

# ETSI EN 301 140-3-2 V1.1.3 (2000-05)

---

*European Standard (Telecommunications series)*

**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 2: Call Party Handling (CPH)**

---



---

**Reference**

DEN/SPS-03038-3-2

---

**Keywords**

IN, CS2, INAP, TSS&amp;TP, SSF

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - 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

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document 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.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at <http://www.etsi.org/tb/status/>

If you find errors in the present document, send your comment to:

editor@etsi.fr

---

**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 .....	6
4 Void.....	7
5 Void.....	7
6 Void.....	7
7 TSS and TP for CPH functions .....	7
7.0 Introduction .....	7
7.1 Preambles and postambles for CPH .....	7
7.1.1 Names of preambles and postambles .....	7
7.1.2 Preamble trees.....	9
7.1.3 TTCN-like notation for preamble description.....	10
7.1.4 Representation of preamble/postamble and test purposes using MSCs .....	10
7.1.5 How to interpret the parameters and their values as used in the MSCs.....	10
7.1.6 Preamble descriptions.....	11
7.1.6.1 O (originating) preamble tree .....	11
7.1.6.2 T (terminating) preamble tree.....	29
7.1.6.3 I (InitiateCallAttempt) preamble tree .....	37
7.1.6.4 Event Detecting/Report rules Preambles.....	43
7.1.7 Postamble descriptions .....	59
7.1.7.1 Postamble ReleaseAll_1.....	59
7.1.7.2 Postamble ReleaseAll_2.....	60
7.1.7.3 Postamble ReleaseAll_3.....	61
7.1.7.4 Postamble ReleaseAll_4.....	62
7.1.7.5 Postamble ReleaseAll_5.....	63
7.1.7.6 Postamble ReleaseAll_6.....	64
7.1.7.7 Postamble ReleaseAll_7.....	65
7.1.7.8 Postamble ReleaseAll_8.....	66
7.2 CPH procedures.....	66
7.2.1 List of procedures for CPH.....	66
7.2.2 Definitions of the CPH procedures .....	67
7.2.2.1 mergeCallSegments procedure.....	67
7.2.2.2 releaseCall procedure .....	67
7.2.2.3 continueWithArgument procedure .....	67
7.2.2.4 disconnectLeg procedure.....	67
7.2.2.5 moveLeg procedure.....	67
7.2.2.6 splitLeg procedure.....	67
7.2.2.7 continueWithArgument procedure .....	67
7.3 Structure of the test suite (TSS) for CPH .....	67
7.4 Test Purpose (TP) descriptions for the test of CPH procedures .....	68
7.4.1 MergeCallSegment procedure (MC).....	69
7.4.2 MoveCallSegment procedure (MCS).....	74
7.4.3 ReleaseCall (CS2 complement) procedure (RC) .....	75
7.4.4 DisconnectLeg Procedure (DL).....	78
7.4.5 MoveLeg Procedure (ML).....	83
7.4.6 RequestReportBCSMEEvent Procedure (CS2 additions) (RR) .....	89
7.4.7 SplitLeg Procedure (SL) .....	92
7.4.8 Continue with argument Procedure (CW).....	95

7.5	Test Purpose (TP) descriptions for the test of call handling capabilities.....	102
7.5.1	originating (O_BCSM) trigger (controlling legId = 1) .....	102
7.5.2	Terminating (T_BCSM) trigger (controlling legId = 2).....	174
7.5.3	Network initiated .....	197
7.6	Test Purpose (TP) descriptions for testing arming/detecting rules .....	208
7.6.1	Originating (O) trigger.....	209
7.6.1.1	O_1 Events coming from the controlling leg (legId=1) .....	209
7.6.1.2	O_2 Events coming from passive legs (legId= 2,3...) .....	219
7.6.2	Terminating (T) trigger .....	238
7.6.2.1	T_1 Events coming from the controlling leg (legId=2),.....	238
7.6.2.2	T_2 Events coming from passive legs (legId= 2,3...).....	245
<b>Annex A (normative):</b>	<b>Preamble trees .....</b>	<b>263</b>
<b>Annex B (normative):</b>	<b>TCAP Parameter values .....</b>	<b>266</b>
<b>Annex C (normative):</b>	<b>Core INAP Parameter values .....</b>	<b>267</b>
History .....		269

## 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 Protocols for Advanced Networks (SPAN).

The present document is part 3, sub-part 2 of a multi-part EN covering the Intelligent Network Application Protocol (INAP) capability set 2, 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 [2] (1997) modified]".

<b>National transposition dates</b>	
Date of adoption of this EN:	28 April 2000
Date of latest announcement of this EN (doa):	31 July 2000
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	31 January 2001
Date of withdrawal of any conflicting National Standard (dow):	31 January 2001

---

## 1 Scope

The present document contains the Test Suite Structure and Test Purposes (TSS&TP) for Call Party Handling (CPH), part of CoreINAP CS2. It complements the initial document EN 301 140-3-1 [1] dedicated to general introduction and TSS&TPs for CoreINAP CS1.

---

## 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 301 140-3-1: "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".

[2] ITU-T Recommendation Q.1224: "Distributed functional plane for intelligent network Capability Set 2".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in EN 301 140-3-1 [1] apply.

### 3.2 Abbreviations

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

BCSM	Basic Call State Model
CCF	Call Control Function
CPH	Call Party Handling
CS	Capability Set
CS	Call Segment
CV	Connection View
INAP	Intelligent Network Application Protocol
MSC	Message Sequence Chart
PDU	Protocol Data Unit
PIXIT	Protocol Implementation eXtra Information for Testing
SCF	Service Control Function
SDL	Service and Description Language
SRF	Specialized Resource Function
SSF	Service Switching Function
SSP	Service Switching Point
TCAP	Transaction Capabilities Application Part

TP	Test Purpose
TSS	Test Suite Structure
TTCN	Tree and Tabular Combined Notation

---

## 4 Void

See EN 301 140-3-1 [1].

---

## 5 Void

See EN 301 140-3-1 [1].

---

## 6 Void

See EN 301 140-3-1 [1].

---

# 7 TSS and TP for CPH functions

## 7.0 Introduction

The Call Party Handling (CPH) includes a set of operations related to the call handling at the switch. The whole functionality of the CPH operations is presented by means of the four Core Capabilities (ITU-T Recommendation Q.1224 [2]):

- Core Capability one: allows the user to enter information during a midcall event;
- Core Capability two: is the ability of the SSF/CCF to connect a call party to an external resource to perform a transfer;
- Core Capability three is the ability of the SSF/CCF to present the current call view to the SCF;
- Core Capability four: is the ability of the SSF/CCF to combine separate calls into one a single call.

The Test Purposes related to CPH are classified into 3 categories:

- 1) TP for the test of Conformance of each CPH procedure: this forms a set of TPs testing the basic functionality of CPH operations;
- 2) TP for the test of the Switch capabilities: a group of TPs that tests the switch capability of handling different calls at the same time;
- 3) Arming/Detecting rules.

## 7.1 Preambles and postambles for CPH

### 7.1.1 Names of preambles and postambles

CPH requires a large set of preambles. Due to the complexity of their description, the Connection View (CV) model is used for an understanding of the configuration, referring to the following CV objects:

- CallSegmentAssociation (always initial);
- CallSegment;

- Connection point;
- Legs.

NOTE 1: The controlling leg can be either joined, shared or surrogate. The controlling leg identifies the physical access to the end user.

NOTE 2: The legs are named by the LegId, and there is an unique correspondence between a LegId and a BCSM.

**Restrictions:** The test configuration is limited to three passive legs within a call segment, and three call segments within a call segment association.

**Comment on T preambles:** 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.

Based on these considerations, using the naming conventions indicated in subclause 4.5.4 of EN 301 140-3-1 [1], and in addition to the CS1 preambles, which are:

O\_OS\_null\_null;

O\_S2P\_null\_null.

CPH uses the following preambles:

- Originating

O\_OH(1)\_S2P\_null(1)

O\_OH(1)\_OH(1)\_S2P

O\_null\_OH(2)\_S2P

O\_null\_OH(2)\_S2P\_controlling (To test the event report rules)

O\_null\_OH(2)\_S2P\_passive (To test the event report rules)

O\_null\_S2P\_OH(3)

O\_S2P\_OH(1)\_OH(3)

O\_S2P\_OH(2)\_OH(3)

O\_S2P\_OH(3)\_OH(3)

O\_null\_S3P\_null

O\_null\_S3P\_OH(3)

O\_null\_null\_S4P

O\_null\_null\_S4P\_controlling (To test the event report rules)

O\_null\_null\_S4P\_passive (To test the event report rules)

O\_null\_S4P\_OH(3)

- Terminating

T\_TS\_null\_null

T\_S2P\_null\_null

T\_S2P\_null\_null\_controlling (To test the event report rules)

T\_S2P\_null\_null\_passive (To test the event report rules)



T\_OH(1)\_S2P\_null  
 T\_null\_OH(2)\_S2P  
 T\_null\_S3P\_null  
 T\_null\_null\_S4P  
 T\_null\_null\_S4P\_controlling (To test the event report rules)  
 T\_null\_null\_S4P\_passive (To test the event report rules)  
 T\_TF(2)\_null\_null

- Initiate Call Attempt (network initiated)

I\_S1P\_null\_null  
 I\_null\_TF(2)\_null  
 I\_null\_TF(2)\_null\_passive(To test the event report rules)  
 I\_null\_null\_TF(3)  
 I\_S1P\_S1P\_null  
 I\_S1P\_S1P\_null\_passive(To test the event report rules)  
 I\_null\_TF(2)\_S1P  
 I\_S1P\_S1P\_S1P

## 7.1.2 Preamble trees

Each preamble is composed of a limited set of operations (listed below) with the indication of the main parameters used. These operations are shown in the preamble descriptions:

- CWA (ContinueWithArgument): CsId;
- CON (Connect): LegToBeCreated (default = 2), CsID;
- ICA (InitiateCallAttempt): LegToBeCreated (default = 1), newCallSegment (default = 1);
- MC (MergeCallSegments): sourceCallSegment, targetCallSegment;
- SL (SplitLeg): LegToBeSplit, newCallSegmentID;
- IDP (Initial Detection Point): trigger;
- RRB (RequestReportBCSMEvent): LegID, eventTypeBCSM.

Each operation shows the value of the required parameters, knowing that the leg numbers successively take the values 1, 2, 3, etc. In the case of CS, the same number is reused when a CS returns to idle following a mergeCallSegment.

The preamble trees are defined in subclause 7.1.6, where each branch or each preamble is numbered 1, 2, 3, etc., except when there is an alternative or a node of two possible branches, which are then numbered 4-1 and 4-2 for example.

Each preamble shows the state from where it starts (idle or a different state reached by the execution of another preamble), then it shows the operations executed in this preamble and finally the state or configuration reached, using the notation described above.

### 7.1.3 TTCN-like notation for preamble description

The notation used to describe the trees and the required operations to move from one preamble to the next one, is a TTCN-like notation, showing what is sent (character !) and received (character ?) by the co-ordination points (CPs) addressing either Signalling Control or user A (CP1-1) or B (CP1-2) or C (CP1-3), etc., or by the main tester L1 playing the role of the SCF.

### 7.1.4 Representation of preamble/postamble and test purposes using MSCs

In addition to the TTCN-like notation, an MSC is drawn from the SDL simulator to represent each preamble or postamble. For each test purpose, an MSC is also given, in addition to the tabular description of each TP.

Each MSC shows the interface between SCF and SSF using TCAP primitives, and the signalling control points. As there can be any number of signalling control points (from 1 up to 8), the MSC shows SigCon A in one column, while all the other SigCon are merged in a second column. The parameter CallRef number makes it possible to identify the SigCon concerned, SigCon B being number 2, SigCon C being 3, etc.

### 7.1.5 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 EN 301 140-3-1 [1].

The list of parameters for the TCAP protocol is repeated here for each TCAP primitive. 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.

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

## 7.1.6 Preamble descriptions

### 7.1.6.1 O (originating) preamble tree

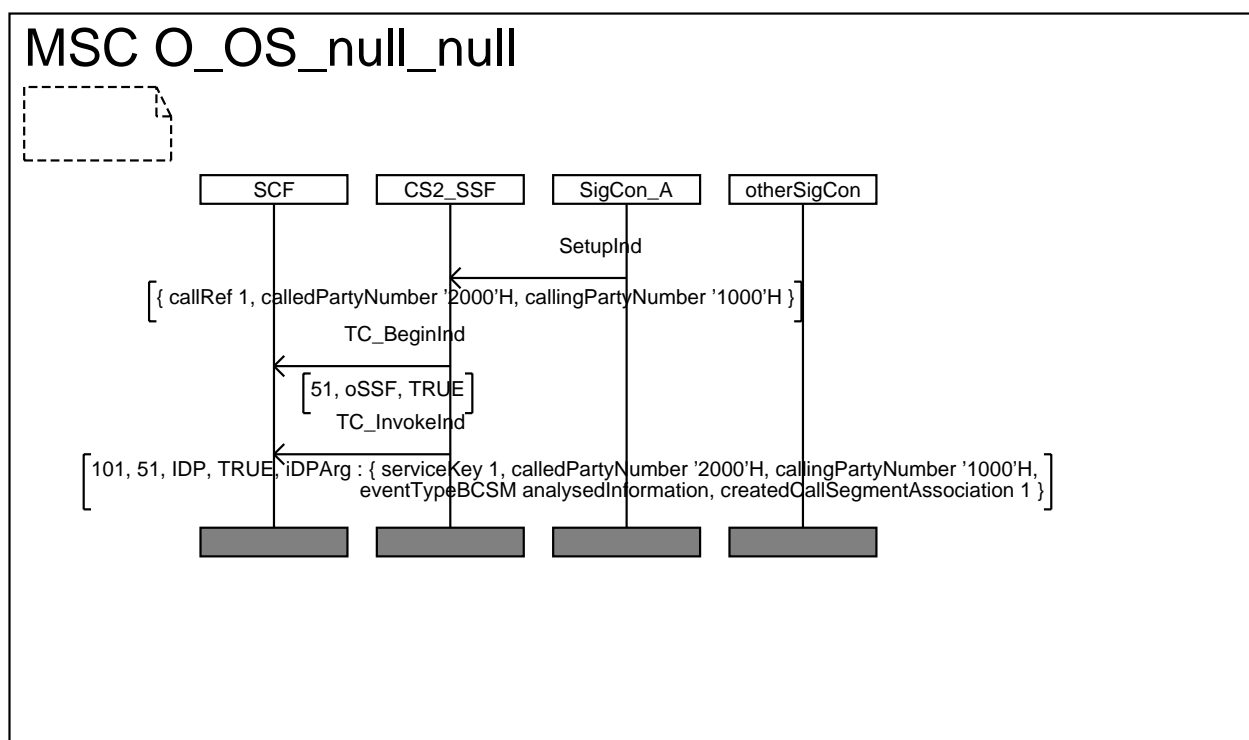
Preamble O\_OS\_null\_null

O\_null

CP1-1! SetUpInd

L1? InitialDP

O\_OS\_null\_null



2 - Preamble O\_S2P\_null\_null

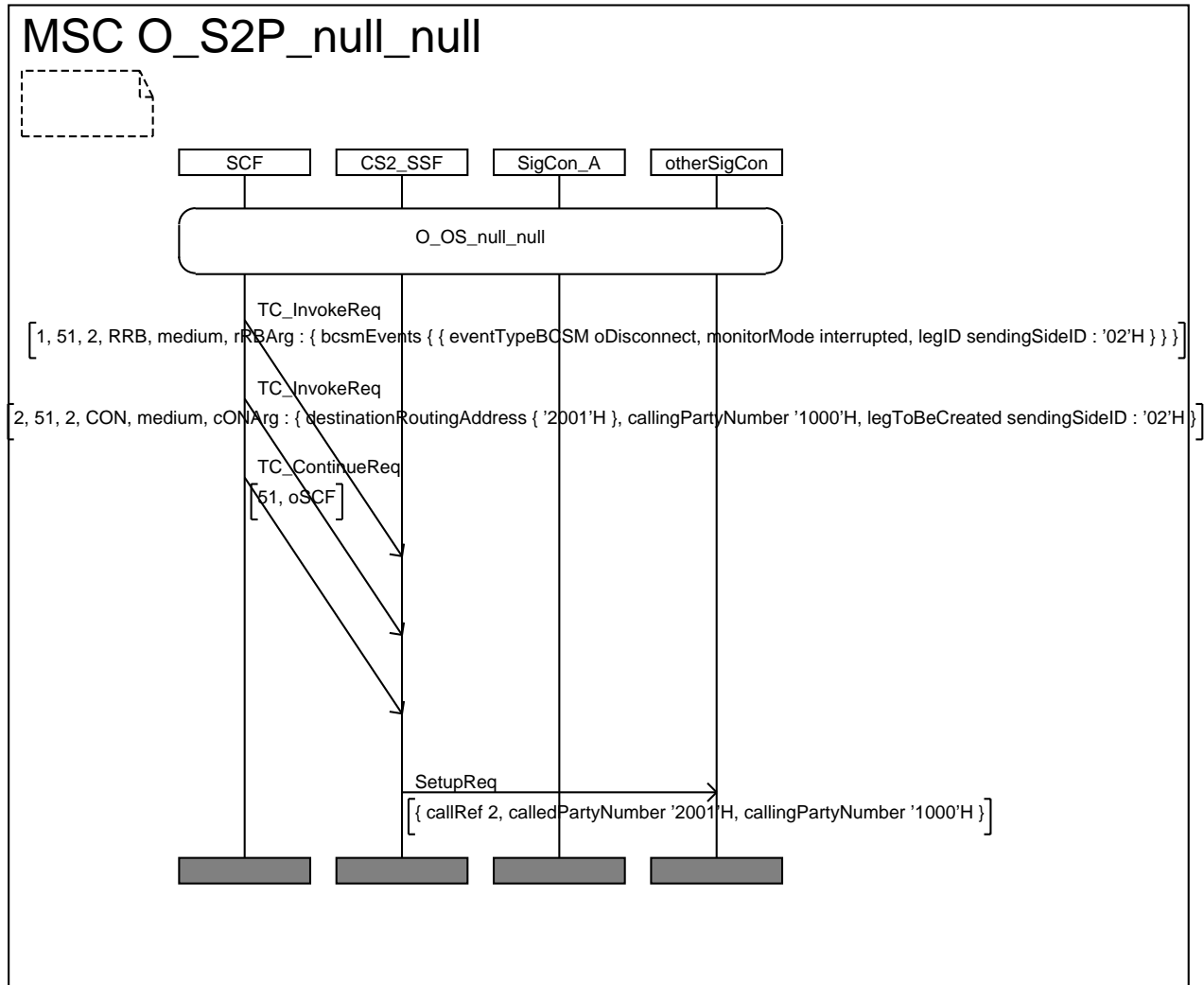
O\_OS\_null\_null

L1! RequestReportBCSMEvent(2,oDisconnect)

L1! Connect(2,1)

CP1-2? SetUpReq

O\_S2P\_null\_null



3 - Preamble O\_OH(1)\_S2P\_null

O\_S2P\_null\_null

L1! SplitLeg(1,2)

L1?SplitLegReturnResult

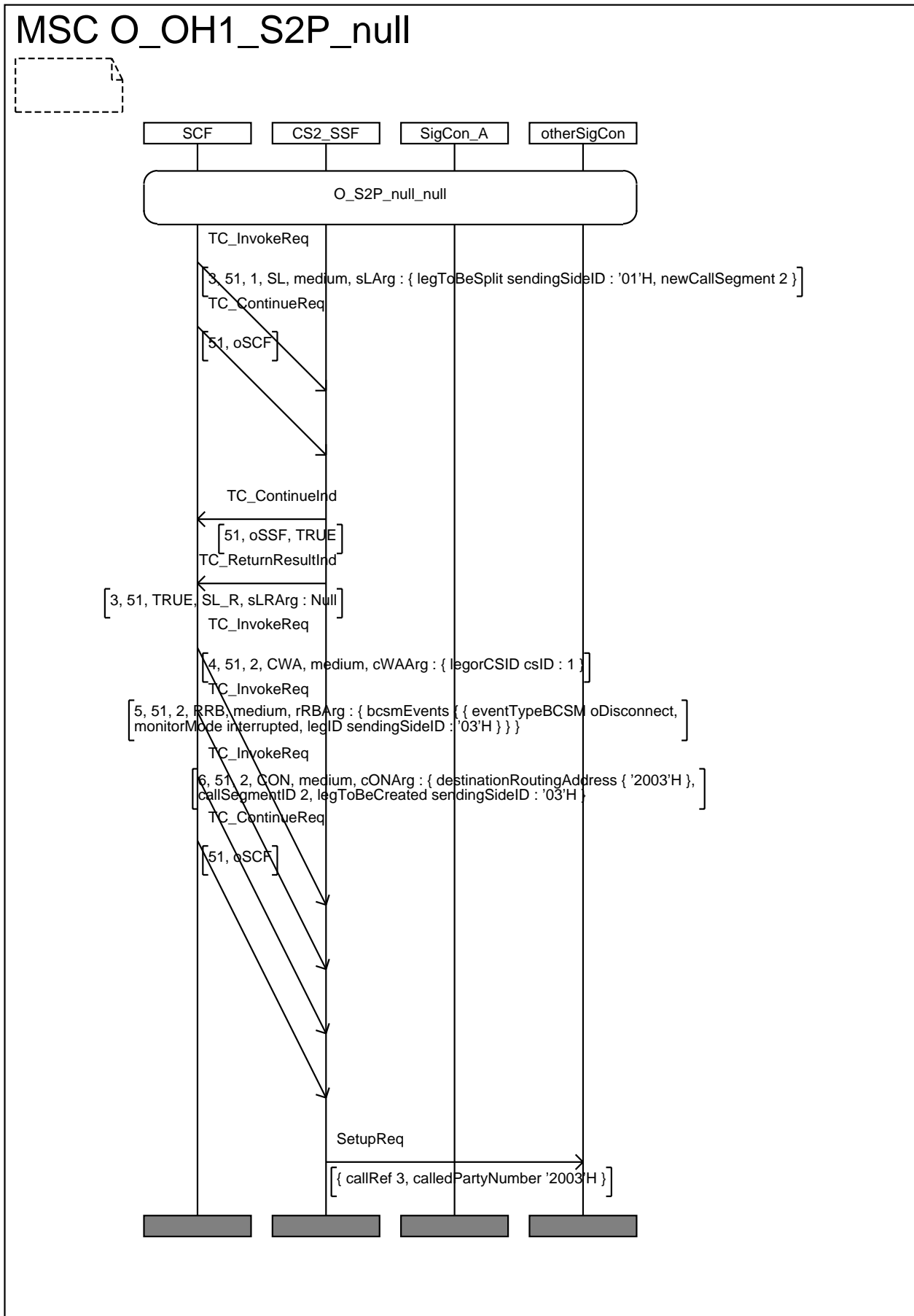
L1! ContinueWithArgument (CsID = 1)

L1! RequestReportBCSMEEvent(3,oDisconnect)

L1! Connect(3,2)

CPI-3? SetUpReq

O\_OH(1)\_S2P\_null



4 1 - Preamble O\_OH(1)\_OH(1)\_S2P

O\_OH(1)\_S2P\_null

L1! SplitLeg(1,3)

L1?SplitLegReturnResult

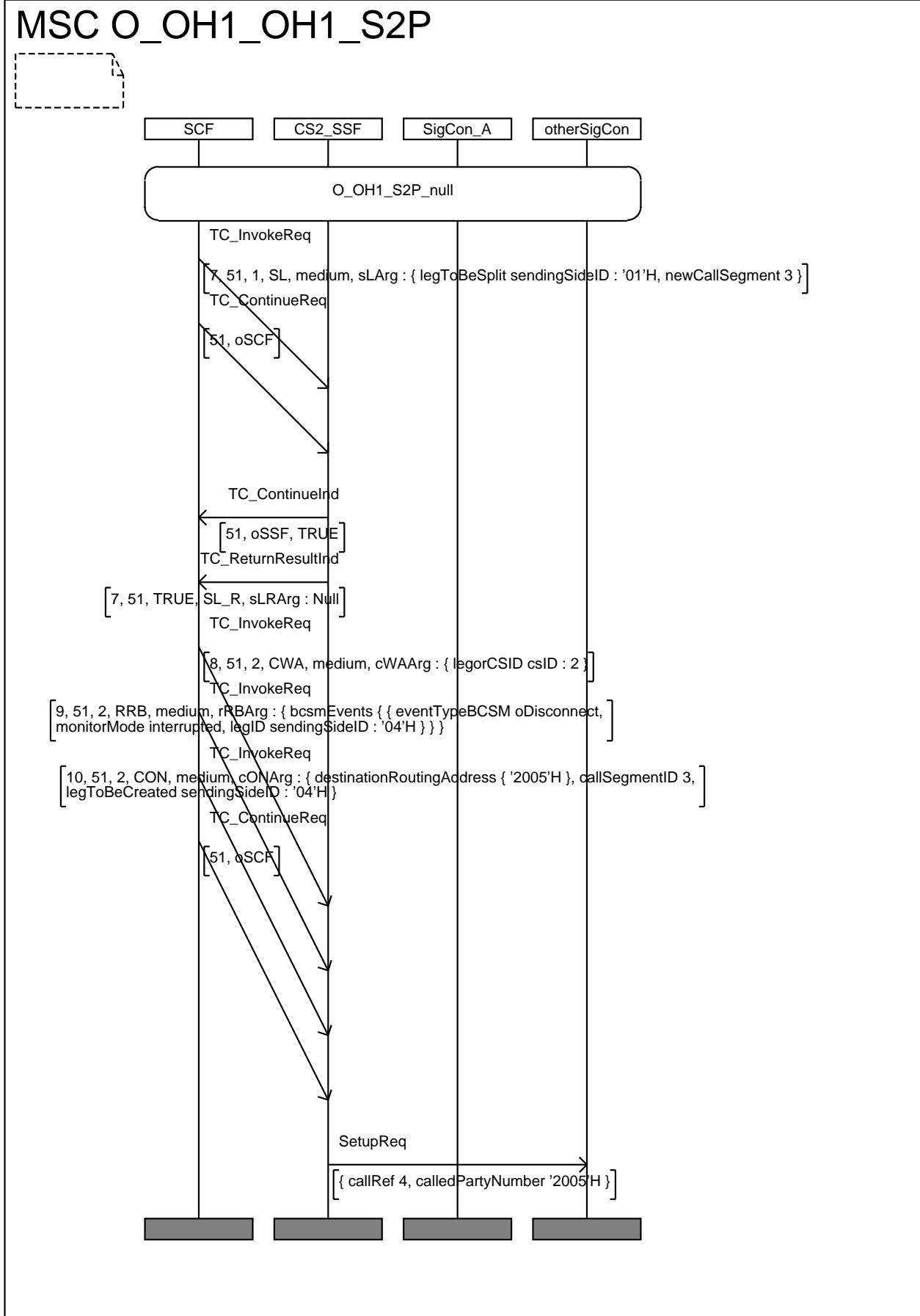
L1! ContinueWithArgument (CsID = 2)

L1! RequestReportBCSMEEvent(4,oDisconnect)

L1! Connect(4,3)

CP1-4? SetUpReq

O\_OH(1)\_OH(1)\_S2P



5 - Preamble O\_null\_OH(2)\_S2P

O\_OH(1)\_OH(1)\_S2P

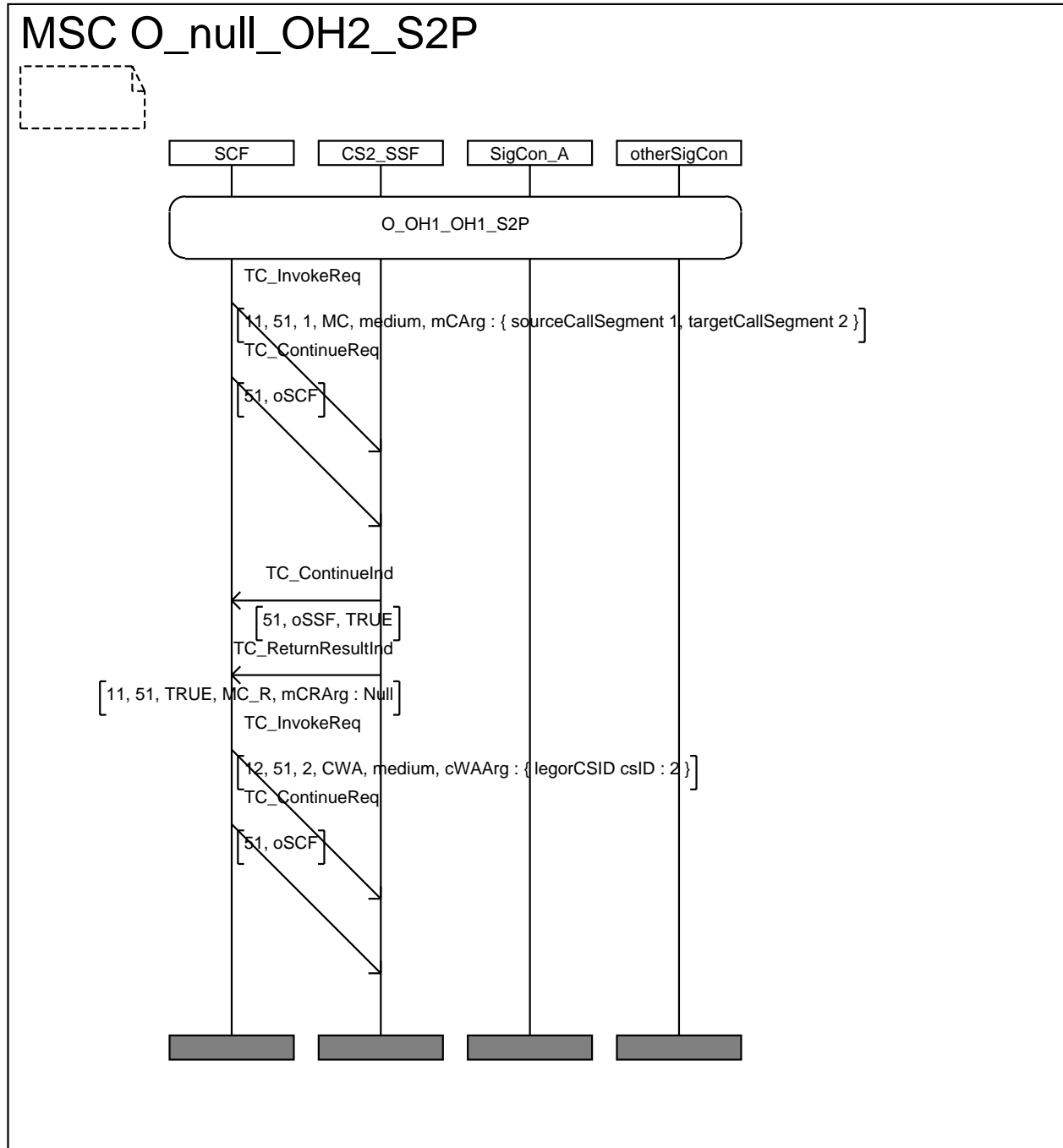
L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=2)

O\_null\_OH(2)\_S2P





6 - Preamble O\_null\_null\_S4P

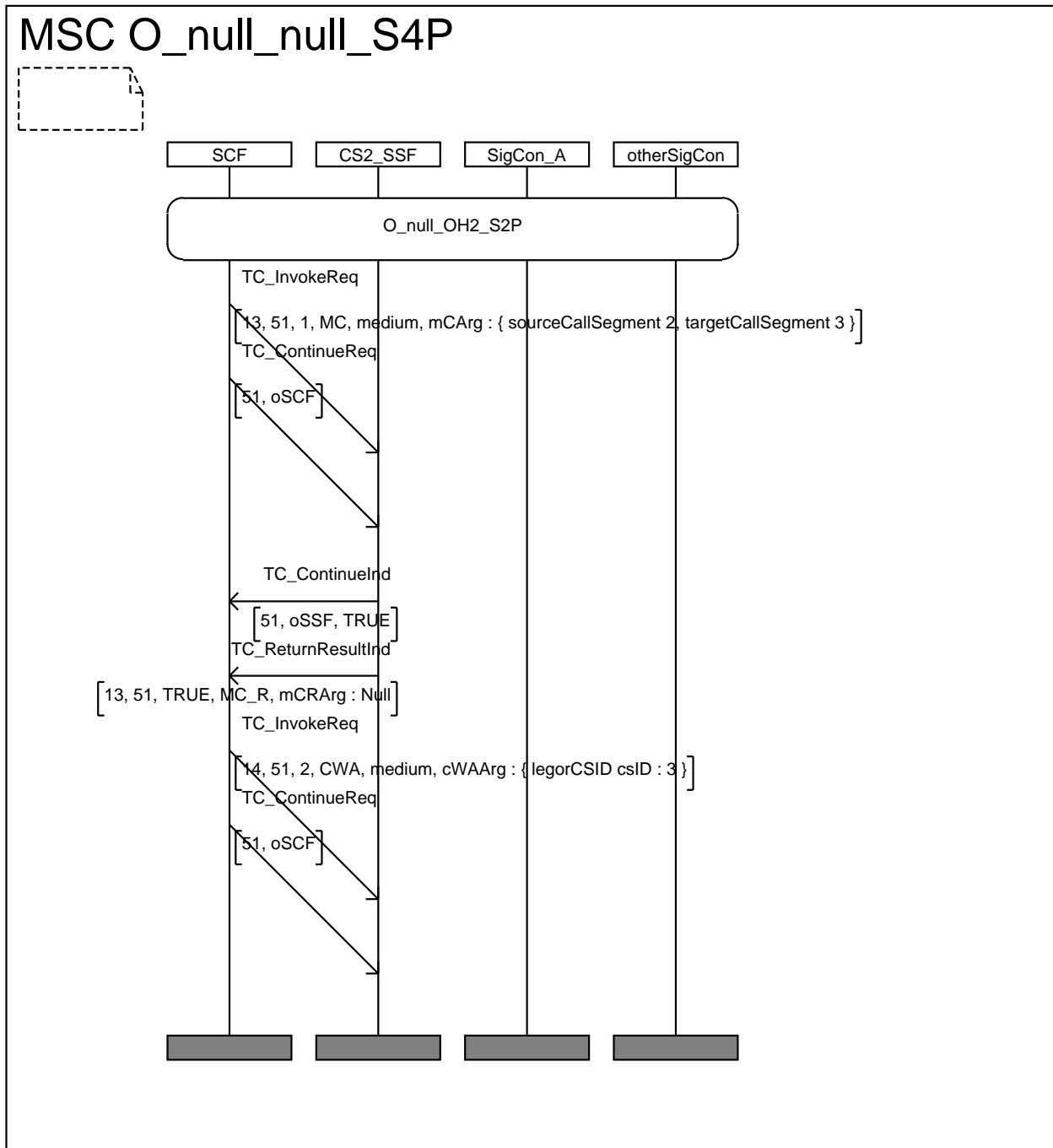
O\_null\_OH(2)\_S2P

L1! MergeCallSegments(2,3)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CsId=3)

O\_null\_null\_S4P



7 - Preamble O\_null\_S2P\_OH(3)

O\_null\_null\_S4P

L1! SplitLeg(1,2)

L1?SplitLegReturnResult

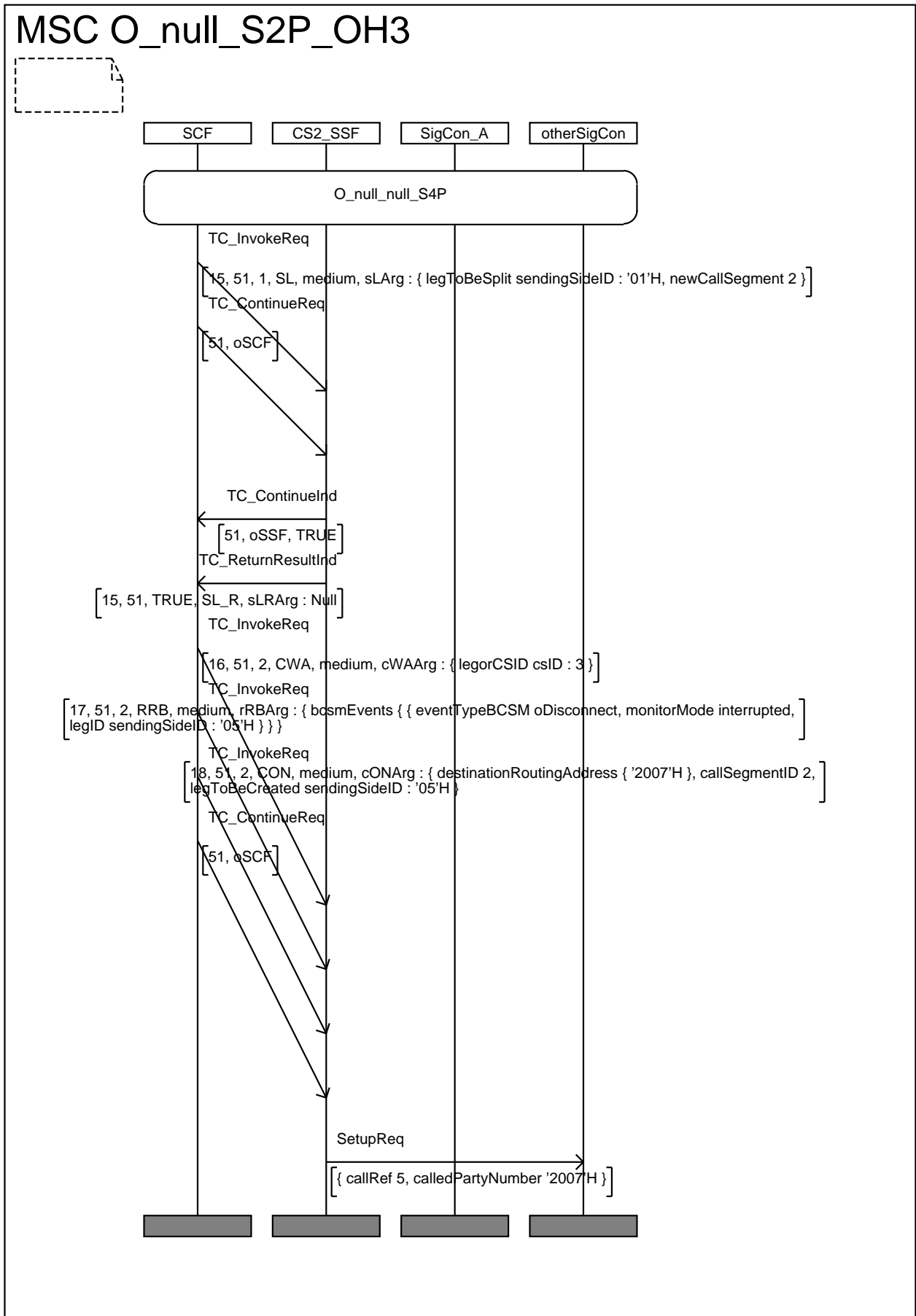
L1! ContinueWithArgument (CsID = 3)

L1! RequestReportBCSMEEvent(5,oDisconnect)

L1! Connect(5,2)

CP1-5? SetUpReq

O\_null\_S2P\_OH(3)



8 - Preamble O\_S2P\_OH(1)\_OH(3)

O\_null\_S2P\_OH(3)

L1! SplitLeg(1,1)

L1?SplitLegReturnResult

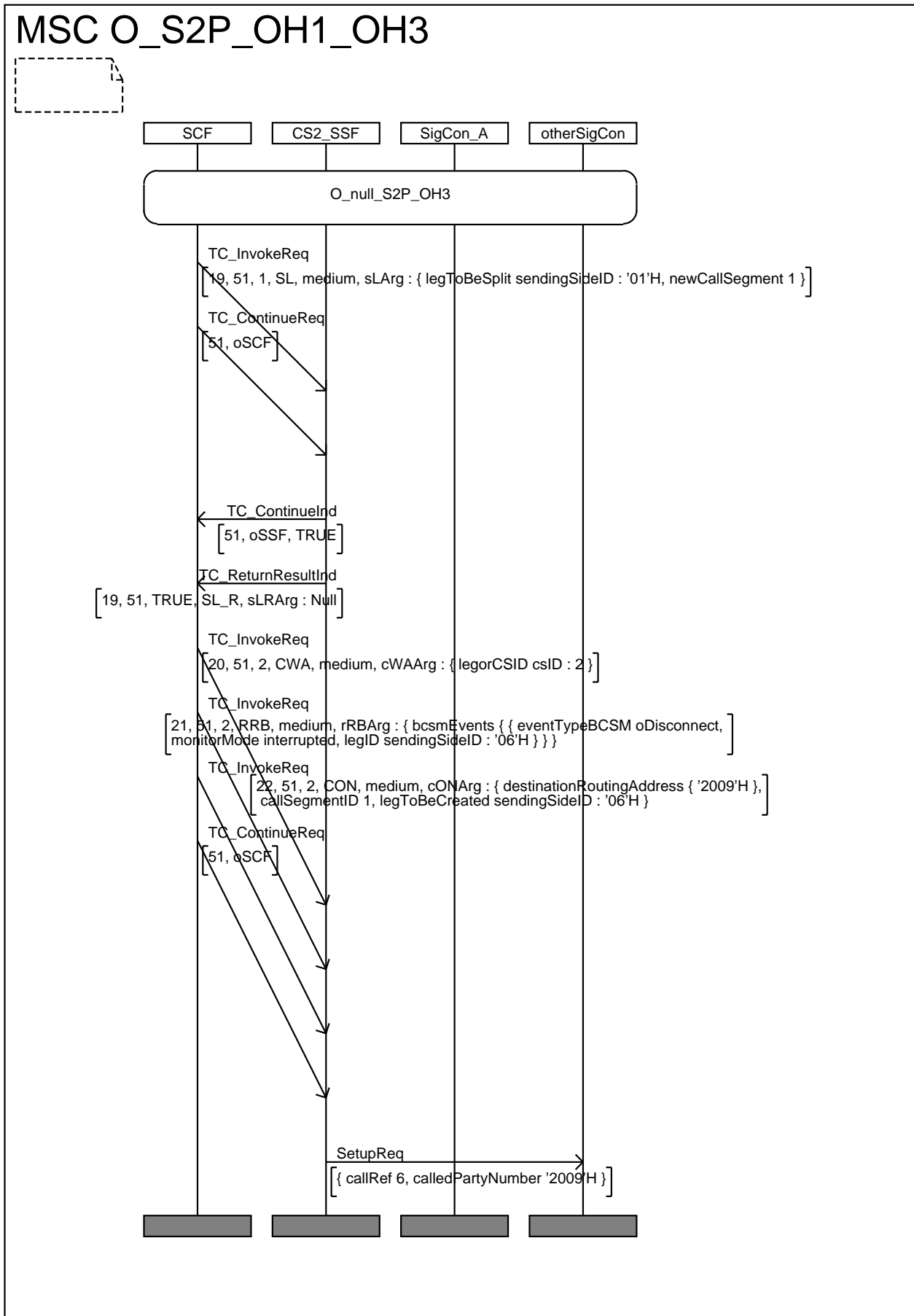
L1! ContinueWithArgument (CsID = 2)

L1! RequestReportBCSMEEvent(6,oDisconnect)

L1! Connect(6,1)

CP1-6? SetUpReq

O\_S2P\_OH(1)\_OH(3)



9 - Preamble O\_null\_S3P\_OH(3)

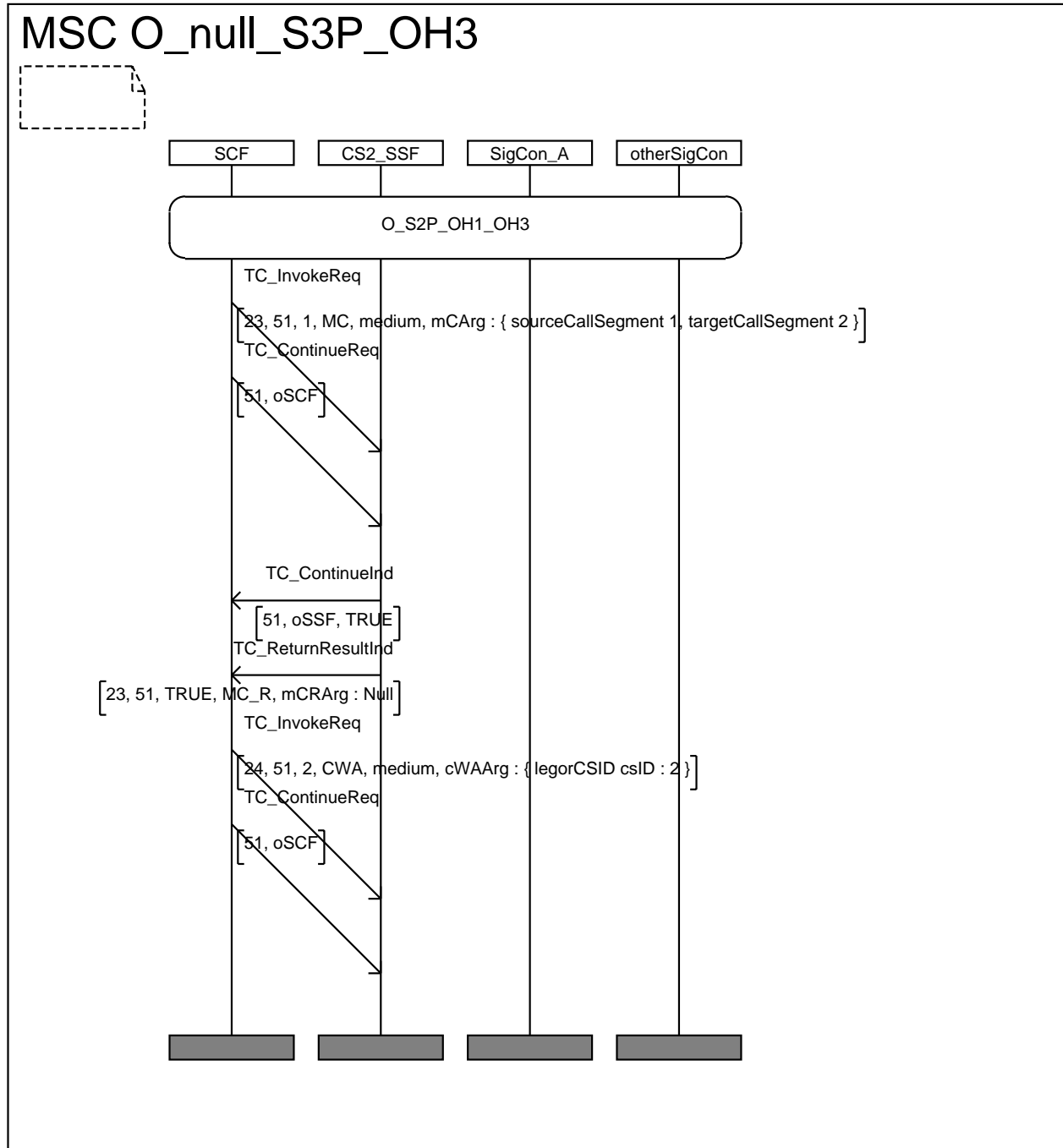
O\_S2P\_OH(1)\_OH(3)

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=2)

O\_null\_S3P\_OH(3)



10 - Preamble O\_S2P\_OH(2)\_OH(3)

O\_null\_S3P\_OH(3)

L1! SplitLeg(1,1)

L1?SplitLegReturnResult

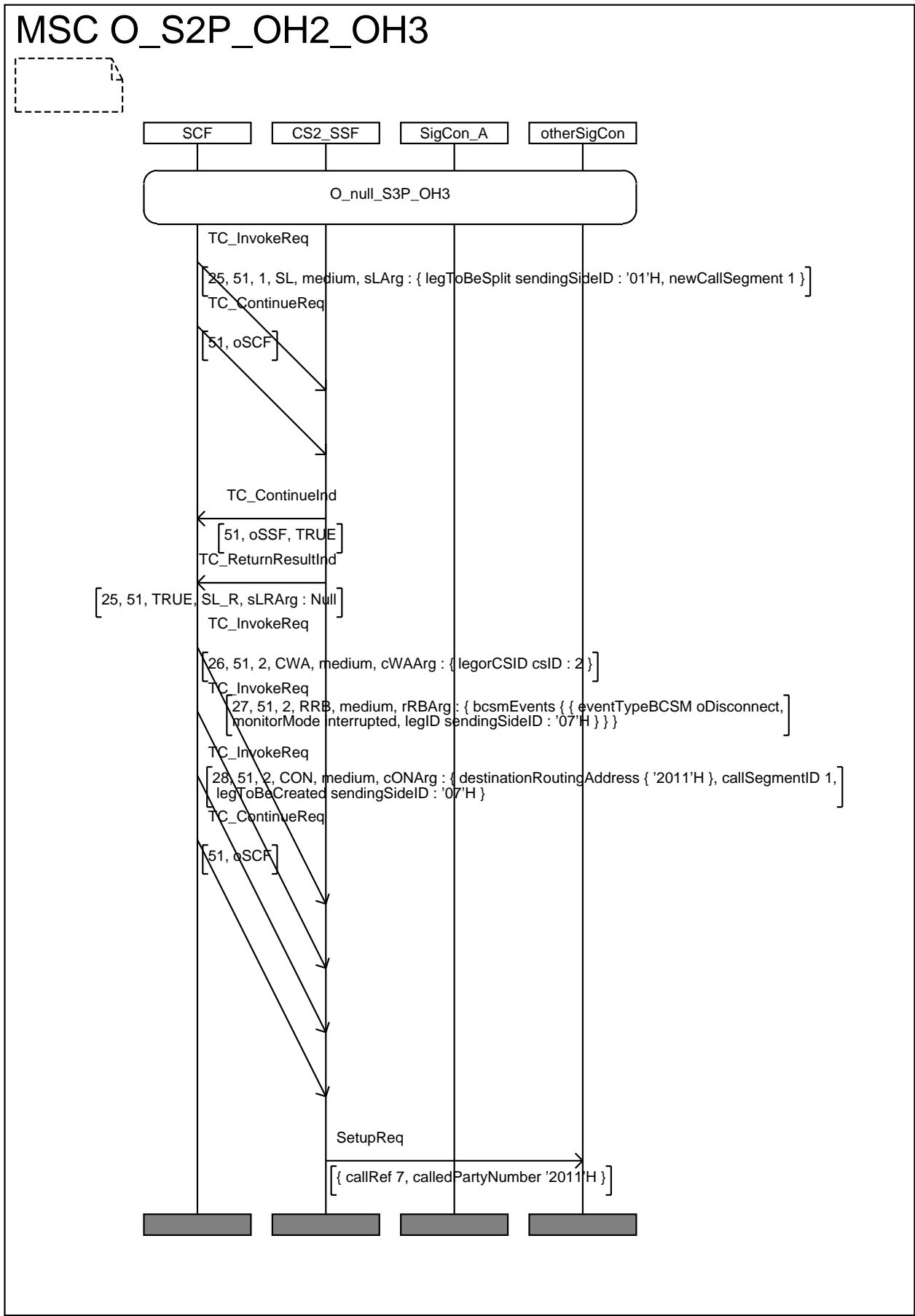
L1! ContinueWithArgument (CsID = 2)

L1! RequestReportBCSMEEvent(7,oDisconnect)

L1! Connect(7,1)

CP1-7? SetUpReq

O\_S2P\_OH(2)\_OH(3)



11 - Preamble O\_null\_S4P\_OH(3)



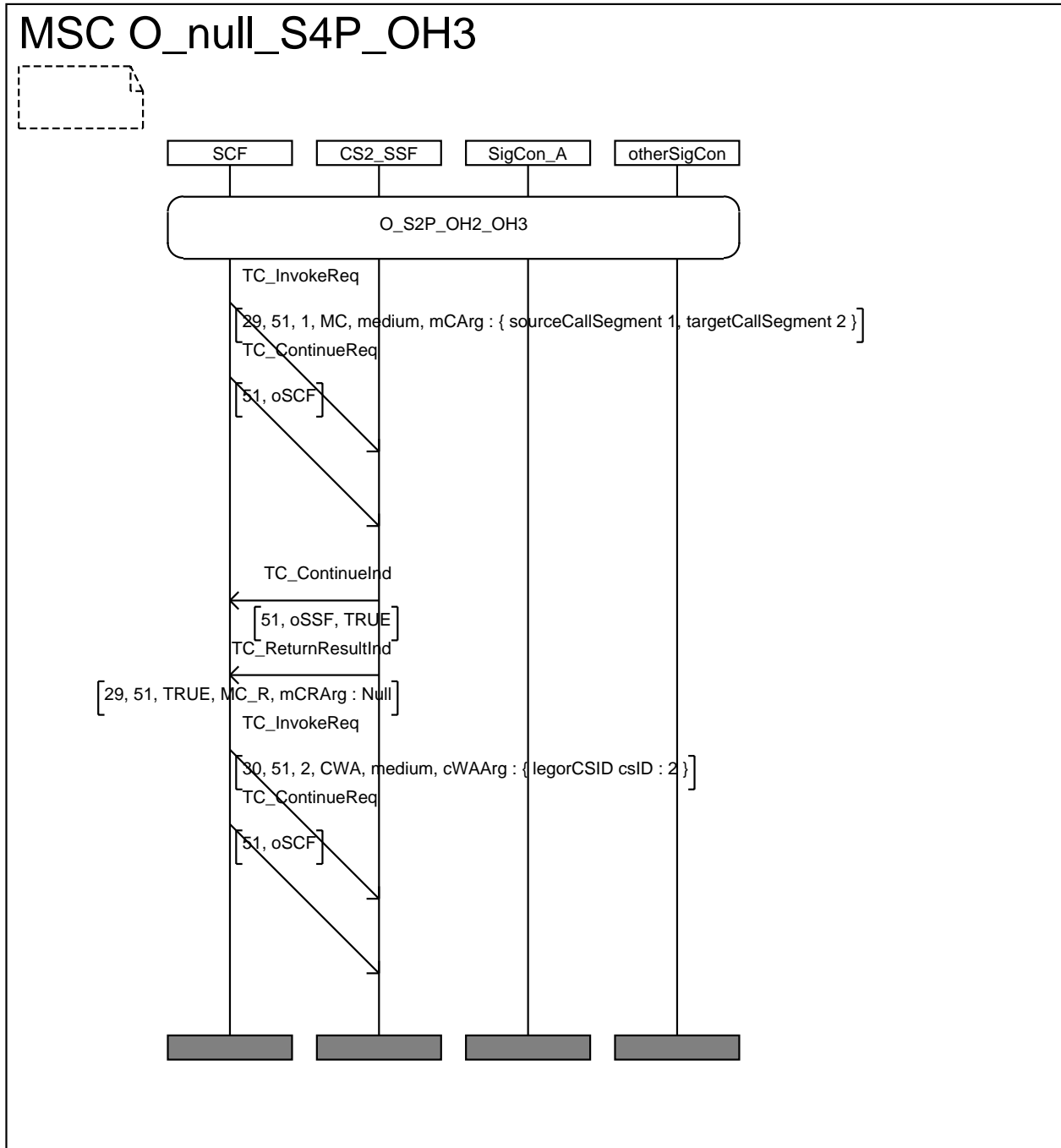
O\_S2P\_OH(2)\_OH(3)

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CsId=2)

O\_null\_S4P\_OH(3)



12 - Preamble O\_S2P\_OH(3)\_OH(3)

O\_null\_S4P\_OH(3)

L1! SplitLeg(1,1)

L1?SplitLegReturnResult

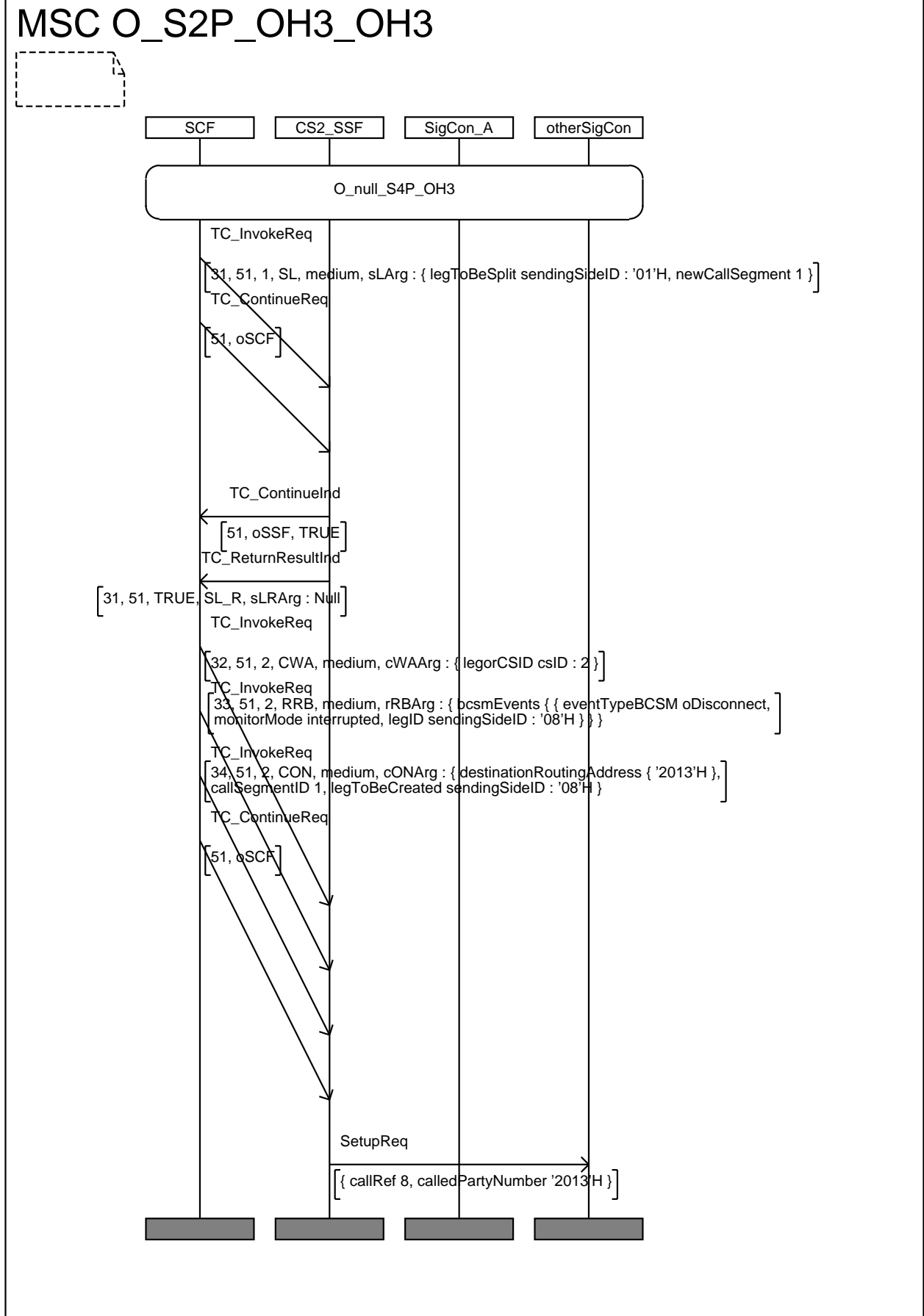
L1! ContinueWithArgument (CsID = 2)

L1! RequestReportBCSMEEvent(8,oDisconnect)

L1! Connect(8,1)

CP1-8? SetUpReq

O\_S2P\_OH(3)\_OH(3)



4-2 - Preamble O\_null\_S3P\_null

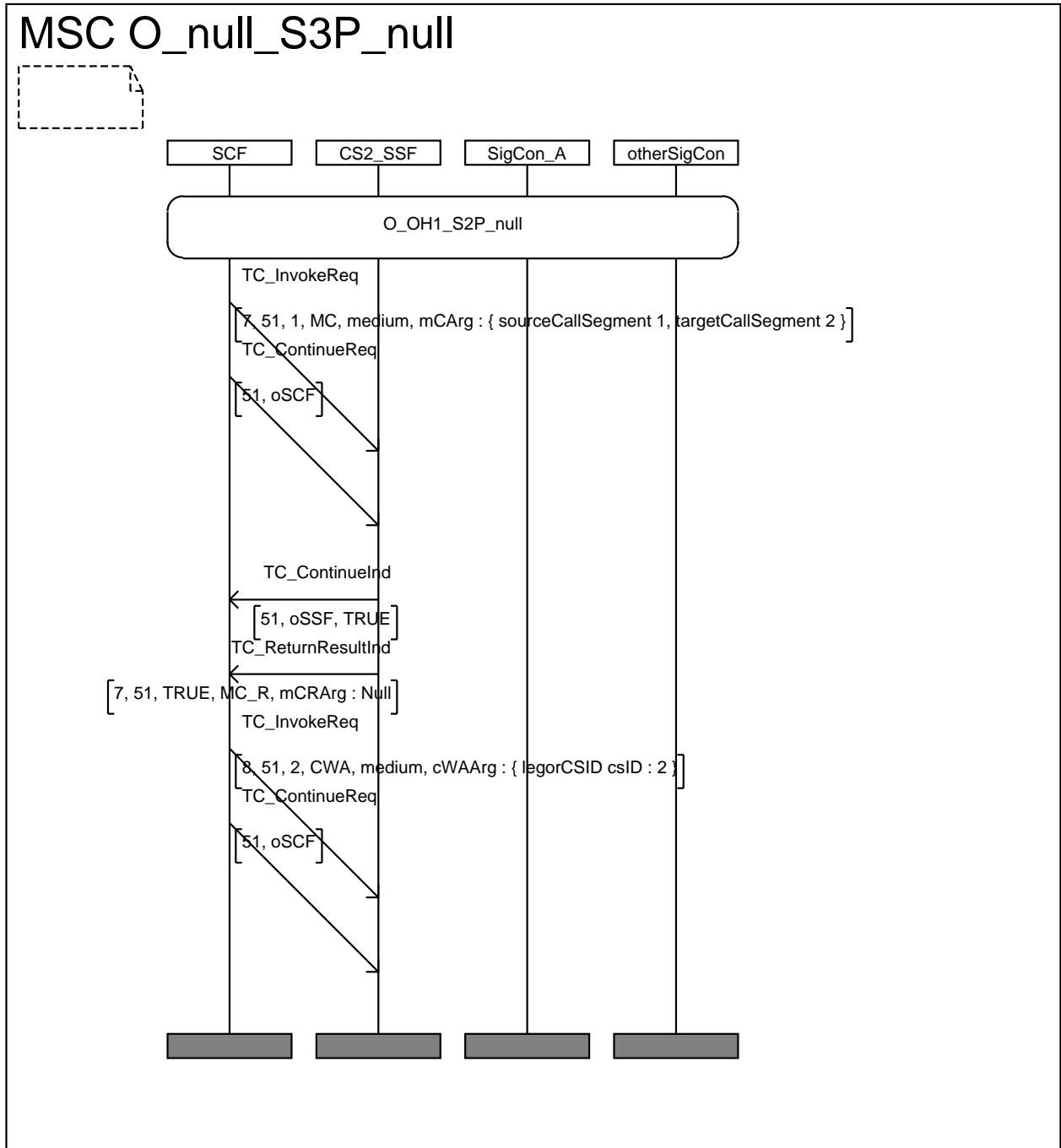
O\_OH(1)\_S2P\_null

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CsId=2)

O\_null\_S3P\_null



### 7.1.6.2 T (terminating) preamble tree

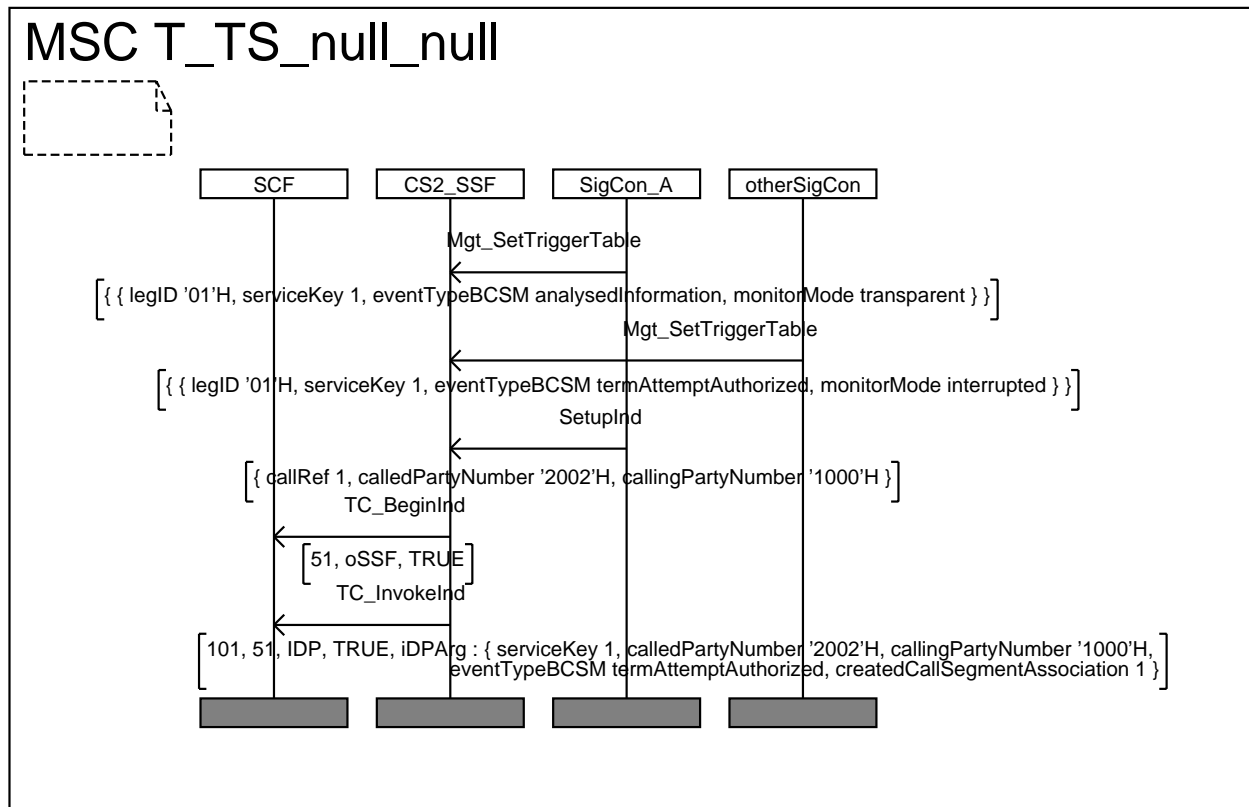
1-1 - Preamble T\_TS\_null\_null

T\_null\_null\_null

CP1-1! SetUpInd

L1? IDP(termAttemptAuthorized)

T\_TS\_null\_null



1-2 - Preamble T\_S2P\_null\_null

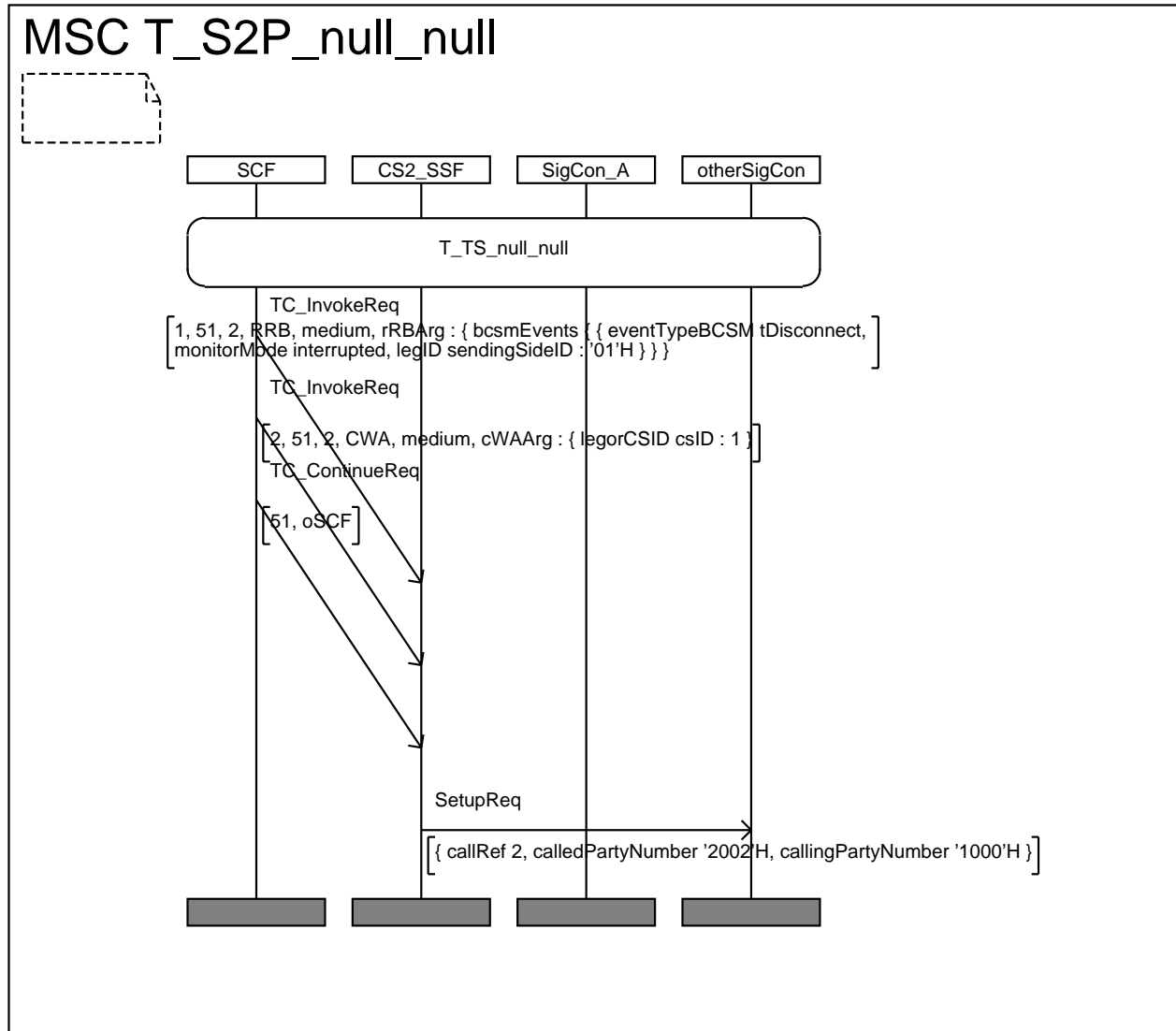
T\_TS\_null\_null

L1! RequestReportBCSM(1, tDisconnect)

L1! ContinueWithArgument(CsId=1)

CP1-2? SetUpReq

T\_S2P\_null\_null



1-3 - Preamble T\_OH(1)\_S2P\_null

T\_S2P\_null\_null

L1! SplitLeg(2,2)

L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 1)

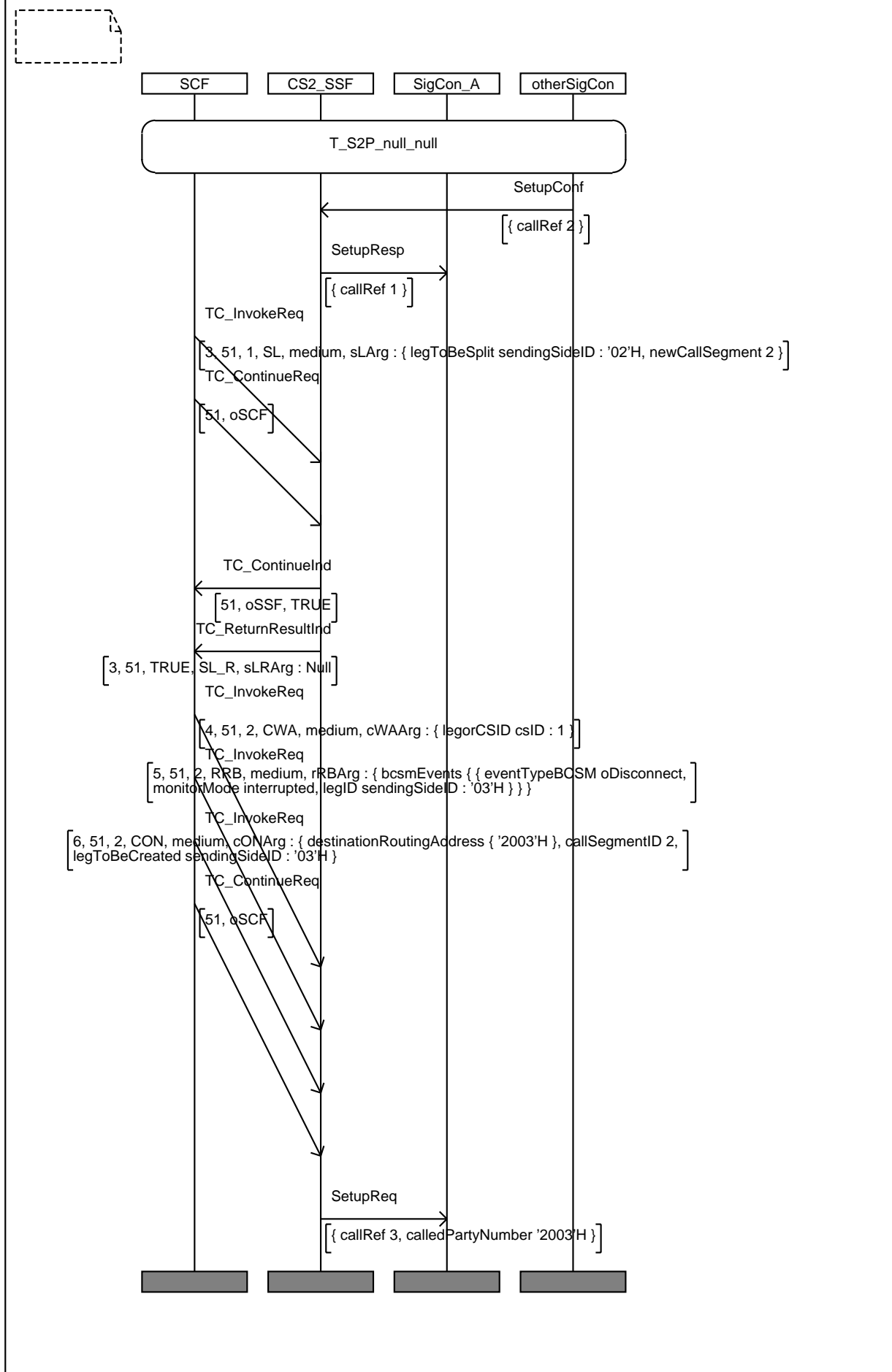
L1! RequestReportBCSMEvent(3,oDisconnect)

L1! Connect (3,2)

CP1-3? SetUpReq

T\_OH(1)\_S2P\_null

# MSC T\_OH1\_S2P\_null





1-4 - Preamble T\_null\_S3P\_null

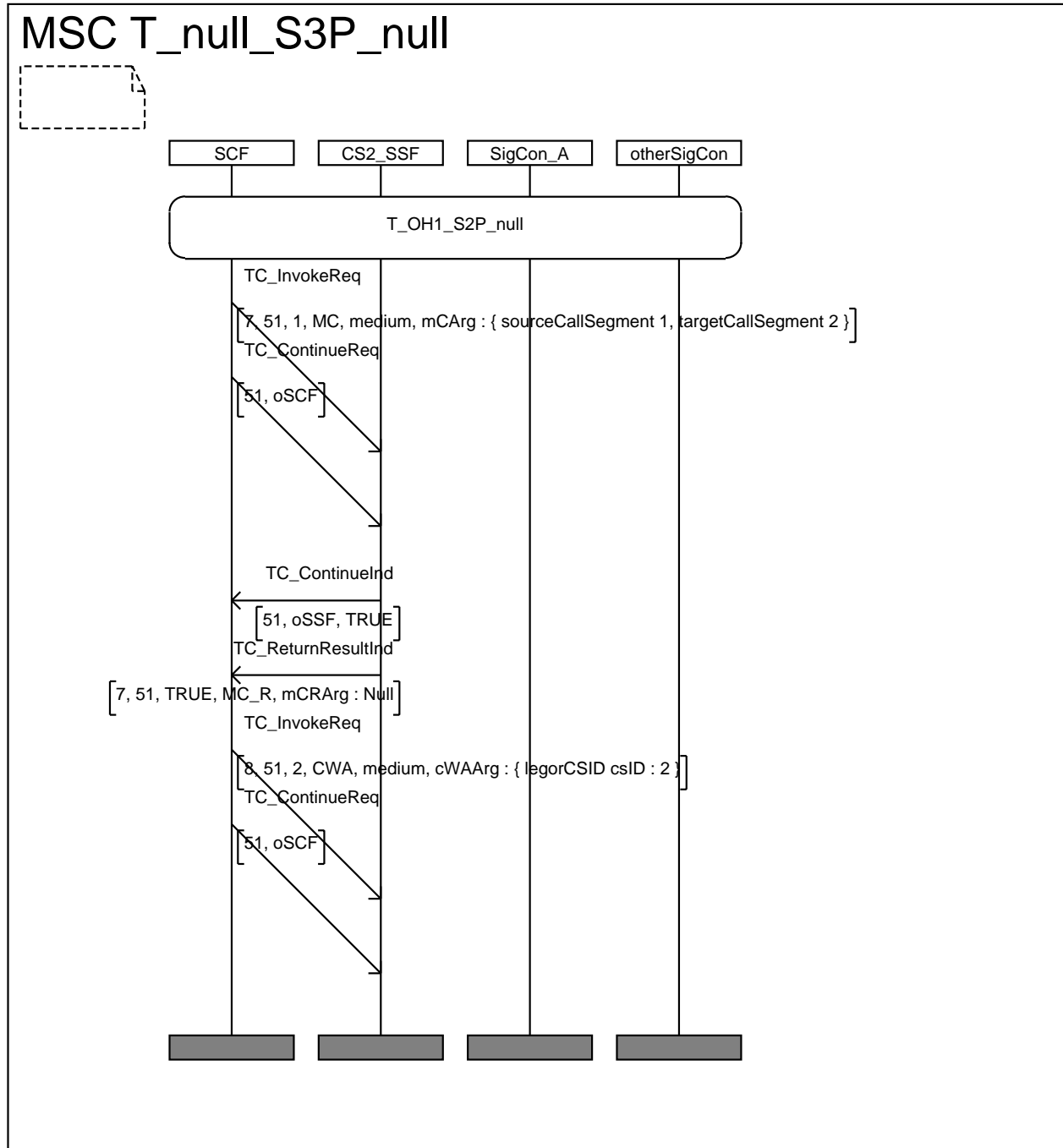
T\_OH(1)\_S2P\_null

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CsId=2)

T\_null\_S3P\_null



1-5 - Preamble T\_null\_OH(2)\_S2P

T\_null\_S3P\_null

L1! SplitLeg(2,3)

