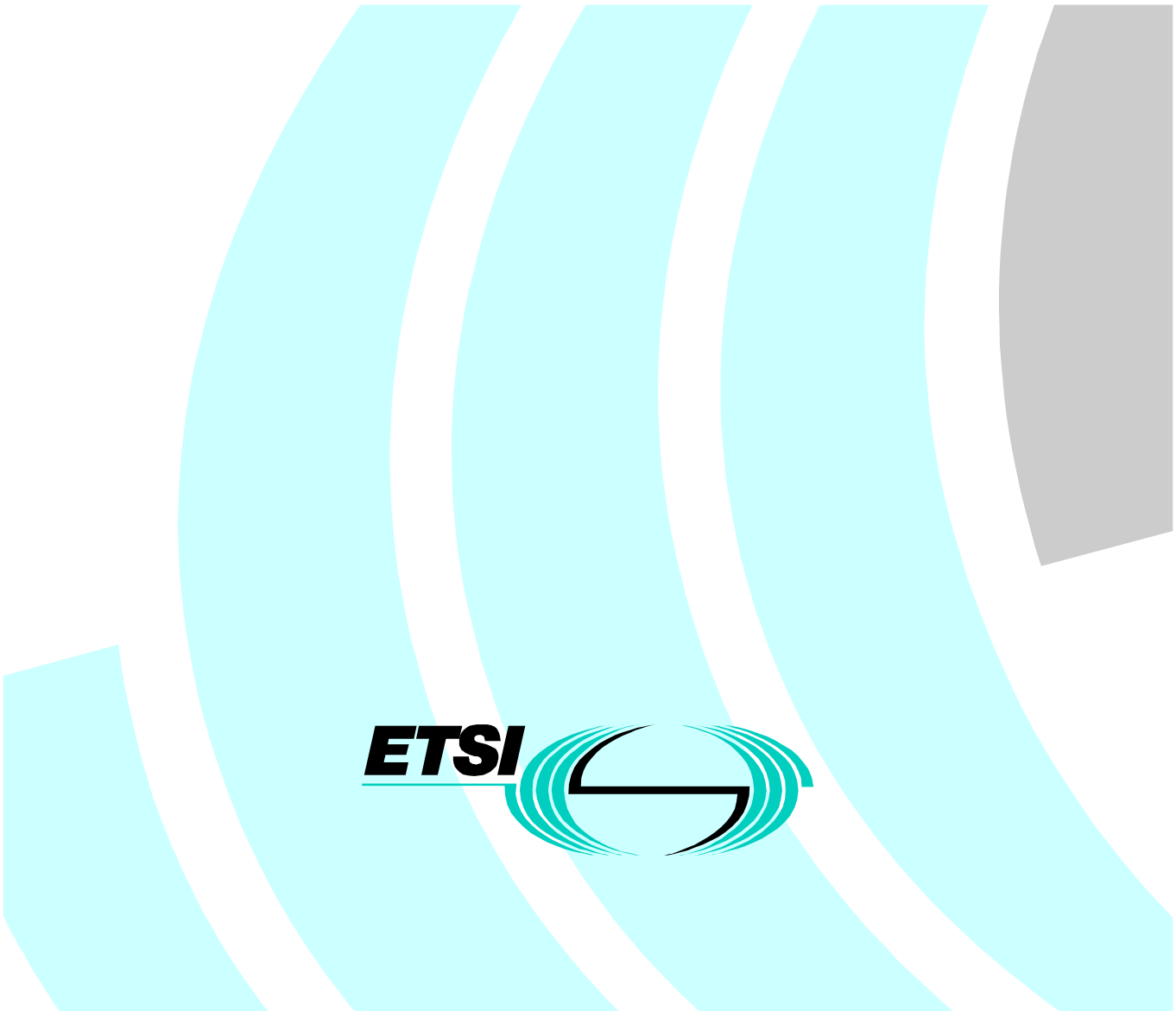
Final draft ETSI EN 301 140-3-2 V1.1.3 (2000-02)

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&TP, SSF

ETSI

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

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

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

Contents

Intellectual Property Rights

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

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

Foreword

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

The present document is part 3, sub-part 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]".

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

1 Scope

The present document 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	Cabability 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

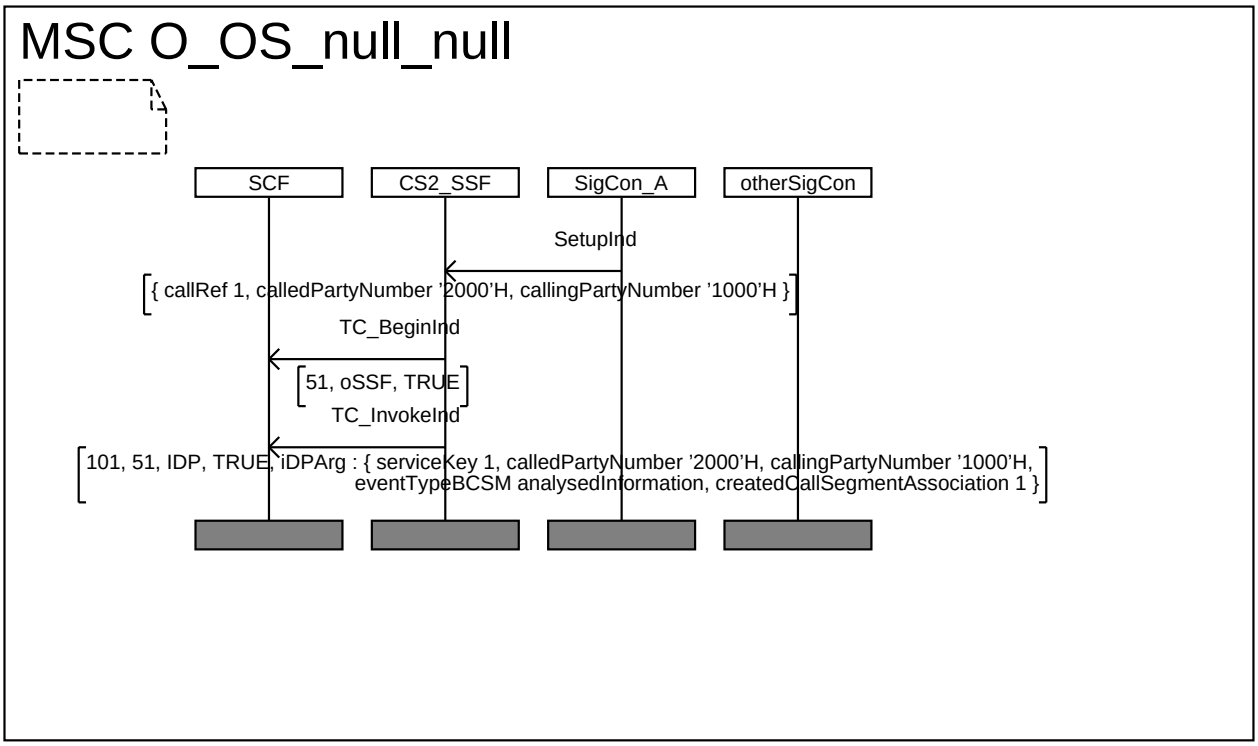
Preamble O_OS_null_null

O_null

CP1-1! SetUpInd

L1? InitialDP

O_OS_null_null



2 - Preamble O_S2P_null_null

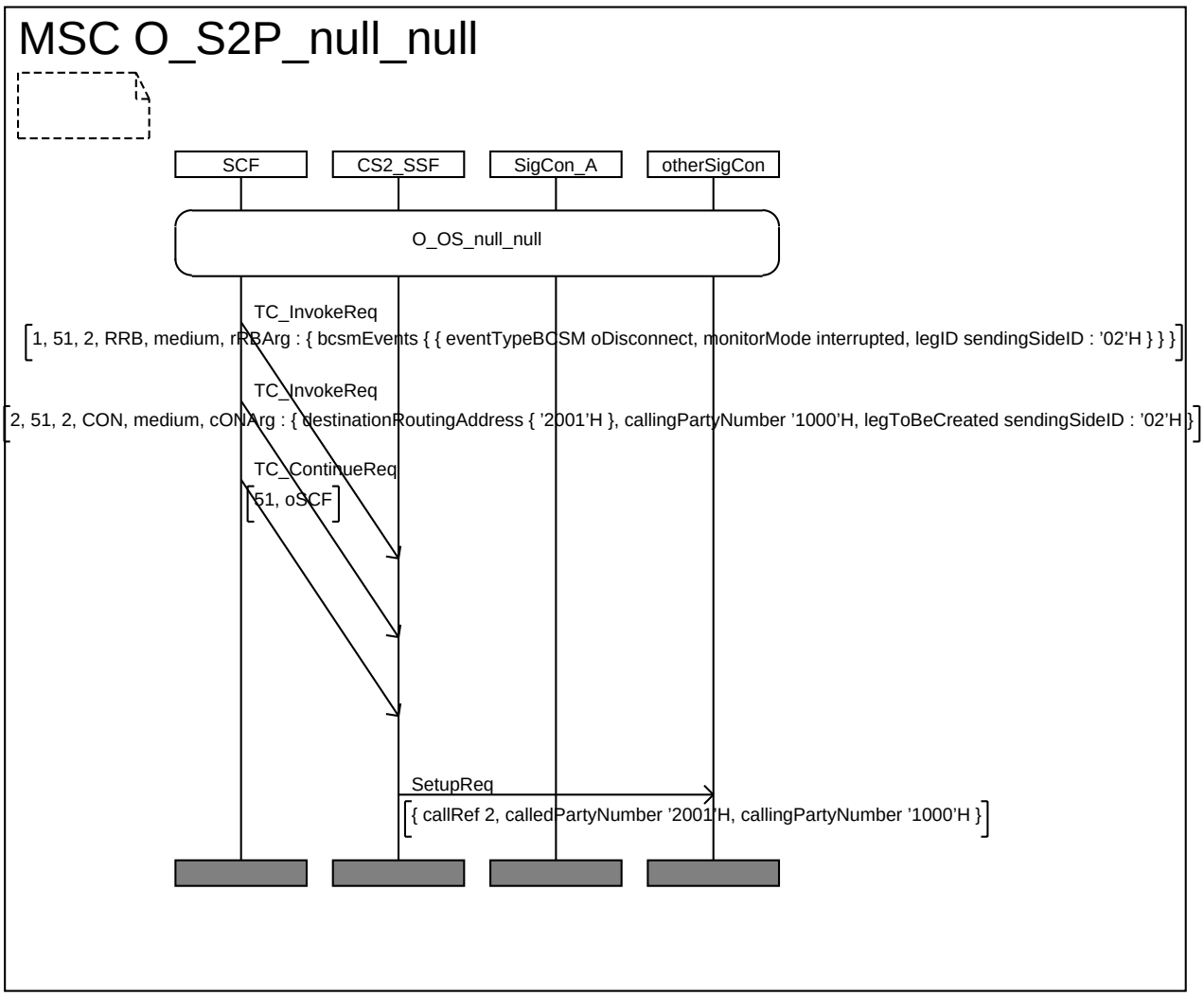
O_OS_null_null

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)

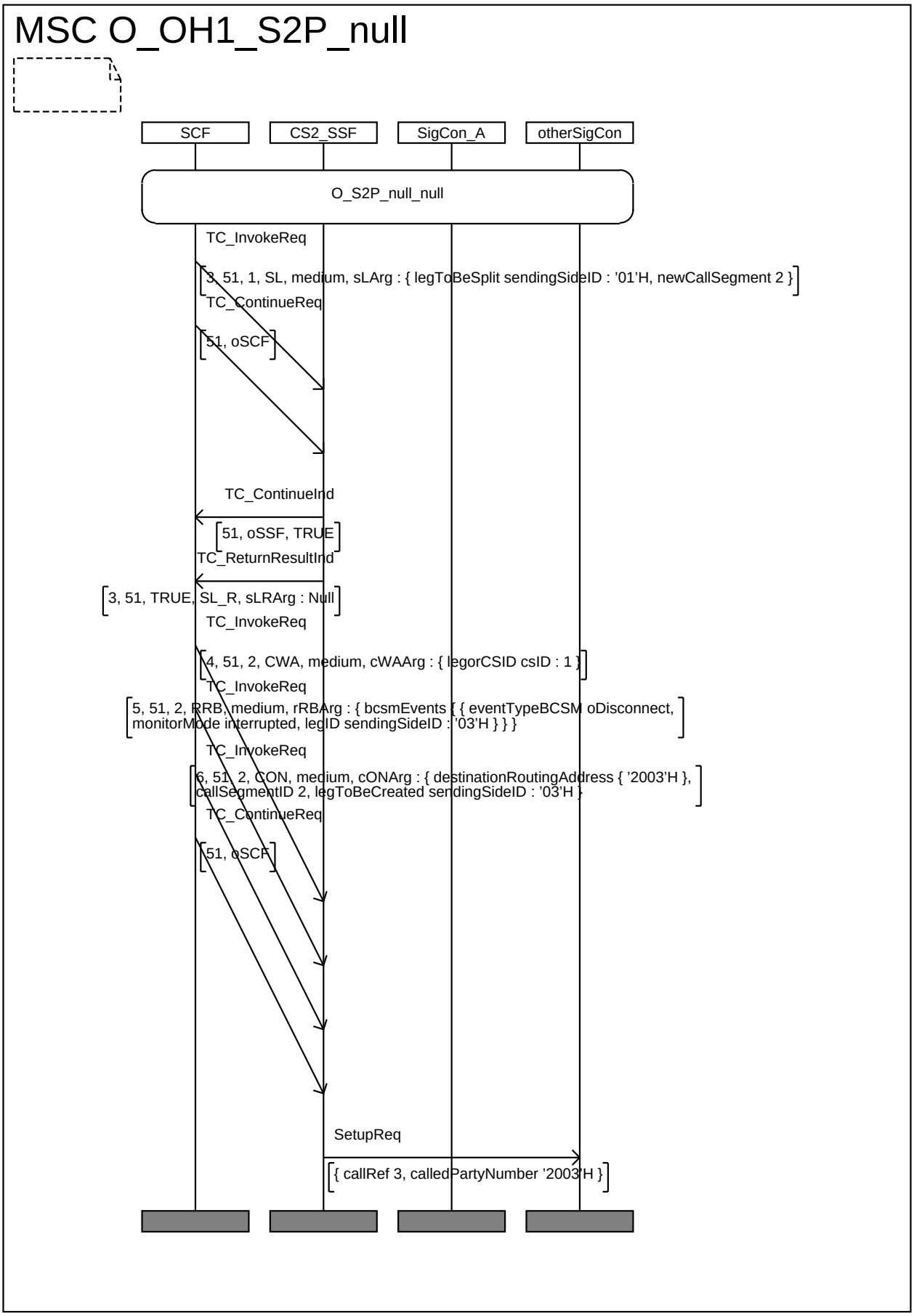
L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 1)
L1! RequestReportBCSMEvent(3,oDisconnect)

L1! Connect(3,2)

