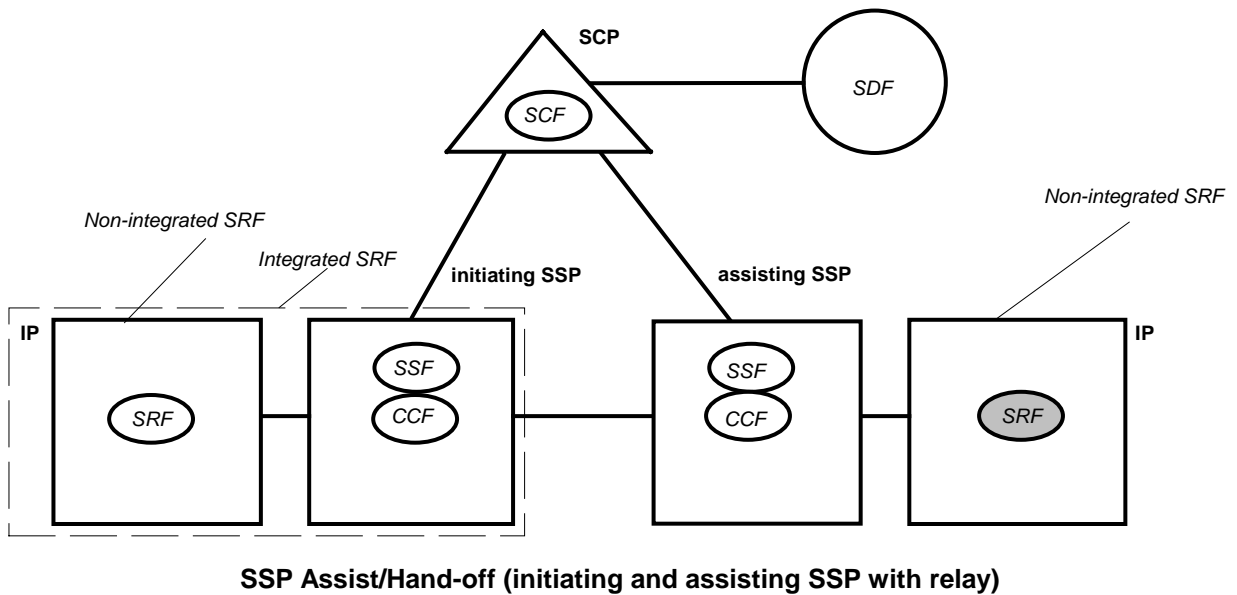
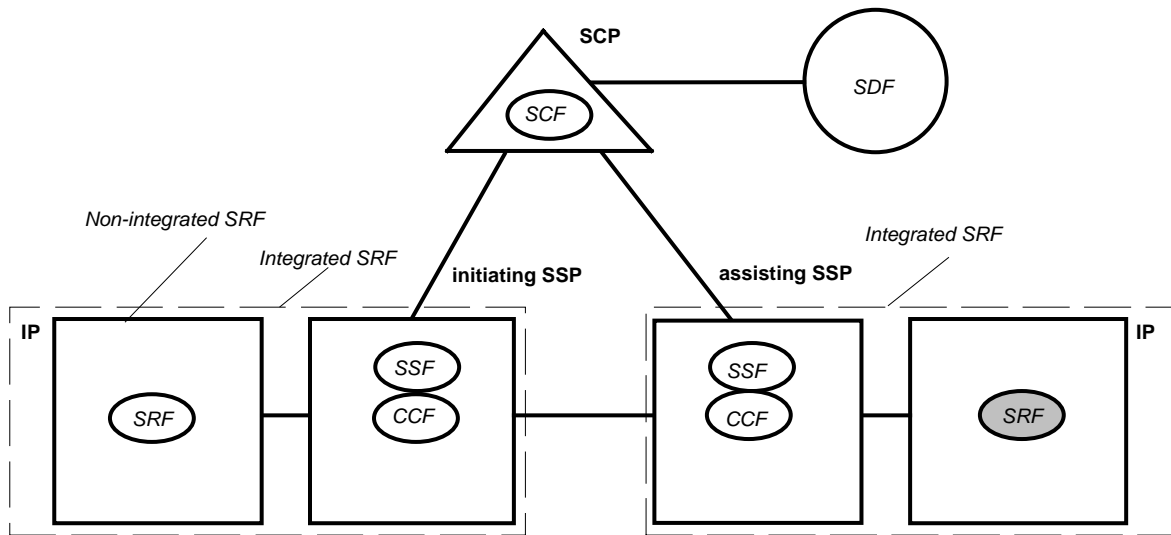