L1?SplitLegReturnResult

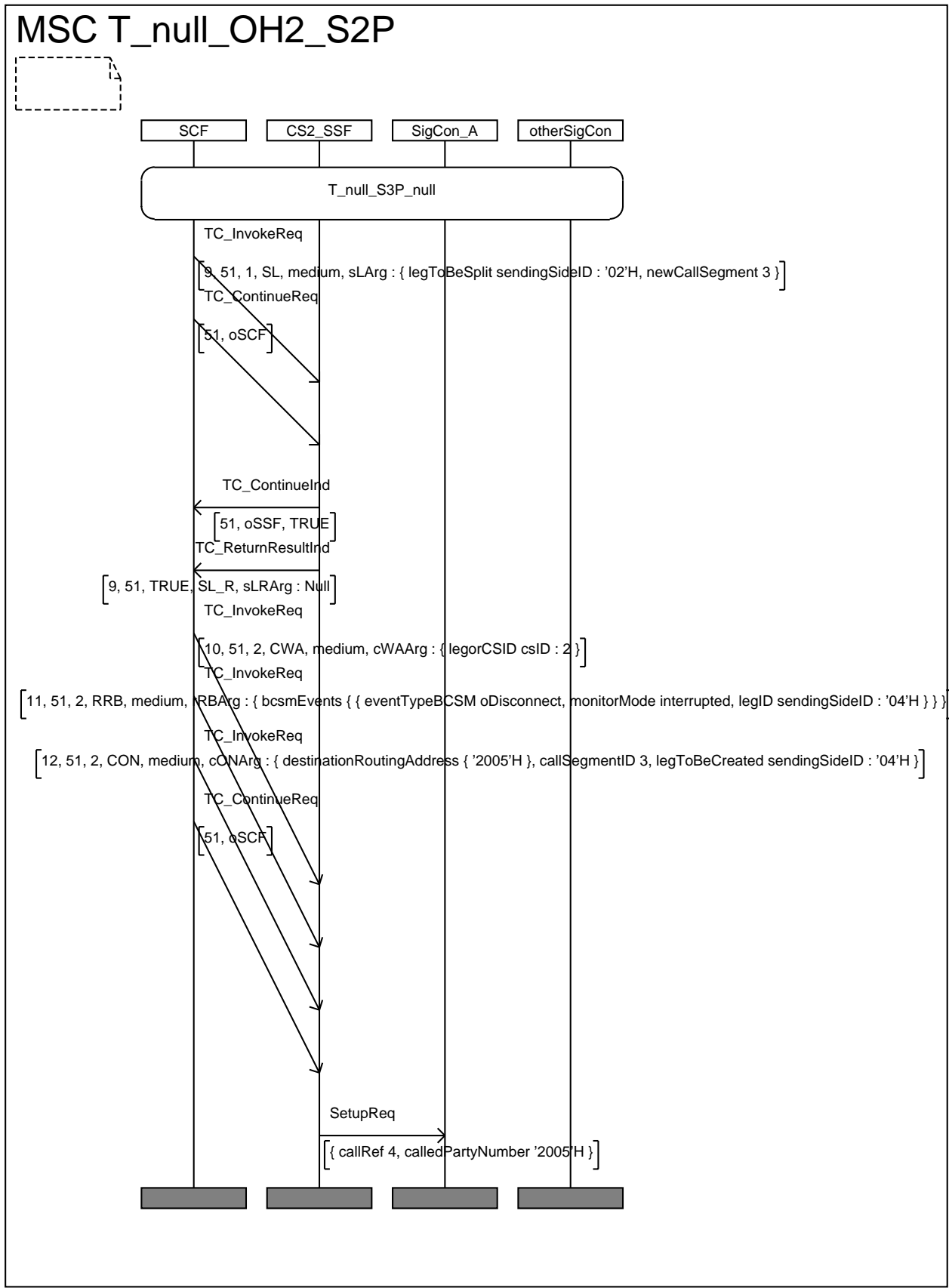
L1! ContinueWithArgument (CsID = 2)

L1! RequestReportBCSMEEvent(4,tDisconnect)

L1! Connect(4,3)

CP1-4? SetUpReq

T\_null\_OH(2)\_S2P



1-6 - Preamble T\_null\_null\_S4P

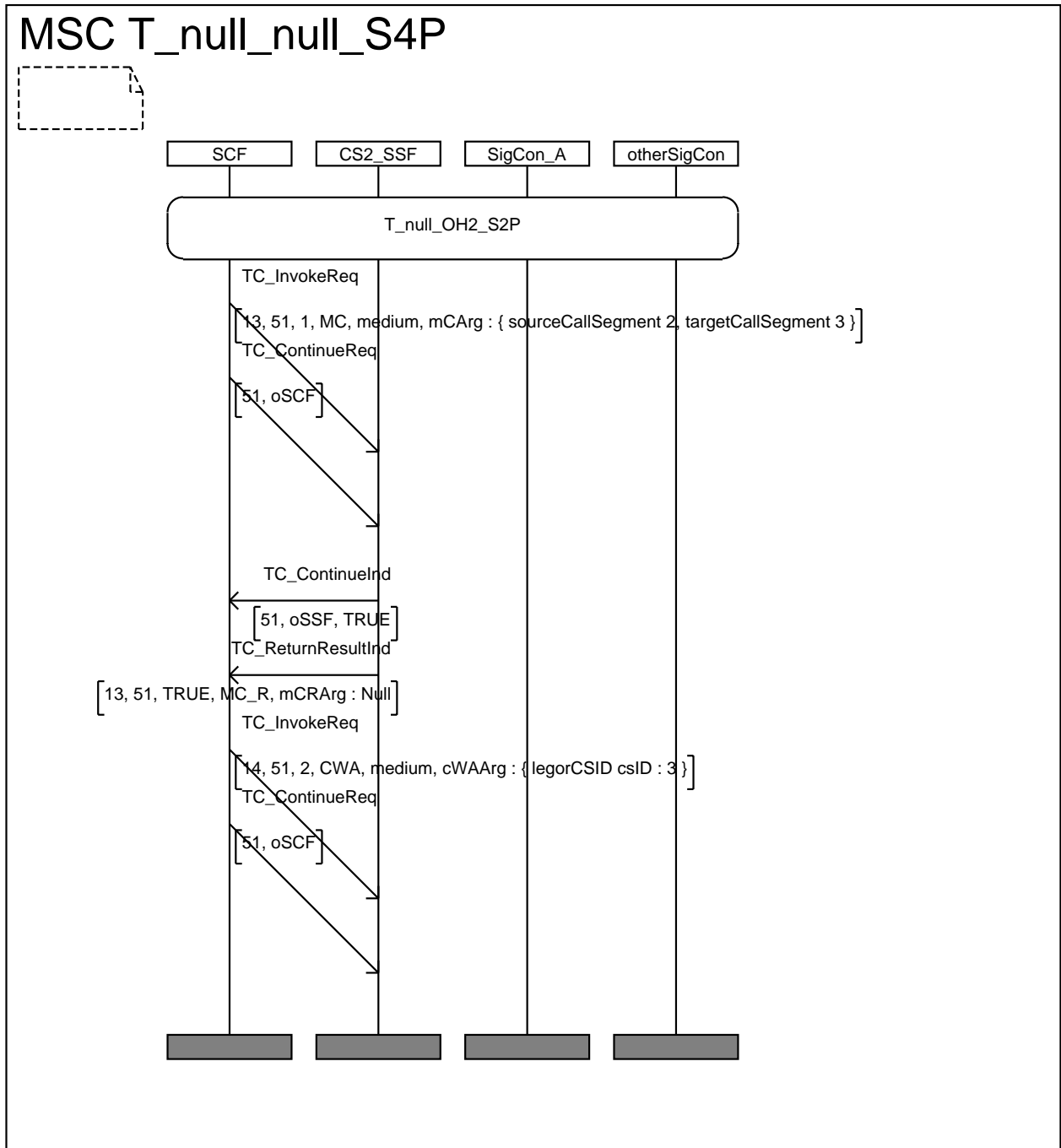
T\_null\_OH(2)\_S2P

L1! MergeCallSegments(2,3)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSID=3)

T\_null\_null\_S4P



2-1 Preamble T\_TF2\_null\_null

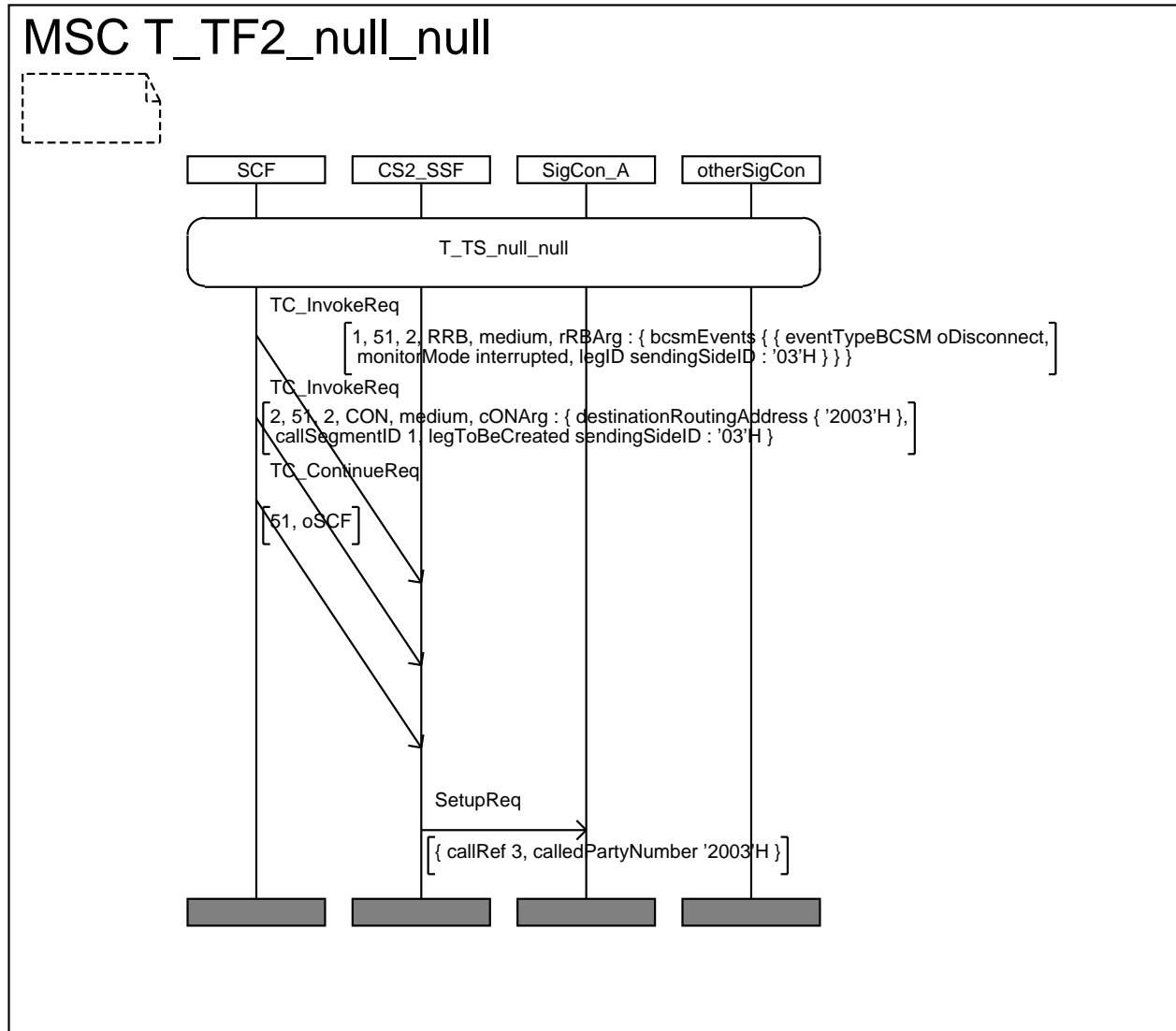
T\_TS\_null\_null

L1! RequestReportBCSMEEvent(3,tDisconnect)

L1! Connect (3,1)

CP1-3? SetUpReq

T\_TF2\_null\_null



### 7.1.6.3 I (InitiateCallAttempt) preamble tree

1 - Preamble I\_S1P\_null\_null

null

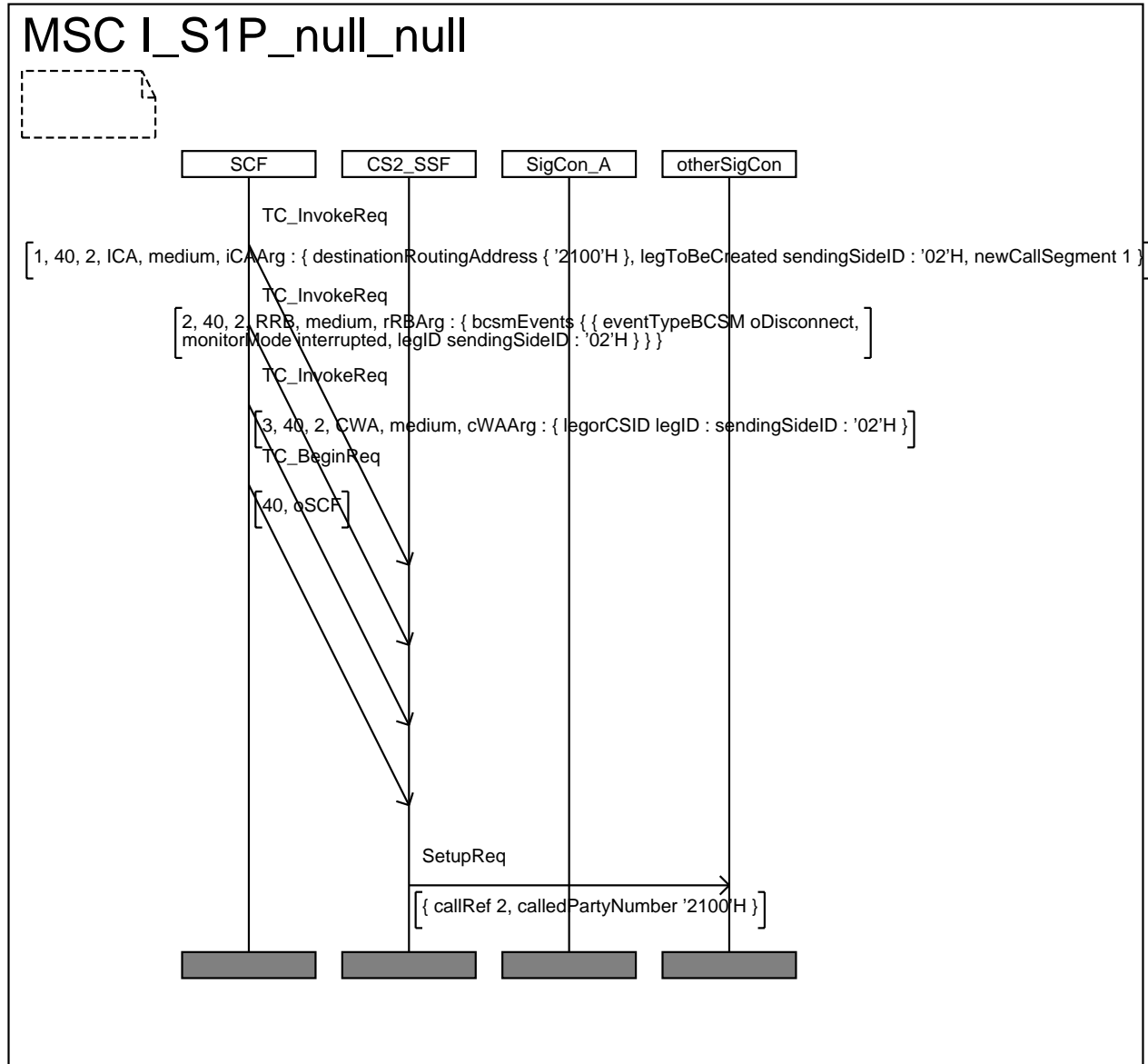
L1! InitiateCallAttempt(2,1)

L1! RequestReportBCSMEvent(2,oDisconnect)

L1! ContinueWithArgument(LegId=2)

CP1-2? SetUpReq

I\_S1P\_null\_null



2 - Preamble I\_S1P\_S1P\_null

I\_S1P\_null\_null

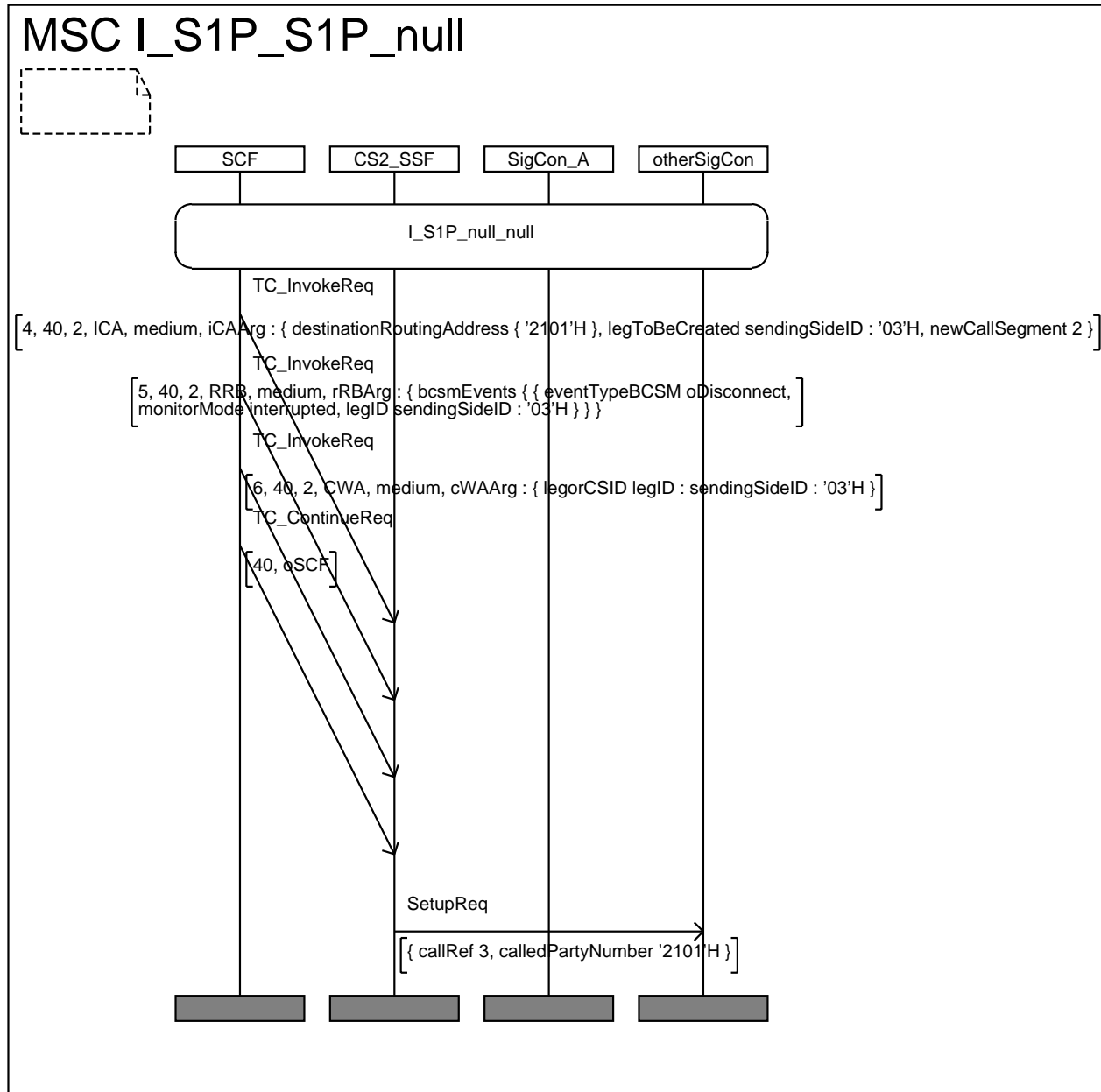
L1! InitiateCallAttempt(3,2)

L1! RequestReportBCSMEvent(3,oDisconnect)

L1! ContinueWithArgument(LegId=3)

CPI-3? SetUpReq

I\_S1P\_S1P\_null



3-1 - Preamble I\_S1P\_S1P\_S1P

I\_S1P\_S1P\_null

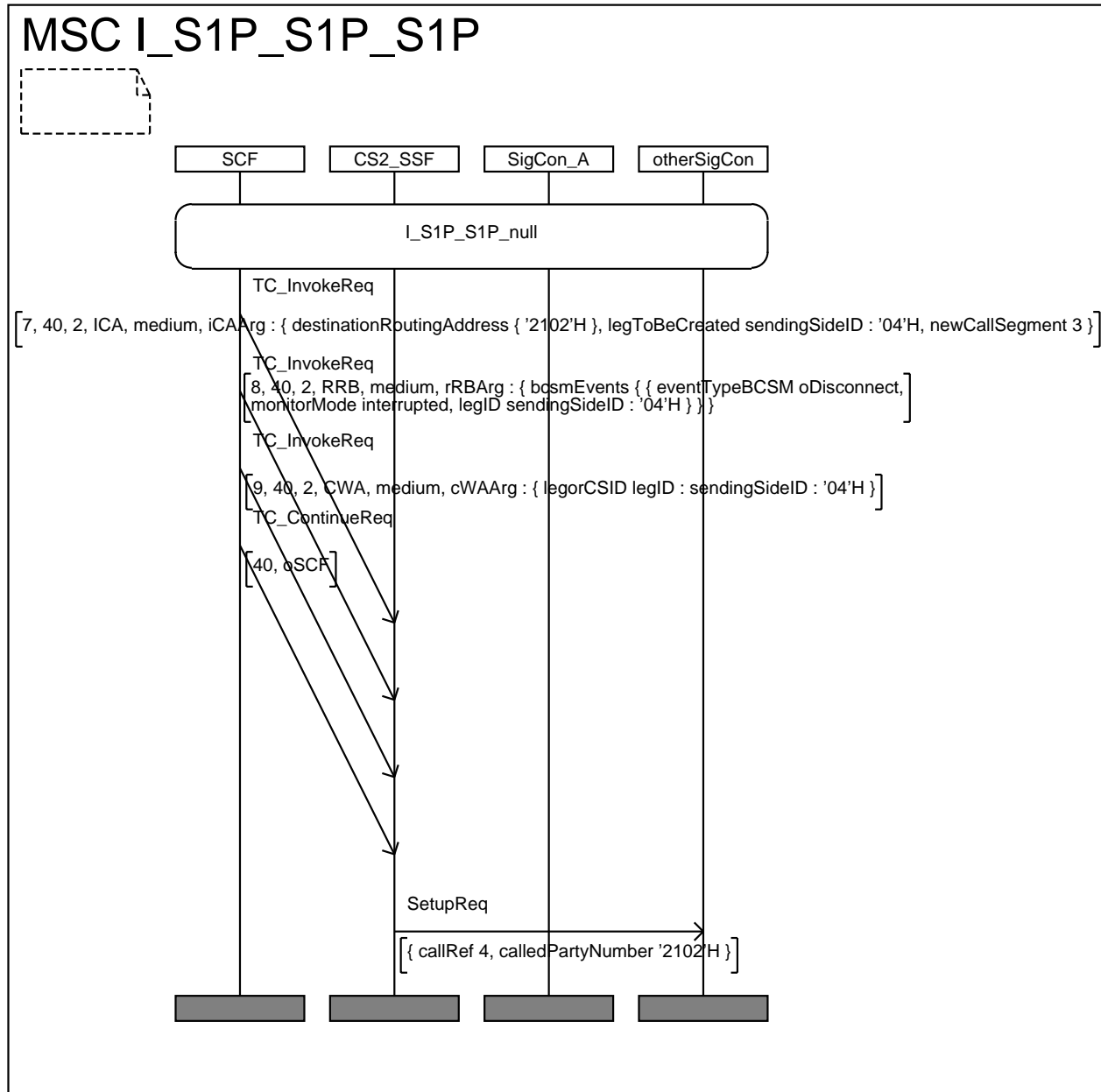
L1! InitiateCallAttempt(4,3)

L1! RequestReportBCSMEvent(4,oDisconnect)

L1! ContinueWithArgument(LegId=4)

CP1-4? SetUpReq

I\_S1P\_S1P\_S1P



4 - Preamble I\_null\_TF(2)\_S1P

I\_S1P\_S1P\_S1P

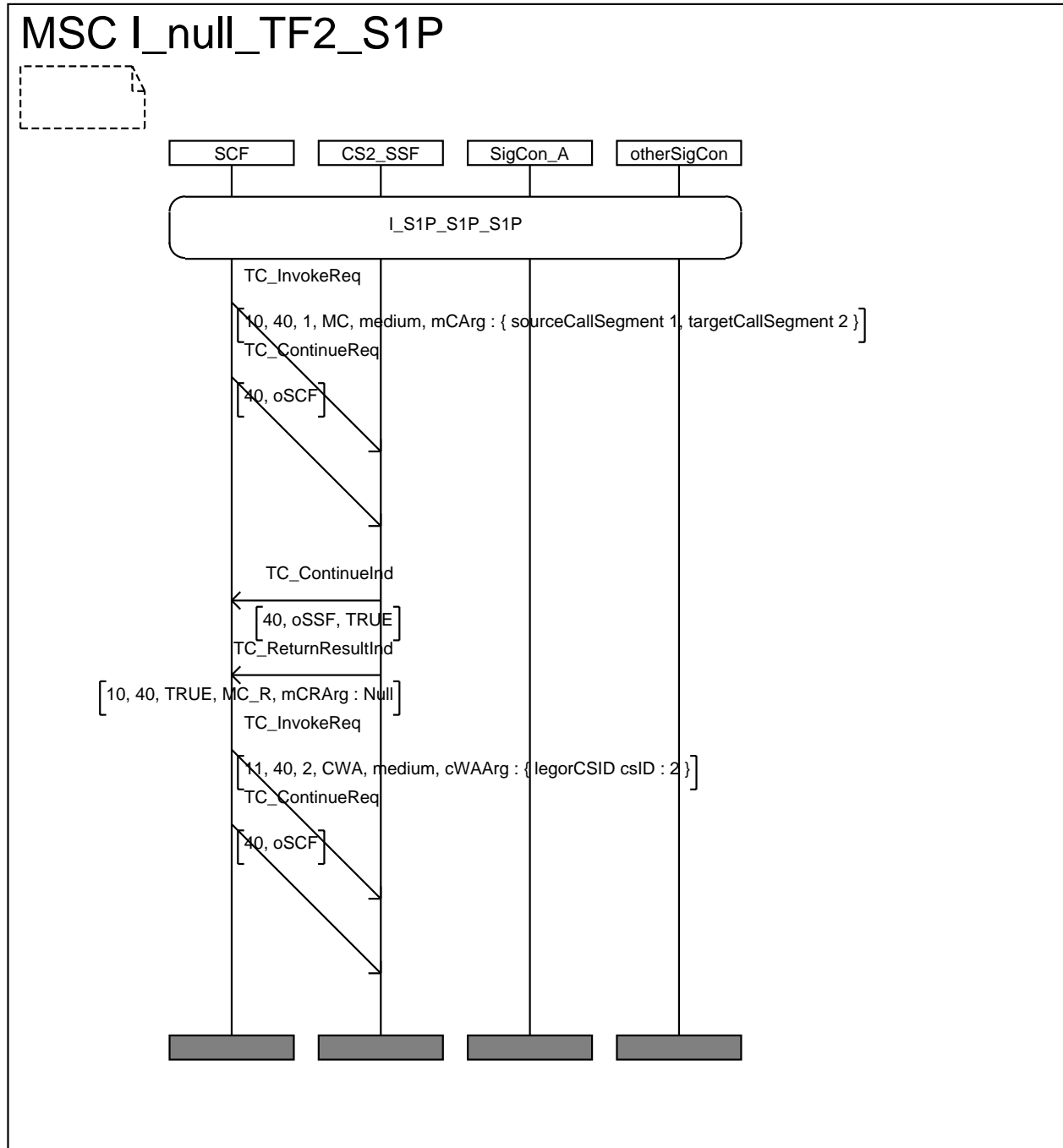
L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=2)

I\_null\_TF(2)\_S1P





5 - Preamble I\_null\_null\_TF(3)

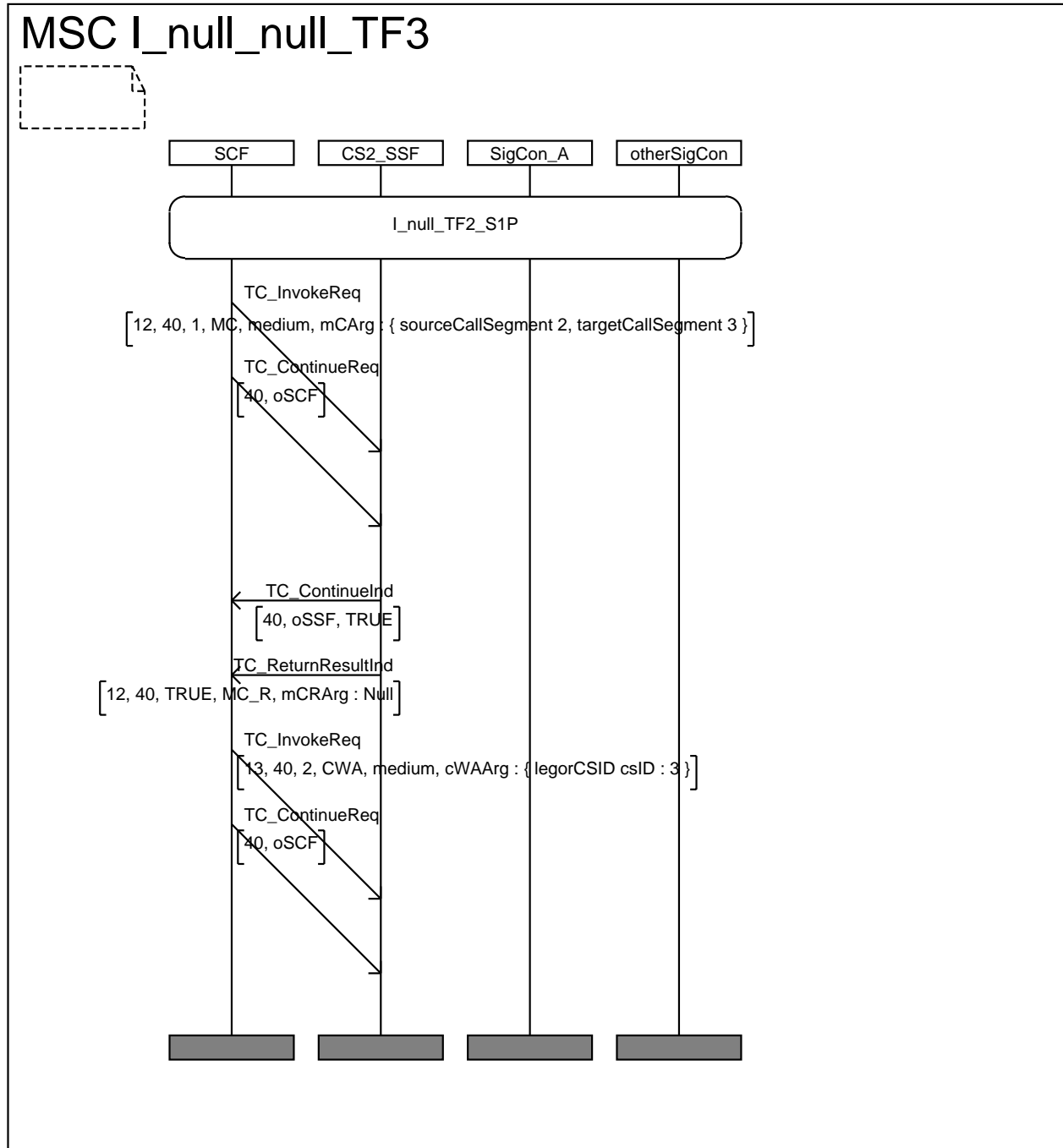
I\_null\_TF(2)\_S1P

L1! MergeCallSegments(2,3)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=3)

I\_null\_null\_TF(3)



3-2 - Preamble I\_null\_TF(2)\_null

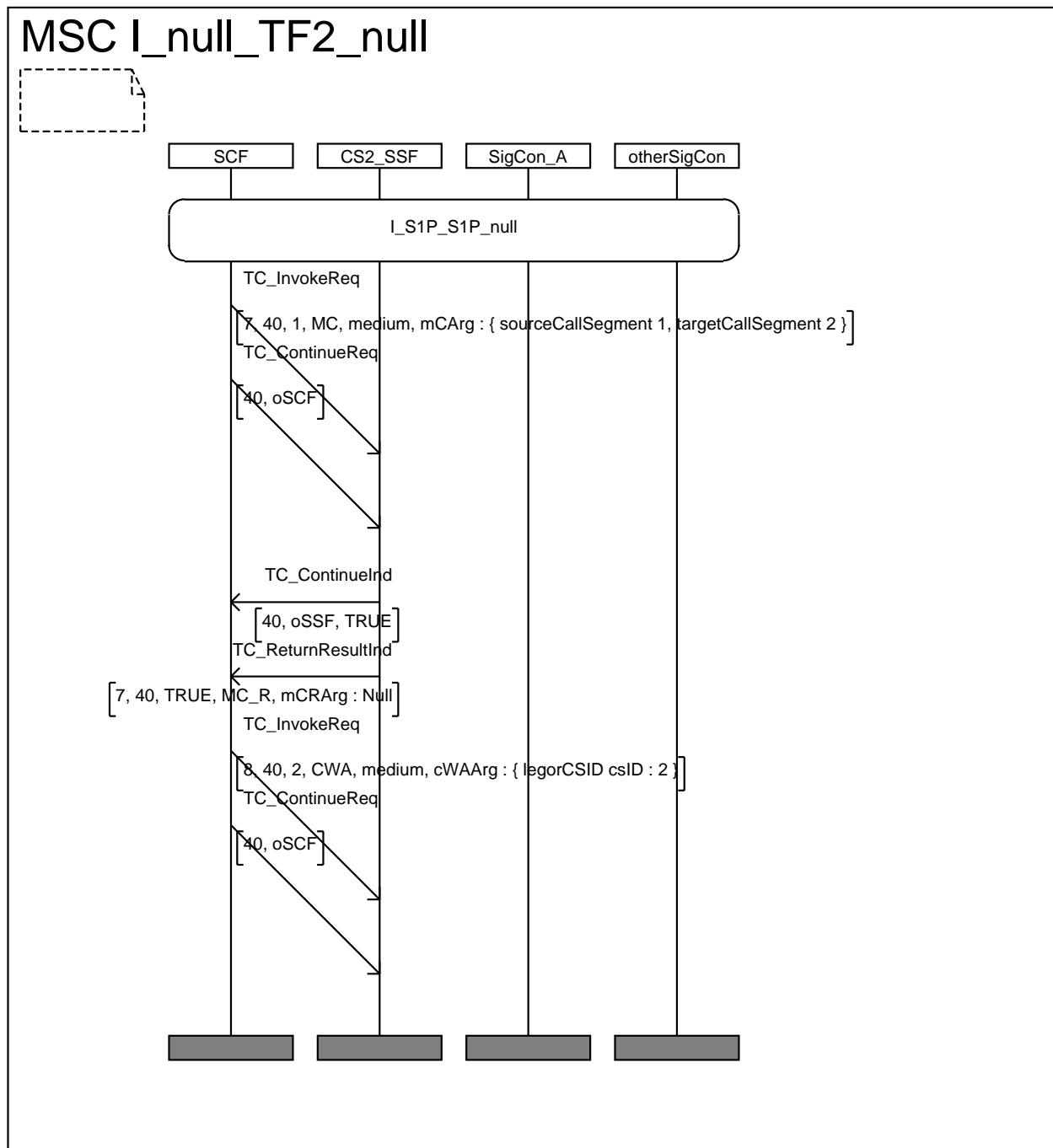
I\_S1P\_S1P\_null

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSID=2)

I\_null\_TF(2)\_null



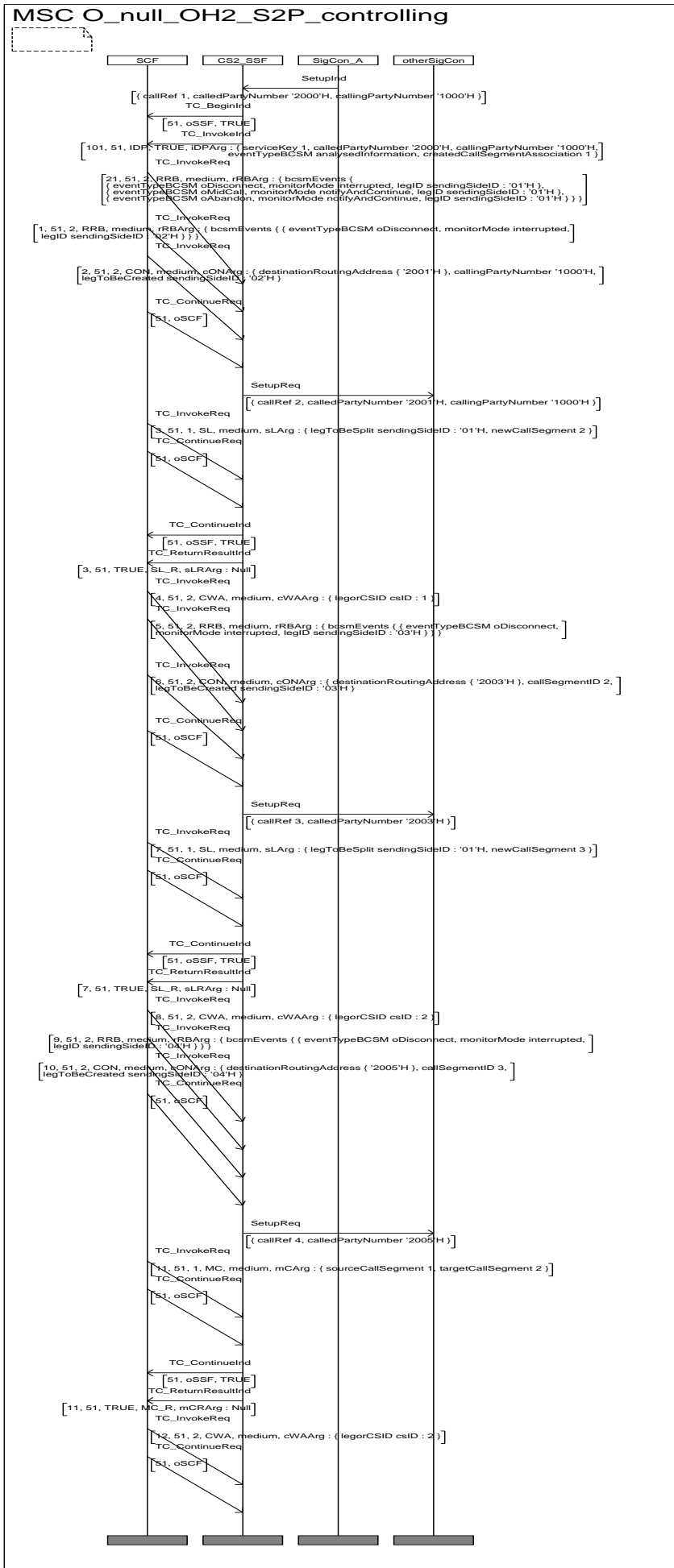
#### 7.1.6.4 Event Detecting/Report rules Preambles

With the aim of testing the event detecting/report rules these preambles arm the legs in order to detect a signalling event coming from the controlling leg or from the passive legs. The steps followed to reach the final configuration are the same as the ones used in the previous preambles (originating, terminating, network initiated). What is different is the number of events armed when a new leg is added to the CallSegment configuration.

##### Preamble O\_null\_OH(2)\_S2P\_controlling

The controlling leg (legId=1) is armed with the events:

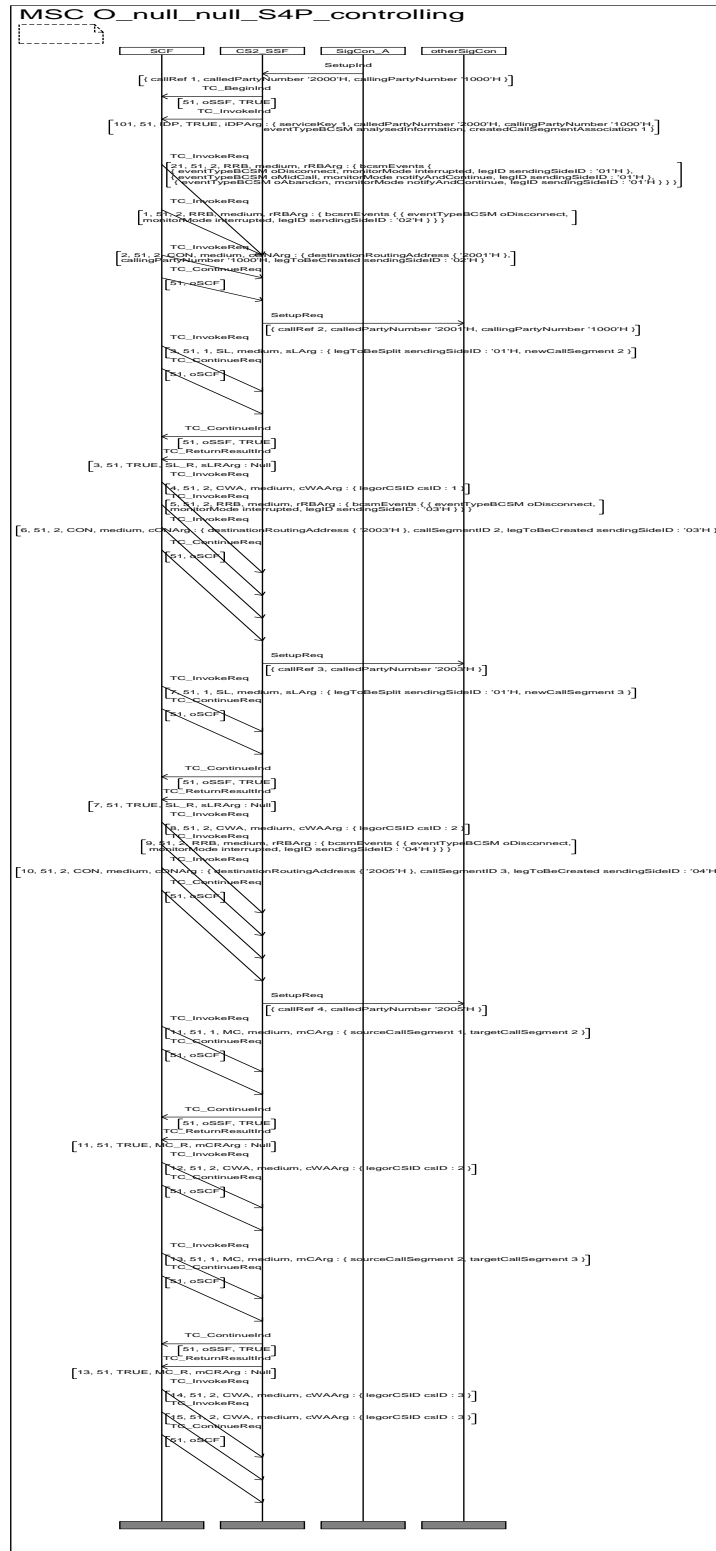
- oDisconnect;
- oAbandon;
- oMidCall.



**Preamble O\_null\_null\_S4P\_controlling**

The controlling leg (legId=1) is armed with the events:

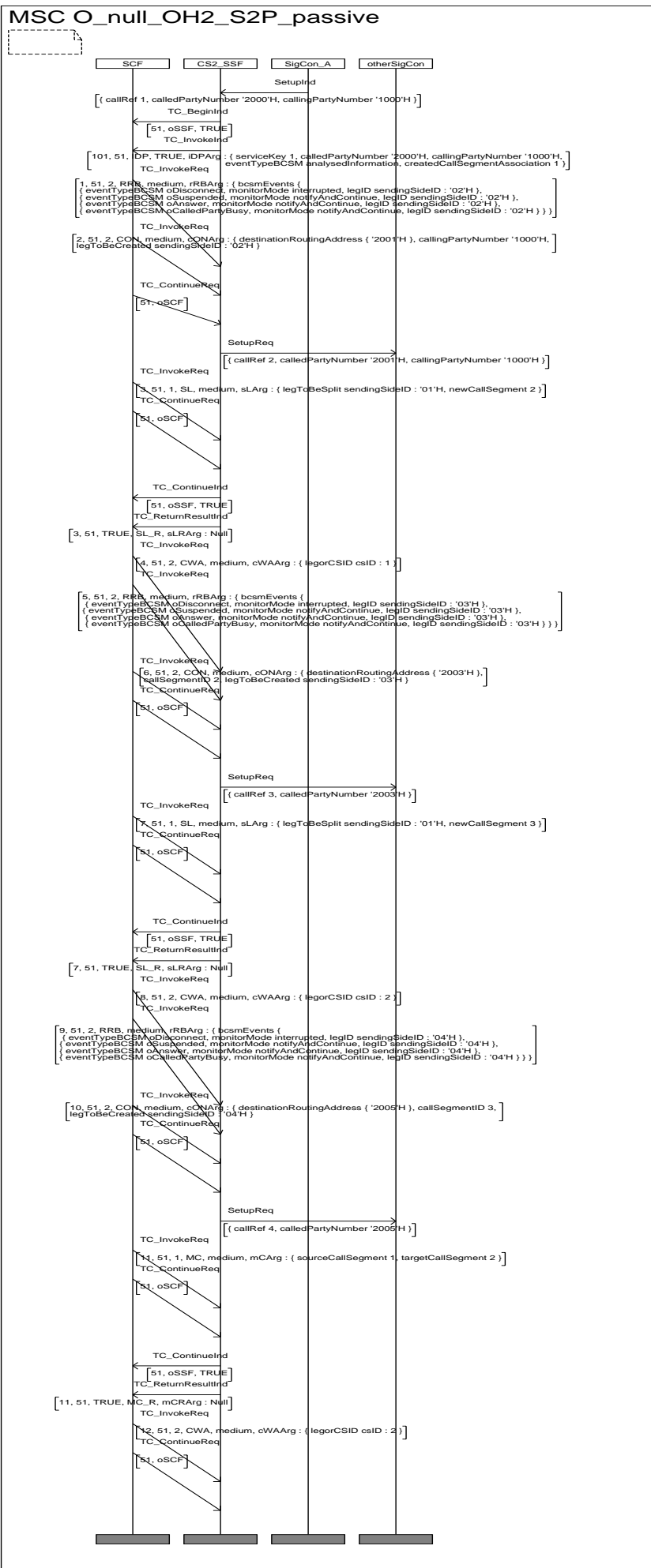
- oDisconnect;
- oAbandon;
- oMidCall.



**Preamble O\_null\_OH(2)\_S2P\_passive**

The passive legs (legId=2,3,4) are armed with the events:

- oDisconnect;
- oAnswer;
- oCalledPartyBusy;
- oSuspended.



**Preamble O\_null\_null\_S4P\_passive**

The passive legs (legId=2,3,4) are armed with the events:

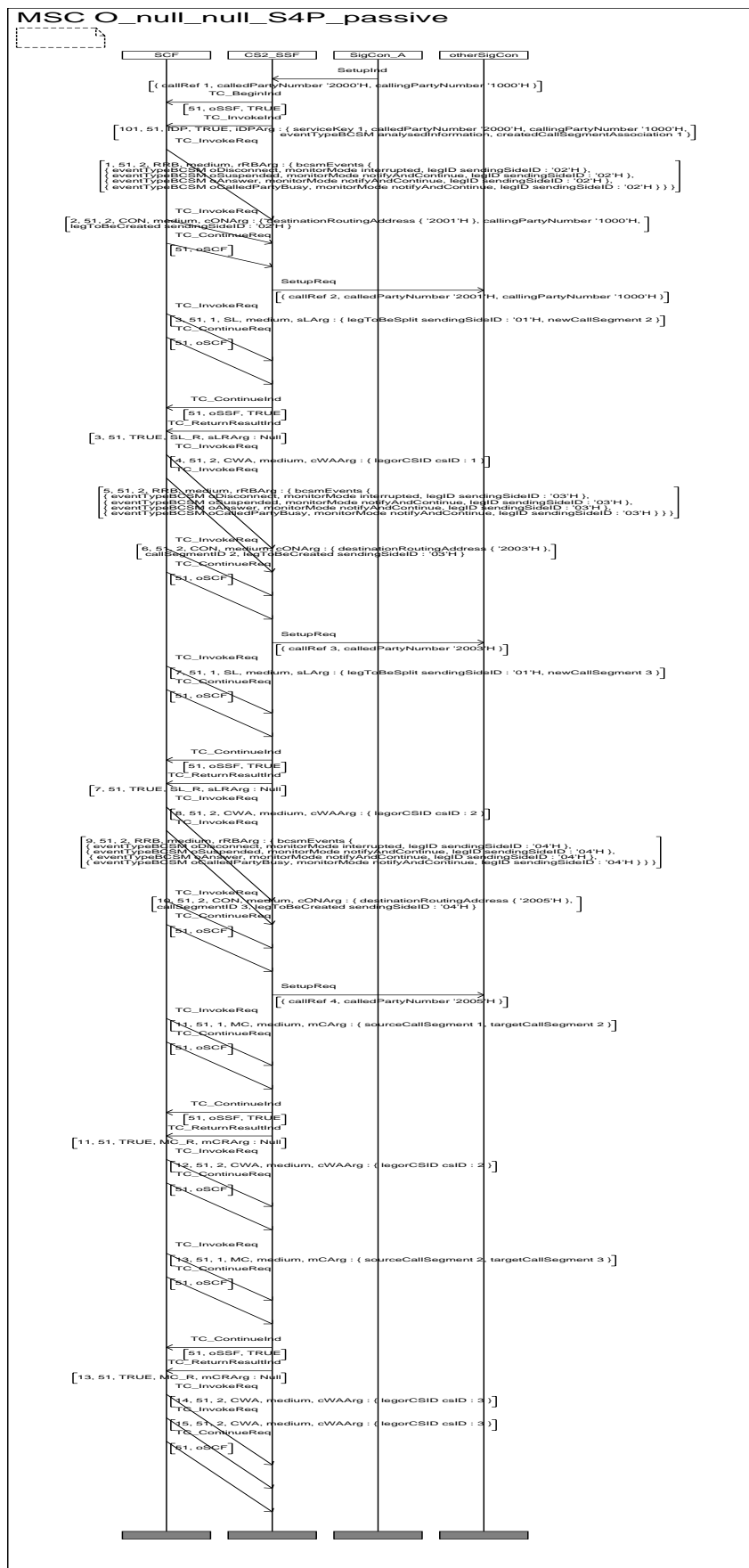
- oDisconnect;

- oAnswer;

- oCalledPartyBusy;

- oSuspended.





Preamble I\_S1P\_S1P\_null\_passive

The passive legs (legId=2,3) are armed with the events:

oDisconnect;

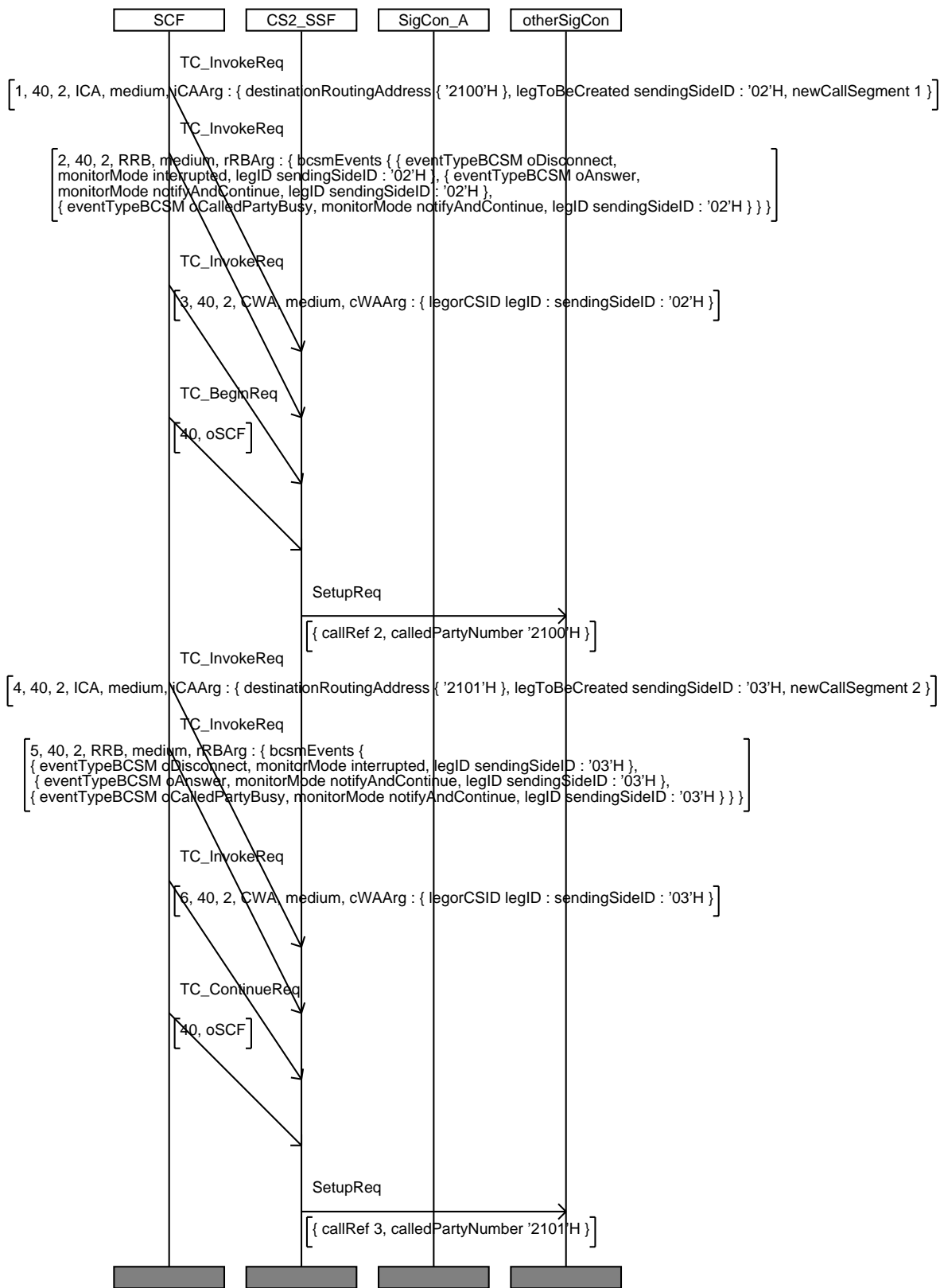
oMidCall;

oAnswer;

oCalledPartyBusy;

oSuspended.

# MSC I\_S1P\_S1P\_null\_passive



**Preamble I\_null\_TF(2)\_null\_passive**

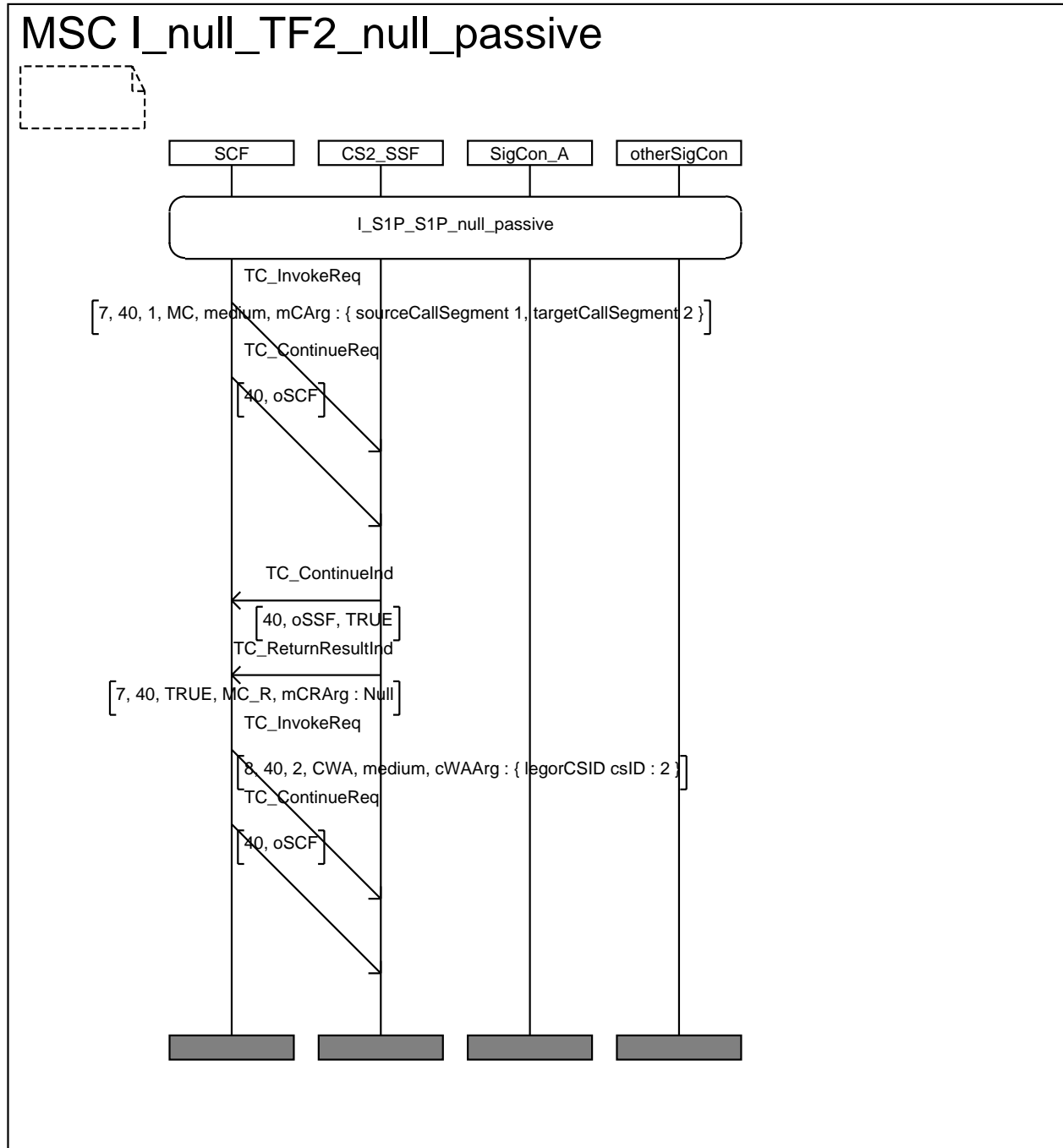
The passive legs (legId=2,3) are armed with the events:

- oDisconnect;

- oAnswer;

- oCalledPartyBusy;

- oSuspended.

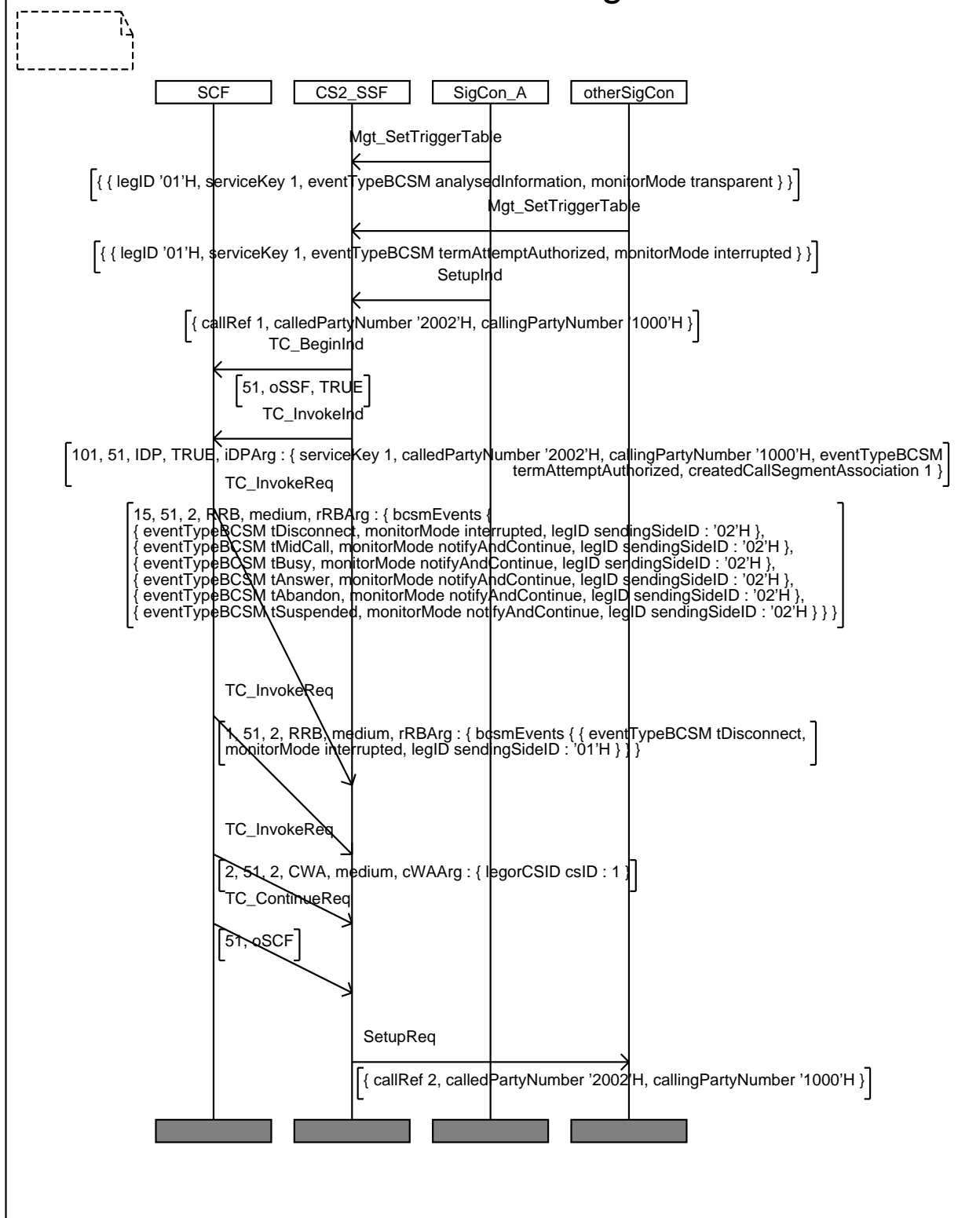


### Preamble T\_S2P\_null\_null\_controlling

The controlling leg (legId=2) is armed with the events:

- tDisconnect;
- tMidCall;
- tAnswer;
- tCalledPartyBusy;
- tSuspended.

## MSC T\_S2P\_null\_null\_controlling



### Preamble T\_null\_null\_S4P\_controlling

The controlling leg (legId=2) is armed with the events:

- tDisconnect;
- tMidCall;
- tAnswer;

tCalledPartyBusy;

tSuspended.

**Preamble T\_S2P\_null\_null\_passive**

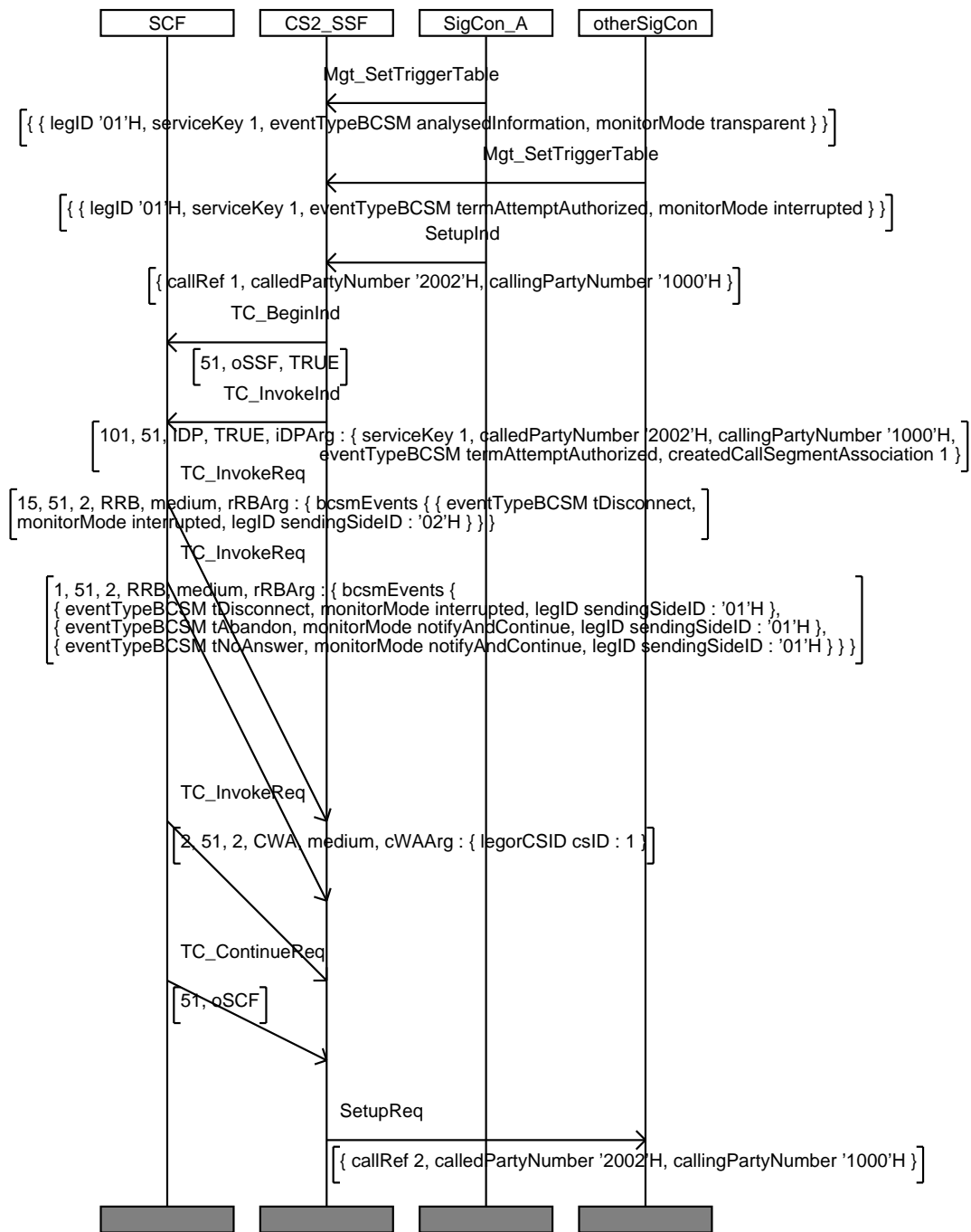
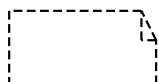
The passive leg (legId=1) is armed with the events:

tDisconnect;

tAbandon;

tAnswer.

# MSC T\_S2P\_null\_null\_passive





**Preamble T\_null\_null\_S4P\_passive**

The passive legs (legId=1,3,4) are armed with the events:

t/oDisconnect;

t/oAbandon;

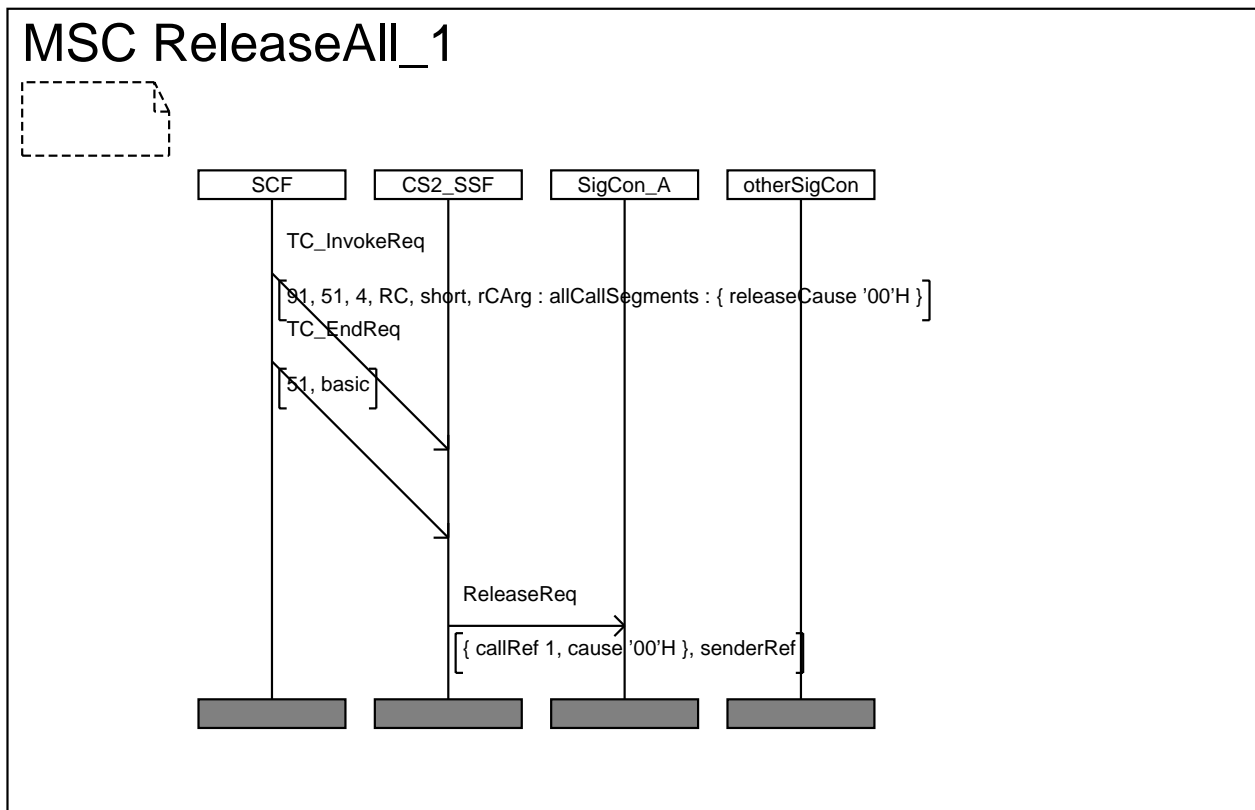
t/oAnswer.



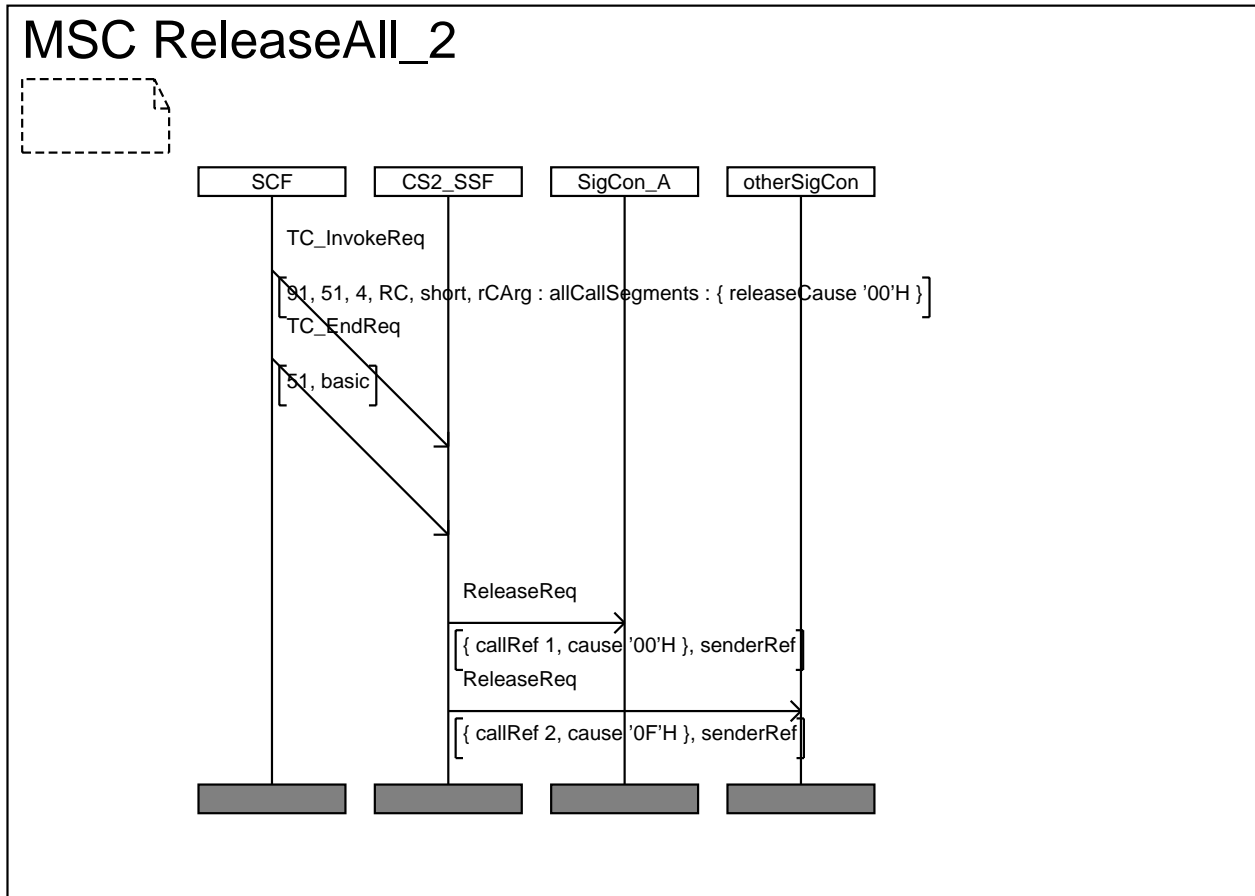
## 7.1.7 Postamble descriptions

Postambles are used to bring the IUT from the state where the test takes place back to the initial state. CPH is using a ReleaseAll postamble, applying to the number of legs active in the call, either 1 or 2 or 3, up to 7. The SCF sends a ReleaseCall operation and the IUT sends ReleaseReq on the legs that are active. The MSCs do not show which exact leg numbers are active, for instance either 123 or 234 in the case where 3 legs are concerned. However, the TTCN generated from these MSCs take the real numbers into account.