CP1-3? 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

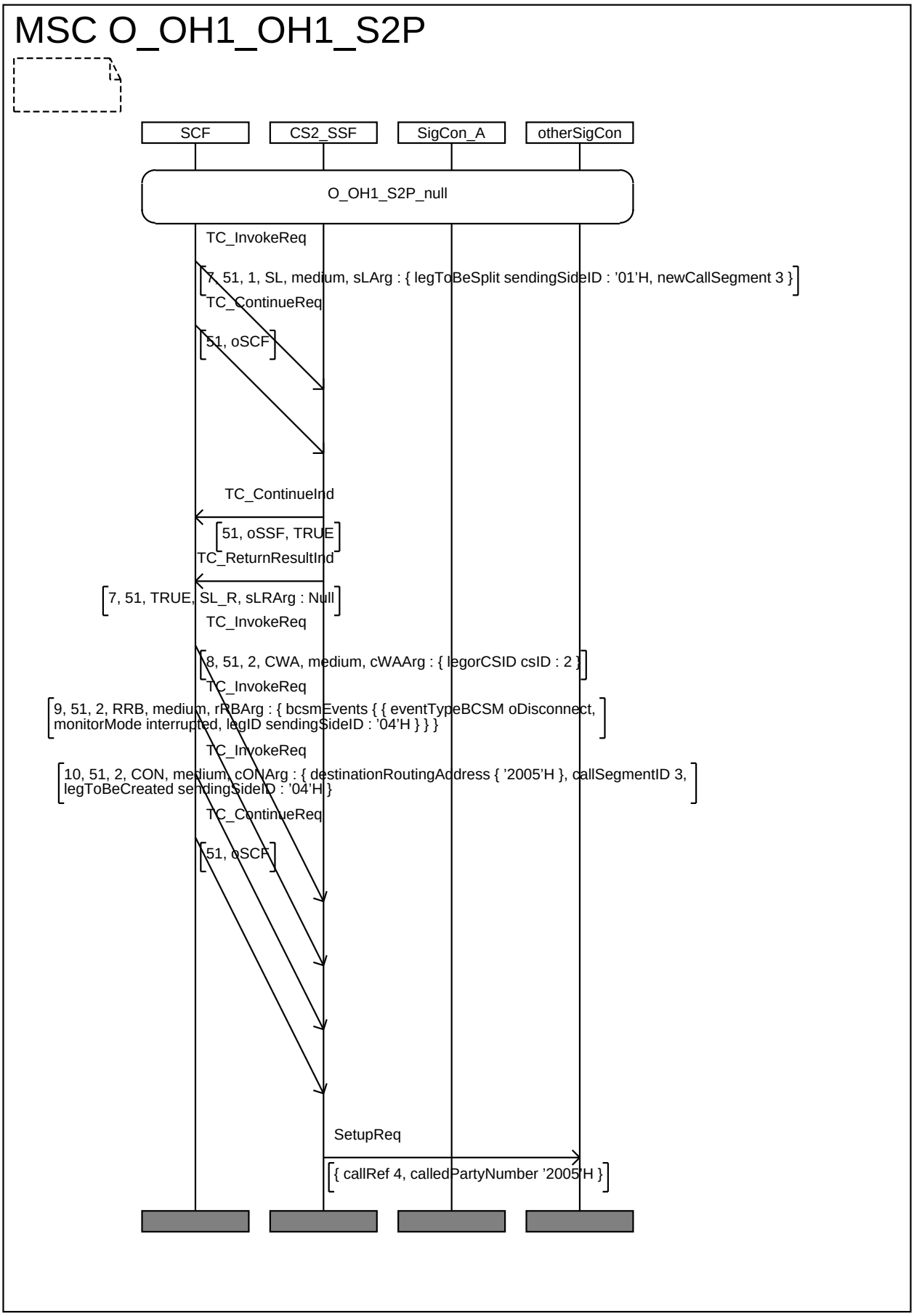
L1! ContinueWithArgument (CsID = 2)

L1! RequestReportBCSMEvent(4,oDisconnect)

L1! Connect(4,3)

CP1-4? SetUpReq

O_OH(1)_OH(1)_S2P



5 - Preamble O_null_OH(2)_S2P

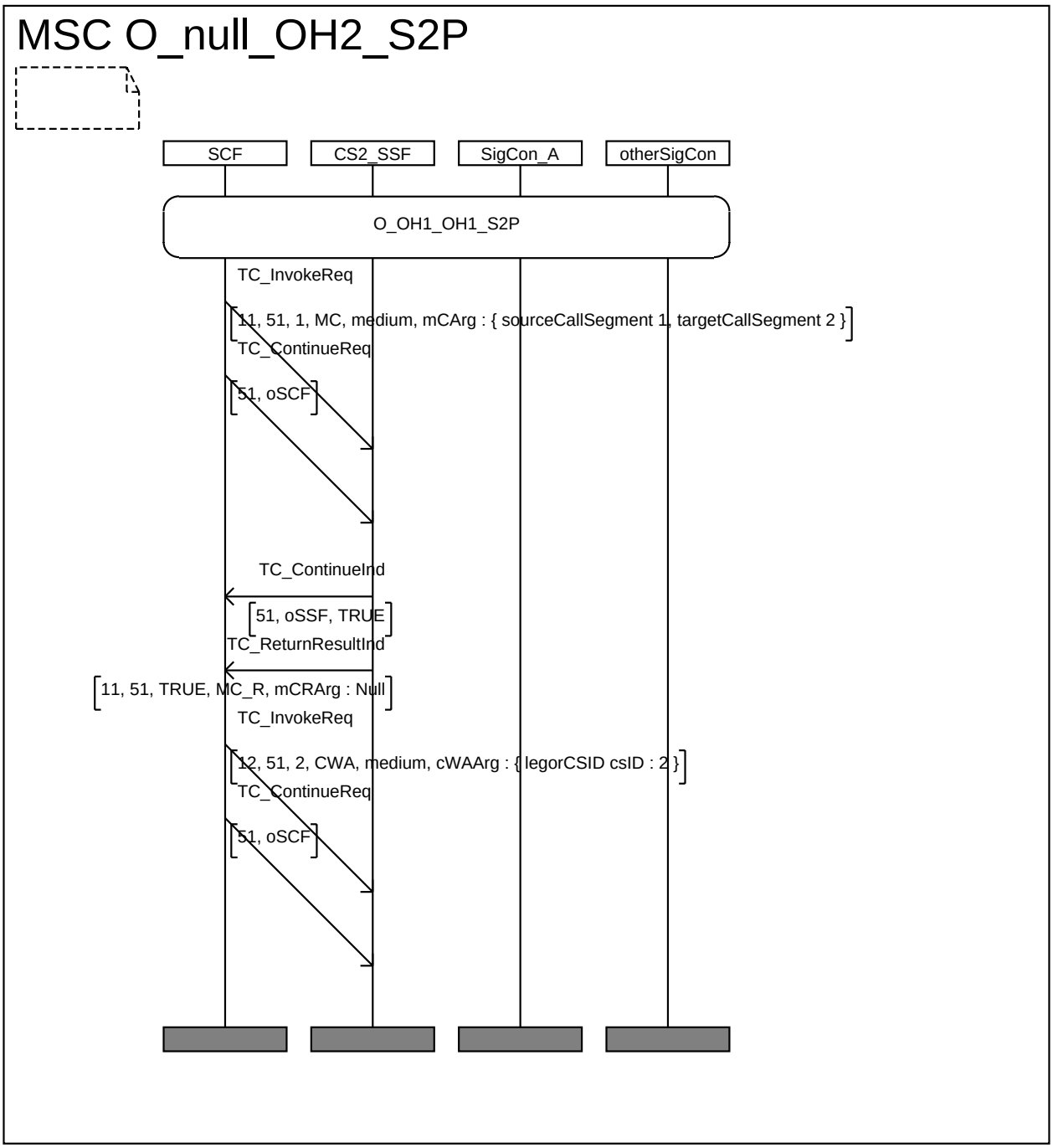
O_OH(1)_OH(1)_S2P

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=2)

O_null_OH(2)_S2P



6 - Preamble O_null_null_S4P

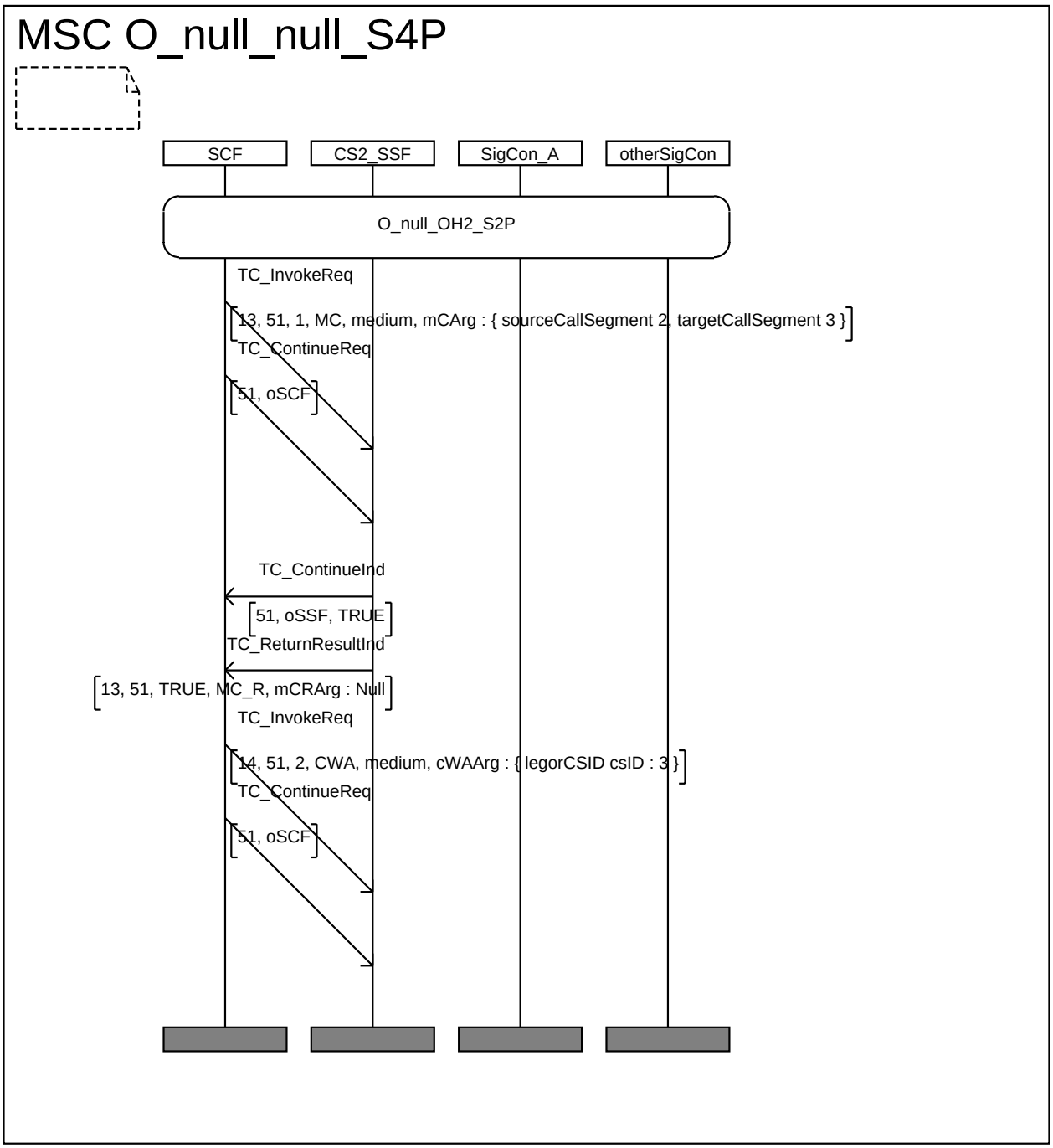
O_null_OH(2)_S2P

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)

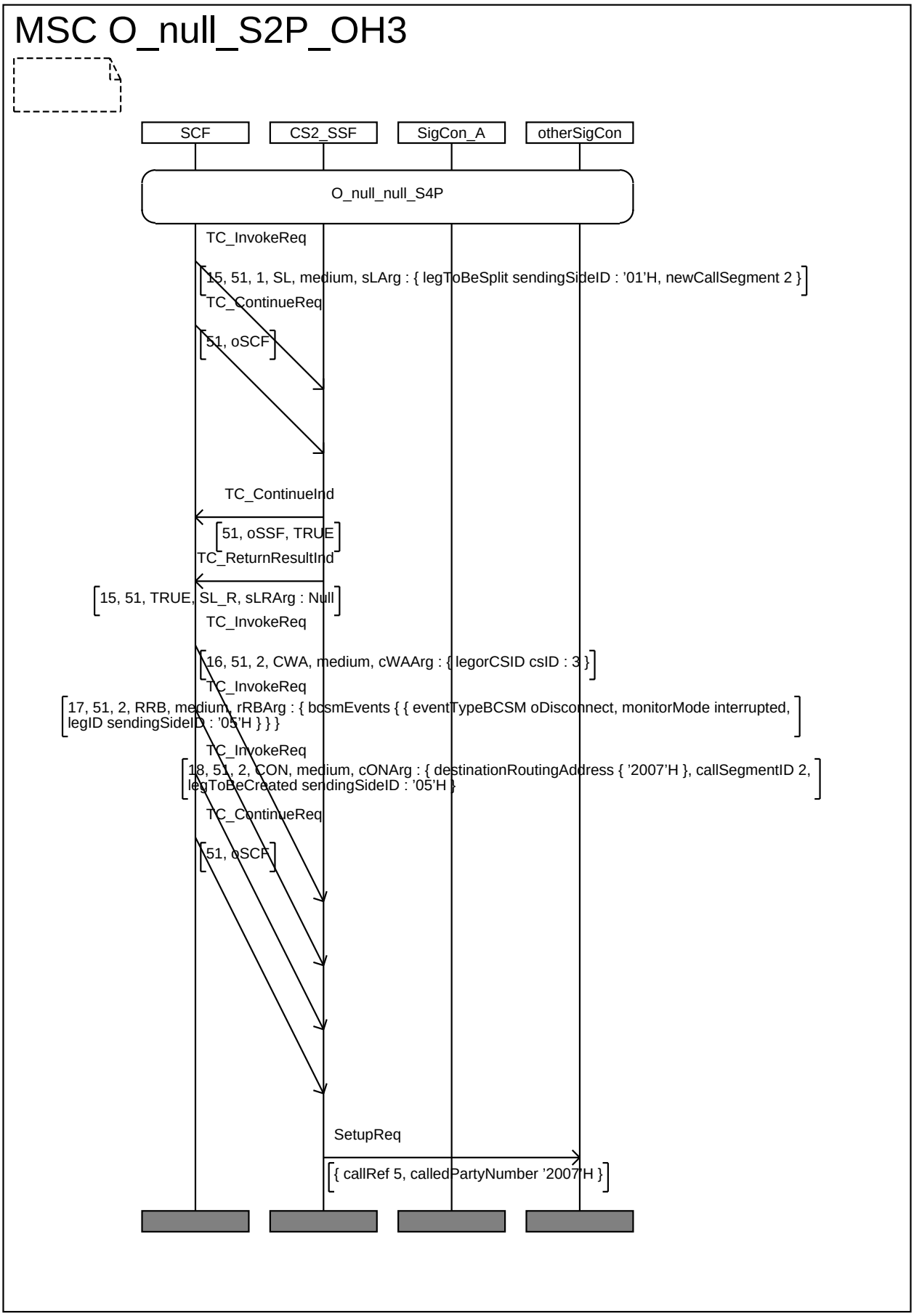
L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 3)
L1! RequestReportBCSMEvent(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)

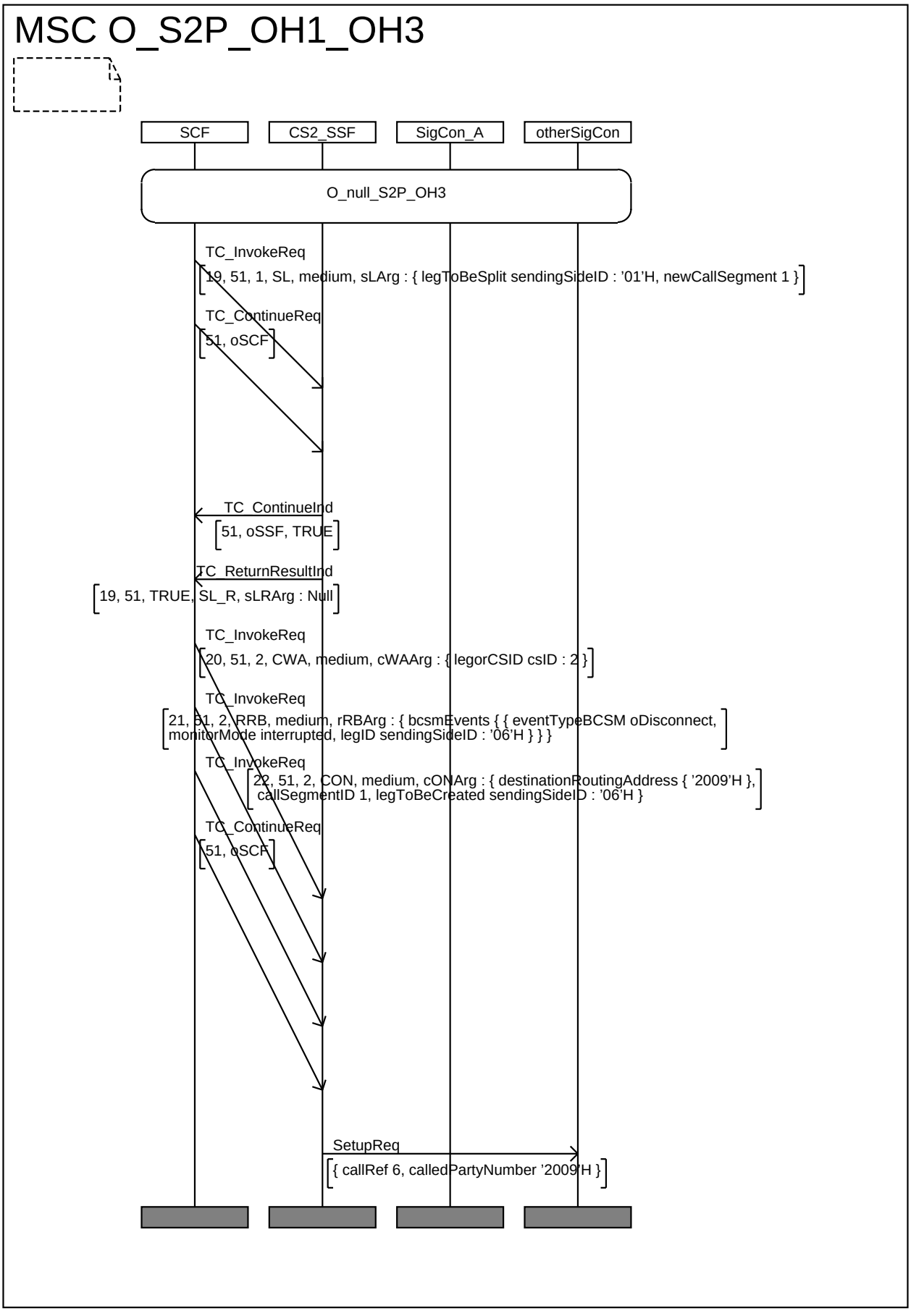
L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 2)
L1! RequestReportBCSMEvent(6,oDisconnect)