Figure A.33: Configuration 5_9: IUT= SRF of assisting SSP(non integrated SRF)



SSP Assist/Hand-off (initiating and assisting SSP with relay)

Figure A.34: Configuration 5_10: IUT= SRF of assisting SSP(integrated SRF)

Annex B (normative): Parameter values used in MSCs for CORE INAP primitives

The following table is an abstract from the PIXIT for CORE INAP CS1, showing the values of the parameters of CORE INAP primitives used to design the MSCs.

Table B.1

Item	Parameter	Parameter type	Explanation/Format	Value
	PIX_AChBillingChargingCharacteristics	AChBillingChargingCharacteristics	"xx"H	44
	PIX_AlertingPattern	AlertingPattern	"xxx"H	123
	PIX_AlertingPattern_ICA	AlertingPattern	"xxx"H	124
	PIX_APtyAbandonCause	Cause	"xx"H	0F
	PIX_APtyDiscCause	Cause	"xx"H	10
	PIX_AssistingSSPIRoutingAddress	AssistingSSPIRoutingAddress	"xxxx"H	7755
	PIX_BPtyBusy_UDUBCause	Cause	"xx"H	0D
	PIX_BPtyNoAnswerCause	Cause	"xx"H	09
	PIX_CalledPartyNumber1_CON	CalledPartyNumber	LegId 2 "xxxx"H	2001
	PIX_CalledPartyNumber2_CON	CalledPartyNumber	LegId 3 "xxxx"H	2003
	PIX_CalledPartyNumber3_CON	CalledPartyNumber	LegId 4 "xxxx"H	2005
	PIX_CalledPartyNumber4_CON	CalledPartyNumber	LegId 5 "xxxx"H	2007
	PIX_CalledPartyNumber5_CON	CalledPartyNumber	LegId 6 "xxxx"H	2009
	PIX_CalledPartyNumber6_CON	CalledPartyNumber	LegId 7 "xxxx"H	2011
	PIX_CalledPartyNumber7_CON	CalledPartyNumber	LegId 8 "xxxx"H	2013
	PIX_CalledPartyNumber8_CON	CalledPartyNumber	LegId 9 "xxxx"H	2015
	PIX_CalledPartyNumberInvalid_CON	CalledPartyNumber	"xxxx"H	AA20
	PIX_CalledPartyNumber1_ICA	CalledPartyNumber	LegId 2 "xxxx"H	2100
	PIX_CalledPartyNumber2_ICA	CalledPartyNumber	LegId 3 "xxxx"H	2101
	PIX_CalledPartyNumber3_ICA	CalledPartyNumber	LegId 4 "xxxx"H	2102
	PIX_CalledPartyNumber4_ICA	CalledPartyNumber	LegId 5 "xxxx"H	2103
	PIX_CalledPartyNumber5_ICA	CalledPartyNumber	LegId 6 "xxxx"H	2104
	PIX_CalledPartyNumber6_ICA	CalledPartyNumber	LegId 7 "xxxx"H	2105
	PIX_CalledPartyNumber7_ICA	CalledPartyNumber	LegId 8 "xxxx"H	2106
	PIX_CalledPartyNumber1_SetupInd	CalledPartyNumber	"xxxx"H	2000
	PIX_CalledPartyNumber2_SetupInd	CalledPartyNumber	"xxxx"H	2002
	PIX_CallingPartyNumber1	CallingPartyNumber	"xxxx"H	1000
	PIX_CallingPartyNumber2	CallingPartyNumber	"xxxx"H	1002
	PIX_CallingPartysCategory_CON	CallingPartysCategory	"xx"H	BB
	PIX_CallingPartysCategory_SetupInd	CallingPartysCategory	"xx"H	CC
	PIX_DateAndTime	DateAndTime	YYMMDDHHMMSS	980115123030
	PIX_Duration	Duration	Seconds	66
	PIX_EventTypeCharging1	EventTypeCharging		"AAAA"
	PIX_EventTypeCharging2	EventTypeCharging		"CCCC"
	PIX_FCIBillingChargingCharacteristics	FCIBillingChargingCharacteristics		55
	PIX_InbandInfo_message	InbandInfo	InformationToSend	"AABB"
	PIX_Interval	Integer	Seconds	33
	PIX_IPRoutingAddress	IPRoutingAddress	"xxx"H	400
	PIX_LocationNumber	LocationNumber	"xxx"H	9001
	PIX_MaximumNumberOfCounters	MaximumNumberOfCounters	"xx"H	14
	PIX_NumberOfCalls	Integer	xx	13
	PIX_OriginalCalledPartyNumber	CalledPartyNumber	"xxxx"H	2211
	PIX_RedirectingPartyNumber	CalledPartyNumber	"xxxx"H	3000
	PIX_RedirectionInformation	RedirectionInformation	"xx"H	AA
	PIX_ReleaseCause	Cause	"xx"H	00
	PIX_RouteSelectFailure1Cause	Cause	"xx"H	0B
	PIX_RouteSelectFailure2Cause	Cause	"xx"H	0C