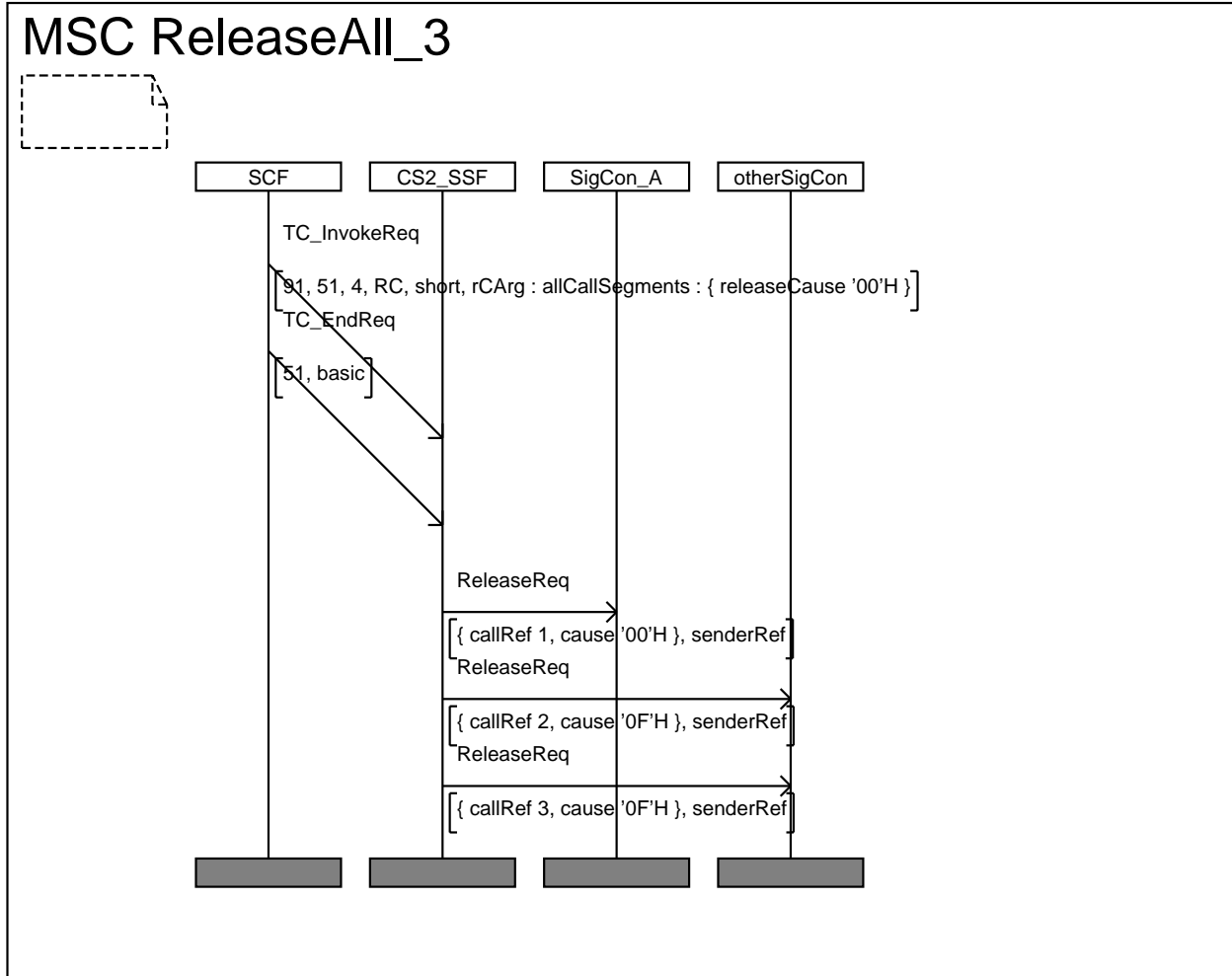
### 7.1.7.1 Postamble ReleaseAll\_1



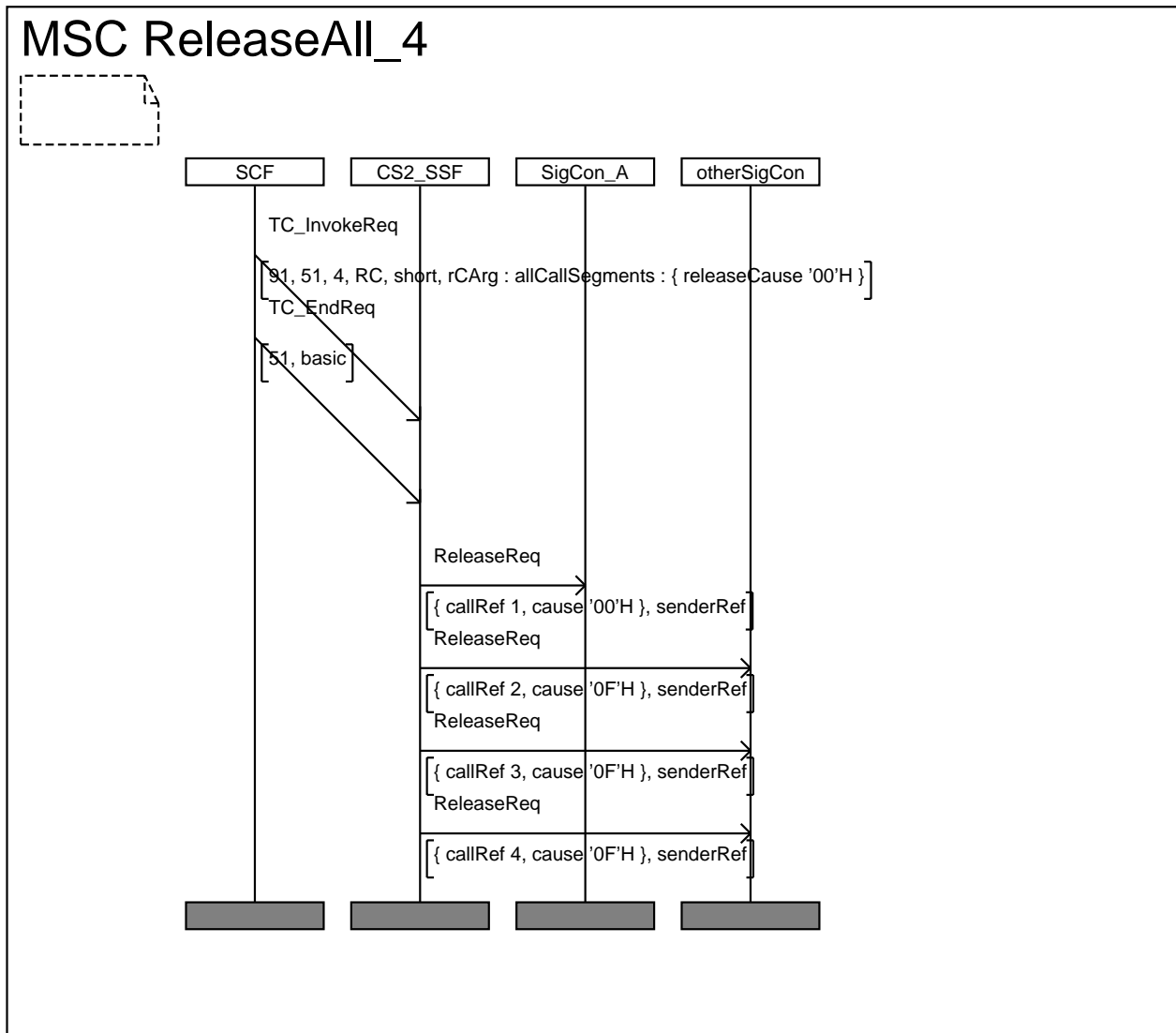
## 7.1.7.2 Postamble ReleaseAll\_2



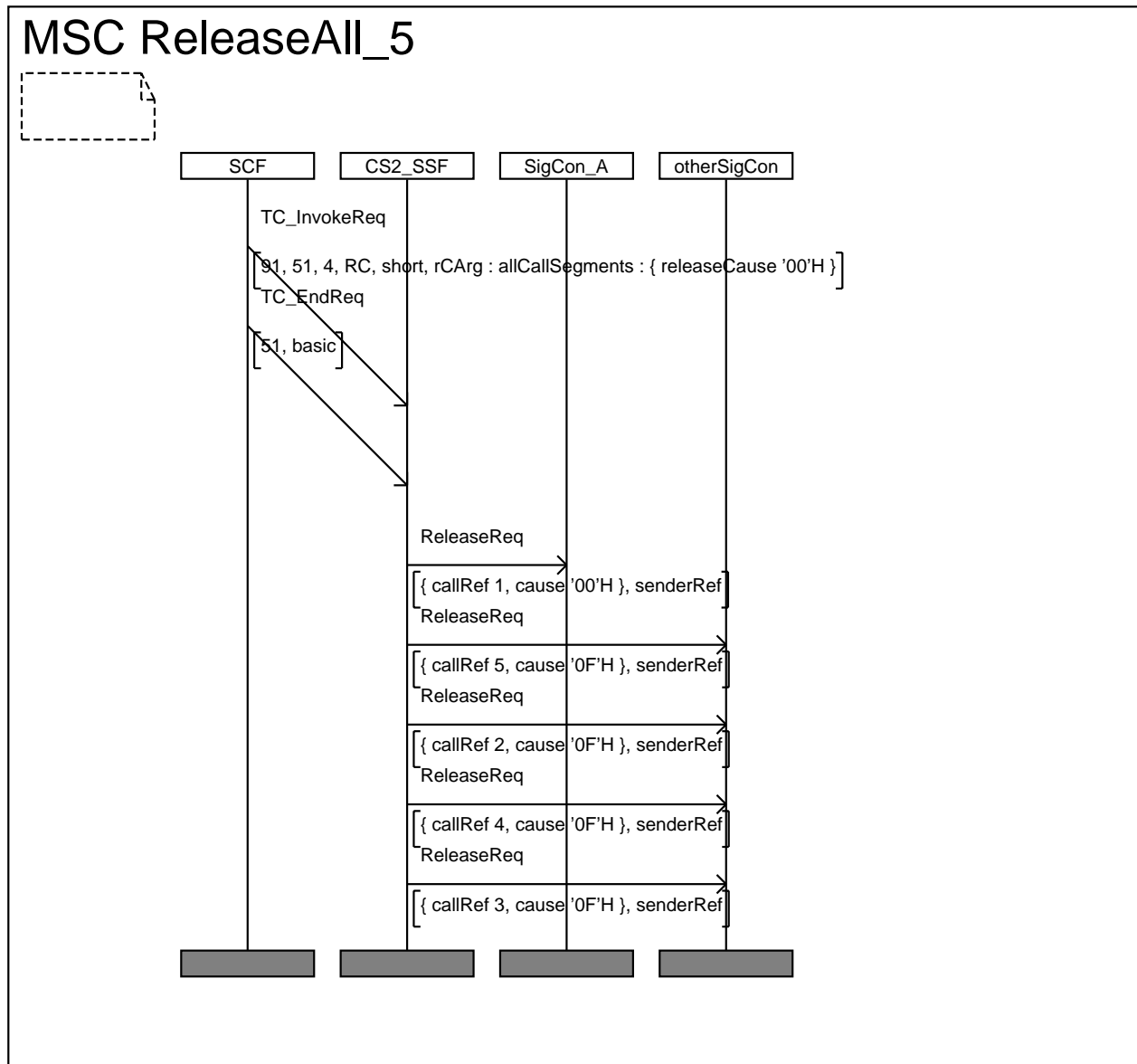
## 7.1.7.3 Postamble ReleaseAll\_3



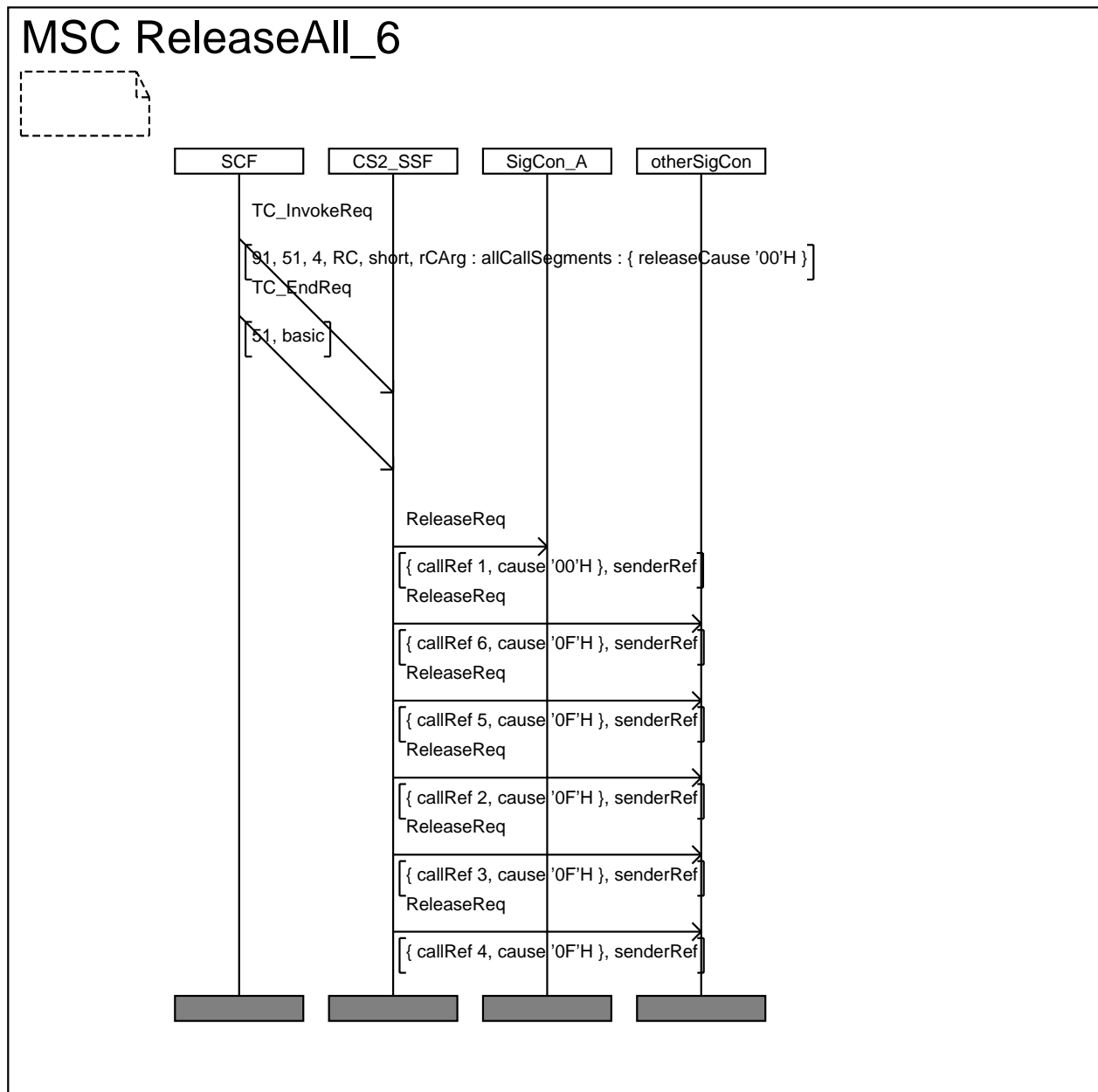
## 7.1.7.4 Postamble ReleaseAll\_4



## 7.1.7.5 Postamble ReleaseAll\_5

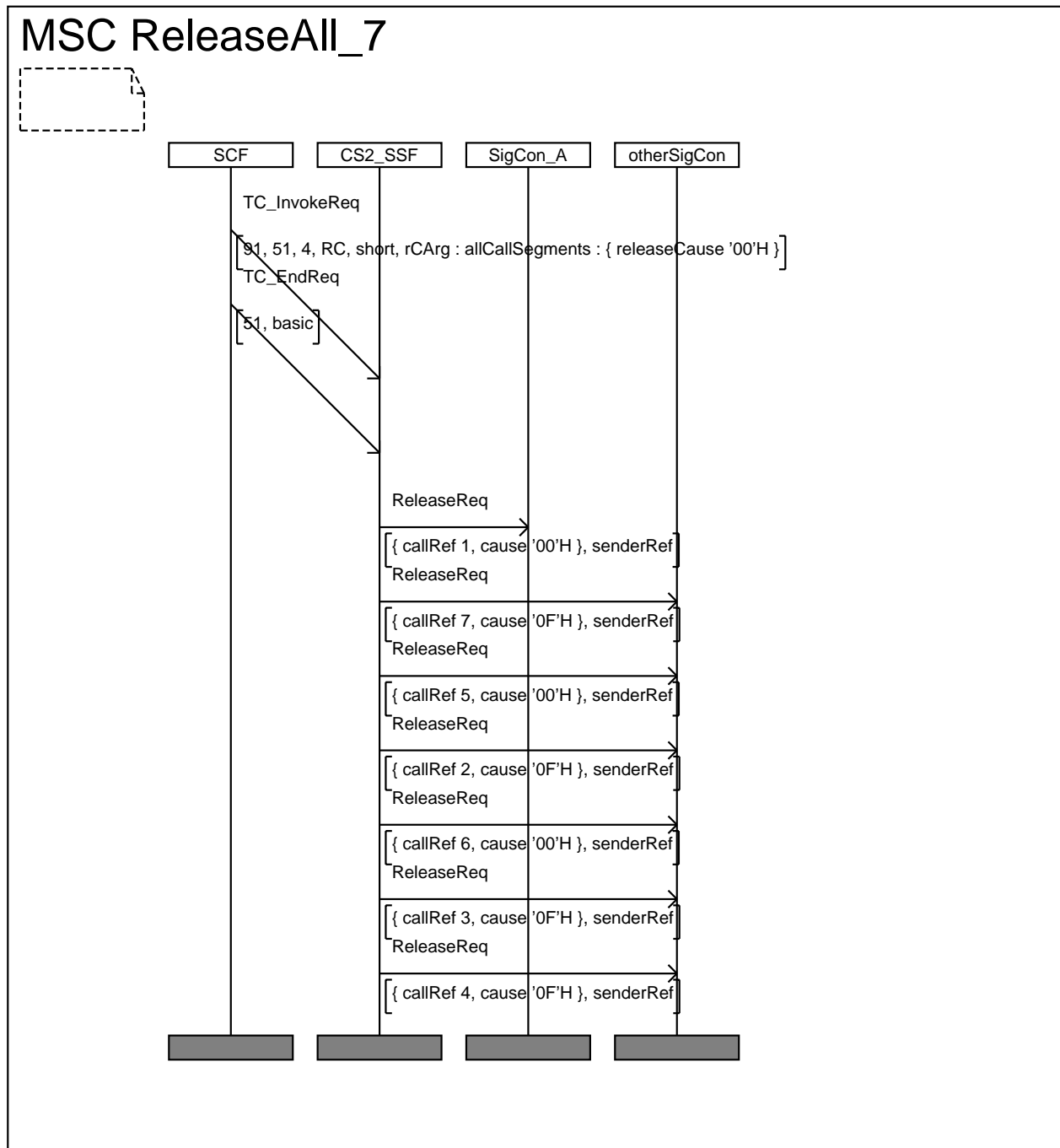


## 7.1.7.6 Postamble ReleaseAll\_6

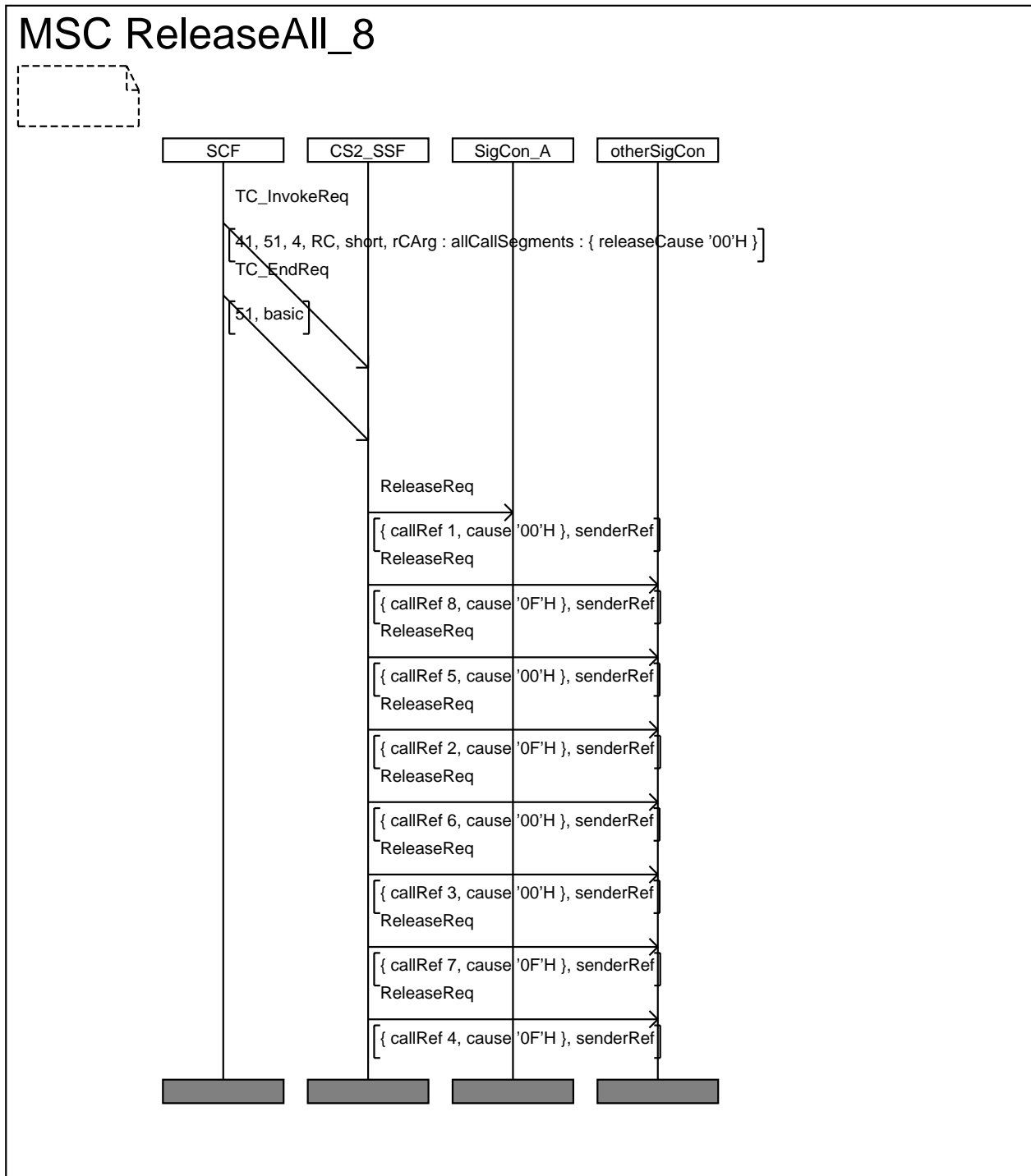




## 7.1.7.7 Postamble ReleaseAll\_7



## 7.1.7.8 Postamble ReleaseAll\_8



## 7.2 CPH procedures

## 7.2.1 List of procedures for CPH

CW	ContinueWithArgument
DL	DisconnectLeg
MC	MergeCallSegment
ML	MoveLeg
RC	ReleaseCall (CPH complement to CS1)
RO	Reconnect

RR RequestReportBCSMEEvent (CPH complement to CS1)  
 SL SplitLeg

## 7.2.2 Definitions of the CPH procedures

### Dealing with CallSegments

See also `splitLeg`.

#### 7.2.2.1 mergeCallSegments procedure

Invoke: `mergeCallSegments`  
 Return Result: `True`  
 Return Error: `mergeCallSegments`

#### 7.2.2.2 releaseCall procedure

Invoke: `releaseCall`  
 Return Result: `None`  
 Return Error: `releaseCall`

#### 7.2.2.3 continueWithArgument procedure

Invoke: `continueWithArgument`  
 Return Result: `None`  
 Return Error: `continueWithArgument`

### Dealing with Legs of CallSegments

See also `Connect` and `Continue` (from CS1).

#### 7.2.2.4 disconnectLeg procedure

Invoke: `disconnectLeg`  
 Return Result: `True`  
 Return Error: `disconnectLeg`

#### 7.2.2.5 moveLeg procedure

Invoke: `moveLeg`  
 Return Result: `True`  
 Return Error: `moveLeg`

#### 7.2.2.6 splitLeg procedure

Invoke: `splitLeg`  
 Return Result: `True`  
 Return Error: `splitLeg`

#### 7.2.2.7 continueWithArgument procedure

Invoke: `continueWithArgument`  
 Return Result: `None`  
 Return Error: `continueWithArgument`

## 7.3 Structure of the test suite (TSS) for CPH

Table 1 shows the structure of the test suites for the test of CPH procedures in the SSF and the number of Test Purposes produced.

In addition, the test suite contains a list of test cases numbered from 1 to 102, with a few holes in the numbering corresponding to TPs dropped during the development.

Table 1: Test suite structure for testing the CPH procedures

IUT	Interface	Protocol component	Procedure	Category and number
SSF	SSF-SCF	CPH subset	MC MergeCall Segment	CA
				BV 2
				BI 1
				BO
			RC ReleaseCall	CA
				BV 3
				BI
				BO
			DL DisconnectLeg	CA
				BV 3
				BI 1
				BO
			ML MoveLeg	CA
				BV 3
				BI 1
				BO
RR RequestReport BCSMEvent	CA			
	BV 2			
	BI			
	BO			
SL SplitLeg	CA			
	BV 2			
	BI 1			
	BO			
CW ContinueWith Argument	CA			
	BV 3			
	BI 2			
	BO			

Total: 24

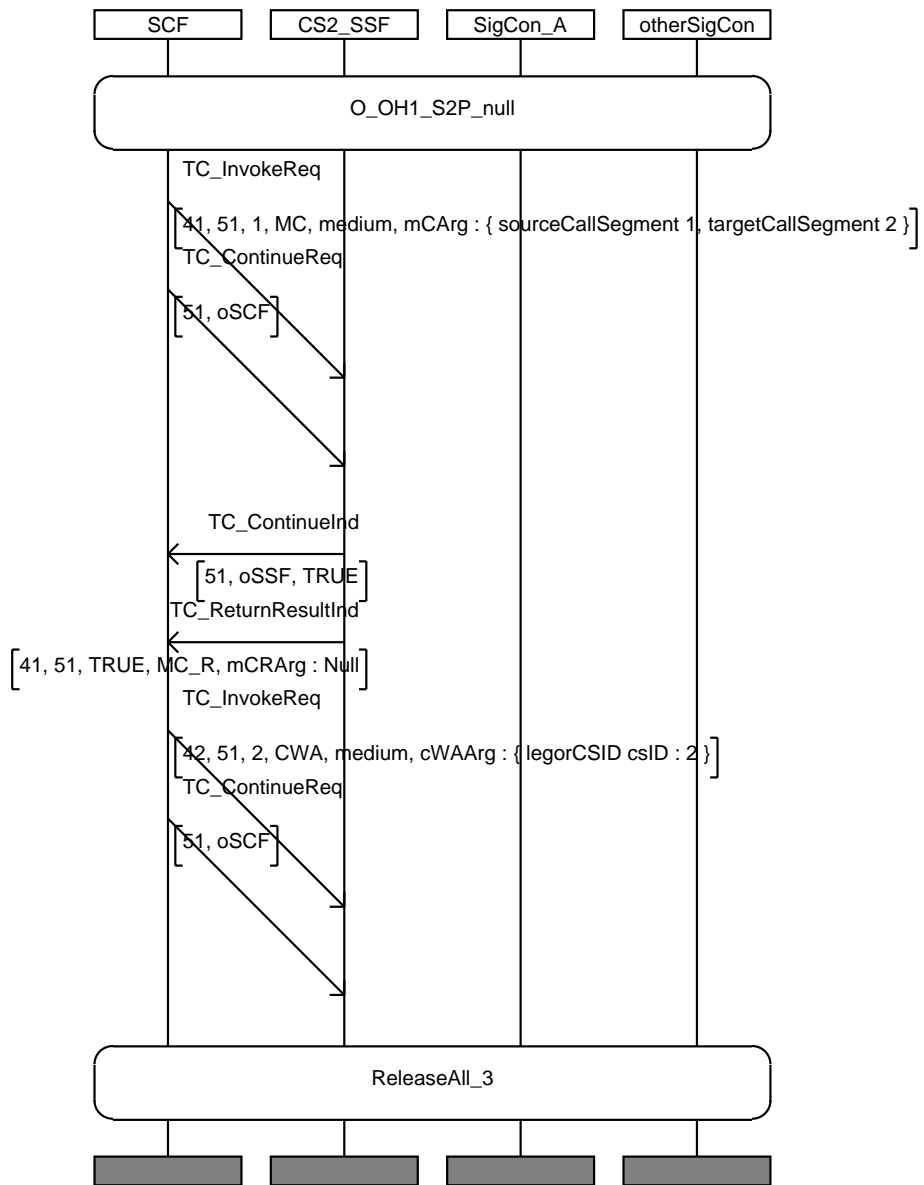
## 7.4 Test Purpose (TP) descriptions for the test of CPH procedures

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

## 7.4.1 MergeCallSegment procedure (MC)

IN2_A_CPH_MC_BV_01	
<b>Purpose:</b>	test MergeCallSegment functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>MergeCS_invoke</b> to SSF with the following parameters: - sourceCS=1 - targetCS=2  followed by <b>ContinueWithArgument</b> with parameter csID = 2
<b>Pass criteria</b>	Check that the SSF sends a MergeCallSegmentsReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_MC\_BV\_01

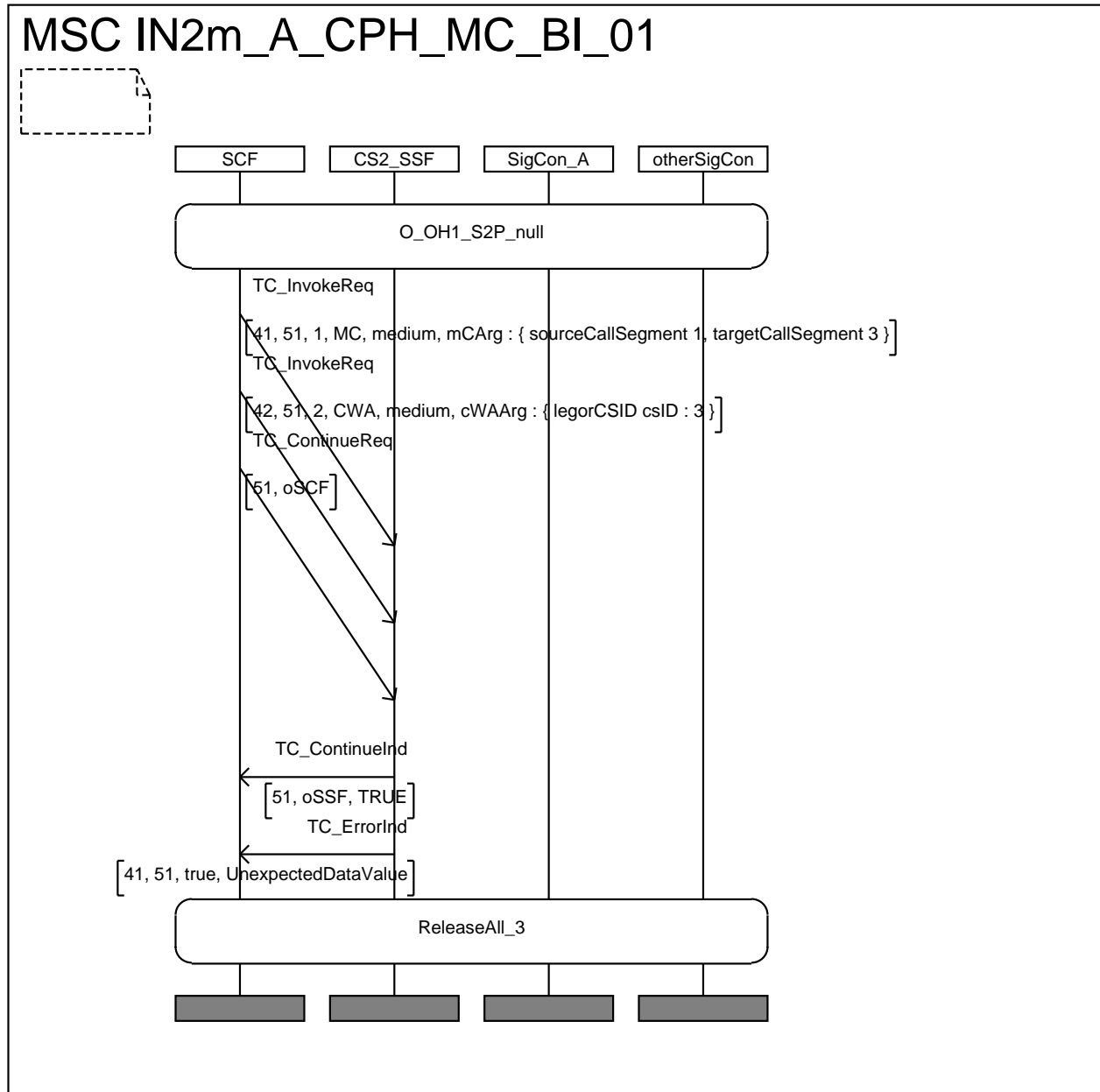


<b>IN2_A_CPH_MC_BV_02</b>	
<b>Purpose:</b>	test that armed DP are kept after MergeCS
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	<p>Arm of the DP, then establishment of configuration:  SCF sends to SSF a <b>SplitLeg_invoke</b> containing at least the parameters:</p> <ul style="list-style-type: none"> <li>- legToBeSplit=1</li> <li>- newCallSegment=2</li> </ul> <p>followed by <b>ContinueWithArgument</b> on csID=1  followed by <b>ReqRepBCSMEvent_invoke</b> containing at least the parameters:</p> <ul style="list-style-type: none"> <li>- eventTypeBCSM= oDisconnect</li> <li>- monitoringMode= interrupted</li> <li>- legId=3</li> </ul> <p>and</p> <ul style="list-style-type: none"> <li>- eventTypeBCSM= oAnswer</li> <li>- monitoringMode= interrupted</li> <li>- legId=3</li> </ul> <p>followed by <b>Connect_invoke</b> containing at least the parameter:</p> <ul style="list-style-type: none"> <li>- legToBeCreated=3</li> <li>- callSegmentID=2</li> </ul> <p>SSF establishes a link to SigCon C (send SetupReq, answered with SetupConf, then SetUpResp to sigconA)</p> <p>then SCF sends <b>MergeCS_invoke</b> with parameters:</p> <ul style="list-style-type: none"> <li>- sourceCS=1</li> <li>- targetCS=2</li> </ul> <p>followed by <b>ContinueWithArgument</b> on csID=2</p>
<b>Pass criteria</b>	<p>Check that SSF sends SplitLegReturnResult, and a MergeCallSegmentsReturnResult</p> <p>Check that the armed DP follows the leg which was merged, i.e. when SigCon C sends ReleaseInd to SSF, SSF sends to SCF an <b>EventReportBCSM</b></p>
<b>Postamble:</b>	ReleaseAll_3





<b>IN2_A_CPH_MC_BI_01</b>	
<b>Purpose:</b>	test that MergeCS is rejected when wrong parameter is used
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>MergeCS_invoke</b> with parameters: <ul style="list-style-type: none"><li>- sourceCS=1</li><li>- targetCS=3</li></ul> followed by <b>ContinueWithArgument</b> on csID=3
<b>Pass criteria</b>	as CS=3 does not exist, Check that SSF sends to SCF a MergeCS_err containing the parameter: <ul style="list-style-type: none"><li>- unexpected data value</li></ul>
<b>Postamble:</b>	ReleaseAll_3



## 7.4.2 MoveCallSegment procedure (MCS)

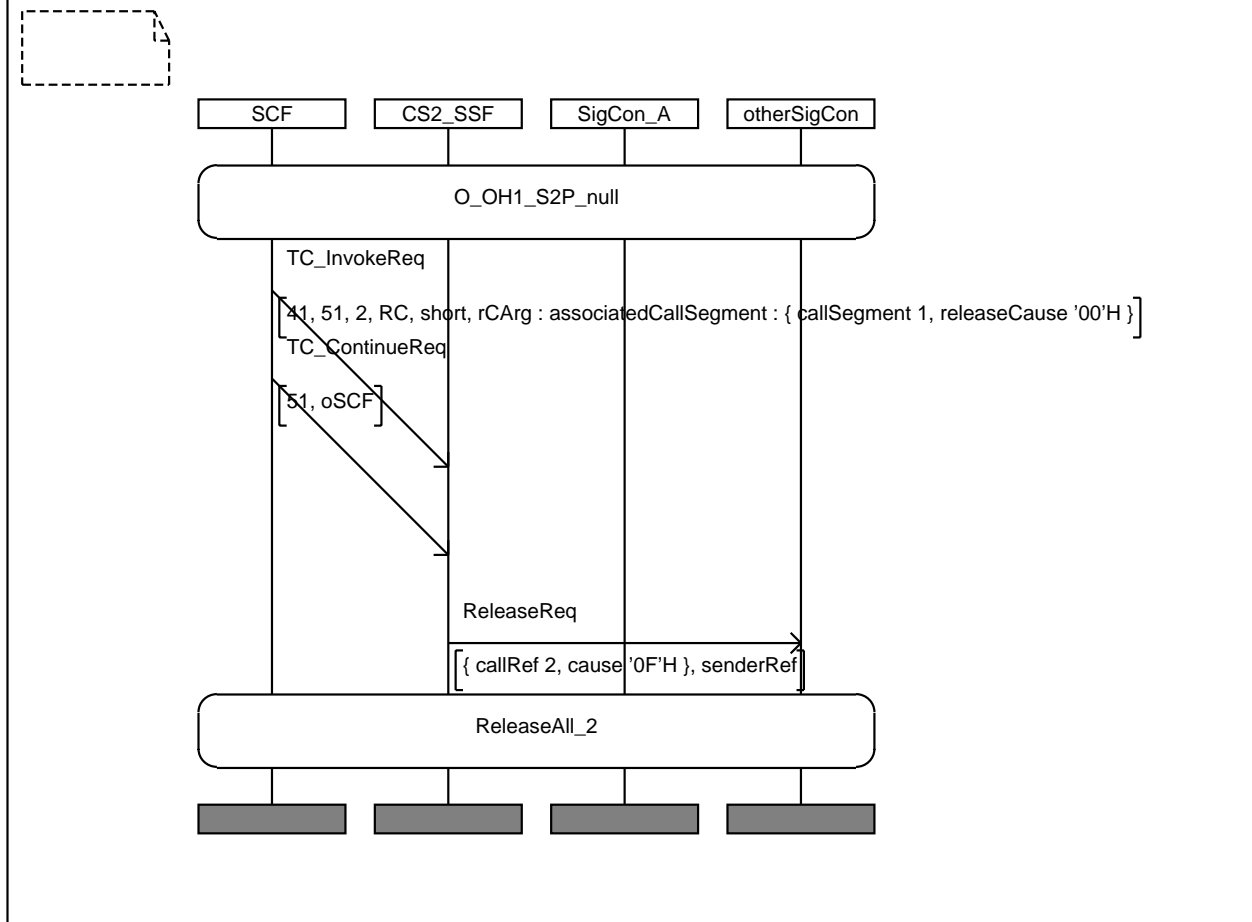
No Test Purpose defined for this procedure, CSAId behaviour being dependant upon implementation.

IN2_A_CPH_MCS_BV_01
IN2_A_CPH_MCS_BI_01

## 7.4.3 ReleaseCall (CS2 complement) procedure (RC)

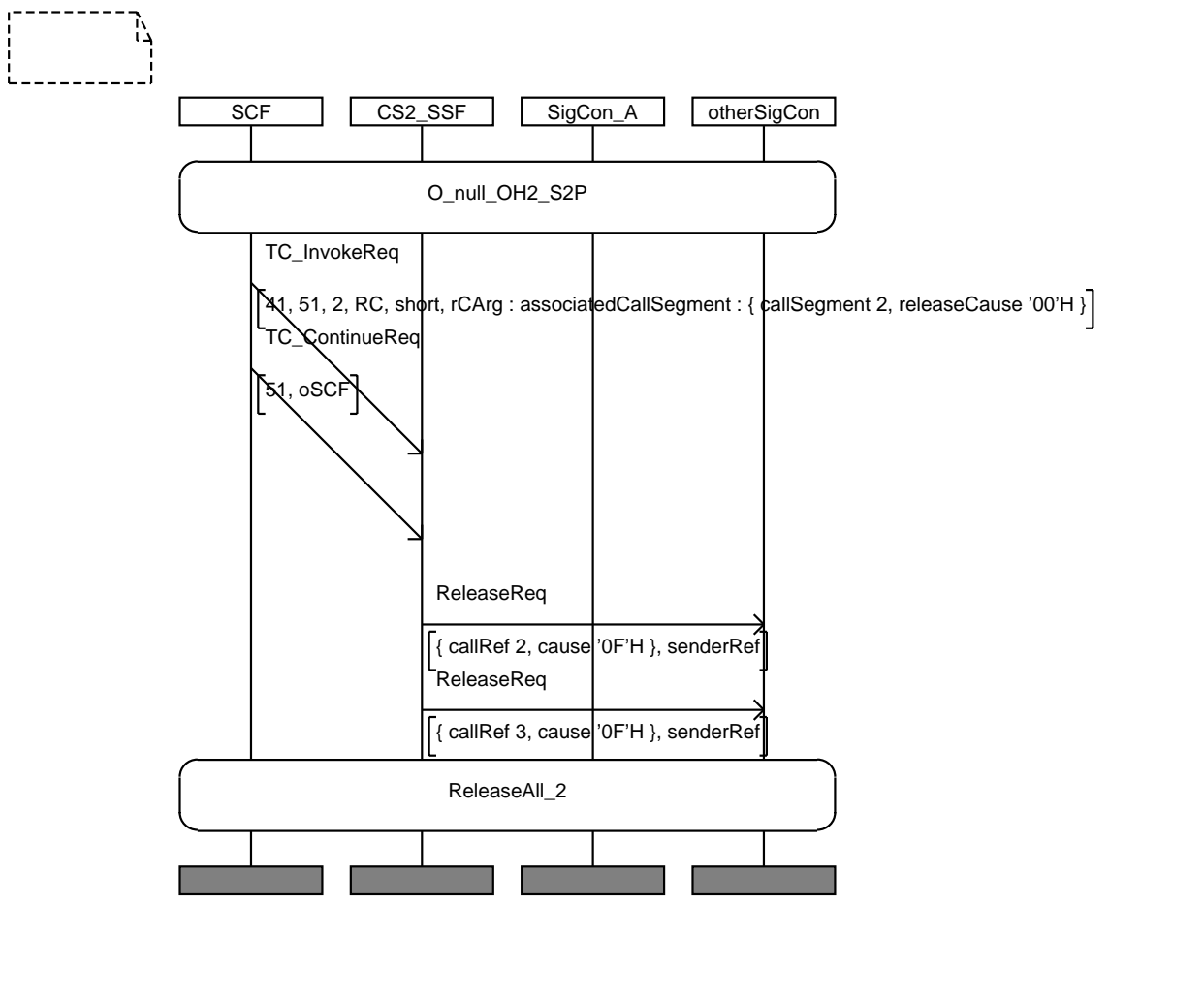
IN2_A_CPH_RC_BV_01	
<b>Purpose:</b>	test release of a given Csid, one leg only
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>ReleaseCall_invoke</b> to SSF with the following parameters: - cause=normal unspecified - Csid=1
<b>Pass criteria</b>	Check that SSF sends to SigConB a ReleaseReq as Csid=1 contains leg=2
<b>Postamble:</b>	ReleaseAll_2

## MSC IN2\_A\_CPH\_RC\_BV\_01



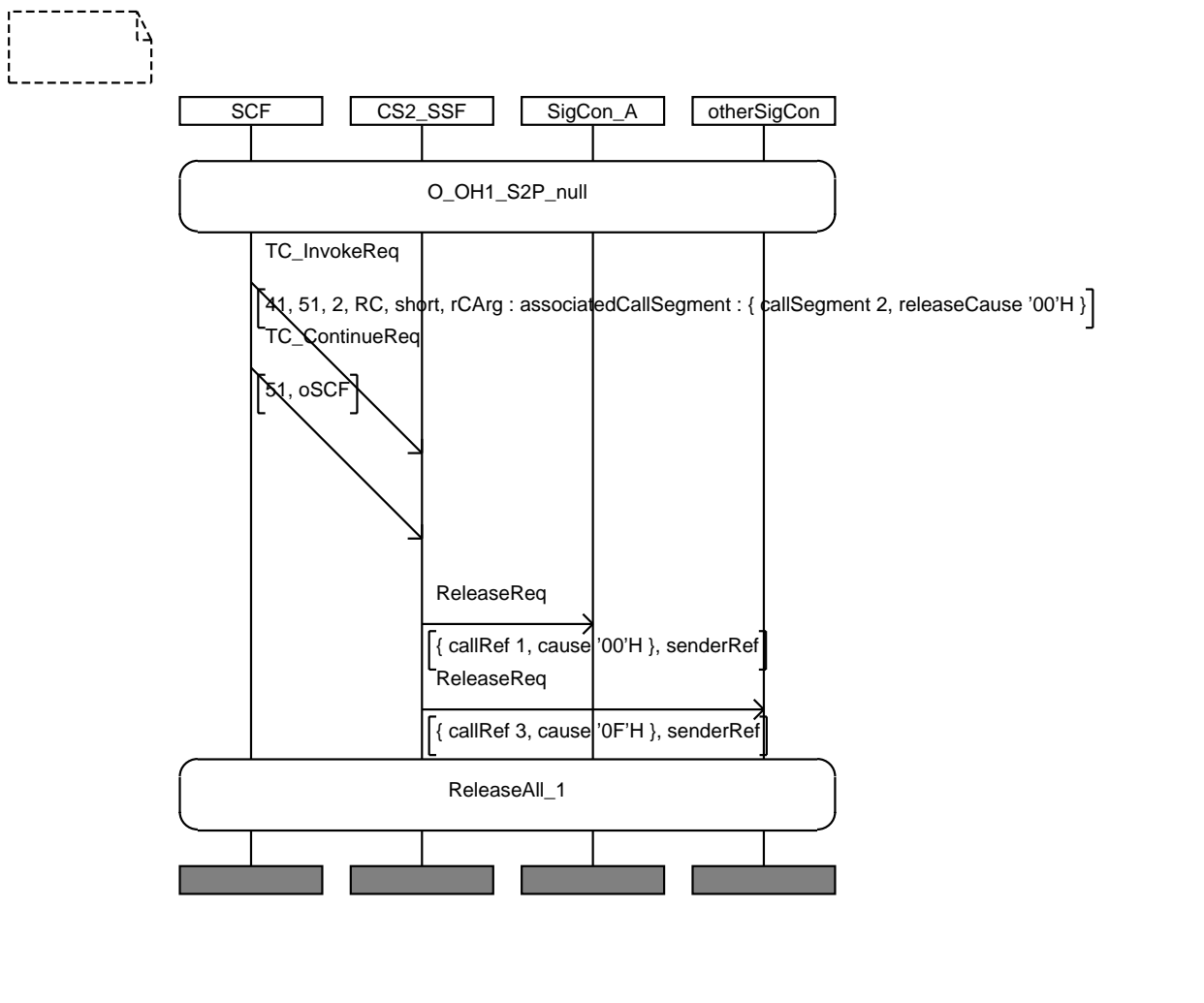
IN2_A_CPH_RC_BV_02	
<b>Purpose:</b>	test release of a given Csid, two legs
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P
<b>Test description</b>	SCF sends <b>ReleaseCall_invoke</b> to SSF with the following parameters: - cause=normal unspecified - CSid=2
<b>Pass criteria</b>	Check that SSF sends to SigConC and to SigConD a ReleaseReq as Csid=2 contains leg=3 to SigCon C and leg4 to SigconD
<b>Postamble:</b>	ReleaseAll_2

## MSC IN2\_A\_CPH\_RC\_BV\_02



<b>IN2_A_CPH_RC_BV_03</b>	
<b>Purpose:</b>	Test of <b>ReleaseCall</b> procedure with two parties
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends to SSF <b>ReleaseCall</b> invoke, with: <ul style="list-style-type: none"><li>- associatedCallSegment     callSegment(CsId=2)     releaseCause(cause=00)</li></ul>
<b>Pass criteria</b>	Check that SSF releases the call (ReleaseReq received by SigConA and SigConC)
<b>Postamble:</b>	ReleaseAll_1

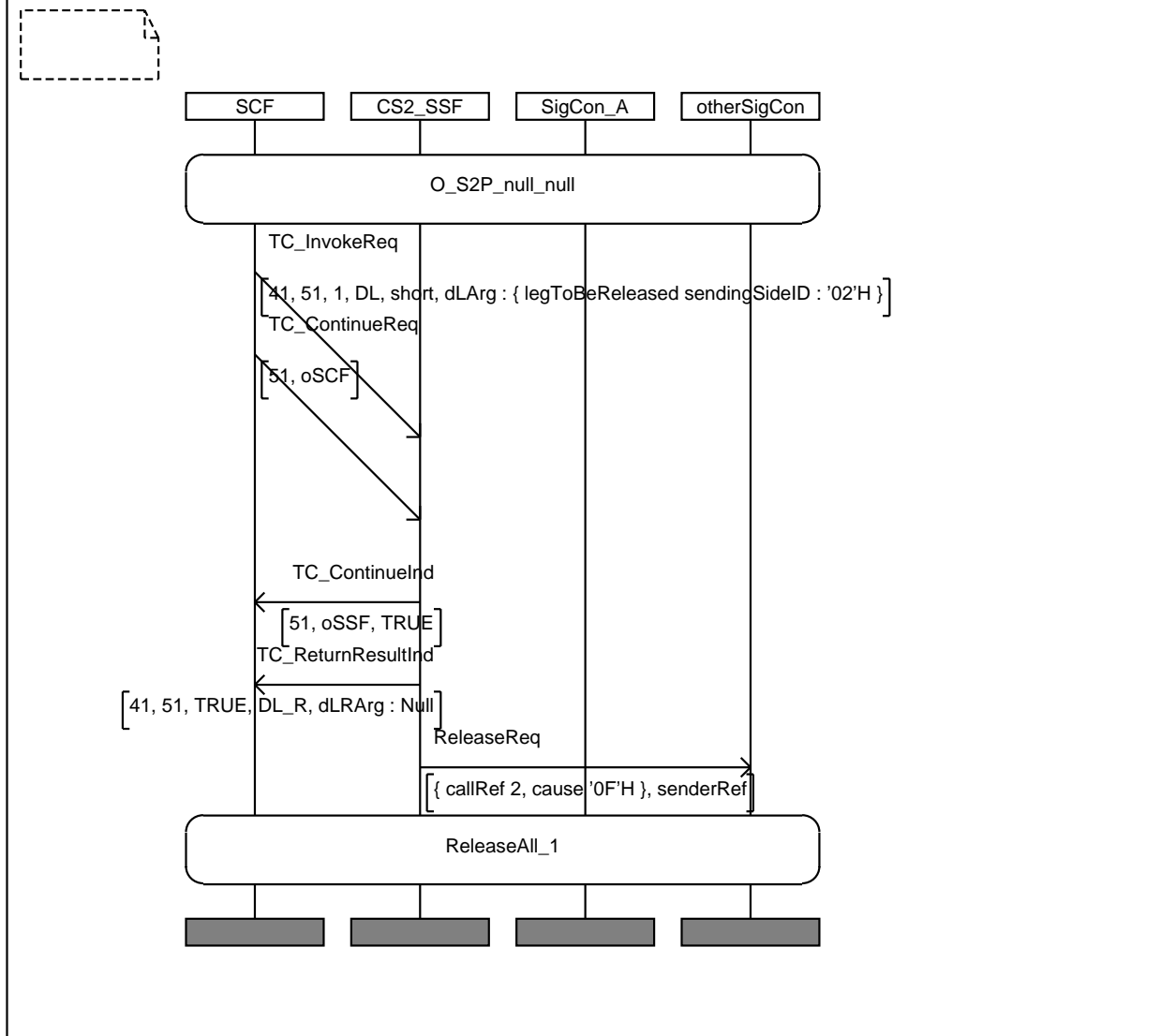
## MSC IN2\_A\_CPH\_RC\_BV\_03



### 7.4.4 DisconnectLeg Procedure (DL)

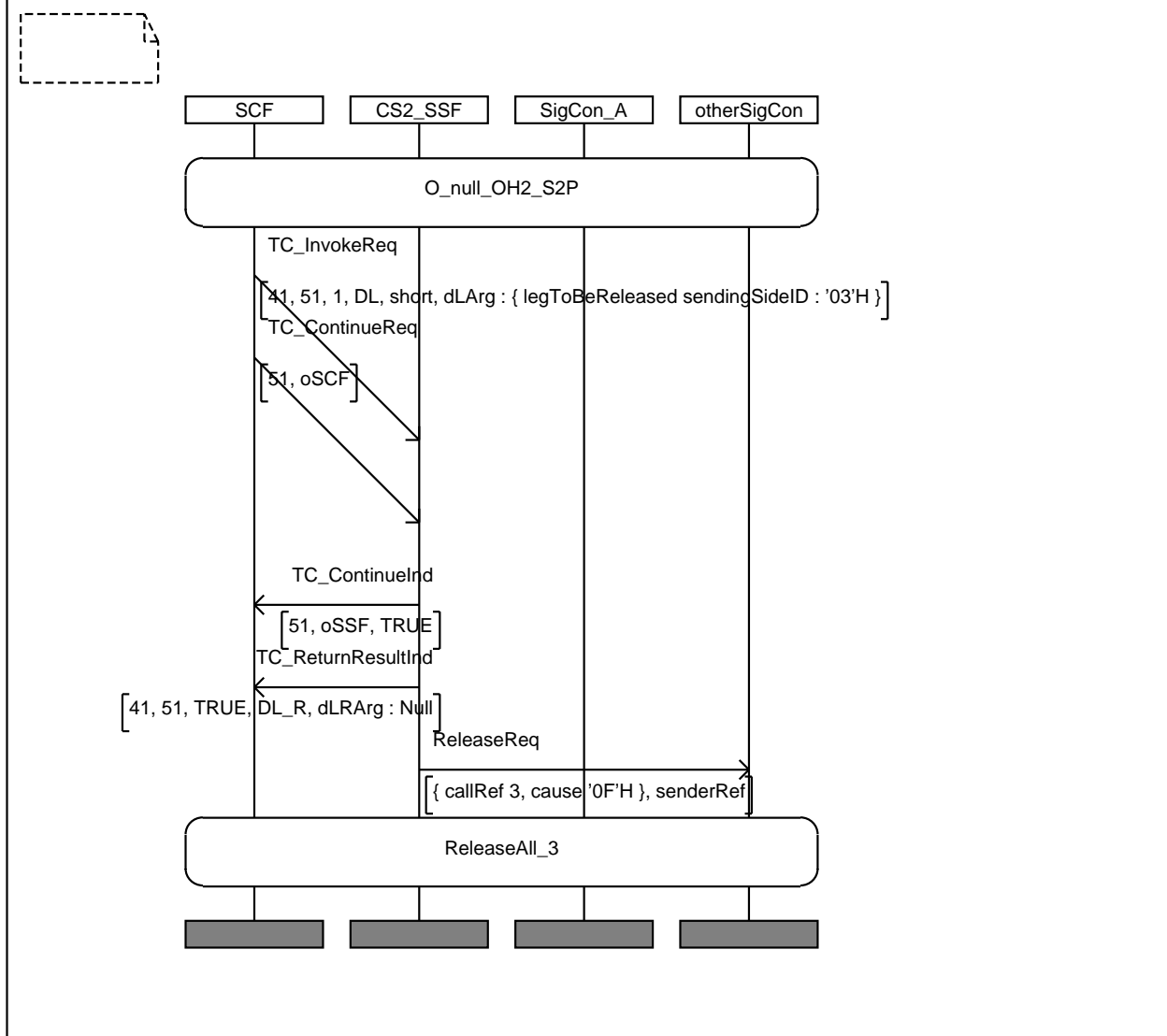
IN2_A_CPH_DL_BV_01	
<b>Purpose:</b>	test Disconnect leg in Csid1
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF sends <b>DisconnectLeg_invoke</b> to SSF with the following parameters: - legToBeReleased=2
<b>Pass criteria</b>	Check SSF sends a DisconnectLegReturnResult  Check that SSF sends to SigConB a ReleaseReq as it is addressed to leg=2 linked to SigCon B
<b>Postamble:</b>	ReleaseAll_1

# MSC IN2\_A\_CPH\_DL\_BV\_01



IN2_A_CPH_DL_BV_02	
<b>Purpose:</b>	test Disconnect one leg of a multileg Csid
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P
<b>Test description</b>	SCF sends <b>DisconnectLeg_invoke</b> to SSF with the following parameters: - legToBeReleased=3
<b>Pass criteria</b>	Check SSF sends a DisconnectLegReturnResult  Check that SSF sends to SigConC a ReleaseReq as it is addressed to leg=3 in Csid2 and is linked to SigCon C
<b>Postamble:</b>	ReleaseAll_3

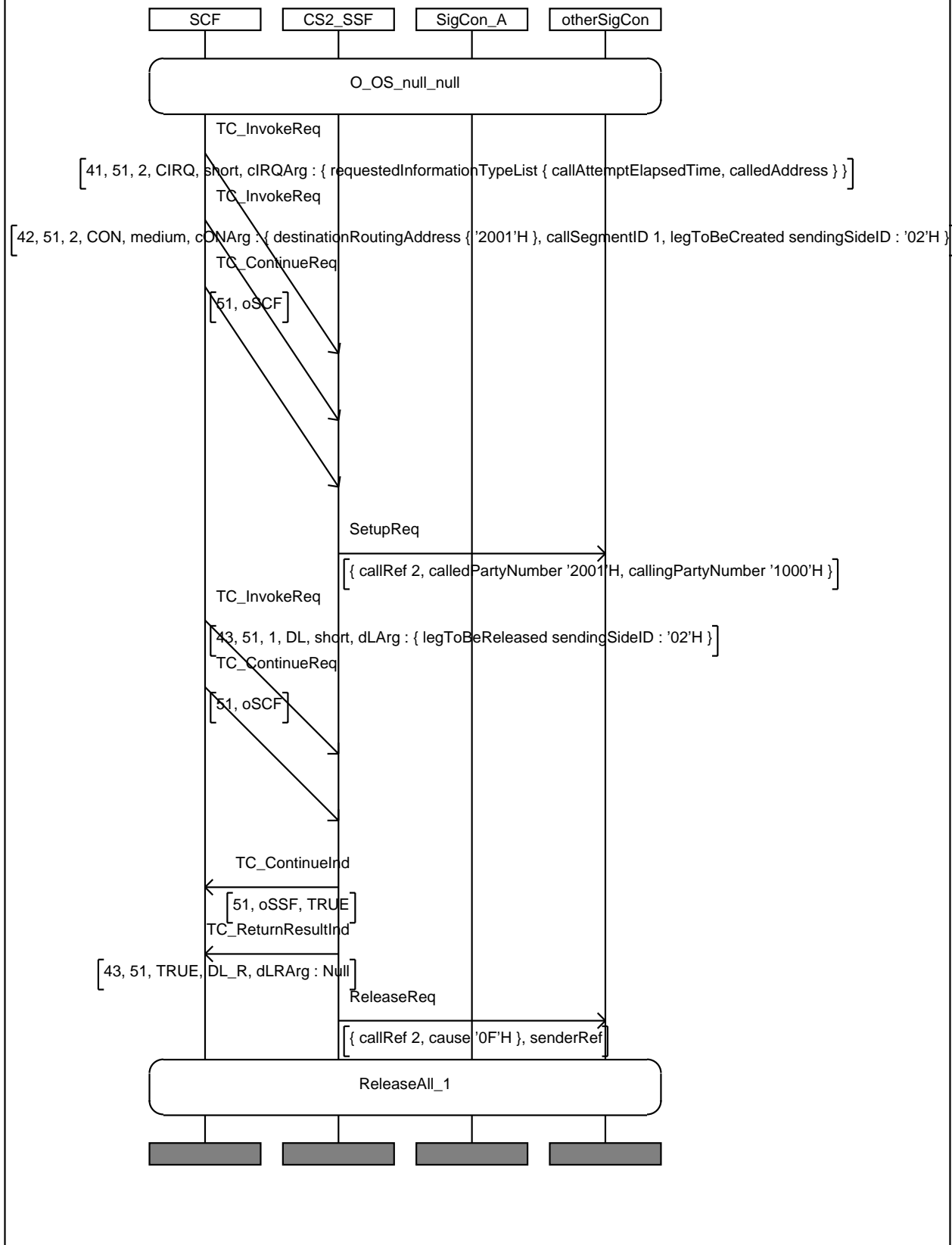
# MSC IN2\_A\_CPH\_DL\_BV\_02



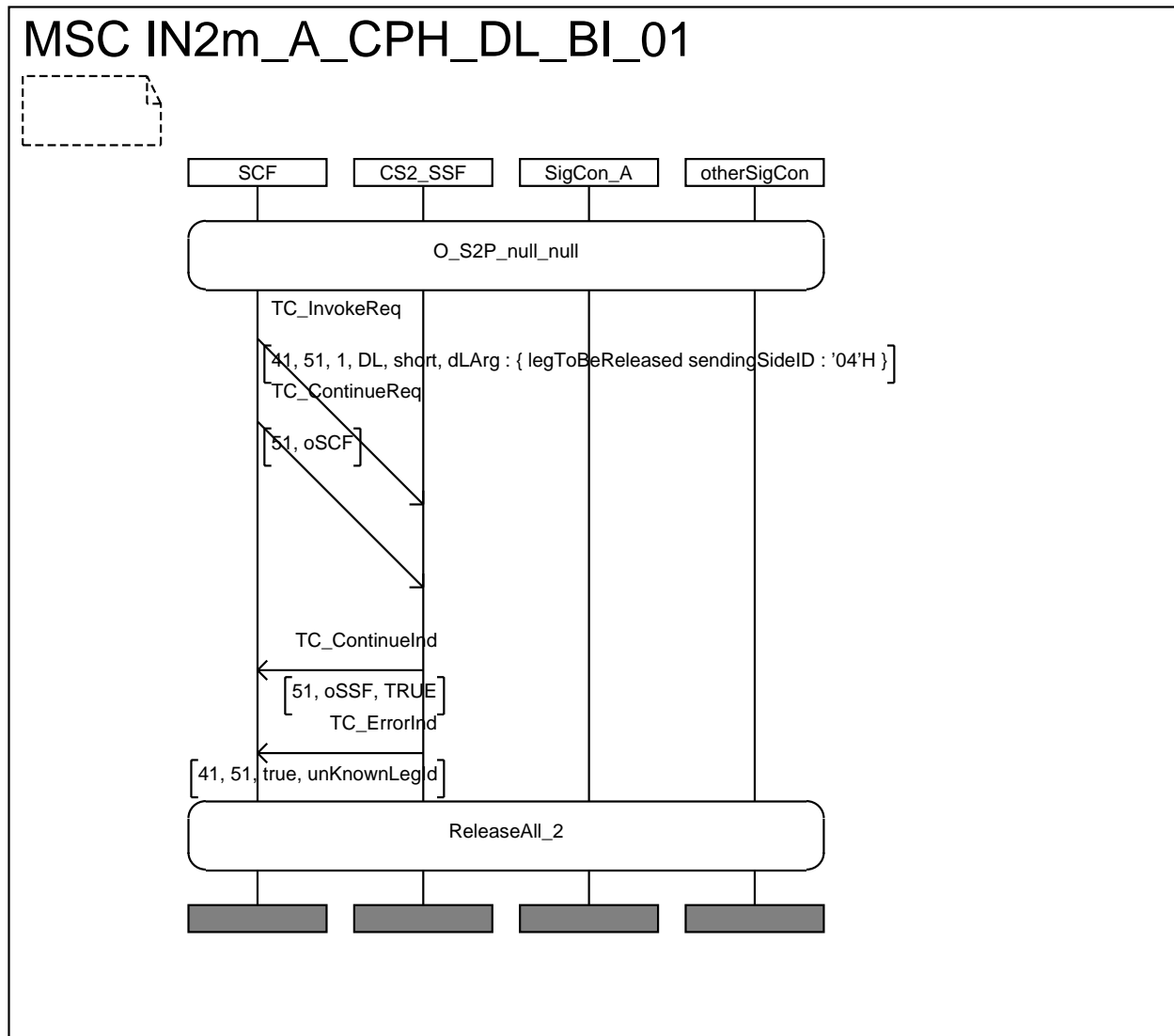
IN2_A_CPH_DL_BV_03	
<b>Purpose:</b>	test that armed DP are disarmed after DisconnectLeg and pending reports are sent
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF sends <b>CallInformationRequest_invoke</b> to SSF with the following parameter: - requestedInformationType=any valid type followed by <b>DisconnectLeg_invoke</b> with the following parameter: - legToBeReleased=2
<b>Pass criteria</b>	Check SSF sends a DisconnectLegReturnResult Check that SSF sends to SCF a CallInformationReport and sends to SigConB a ReleaseReq
<b>Postamble:</b>	ReleaseAll_1



# MSC IN2\_A\_CPH\_DL\_BV\_03



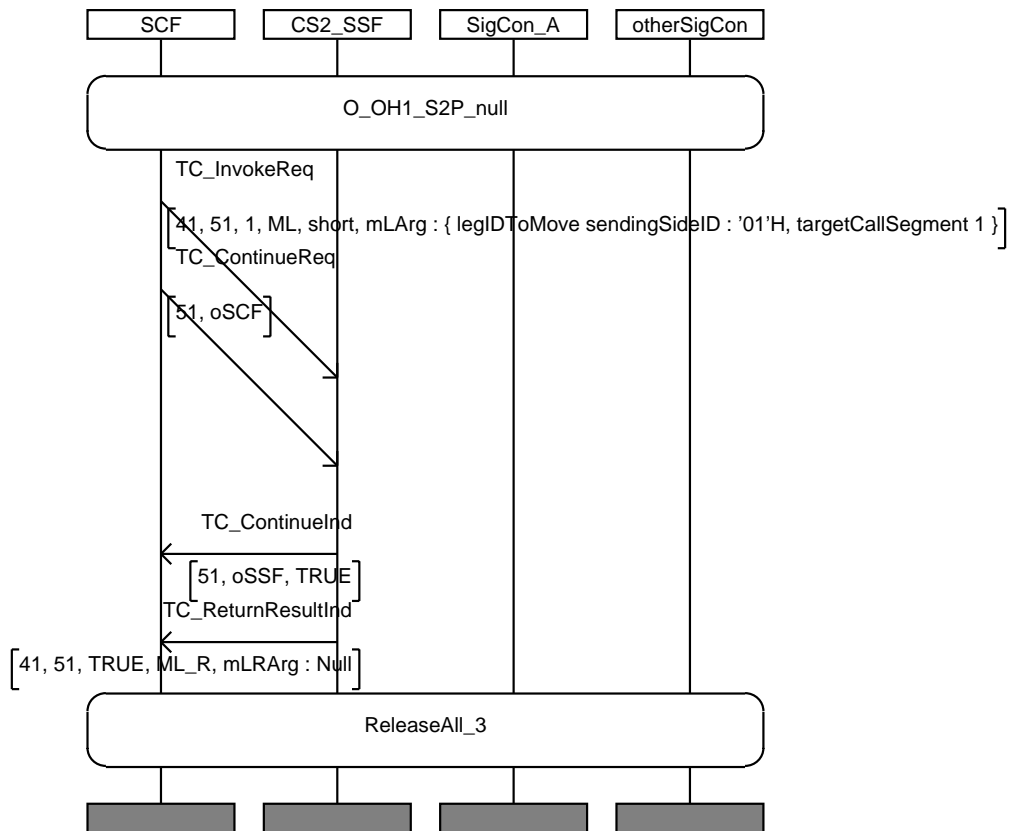
<b>IN2_A_CPH_DL_BI_01</b>	
<b>Purpose:</b>	test that DisconnectLeg is rejected when wrong parameter is used
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF sends <b>DisconnectLeg_invoke</b> with parameters: - legId=4
<b>Pass criteria</b>	as legId=4 does not exist, Check that SSF sends to SCF a DisconnectLeg_err containing the parameter: - unknownLegId
<b>Postamble:</b>	ReleaseAll_2



#### 7.4.5 MoveLeg Procedure (ML)

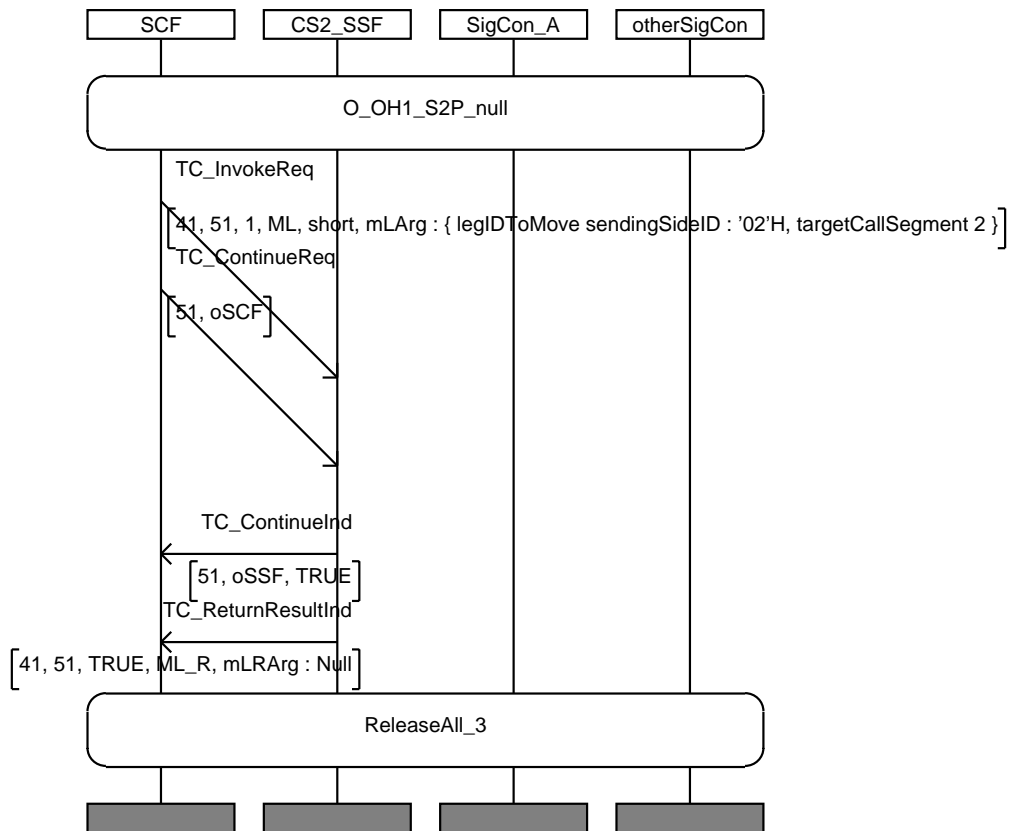
IN2_A_CPH_ML_BV_01	
<b>Purpose:</b>	test Moveleg functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>MoveLeg_invoke</b> to SSF with the following parameters: - legIdToMove=1 - targetCallSegment=1
<b>Pass criteria</b>	Check that the SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_ML\_BV\_01

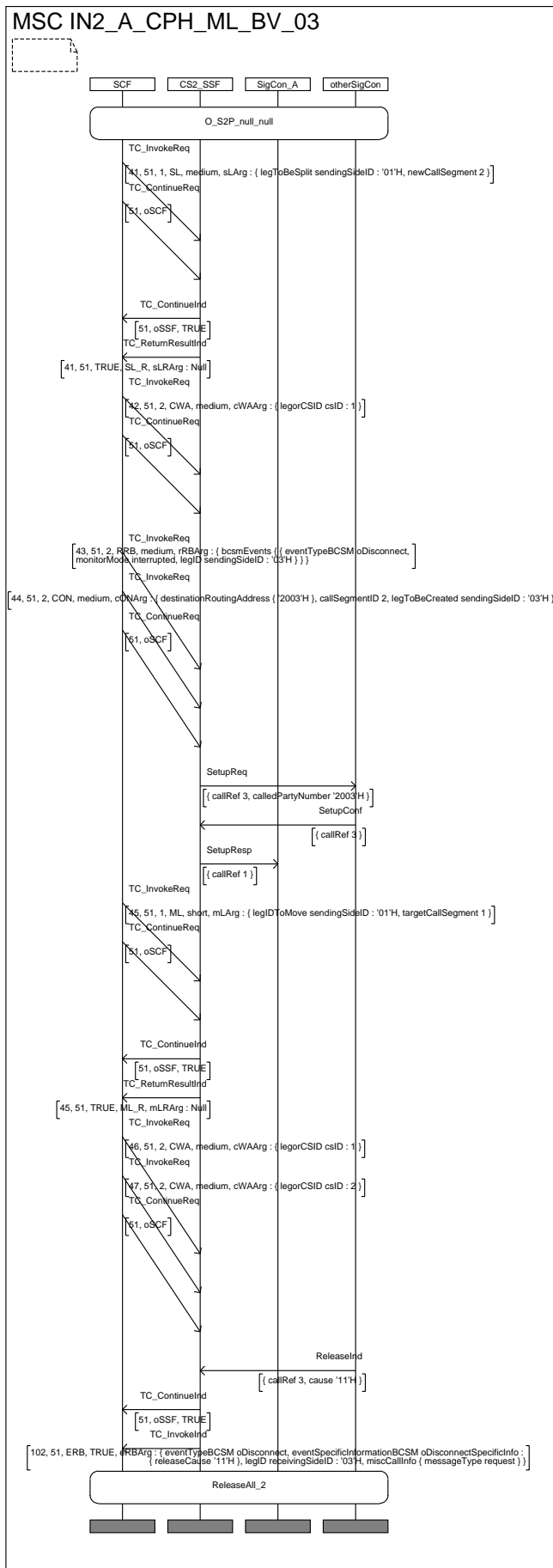


IN2_A_CPH_ML_BV_02	
<b>Purpose:</b>	test Moveleg functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>MoveLeg_invoke</b> to SSF with the following parameters: - legIDToMove=2 - targetCallSegment=2
<b>Pass criteria</b>	Check that the SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_ML\_BV\_02

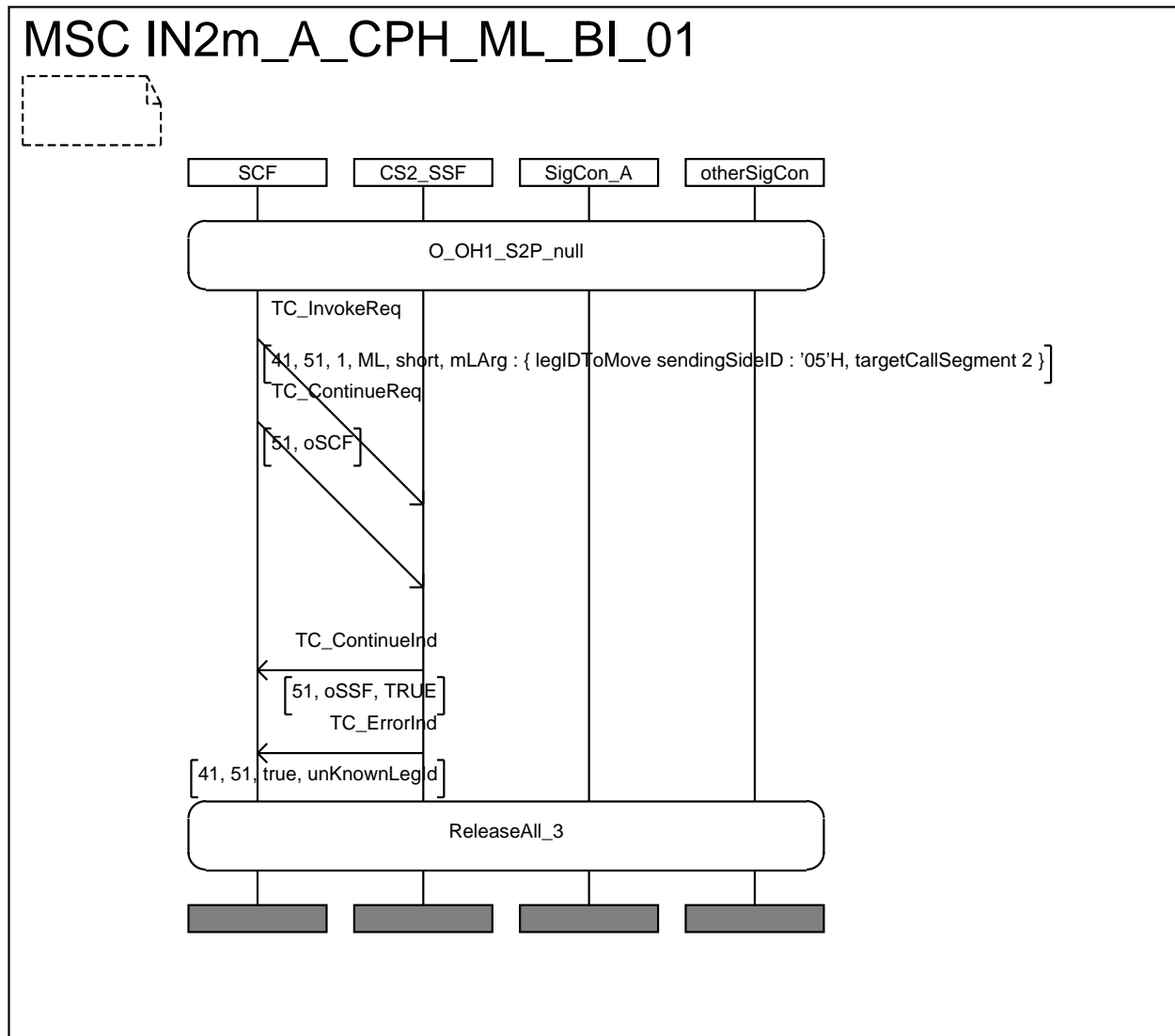


<b>IN2_A_CPH_ML_BV_03</b>	
<b>Purpose:</b>	test Event reporting rules apply after MoveLeg
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	<p>SigConB sends SetupConf,  SCF sends <b>SplitLeg</b></p> <ul style="list-style-type: none"> <li>- legToBeSplit=1</li> <li>- newCallSegment=2</li> </ul> <p>followed by <b>ContinueWithArgument</b> for csID=1  SCF sends <b>RequestReportBCSM</b> (oDisconnect, legID=3),  SCF send <b>Connect</b> (callSegmentID=2,legToBeCreated=3).</p> <p>After SetupReq is received at SigConC, which is answered with SetUpConf,  SCF sends <b>MoveLeg_invoke</b> to SSFwith the following parameters:</p> <ul style="list-style-type: none"> <li>- legIDtoMove=1</li> <li>- targetCallSegment=1</li> </ul> <p>followed by <b>ContinueWithArgument</b> for csID=1 and 2.  SigConC send ReleaseInd with cause BptyDisc.</p>
<b>Pass criteria</b>	Check that the SSF sends MoveLegReturnResult, Check that after ReleaseInd from SigConC, the <b>EventReportBCSM</b> is received with indication oDisconnect.
<b>Postamble:</b>	ReleaseAll_2



<b>IN2_A_CPH_ML_BI_01</b>	
<b>Purpose:</b>	test that SSF sends an error after a MoveLeg with wrong parameters
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>MoveLeg_invoke</b> to SSF with the following parameters: <ul style="list-style-type: none"><li>- legIdToMove=5</li><li>- targetCallSegment=2</li></ul>
<b>Pass criteria</b>	as legId=5 does not exist, check that SSF sends to SCF a MoveLeg_err containing the parameter: <ul style="list-style-type: none"><li>- unknownLegId</li></ul>
<b>Postamble:</b>	ReleaseAll_3

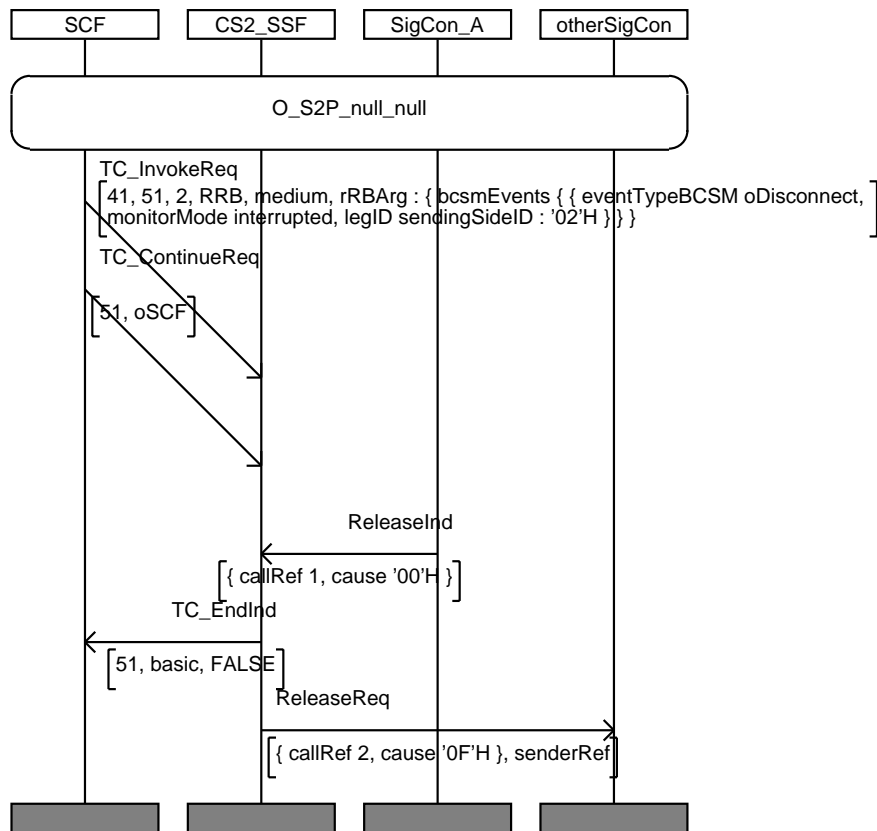




#### 7.4.6 RequestReportBCSMEvent Procedure (CS2 additions) (RR)

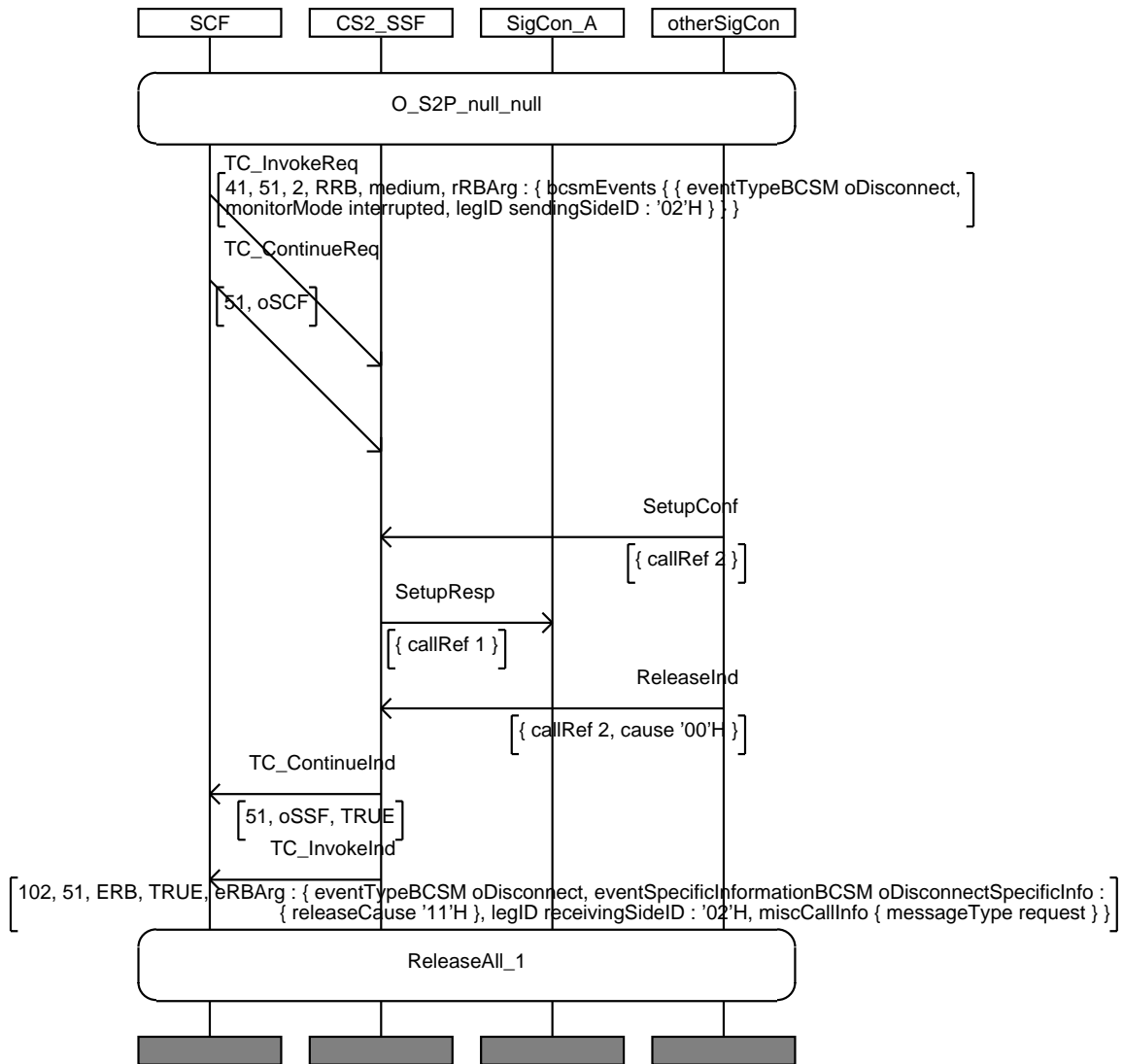
IN2_A_CPH_RR_BV_01	
<b>Purpose:</b>	Test of <b>RequestReportBCSMEvent</b> procedure and oDisconnect indication.
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF SCF sends to SSF <b>RequestReportBCSMEvent</b> invoke containing parameters <ul style="list-style-type: none"> <li>- eventTypeBCSM= oDisconnect</li> <li>- monitoringMode=interrupted</li> <li>- legid=2</li> </ul> Then SigCon A (calling party) clears the call before it is answered (ReleaseInd sent)
<b>Pass criteria</b>	<ul style="list-style-type: none"> <li>- Check that SSF <b>does</b> not send to SCF an <b>EventReportBCSM</b> because the detection point was armed for leg2</li> <li>- Check that SSF is ending the dialogue with SCF and releasing SigConB (ReleaseReq sent)</li> </ul>
<b>Postamble:</b>	none