L1! Connect(6,1)

CP1-6? SetUpReq

O_S2P_OH(1)_OH(3)



9 - Preamble O_null_S3P_OH(3)

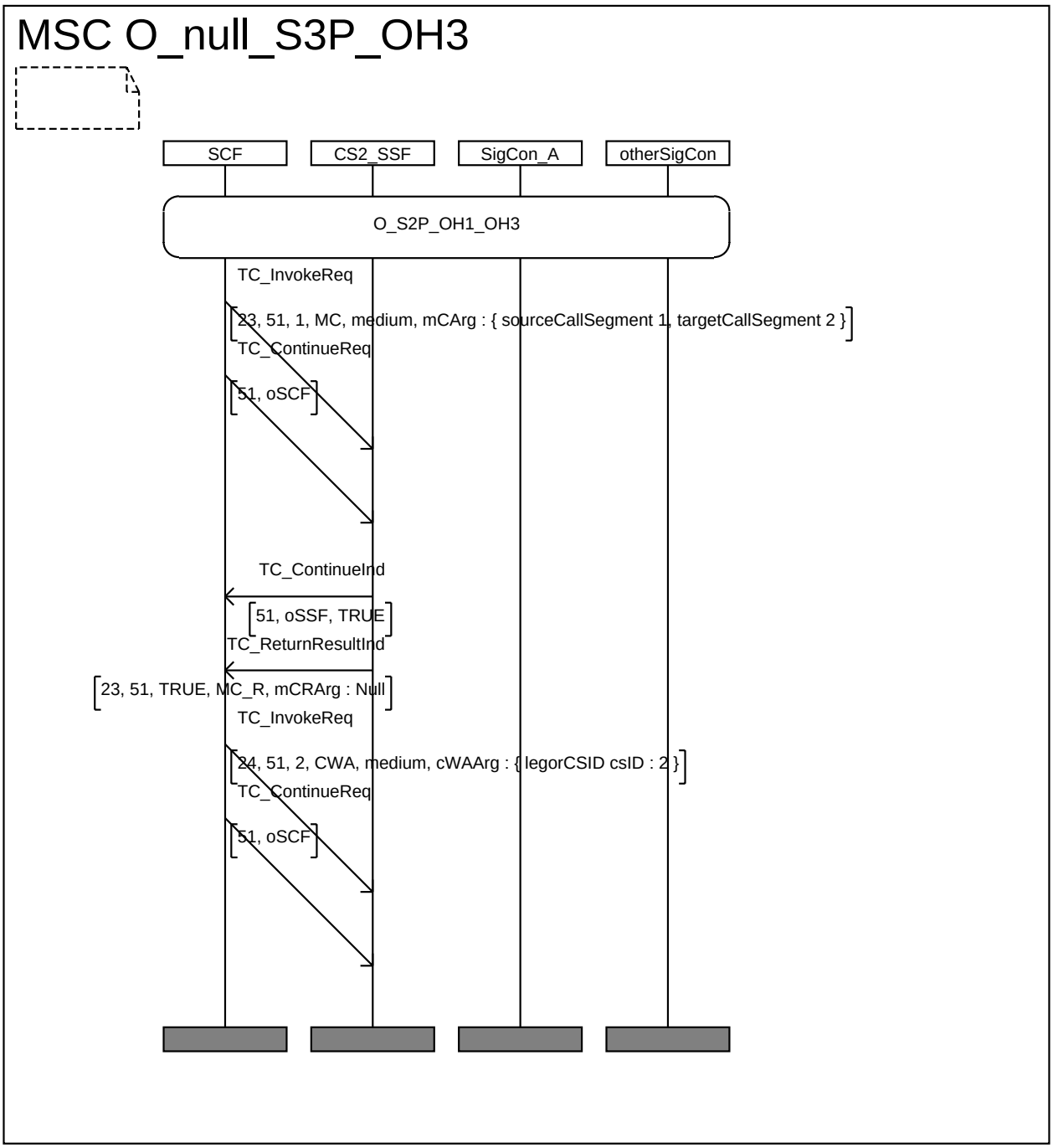
O_S2P_OH(1)_OH(3)

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)

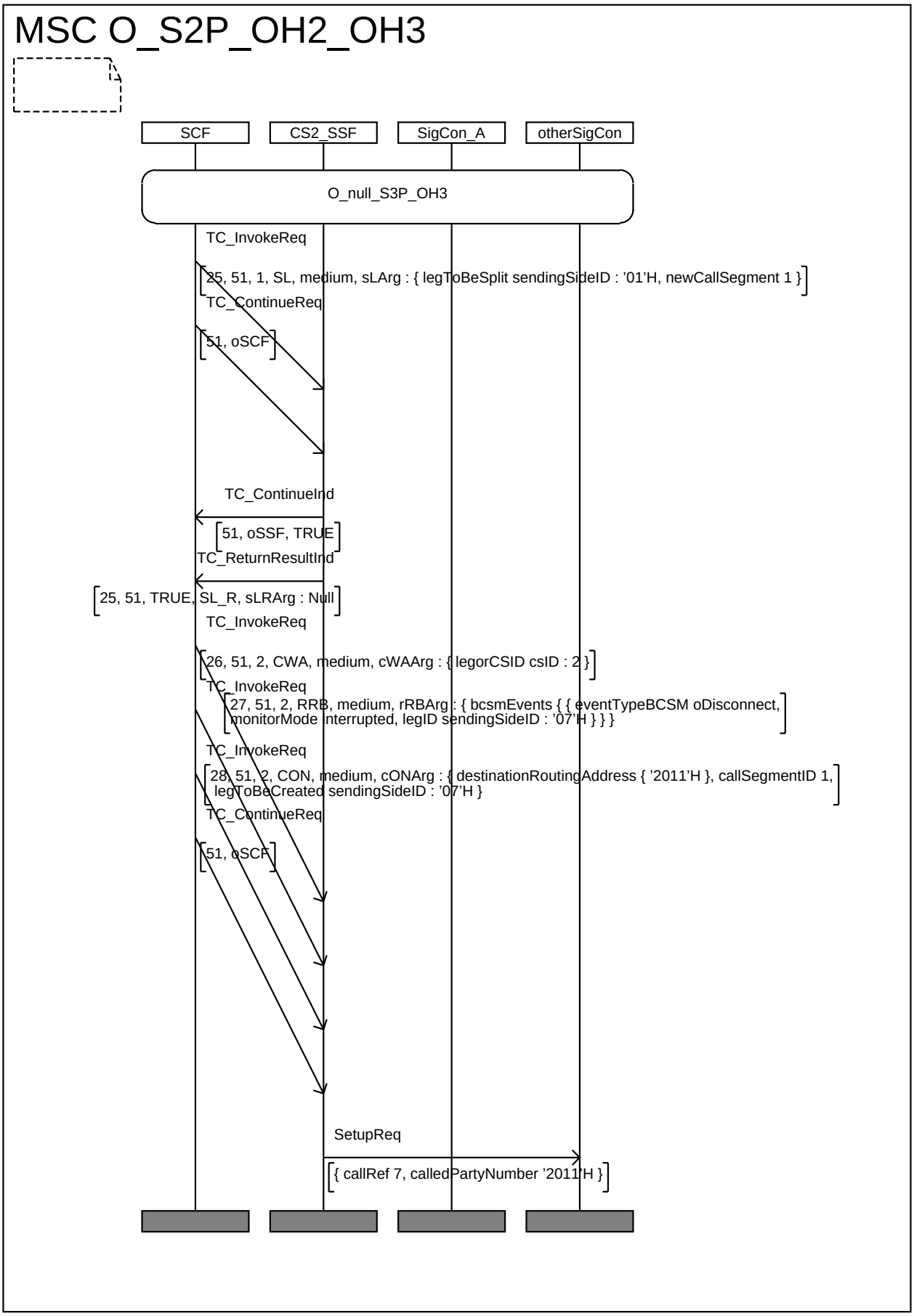
L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 2)
L1! RequestReportBCSMEvent(7,oDisconnect)

L1! Connect(7,1)

CP1-7? SetUpReq

O_S2P_OH(2)_OH(3)



11 - Preamble O_null_S4P_OH(3)

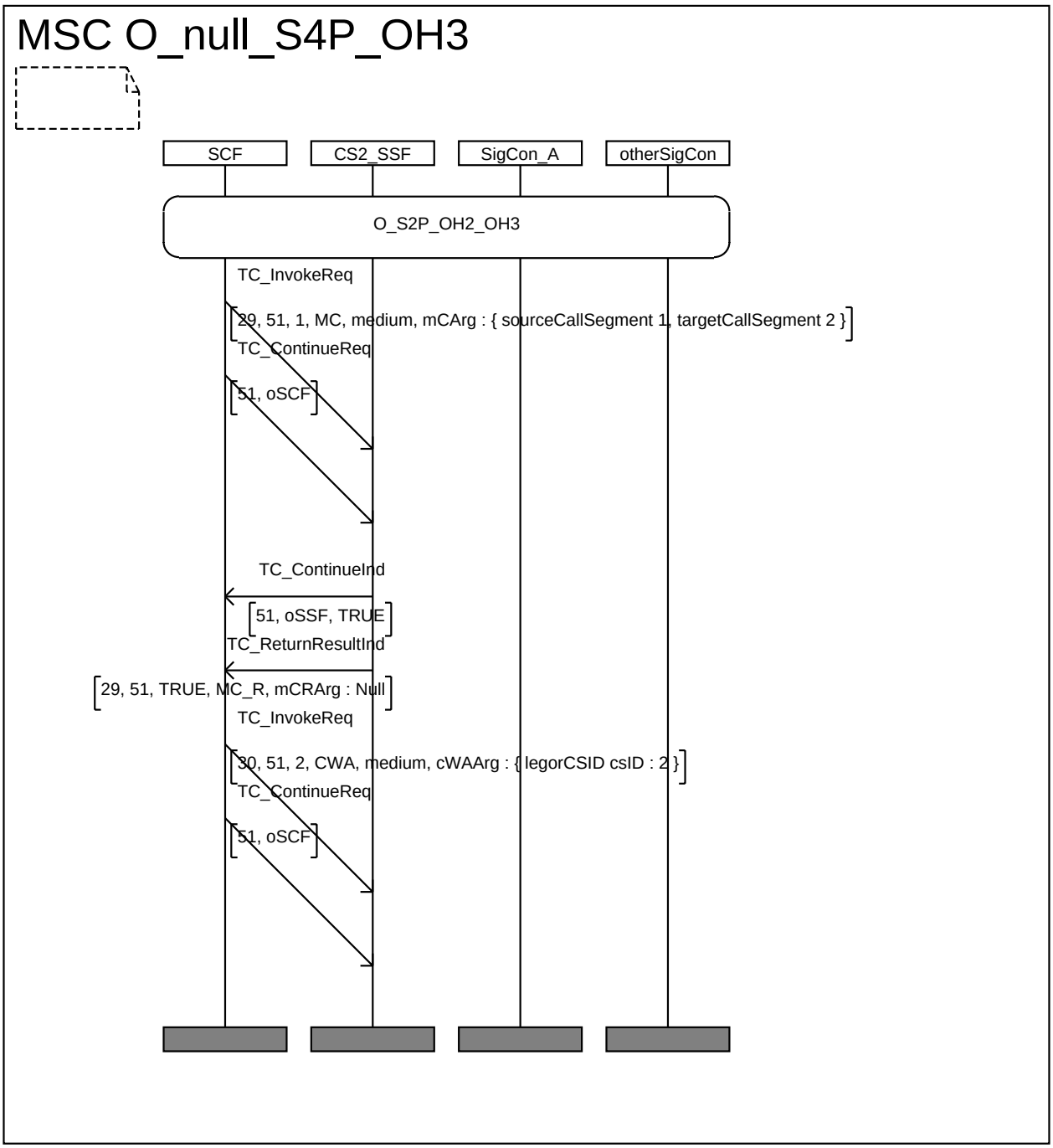
O_S2P_OH(2)_OH(3)

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)

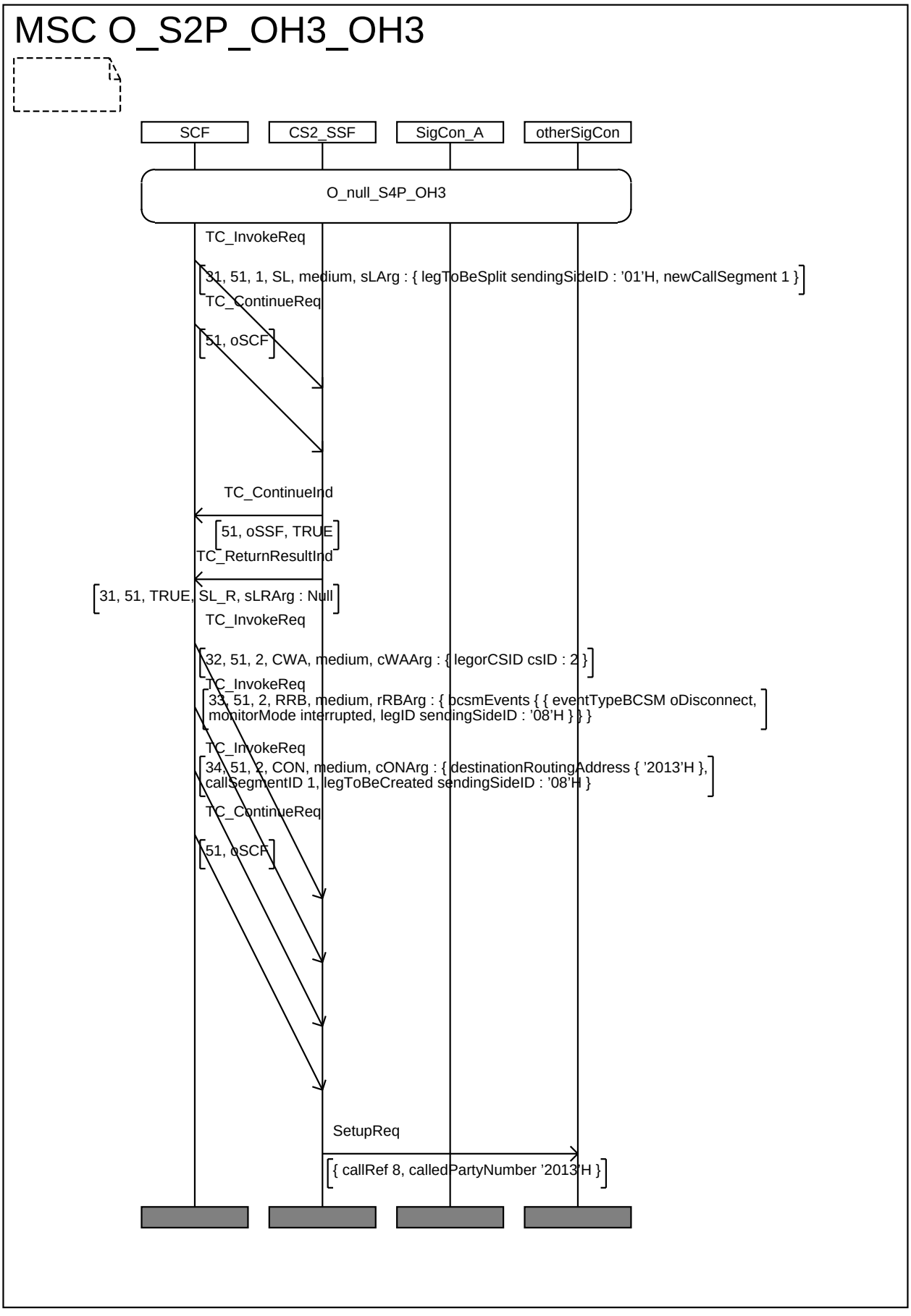
L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 2)
L1! RequestReportBCSMEvent(8,oDisconnect)