Item	Parameter	Parameter type	Explanation/Format	Value
	PIX_ScfID	ScfID	"xxx"H	8881
	PIX_ServiceInteractionIndicators	ServiceInteractionIndicators	"xx"H	22
	PIX_ServiceKey1	ServiceKey	"xx"H	27
	PIX_ServiceKey2	ServiceKey	"xx"H	28
	PIX_SFBillingChargingCharacteristics	SFBillingChargingCharacteristics	"xxx"H	BBBB
	PIX_StartTime	DateAndTime	YYMMDDHHMMSS	971128113015
	PIX_StopTime	DateAndTime	YYMMDDHHMMSS	971212113015
	PIX_ElementaryMessageID	integer	xxx	191
	PIX_CorrelationId	correlationID	"xxx"H	AAA
	PIX_UiScriptID1	integer	xxx	202
	PIX_UiScriptID2	integer	xxx	203
	PIX_UiScriptIDInvalid	integer	xxx	210
	PIX_UiScriptResult	UiScriptResult	"xxx"H	5110
	PIX_UiScriptSpecificInfo	UiScriptSpecificInfo	"xxx"H	5220

Annex C (normative): Parameter values used in MSCs for TCAP primitives

The following table is an abstract from the PIXIT for CORE INAP CS1, showing the values of the parameters of TCAP primitives used to design the MSCs.

Table C.1: Parameter values

Item	Parameter	Parameter type	Explanation	Value
	PIX_Invokeld	InvokeIDType	Direction SCF ->SSF	1-100
			Direction SSF->SCF	101-200
	PIX_DialogueId	DialogueIDType	Direction SCF ->SSF	1-50
			Direction SSF->SCF	51-100

Bibliography

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

- ETSI EN 301 140-2: "Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 2 (CS2); Part 2: Protocol Implementation Conformance Statement (PICS) proforma specification".

History

Document history			
V1.1.2	June 1999	Public Enquiry	PE 9947: 1999-06-23 to 1999-11-19
V1.1.3	February 2000	Vote	V 200017: 2000-02-28 to 2000-04-28