# MSC IN2\_A\_CPH\_RR\_BV\_01



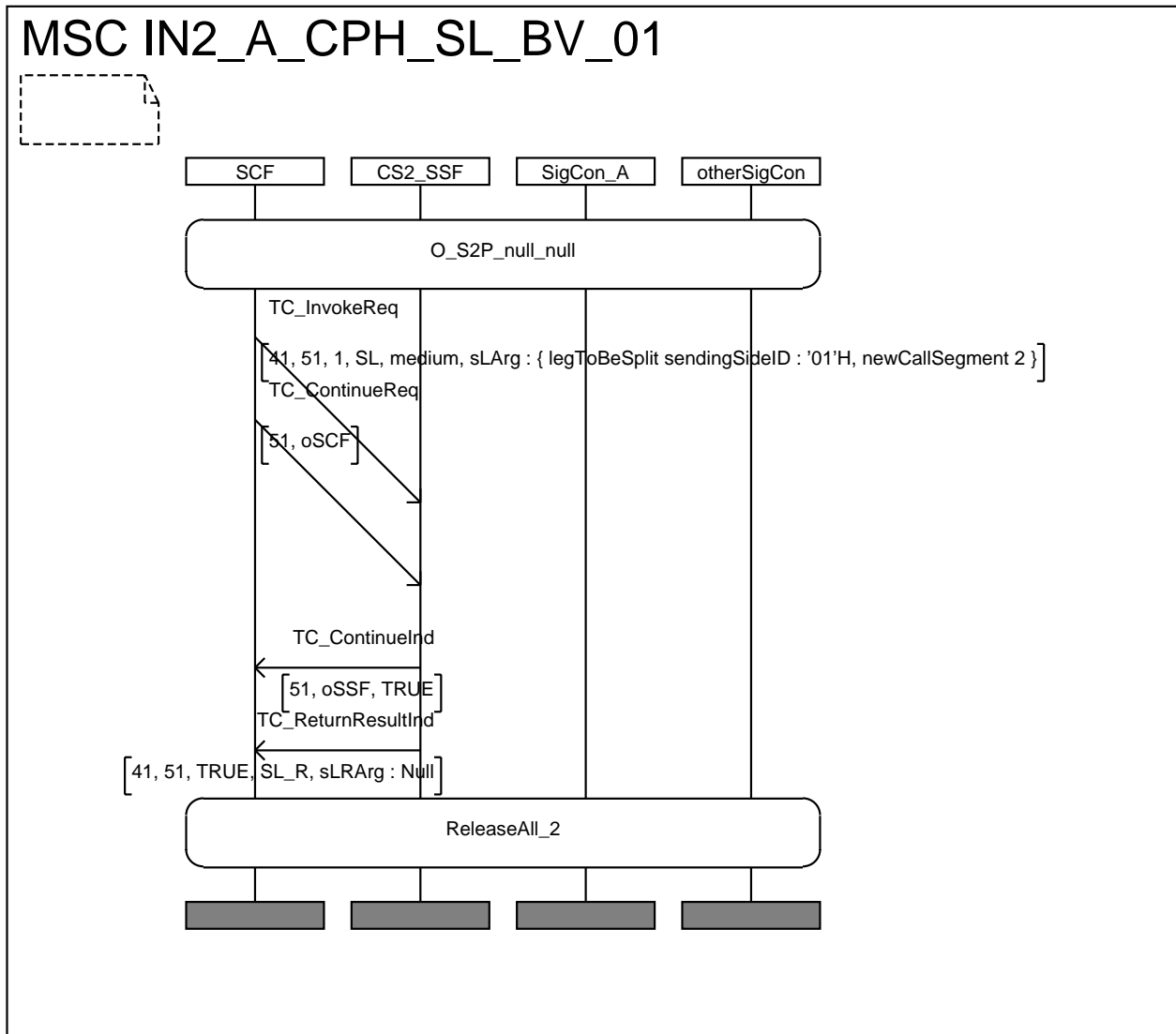
IN2_A_CPH_RR_BV_02	
<b>Purpose:</b>	Test of <b>RequestReportBCSMEvent</b> procedure and oDisconnect indication.
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	<p>SCF SCF sends to SSF <b>RequestReportBCSMEvent</b> invoke containing parameters</p> <ul style="list-style-type: none"> <li>- eventTypeBCSM= oDisconnect</li> <li>- monitoringMode=interrupted</li> <li>- legid=2</li> </ul> <p>Then SigCon B (called party) clears the call after it is answered (SetupConf then ReleaseInd sent)</p>
<b>Pass criteria</b>	Check that SSF sends to SCF an <b>EventReportBCSM</b> with the indication of <b>eventTypeBCSM= oDisconnect</b>
<b>Postamble:</b>	ReleaseAll_1

# MSC IN2\_A\_CPH\_RR\_BV\_02

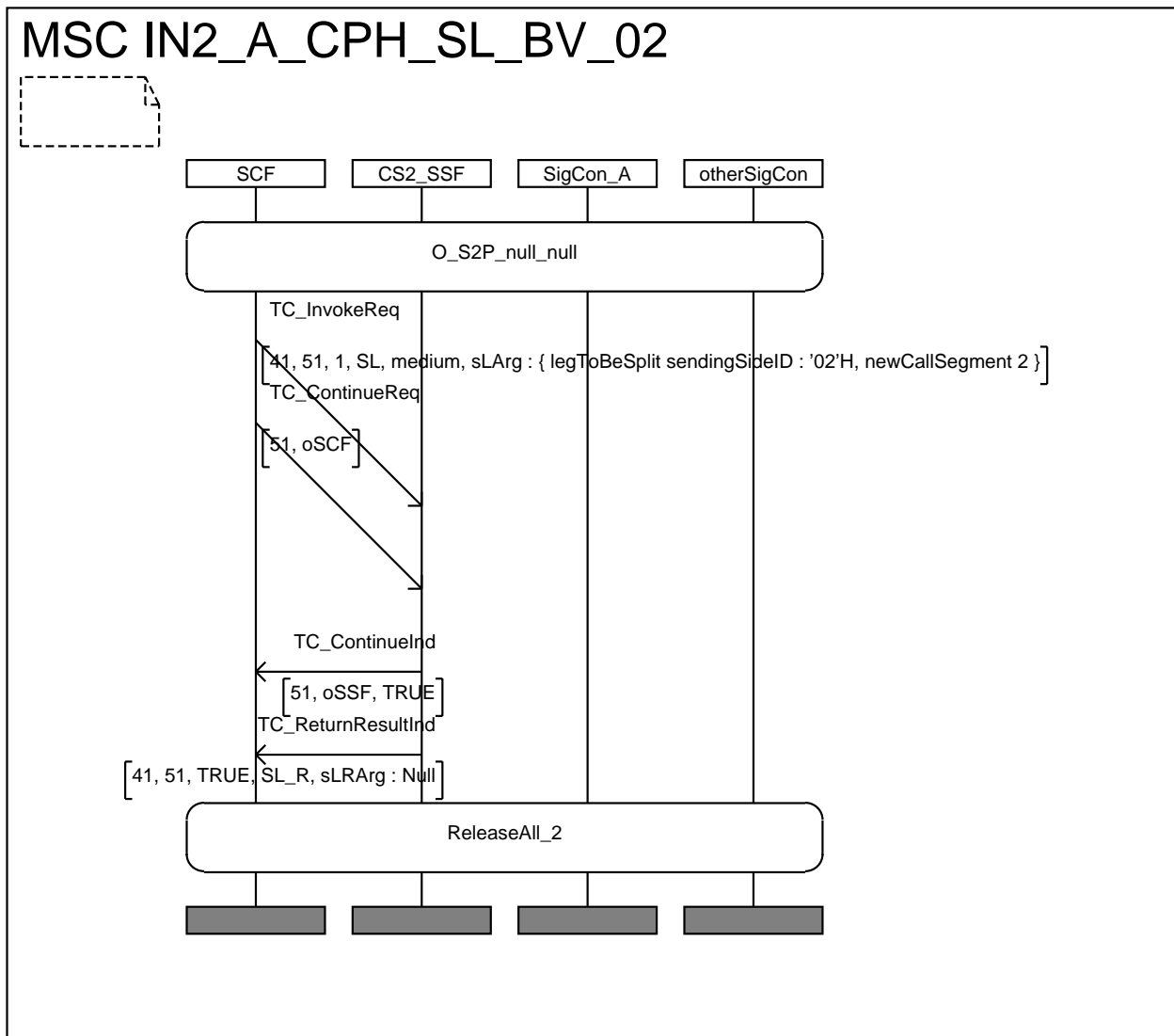


### 7.4.7 SplitLeg Procedure (SL)

IN2_A_CPH_SL_BV_01	
<b>Purpose:</b>	Test of <b>splitLeg</b> procedure functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF sends to SSF a <b>SplitLeg_invoke</b> containing at least the parameters: - legToBeSplit=1 - newCallSegment=2
<b>Pass criteria</b>	Check that the SSF sends SplitLegReturnResult
<b>Postamble:</b>	ReleaseAll_2

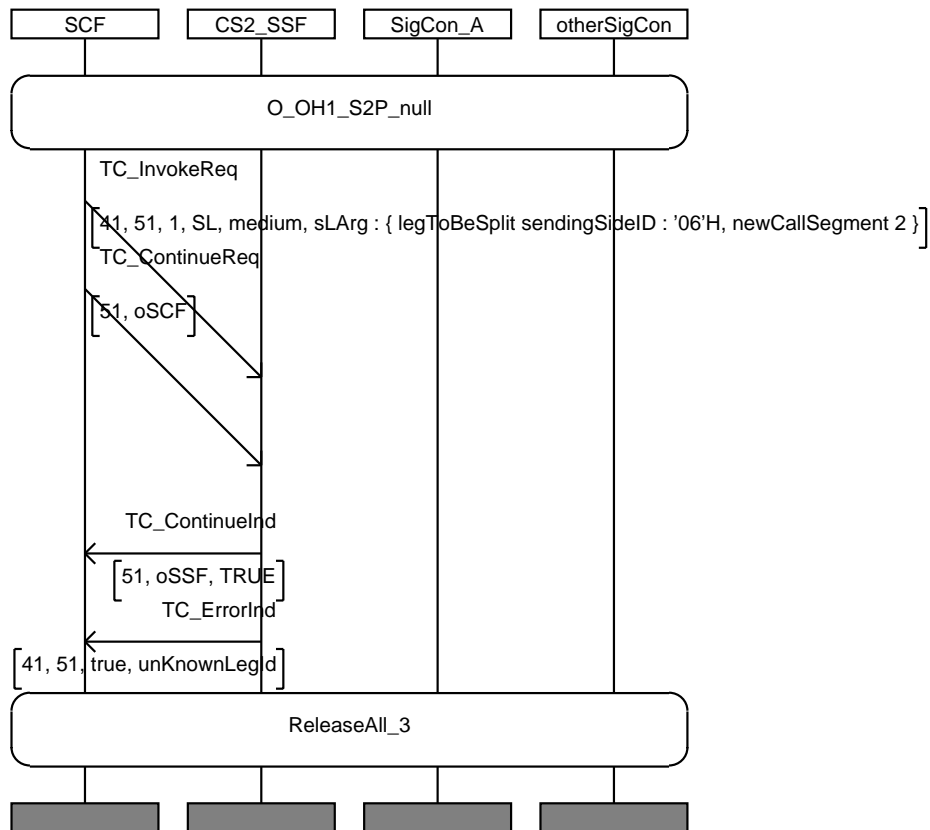


IN2_A_CPH_SL_BV_02	
<b>Purpose:</b>	Test of <b>splitLeg</b> procedure functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF sends to SSF a <b>SplitLeg_invoke</b> containing at least the parameters: - legToBeSplit=2 - newCallSegment=2
<b>Pass criteria</b>	Check that the SSF sends SplitLegReturnResult
<b>Postamble:</b>	ReleaseAll_2



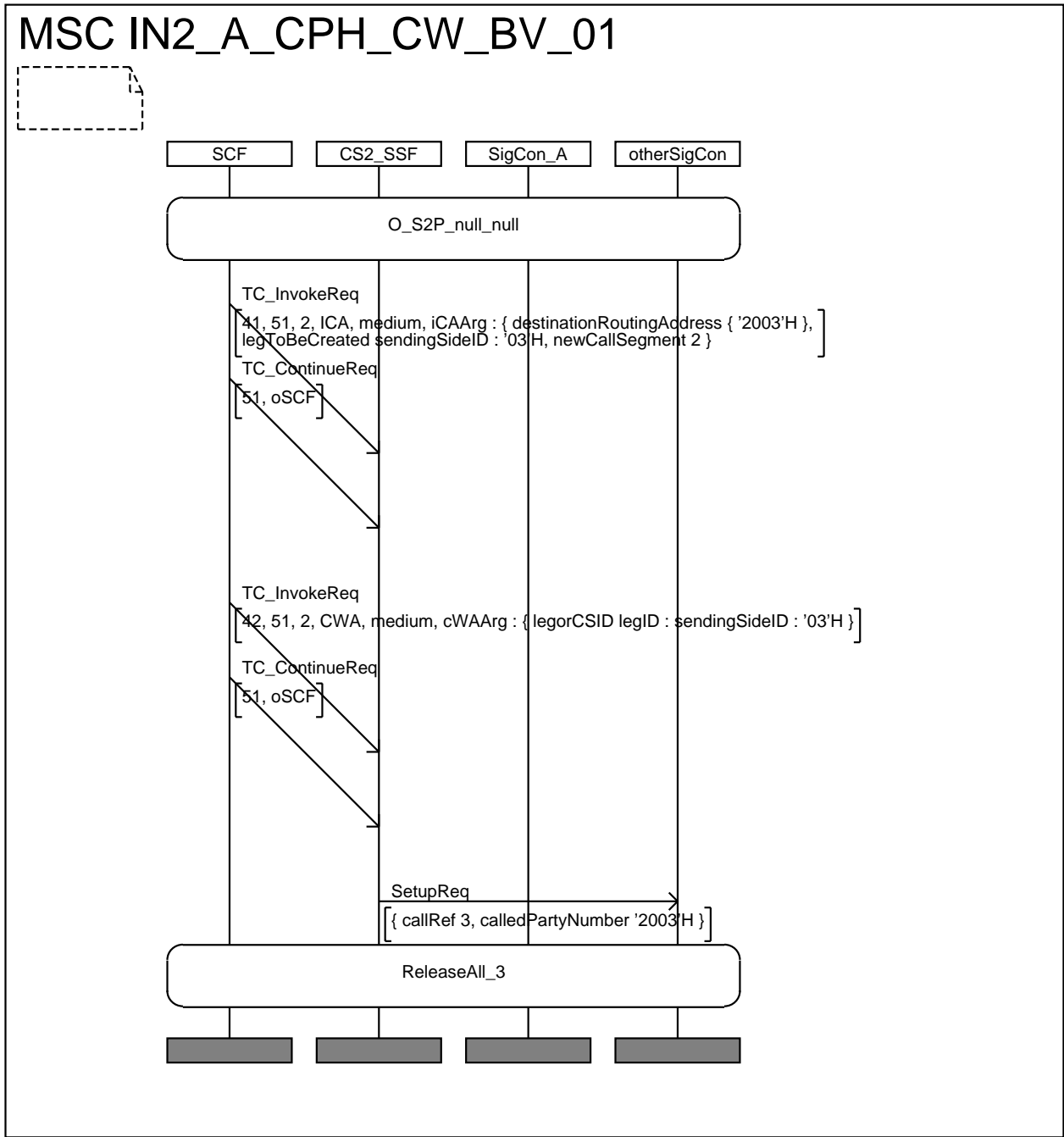
IN2_A_CPH_SL_BI_01	
<b>Purpose:</b>	test that SSF sends an error after a SplitLeg with wrong parameters
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends <b>SplitLeg_invoke</b> to SSF with the following parameters: - legIdToBeSplit=6 - newCallSegment=2
<b>Pass criteria</b>	check that SSF sends to SCF a splitLeg_err containing the parameter: - unknownLegId
<b>Postamble:</b>	ReleaseAll_3

## MSC IN2m\_A\_CPH\_SL\_BI\_01



## 7.4.8 Continue with argument Procedure (CW)

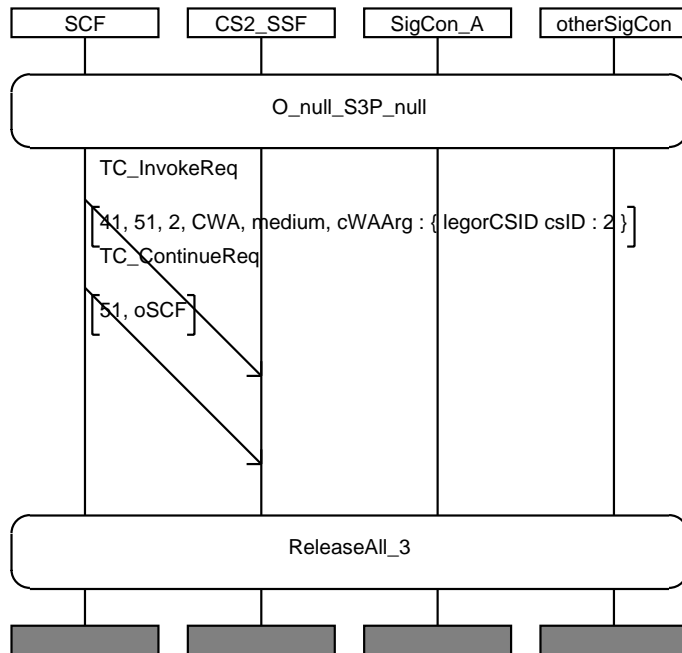
<b>IN2_A_CPH_CW_BV_01</b>	
<b>Purpose:</b>	Test of Continue with argument procedure functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	SCF sends to SSF a <b>InitiateCallAttempt_invoke</b> containing at least the parameters: <ul style="list-style-type: none"> <li>- legToBeCreated=3</li> <li>- newCallSegment=2</li> </ul> followed by <b>ContinueWithArgument_invoke</b> with the following parameters: <ul style="list-style-type: none"> <li>- legId=3</li> </ul>
<b>Pass criteria</b>	Check that SSF sends to SigconC a SetupRequest meaning that BCSM continues
<b>Postamble:</b>	ReleaseAll_3



IN2_A_CPH_CW_BV_02	
<b>Purpose:</b>	Test of Continue with argument procedure functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S3P_null
<b>Test description</b>	SCF sends <b>ContinueWithArgument_invoke</b> with the following parameters CSId=2
<b>Pass criteria</b>	The operation is accepted
<b>Postamble:</b>	ReleaseAll_3

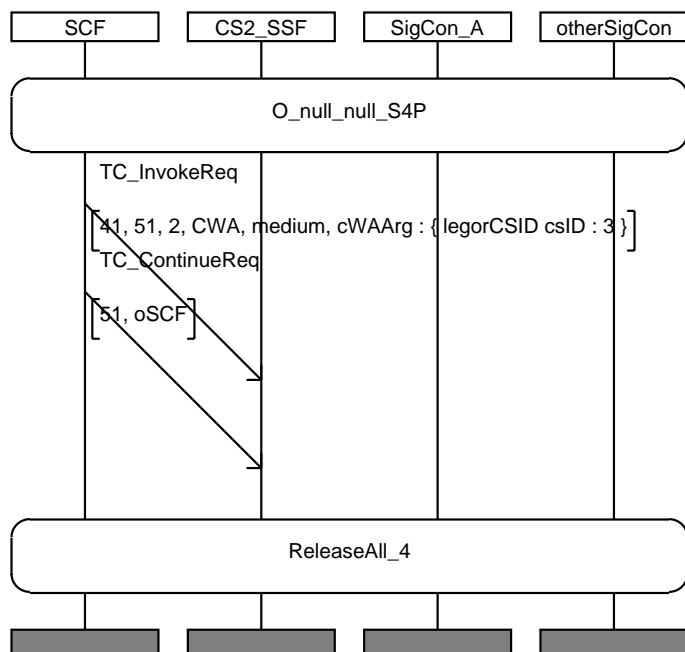


## MSC IN2\_A\_CPH\_CW\_BV\_02



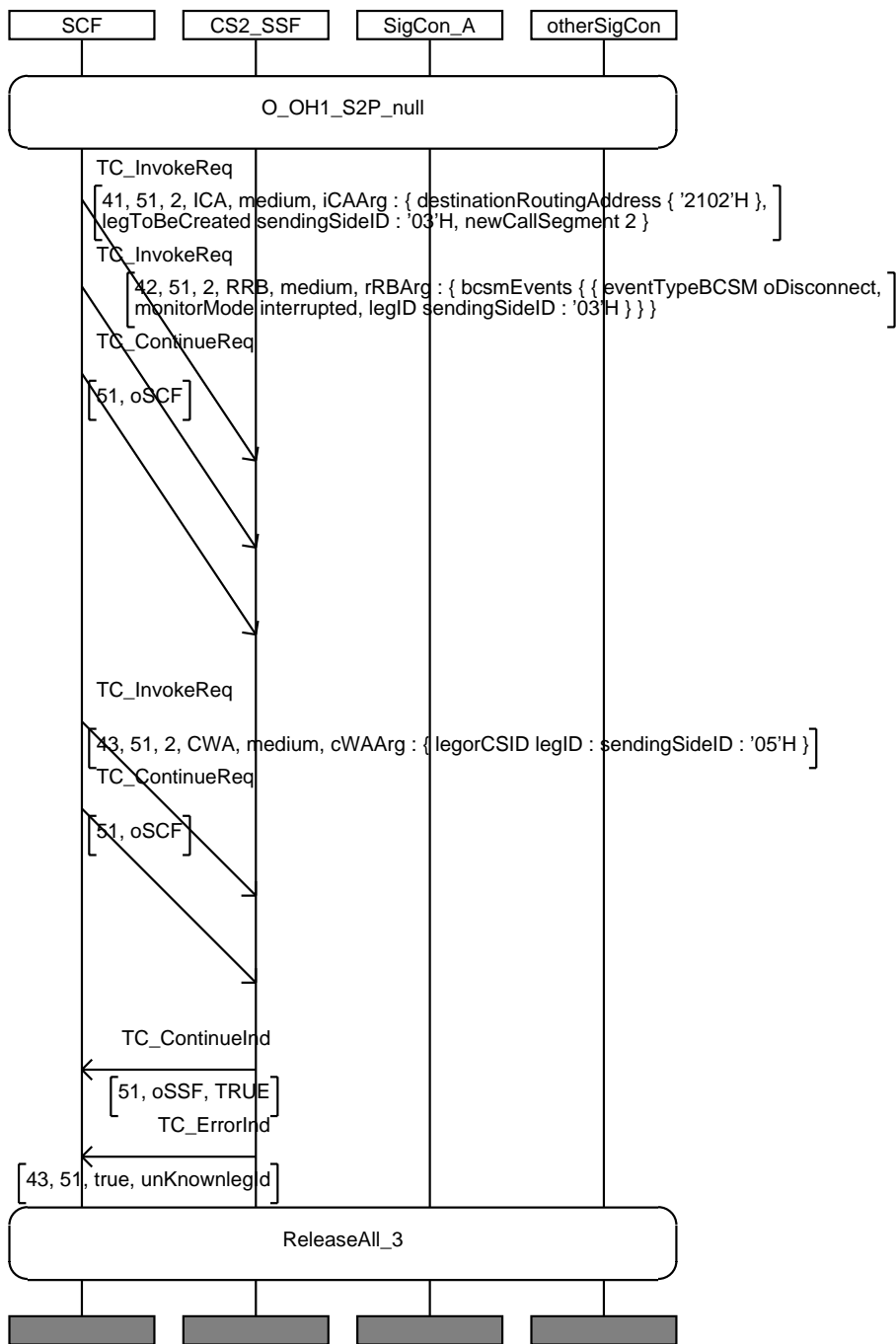
IN2_A_CPH_CW_BV_03	
<b>Purpose:</b>	Test of Continue with argument procedure functionality
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_nul_S4P
<b>Test description</b>	SCF sends <b>ContinueWithArgument_invoke</b> with the following parameters LegId = 4, D party
<b>Pass criteria</b>	The operation is accepted
<b>Postamble:</b>	ReleaseAll_4

## MSC IN2\_A\_CPH\_CW\_BV\_03



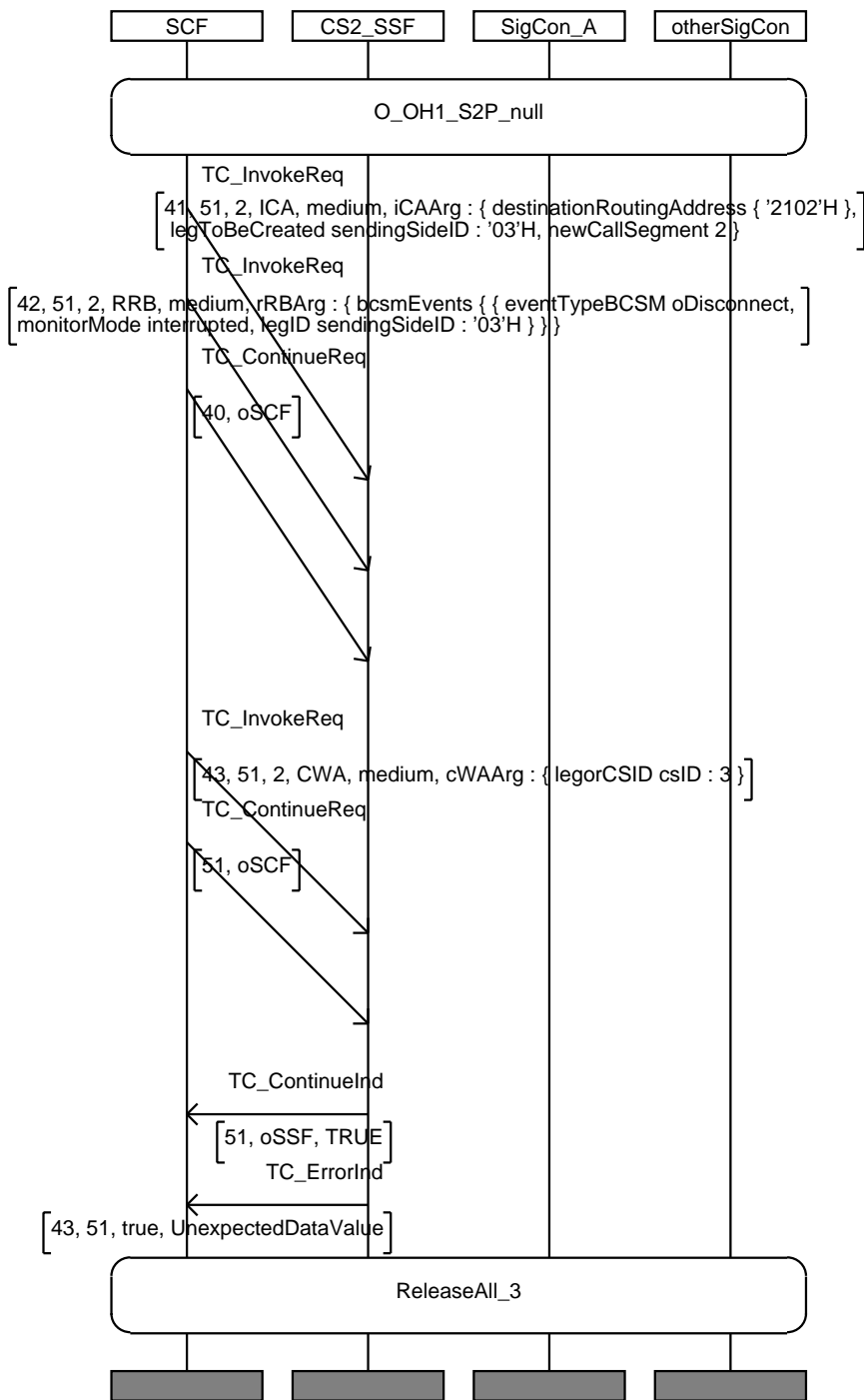
IN2_A_CPH_CW_BI_01	
<b>Purpose:</b>	test that SSF sends an error after a Continue with argument with wrong parameters
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends to SSF a <b>InitiateCallAttempt_invoke</b> containing at least the parameters: <ul style="list-style-type: none"> <li>- legToBeCreated=3</li> <li>- newCallSegment=2</li> </ul> followed by <b>ContinueWithArgument_invoke</b> with the following parameters: <ul style="list-style-type: none"> <li>- legId=5</li> </ul>
<b>Pass criteria</b>	check that SSF sends to SCF a Continue with argument_err containing the parameter: <ul style="list-style-type: none"> <li>- unknownLegId</li> </ul>
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2m\_A\_CPH\_CW\_BI\_01



<b>IN2_A_CPH_CW_BI_02</b>	
<b>Purpose:</b>	test that SSF sends an error after a Continue with argument with wrong parameters
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	SCF sends to SSF a <b>InitiateCallAttempt_invoke</b> containing at least the parameters: <ul style="list-style-type: none"> <li>- legToBeCreated=3</li> <li>- newCallSegment=2</li> </ul> followed by <b>ContinueWithArgument_invoke</b> with the following parameters: <ul style="list-style-type: none"> <li>- Csid=3</li> </ul>
<b>Pass criteria</b>	check that SSF sends to SCF a Continue with argument_err containing the parameter: <ul style="list-style-type: none"> <li>- UnexpectedDataValue</li> </ul>
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2m\_A\_CPH\_CW\_BI\_02

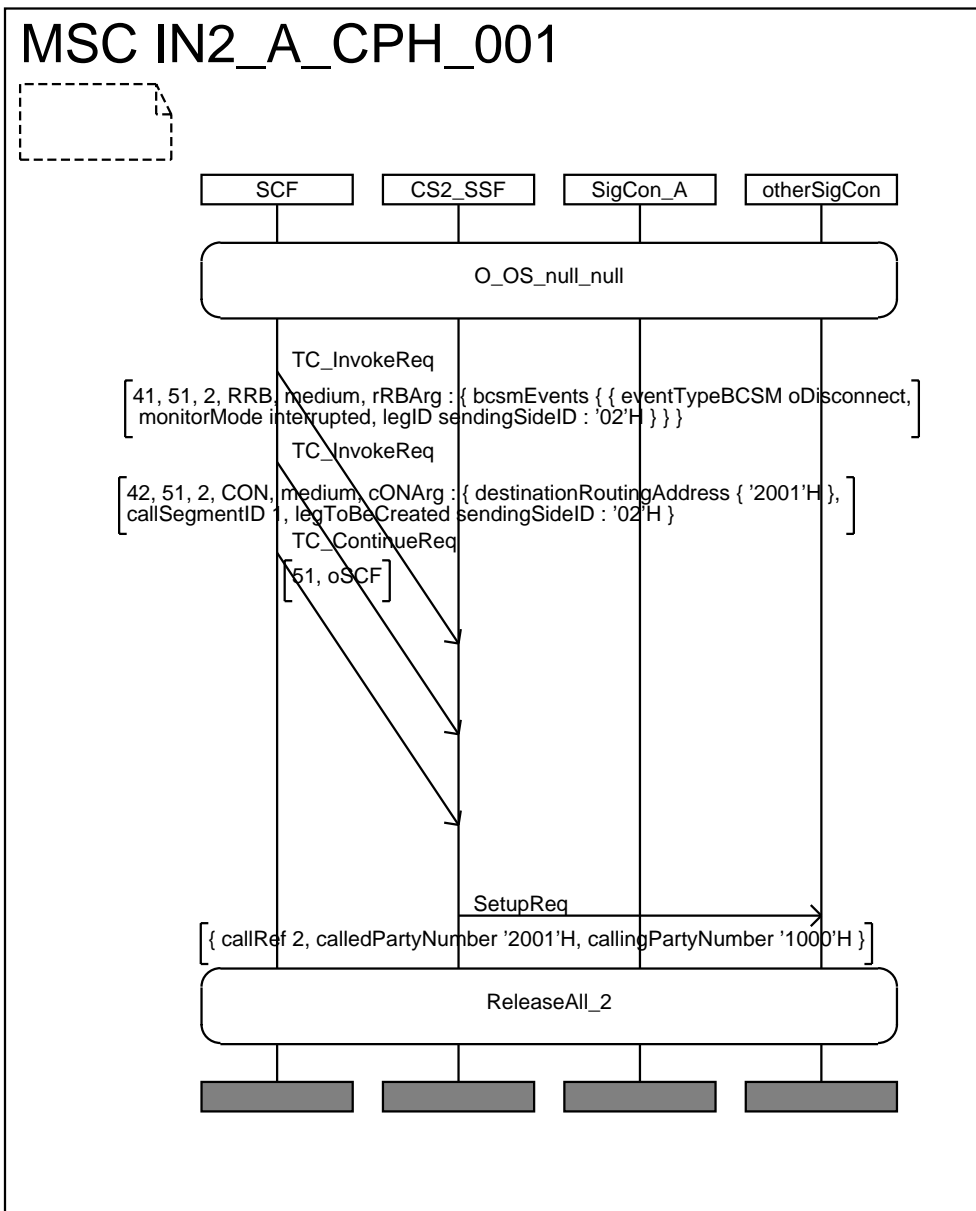


## 7.5 Test Purpose (TP) descriptions for the test of call handling capabilities

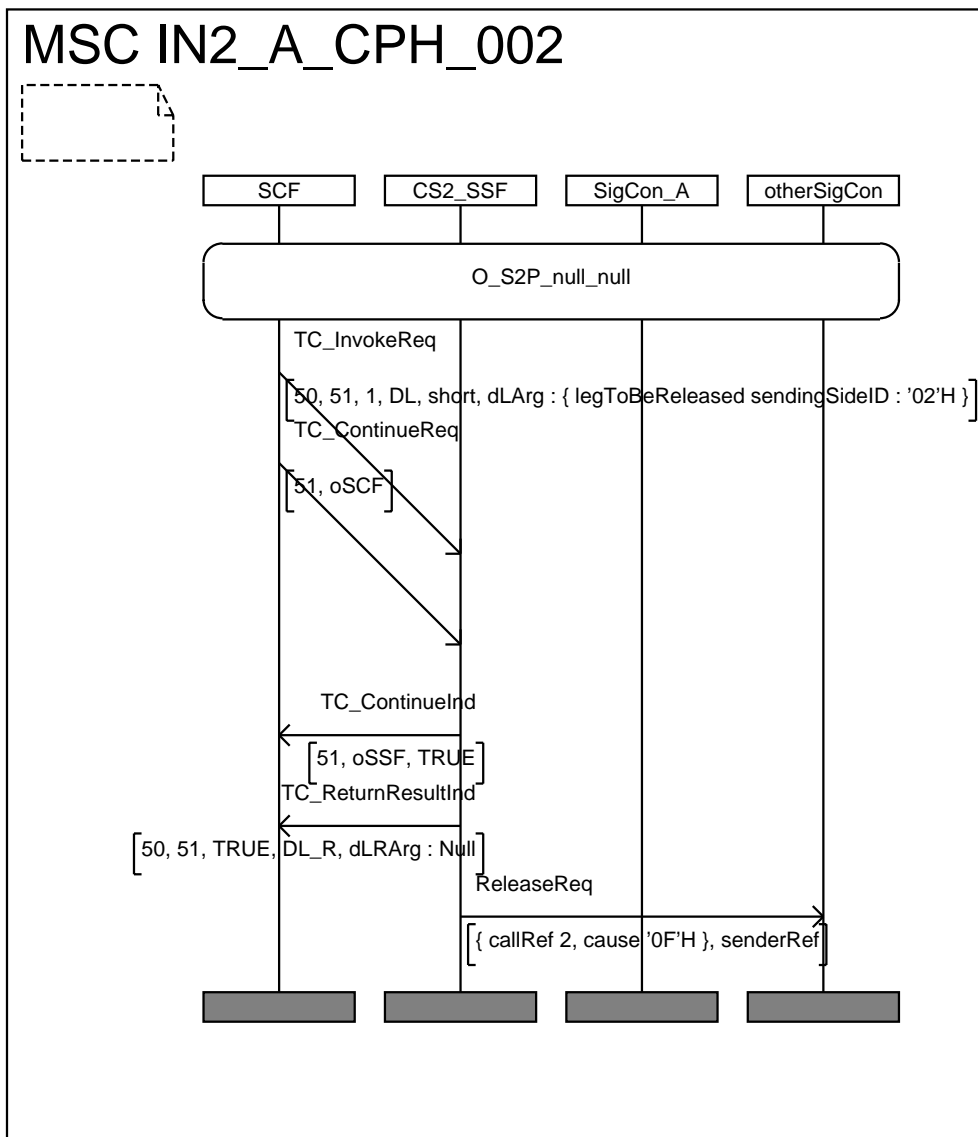
This section includes the TPs for testing the switch call handling capabilities. Due to the great number of possible transitions, the TP are restricted to a limited number of cases covering the switch functionality. Taking the preamble tree as starting point and successively starting from each preamble state, a sequence of operations is applied bringing the IUT to a new state, to be tested.

### 7.5.1 originating (O\_BCSM) trigger (controlling legId = 1)

IN2_A_CPH_001	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OS_null_null
<b>Test description</b>	L1! RequestReportBCSMEvent(2,oDisconnect) L1! Connect(2,1) reaching state O_S2P_null_null
<b>Pass criteria</b>	CP1-2? SetUpReq
<b>Postamble:</b>	ReleaseAll_2

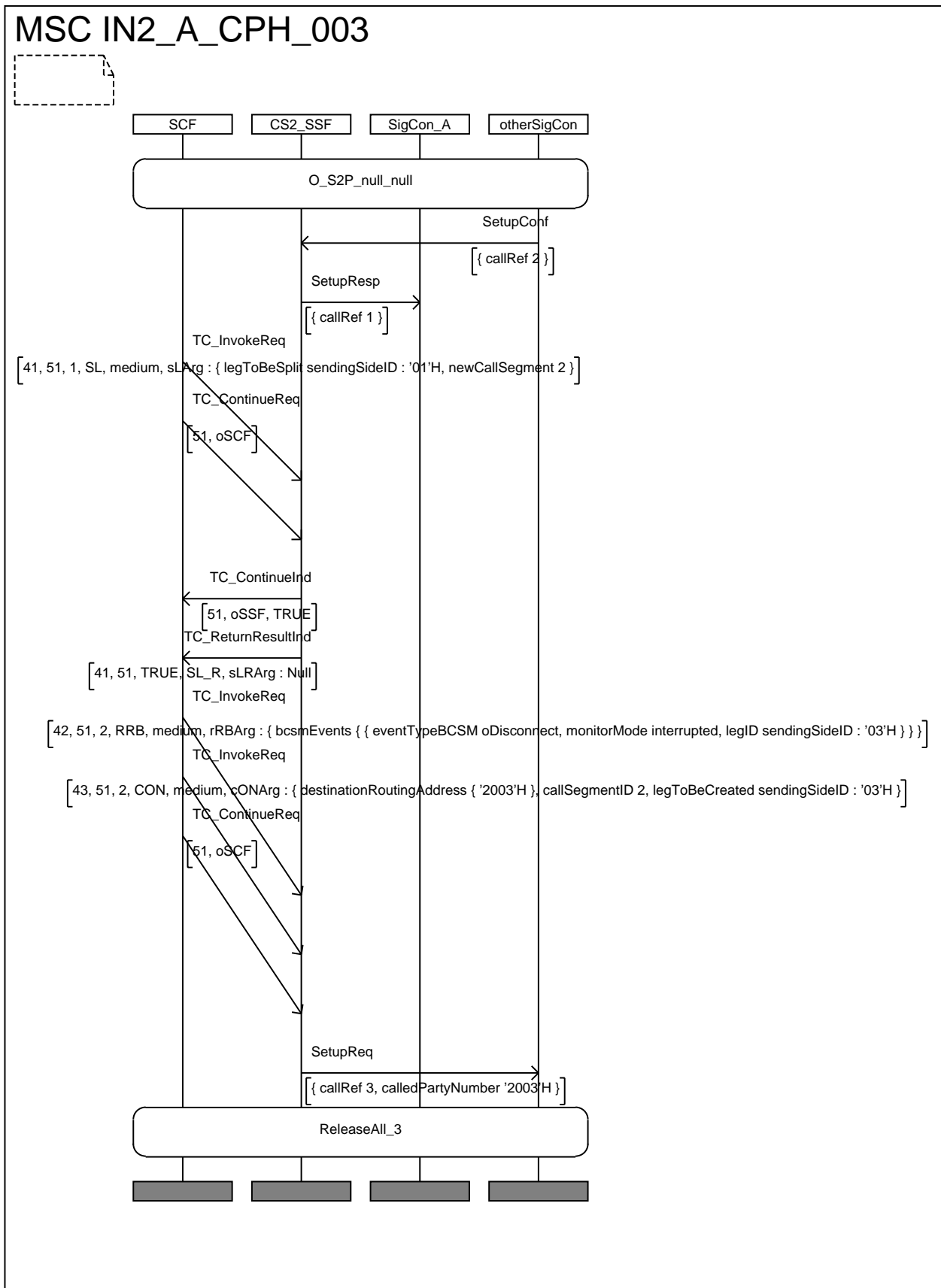


IN2_A_CPH_002	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult reaching state Null
<b>Pass criteria</b>	CP1_1? ReleaseReq
<b>Postamble:</b>	None



IN2_A_CPH_003	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	CP1-2! SetupConf, B party answers L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument (CsID = 1) L1! RequestReportBCSMEvent(3,oDisconnect) L1! Connect(3,2) reaching state O_OH(1)_S2P_null
<b>Pass criteria</b>	CP1-3? SetUpReq
<b>Postamble:</b>	ReleaseAll_3

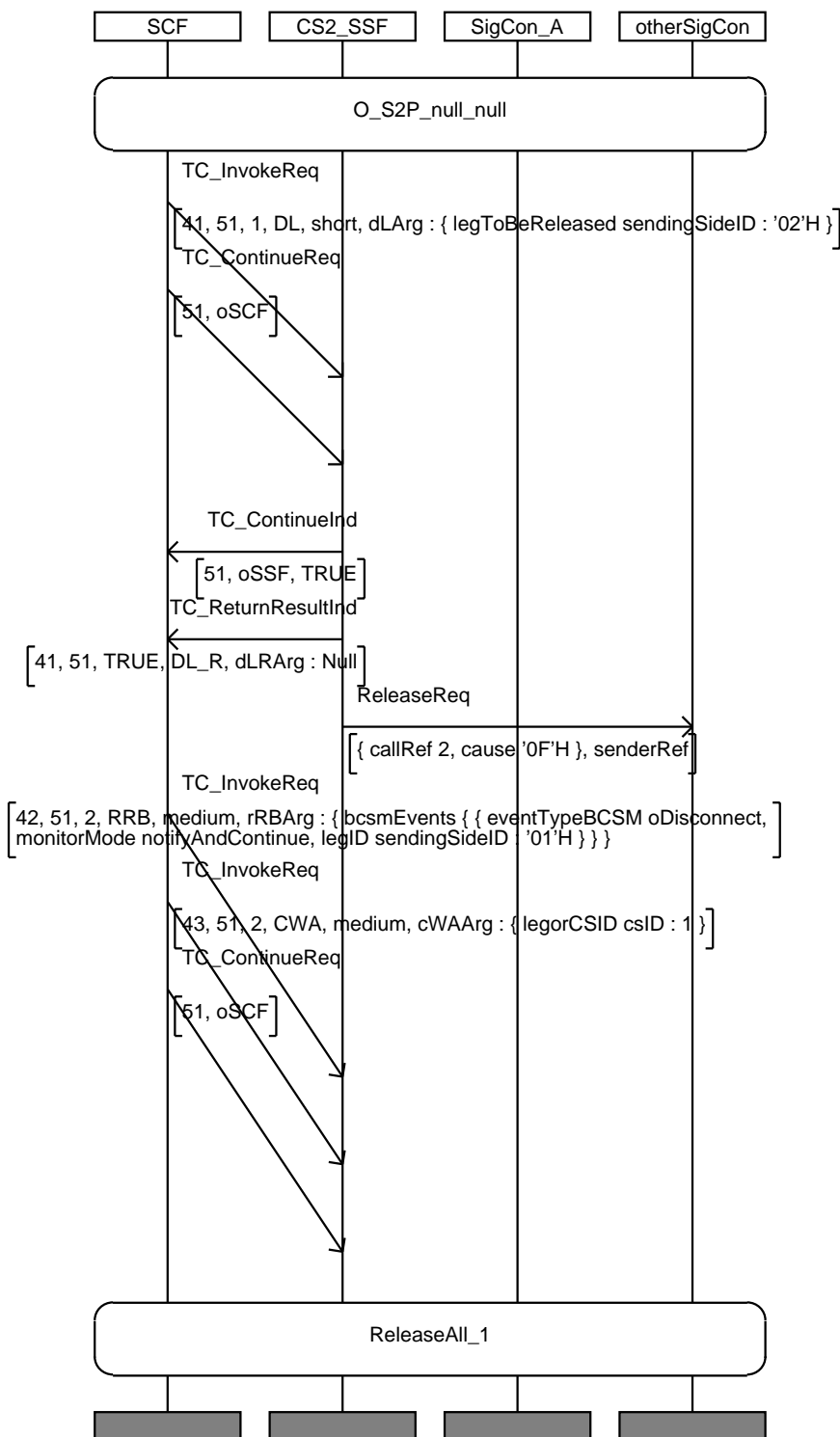




IN2_A_CPH_004	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	

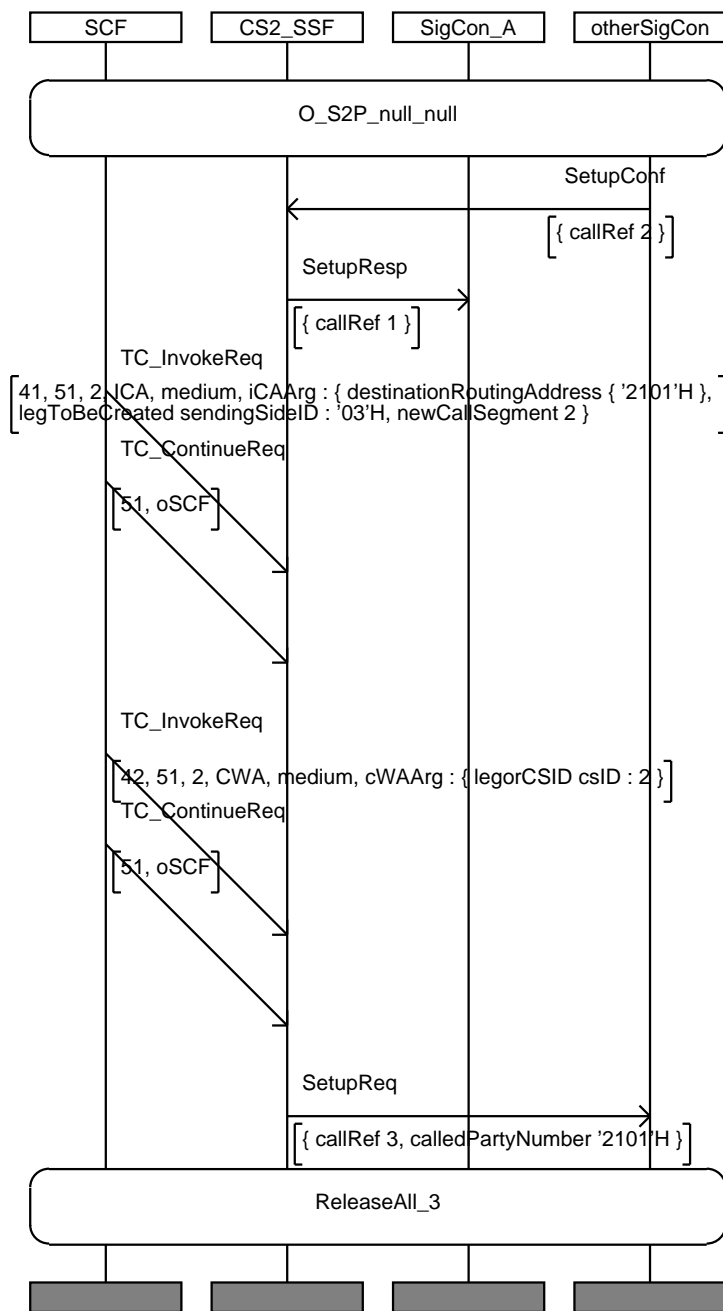
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult L1! ContinueWithArgument (CsID = 1) Reaching state O_S1P_null_null:
<b>Pass criteria</b>	CP1_2? ReleaseReq
<b>Postamble:</b>	ReleaseAll_1

# MSC IN2\_A\_CPH\_004



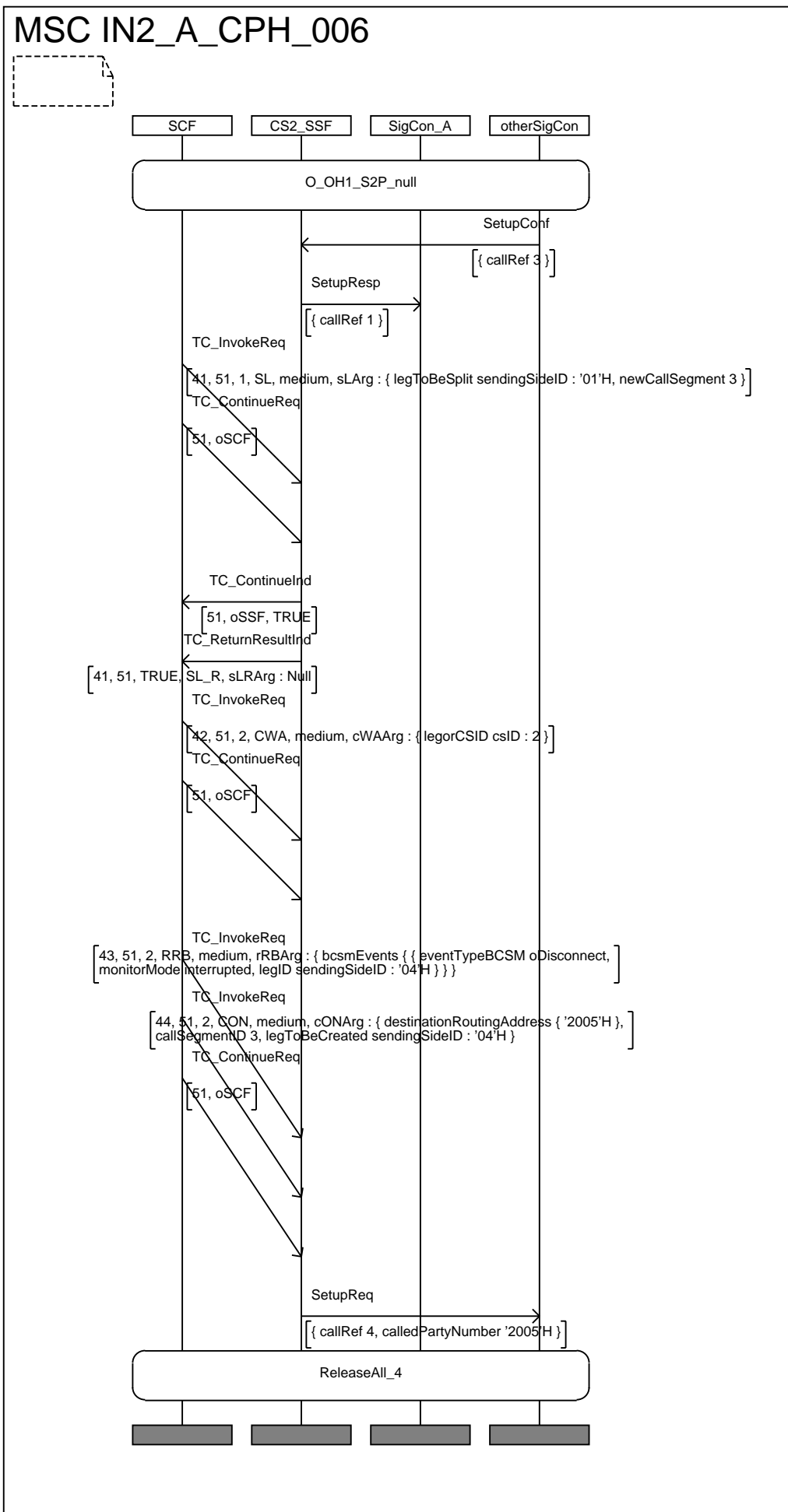
IN2_A_CPH_005	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test description</b>	CP1-2! SetupConf, B party answers L1! InitiateCallAttempt(3,2) L1! ContinueWithArgument(LegId=3) Reaching state O_S2P_S1P_null
<b>Pass criteria</b>	CP1_3? SetupReq
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_005



IN2_A_CPH_006	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	CP1-3! SetupConf, C party answers L1! SplitLeg(1,3) L1? SplitLegReturnResult L1! ContinueWithArgument (CslD = 2) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(4,3) Reaching state O_OH(1)_OH(1)_S2P
<b>Pass criteria</b>	CP1_4? SetupReq
<b>Postamble:</b>	ReleaseAll_4

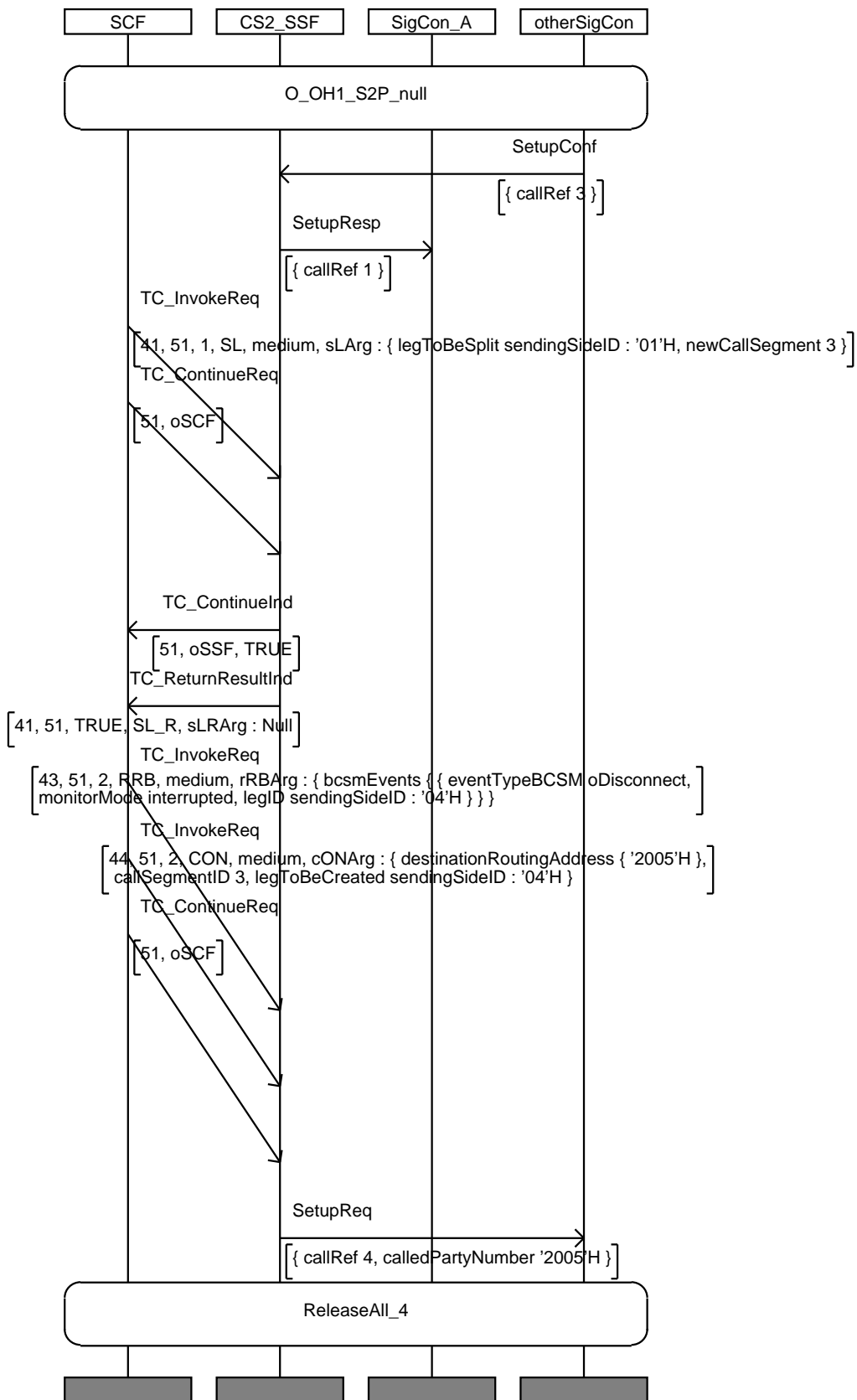
# MSC IN2\_A\_CPH\_006



<b>IN2_A_CPH_007</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	CP1-3! SetupConf, C party answers L1! SplitLeg(1,3) L1? SplitLegReturnResult L1! ContinueWithArgument (CsID = 2) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(4,3) Reaching state O_OH(1)_OH(1)_S2P
<b>Pass criteria</b>	CP1_4? SetupReq
<b>Postamble:</b>	ReleaseAll_4

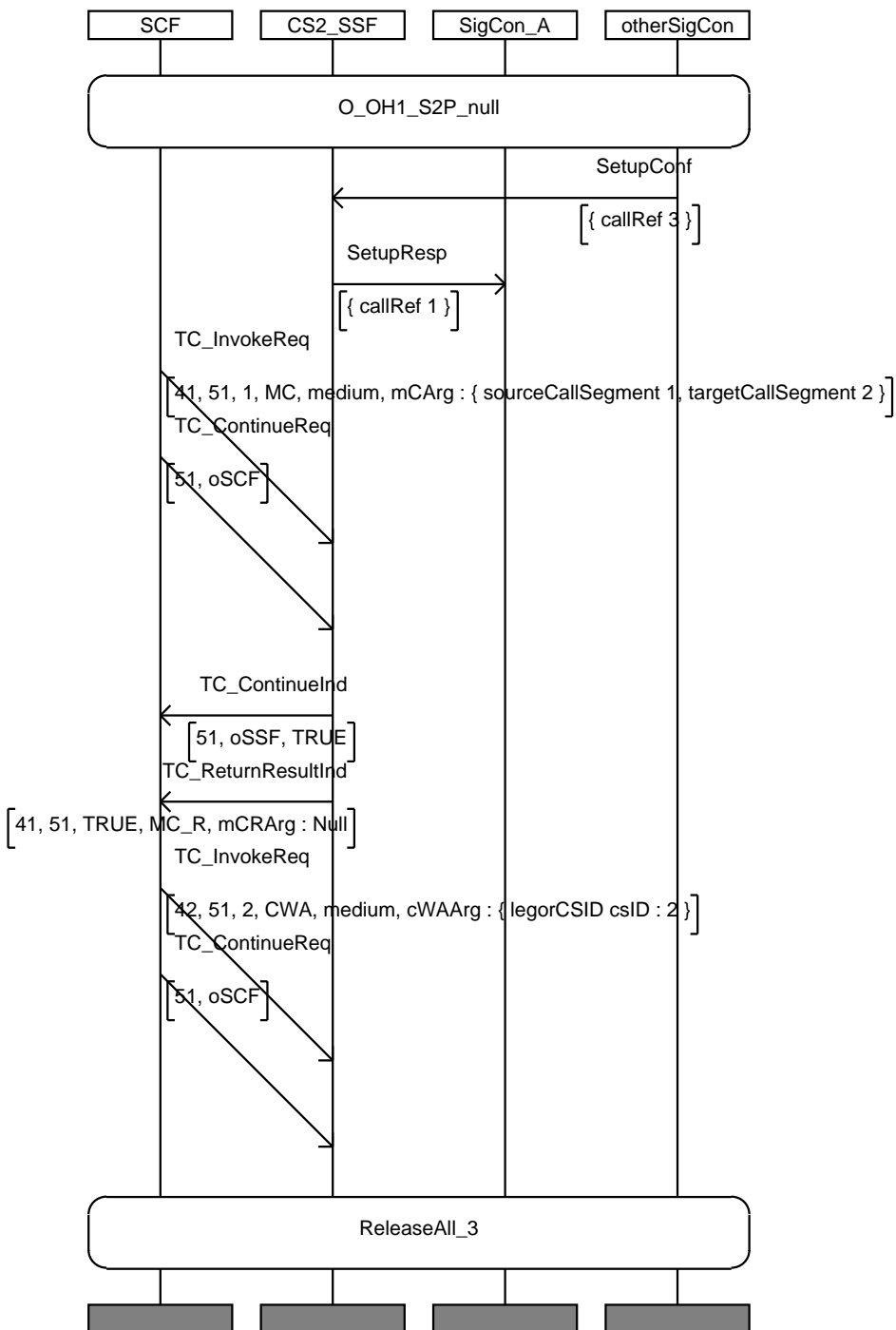


# MSC IN2\_A\_CPH\_007



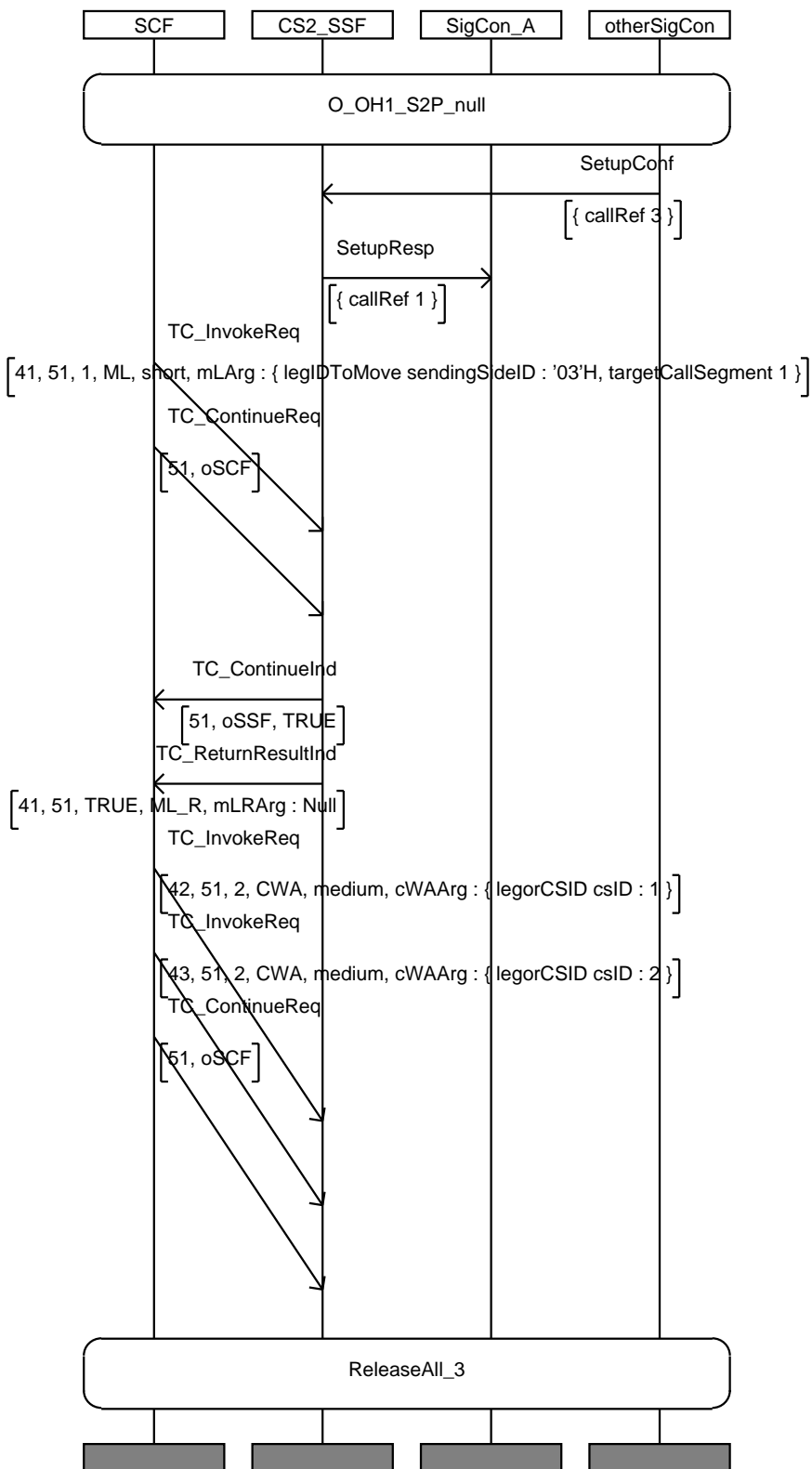
IN2_A_CPH_008	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	CP1-3! SetupConf, C party answers L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_S3P_null
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_008



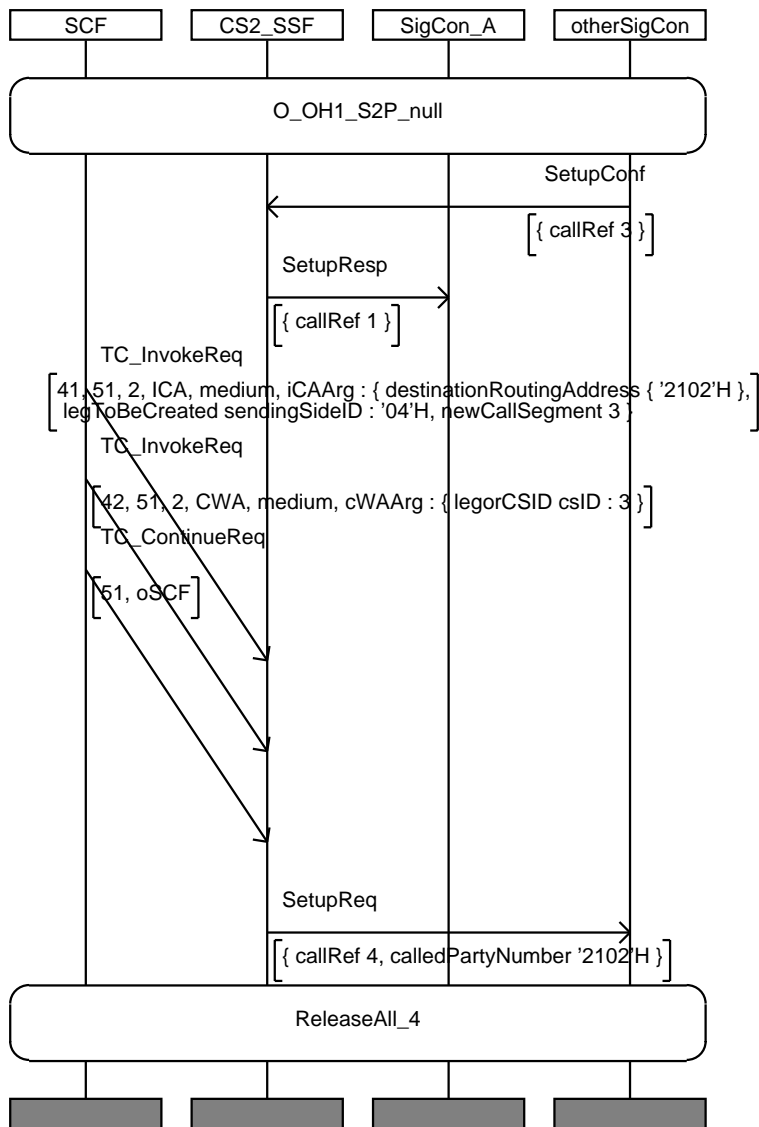
IN2_A_CPH_009	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	CP1-3! SetupConf, C party answers L1! MoveLeg(3,1) L1! ContinueWithArgument(Csld=1) L1! ContinueWithArgument(Csld=2) Reaching state O_OH(2)_1P_null
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_009



IN2_A_CPH_010	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_S2P_null
<b>Test description</b>	CP1-3! SetupConf, C party answers L1! InitiateCallAttempt(4,3) L1! ContinueWithArgument(LegId=4) Reaching state O_OH(1)_S2P_S1P
<b>Pass criteria</b>	CP1_4? SetUpReq
<b>Postamble:</b>	ReleaseAll_4

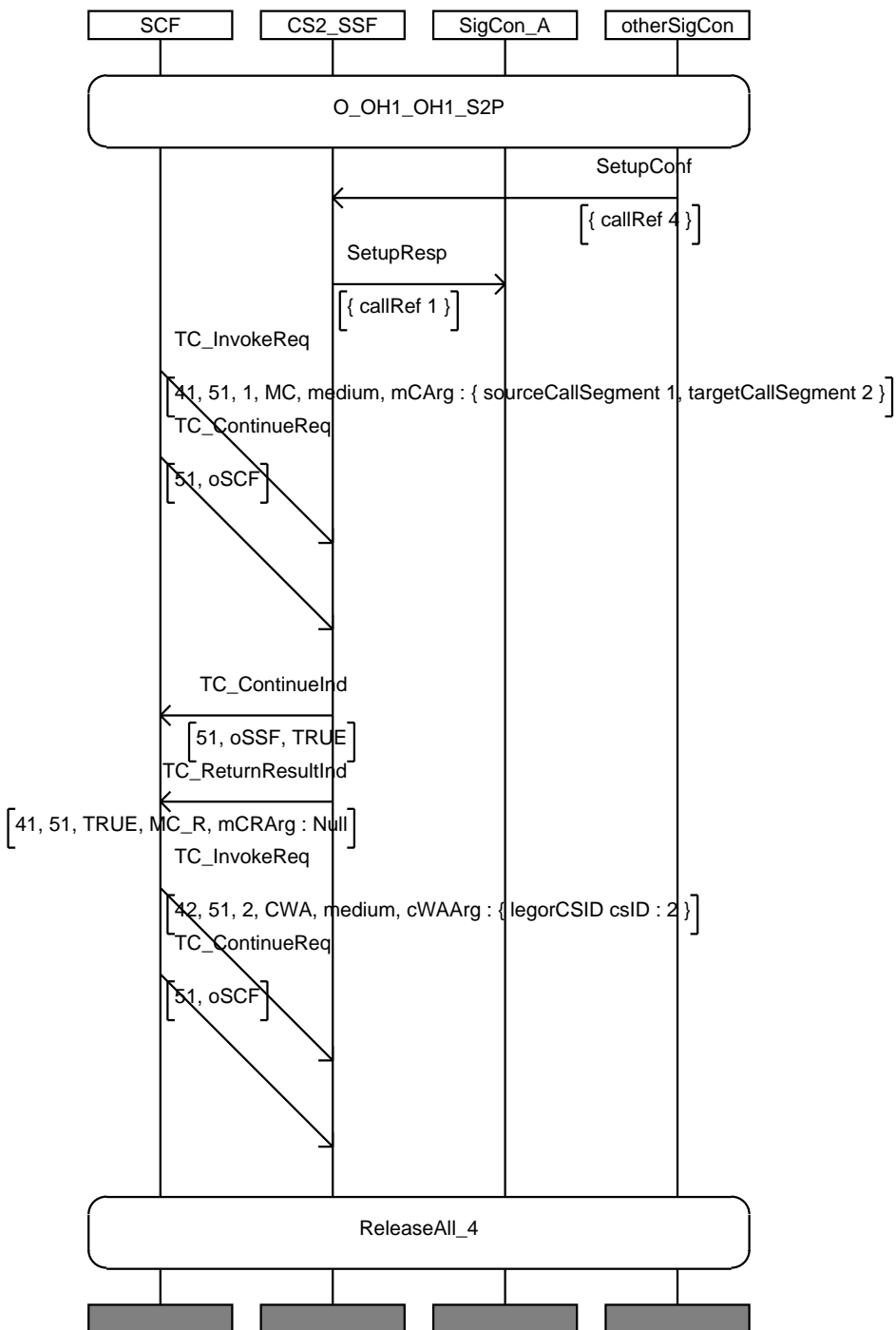
# MSC IN2\_A\_CPH\_010



IN2_A_CPH_011	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_OH(1)_S2P
<b>Test description</b>	CP1-4! SetupConf L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_OH(2)_S2P
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_4

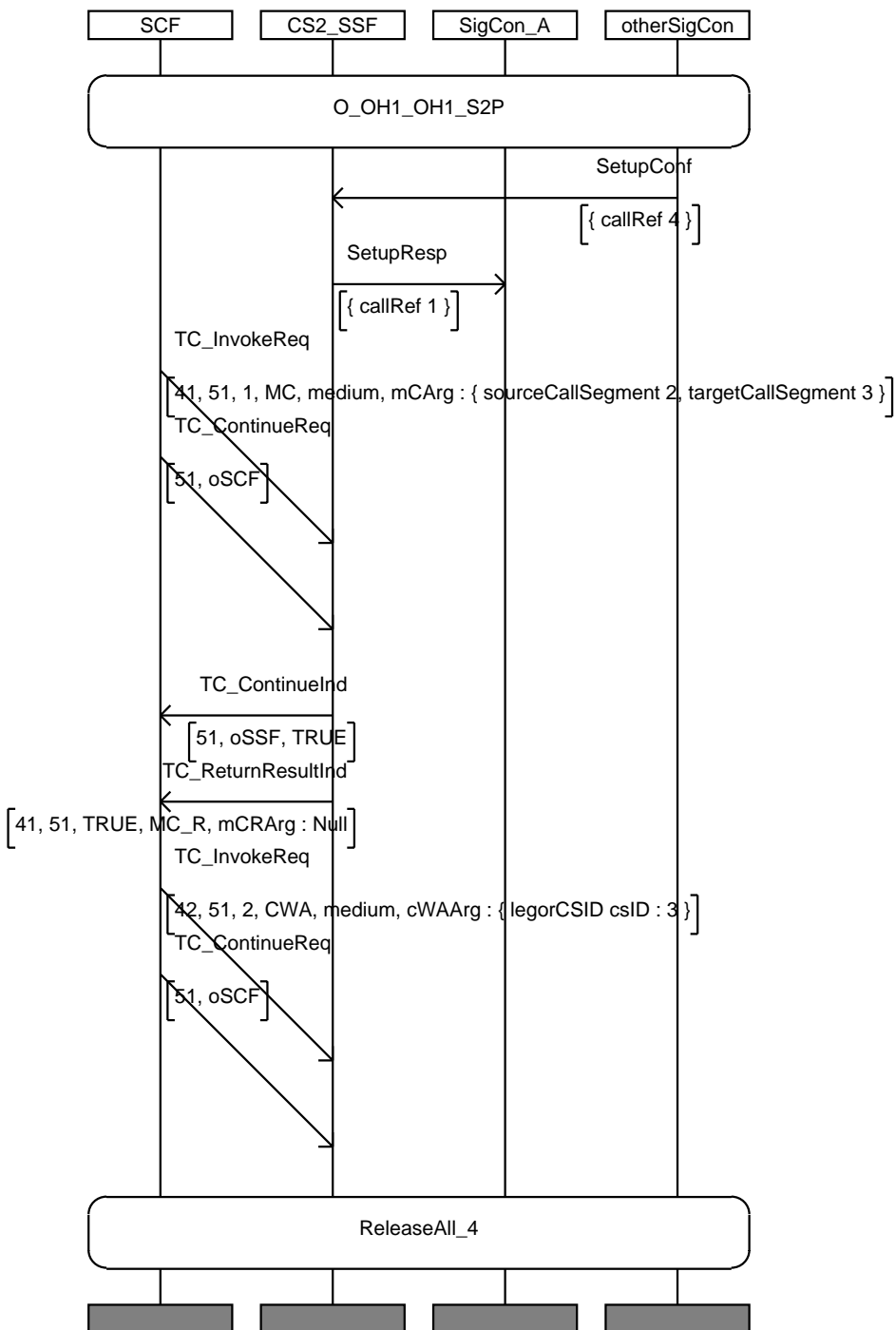


# MSC IN2\_A\_CPH\_011



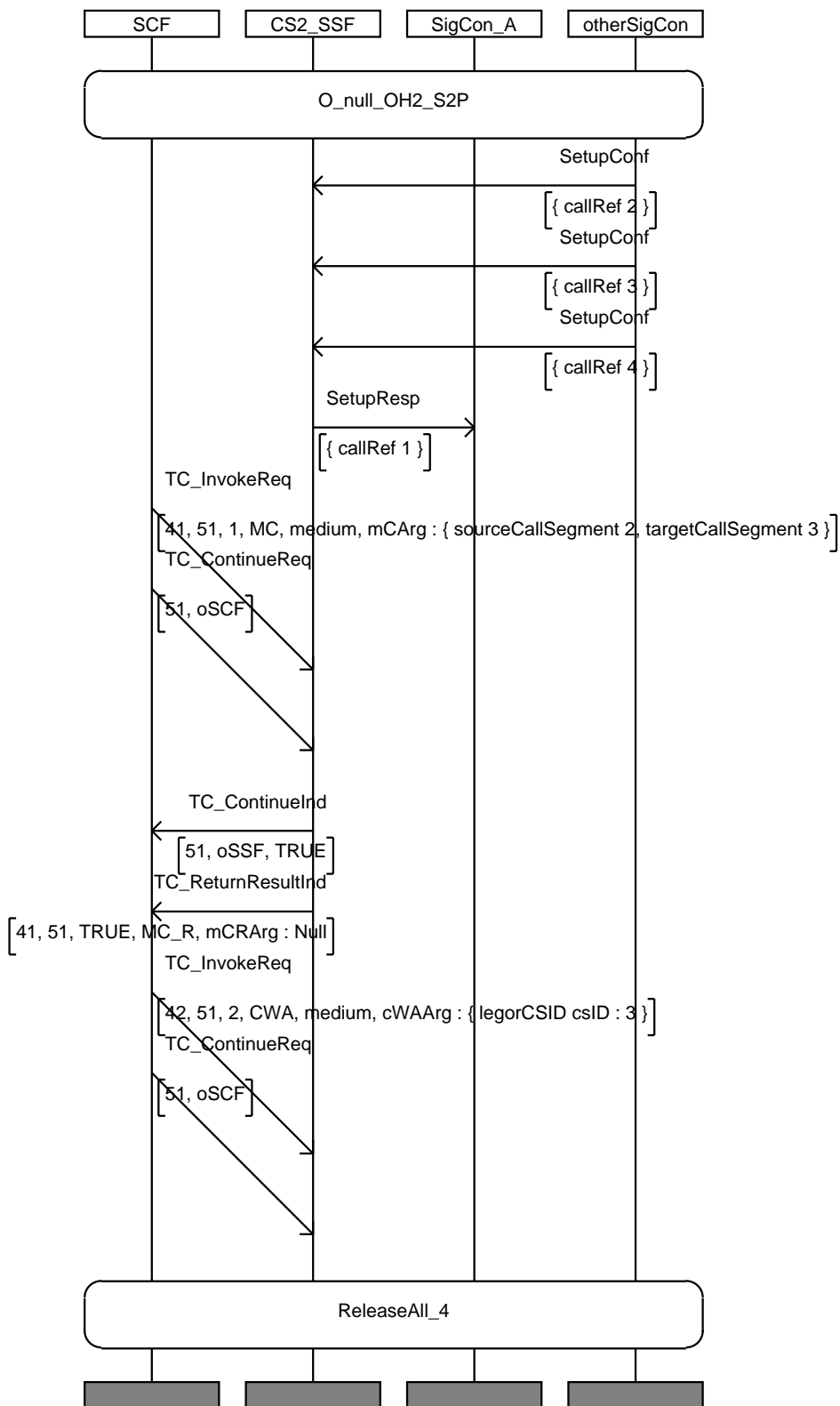
IN2_A_CPH_012	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_OH(1)_OH(1)_S2P
<b>Test description</b>	CP1-4! SetupConf L1! MergeCallSegments(2,3) L1! ContinueWithArgument(CsId=3) Reaching state O_OH(1)_null_S3P
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_4