L1! Connect(8,1)

CP1-8? SetUpReq

O_S2P_OH(3)_OH(3)



4-2 - Preamble O_null_S3P_null

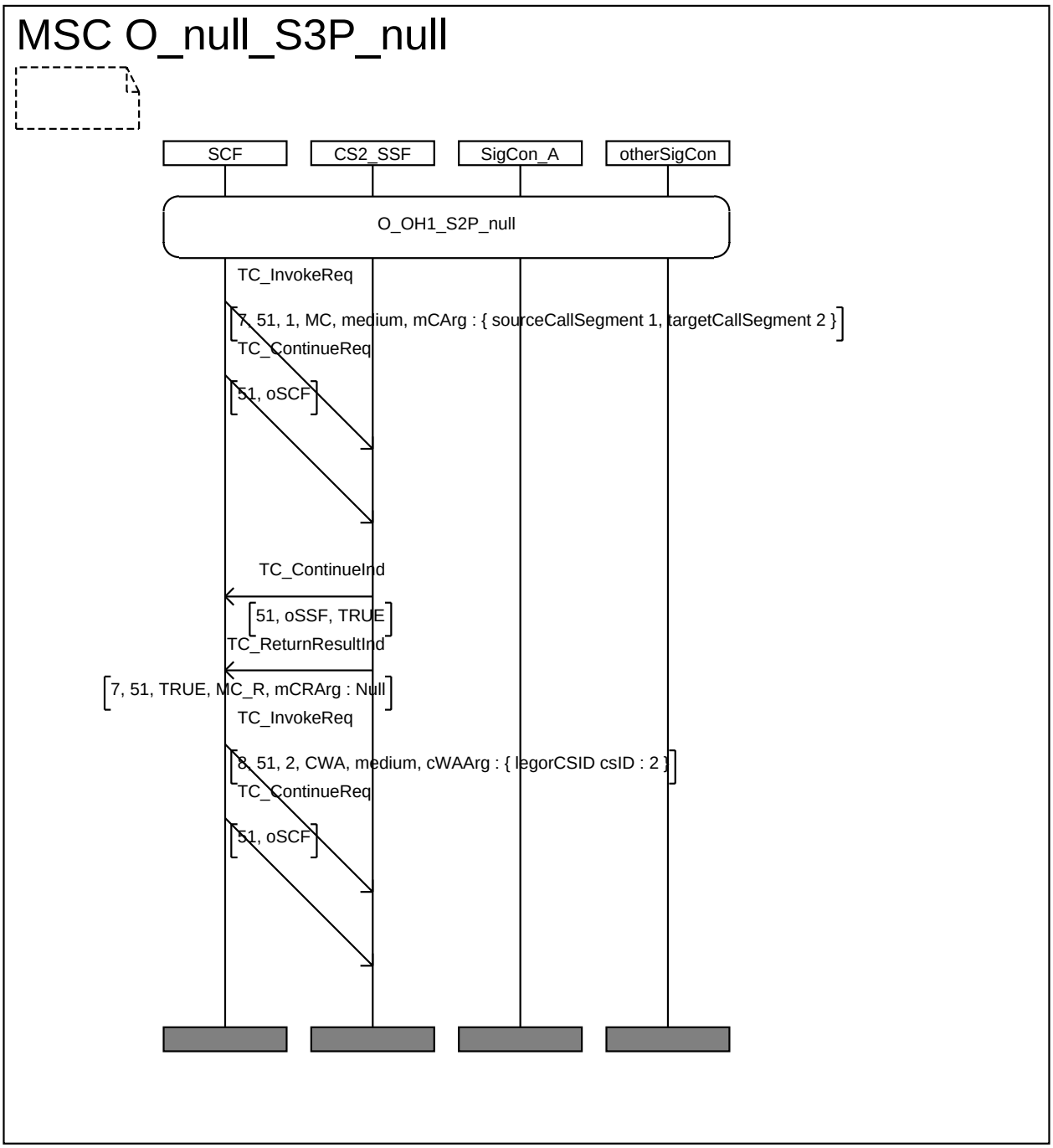
O_OH(1)_S2P_null

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CsId=2)

O_null_S3P_null



7.1.6.2 T (terminating) preamble tree

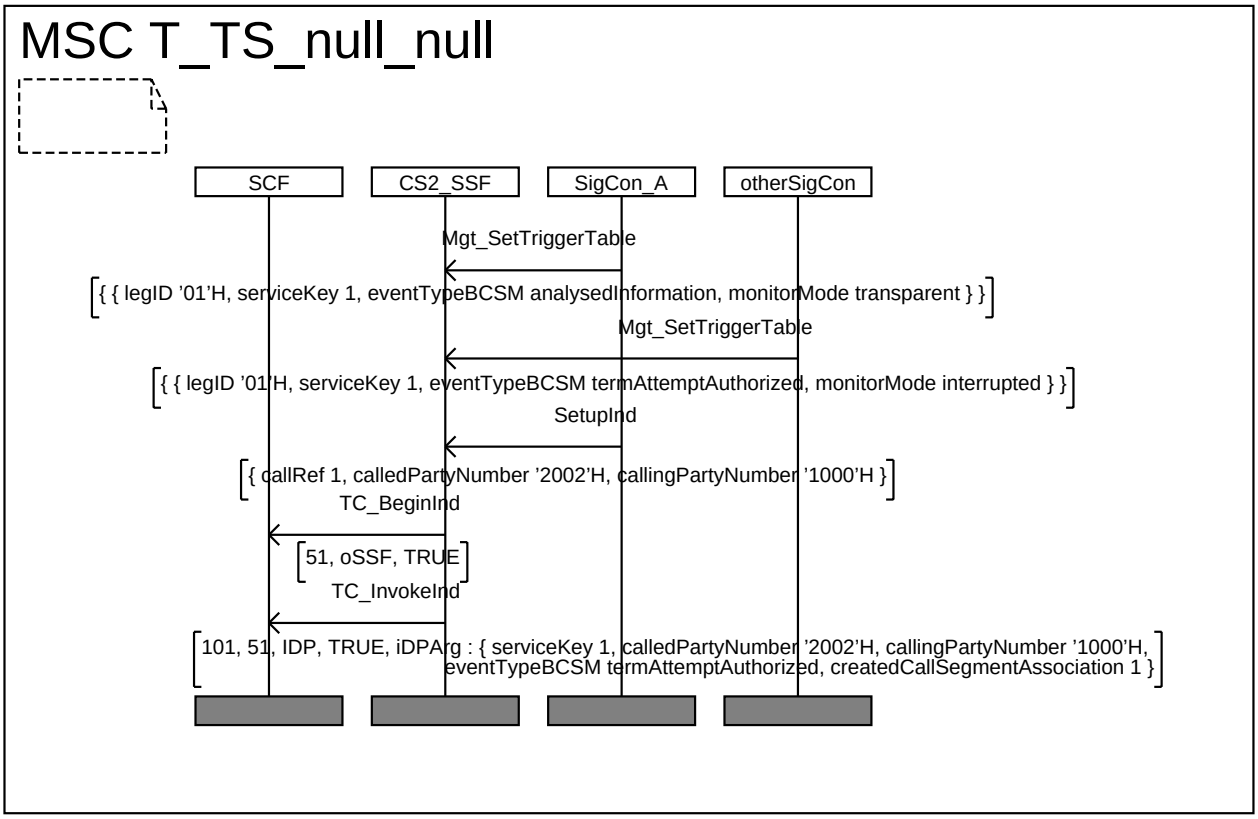
1-1 - Preamble T_TS_null_null

T_null_null_null

CP1-1! SetUpInd

L1? IDP(termAttemptAuthorized)

T_TS_null_null



1-2 - Preamble T_S2P_null_null

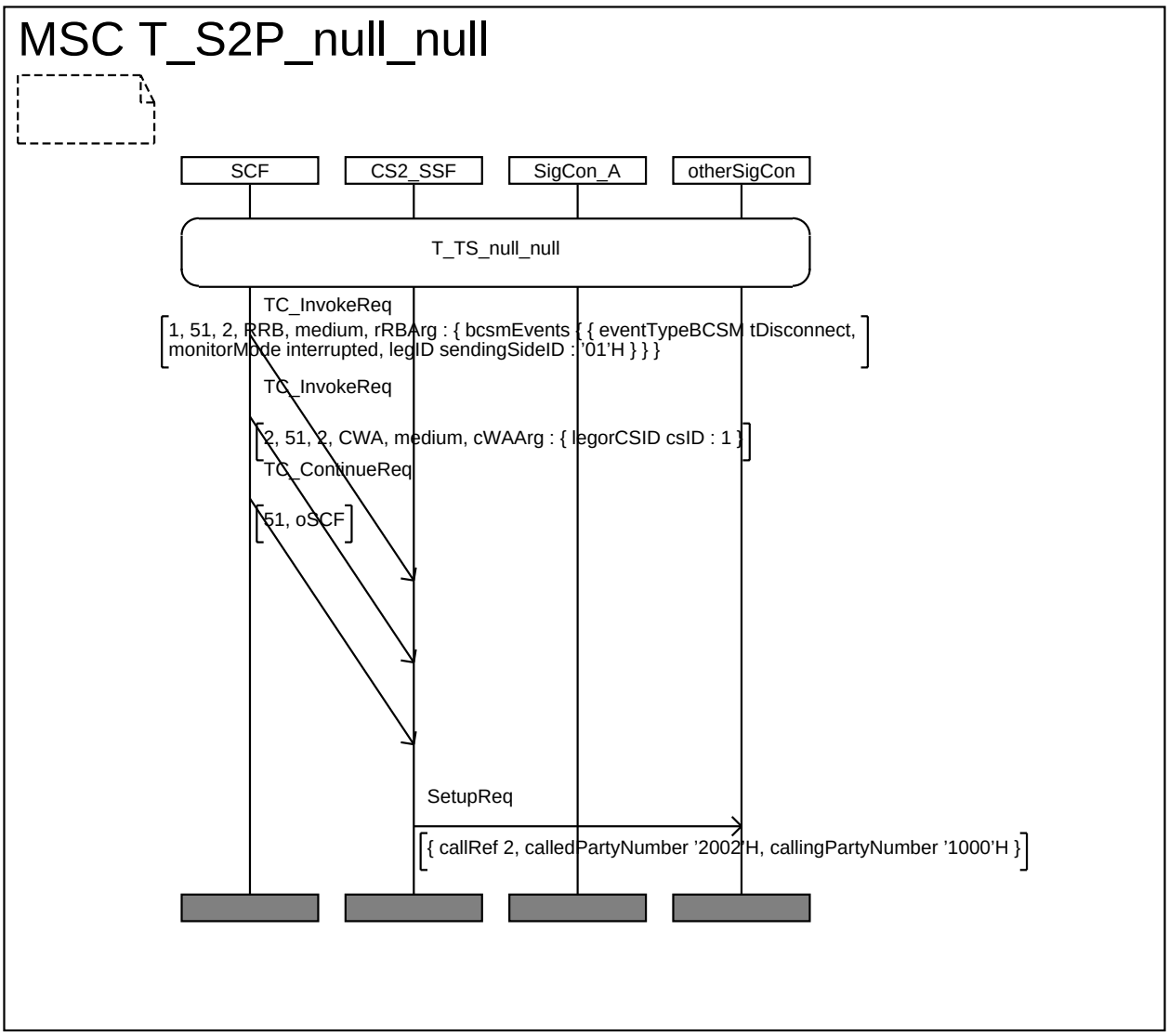
T_TS_null_null

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)

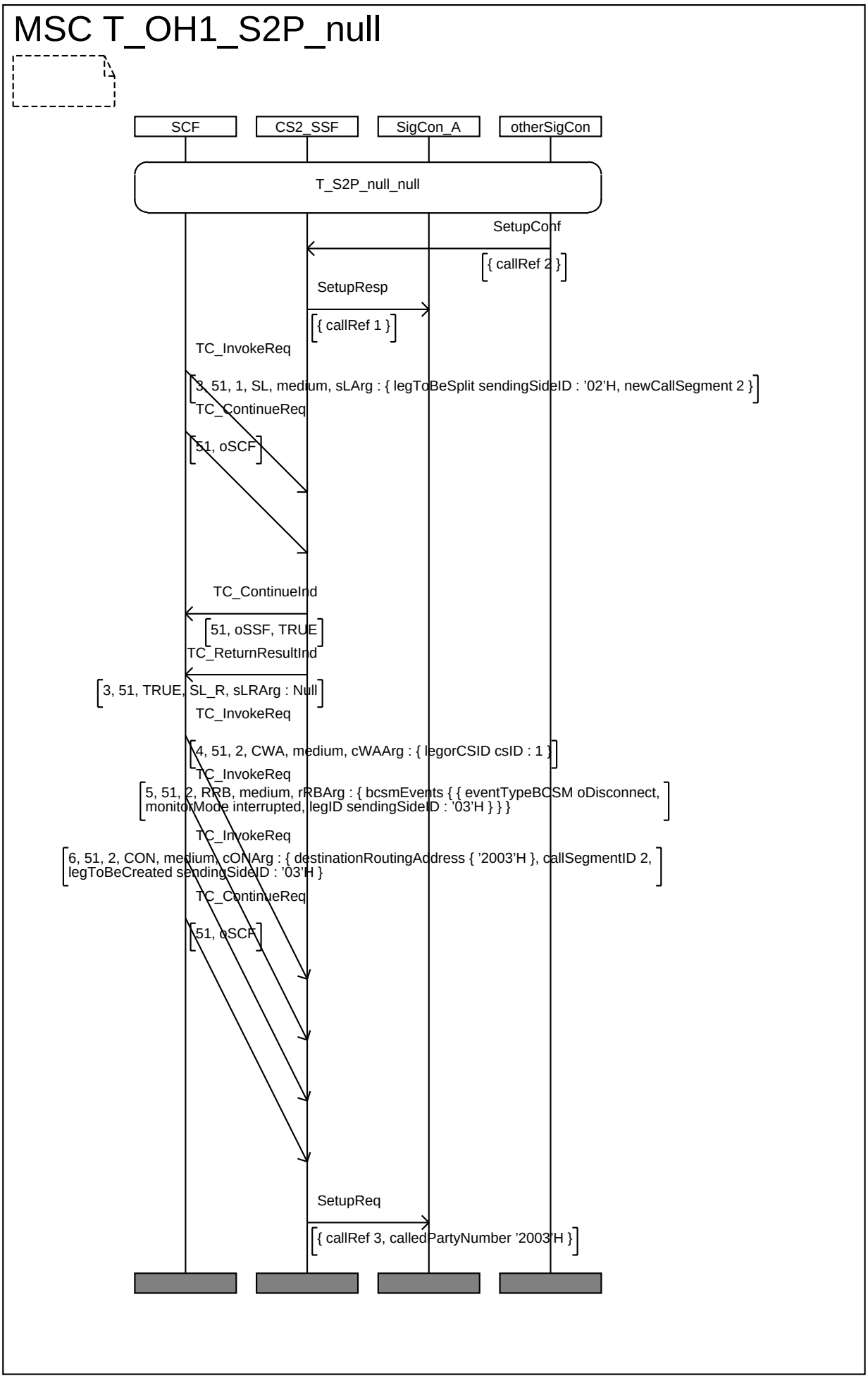
L1?SplitLegReturnResult

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
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
3, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
3, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
4, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
5, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
6, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }

1-4 - Preamble T_null_S3P_null

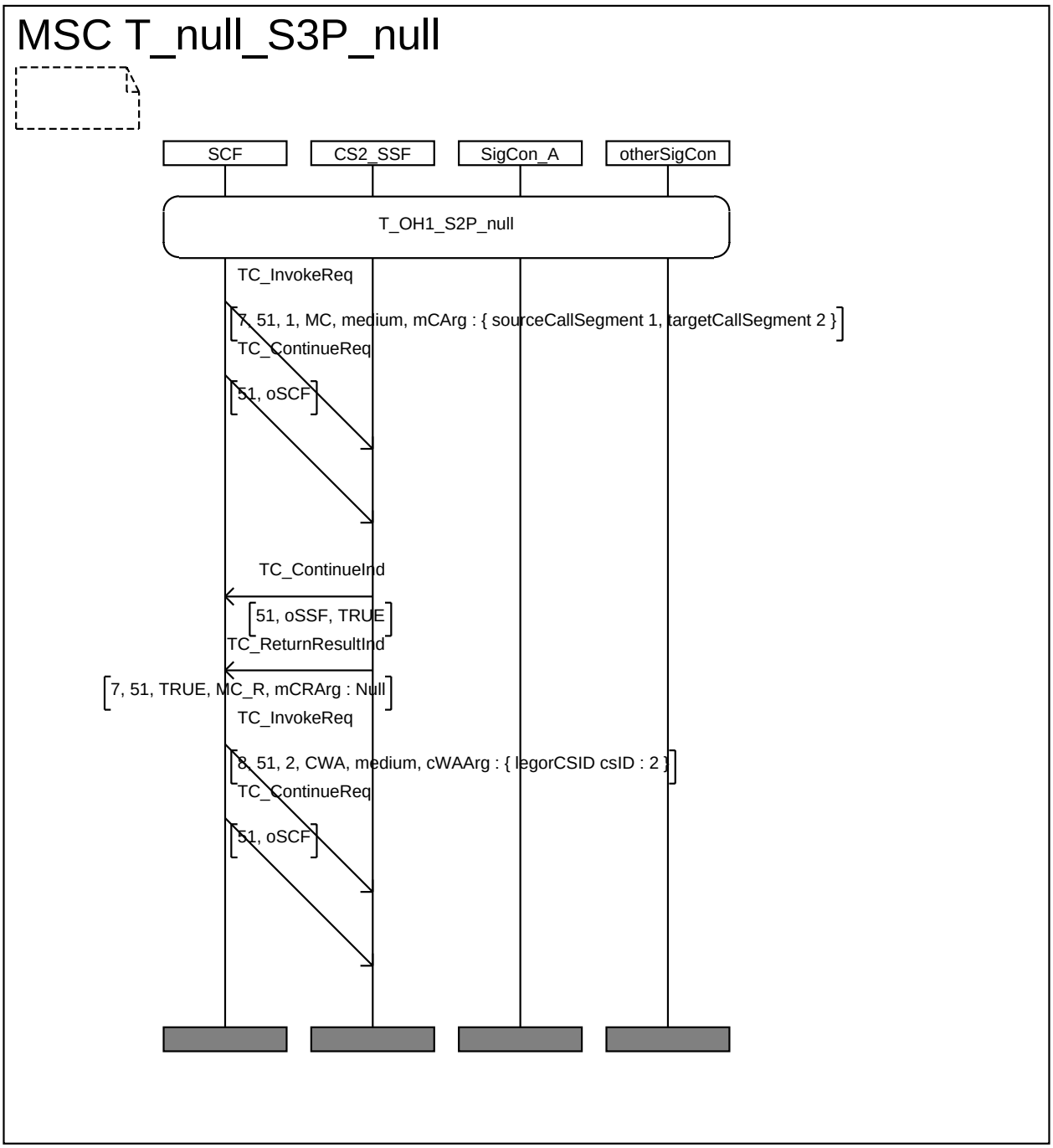
T_OH(1)_S2P_null

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)

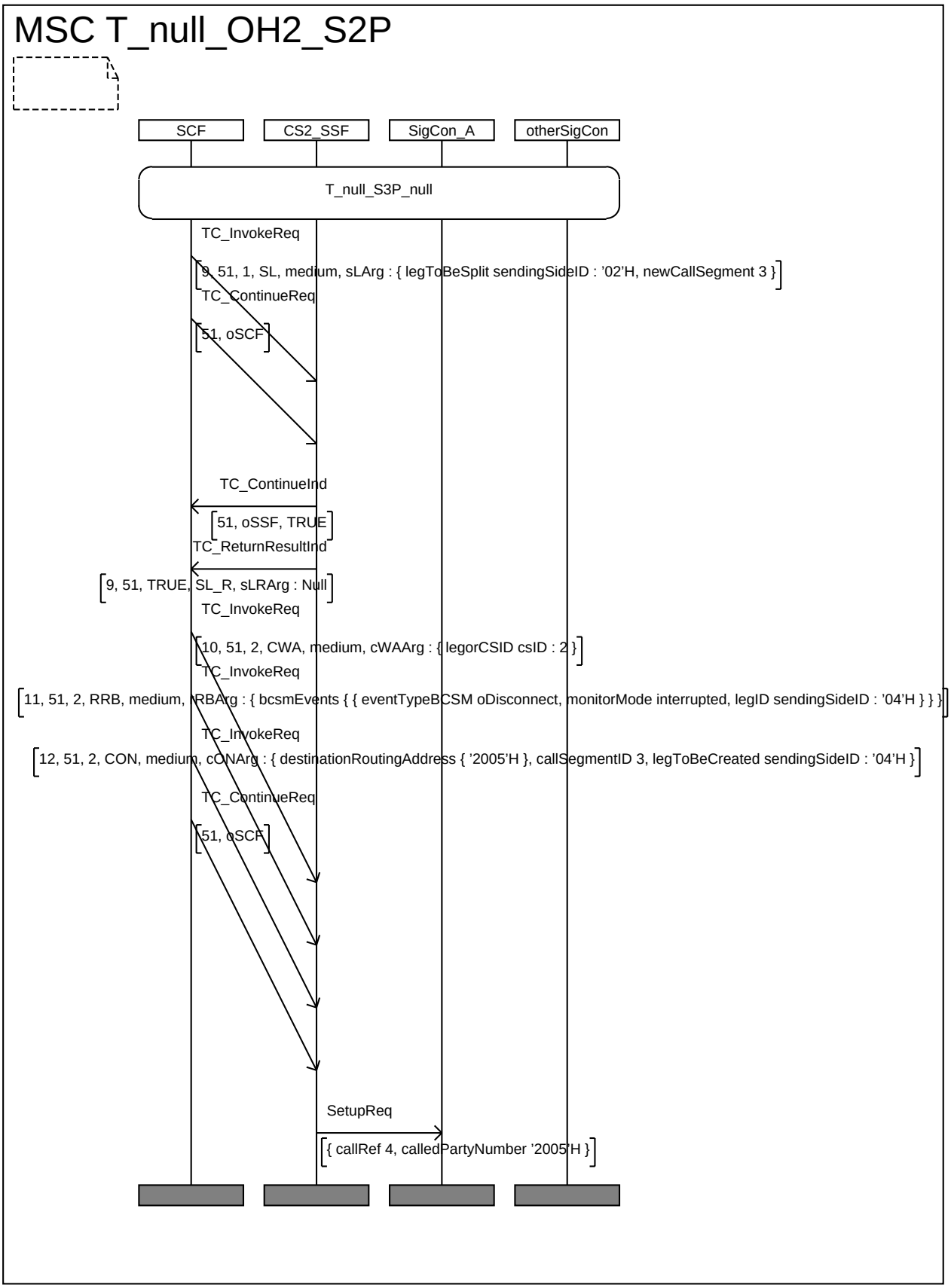
L1?SplitLegReturnResult

L1! ContinueWithArgument (CsID = 2)
L1! RequestReportBCSMEvent(4,tDisconnect)

L1! Connect(4,3)

CP1-4? SetUpReq

T_null_OH(2)_S2P



1-6 - Preamble T_null_null_S4P

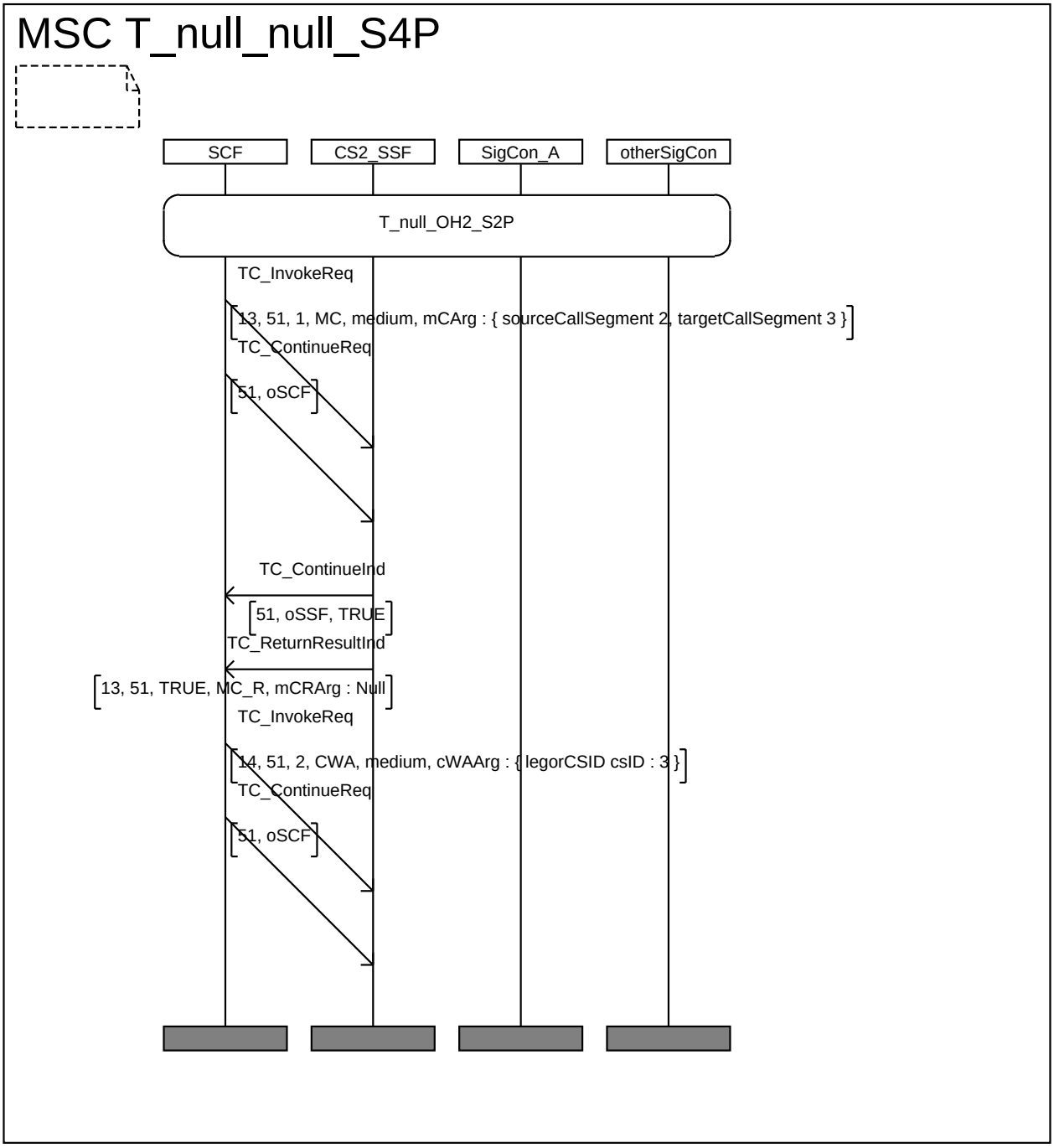
T_null_OH(2)_S2P

L1! MergeCallSegments(2,3)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=3)

T_null_null_S4P



2-1 Preamble T_TF2_null_null

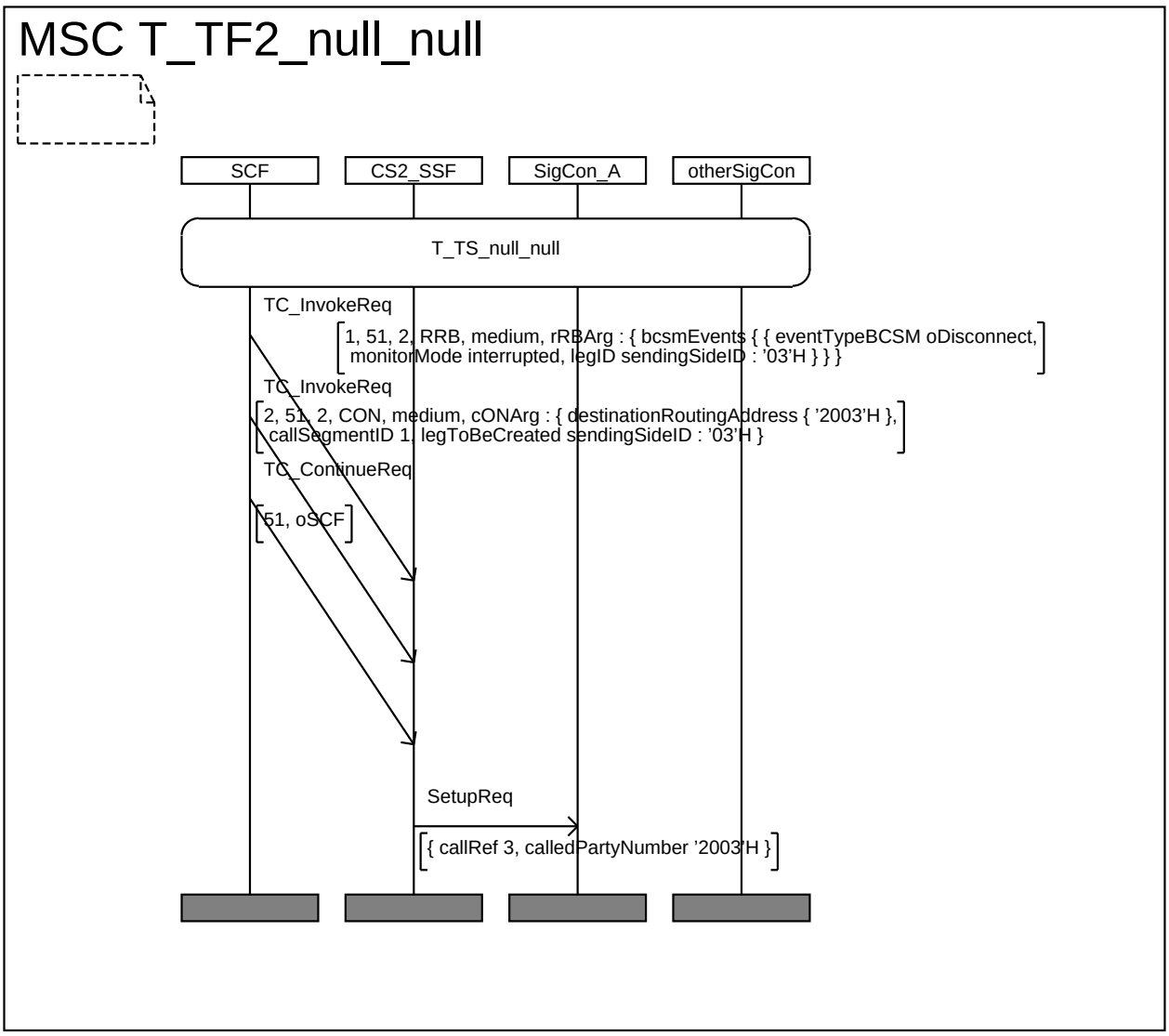
T_TS_null_null

L1! RequestReportBCSMEvent(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

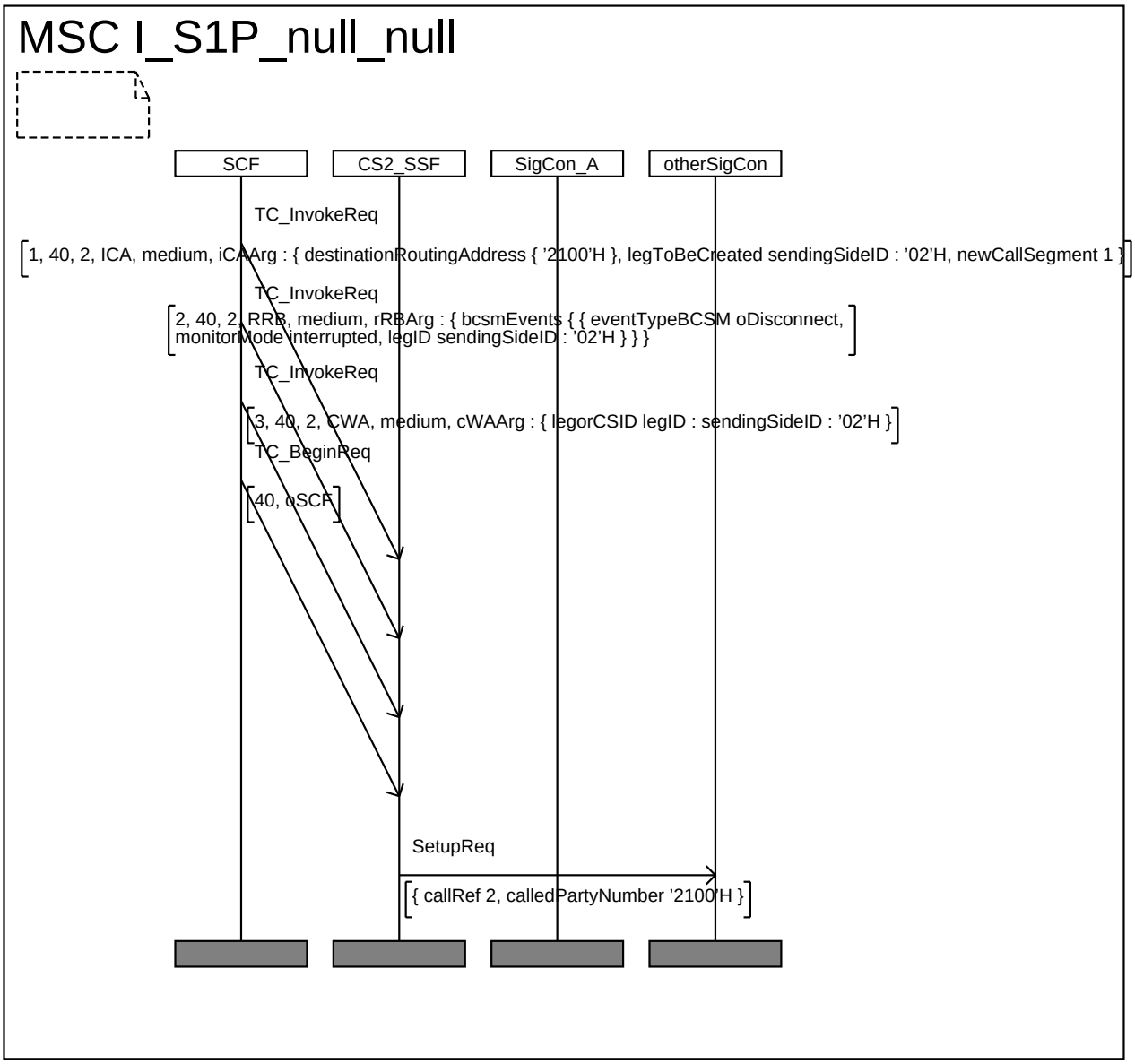
L1! InitiateCallAttempt(2,1)