# MSC IN2\_A\_CPH\_012



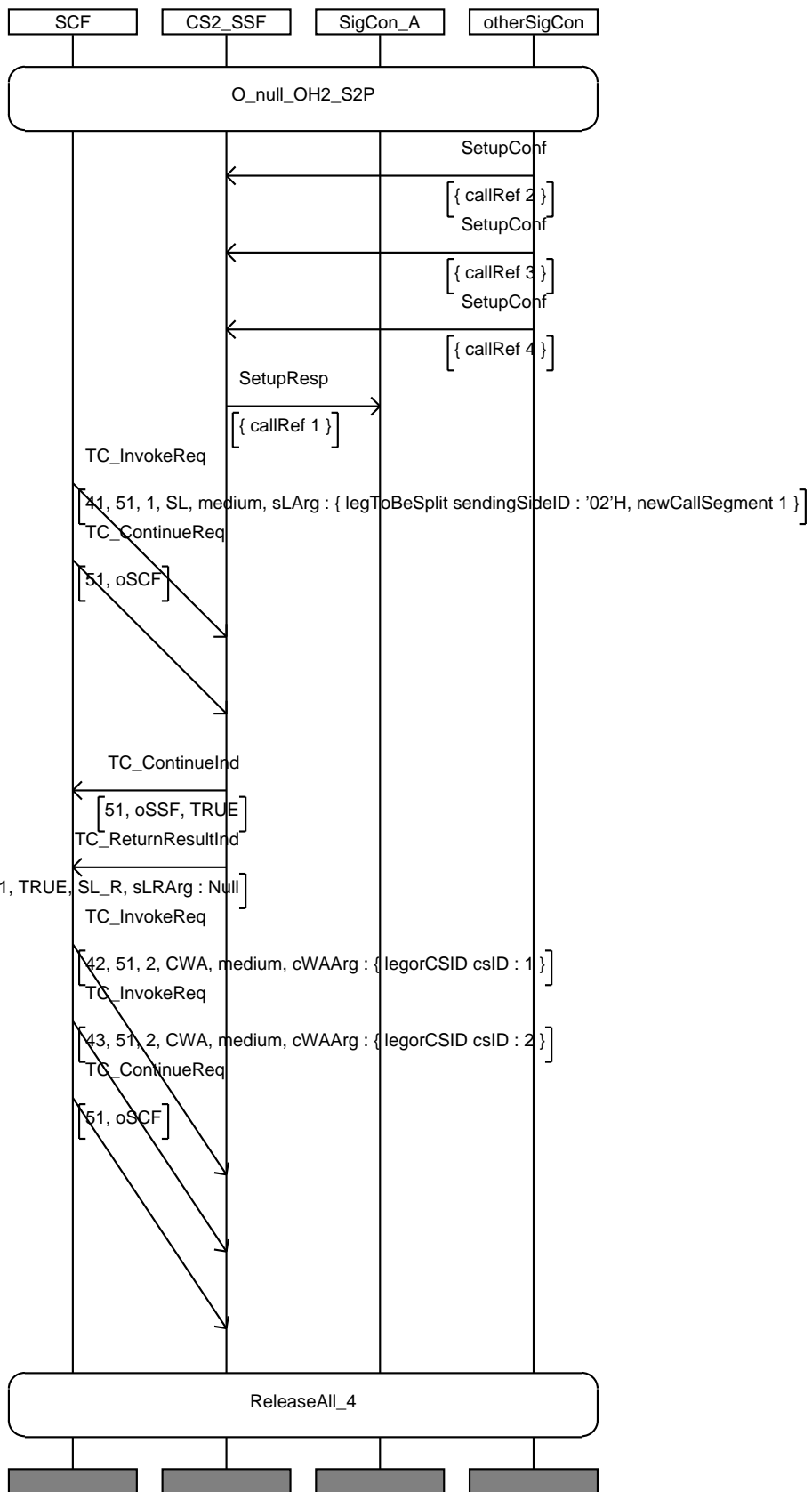
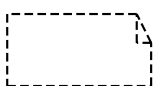
<b>IN2_A_CPH_013</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P
<b>Test description</b>	CP1-2! SetupConf CP1-3! SetupConf CP1-4! SetupConf L1! MergeCallSegments(2,3) L1! ContinueWithArgument(CsId=3) Reaching state O_null_null_S4P
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_4

# MSC IN2\_A\_CPH\_013



<b>IN2_A_CPH_014</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P
<b>Test description</b>	CP1-2! SetupConf CP1-3! SetupConf CP1-4! SetupConf L1! SplitLeg(2,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=2) Reaching state O_OH(1)_OH(1)_S2P
<b>Pass criteria</b>	SSF sends SplitLegReturnResult
<b>Postamble:</b>	ReleaseAll_4

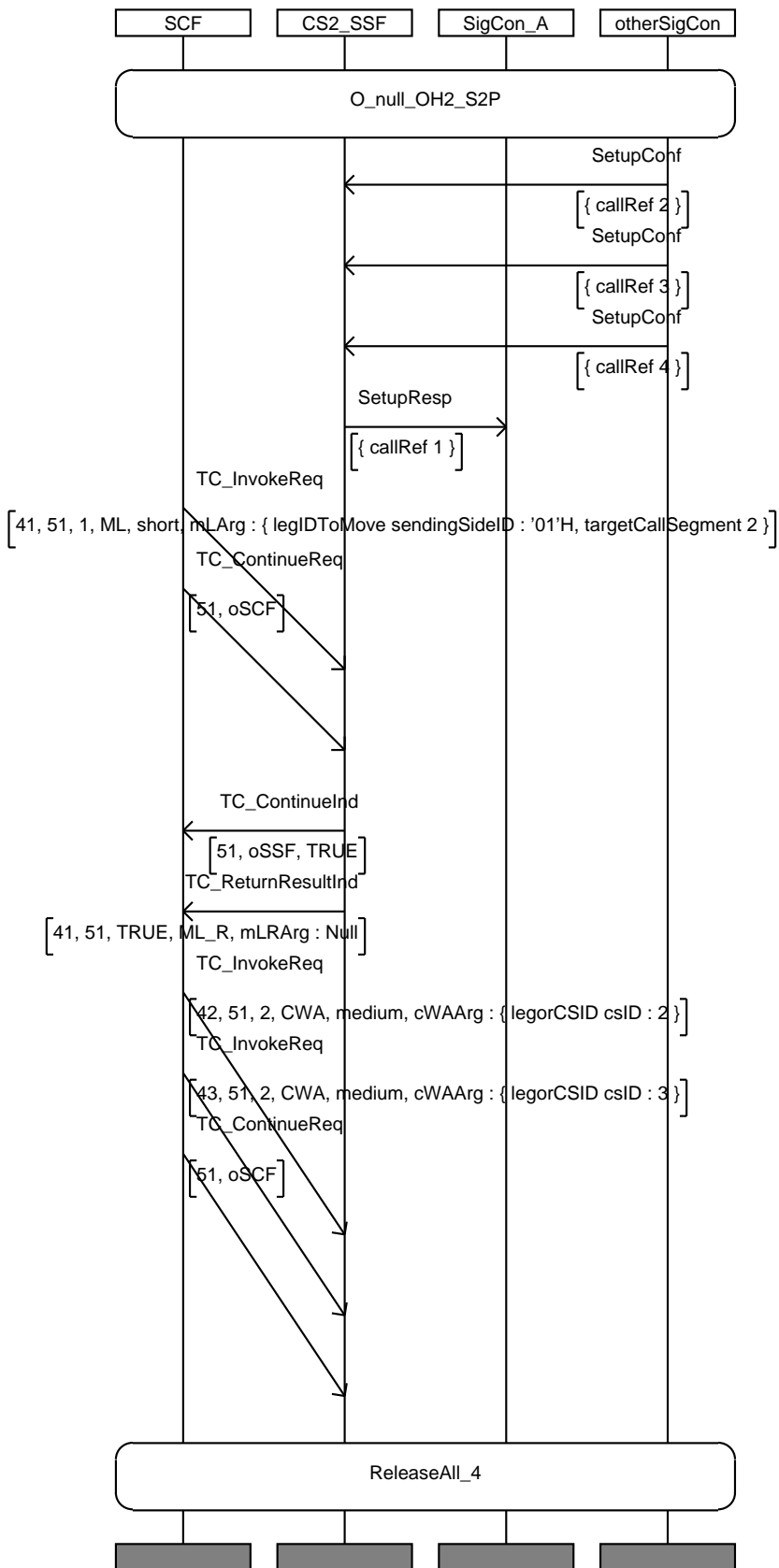
# MSC IN2\_A\_CPH\_014



IN2_A_CPH_015	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P
<b>Test description</b>	CP1-2! SetupConf CP1-3! SetupConf, C party answers CP1-4! SetupConf, D party answers L1! MoveLeg(1,2) L1!ContinueWithArgument(Csld=2) L1!ContinueWithArgument(Csld=3) Reaching state O_null_S3P_OH(1)
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_4

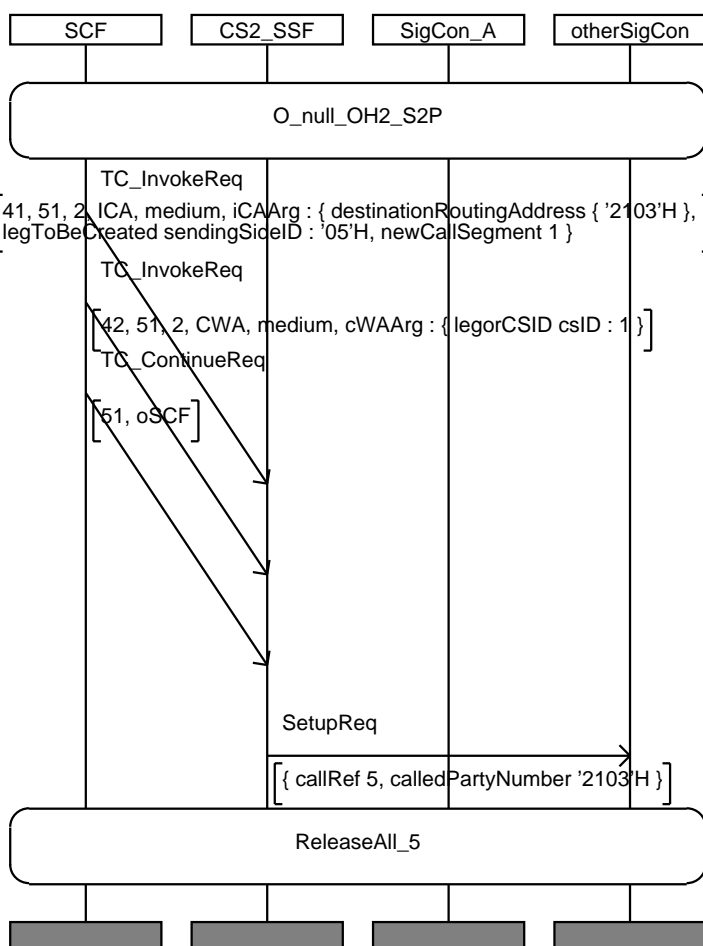


# MSC IN2\_A\_CPH\_015



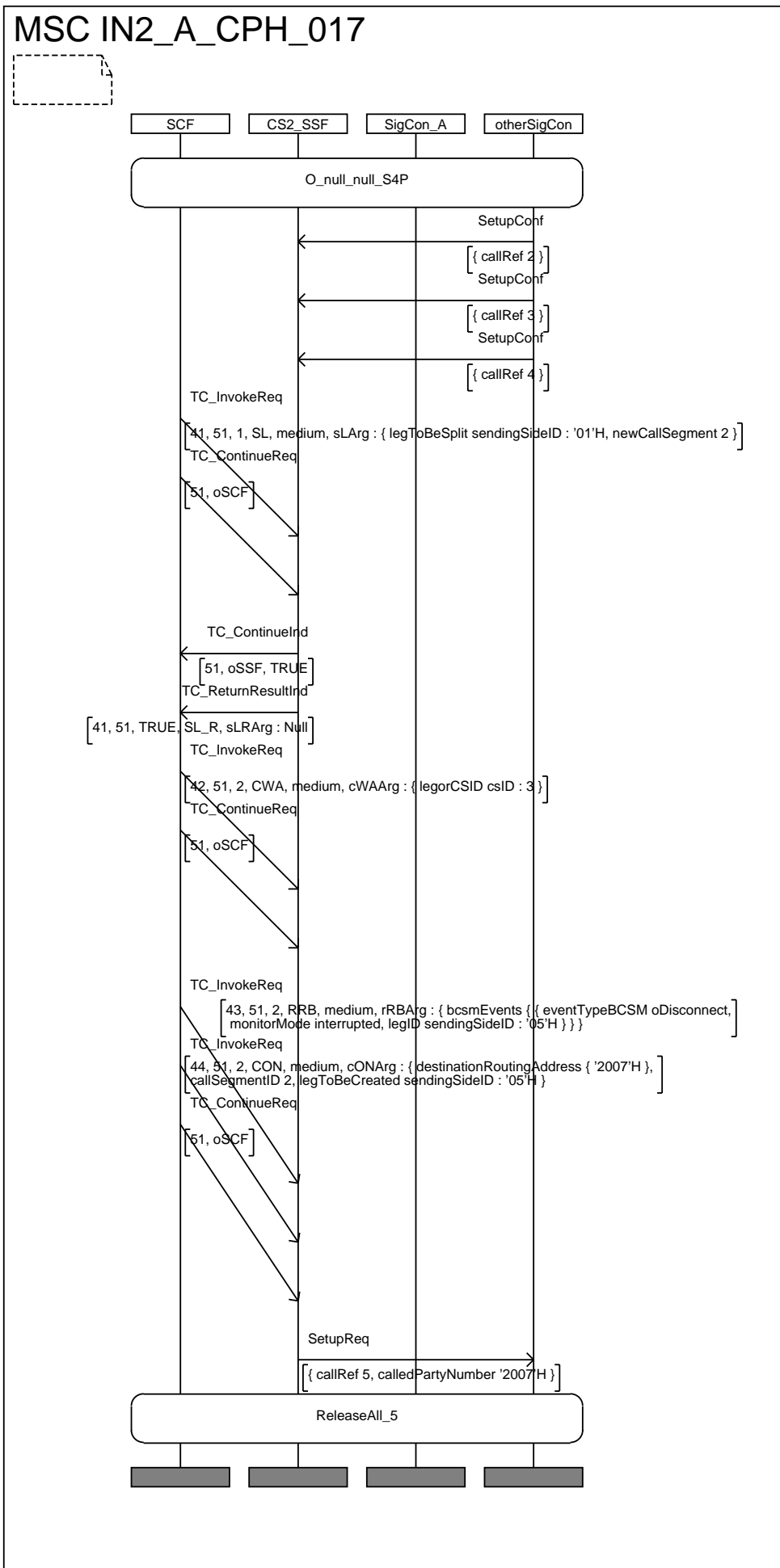
IN2_A_CPH_016	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P
<b>Test description</b>	L1! InitiateCallAttempt(5,1) L1! ContinueWithArgument(LegId=5) Reaching state O_S1P_OH(2)_S2P
<b>Pass criteria</b>	CP1_5? SetUpReq
<b>Postamble:</b>	ReleaseAll_5

## MSC IN2\_A\_CPH\_016

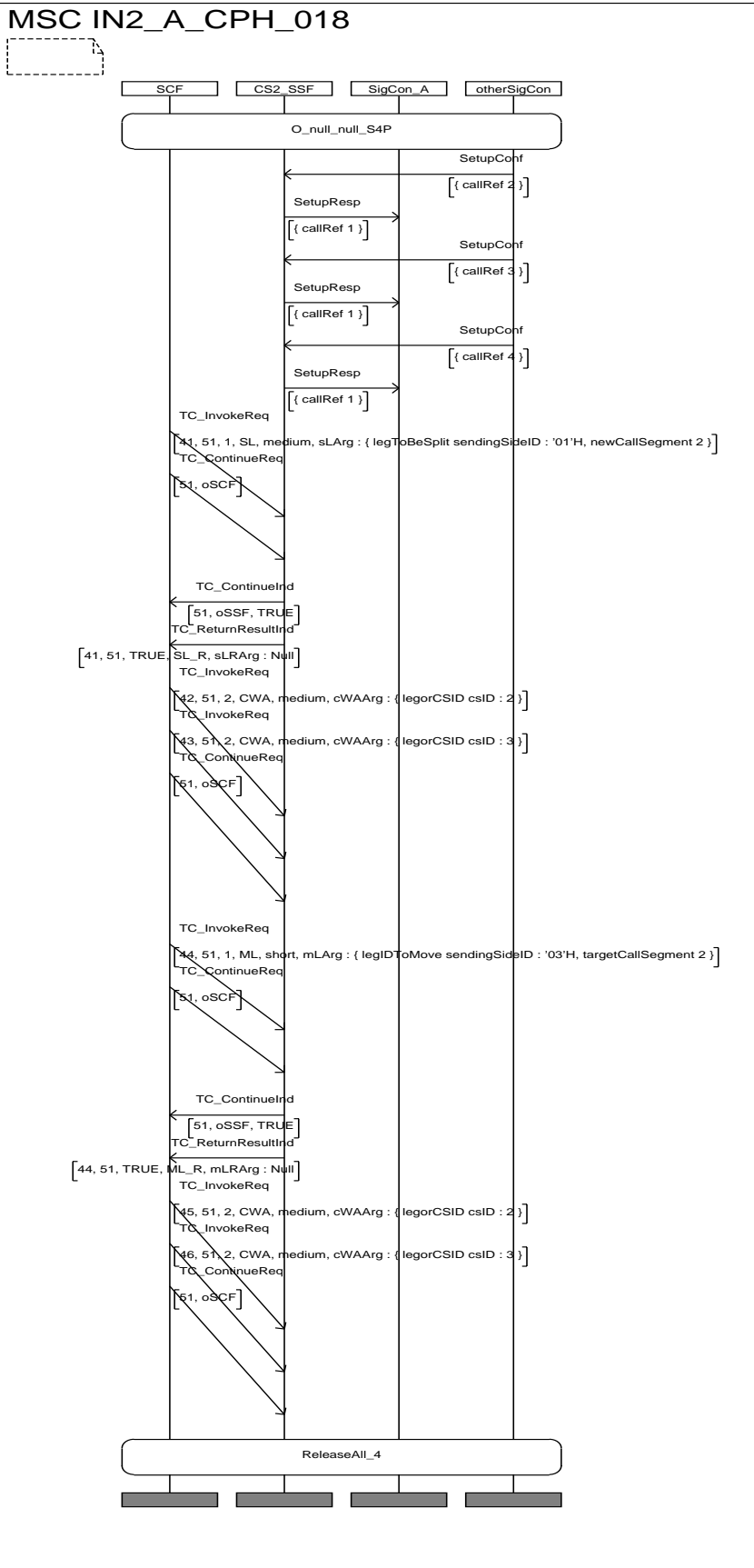


<b>IN2_A_CPH_017</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P
<b>Test description</b>	CP1-2! SetupConf CP1-3! SetupConf, C party answers CP1-4! SetupConf, D party answers L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RequestReportBCSMEvent(5,oDisconnect) L1! Connect(5,2) Reaching state O_null_S2P_OH(3)
<b>Pass criteria</b>	CP1_5? SetUpReq
<b>Postamble:</b>	ReleaseAll_5

# MSC IN2\_A\_CPH\_017

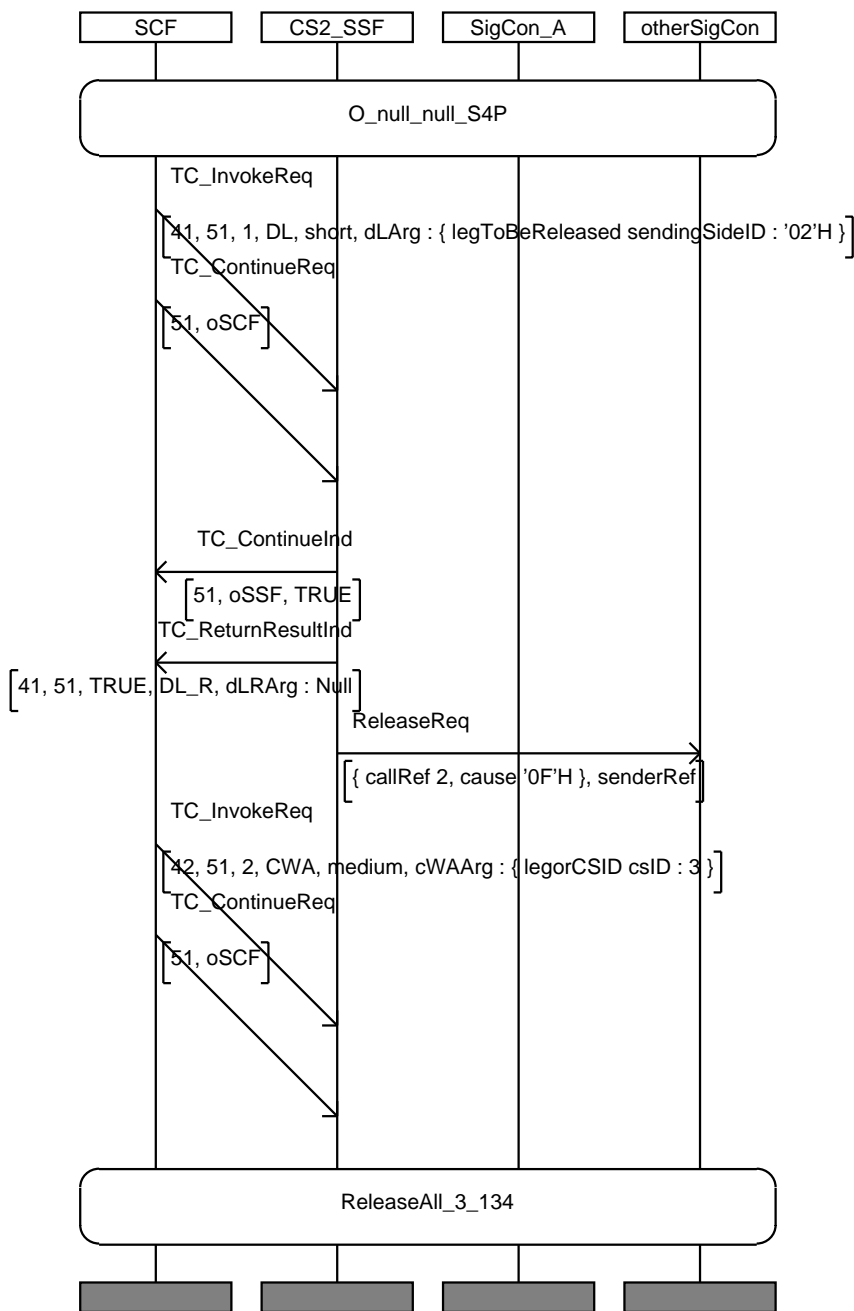


<b>IN2_A_CPH_018</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P
<b>Test description</b>	CP1-2! SetupConf CP1-3! SetupConf, C party answers CP1-4! SetupConf, D party answers L1! SplitLeg(1,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) L1! MoveLeg(3,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state O_null_S2P_OH(2)
<b>Pass criteria</b>	SSF sends SplitLegReturnResult and SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_4



IN2_A_CPH_019	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P
<b>Test description</b>	L1! DisconnectLeg(2) L1!ContinueWithArgument(CsId=3) L1? DisconnectLegReturnResult Reaching state O_null_null_S3P
<b>Pass criteria</b>	CP1_2? ReleaseReq
<b>Postamble:</b>	ReleaseAll_3

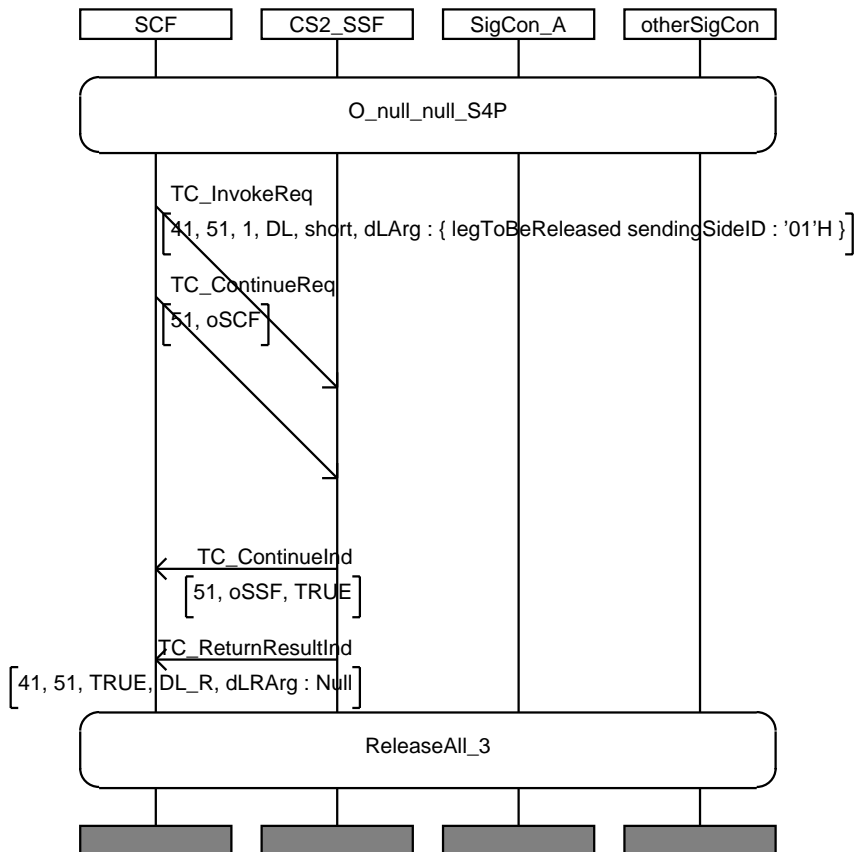
# MSC IN2\_A\_CPH\_019





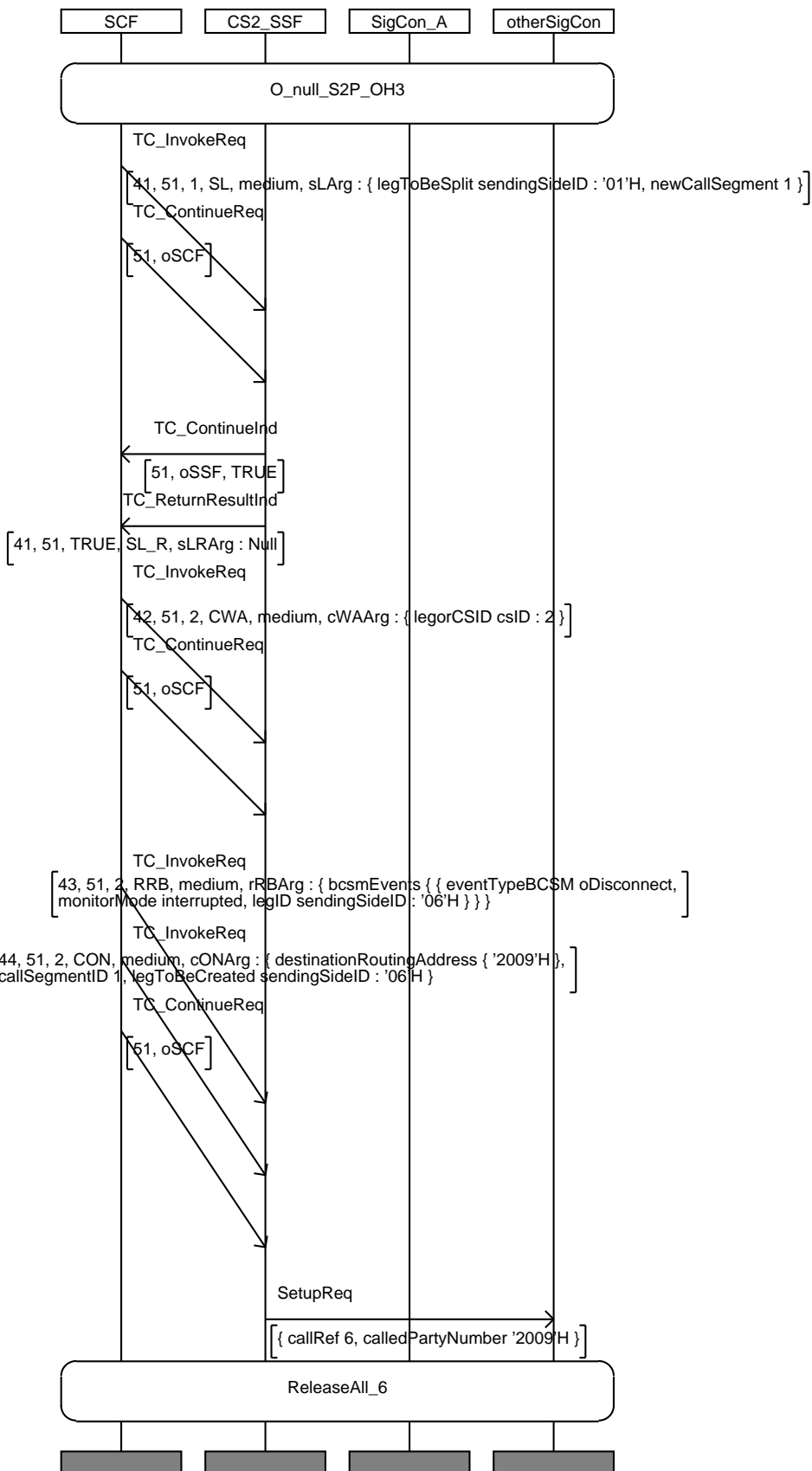
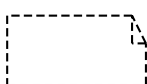
IN2_A_CPH_020	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P
<b>Test description</b>	L1! DisconnectLeg(1) L1!ContinueWithArgument(CsId=3) L1? DisconnectLegReturnResult Reaching state O_null_null_TF(3): (it could be reconnected afterwards)
<b>Pass criteria</b>	CP1_1? ReleaseReq
<b>Postamble:</b>	ReleaseAll_3

## MSC IN2\_A\_CPH\_020



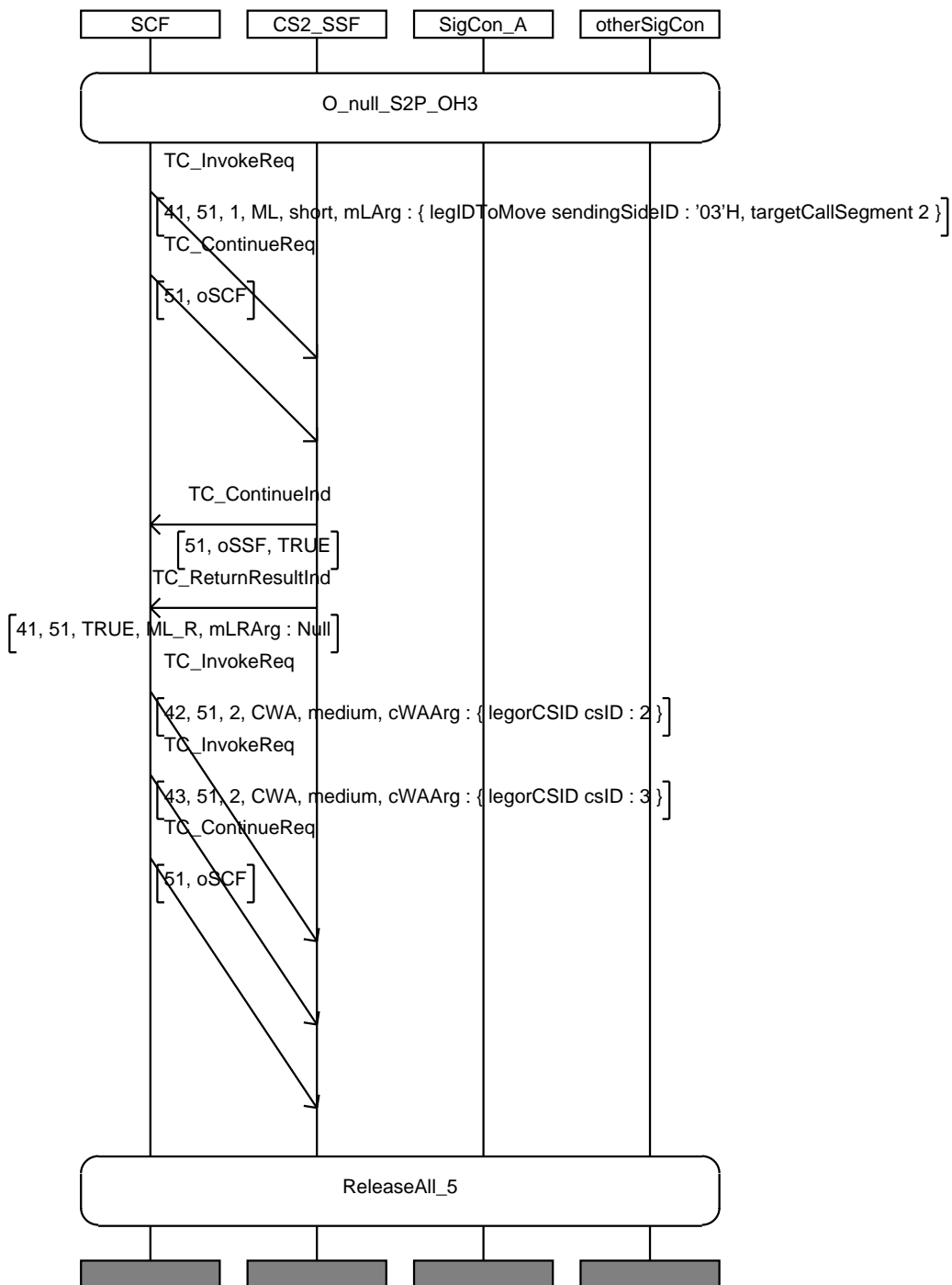
<b>IN2_A_CPH_021</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S2P_OH(3)
<b>Test description</b>	L1! SplitLeg(1,1)  L1? SplitLegReturnResult L1! ContinueWithArgument(CSIId=2) L1! RequestReportBCSMEvent(6,oDisconnect)  L1! Connect(6,1)  Reaching state O_S2P_OH(1)_OH(3)
<b>Pass criteria</b>	CP1-6? SetUpReq
<b>Postamble:</b>	ReleaseAll_6

# MSC IN2\_A\_CPH\_021



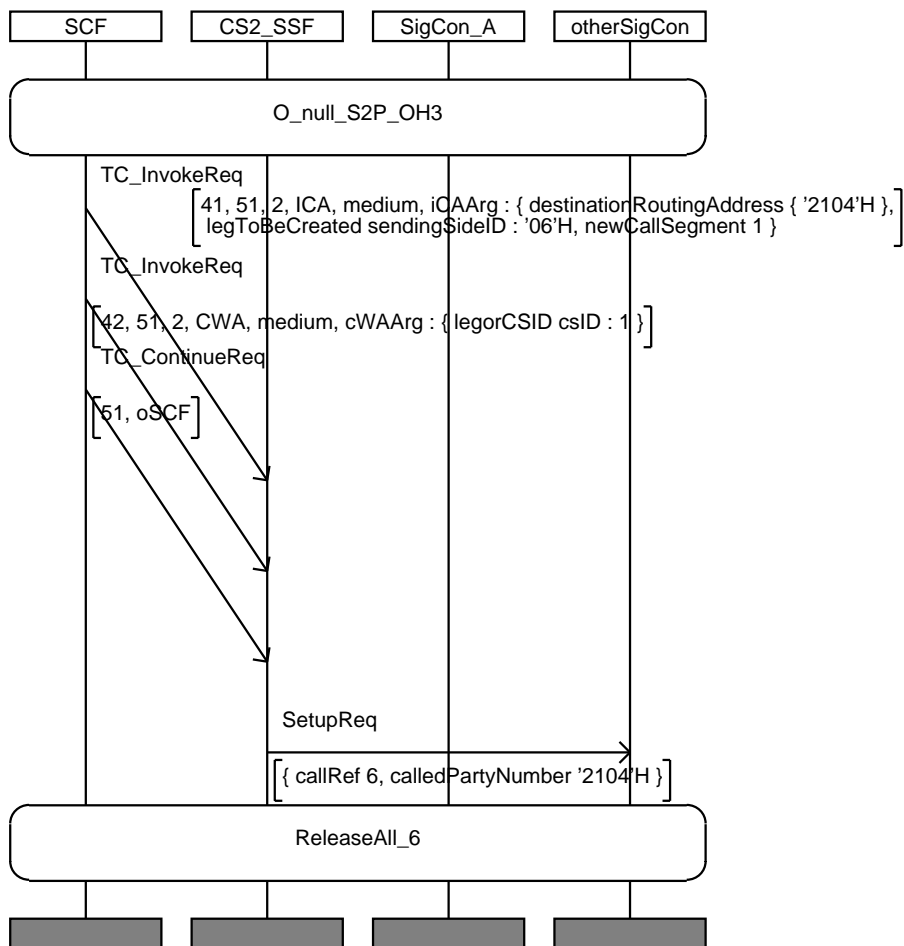
IN2_A_CPH_022	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S2P_OH(3)
<b>Test description</b>	L1! MoveLeg(3,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state O_null_S3P_OH(2)
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_5

# MSC IN2\_A\_CPH\_022

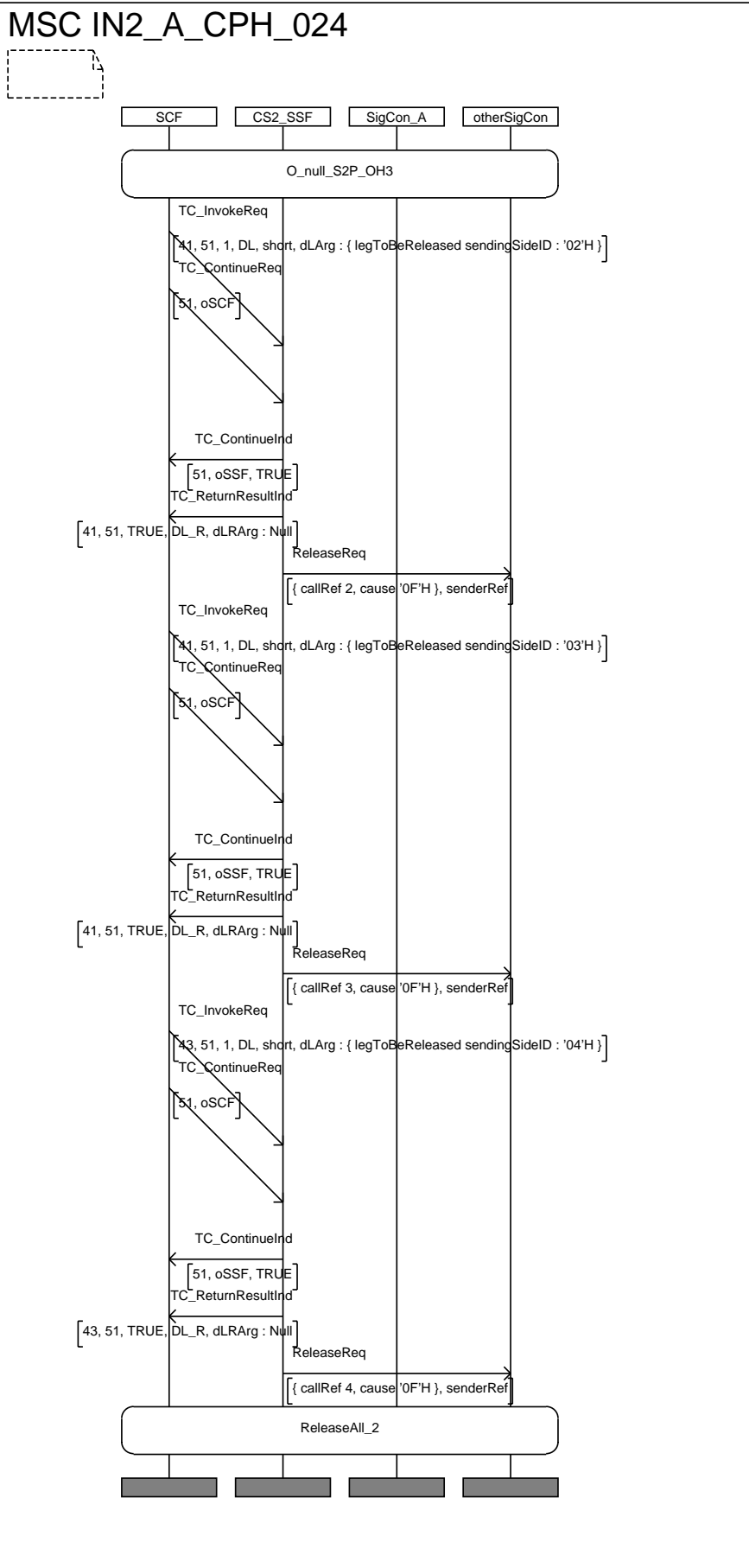


IN2_A_CPH_023	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S2P_OH(3)
<b>Test description</b>	L1! InitiateCallAttempt(6,1) L1! ContinueWithArgument(LegId=6) Reaching state O_S1P_S2P_OH(3)
<b>Pass criteria</b>	CP1_6? SetupReq
<b>Postamble:</b>	ReleaseAll_6

## MSC IN2\_A\_CPH\_023



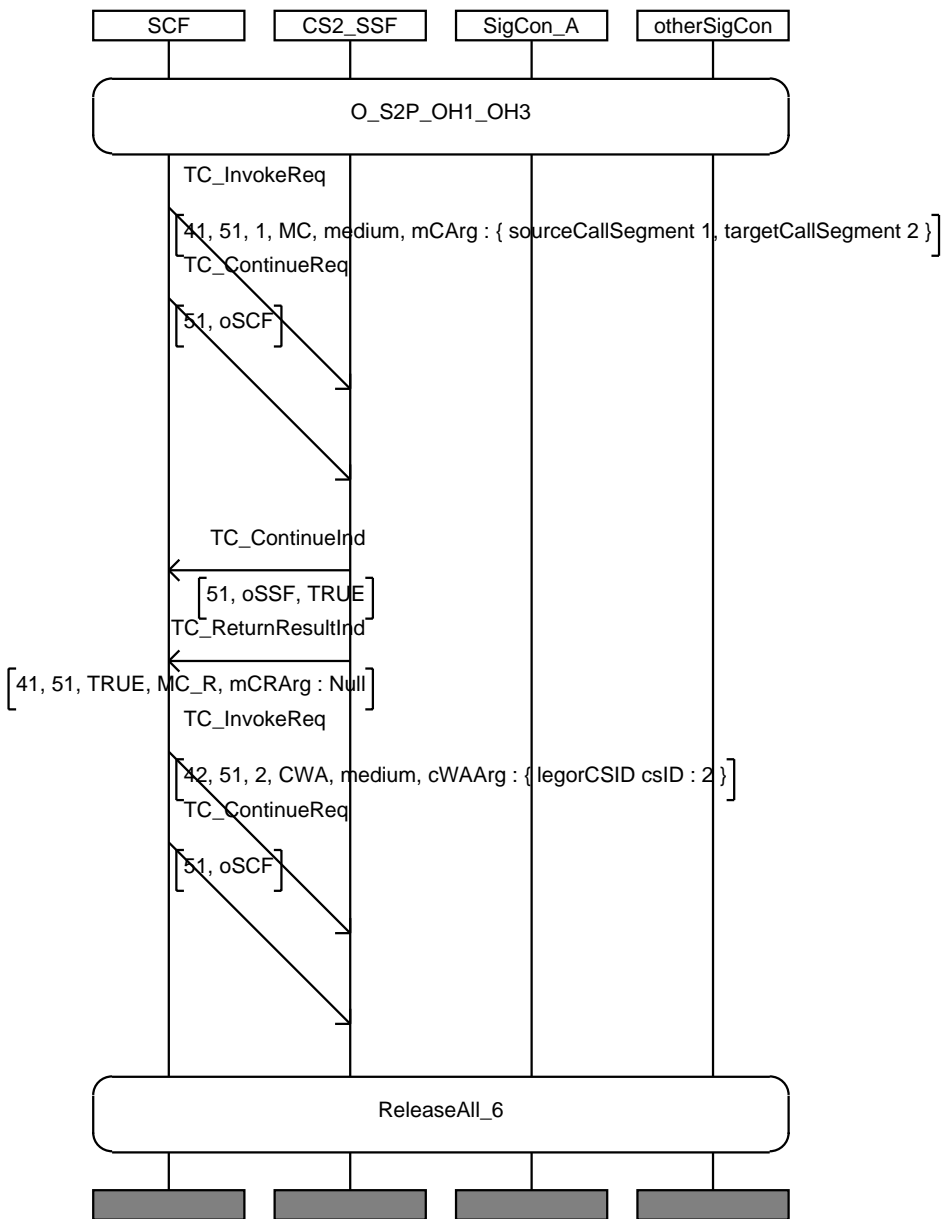
<b>IN2_A_CPH_024</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S2P_OH(3)
<b>Test description</b>	L1! DisconnectLeg(2) L1! DisconnectLeg(3) L1! DisconnectLeg(4) L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult Reaching state O_null_S2P_null
<b>Pass criteria</b>	CP1_2? ReleaseReq CP1_3? ReleaseReq CP1_4? ReleaseReq
<b>Postamble:</b>	ReleaseAll_2



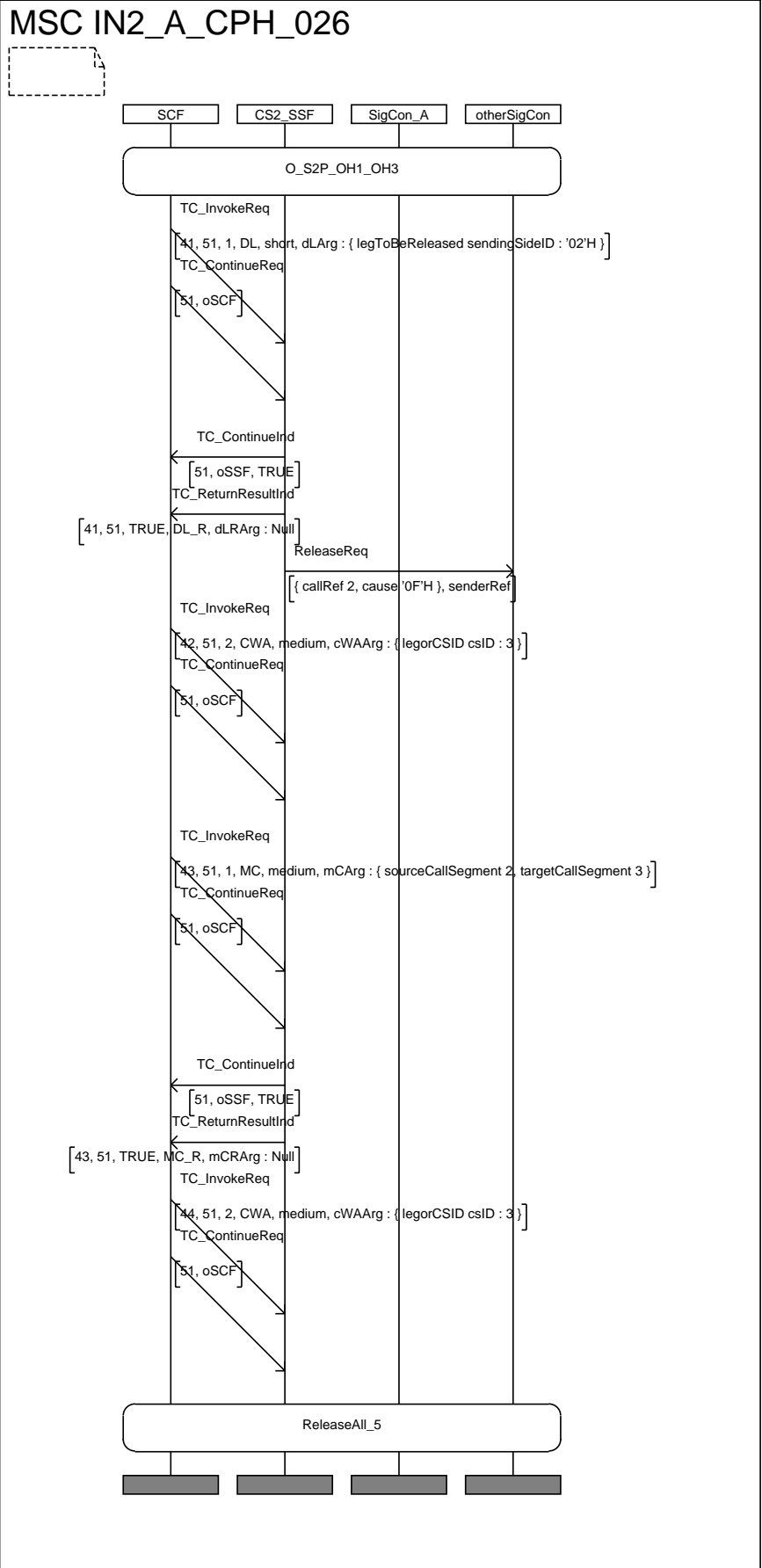


IN2_A_CPH_025	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(1)_OH(3)
<b>Test description</b>	L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_S3P_OH(3)
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_6

# MSC IN2\_A\_CPH\_025

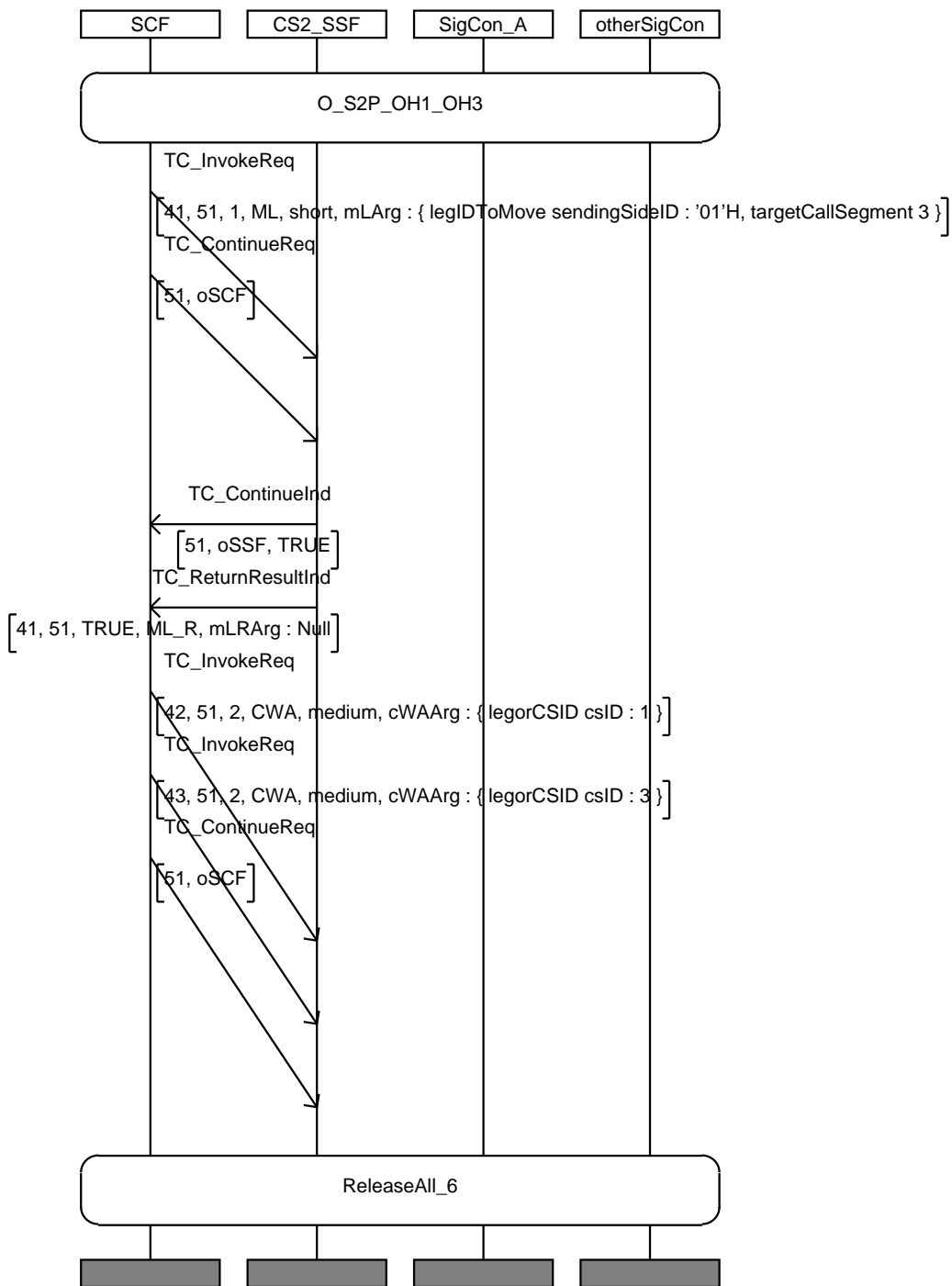


<b>IN2_A_CPH_026</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(1)_OH(3)
<b>Test description</b>	L1! DisconnectLeg(2) L1! ContinueWithArgument(CsId=3) L1? DisconnectLegReturnResult CP1-2? ReleaseReq L1! MergeCallSegment(2,3) L1? MergeCallSegmentReturnResult L1! ContinueWithArgument(CsId=3) Reaching state O_S2P_null_OH(3)
<b>Pass criteria</b>	CP1_2? ReleaseReq
<b>Postamble:</b>	ReleaseAll_5



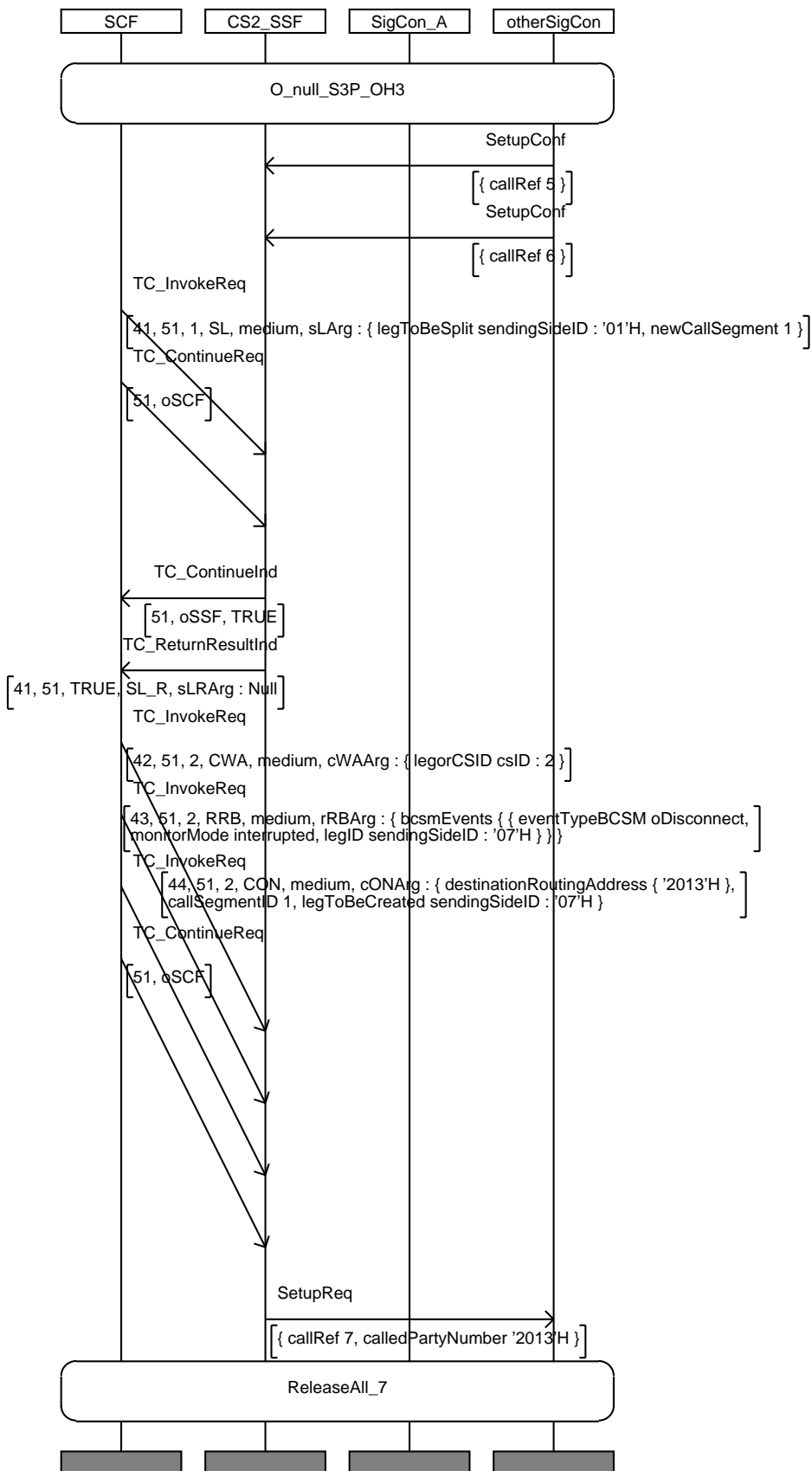
<b>IN2_A_CPH_027</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(1)_OH(3)
<b>Test description</b>	L1! MoveLeg(1,3) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=3) Reaching state O_OH(1)_OH(1)_S4P
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_6

# MSC IN2\_A\_CPH\_027



<b>IN2_A_CPH_028</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S3P_OH(3)
<b>Test description</b>	CP1-5! SetupCon, E party answers CP1-6! SetupConf, F party answers L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CSIId=2) L1! RequestReportBCSMEvent(7,oDisconnect) L1! Connect(7,1) Reaching state O_S2P_OH(2)_OH(3)
<b>Pass criteria</b>	CP1-7? SetUpReq
<b>Postamble:</b>	ReleaseAll_7

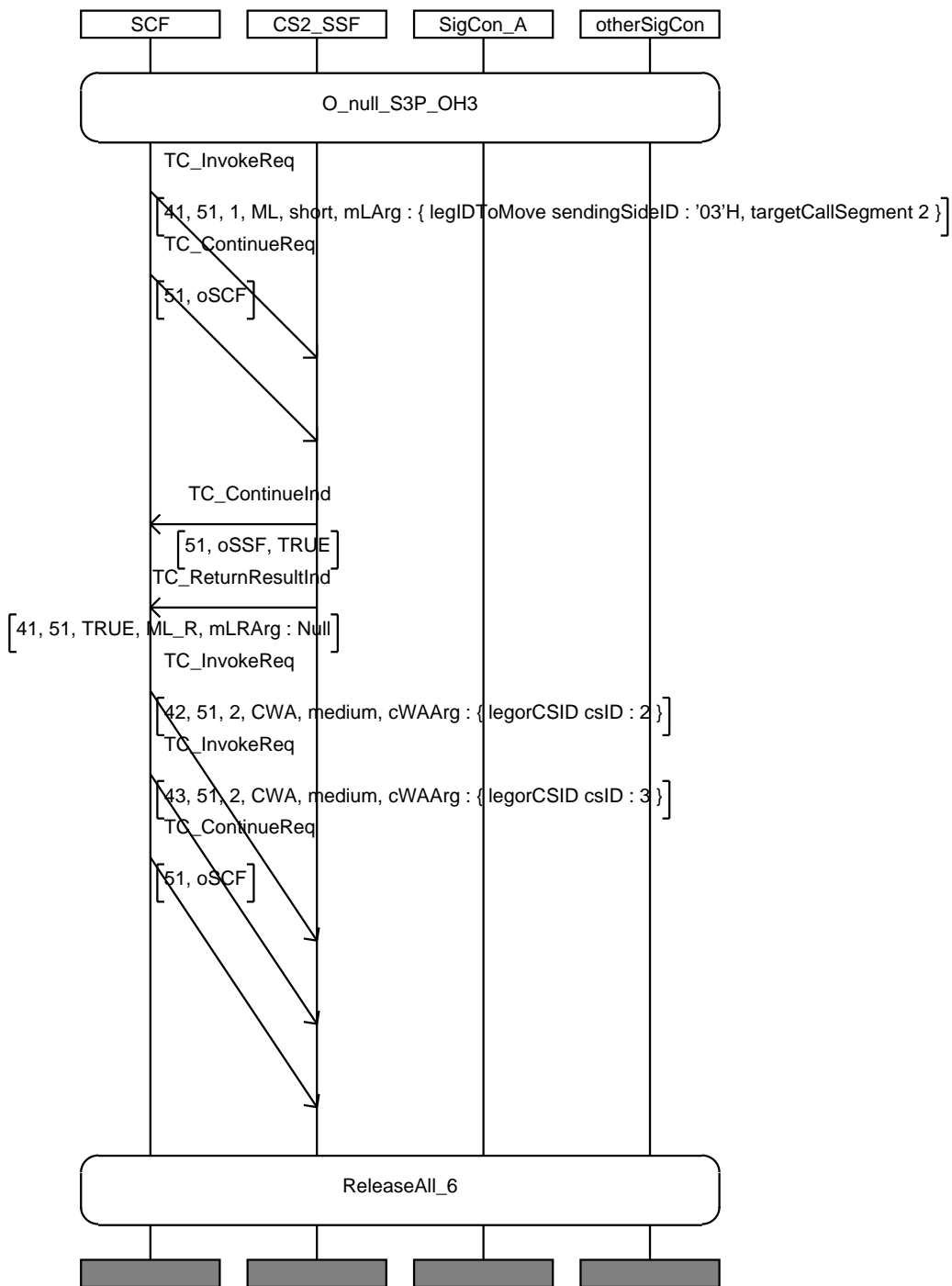
# MSC IN2\_A\_CPH\_028





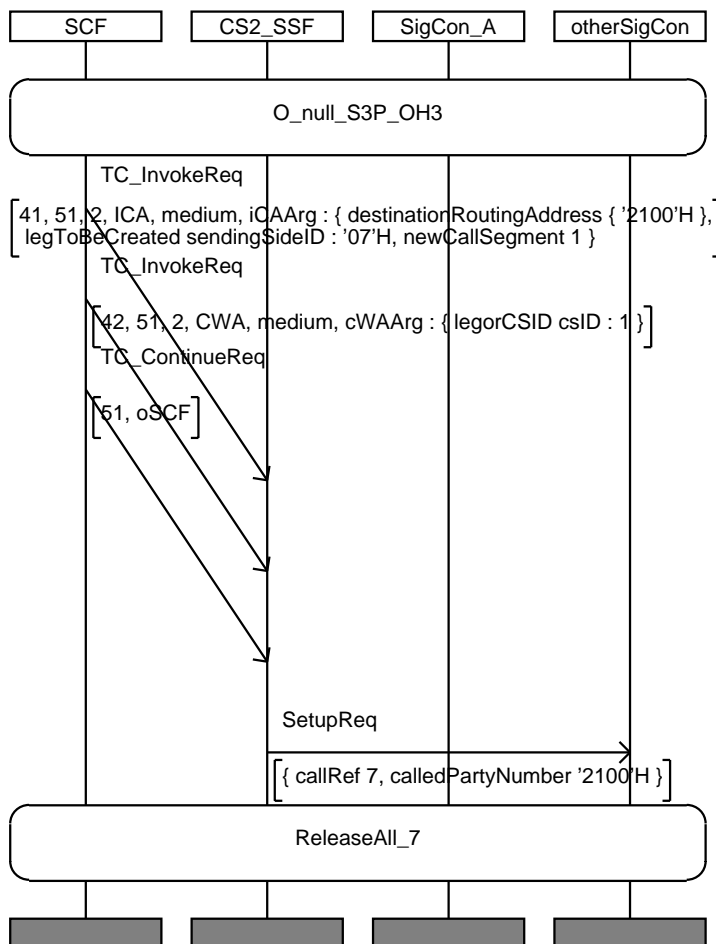
IN2_A_CPH_029	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S3P_OH(3)
<b>Test description</b>	L1! MoveLeg(3,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state O_null_S4P_OH(2)
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_6

# MSC IN2\_A\_CPH\_029

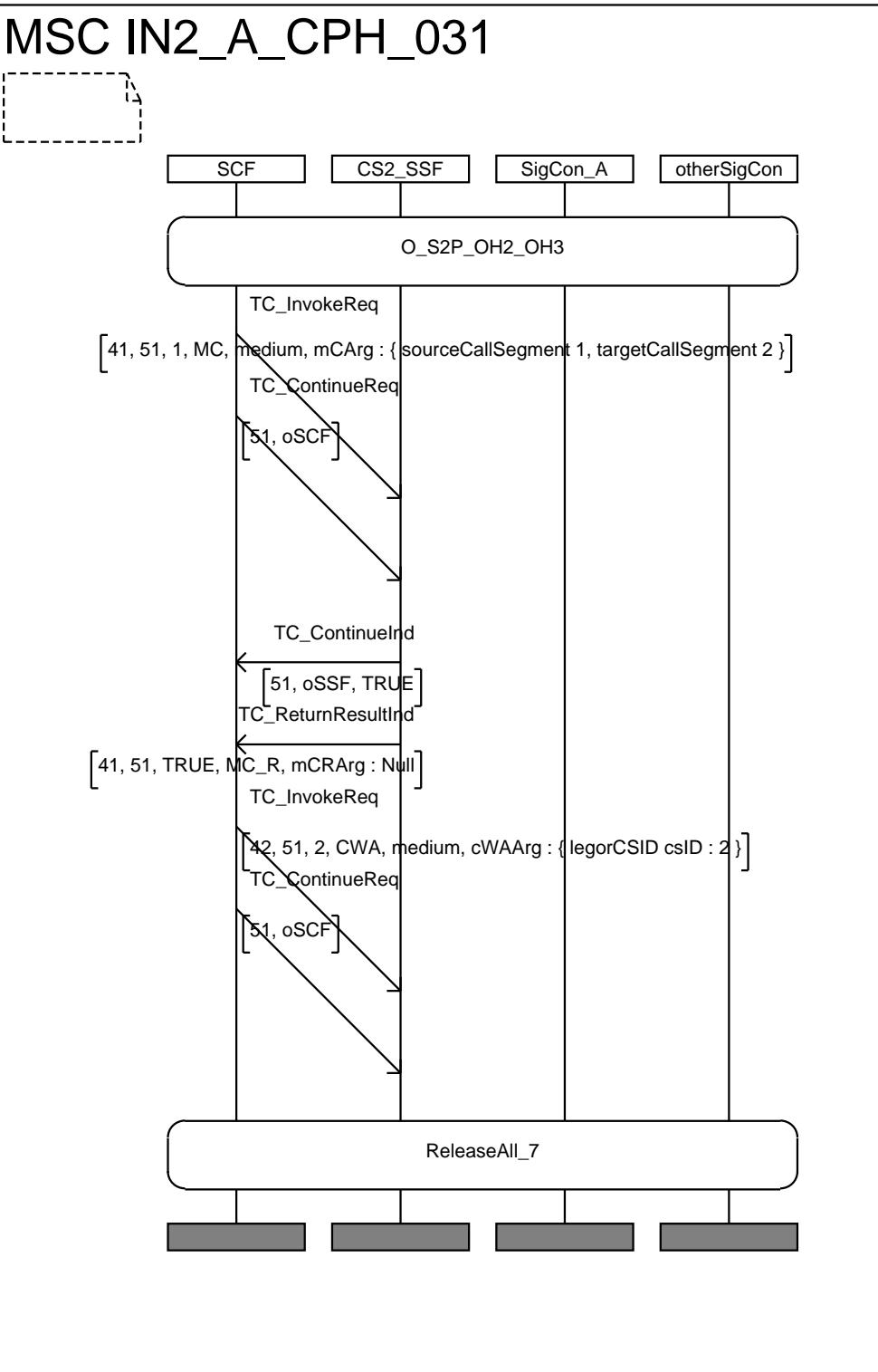


IN2_A_CPH_030	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S3P_OH(3)
<b>Test description</b>	L1! InitiateCallAttempt(7,1) L1! ContinueWithArgument(7) Reaching state O_S1P_S3P_OH(3)
<b>Pass criteria</b>	CP1_7? SetUpReq
<b>Postamble:</b>	ReleaseAll_7

## MSC IN2\_A\_CPH\_030

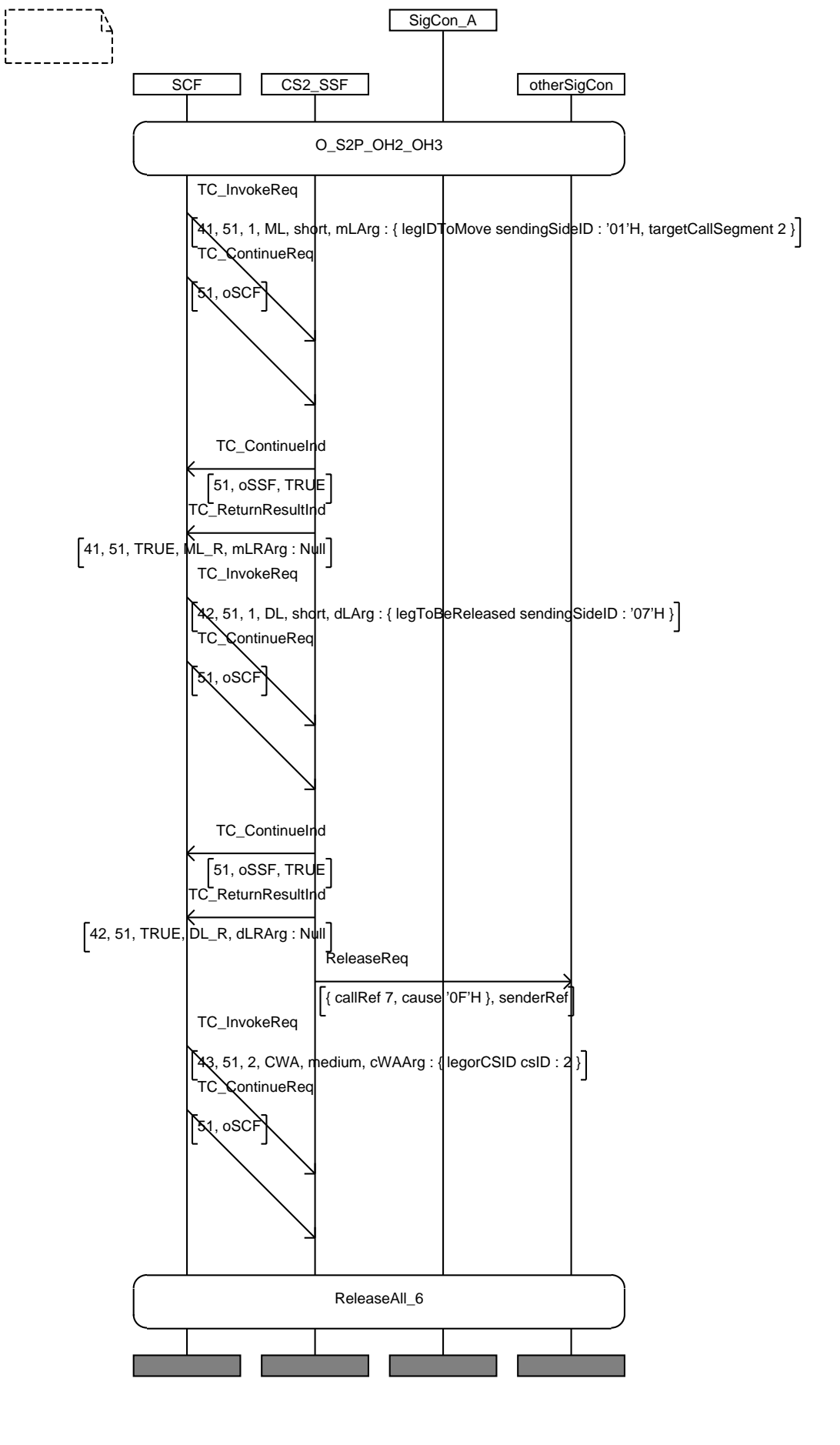


IN2_A_CPH_031	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(2)_OH(3)
<b>Test description</b>	L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_S4P_OH(3)
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_7



<b>IN2_A_CPH_032</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(2)_OH(3)
<b>Test description</b>	L1! MoveLeg(1,2) L1? MoveLegReturnResult L1! DisconnectLeg(7) L1! ContinueWithArgument(CsId=2) L1? DisconnectLegReturnResult Reaching state O_null_S3P_OH(3)
<b>Pass criteria</b>	CP1_7? ReleaseReq
<b>Postamble:</b>	ReleaseAll_6

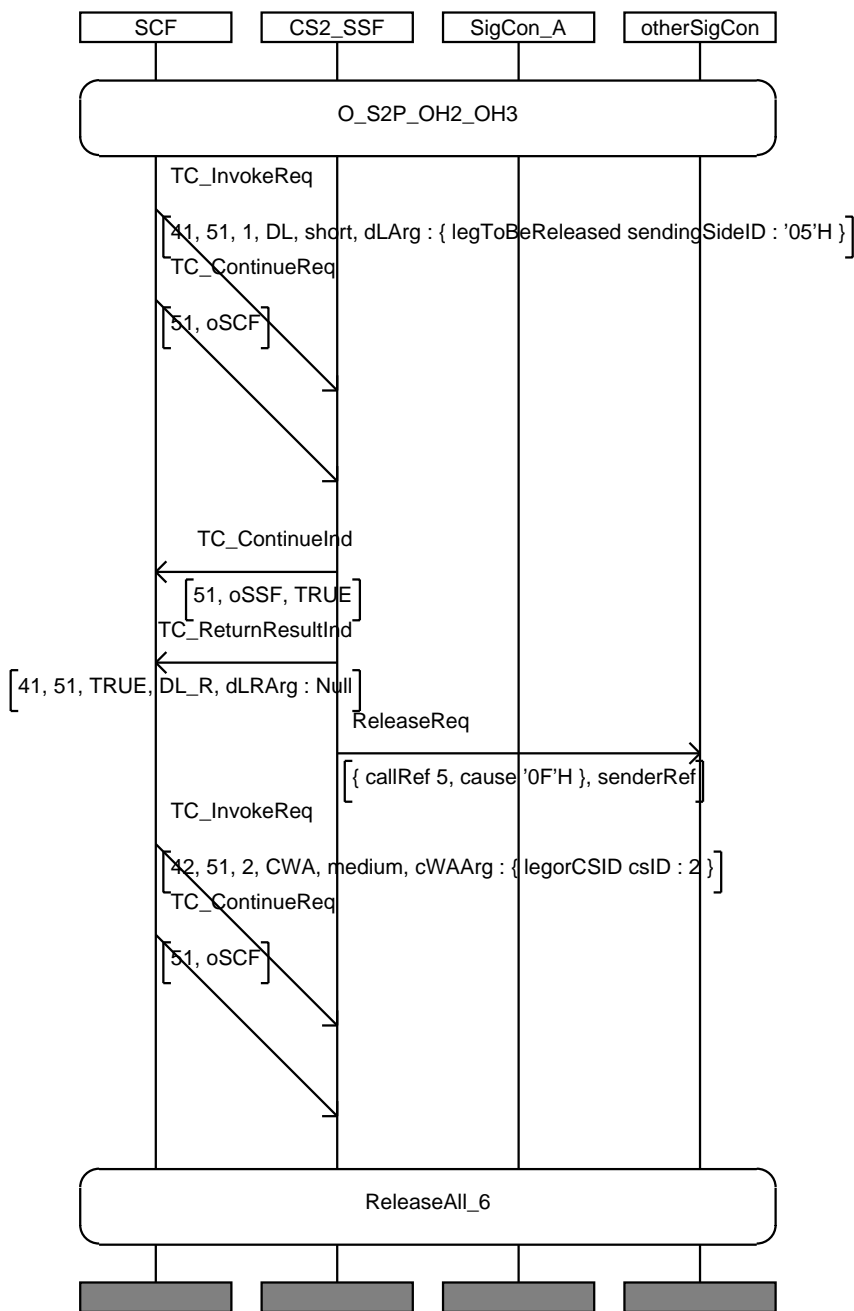
# MSC IN2\_A\_CPH\_032



<b>IN2_A_CPH_033</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(2)_OH(3)
<b>Test description</b>	L1! DisconnectLeg(5) L1! ContinueWithArgument(CsId=2) L1? DisconnectLegReturnResult Reaching state O_S2P_OH(1)_OH(3)
<b>Pass criteria</b>	CP1_5? ReleaseReq
<b>Postamble:</b>	ReleaseAll_6

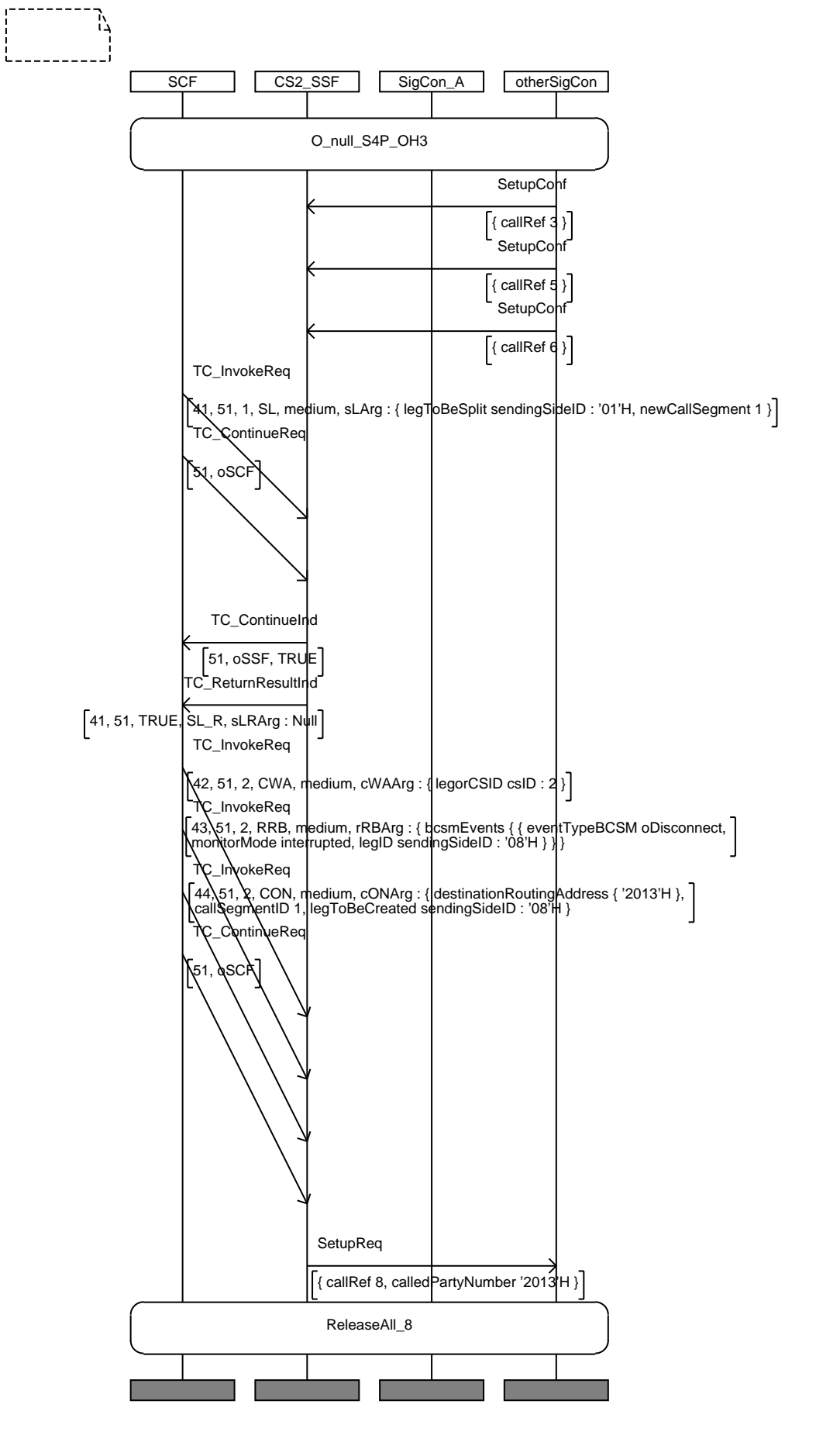


# MSC IN2\_A\_CPH\_033

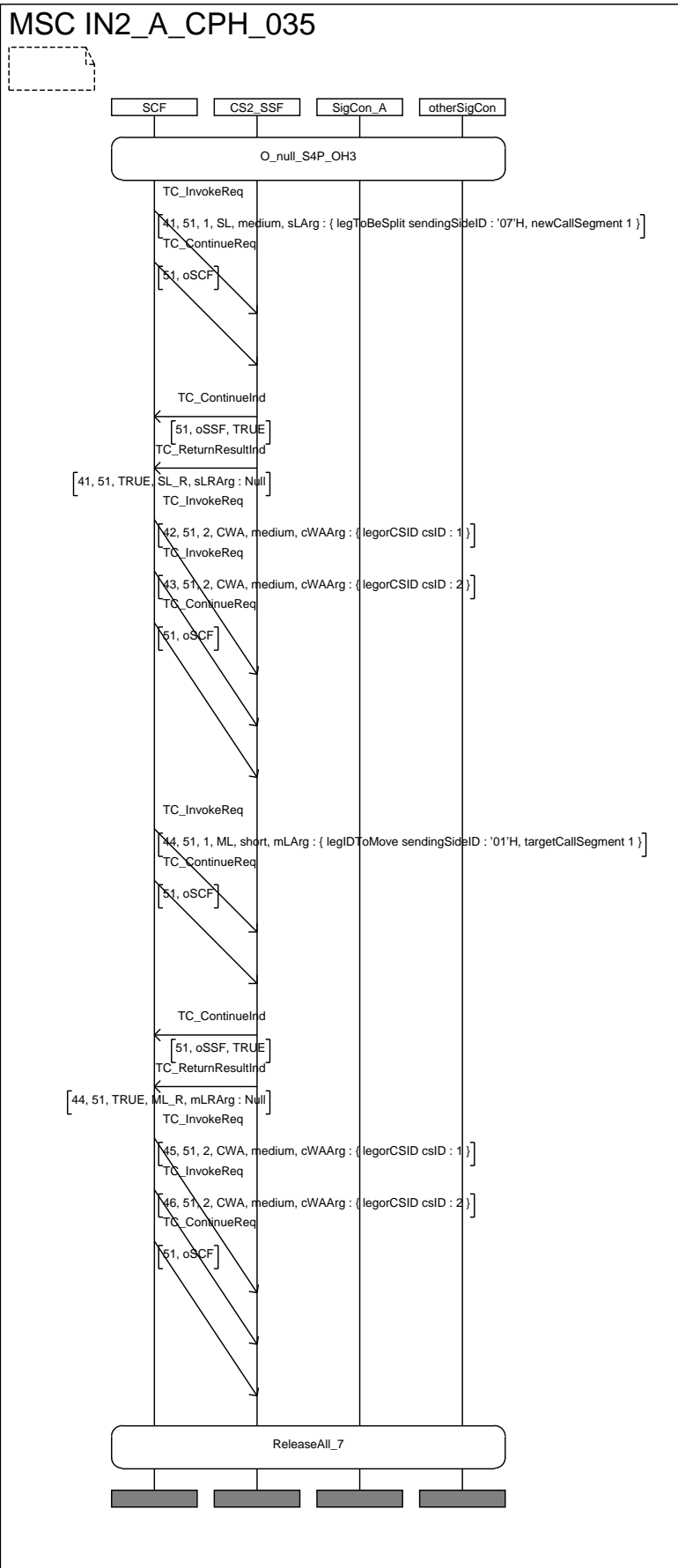


<b>IN2_A_CPH_034</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S4P_OH(3)
<b>Test description</b>	CP1-3! SetupConf, C party answers CP1-5! SetupConf, E party answers CP1-6! SetupConf, F party answers L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! ContinueWithAgument(CsId=2) L1! RequestReportBCSMEvent(8,oDisconnect) L1! Connect(8,1) Reaching state O-S2P_OH(3)_OH(3)
<b>Pass criteria</b>	CP1-8? SetUpReq
<b>Postamble:</b>	ReleaseAll_8

# MSC IN2\_A\_CPH\_034

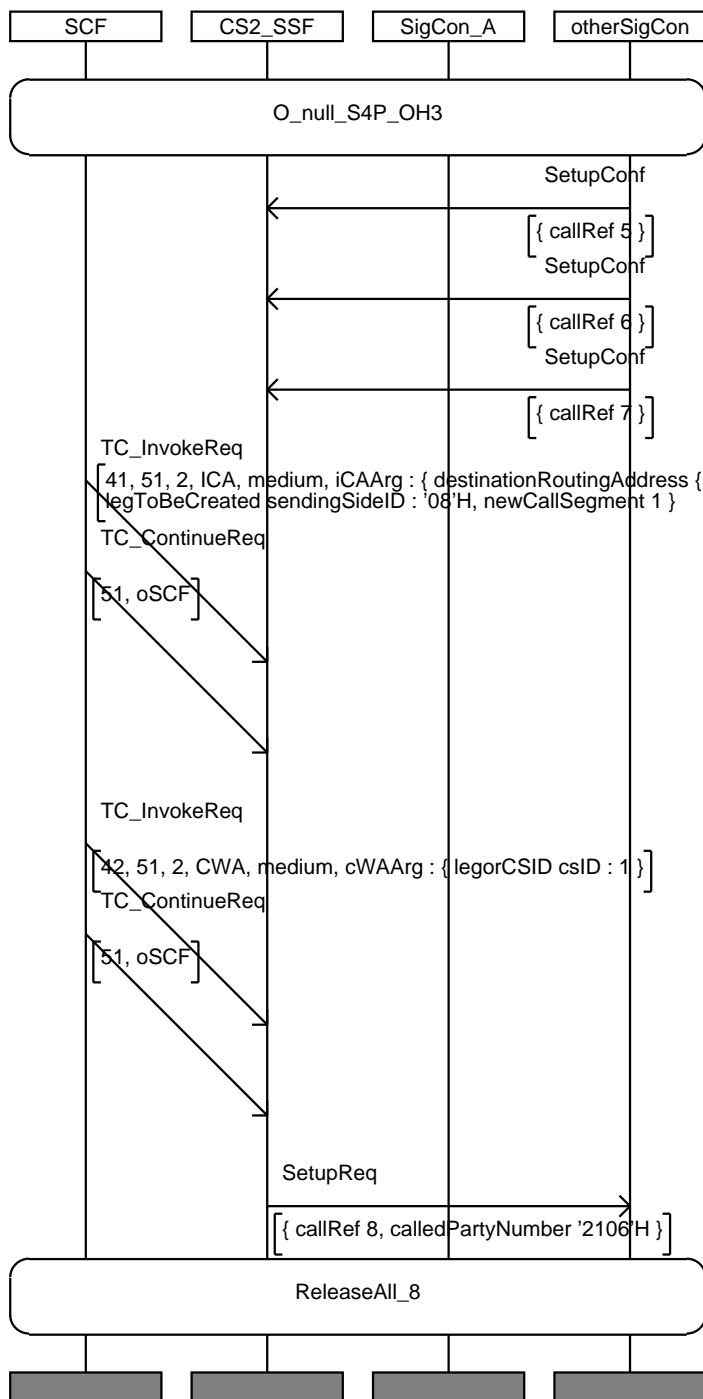


<b>IN2_A_CPH_035</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S4P_OH(3)
<b>Test description</b>	L1! SplitLeg(7,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=2) L1! MoveLeg(1,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=2) Reaching state O_S2P_OH(2)_OH(3)
<b>Pass criteria</b>	SSF sends SplitLegReturnResult SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_7



<b>IN2_A_CPH_036</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S4P_OH(3)
<b>Test description</b>	CP1-3! SetupConf, C party answers CP1-5! SetupConf, E party answers CP1-6! SetupConf, F party answers L1! InitiateCallAttempt(8,1) L1! ContinueWithArgument(8) Reaching state O_S1P_S4P_OH(3)
<b>Pass criteria</b>	CP1_8? SetUpReq
<b>Postamble:</b>	ReleaseAll_8

# MSC IN2\_A\_CPH\_036



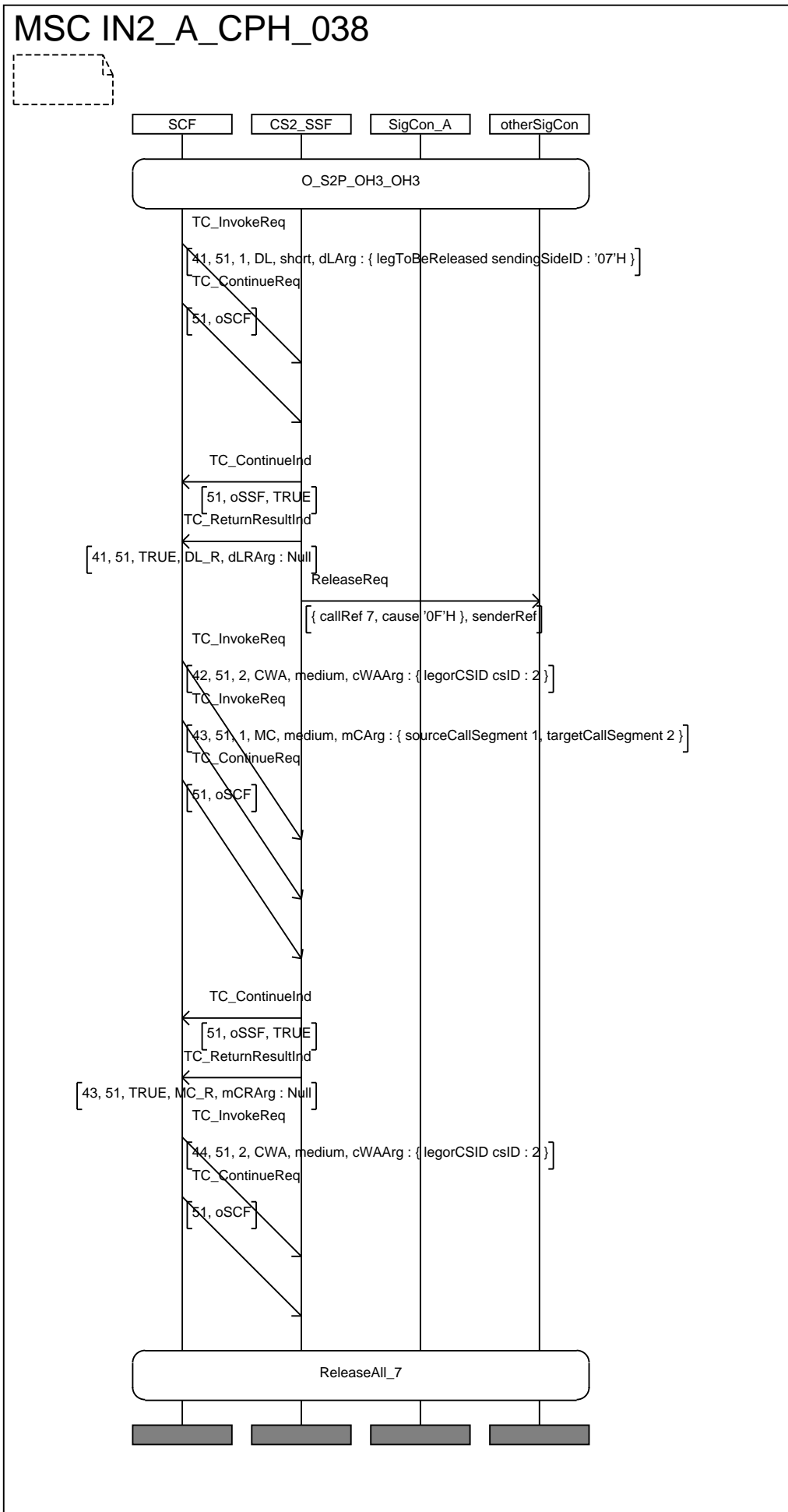
<b>IN2_A_CPH_037</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_S4P_OH(3)
<b>Test description</b>	CP1-3! SetupConf, C party answers CP1-5! SetupConf, E party answers CP1-6! SetupConf, F party answers L1! DisconnectLeg(2) L1! DisconnectLeg(3) L1! DisconnectLeg(4) L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult Reaching state O_null_S4P_null
<b>Pass criteria</b>	CP1_2? ReleaseReq CP1_3? ReleaseReq CP1_4? ReleaseReq
<b>Postamble:</b>	ReleaseAll_4





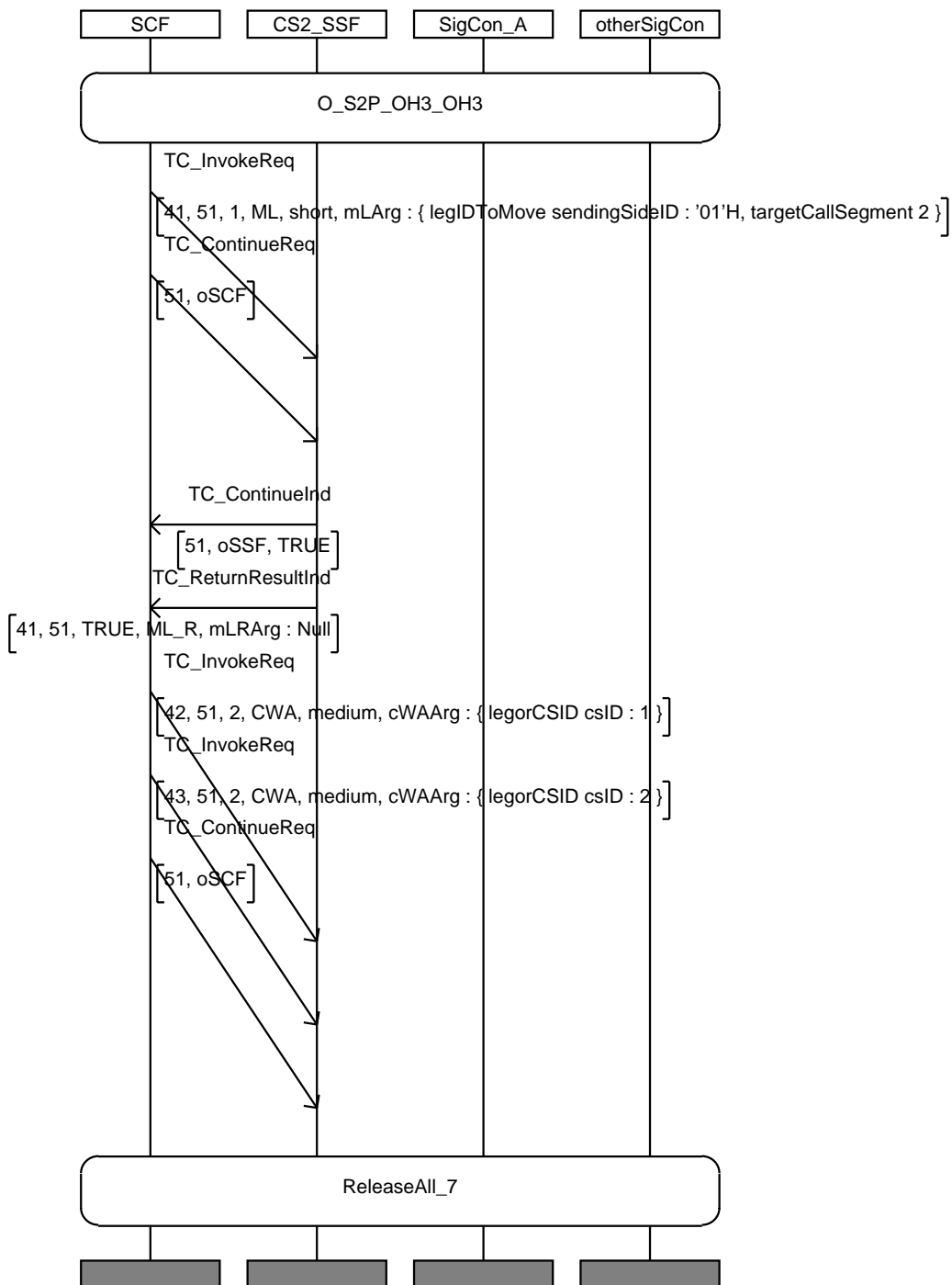
<b>IN2_A_CPH_038</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(3)_OH(3)
<b>Test description</b>	L1! DisconnectLeg(7) L1! ContinueWithArgument(CsId=2) L1! MergeCallSegment(1,2) L1? MergeCallSegmentReturnResult L1! ContinueWithArgument(CsId=2) L1? DisconnectLegReturnResult Reaching O_null_S4P_OH(3)
<b>Pass criteria</b>	CP1_7? ReleaseReq
<b>Postamble:</b>	ReleaseAll_7

# MSC IN2\_A\_CPH\_038



IN2_A_CPH_039	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_OH(3)_OH(3)
<b>Test description</b>	L1! MoveLeg(1,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=1) Reaching state O_OH(1)_S4P_OH(3)
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_7

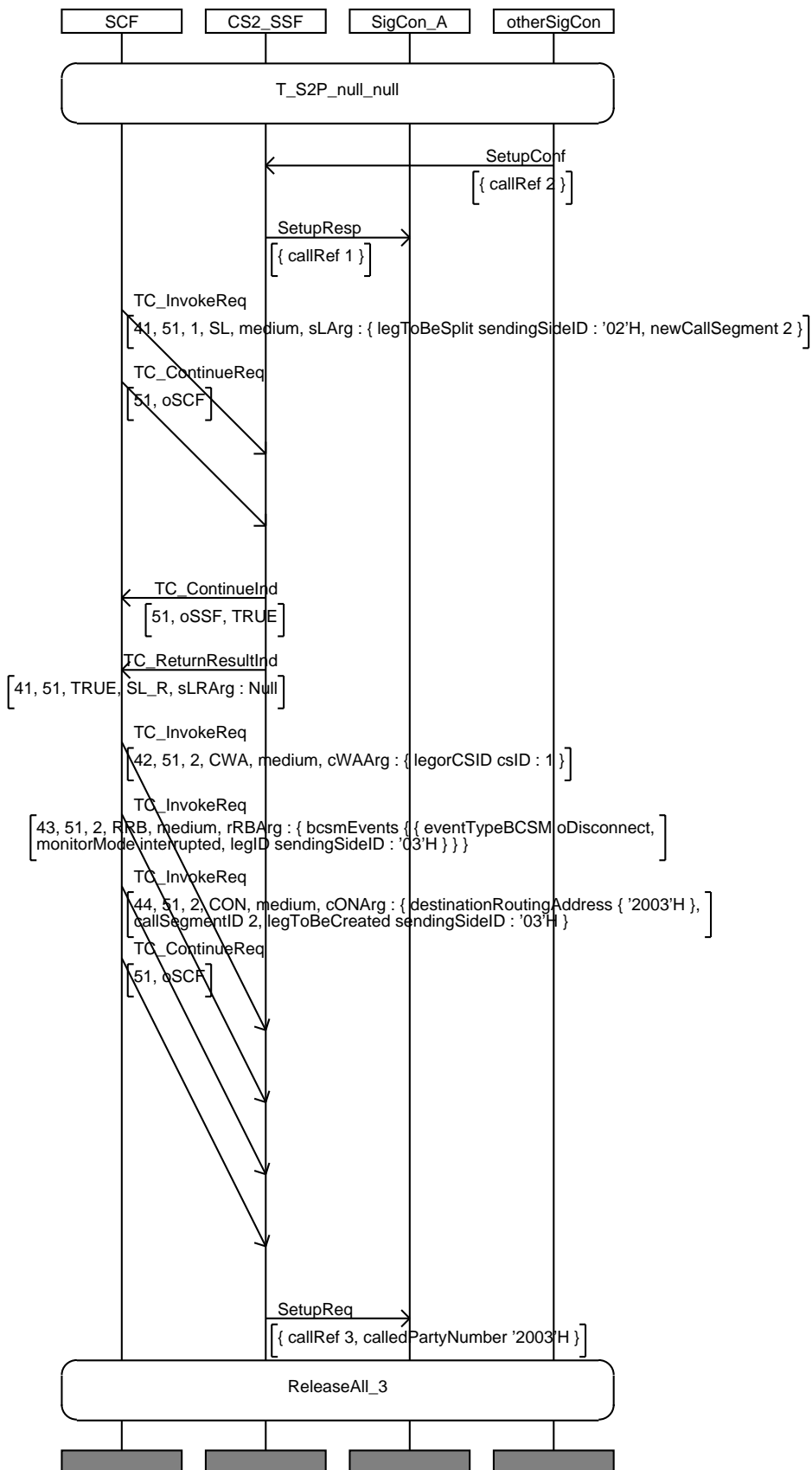
# MSC IN2\_A\_CPH\_039



## 7.5.2 Terminating (T\_BCSM) trigger (controlling legId = 2)

<b>IN2_A_CPH_040</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null
<b>Test description</b>	CP1-2! SetupConf L1! SplitLeg(2,2)  L1? SplitLegReturnResult  L1! ContinueWithArgument(CSId=1) L1! RequestReportBCSMEvent(3,oDisconnect)  L1! Connect(3,2)  Reaching state T_OH(1)_S2P_null
<b>Pass criteria</b>	CP1-3? SetUpReq
<b>Postamble:</b>	ReleaseAll_3

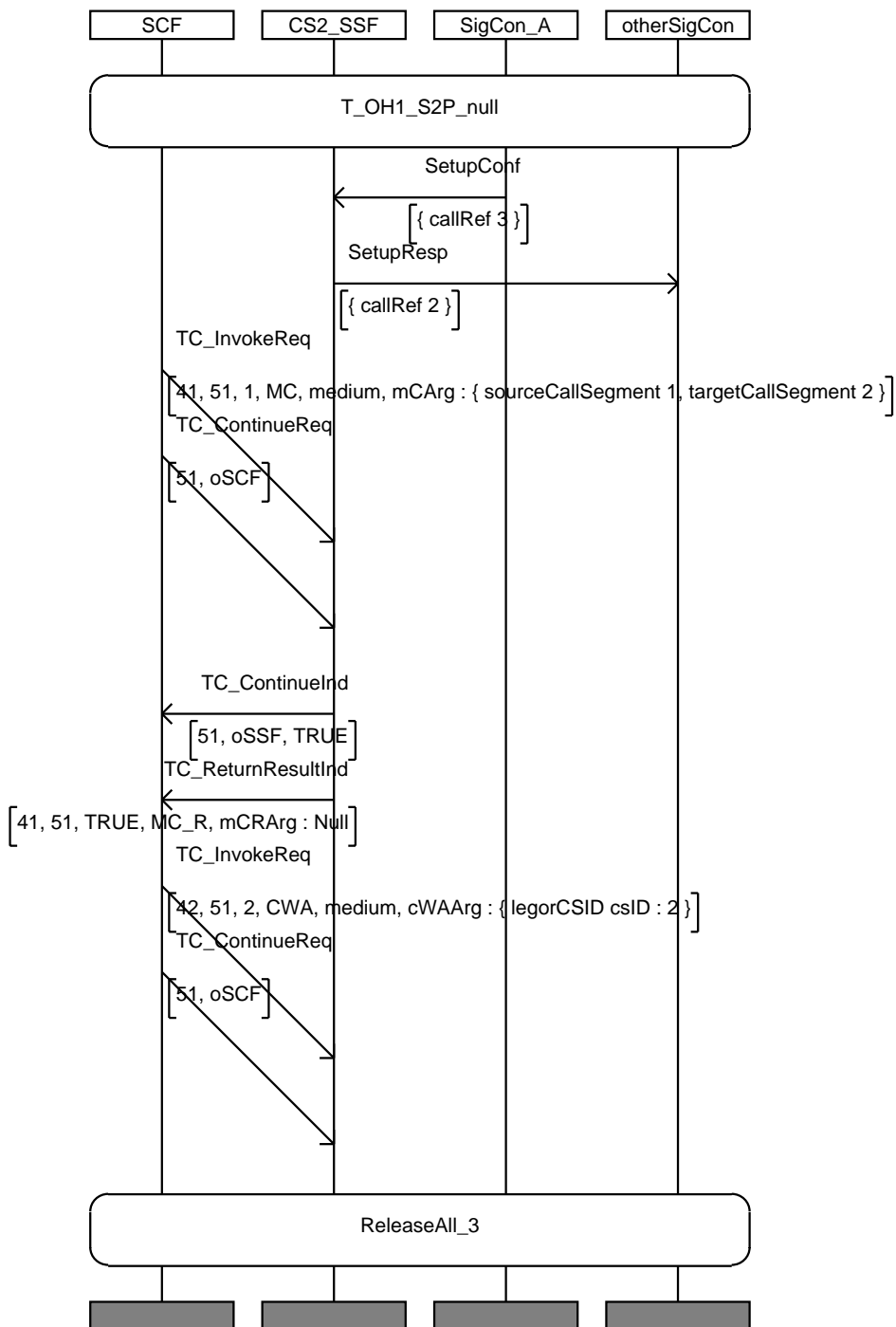
# MSC IN2\_A\_CPH\_040



IN2_A_CPH_041	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_OH(1)_S2P_null
<b>Test description</b>	CP1-3! SetupConf, C party answer L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2 Reaching state T_null_S3P_null
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_3

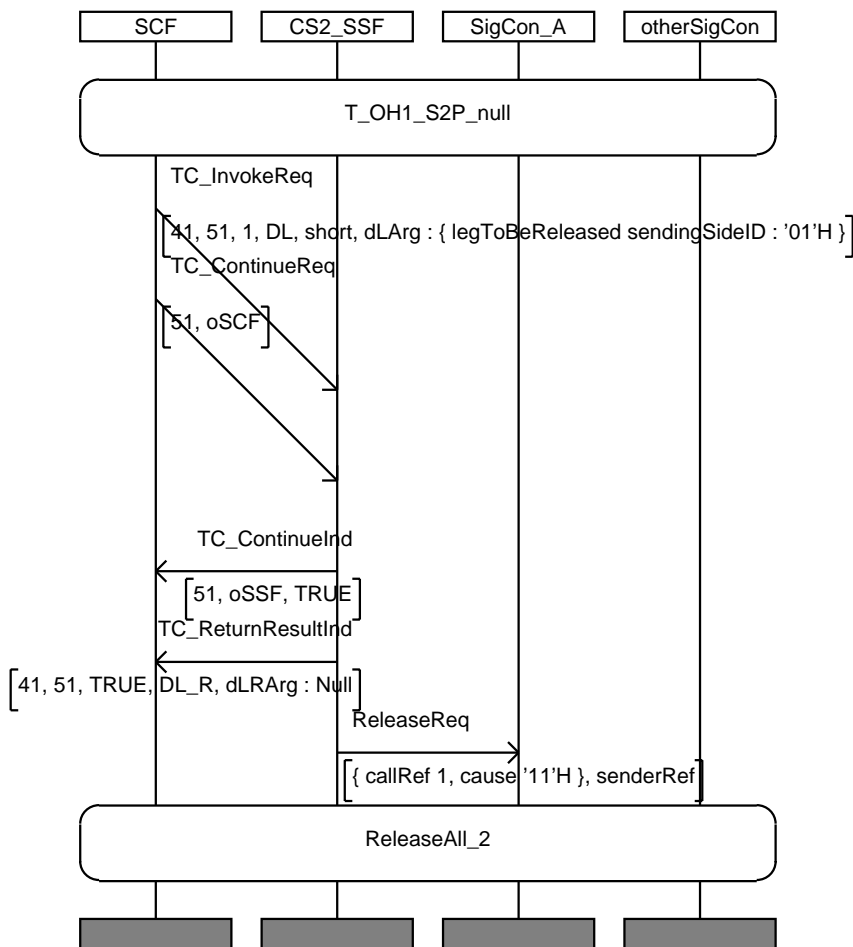


# MSC IN2\_A\_CPH\_041



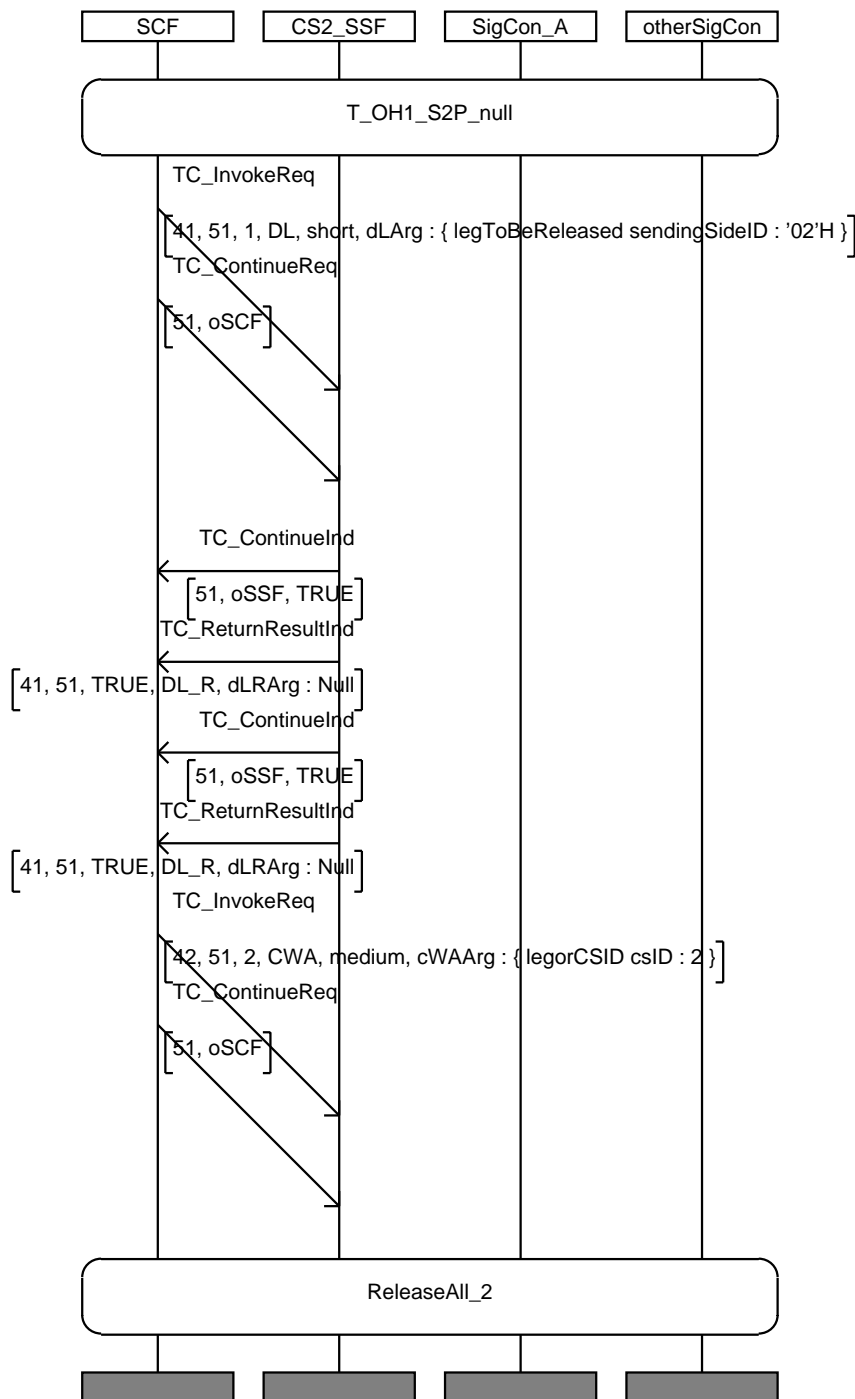
IN2_A_CPH_042	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_OH(1)_S2P_null
<b>Test description</b>	L1! DisconnectLeg(1) L1? DisconnectLegReturnResult Reaching state T_null_S2P_null
<b>Pass criteria</b>	CP1_1? ReleaseReq
<b>Postamble:</b>	ReleaseAll_2

# MSC IN2\_A\_CPH\_042



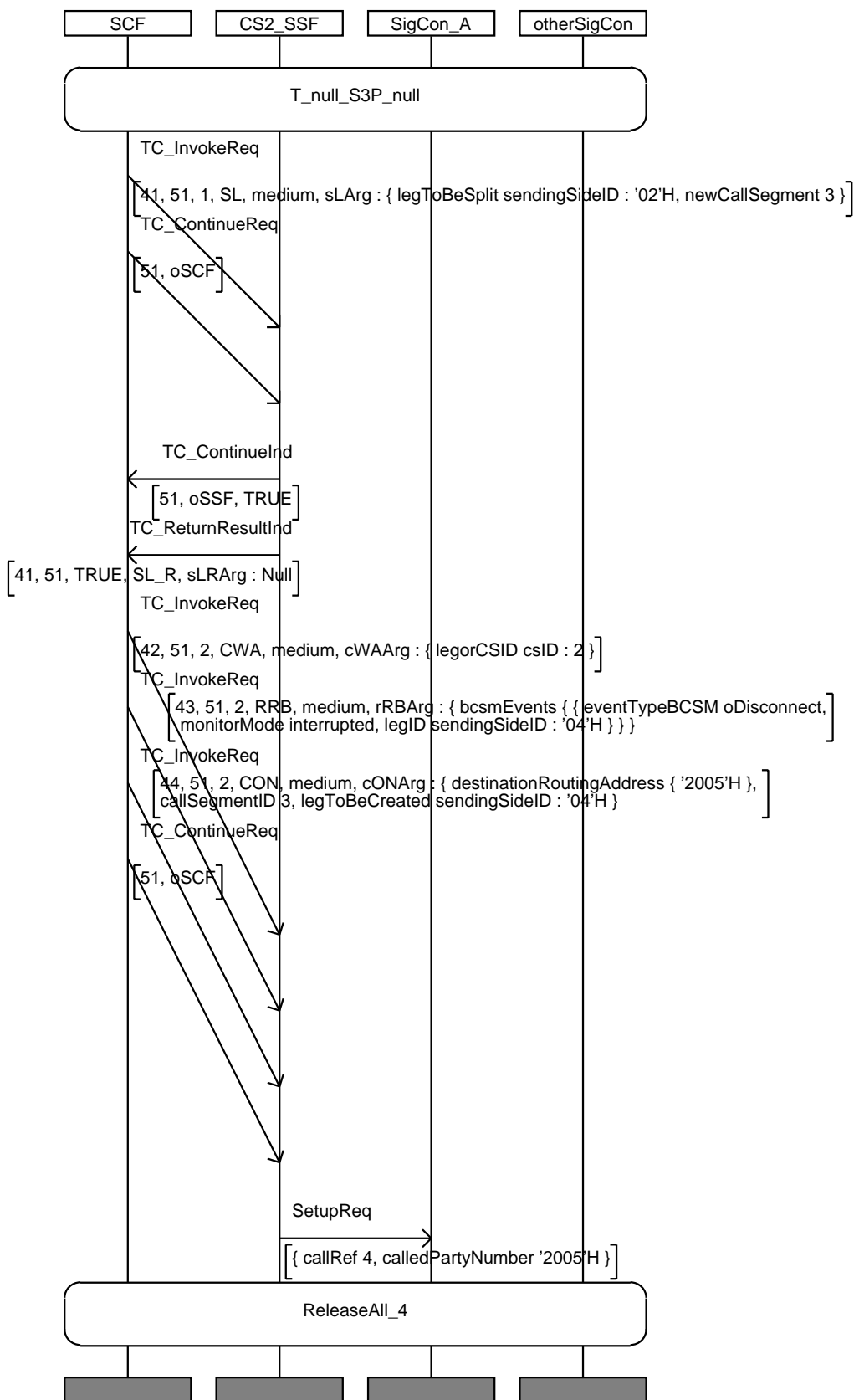
IN2_A_CPH_043	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_OH(1)_S2P_null
<b>Test description</b>	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult L1! ContinueWithArgument(CsId=2) Reaching state T_OH(1)_OH(1)_null
<b>Pass criteria</b>	CP1_2? ReleaseReq
<b>Postamble:</b>	ReleaseAll_2

# MSC IN2\_A\_CPH\_043



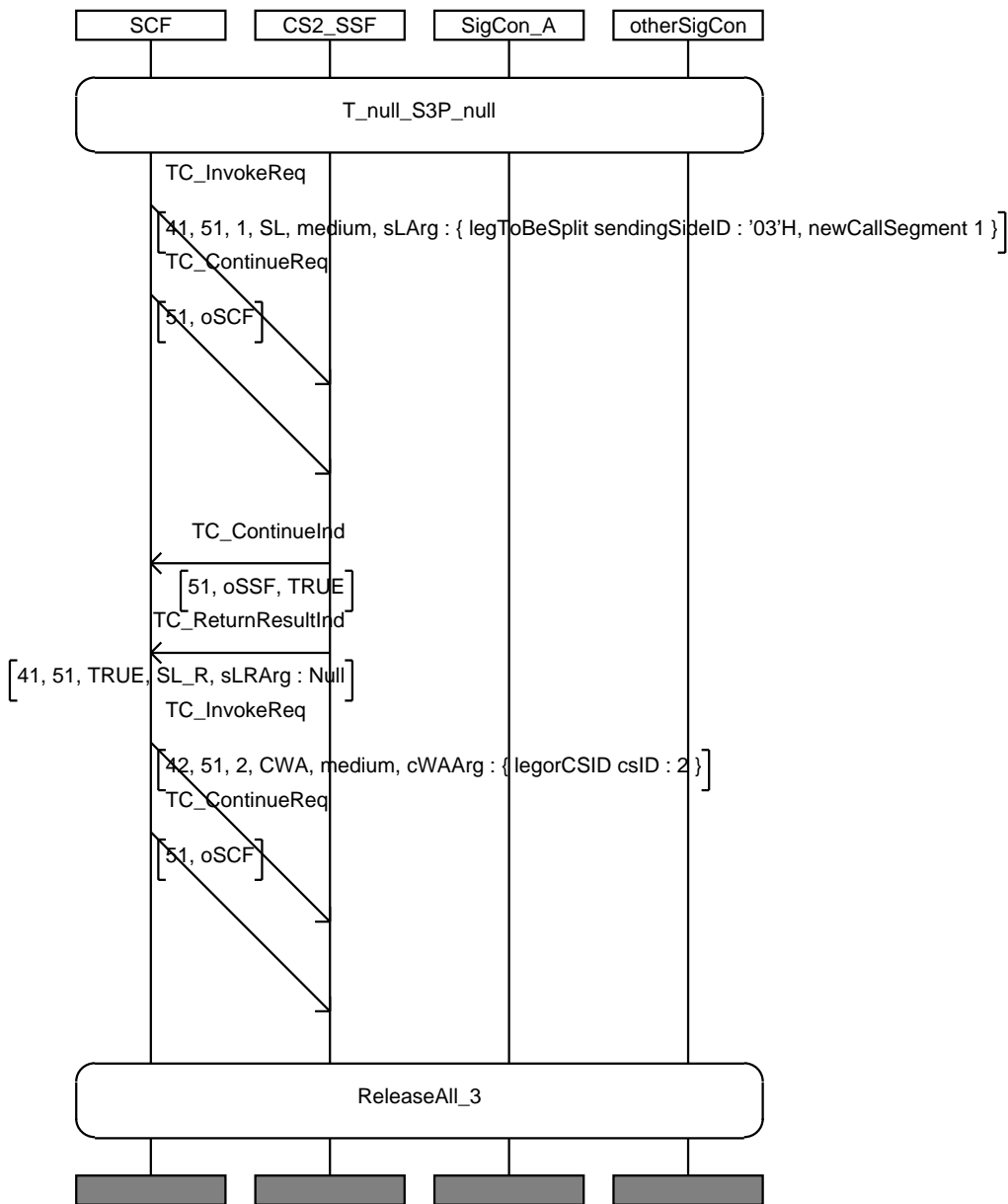
<b>IN2_A_CPH_044</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_S3P_null
<b>Test description</b>	L1! SplitLeg(2,3) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=2) L1! RequestReportBCSMEEvent(4,oDisconnect) L1! Connect(4,3) Reaching state T_null_OH(2)_S2P
<b>Pass criteria</b>	Pass criteria CP1-4? SetUpReq
<b>Postamble:</b>	ReleaseAll_4

# MSC IN2\_A\_CPH\_044



IN2_A_CPH_045	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_S3P_null
<b>Test description</b>	L1! SplitLeg(3,1) L1! ContinueWithArgument(CsId=2) Reaching state T_OH(1)_S2P_null
<b>Pass criteria</b>	SSF sends SplitLegReturnResult
<b>Postamble:</b>	ReleaseAll_3

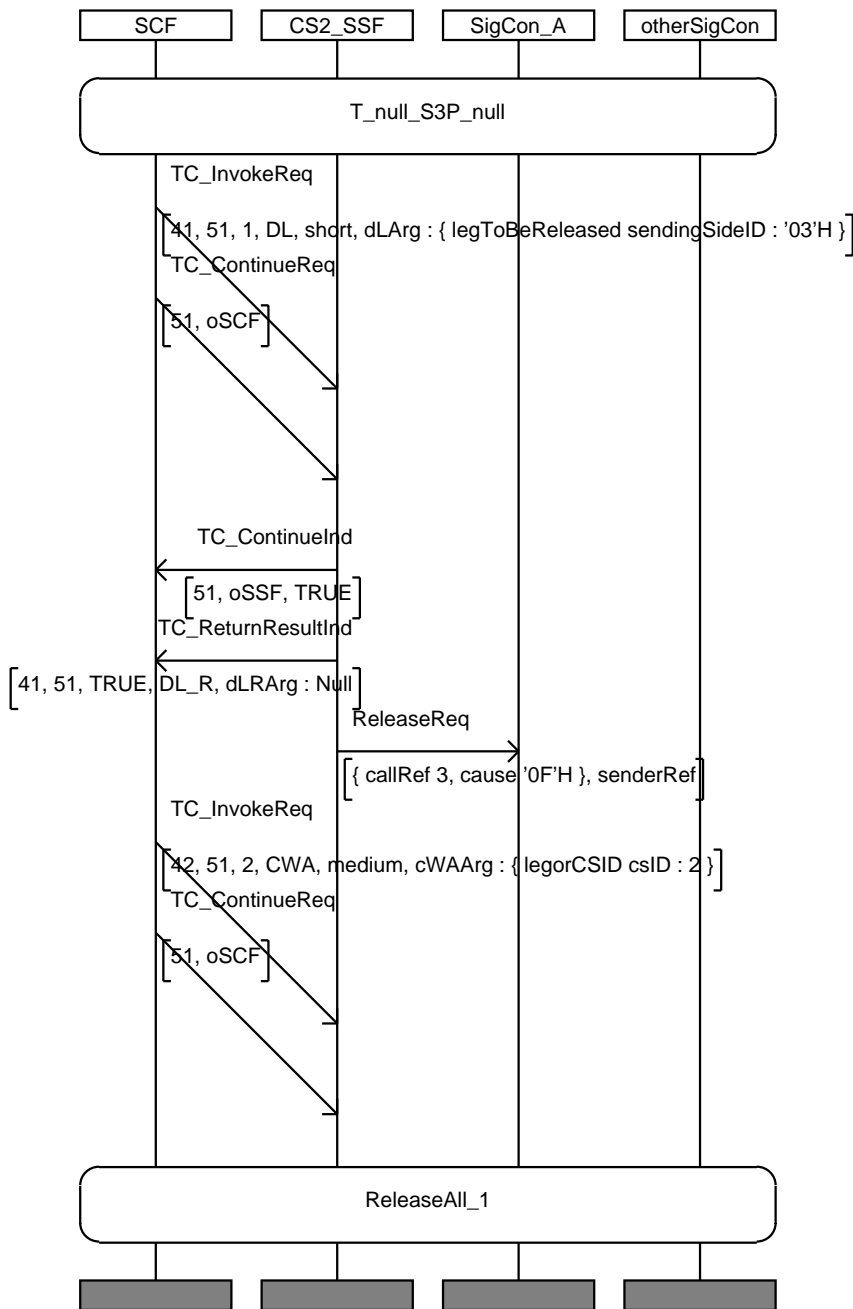
# MSC IN2\_A\_CPH\_045





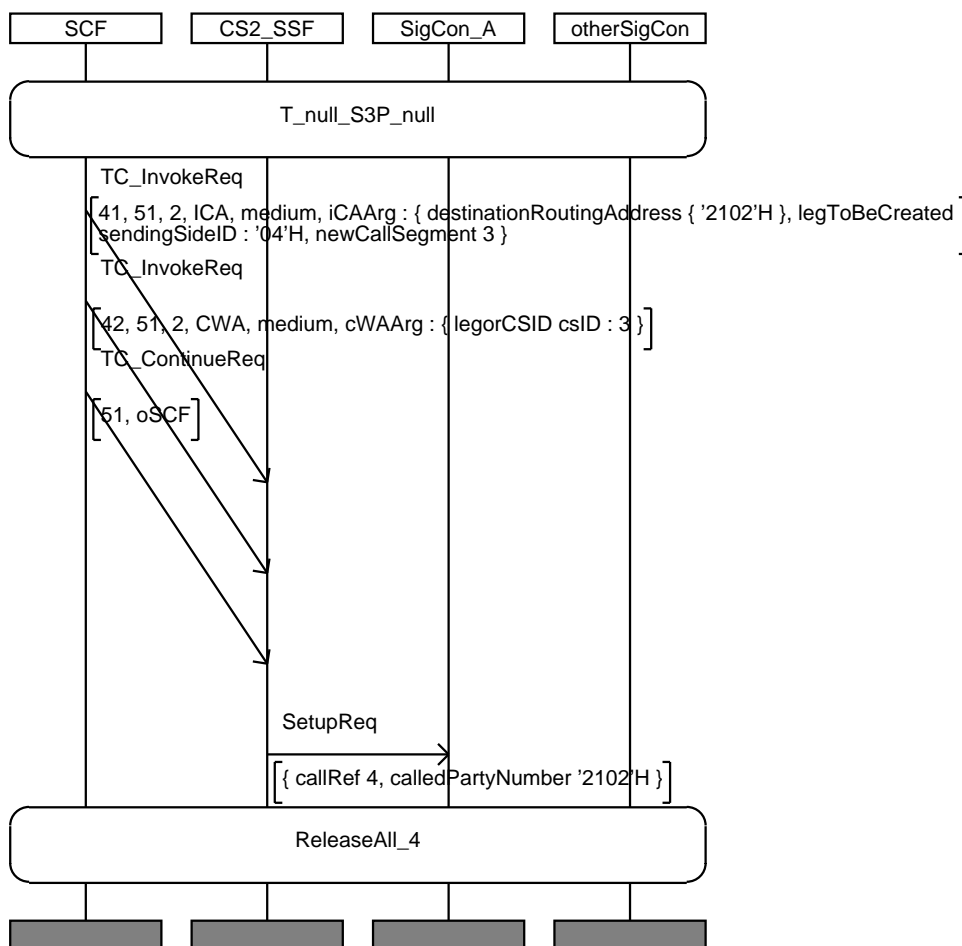
<b>IN2_A_CPH_046</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_S3P_null
<b>Test description</b>	L1! DisconnectLeg(3) L1! ContinueWithArgument(CsId=2) L1?DisconnectLegReturnResult Reaching state T_null_S2P_null
<b>Pass criteria</b>	CP1_3? ReleaseReq
<b>Postamble:</b>	ReleaseAll_1

# MSC IN2\_A\_CPH\_046



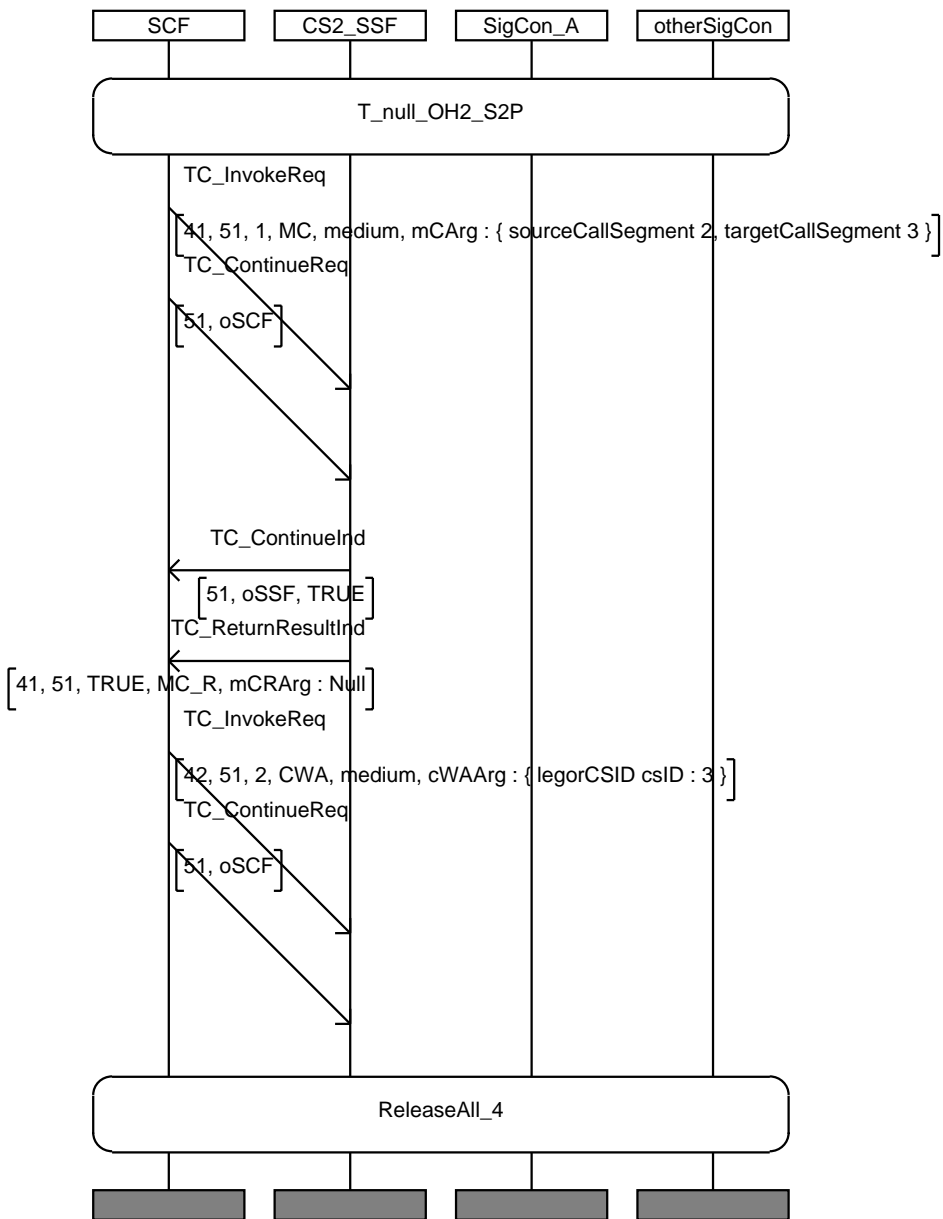
IN2_A_CPH_047	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_S3P_null
<b>Test description</b>	L1! InitiateCallAttempt(4,3) L1! ContinueWithArgument(CsId=3) Reaching state T_null_S3P_S1P
<b>Pass criteria</b>	CP1_4? SetUpReq
<b>Postamble:</b>	ReleaseAll_4

## MSC IN2\_A\_CPH\_047



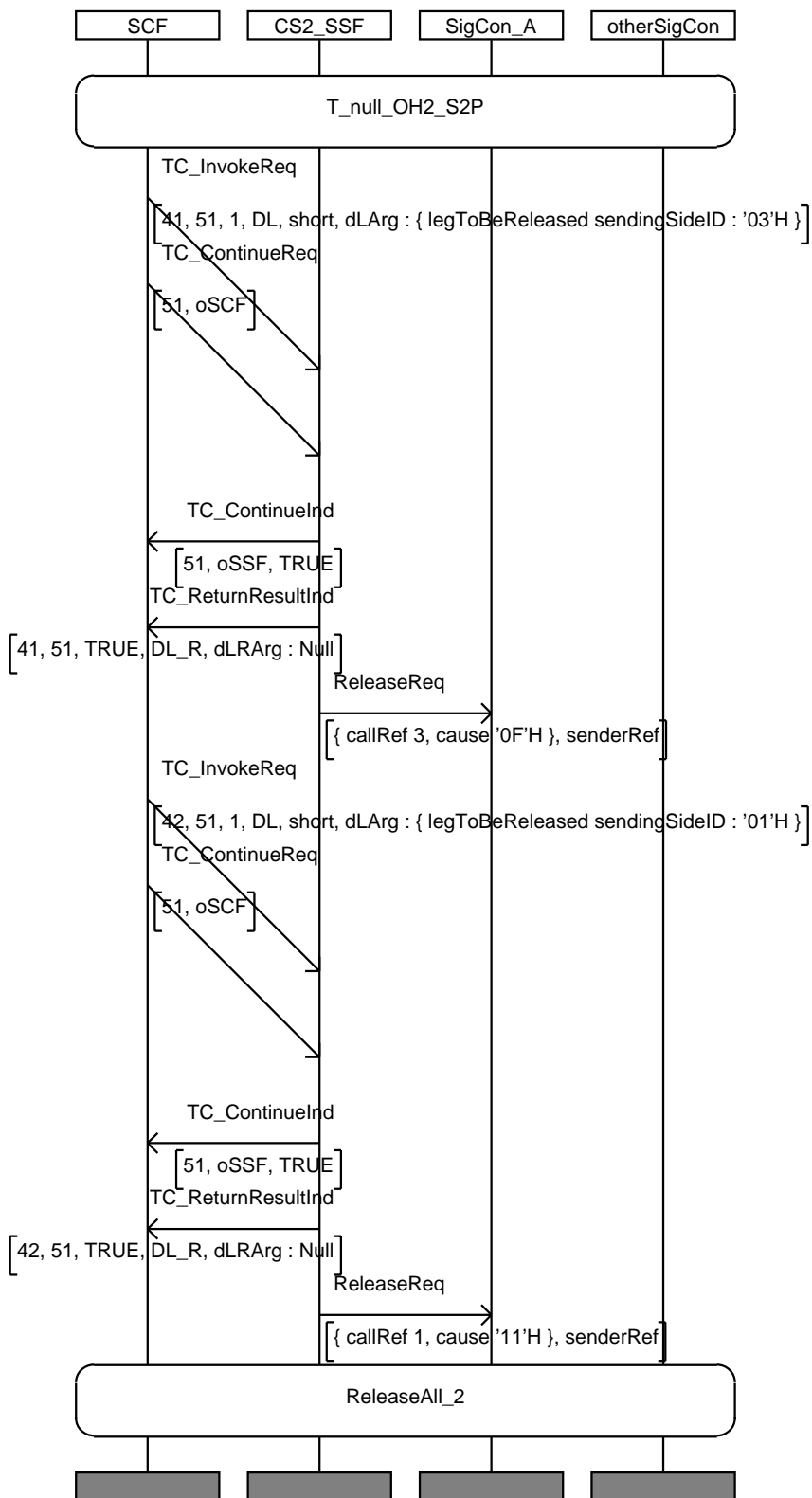
IN2_A_CPH_048	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_OH(2)_S2P
<b>Test description</b>	L1! MergeCallSegments(2,3) L1! ContinueWithArgument(CsId=3) Reaching state T_null_null_S4P
<b>Pass criteria</b>	SSF send MergeCallSegmentsReturnResult
<b>Postamble:</b>	ReleaseAll_4

# MSC IN2\_A\_CPH\_048



<b>IN2_A_CPH_049</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_OH(2)_S2P
<b>Test description</b>	L1! DisconnectLeg(3) L1! DisconnectLeg(1) L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult Reaching state T_null_null_S2P
<b>Pass criteria</b>	CP1_3? ReleaseReq CP1_1? ReleaseReq
<b>Postamble:</b>	ReleaseAll_2

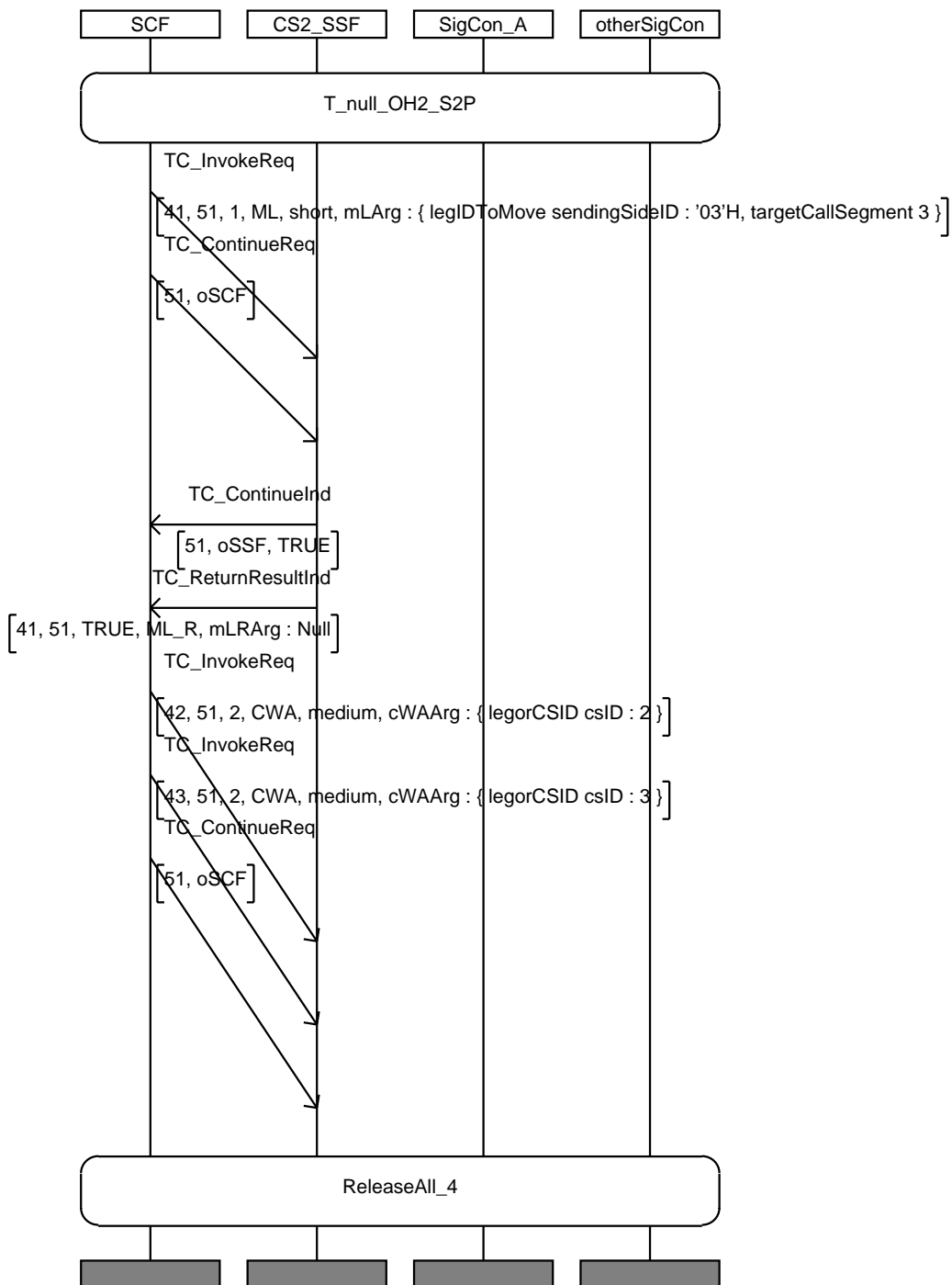
# MSC IN2\_A\_CPH\_049



IN2_A_CPH_050	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_OH(2)_S2P
<b>Test description</b>	L1! MoveLeg(3,3) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state T_null_OH(1)_S3P
<b>Pass criteria</b>	SSF sends MoveLegReturnResult
<b>Postamble:</b>	ReleaseAll_4

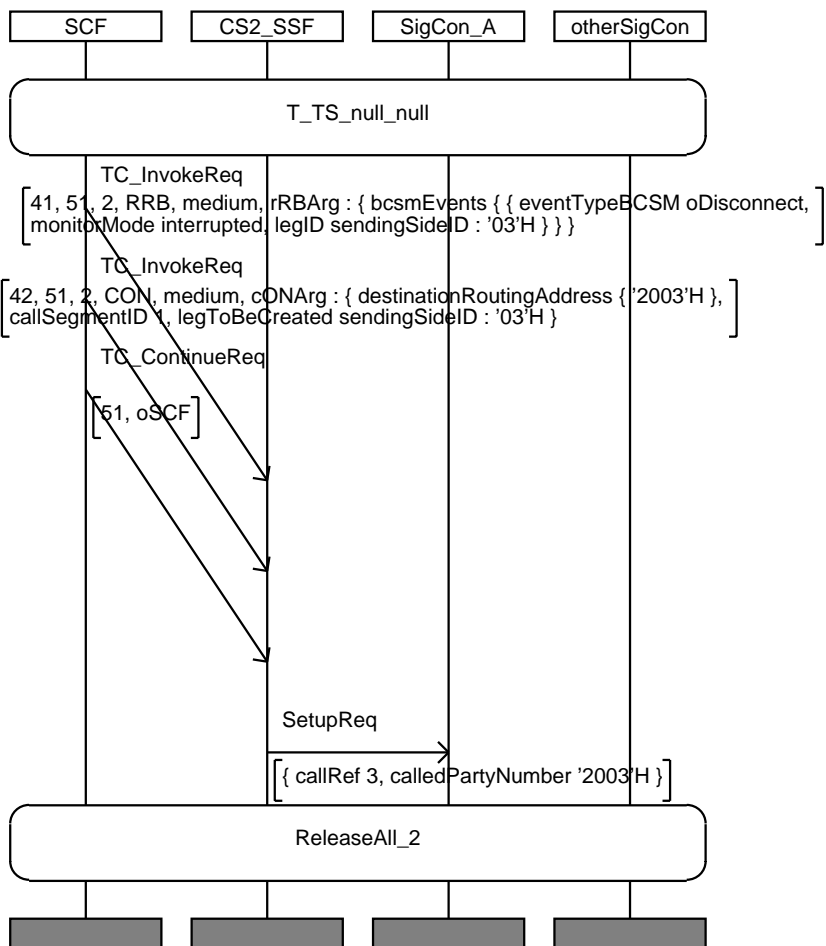


# MSC IN2\_A\_CPH\_050



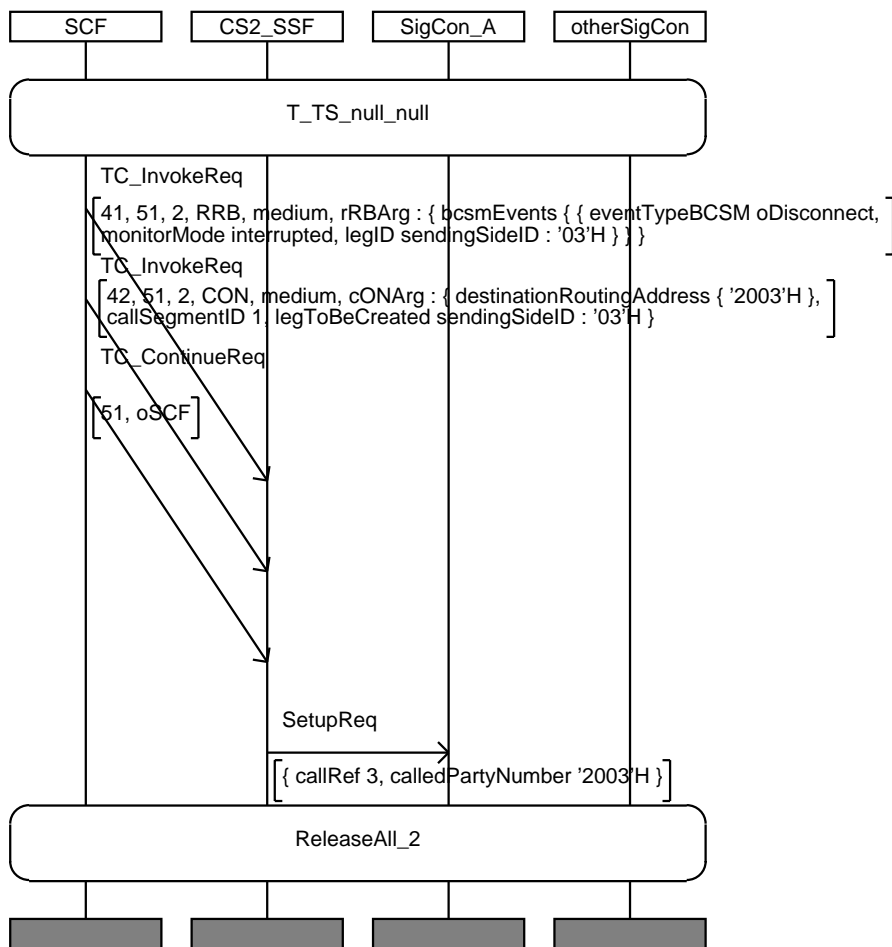
IN2_A_CPH_051	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_TS_null_null
<b>Test description</b>	L1! RequestReportBCSMEvent(3,oDisconnect)  L1! Connect (3,1)  Reaching state T_TF(2)_null_null
<b>Pass criteria</b>	CP1-3? SetupReq
<b>Postamble:</b>	ReleaseAll_2

## MSC IN2\_A\_CPH\_051



<b>IN2_A_CPH_052</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_TS_null_null
<b>Test description</b>	L1! RequestReportBCSMEvent(3,oDisconnect) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(3,1) L1! Connect(4,1) Reaching state T_TF(3)_null_null
<b>Pass criteria</b>	CP1_3? SetUpReq CP1_4? SetUpReq
<b>Postamble:</b>	ReleaseAll_2

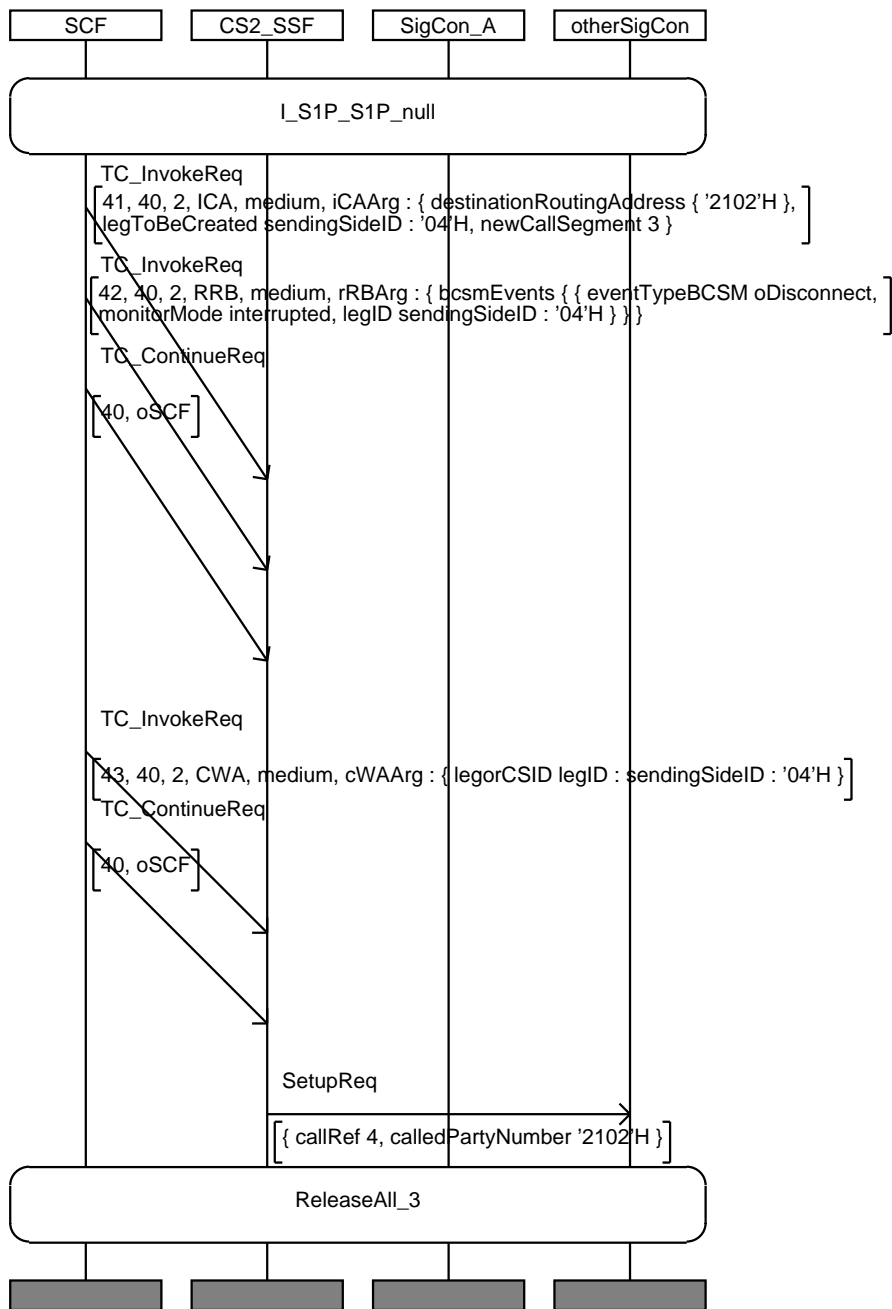
# MSC IN2\_A\_CPH\_052



## 7.5.3 Network initiated

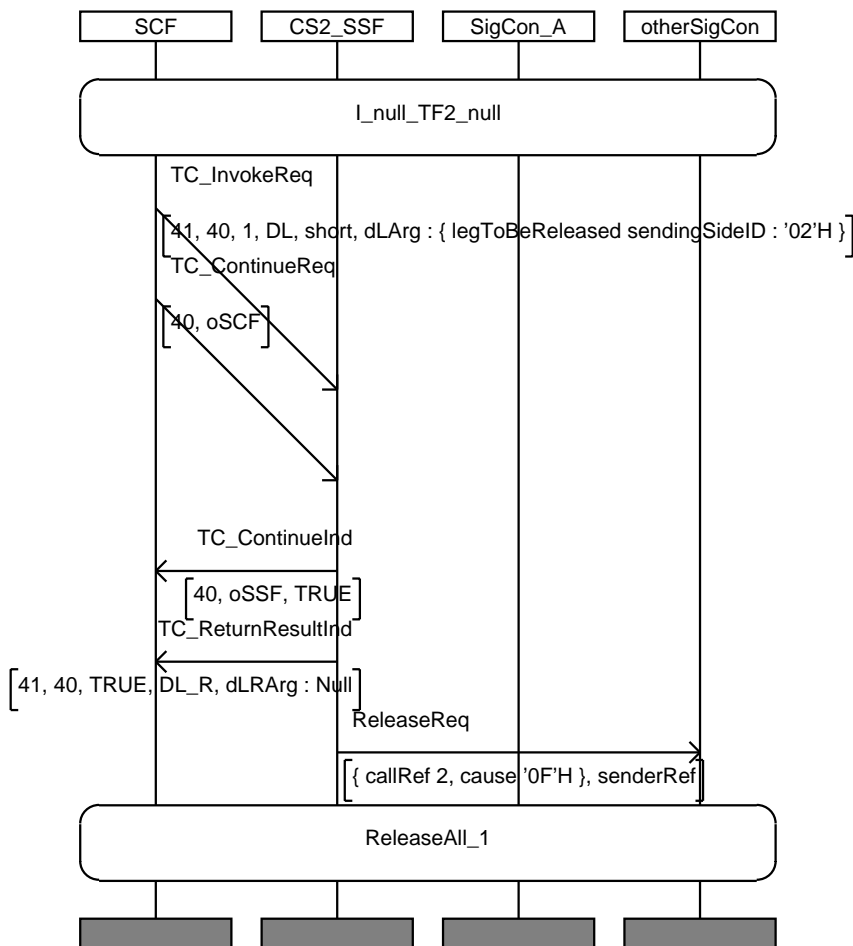
<b>IN2_A_CPH_053</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_S1P_S1P_null
<b>Test description</b>	L1! InitiateCallAttempt(4,3) L1! RequestReportBCSMEvent(4,oDisconnect) L1! ContinueWithArgument(LegId=4) Reaching state I_S1P_S1P_S1P
<b>Pass criteria</b>	CP1_4? SetUpIpd
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_053



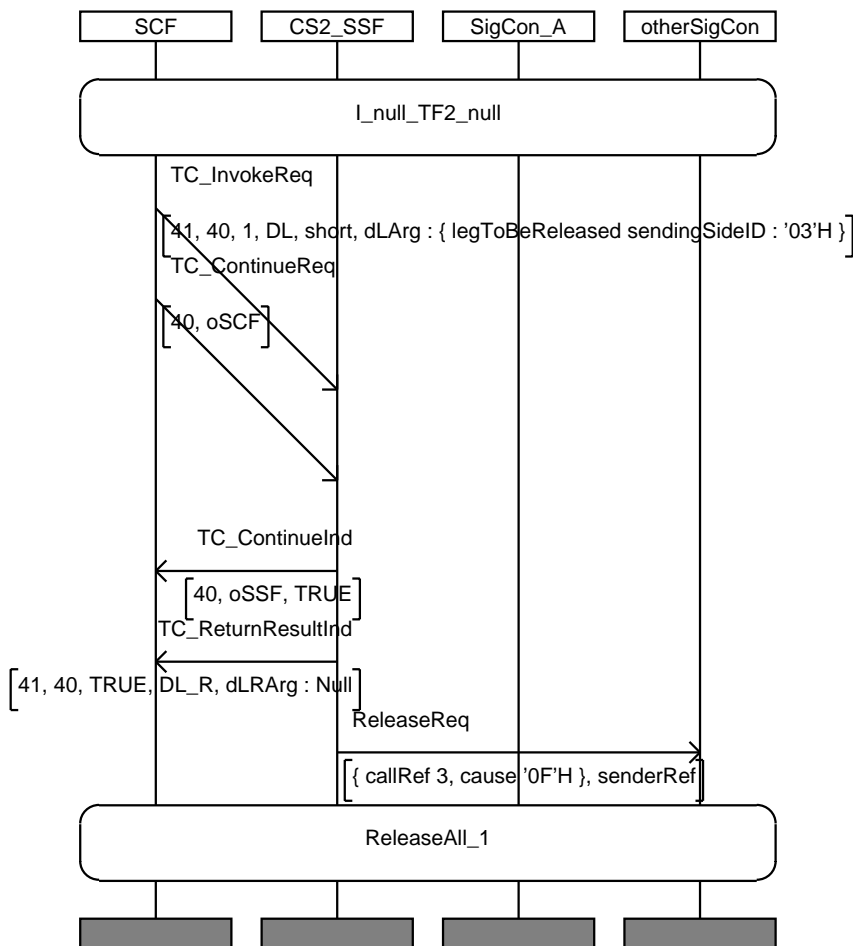
IN2_A_CPH_054	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF2_null
<b>Test description</b>	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult Reaching state I_null_S1P_null
<b>Pass criteria</b>	CP1_2? ReleaseReq
<b>Postamble:</b>	ReleaseAll_1

### MSC IN2\_A\_CPH\_054



IN2_A_CPH_055	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF2_null
<b>Test description</b>	L1! DisconnectLeg(3) L1? DisconnectLegReturnResult Reaching state I_null_S1P_null
<b>Pass criteria</b>	CP1_3? ReleaseReq
<b>Postamble:</b>	ReleaseAll_1