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

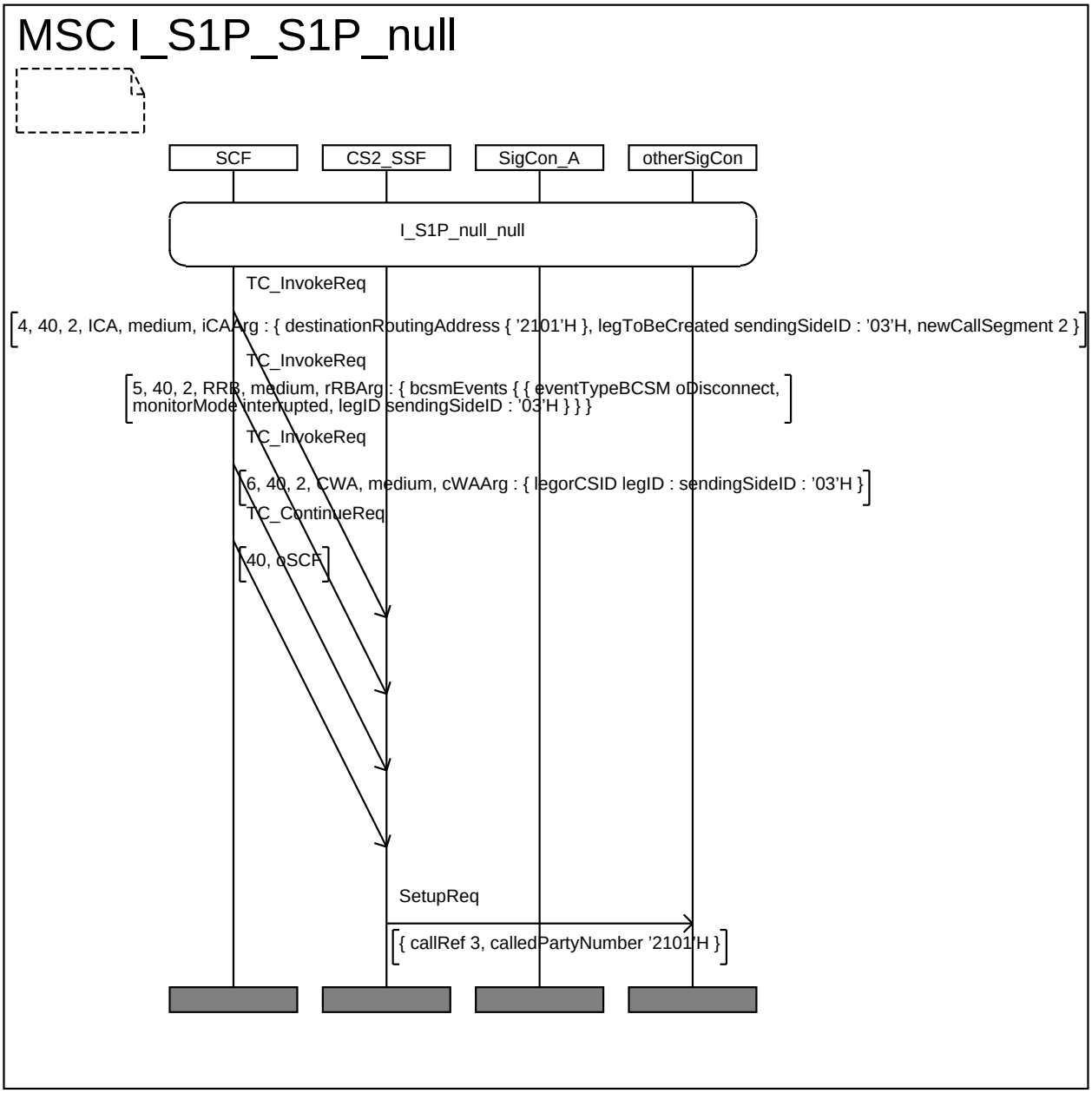
L1! InitiateCallAttempt(3,2)

L1! RequestReportBCSMEvent(3,oDisconnect)

L1! ContinueWithArgument(LegId=3)

CP1-3? SetUpReq

I_S1P_S1P_null



3-1 - Preamble I_S1P_S1P_S1P

I_S1P_S1P_null

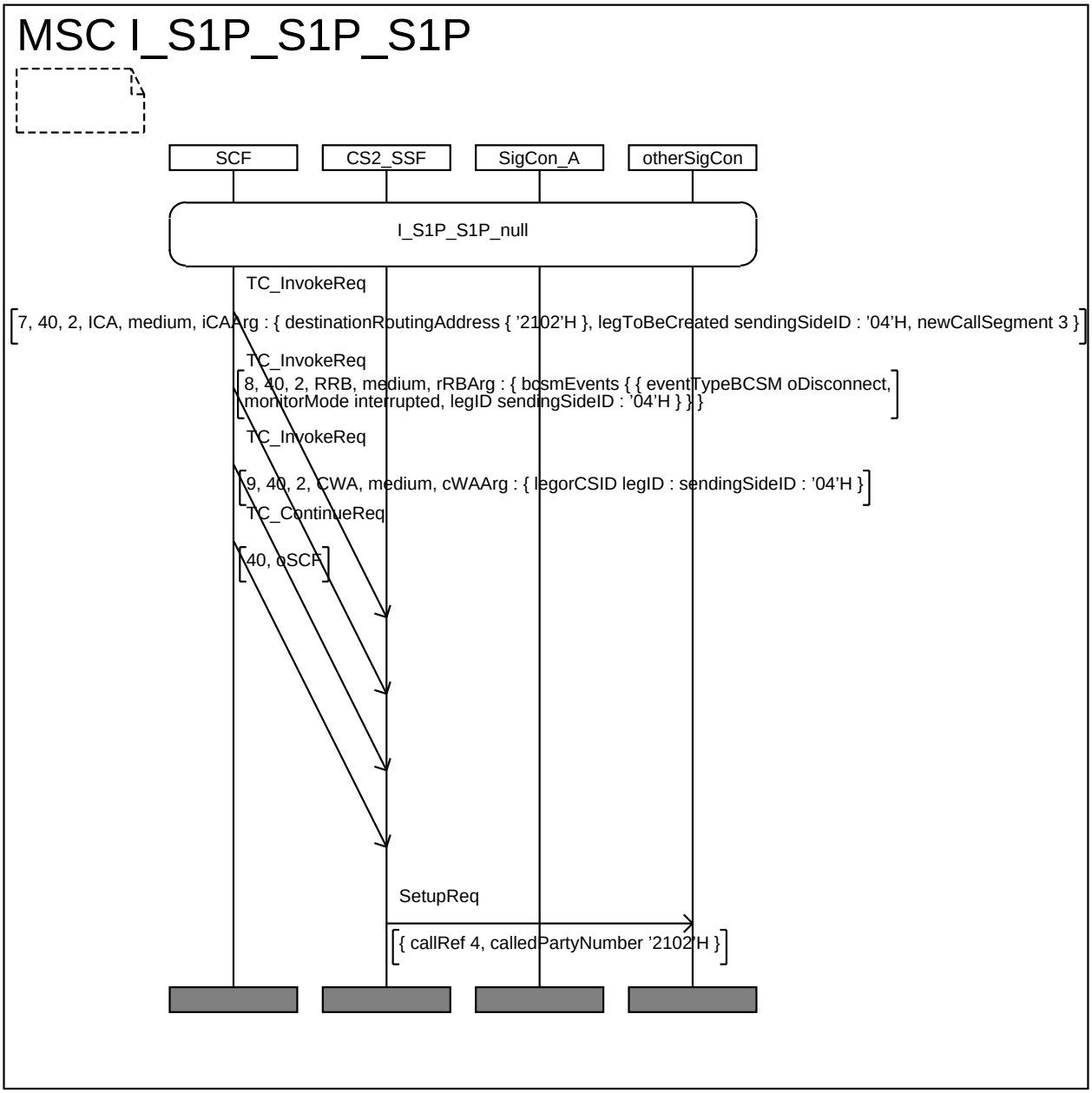
L1! InitiateCallAttempt(4,3)

L1! RequestReportBCSMEvent(4,oDisconnect)

L1! ContinueWithArgument(LegId=4)

CP1-4? SetUpReq

I_S1P_S1P_S1P



4 - Preamble I_null_TF(2)_S1P

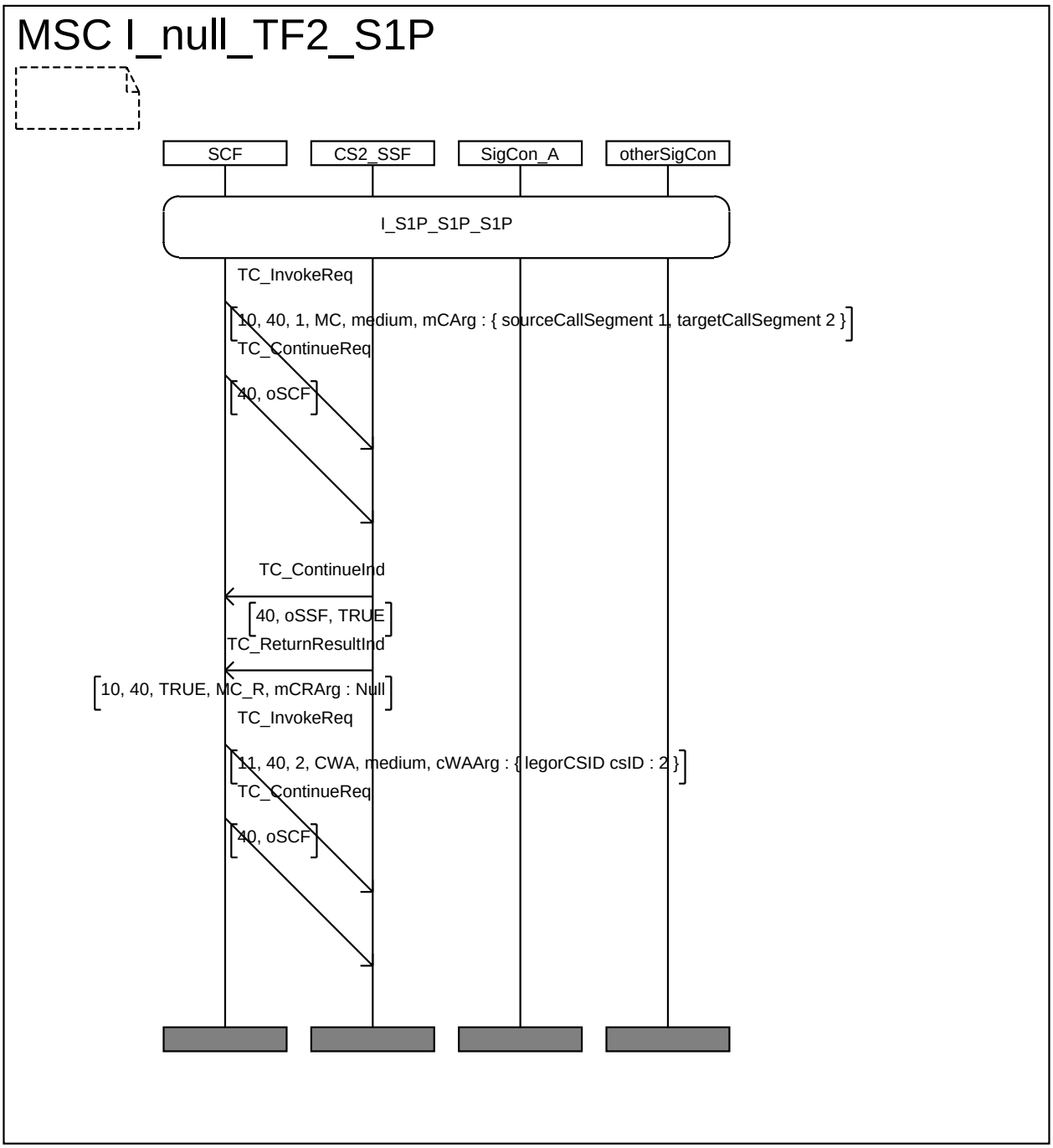
I_S1P_S1P_S1P

L1! MergeCallSegments(1,2)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=2)

I_null_TF(2)_S1P



5 - Preamble I_null_null_TF(3)