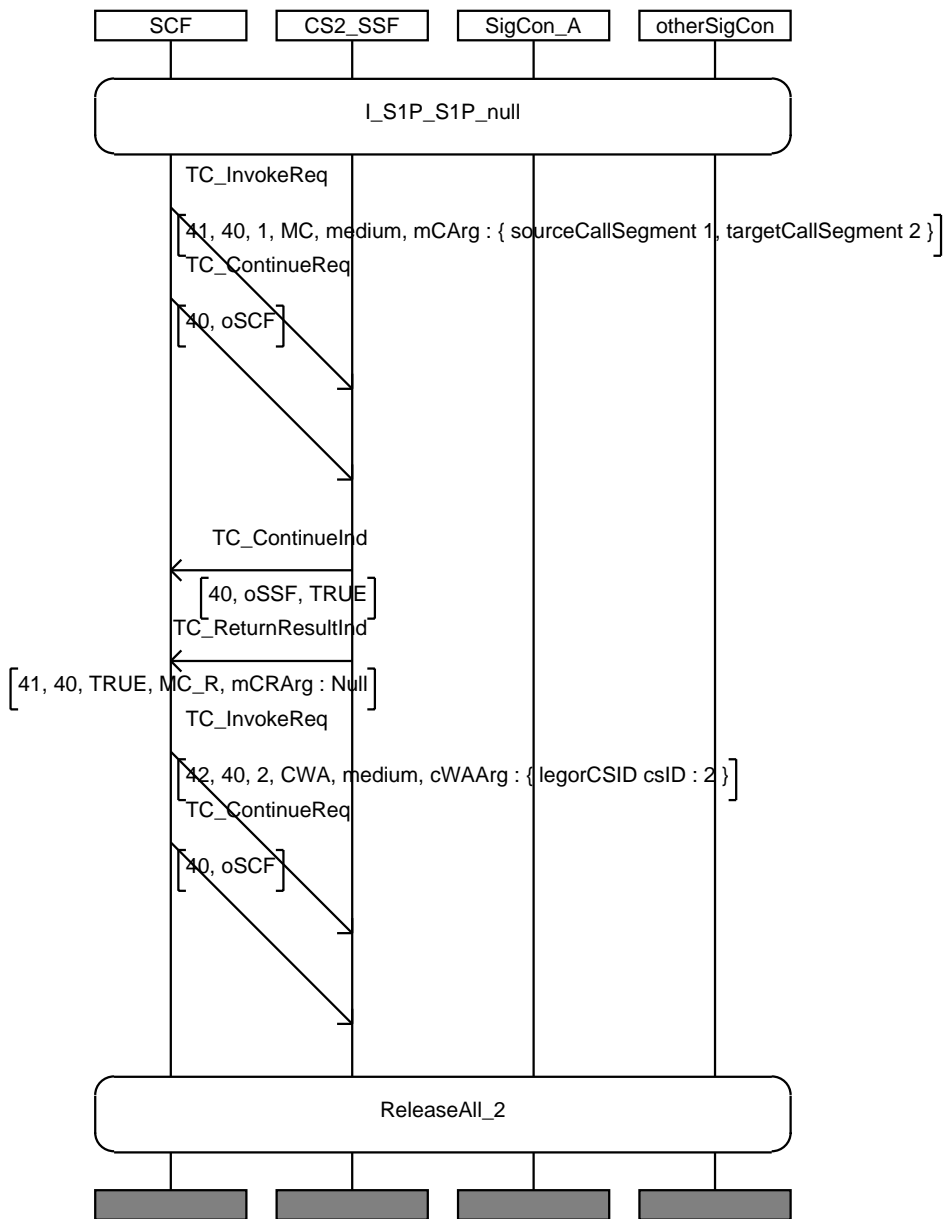
## MSC IN2\_A\_CPH\_055





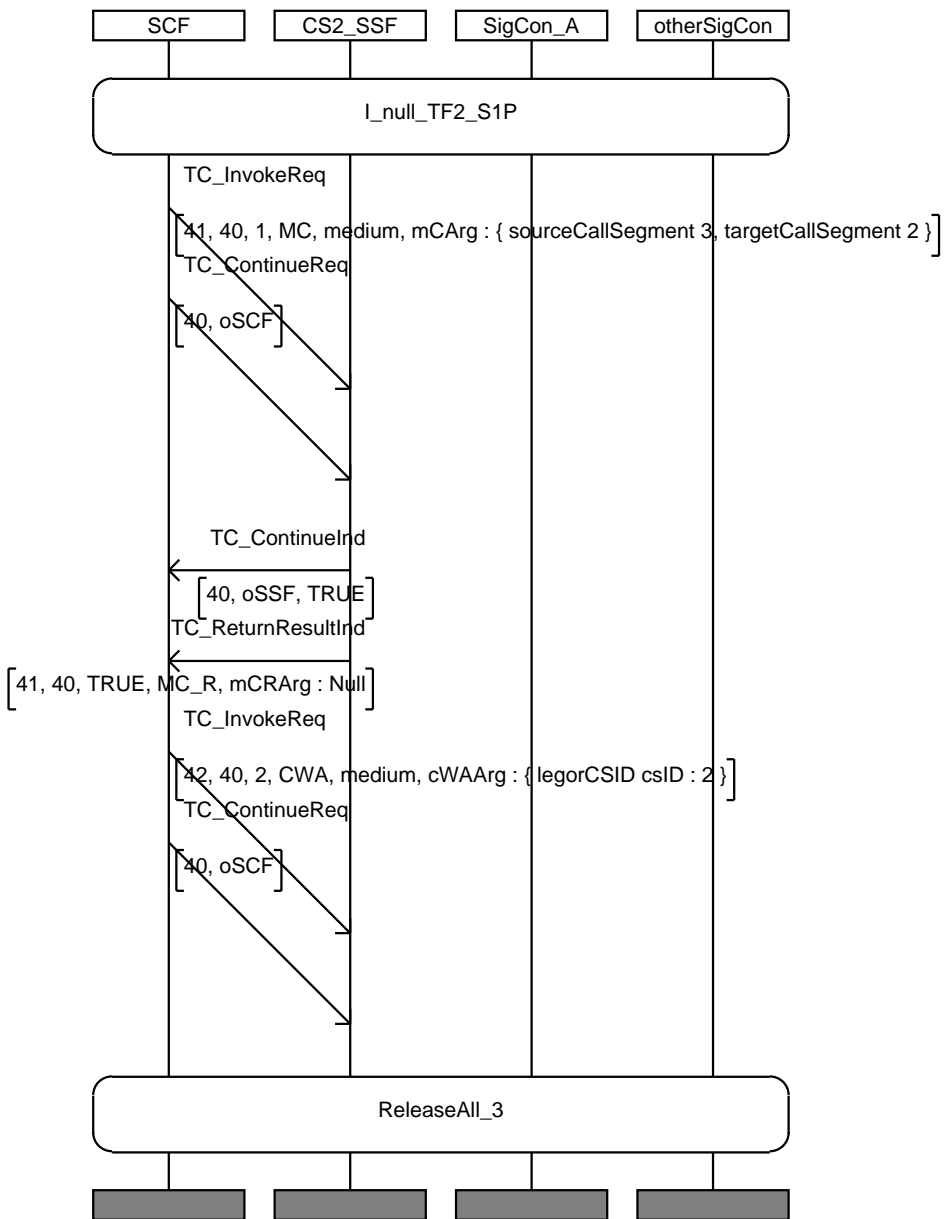
<b>IN2_A_CPH_056</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_S1P_S1P_null
<b>Test description</b>	L1! MergeCallSegments(1,2) L1! ContinueWithArgument(CsId=2) Reaching state I_null_TF(2)_null
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_2

# MSC IN2\_A\_CPH\_056



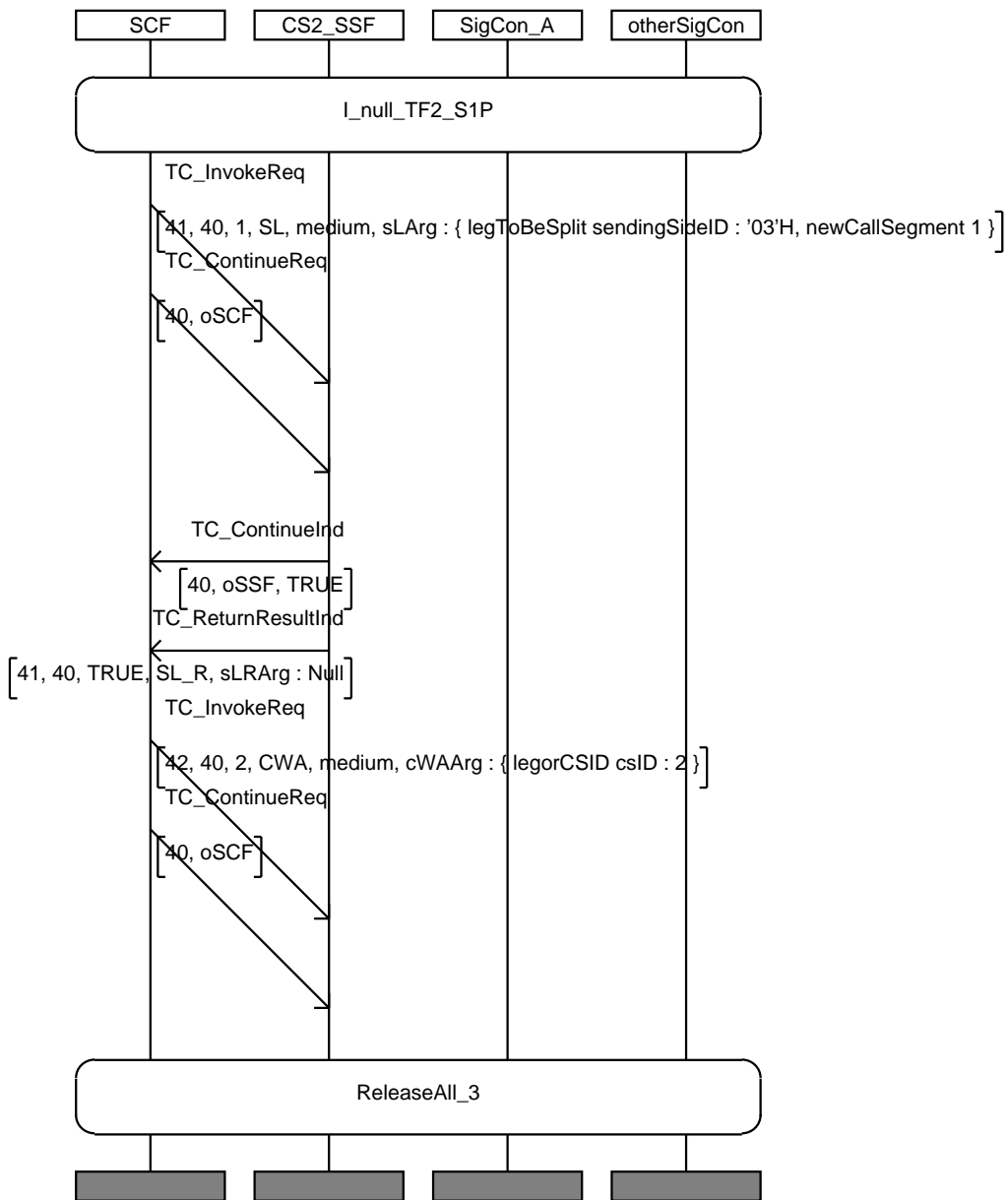
<b>IN2_A_CPH_057</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF(2)_S1P
<b>Test description</b>	L1! MergeCallSegments(3,2) L1! ContinueWithArgument(CsId=2) Reaching state I_null_TF(3)_null
<b>Pass criteria</b>	SSF sends MergeCallSegmentReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_057



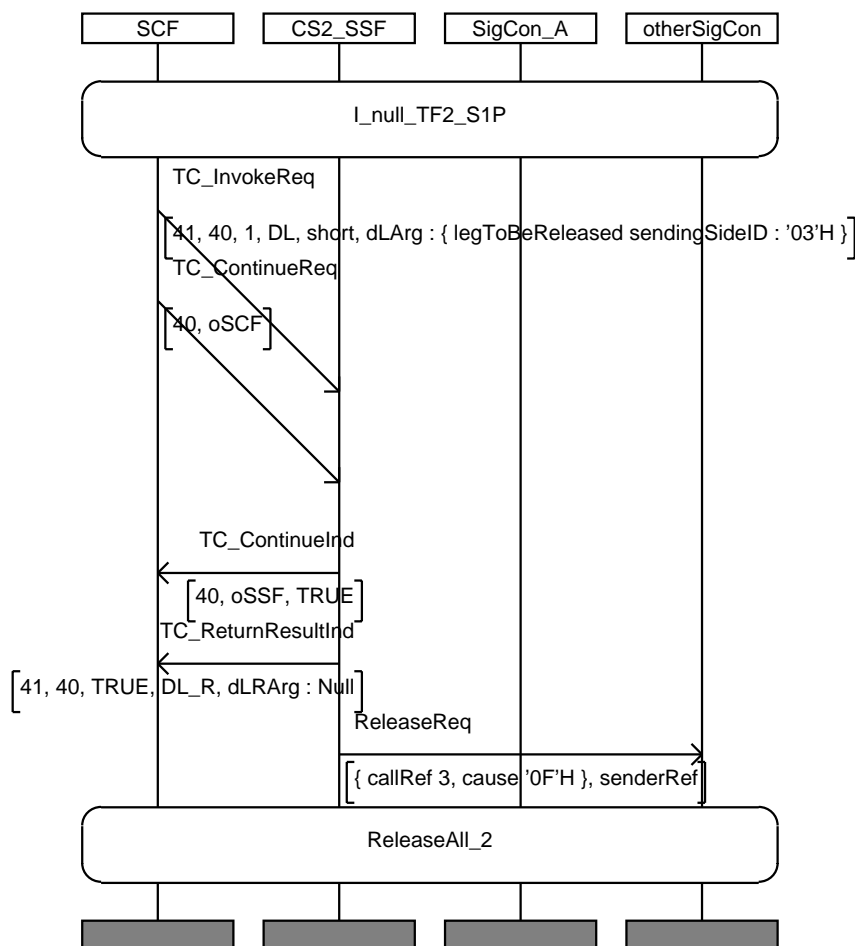
<b>IN2_A_CPH_058</b>	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF(2)_S1P
<b>Test description</b>	L1! SplitLeg(3,1) L1! ContinueWithArgument(CsId=2) Reaching state I_S1P_S1P_S1P
<b>Pass criteria</b>	SSF sends SplitLegReturnResult
<b>Postamble:</b>	ReleaseAll_3

# MSC IN2\_A\_CPH\_058

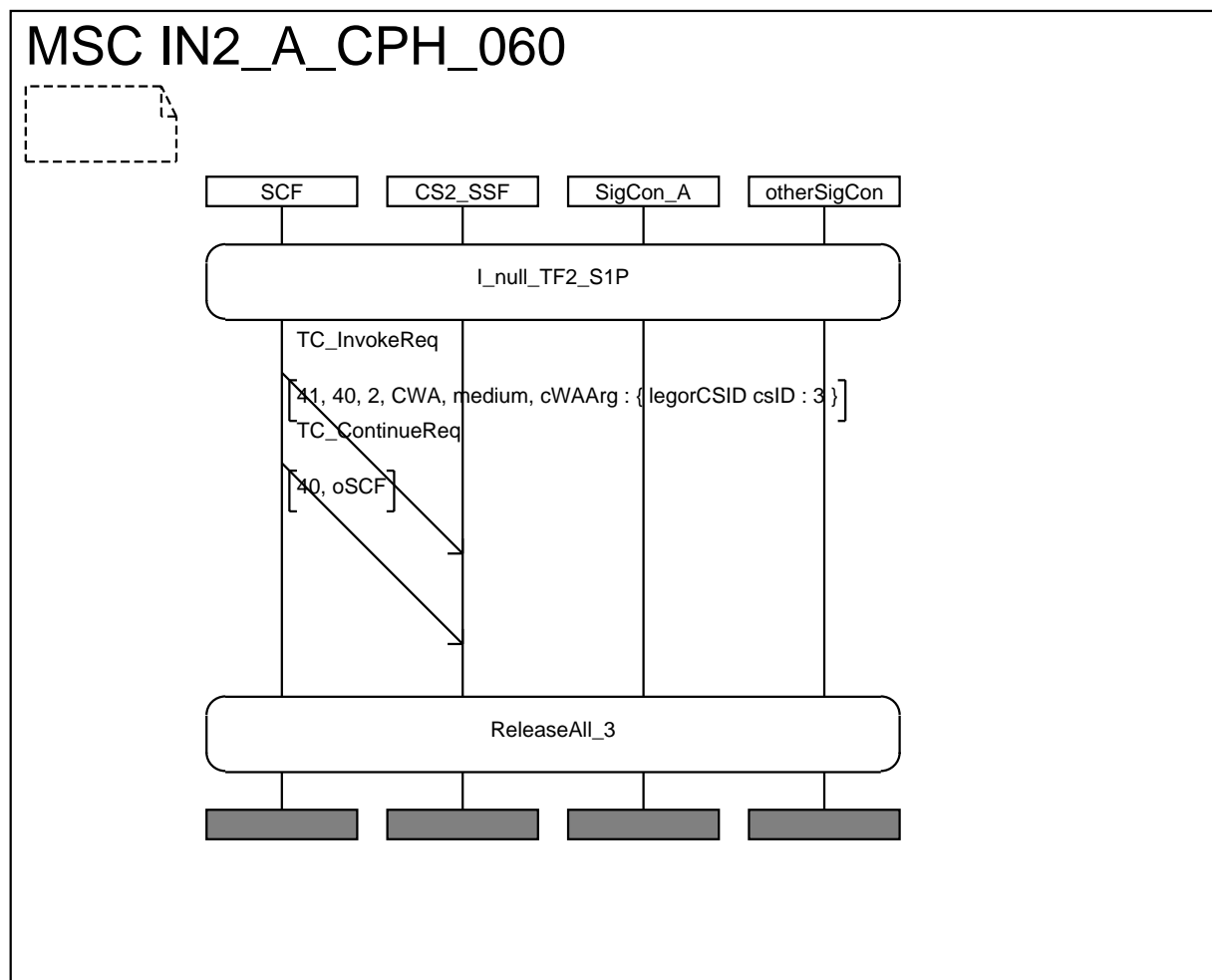


IN2_A_CPH_059	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF(2)_S1P
<b>Test description</b>	L1! DisconnectLeg(3) L1? DisconnectLegReturnResult Reaching state I_null_S1P_S1P
<b>Pass criteria</b>	Pass criteria: CP1_3? ReleaseReq
<b>Postamble:</b>	ReleaseAll_2

## MSC IN2\_A\_CPH\_059



IN2_A_CPH_060	
<b>Purpose:</b>	Test CPH capabilities
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF(2)_S1P
<b>Test description</b>	L1! ContinuewithArgument(Csld=3)
<b>Pass criteria</b>	The operation is accepted
<b>Postamble:</b>	ReleaseAll_3



## 7.6 Test Purpose (TP) descriptions for testing arming/detecting rules

This section includes a set the TPs to check that the IUT acts according the arming/detecting rules.

TPs are listed first for the Originating (O) trigger, then for the Terminating (T) trigger situations.

In these scenarios, the passive legs detection points are armed for signalling events coming from the controlling leg, this means that filtering rules should apply.

Then, the passive legs detection points are armed for signalling events coming from the passive legs.

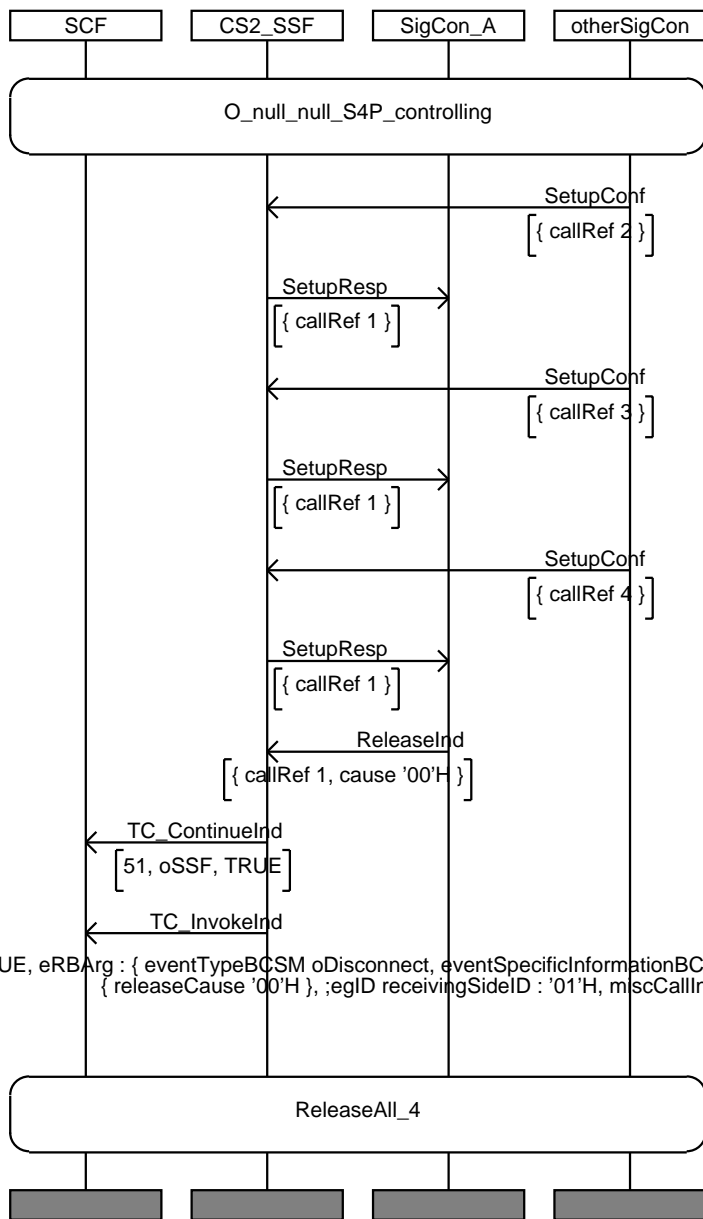


## 7.6.1 Originating (O) trigger

## 7.6.1.1 O\_1 Events coming from the controlling leg (legId=1)

<b>IN2_A_CPH_061</b>	
<b>Purpose:</b>	Check that event filtering rules are applied eventType = oDisconnect
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P_controlling
<b>Test description</b>	CP1_2! SetUpConf CP1_3! SetUpConf CP1_4! SetUpConf CP1_1! ReleaseInd
<b>Pass criteria</b>	L1?: ERB (oDisconnect) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_4

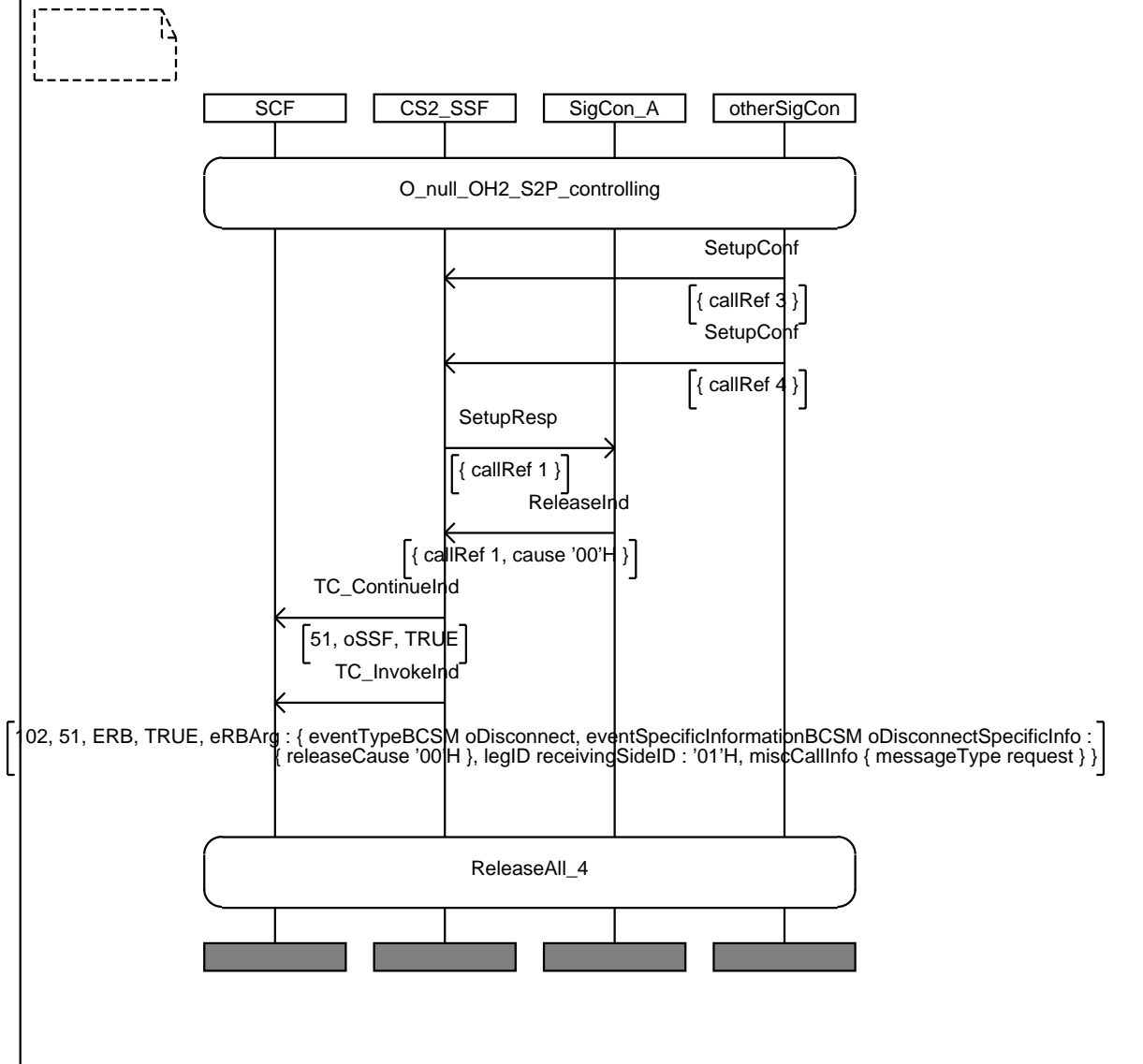
# MSC IN2\_A\_CPH\_061



[ 102, 51, ERB, TRUE, eRBAArg : { eventTypeBCSM oDisconnect, eventSpecificInformationBCSM oDisconnectSpecificInfo : { releaseCause '00'H }, ;egID receivingSideID : '01'H, miscCallInfo { messageType request } } ]

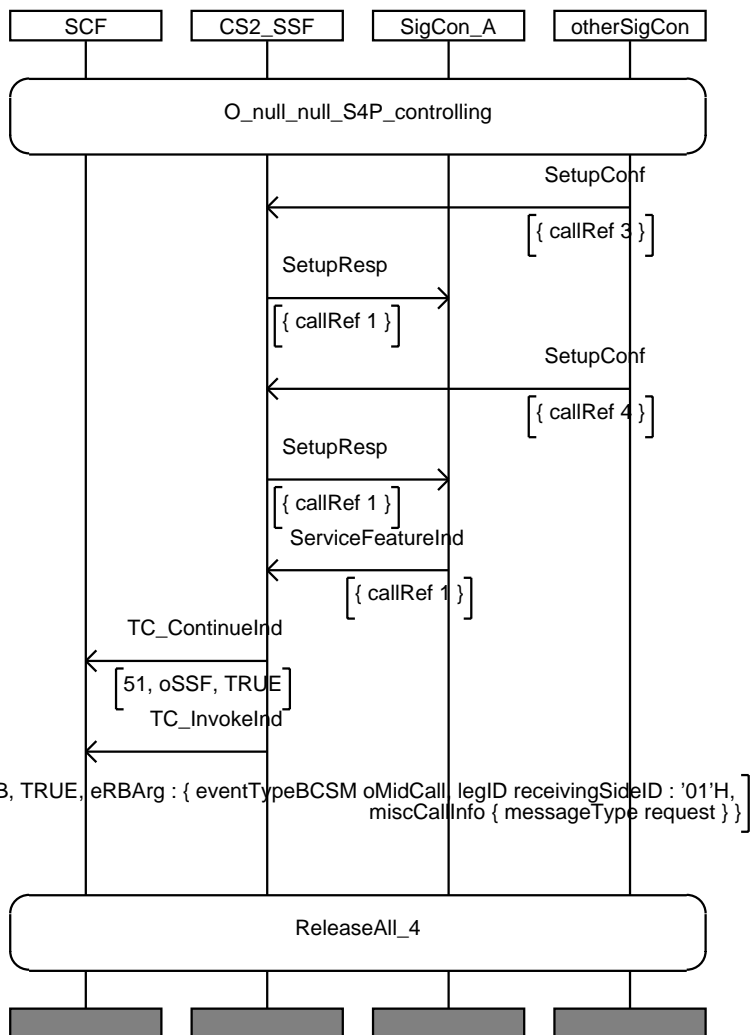
IN2_A_CPH_062	
<b>Purpose:</b>	Check that event filtering rules are applied for multiple callsegments eventType = oDisconnect
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P_controlling
<b>Test description</b>	CP1_3! SetUpConf CP1_4! SetUpConf CP1_1! ReleaseInd
<b>Pass criteria</b>	L1?: ERB (oDisconnect) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_4

## MSC IN2\_A\_CPH\_062



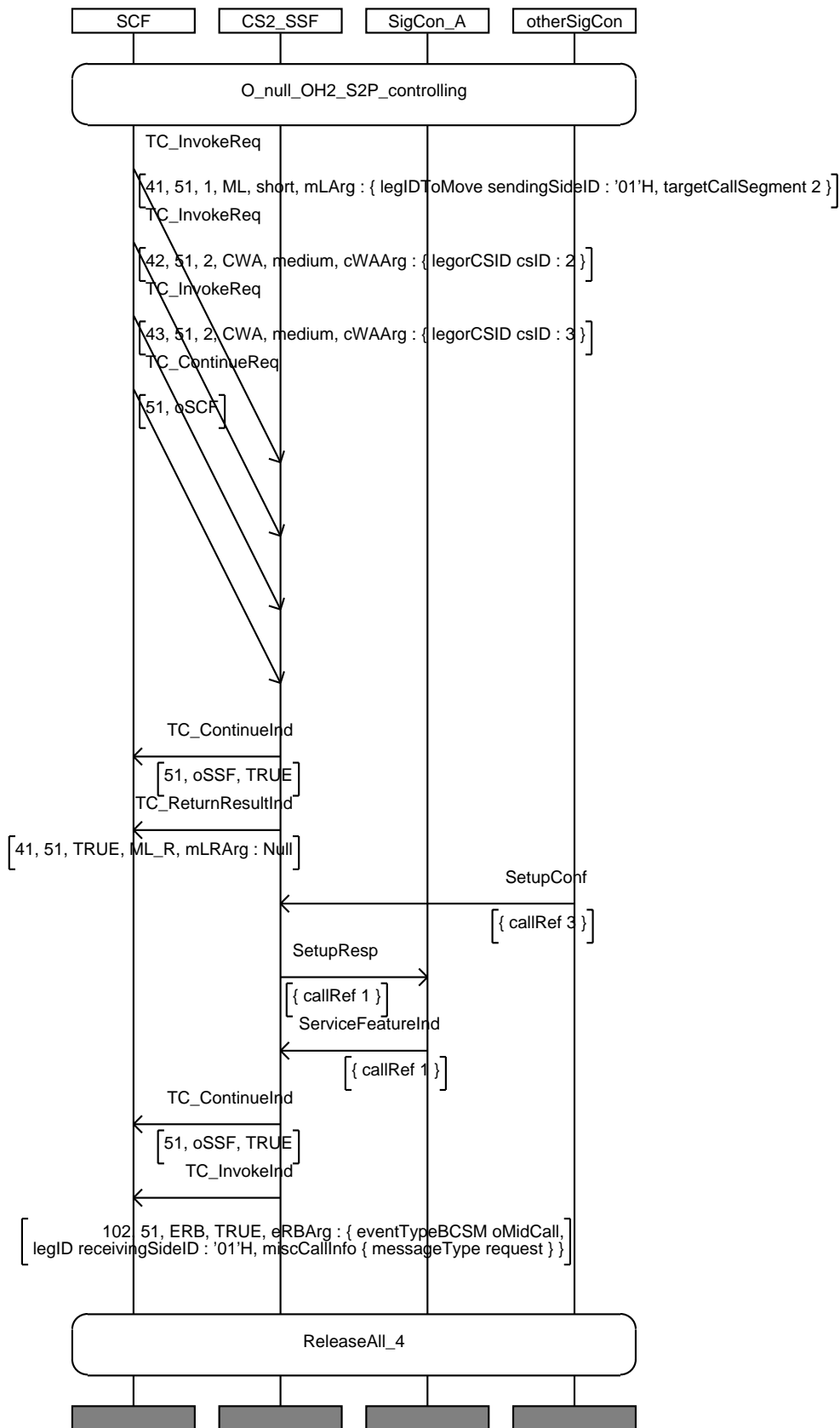
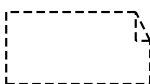
<b>IN2_A_CPH_063</b>	
<b>Purpose:</b>	Check that event filtering rules are applied  eventType = oMidcall
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P_controlling
<b>Test description</b>	CP1_3! SetUpConf  CP1_4! SetUpConf  CP1_1! ServiceFeatureIndication
<b>Pass criteria</b>	L1?: ERB (oMidCall) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_4

# MSC IN2\_A\_CPH\_063



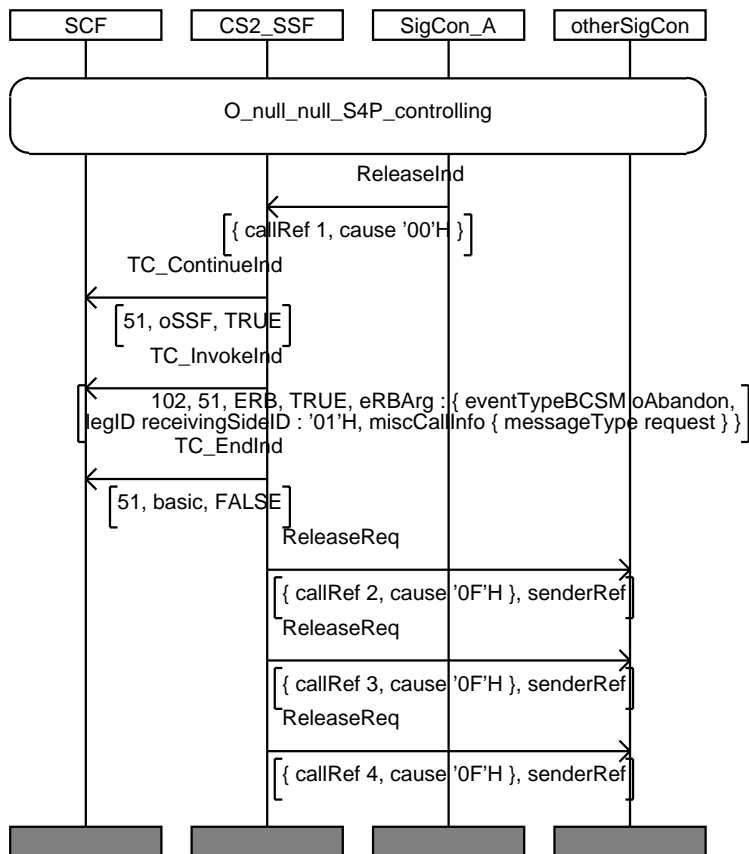
<b>IN2_A_CPH_064</b>	
<b>Purpose:</b>	Check that event filtering rules are applied for multiple callsegments  eventType = oMidcall
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P_controlling
<b>Test description</b>	L1! MoveLeg(1,2)  L1? MoveLegReturnResult  L1! ContinueWithArgument(CsId=2)  L1! ContinueWithArgument(CsId=3)  CP1_3! SetupConf  CP1_1! MidCallInd
<b>Pass criteria</b>	L1?: ERB (oMidCall) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_4

# MSC IN2\_A\_CPH\_064



IN2_A_CPH_065	
<b>Purpose:</b>	Check that event filtering rules are applied eventType = oAbandon
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P_controlling
<b>Test description</b>	CP1_! ReleaseInd
<b>Pass criteria</b>	L1?: ERB (oAbandon) is received only once (filtering rules apply)
<b>Postamble:</b>	none

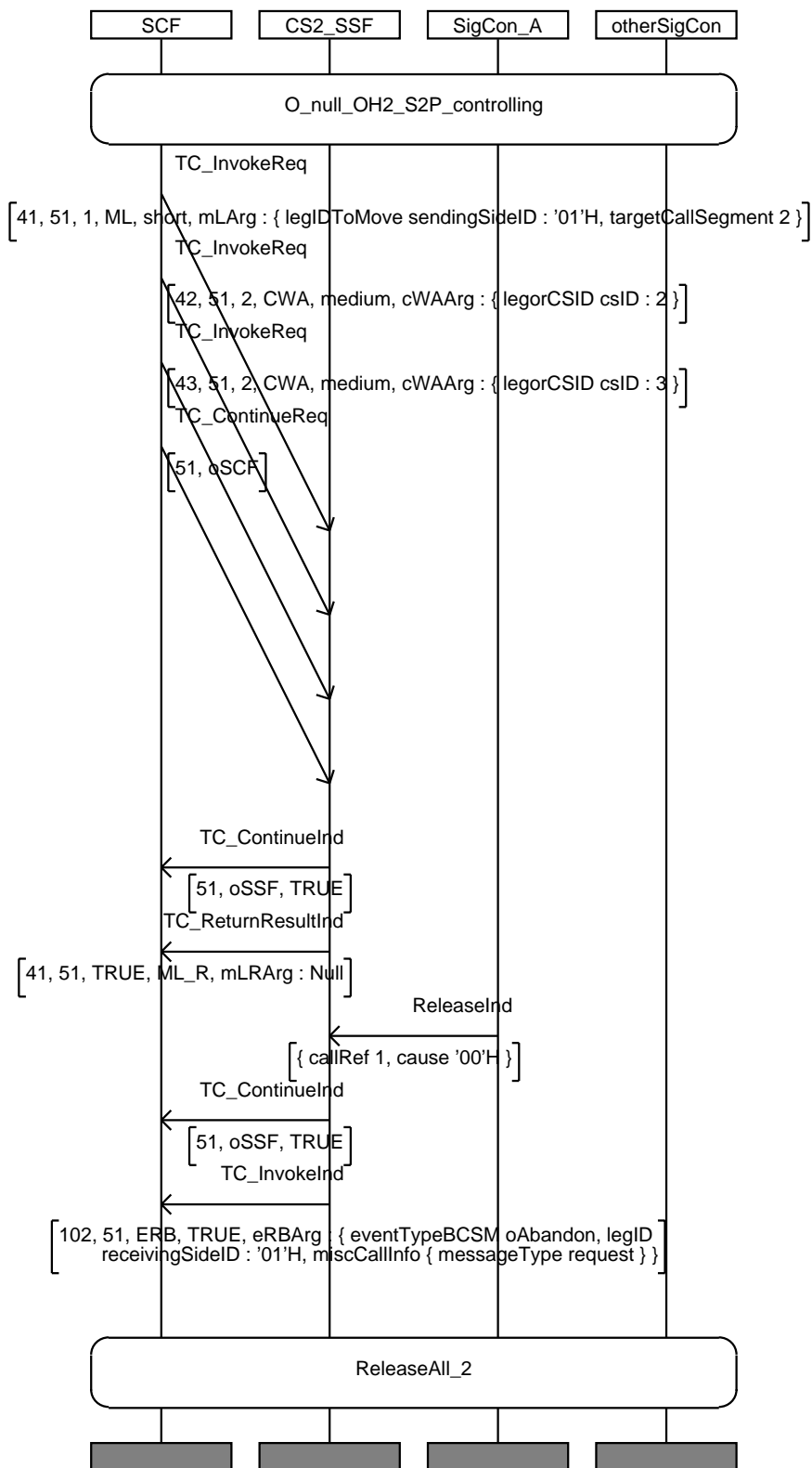
## MSC IN2\_A\_CPH\_065





<b>IN2_A_CPH_066</b>	
<b>Purpose:</b>	Check that event filtering rules are applied for multiple callsegments  eventType = oAbandon
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P_controlling
<b>Test</b>	L1! MoveLeg(1,2)  L1? MoveLegReturnResult  L1! ContinueWithArgumentCsid=2)  L1! ContinueWithArgumentCsid=3)  CP1_1!:ReleaseInd
<b>Pass criteria</b>	L1?: ERB (oAbandon) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_2

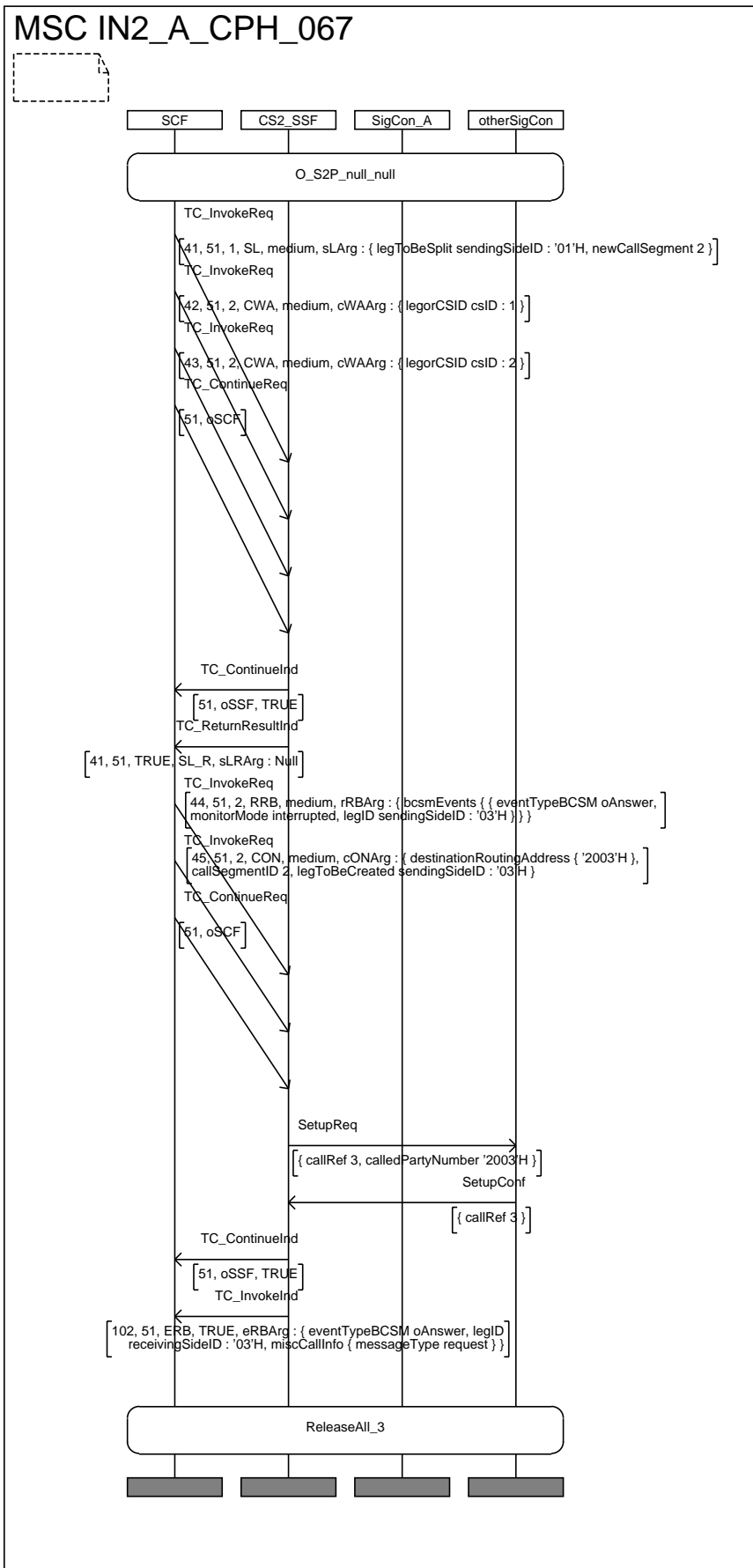
# MSC IN2\_A\_CPH\_066



## 7.6.1.2 O\_2 Events coming from passive legs (legId= 2,3...)

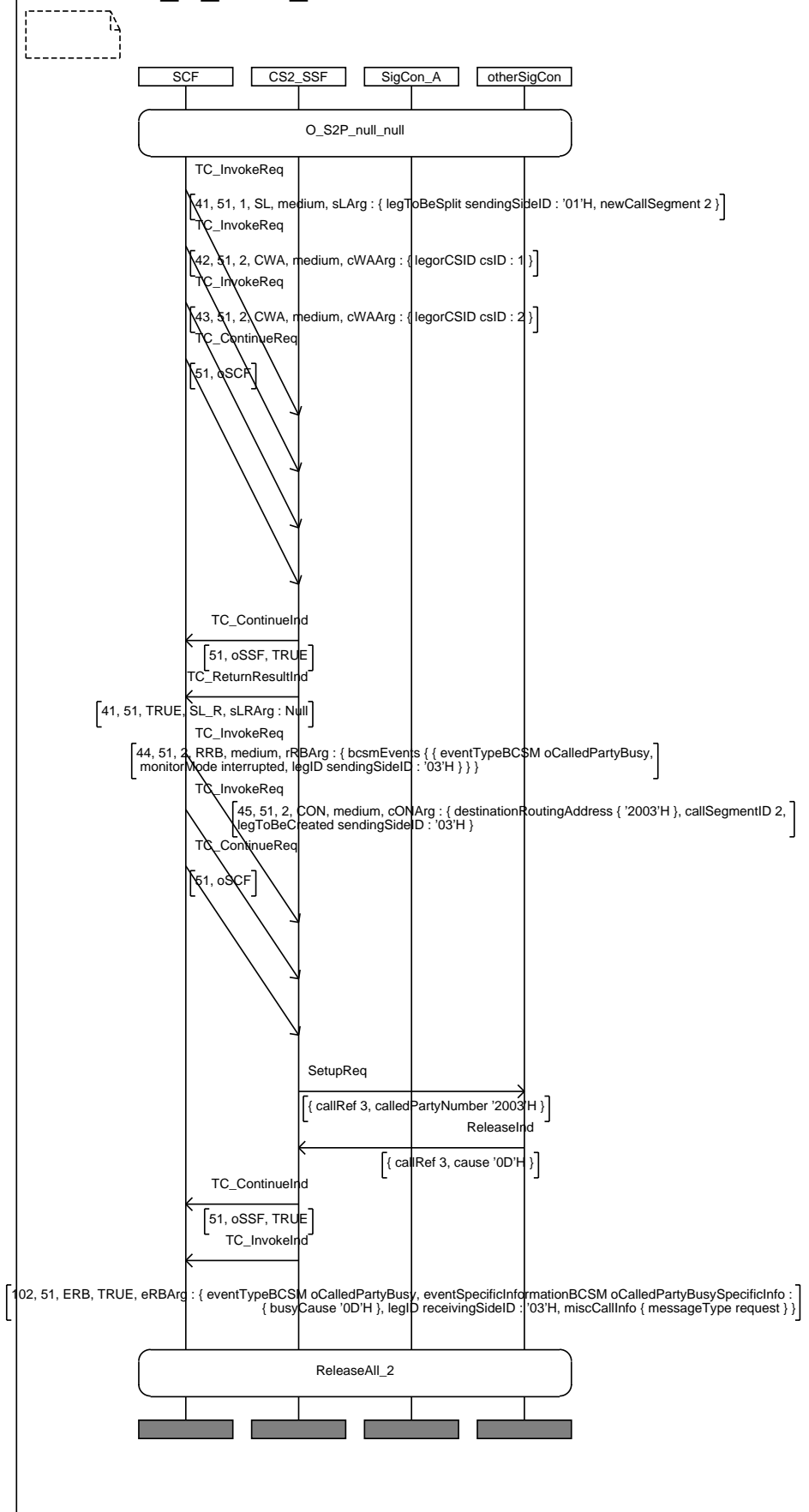
<b>IN2_A_CPH_067</b>	
<b>Purpose:</b>	Check that a single event oAnswer is detected on a passive leg
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test</b>	L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oAnswer) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! SetupConf
<b>Pass criteria</b>	L1? ERB(3,interrupted,oAnswer)
<b>Postamble:</b>	ReleaseAll_3

MSC IN2\_A\_CPH\_067

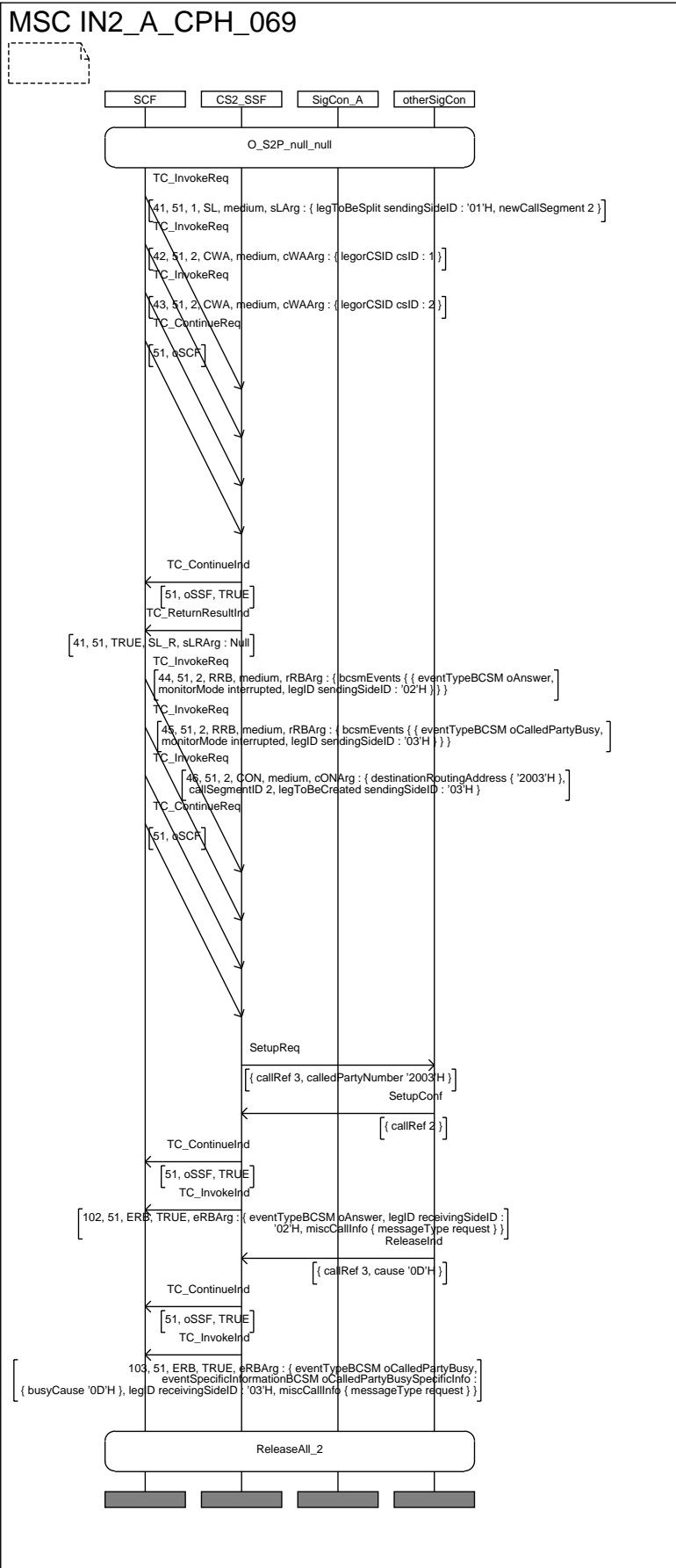


<b>IN2_A_CPH_068</b>	
<b>Purpose:</b>	Check that a single event oCalledPartyBusy is detected on a passive leg
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test</b>	L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oBusy, notify, oAnswer) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(3,interrupted,oBusy)
<b>Postamble:</b>	ReleaseAll_2

# MSC IN2\_A\_CPH\_068



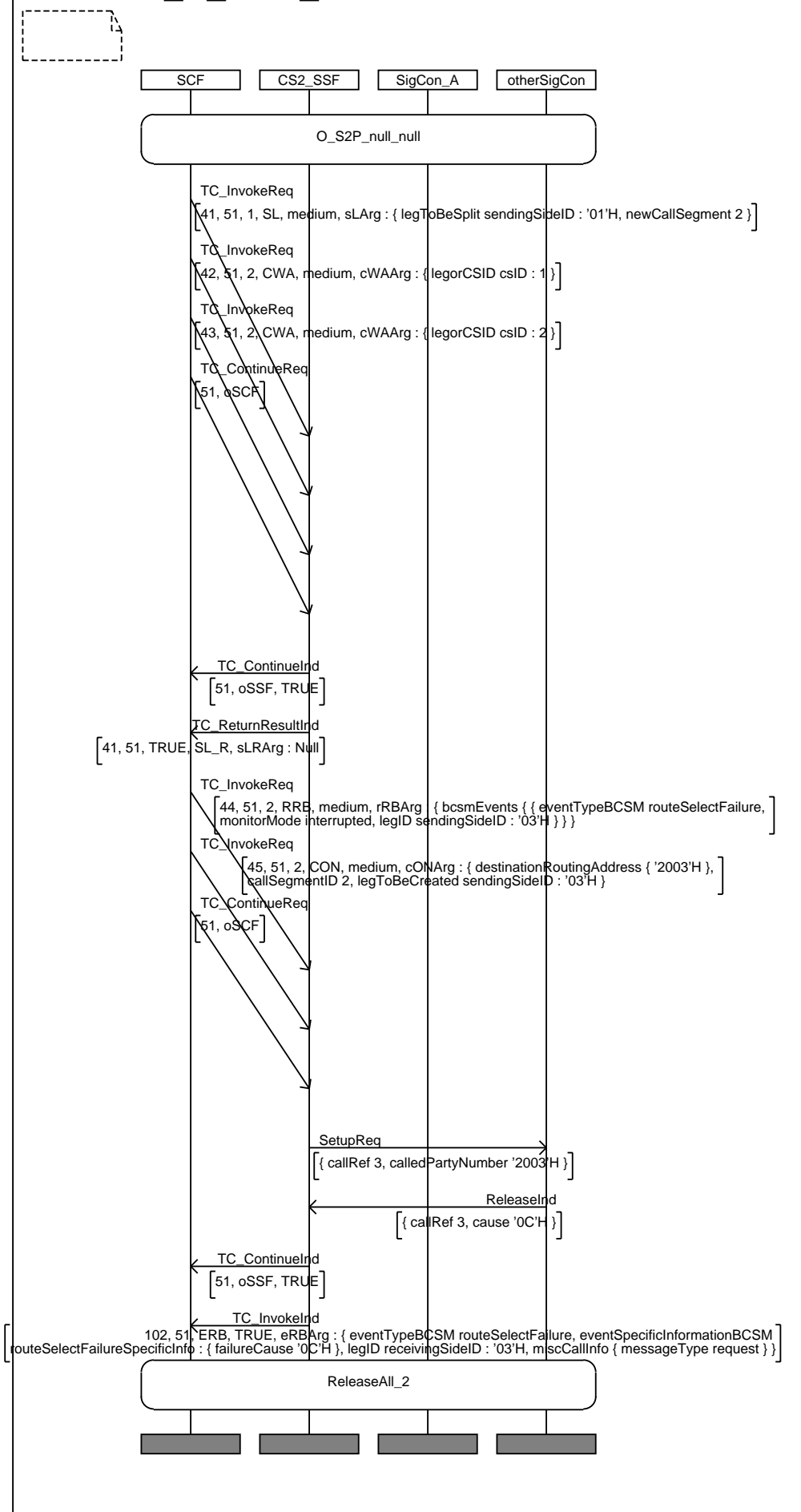
<b>IN2_A_CPH_069</b>	
<b>Purpose:</b>	Check after two events are armed, they are reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test</b>	L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(2, interrupted, oAnswer) L1! RRB(3, interrupted, oBusy) L1! Connect(3,2) CP1_3? SetUpReq CP1_2! SetupConf CP1_3! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(2,interrupted,oAnswer) L1? ERB(3,interrupted,oBusy)
<b>Postamble:</b>	ReleaseAll_2





<b>IN2_A_CPH_070</b>	
<b>Purpose:</b>	Check that a single event routeSelectFailure is detected on a passive leg
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_S2P_null_null
<b>Test</b>	L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oRouteSelectFailure) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! ReleaseInd(RouteSelectFailure cause)
<b>Pass criteria</b>	L1? ERB(3,interrupted, oRouteSelectFailure)
<b>Postamble:</b>	ReleaseAll_2

# MSC IN2\_A\_CPH\_070



<b>IN2_A_CPH_071</b>
----------------------

This test purpose is not included.

<b>IN2_A_CPH_072</b>
----------------------

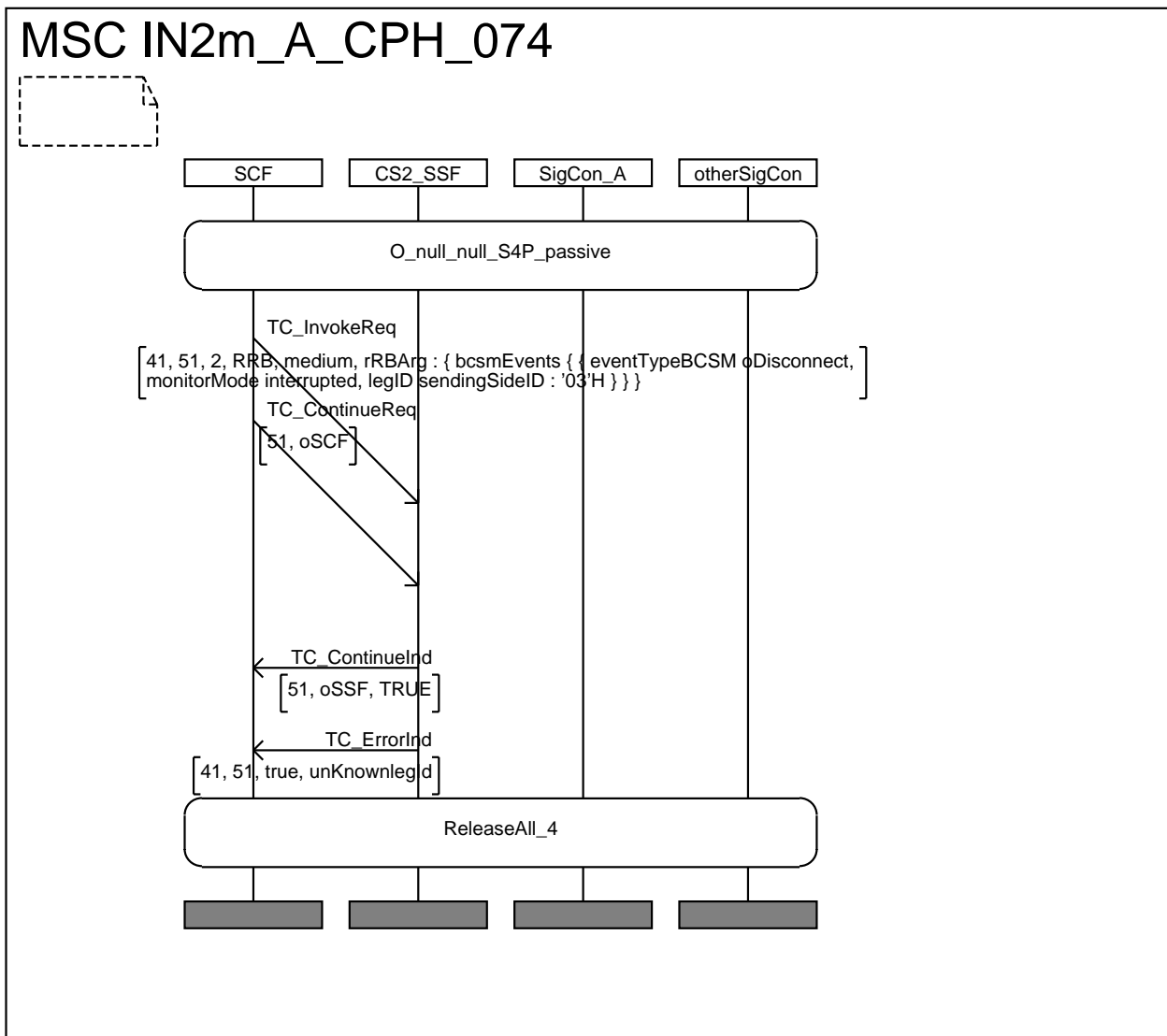
This test purpose is not included.

<b>IN2_A_CPH_073</b>
----------------------

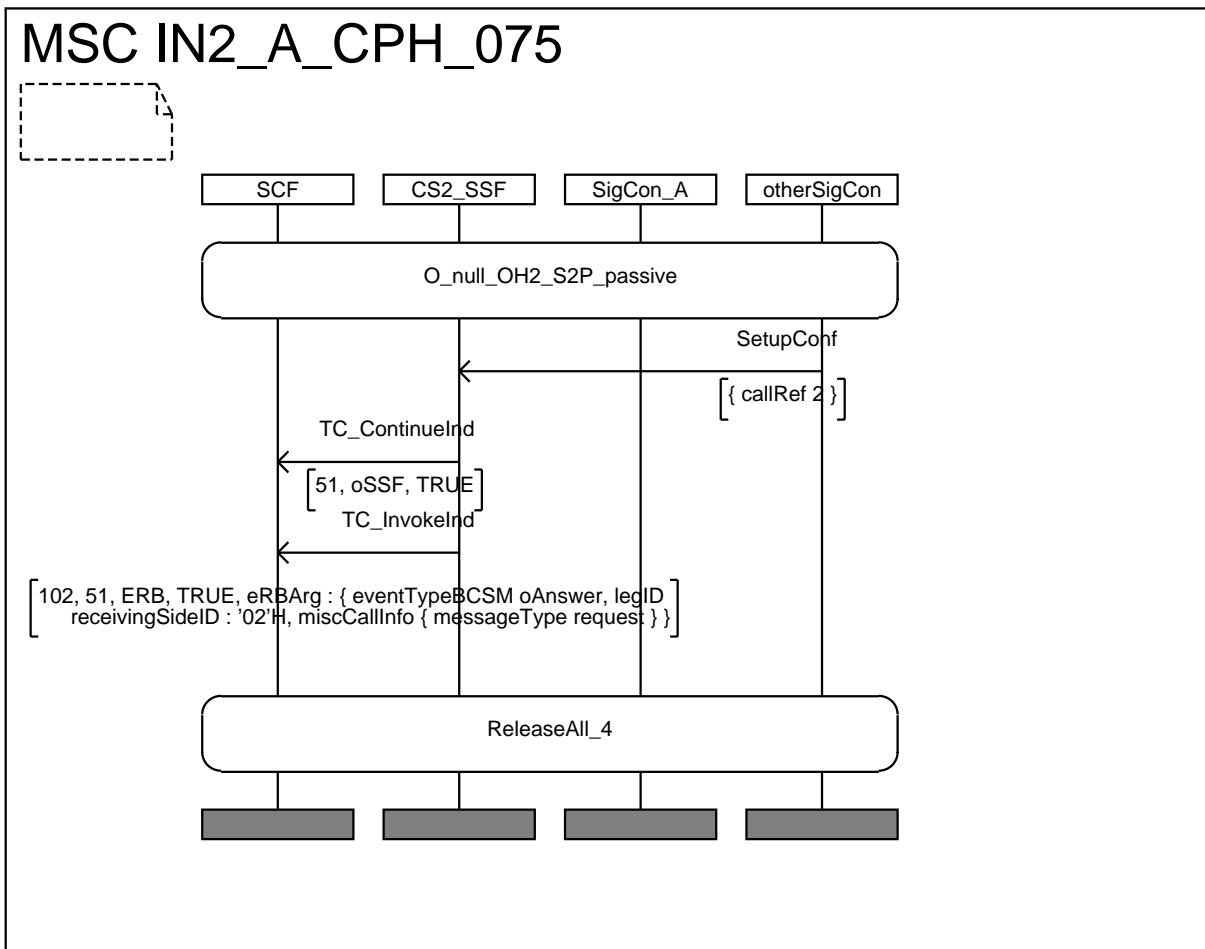
<b>IN2_A_CPH_073</b>	
<b>Purpose:</b>	Check that after two events are armed, then both are reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P_passive
<b>Test description</b>	CP1_2! SetupConf CP1_2! NetworkSuspendInd
<b>Pass criteria</b>	L1? ERB(2,notify, oAnswer) L1? ERB(2,notify, oSuspended)
<b>Postamble:</b>	ReleaseAll_4



IN2_A_CPH_074	
<b>Purpose:</b>	Check that after sending an erroneous component an error is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P_passive
<b>Test description</b>	L1! RRB(6, notify, oSuspend)
<b>Pass criteria</b>	L1? RRB_err (this leg doesn't exist)
<b>Postamble:</b>	ReleaseAll_4



IN2_A_CPH_075	
<b>Purpose:</b>	Check that a single event oAnswer is detected on a passive leg
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P_passive
<b>Test description</b>	CP1_2! SetupConf
<b>Pass criteria</b>	L1? ERB(2, notify, oAnswer)
<b>Postamble:</b>	ReleaseAll_4

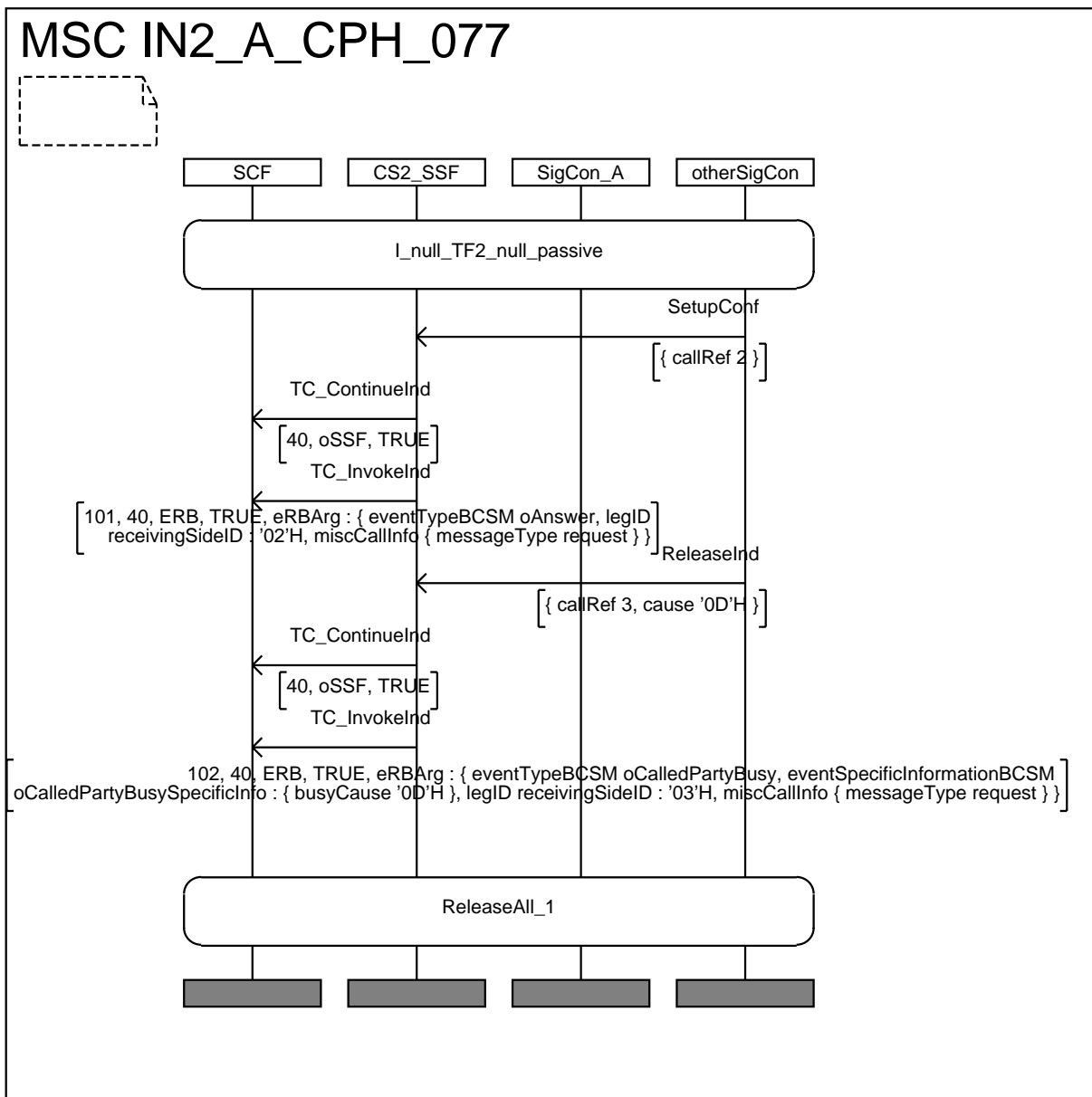


IN2_A_CPH_076
---------------

This test purpose is not included.

IN2_A_CPH_077	
<b>Purpose:</b>	Check that after two events are armed, then both are reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_null_TF(2)_null_passive
<b>Test description</b>	CP1_2! SetUpConf CP1_3! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(2,notify, oAnswer) L1? ERB(3,notify, oBusy)
<b>Postamble:</b>	ReleaseAll_1

## MSC IN2\_A\_CPH\_077

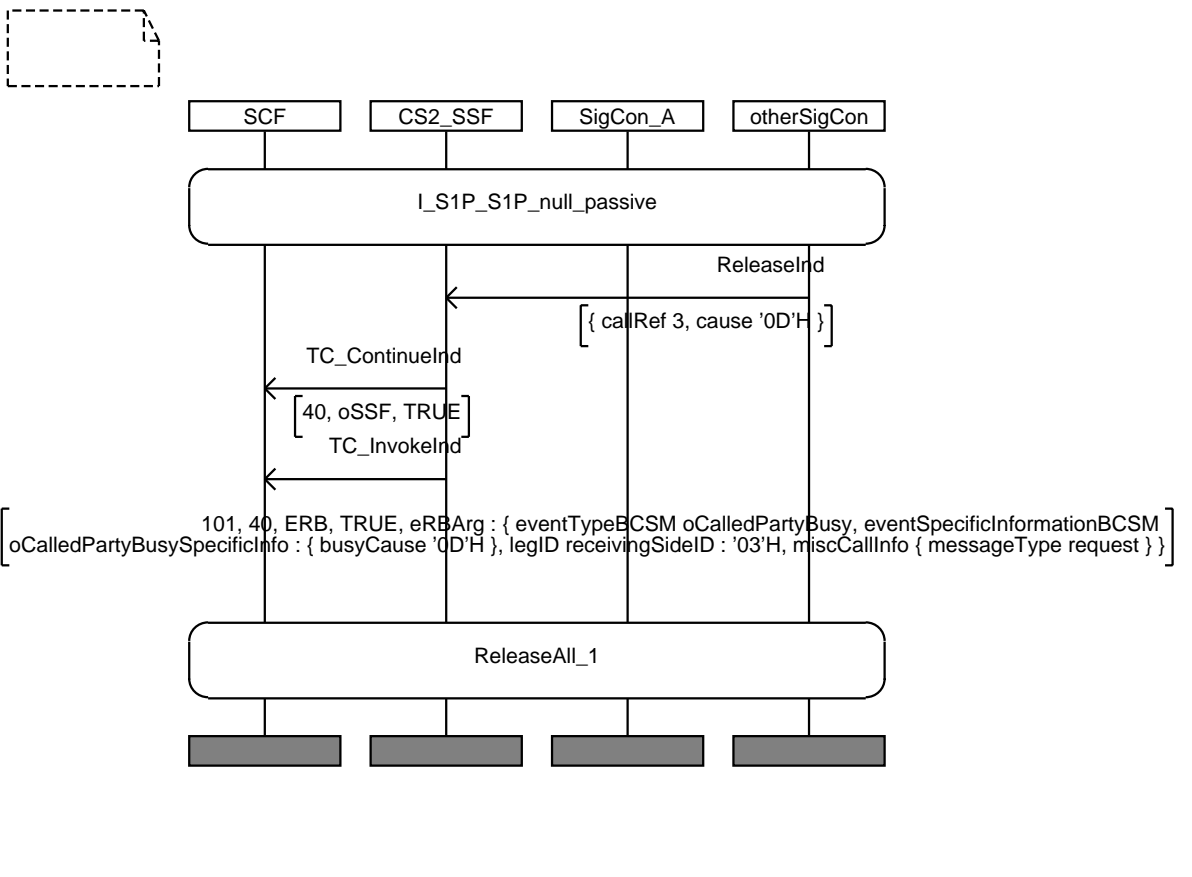


IN2\_A\_CPH\_078

This test purpose is not included.

IN2_A_CPH_079	
<b>Purpose:</b>	Check that after two events are armed, then only one is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_S1P_S1P_null_passive
<b>Test description</b>	CP1_3! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(3,notify, oBusy)
<b>Postamble:</b>	ReleaseAll_1

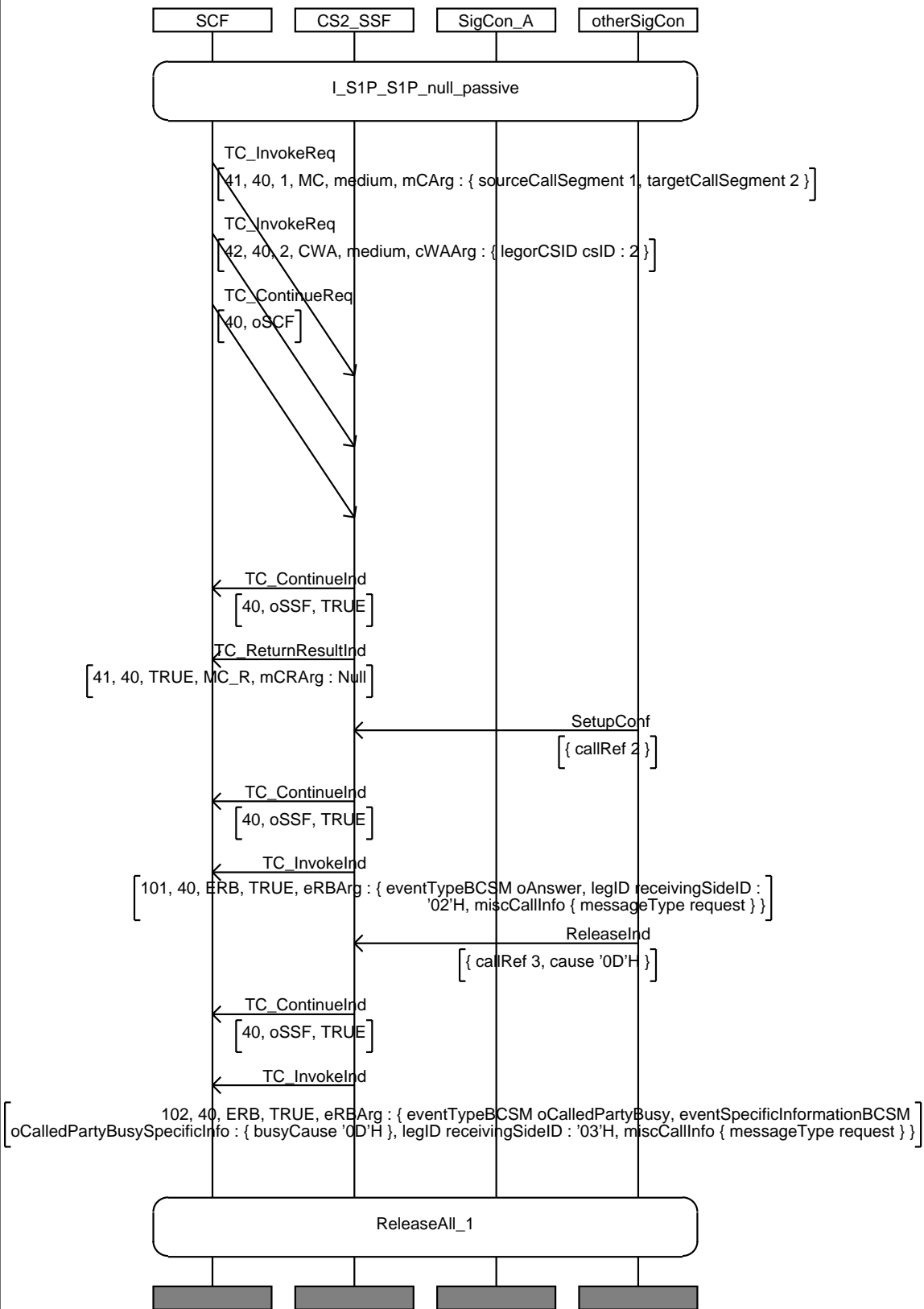
## MSC IN2\_A\_CPH\_079





<b>IN2_A_CPH_080</b>	
<b>Purpose:</b>	Test that after a Merge Call segments operation the events armed remain the same for the passive legs.
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	I_S1P_S1P_null_passive
<b>Test description</b>	L1! MergeCallSegments(1,2) L1? MergeCallSegmentReturnResult L1! ContinueWithArgument(CsId=2) CP1_2! SetUpConf CP1_3! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(2,notify, oAnswer) L1? ERB(3,notify, oBusy)
<b>Postamble:</b>	ReleaseAll_1

# MSC IN2\_A\_CPH\_080

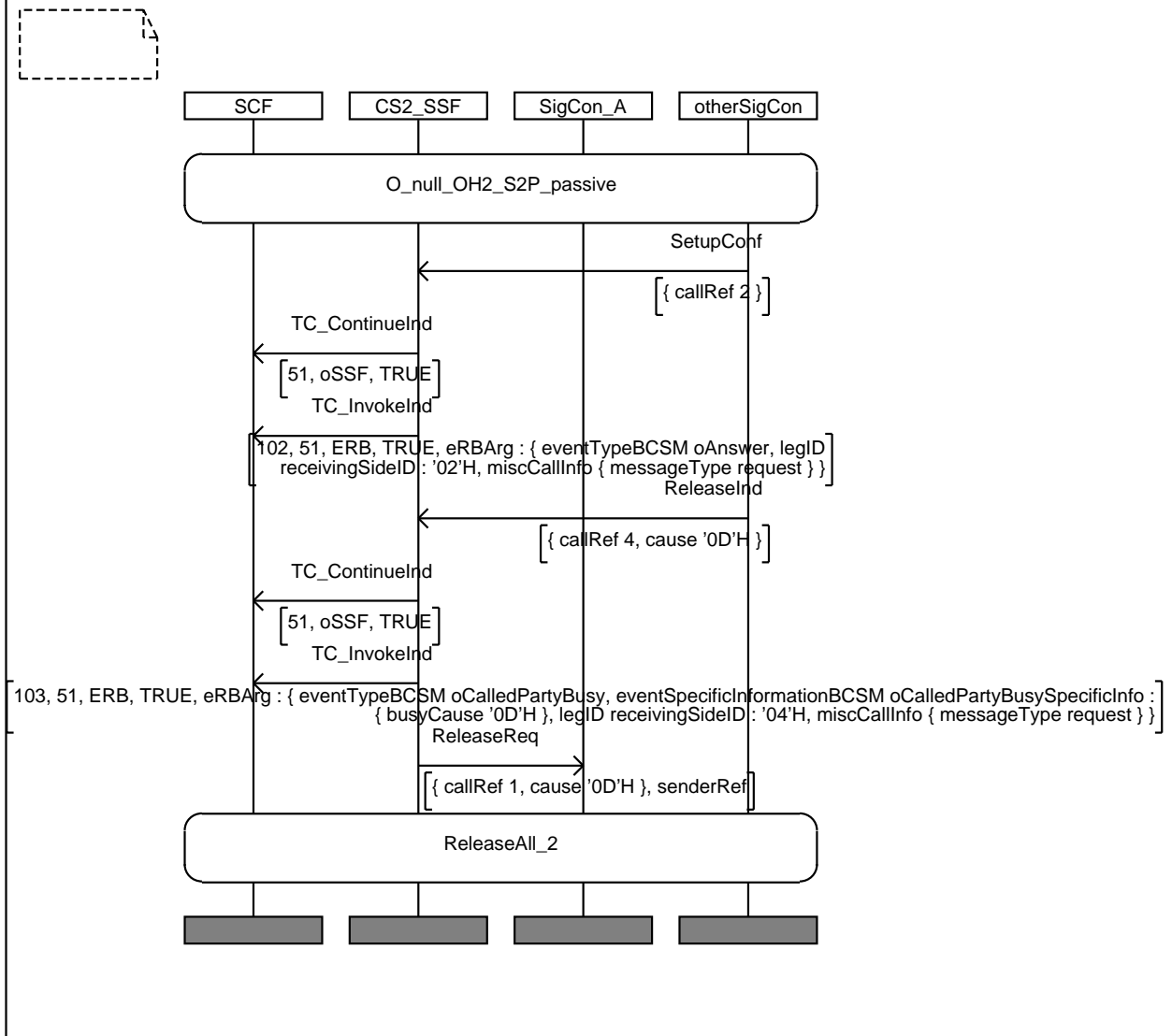


IN2\_A\_CPH\_081

This test purpose is not included.

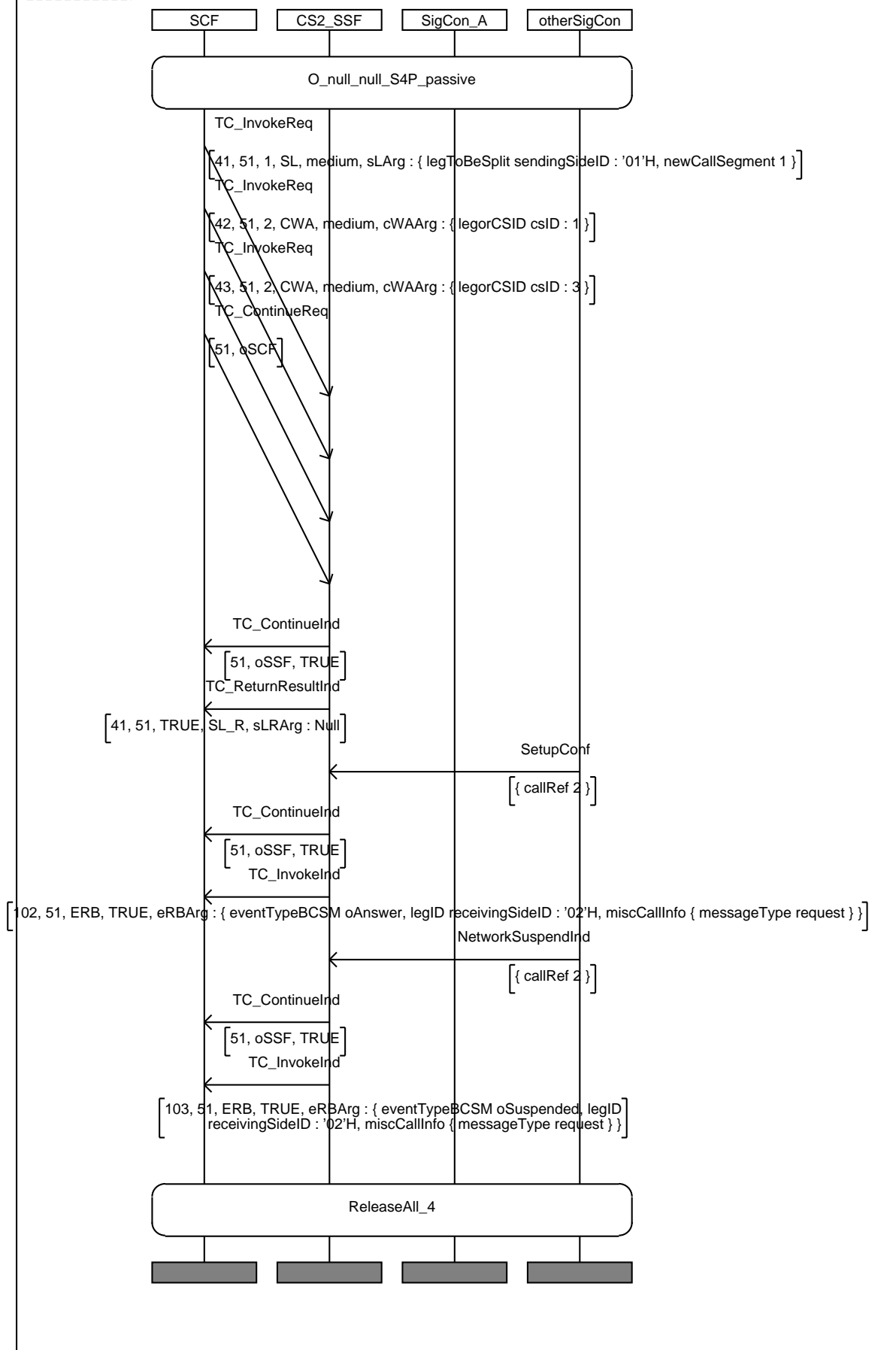
IN2_A_CPH_082	
<b>Purpose:</b>	Check that after two events are armed, then both are reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_OH(2)_S2P_passive
<b>Test description</b>	CP1_2! SetupConf CP1_4! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(2,notify, oAnswer) L1? ERB(4,notify, oBusy)
<b>Postamble:</b>	ReleaseAll_2

## MSC IN2\_A\_CPH\_082



<b>IN2_A_CPH_083</b>	
<b>Purpose:</b>	Test that after a Split leg operation is sent the armed events remain the same
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	O_null_null_S4P_passive
<b>Test description</b>	L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=3) CP1_2! SetupConf CP1_2! NetworkSuspendInd
<b>Pass criteria</b>	L1? ERB(2, notify, oAnswer) L1? ERB(2, notify, oSuspend)
<b>Postamble:</b>	ReleaseAll_4

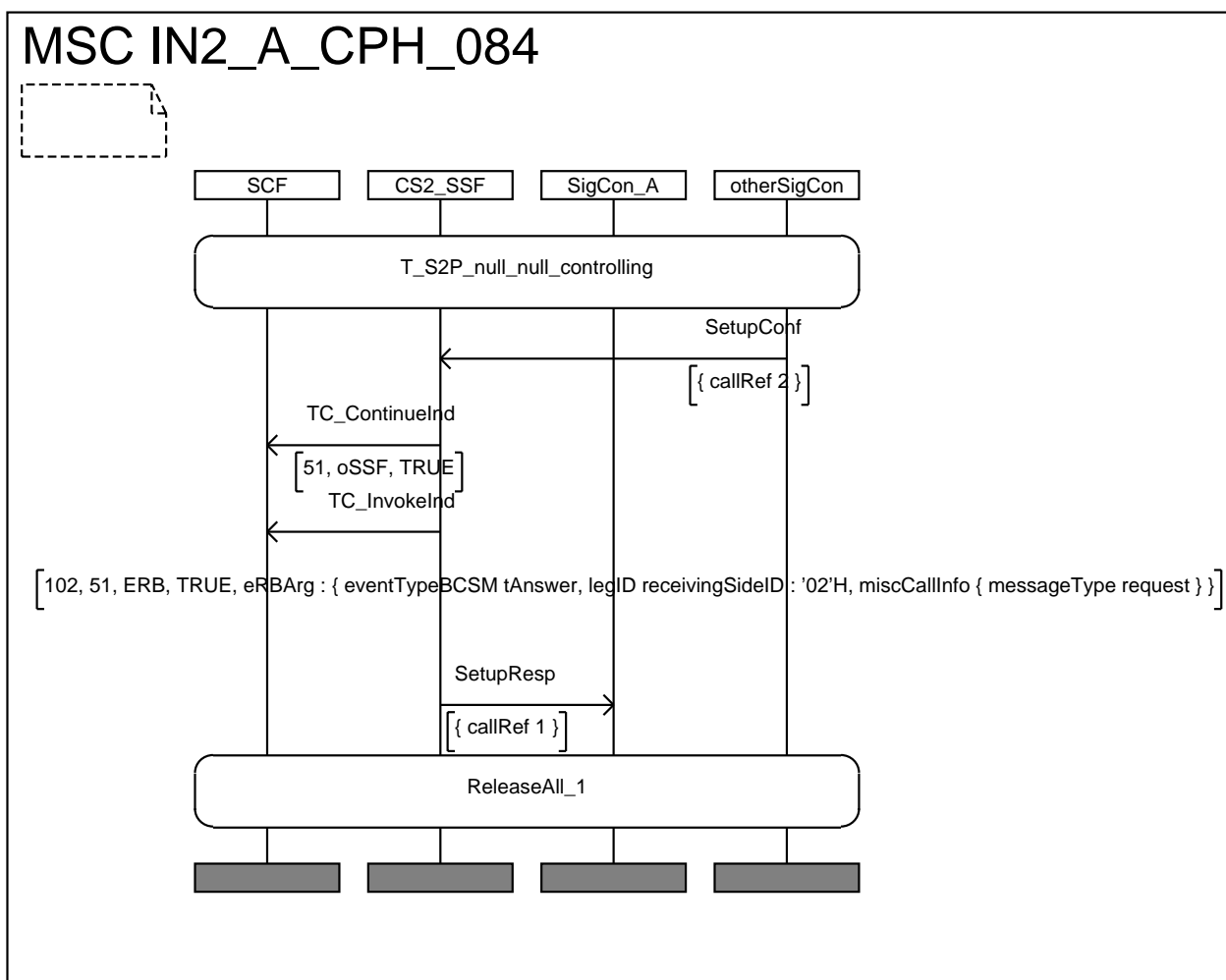
# MSC IN2\_A\_CPH\_083



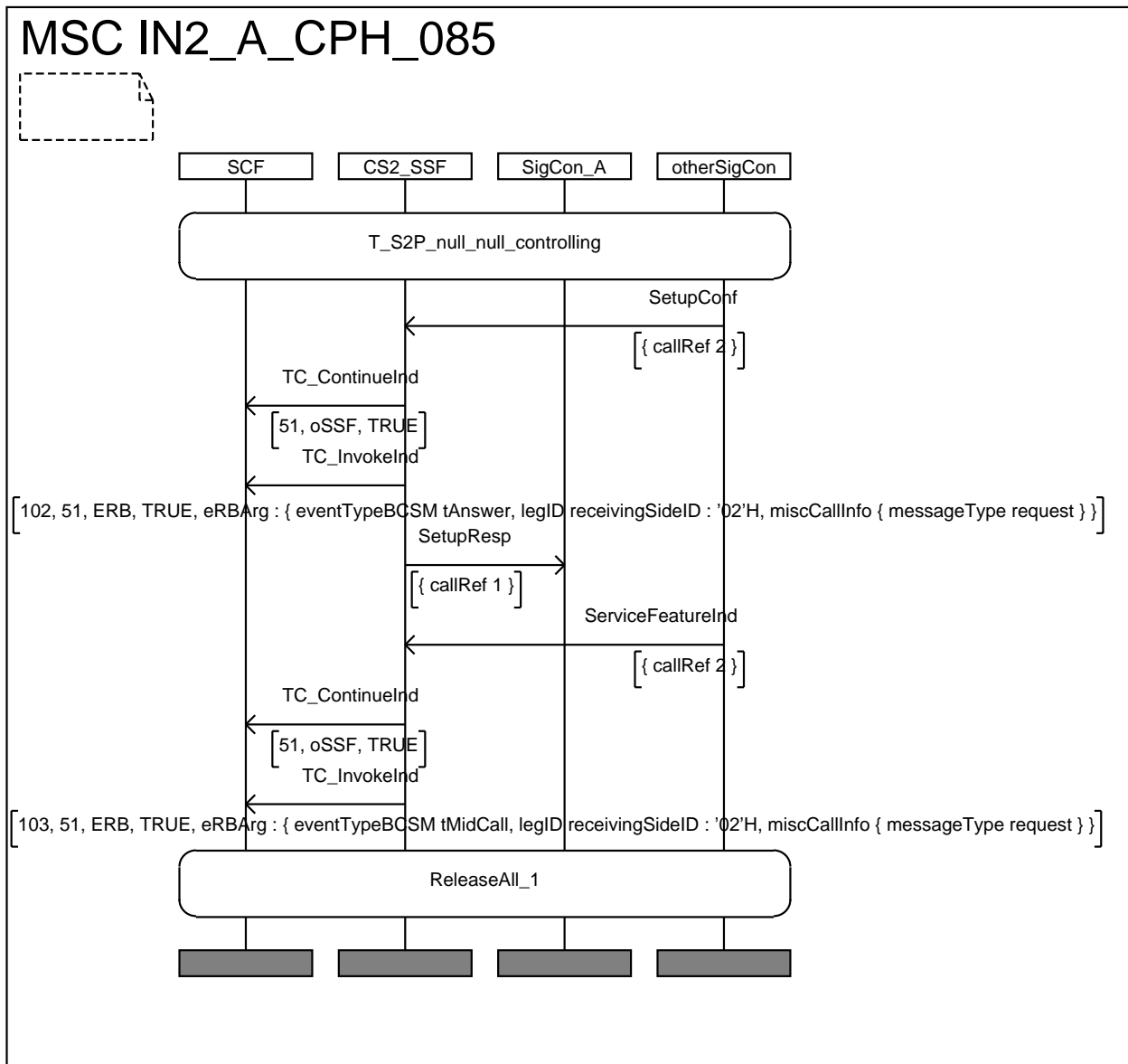
## 7.6.2 Terminating (T) trigger

### 7.6.2.1 T\_1 Events coming from the controlling leg (legId=2),

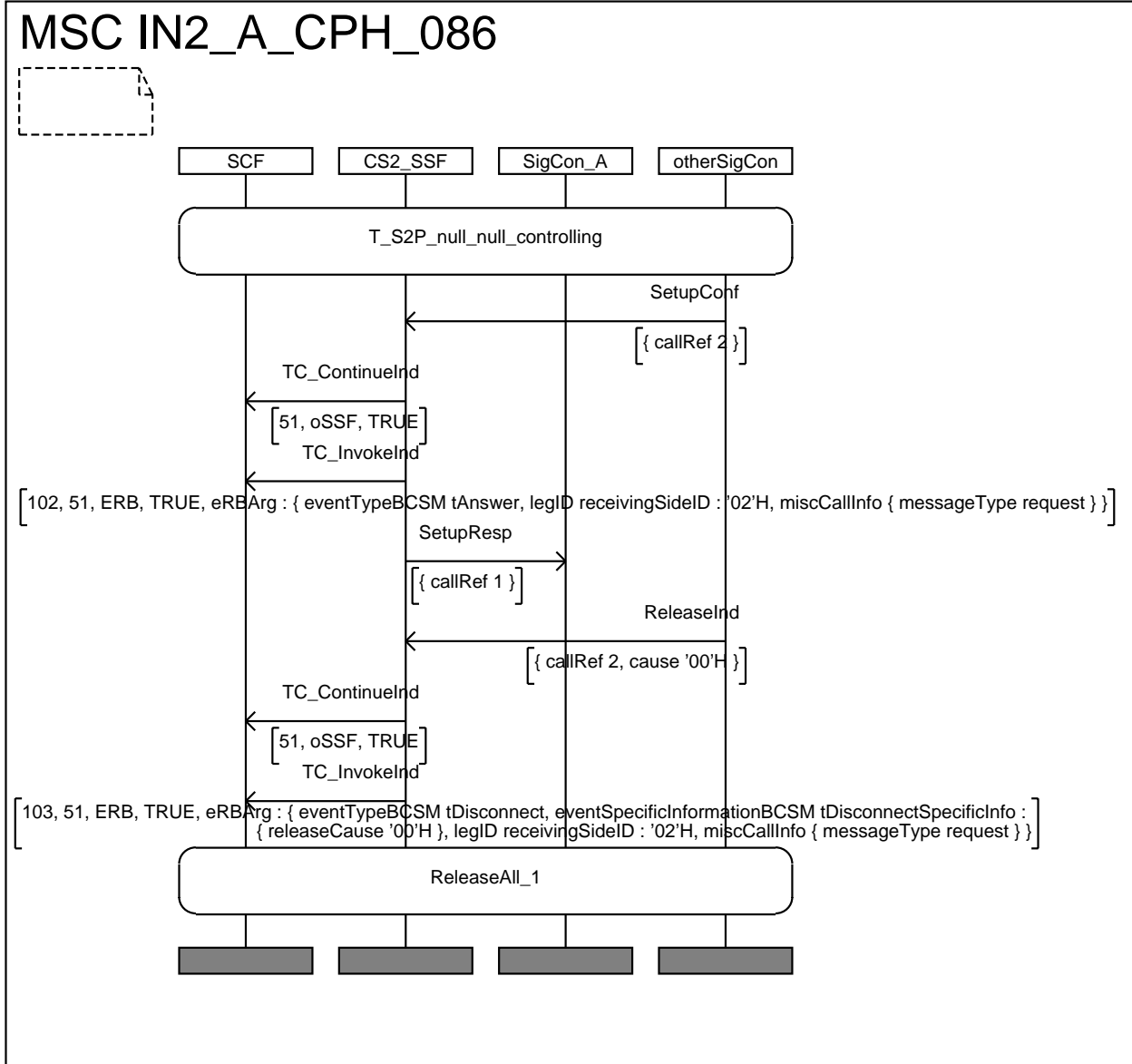
IN2_A_CPH_084	
<b>Purpose:</b>	Check that an event coming from the controlling leg is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_controlling
<b>Test description</b>	CP1_2! SetupConf
<b>Pass criteria</b>	L1?: ERB (tAnswer) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_1



IN2_A_CPH_085	
<b>Purpose:</b>	Check that an event coming from the controlling leg is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_controlling
<b>Test description</b>	CP1_2!:SetupConf CP1_2!:ServiceFeatureIndication
<b>Pass criteria</b>	L1?: ERB (tAnswer) L1?: ERB (tMidCall)
<b>Postamble:</b>	ReleaseAll_1



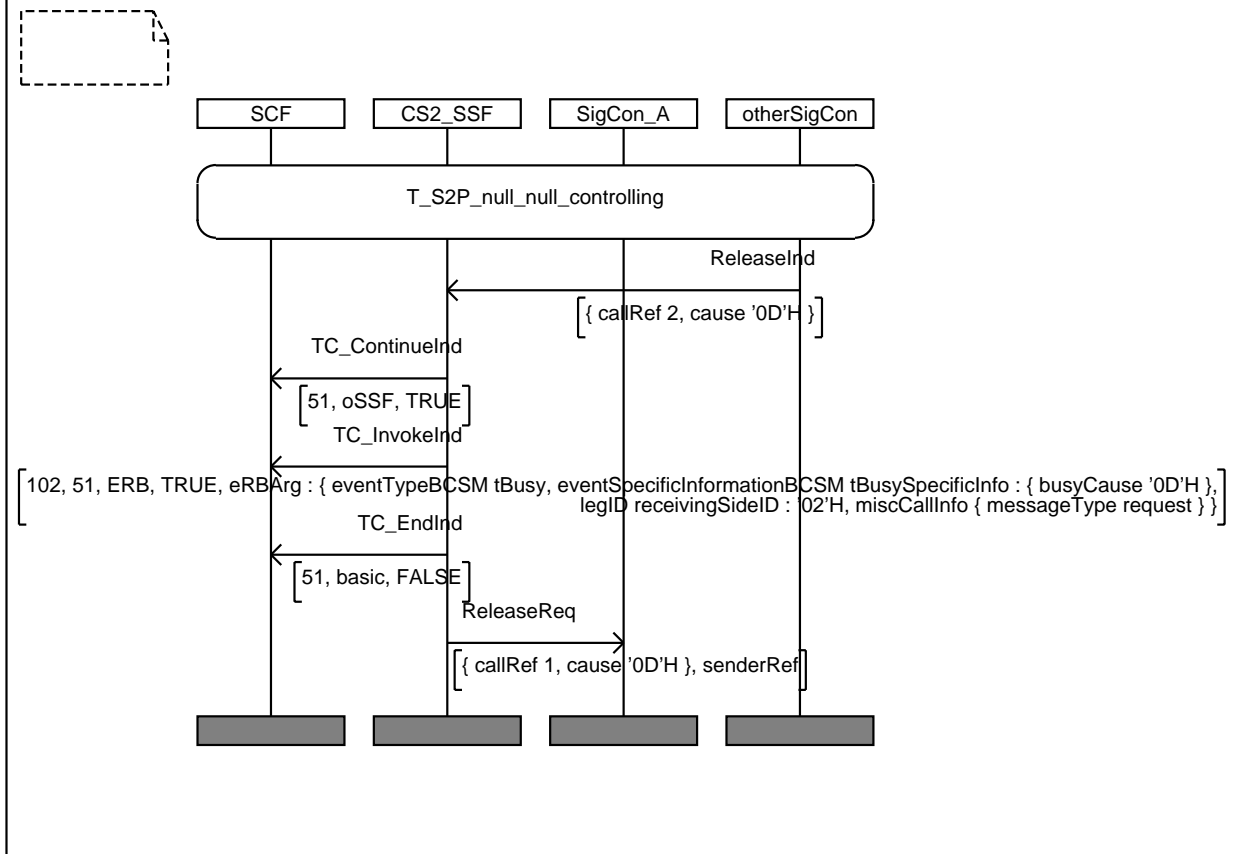
IN2_A_CPH_086	
<b>Purpose:</b>	Check that an event coming from the controlling leg is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_controlling
<b>Test description</b>	CP1_2!: SetupConf CP1_2: ReleaseInd
<b>Pass criteria</b>	L1?: ERB (tAnswer) L1?: ERB(tDisconnect)
<b>Postamble:</b>	ReleaseAll_1





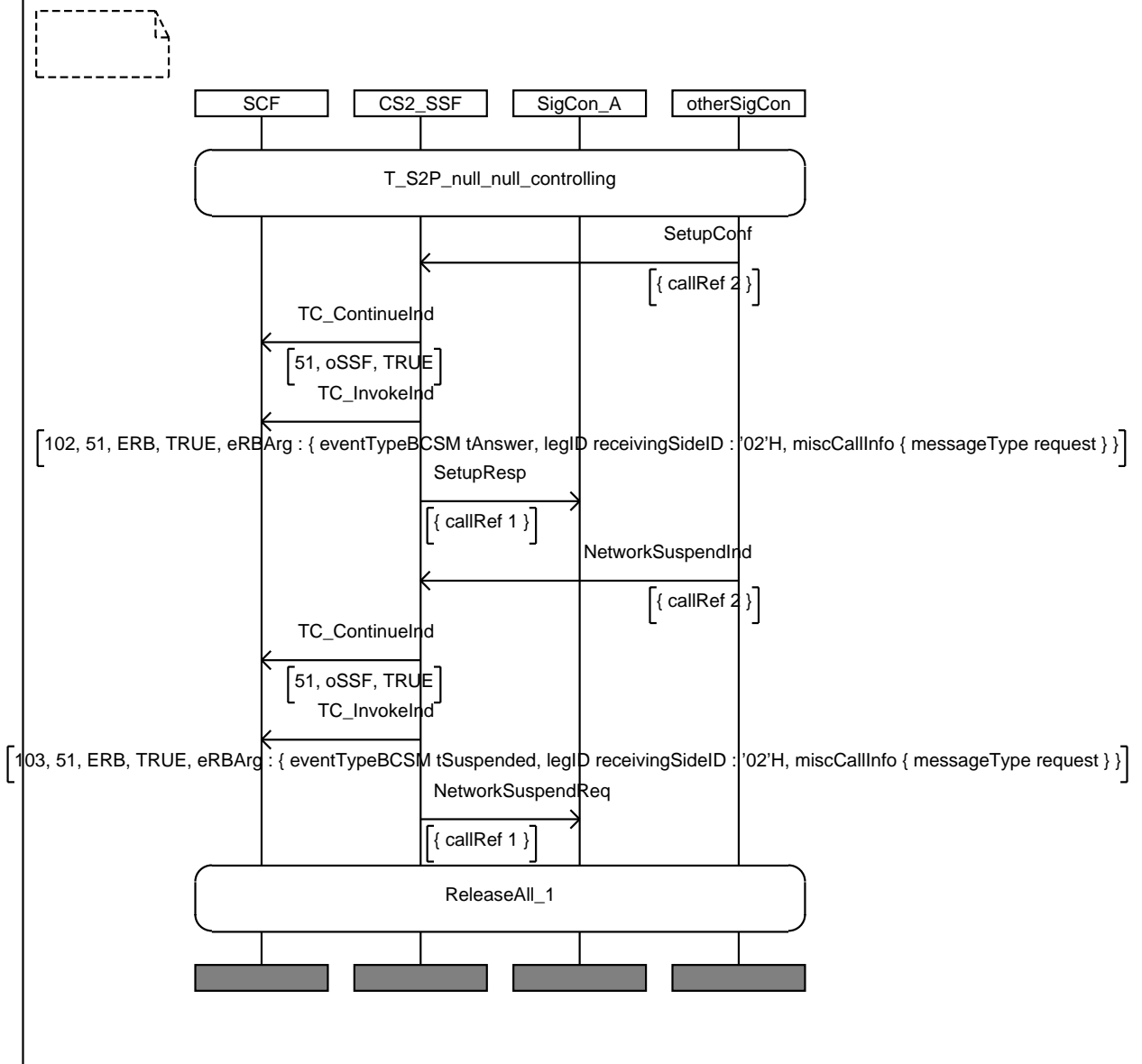
IN2_A_CPH_087	
<b>Purpose:</b>	Check that an event coming from the controlling leg is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_controlling
<b>Test description</b>	CP1_2!: ReleaseIndication(busy)
<b>Pass criteria</b>	L1?: ERB (tBusy))
<b>Postamble:</b>	none

### MSC IN2\_A\_CPH\_087

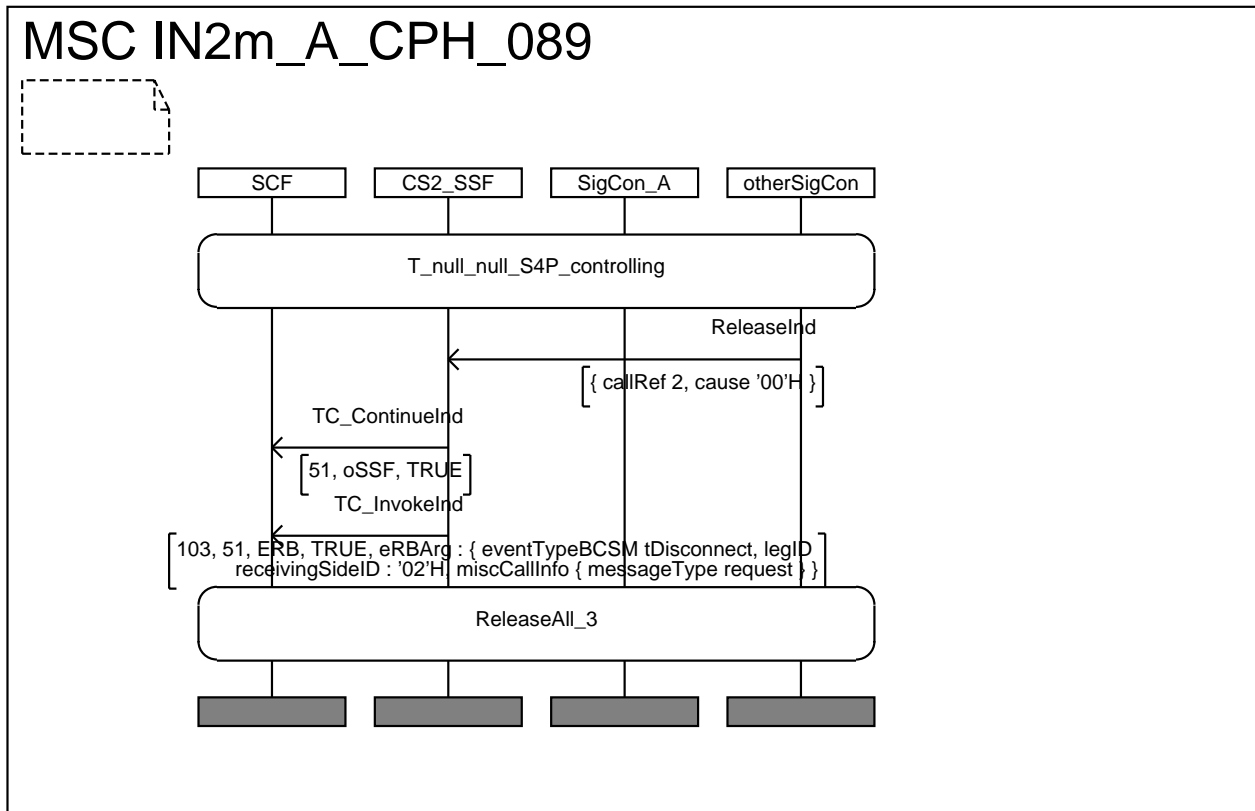


IN2_A_CPH_088	
<b>Purpose:</b>	Check that an event coming from the controlling leg is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_controlling
<b>Test description</b>	CP1_2!: SetupConf CP1_2!: SuspendIndication
<b>Pass criteria</b>	L1?: ERB (tAnswer) L1?: ERB(tSuspended)
<b>Postamble:</b>	ReleaseAll_1

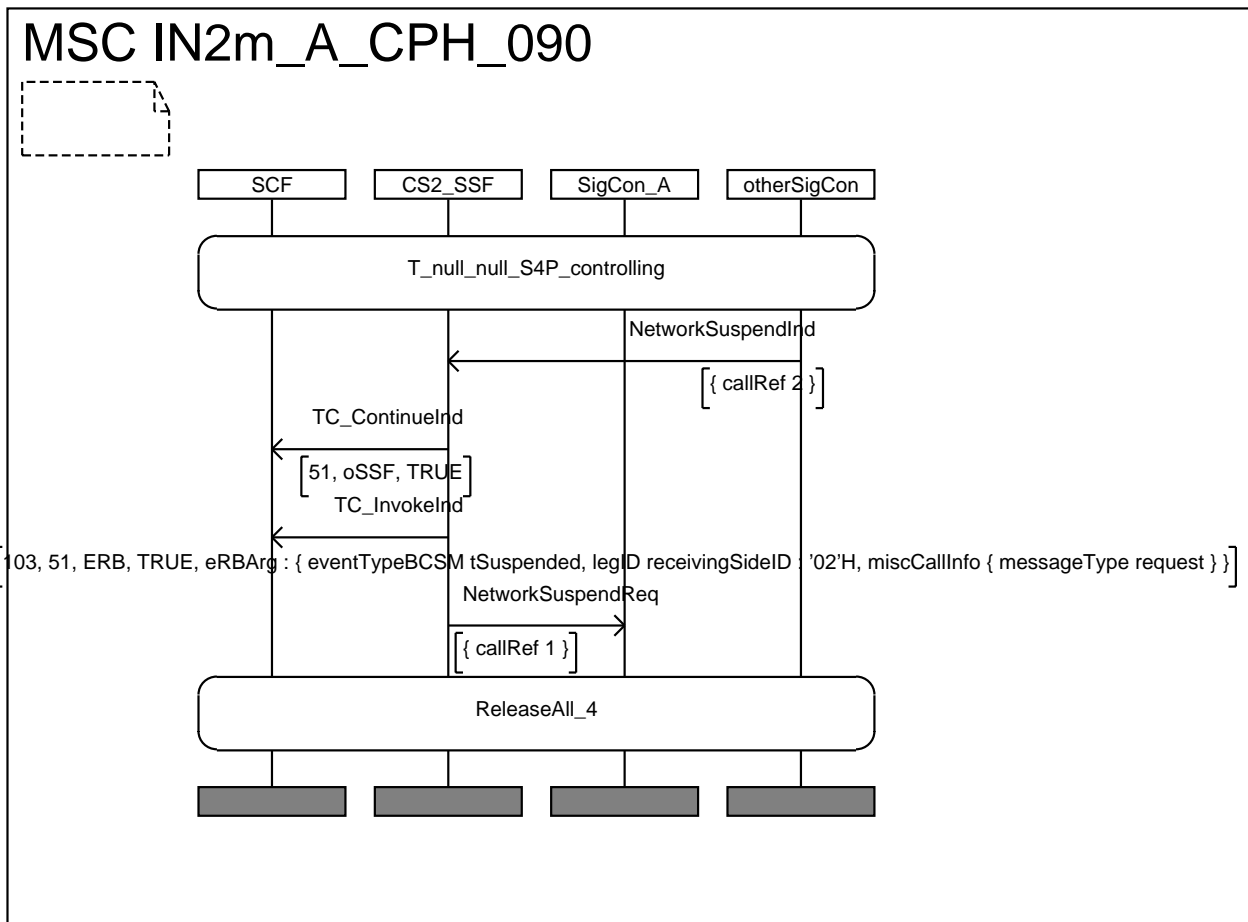
### MSC IN2\_A\_CPH\_088



IN2_A_CPH_089	
<b>Purpose:</b>	Check that event filtering rules are applied
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_null_S4P_controlling
<b>Test description</b>	CP1_2! ReleaseInd
<b>Pass criteria</b>	L1?: ERB (tDisconnect) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_3



IN2_A_CPH_090	
<b>Purpose:</b>	Check that event filtering rules are applied
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_null_S4P_controlling
<b>Test description</b>	CP1_2!: SuspendIndication
<b>Pass criteria</b>	L1?: ERB (tSuspended) is received only once (filtering rules apply)
<b>Postamble:</b>	ReleaseAll_4

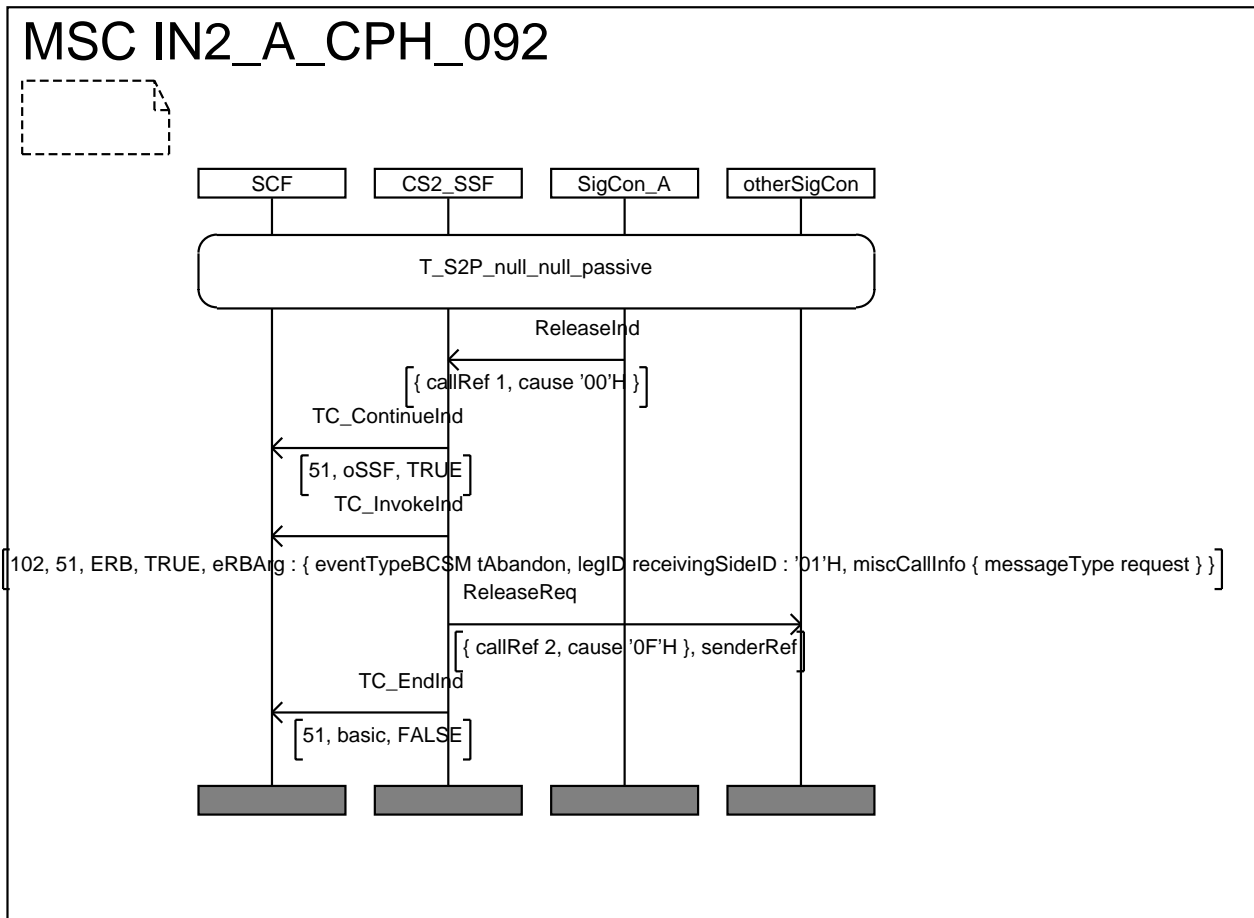


IN2_A_CPH_091
---------------

This test purpose is not included.

7.6.2.2 T\_2 Events coming from passive legs (legId= 2,3...)

IN2_A_CPH_092	
<b>Purpose:</b>	Check that a single event tAbandon is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_passive
<b>Test description</b>	CP1_1! ReleaseInd
<b>Pass criteria</b>	L1? ERB(1,interrupted, tAbandon)
<b>Postamble:</b>	none

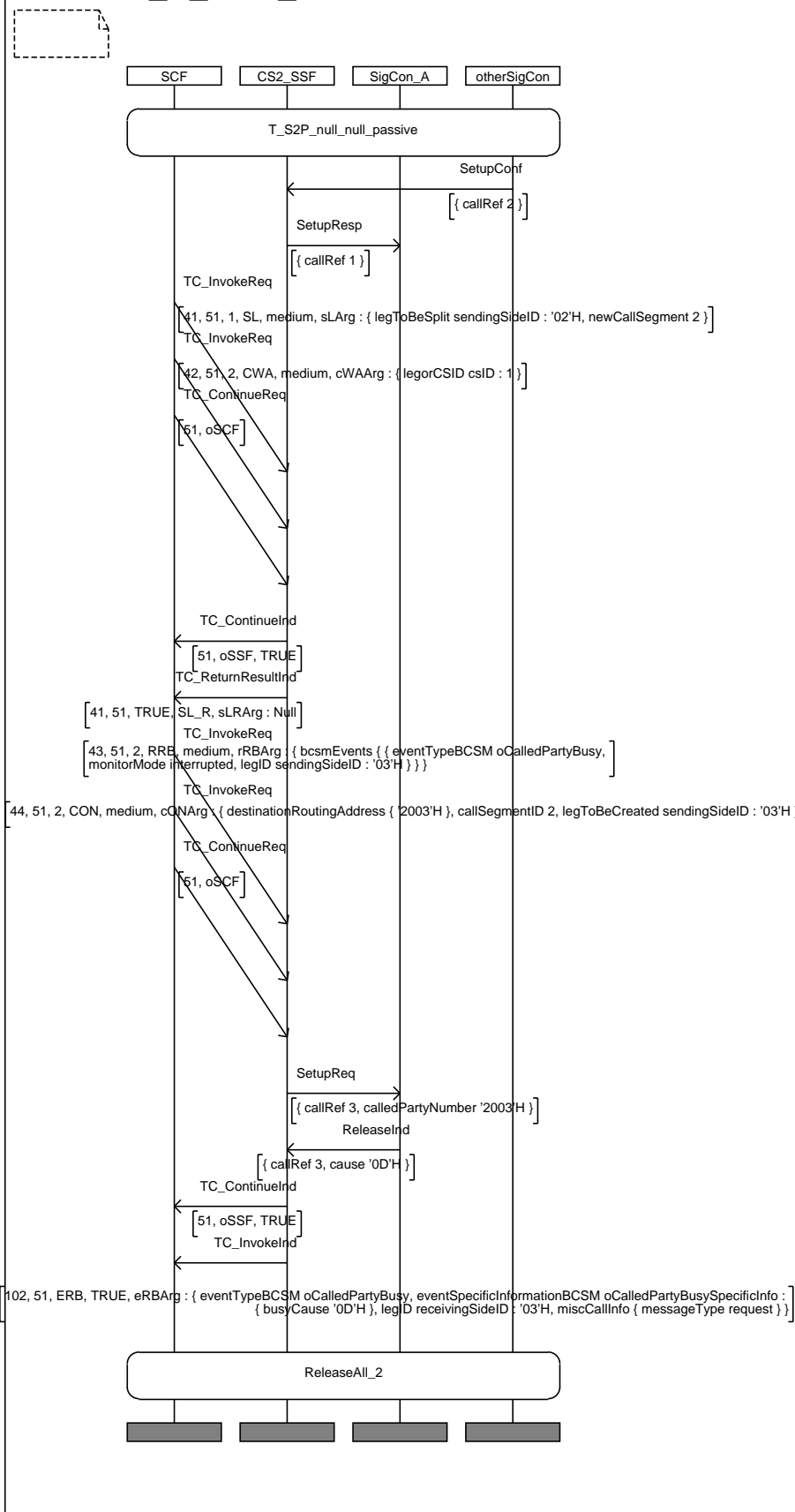


IN2_A_CPH_093
---------------

This test purpose is not included.

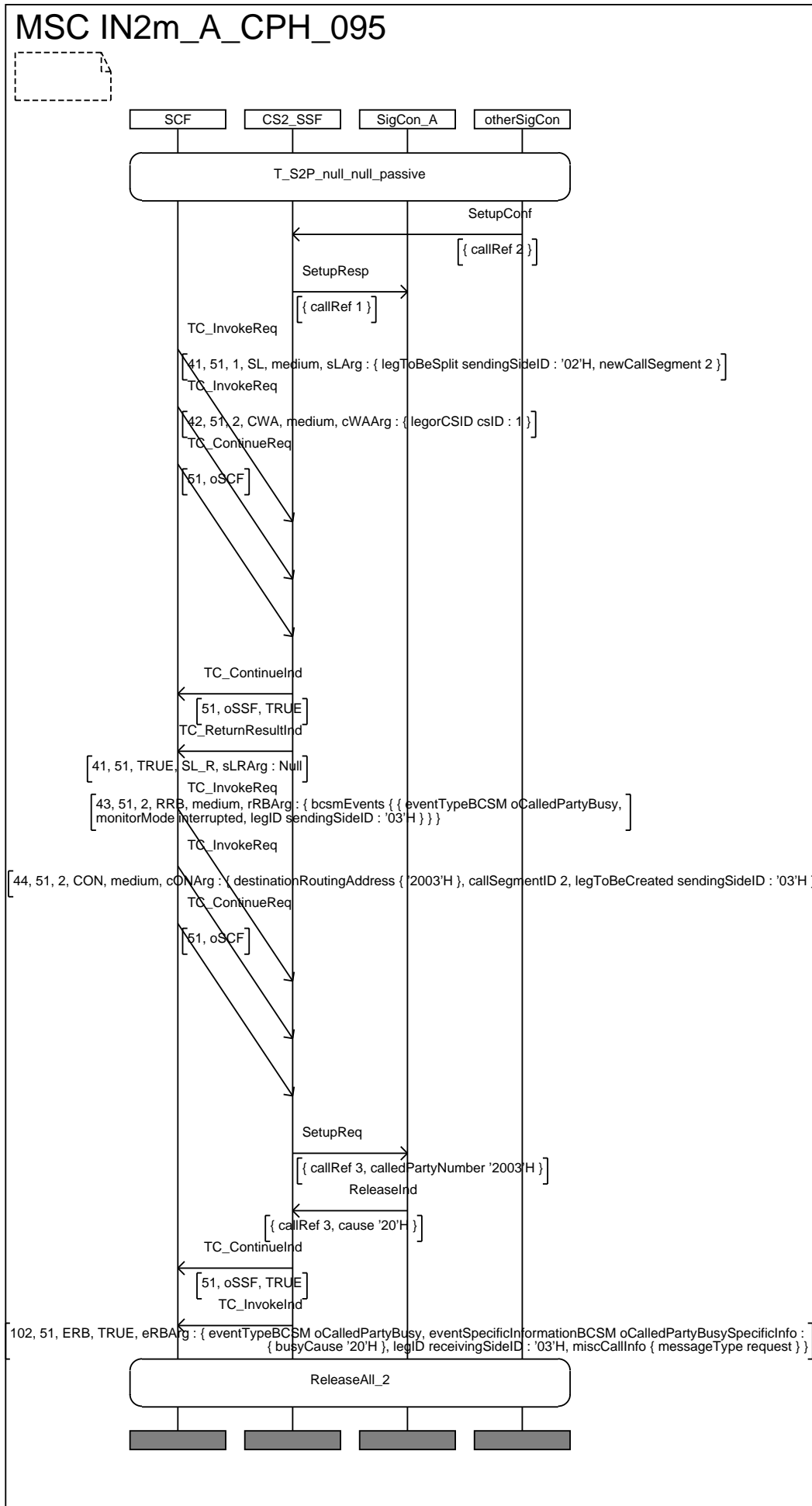
<b>IN2_A_CPH_094</b>	
<b>Purpose:</b>	Check that a single event oCalledpartyBusy is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_passive
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oCalledPartyBusy) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! ReleaseInd(Busy cause)
<b>Pass criteria</b>	L1? ERB(3,interrupted, oCalledPartyBusy)
<b>Postamble:</b>	ReleaseAll_2

MSC IN2\_A\_CPH\_094

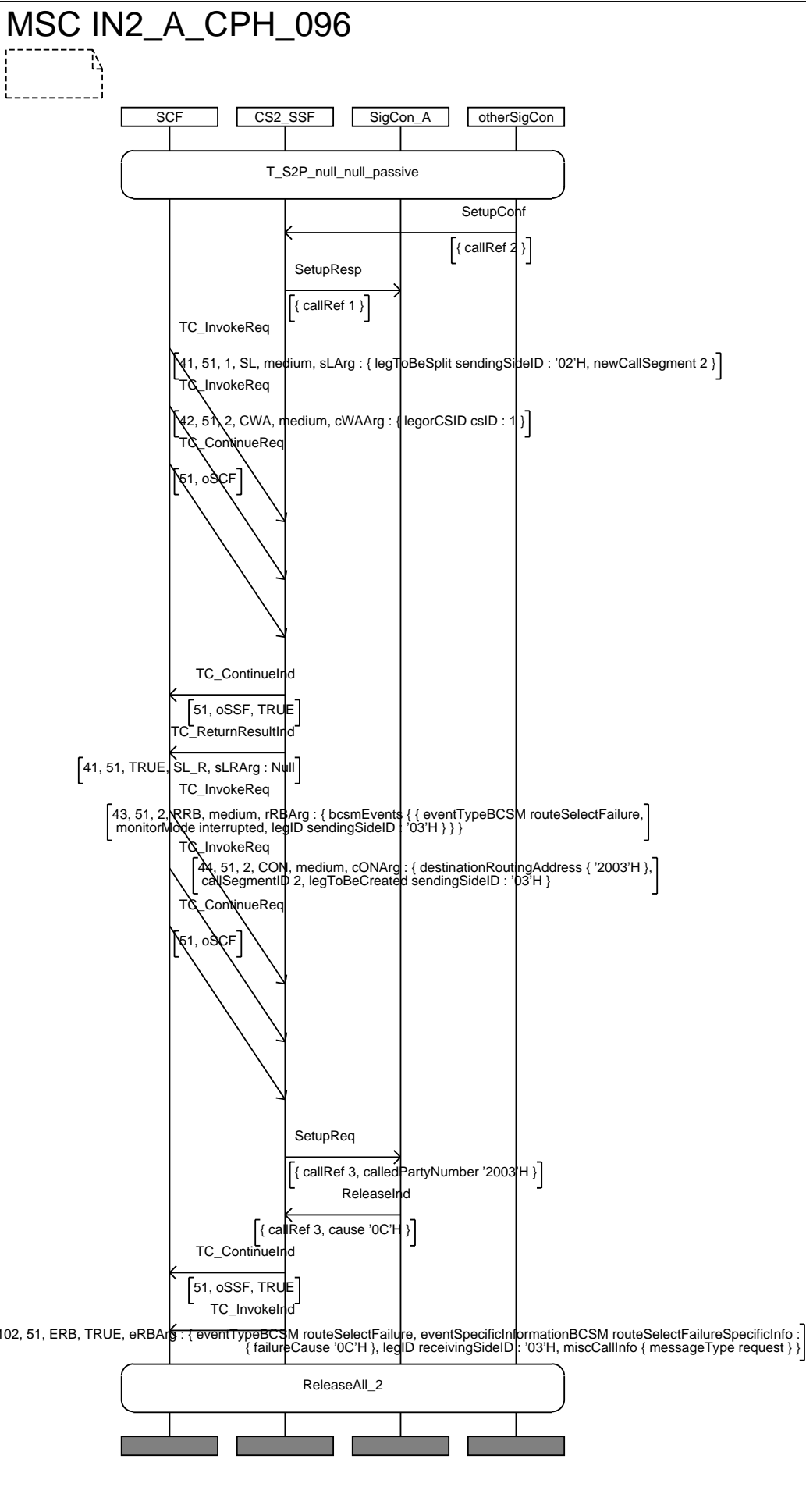


<b>IN2_A_CPH_095</b>	
<b>Purpose:</b>	Check that a single event oCalledPartyBusy is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_passive
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oCalledPartyBusy) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! ReleaseInd(cause)
<b>Pass criteria</b>	L1? ERB(3,interrupted, oCalledPartyBusy, cause)
<b>Postamble:</b>	ReleaseAll_2



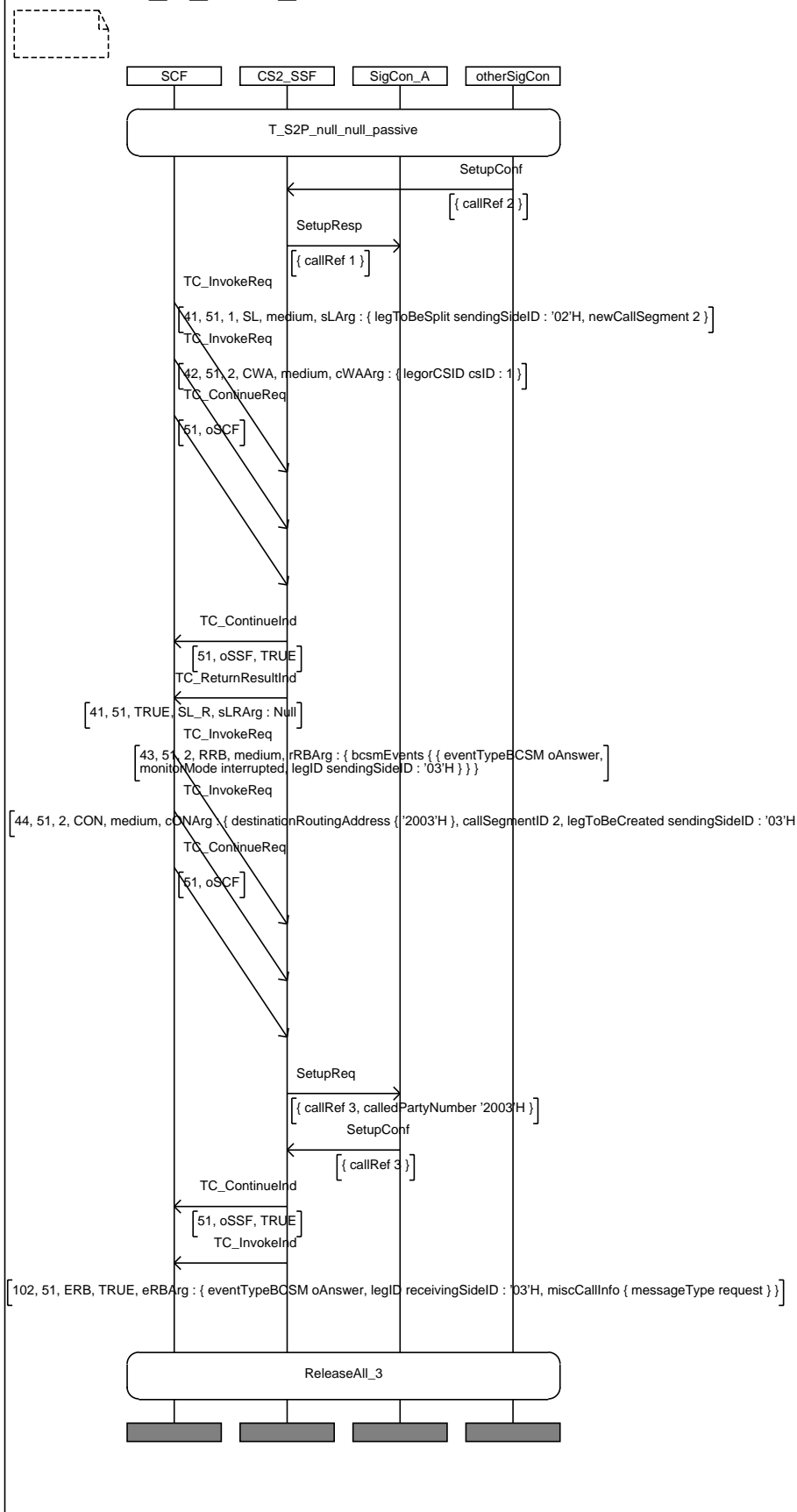


<b>IN2_A_CPH_096</b>	
<b>Purpose:</b>	Check that a single event routeSelectFailure is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_passive
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oRouteSelectFailure) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! ReleaseInd(RouteSelectFailure cause)
<b>Pass criteria</b>	L1? ERB(3,interrupted, oRouteSelectFailure)
<b>Postamble:</b>	ReleaseAll_2



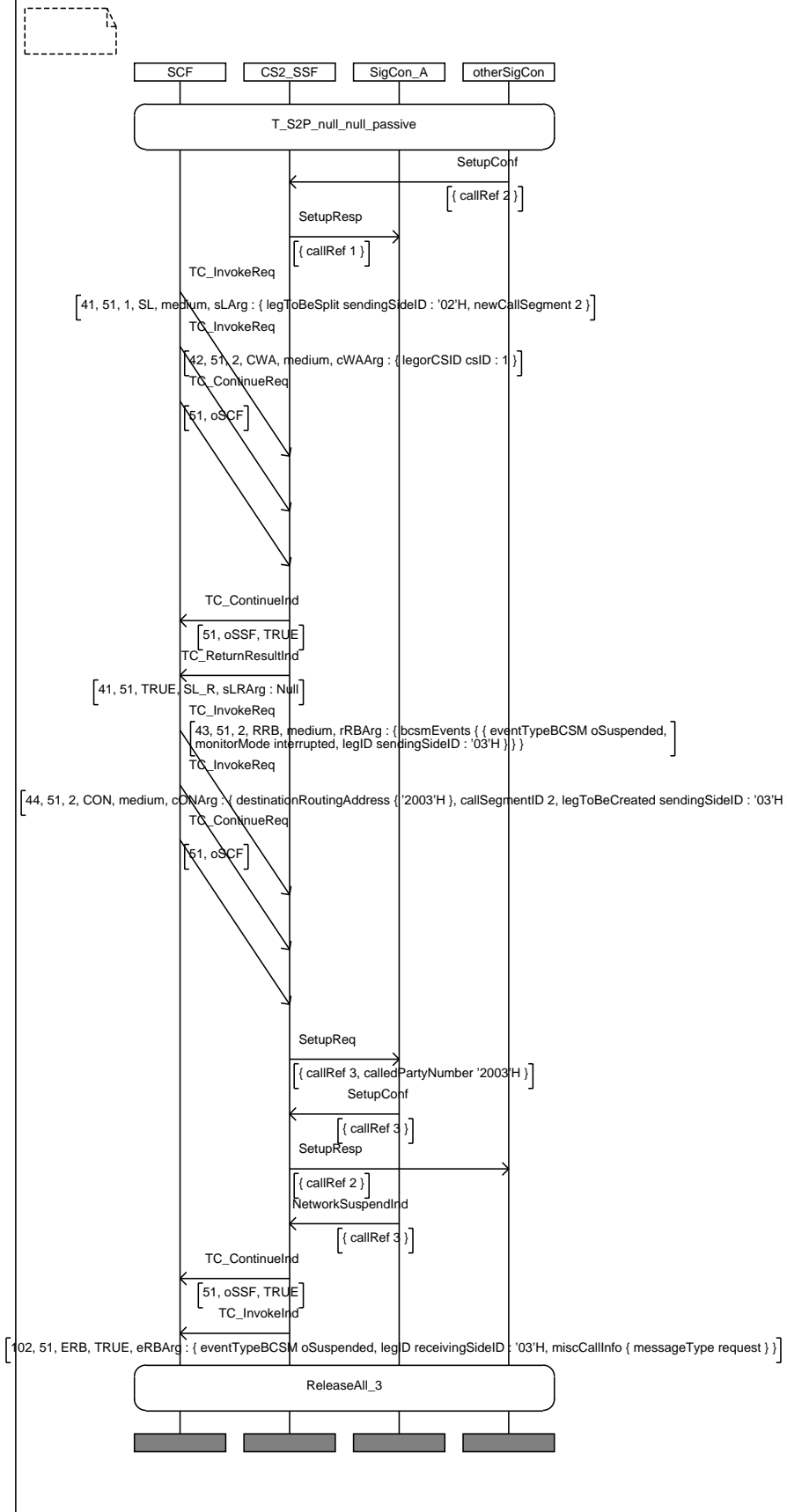
<b>IN2_A_CPH_097</b>	
<b>Purpose:</b>	Check that a single event oAnswer is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_passive
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oAnswer) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! SetUpConf
<b>Pass criteria</b>	L1? ERB(3,interrupted, oAnswer)
<b>Postamble:</b>	ReleaseAll_3

MSC IN2\_A\_CPH\_097



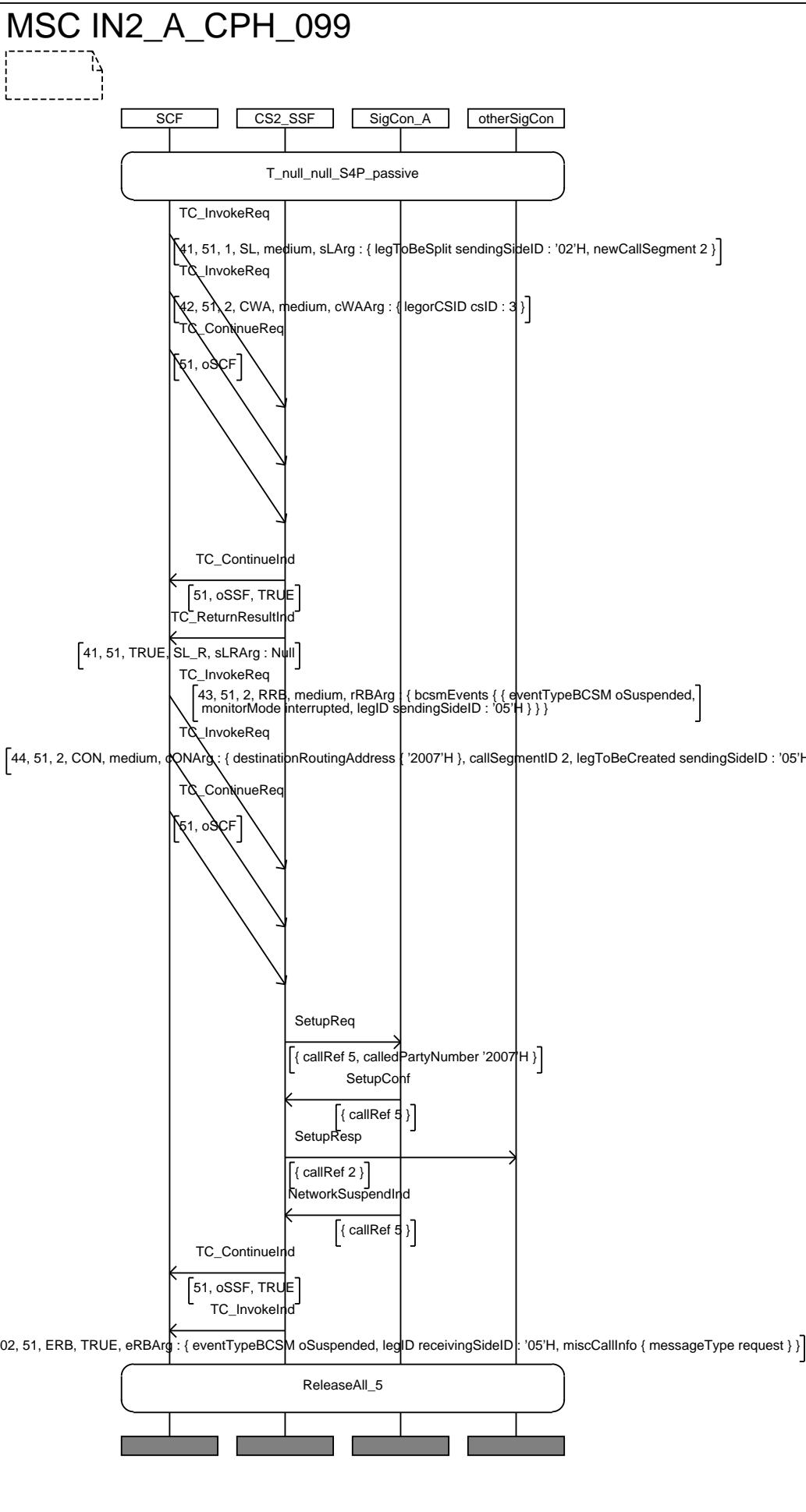
<b>IN2_A_CPH_098</b>	
<b>Purpose:</b>	Check that a single event oSuspended is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_S2P_null_null_passive
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oSuspend) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! SetUpConf CP1_3! NetworkSuspendInd
<b>Pass criteria</b>	L1? ERB(3,interrupted, oSuspend)
<b>Postamble:</b>	ReleaseAll_3

MSC IN2\_A\_CPH\_098



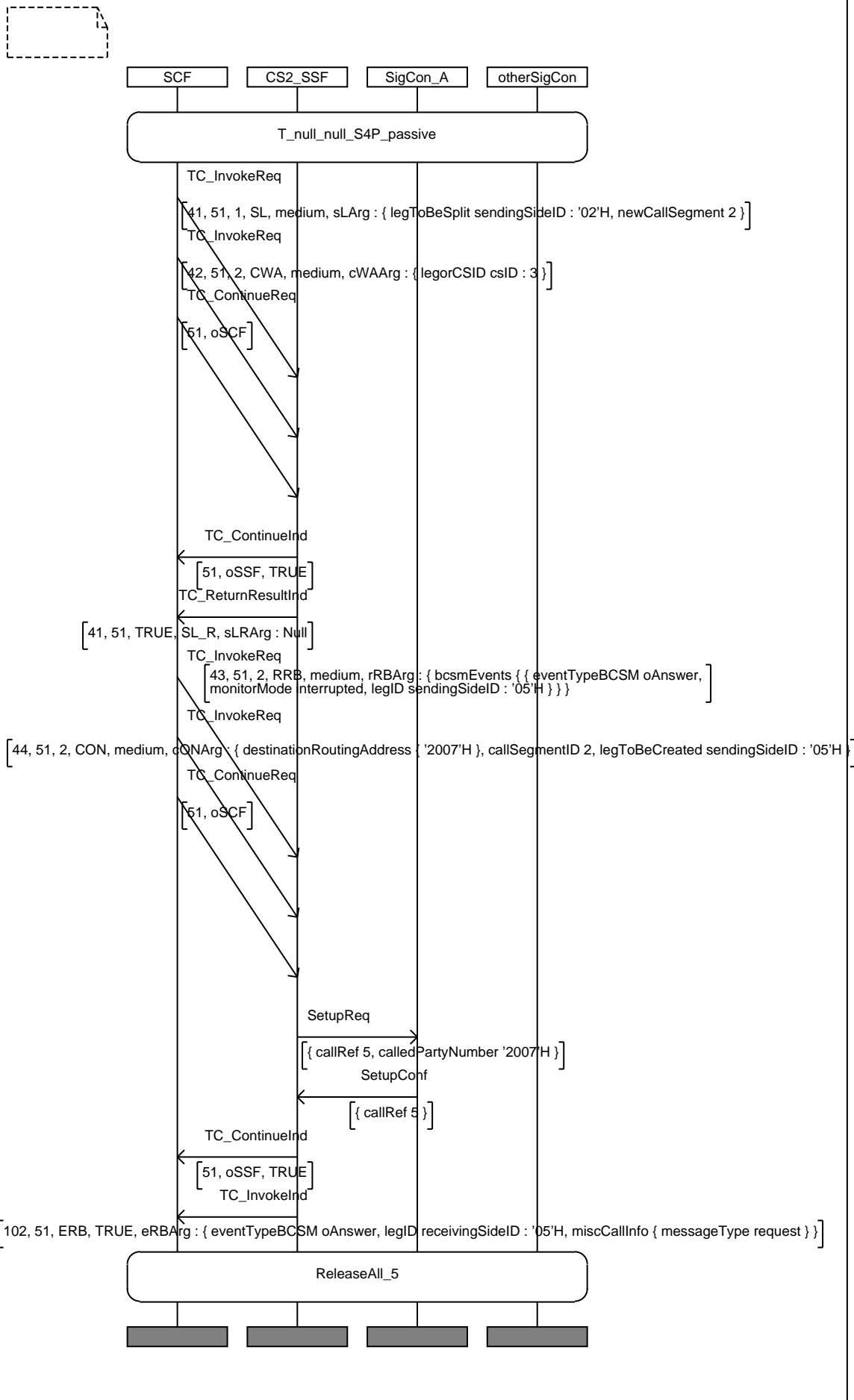
<b>IN2_A_CPH_099</b>	
<b>Purpose:</b>	Check that a single event oSuspended is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_null_S4P
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RRB(5, interrupted, oSuspend) L1! Connect(5,2) CP1_5? SetUpReq CP1_5! SetUpConf CP1_5! NetworkSuspendInd
<b>Pass criteria</b>	L1? ERB(5,interrupted, oSuspend)
<b>Postamble:</b>	ReleaseAll_5





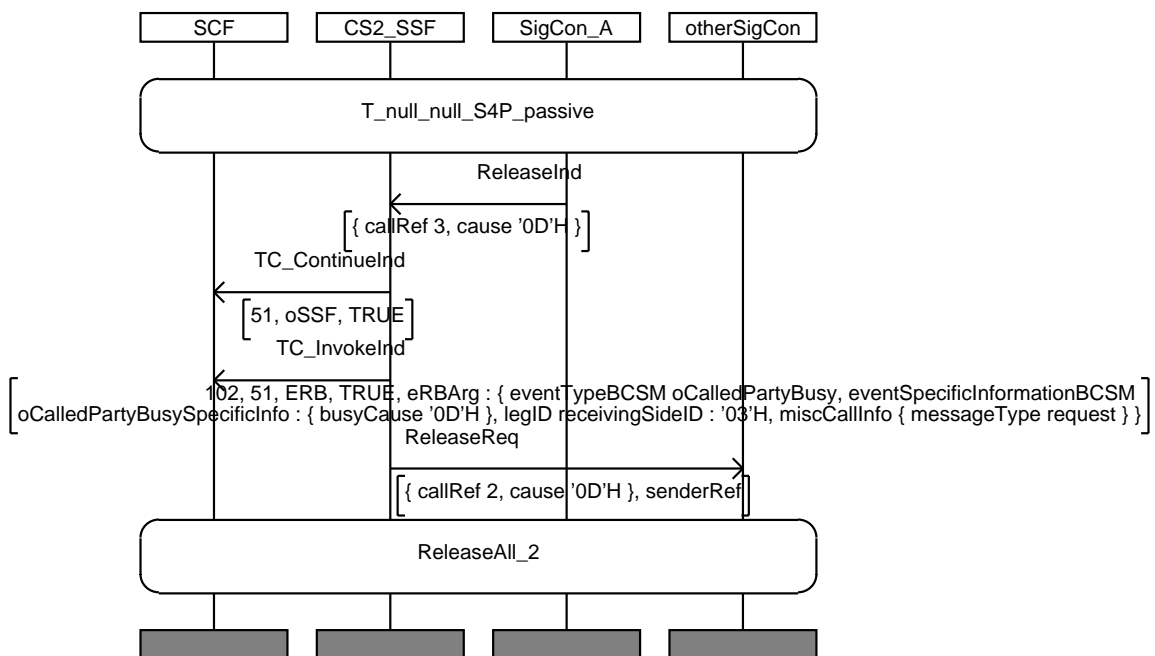
<b>IN2_A_CPH_100</b>	
<b>Purpose:</b>	Check that a single event oAnswer is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_null_S4P
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RRB(5, interrupted, oAnswer) L1! Connect(5,2) CP1_5? SetUpReq CP1_5! SetUpConf
<b>Pass criteria</b>	L1? ERB(5,interrupted, oAnswer)
<b>Postamble:</b>	ReleaseAll_5

# MSC IN2\_A\_CPH\_100

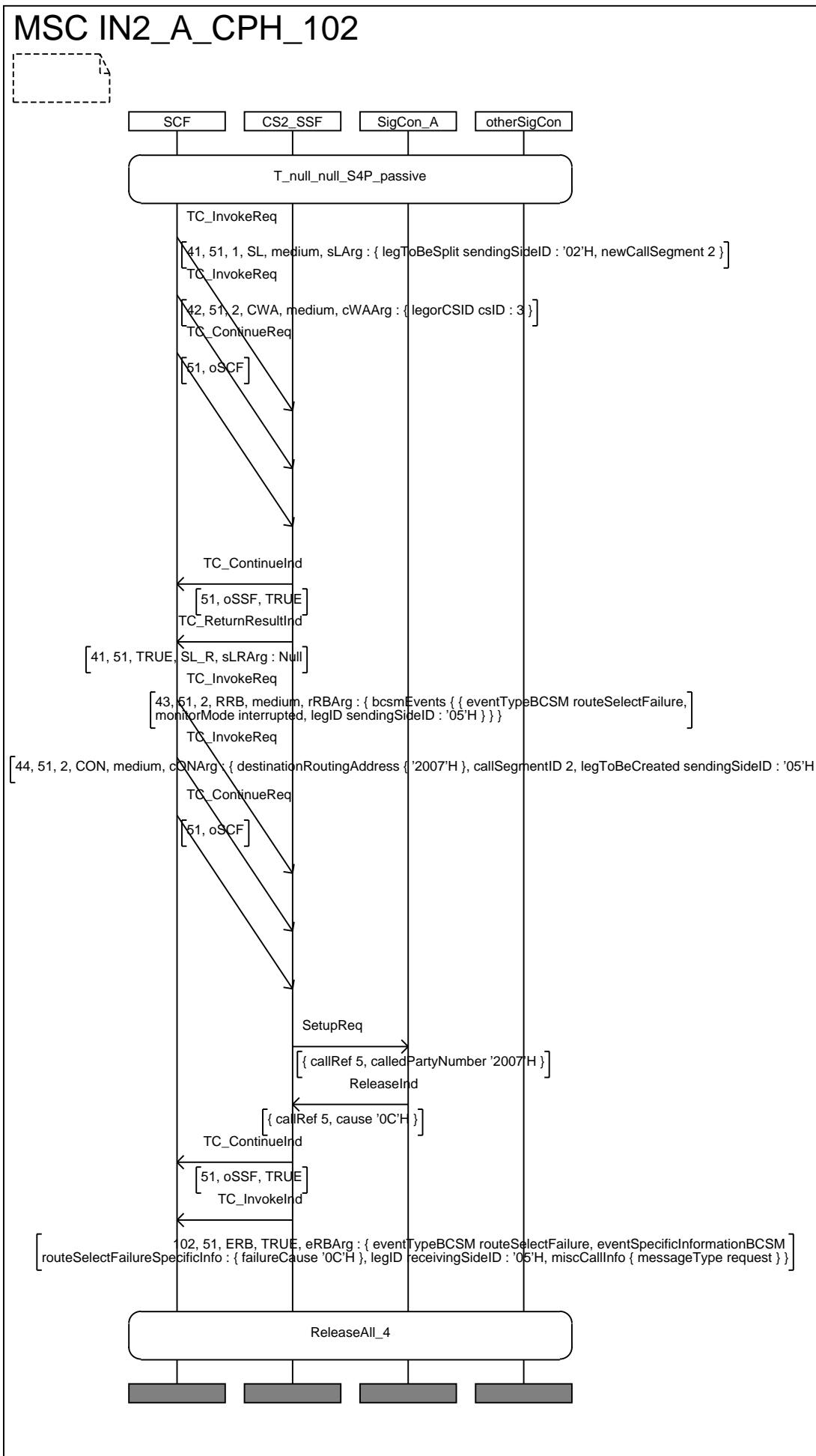


IN2_A_CPH_101	
<b>Purpose:</b>	Check that a single event oCalledPartyBusy is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_null_S4P_passive
<b>Test description</b>	CP1_3! ReleaseInd (Busy cause)
<b>Pass criteria</b>	L1? ERB(3, notify, oCalledPartyBusy)
<b>Postamble:</b>	ReleaseAll_2

### MSC IN2\_A\_CPH\_101



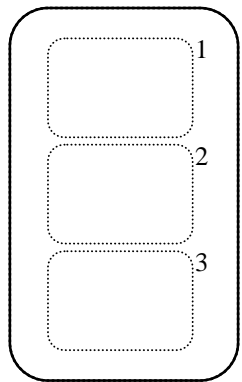
<b>IN2_A_CPH_102</b>	
<b>Purpose:</b>	Check that a single event routeSelectFailure is reported
<b>Requirement ref</b>	
<b>Selection Cond.</b>	
<b>Preamble:</b>	T_null_null_S4P
<b>Test description</b>	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RRB(5, interrupted, routeSelectFailure) L1! Connect(5,2) CP1_5?SetupReq CP1_5! ReleaseInd (routeSelectFailure cause)
	L1? ERB(5, interrupted, routeSelectFailure)
<b>Postamble:</b>	ReleaseAll_4



---

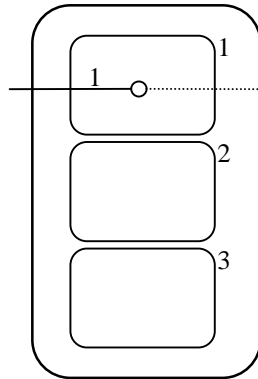
## Annex A (normative): Preamble trees

### **Terminating Preamble (1)**



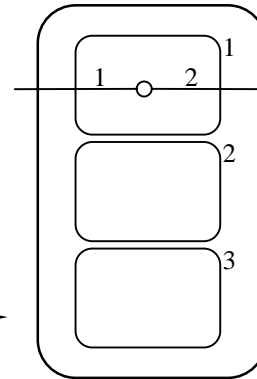
**T\_null\_null\_null**

CP1-1? SetupInd  
L1? IDP(termAttemptAuthorized)



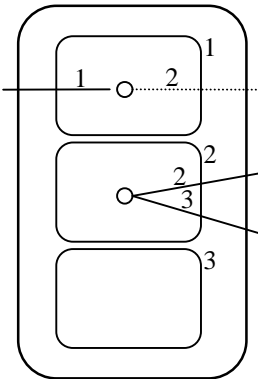
**T\_TS\_null\_null**

L1! RRB(1,tDisconnect)  
L1! CWA(csId=1)  
CP1-2? SetupReq



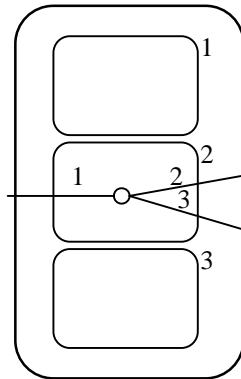
**T\_S2P\_null\_null**

L1! SL(2,2)  
L1? SL\_R  
L1! RRB(3,tDisconnect)



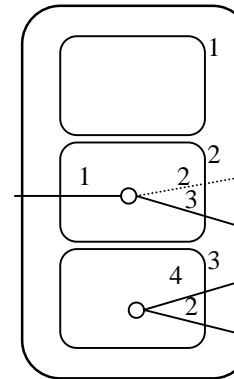
**T\_OH1\_S2P\_null**

CP1-2! SetupConf  
L1! MC(1,2)



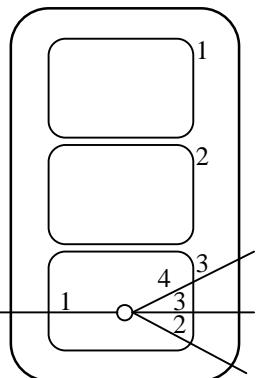
**T\_null\_S3P\_null**

L1! SL(2,3)  
L1? SL\_R  
L1! RRB(4,tDisconnect)



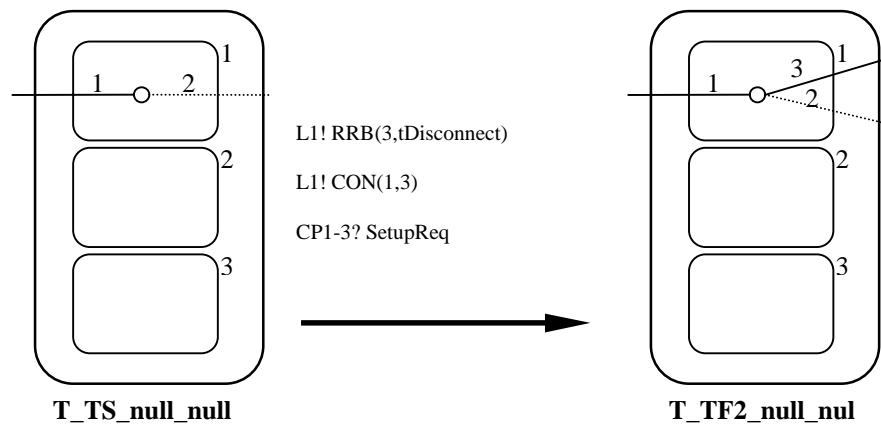
**T\_null\_OH2\_S2P**

L1! MC(2,3)  
L1? MC\_R



**T\_null\_null\_S4P**



**Terminating Preamble (2)**

## Annex B (normative): TCAP Parameter values

**Table B.1: Parameter values**

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

For the purpose of documenting the MSCs, specially for complex configuration involved in CPH, the following values are used, in order to use unique values for every Id:

### INVOKE ID:

- In O\_PREAMBLES: from 1 to 39;
- In T\_PREAMBLES: from 1 to 39;
- In I\_PREAMBLES: from 1 to 39;
- In TEST PURPOSES: from 41 to 90;
- In POSTAMBLES: from 91 to 100.

### DIALOG ID:

- When using O\_PREAMBLES: value 51;
- When using T\_PREAMBLES: value 51;
- When using I\_PREAMBLES: value 40.

## Annex C (normative): Core INAP Parameter values

Table C.1: Parameter values

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	"xx"H	400
	PIX_LocationNumber	LocationNumber	"xxxx"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
	PIX_ScflD	ScflD	"xxxx"H	8881
	PIX_ServiceInteractionIndicators	ServiceInteractionIndicators	"xx"H	22
	PIX_ServiceKey1	ServiceKey	"xx"H	27
	PIX_ServiceKey2	ServiceKey	"xx"H	28

Item	Parameter	Parameter type	Explanation/Format	Value
	PIX_SFBillingChargingCharacteristics	SFBillingChargingCharacteristics	"xxxx"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	"xxxx"H	5110
	PIX_UiScriptSpecificInfo	UiScriptSpecificInfo	"xxxx"H	5220

## History

<b>Document history</b>				
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
V1.1.3	May 2000	Publication		