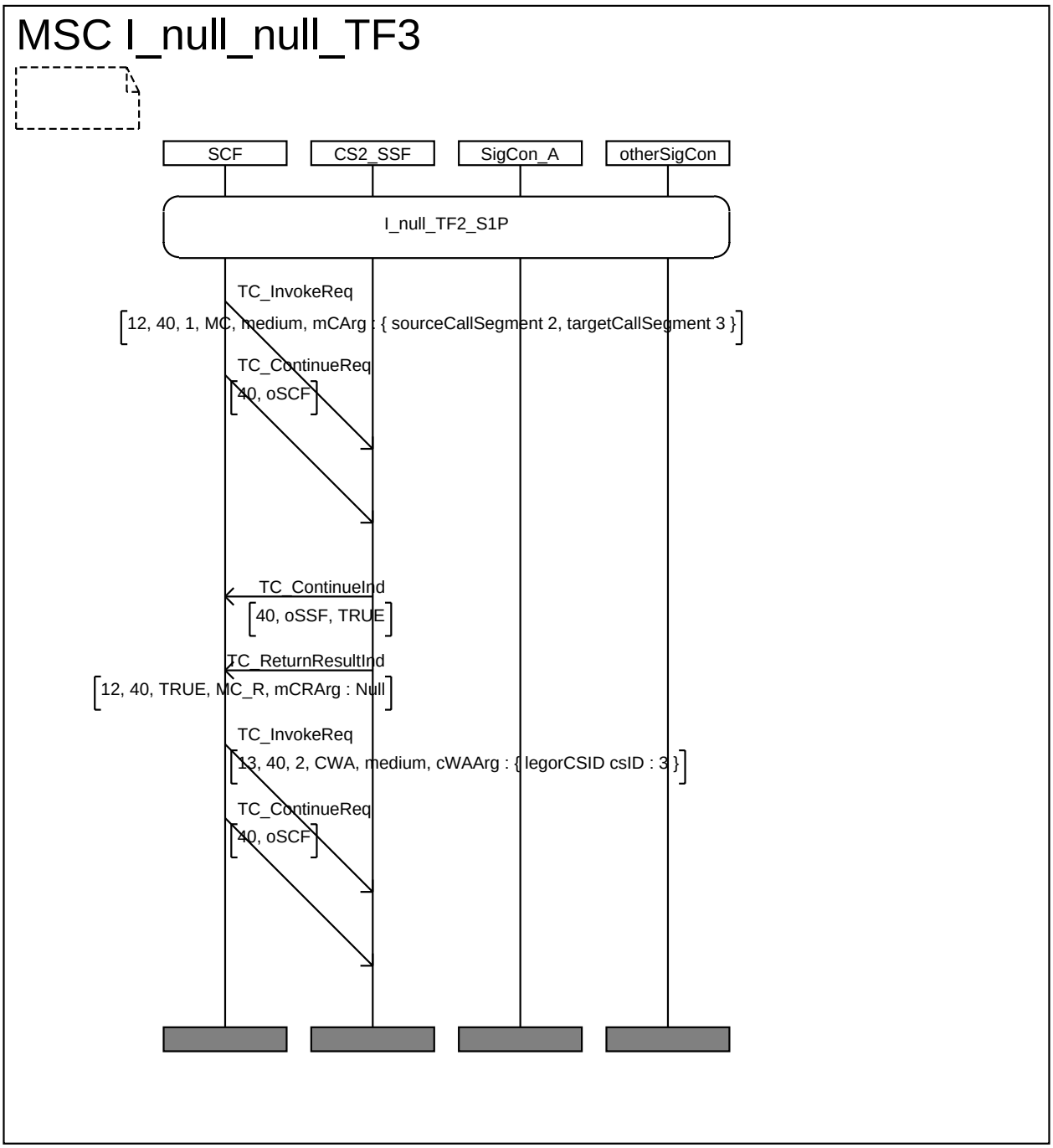
I_null_TF(2)_S1P

L1! MergeCallSegments(2,3)

L1?MergeCallSegmentsReturnResult

L1! ContinueWithArgument(CSId=3)

I_null_null_TF(3)



3-2 - Preamble I_null_TF(2)_null

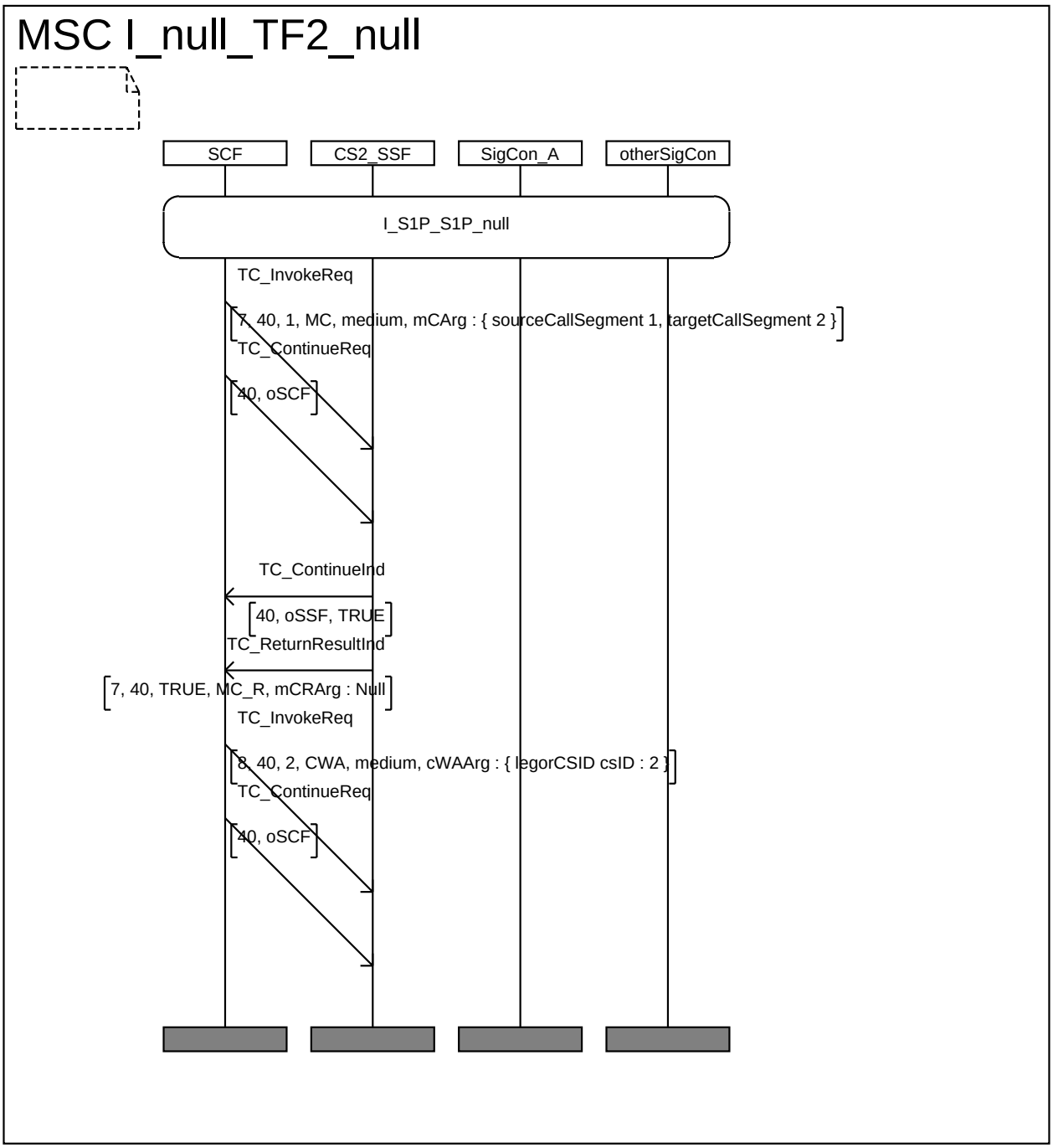
I_S1P_S1P_null

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.

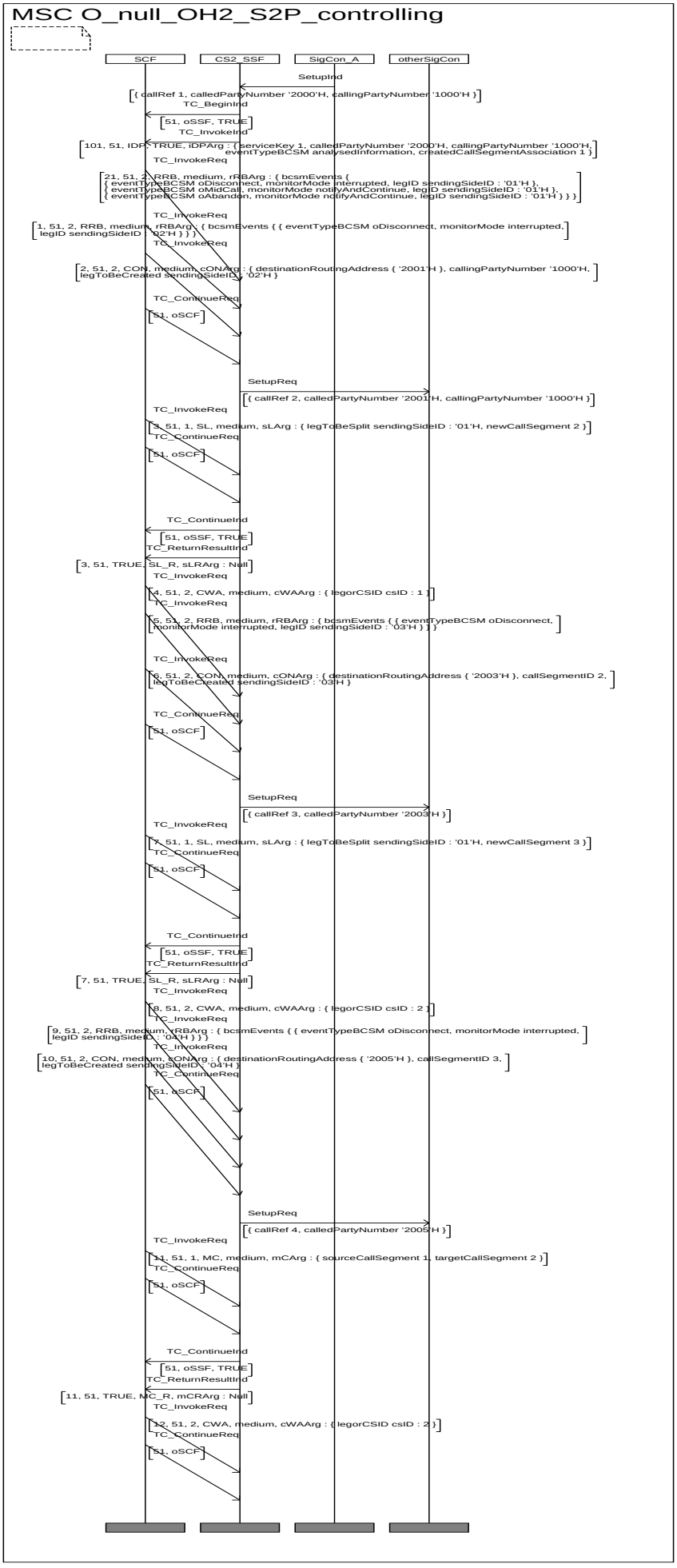
Preamble O_null_OH(2)_S2P_controlling

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

oDisconnect;

oAbandon;

oMidCall.


MSC O_null_OH2_S2P_controlling
SCF
CS2_SSF
SigCon_A
otherSigCon
SetupInd
{ callRef 1, calledPartyNumber '2000'H, callingPartyNumber '1000'H }
TC_BeginInd
51, oSSF, TRUE
TC_InvokeInd
101, 51, IDP, TRUE, iDPArg : { serviceKey 1, calledPartyNumber '2000'H, callingPartyNumber '1000'H, eventTypeBCSM analysedInformation, createdCallSegmentAssociation 1 }
TC_InvokeReq
21, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '01'H }, { eventTypeBCSM oMidCall, monitorMode notifyAndContinue, legID sendingSideID : '01'H }, { eventTypeBCSM oAbandon, monitorMode notifyAndContinue, legID sendingSideID : '01'H } } }
TC_InvokeReq
1, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '02'H } } }
TC_InvokeReq
2, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2001'H }, callingPartyNumber '1000'H, legToBeCreated sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1000'H }
TC_InvokeReq
3, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
3, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
4, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
5, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
6, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
TC_InvokeReq
7, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
7, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
8, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
9, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '04'H } } }
TC_InvokeReq
10, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2005'H }, callSegmentID 3, legToBeCreated sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2005'H }
TC_InvokeReq
11, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
11, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
12, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF

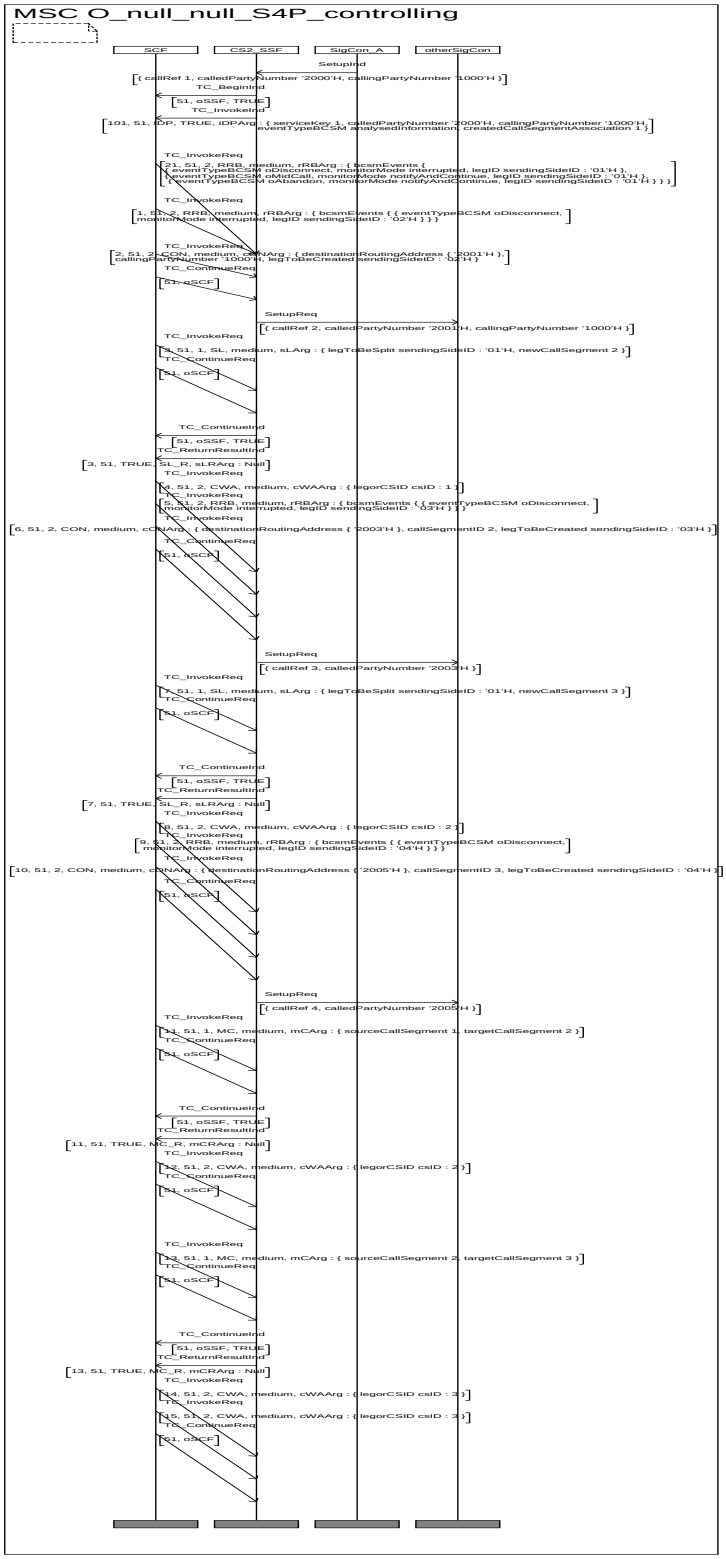
Preamble O_null_null_S4P_controlling

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

oDisconnect;

oAbandon;

oMidCall.



Preamble O_null_OH(2)_S2P_passive

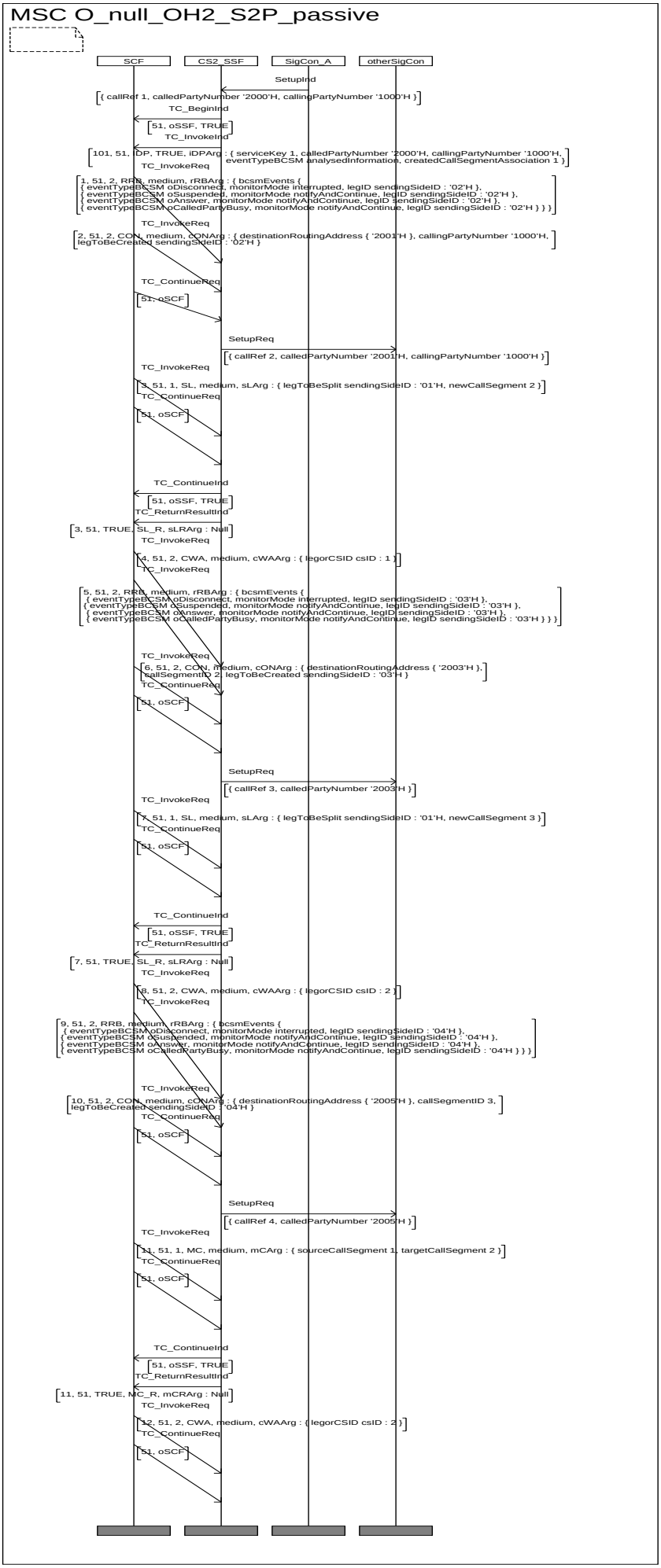
The passive legs (legId=2,3,4) are armed with the events:

oDisconnect;

oAnswer;

oCalledPartyBusy;

oSuspended.


MSC O_null_OH2_S2P_passive
SCF
CS2_SSF
SigCon_A
otherSigCon
SetupInd
{ callRef 1, calledPartyNumber '2000'H, callingPartyNumber '1000'H }
TC_BeginInd
51, oSSF, TRUE
TC_InvokeInd
101, 51, DP, TRUE, iDPArg : { serviceKey 1, calledPartyNumber '2000'H, callingPartyNumber '1000'H, eventTypeBCSM analysedInformation, createdCallSegmentAssociation 1 }
TC_InvokeReq
1, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '02'H }, { eventTypeBCSM oSuspended, monitorMode notifyAndContinue, legID sendingSideID : '02'H }, { eventTypeBCSM oAnswer, monitorMode notifyAndContinue, legID sendingSideID : '02'H }, { eventTypeBCSM oCalledPartyBusy, monitorMode notifyAndContinue, legID sendingSideID : '02'H } } }
TC_InvokeReq
2, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2001'H }, callingPartyNumber '1000'H, legToBeCreated sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1000'H }
TC_InvokeReq
3, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
3, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
4, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
5, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H }, { eventTypeBCSM oSuspended, monitorMode notifyAndContinue, legID sendingSideID : '03'H }, { eventTypeBCSM oAnswer, monitorMode notifyAndContinue, legID sendingSideID : '03'H }, { eventTypeBCSM oCalledPartyBusy, monitorMode notifyAndContinue, legID sendingSideID : '03'H } } }
TC_InvokeReq
6, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
TC_InvokeReq
7, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
7, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
8, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
9, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '04'H }, { eventTypeBCSM oSuspended, monitorMode notifyAndContinue, legID sendingSideID : '04'H }, { eventTypeBCSM oAnswer, monitorMode notifyAndContinue, legID sendingSideID : '04'H }, { eventTypeBCSM oCalledPartyBusy, monitorMode notifyAndContinue, legID sendingSideID : '04'H } } }
TC_InvokeReq
10, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2005'H }, callSegmentID 3, legToBeCreated sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2005'H }
TC_InvokeReq
11, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
11, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
12, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF

Preamble O_null_null_S4P_passive

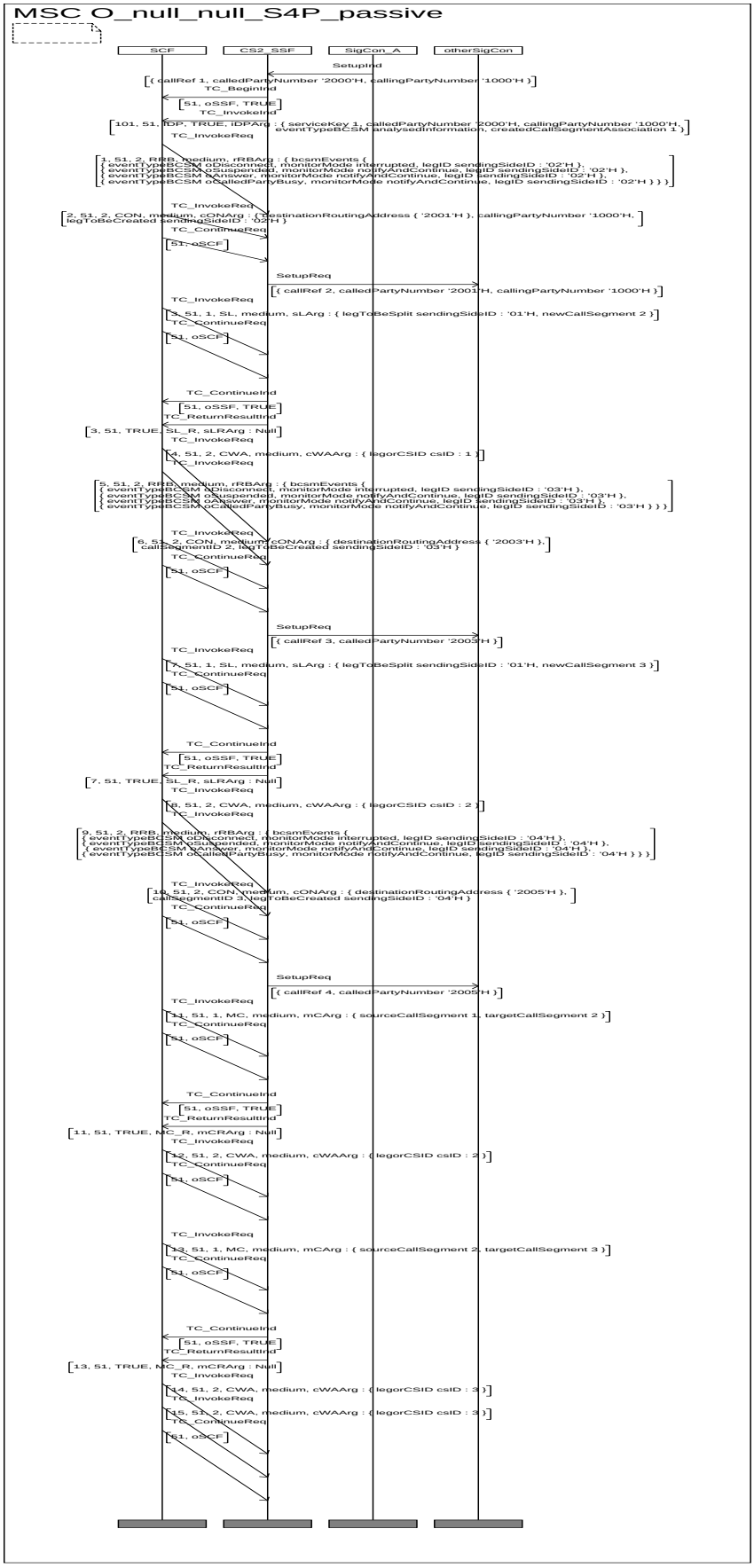
The passive legs (legId=2,3,4) are armed with the events:

oDisconnect;

oAnswer;

oCalledPartyBusy;

oSuspended.



Preamble I_S1P_S1P_null_passive

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

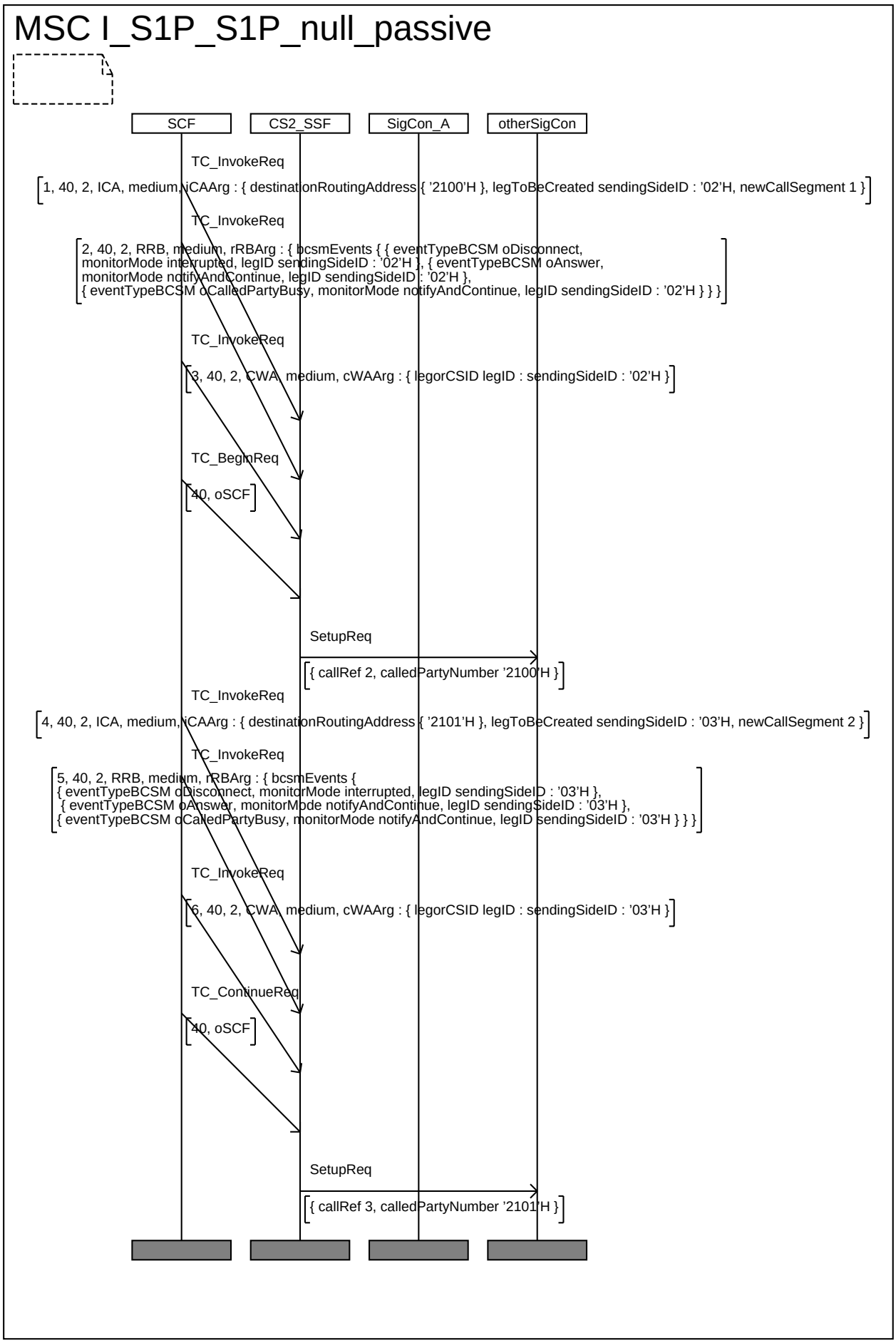
oDisconnect;

oMidCall;

oAnswer;

oCalledPartyBusy;

oSuspended.


MSC I_S1P_S1P_null_passive
SCF
CS2_SSF
SigCon_A
otherSigCon
TC_InvokeReq
1, 40, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2100'H }, legToBeCreated sendingSideID : '02'H, newCallSegment 1 }
TC_InvokeReq
2, 40, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '02'H }, { eventTypeBCSM oAnswer, monitorMode notifyAndContinue, legID sendingSideID : '02'H }, { eventTypeBCSM oCalledPartyBusy, monitorMode notifyAndContinue, legID sendingSideID : '02'H } } }
TC_InvokeReq
3, 40, 2, CWA, medium, cWAArg : { legorCSID legID : sendingSideID : '02'H }
TC_BeginReq
40, oSCF
SetupReq
{ callRef 2, calledPartyNumber '2100'H }
TC_InvokeReq
4, 40, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2101'H }, legToBeCreated sendingSideID : '03'H, newCallSegment 2 }
TC_InvokeReq
5, 40, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H }, { eventTypeBCSM oAnswer, monitorMode notifyAndContinue, legID sendingSideID : '03'H }, { eventTypeBCSM oCalledPartyBusy, monitorMode notifyAndContinue, legID sendingSideID : '03'H } } }
TC_InvokeReq
6, 40, 2, CWA, medium, cWAArg : { legorCSID legID : sendingSideID : '03'H }
TC_ContinueReq
40, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2101'H }

Preamble I_null_TF(2)_null_passive

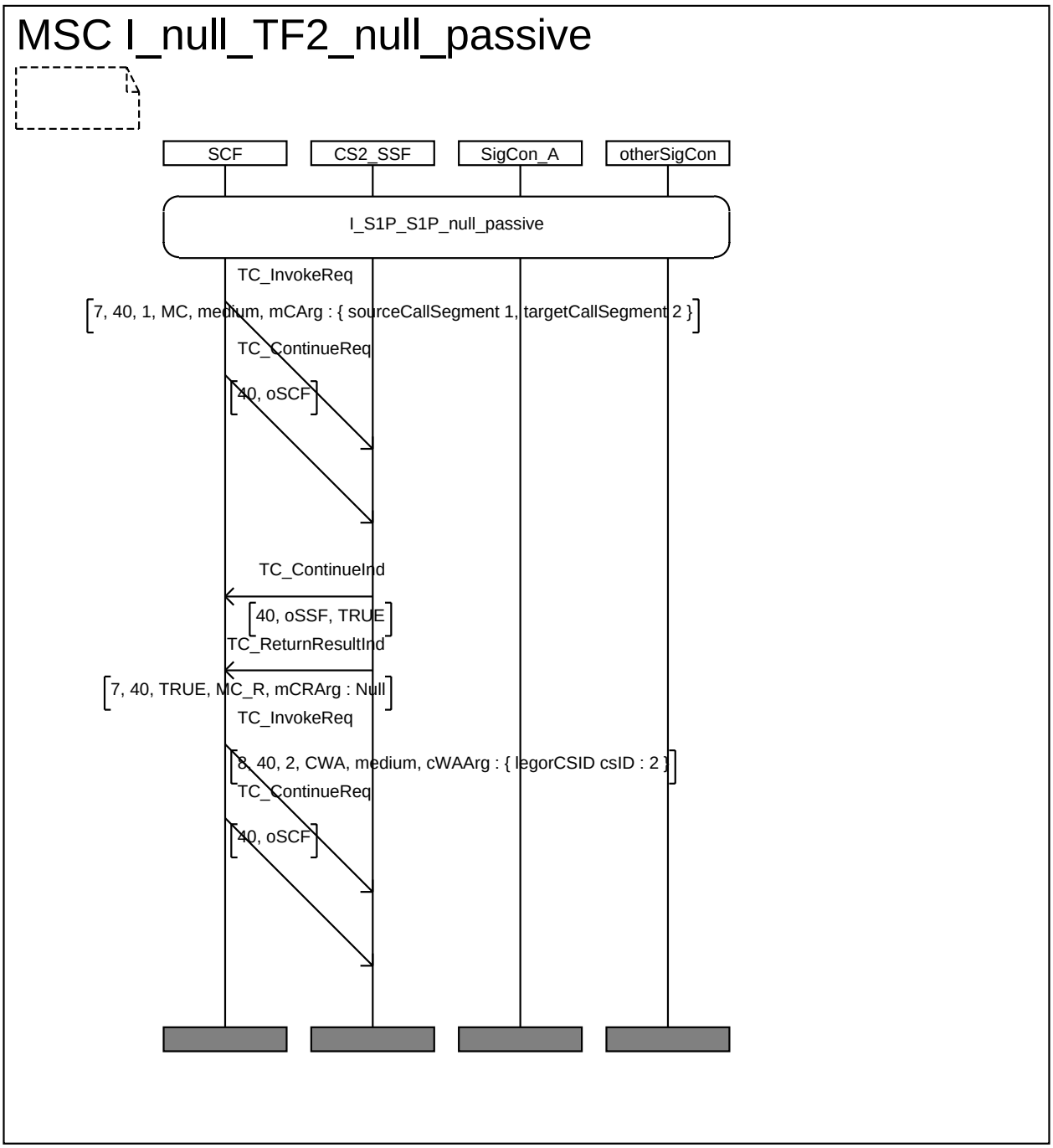
The passive legs (legId=2,3) are armed with the events:

oDisconnect;

oAnswer;

oCalledPartyBusy;

oSuspended.



Preamble T_S2P_null_null_controlling

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

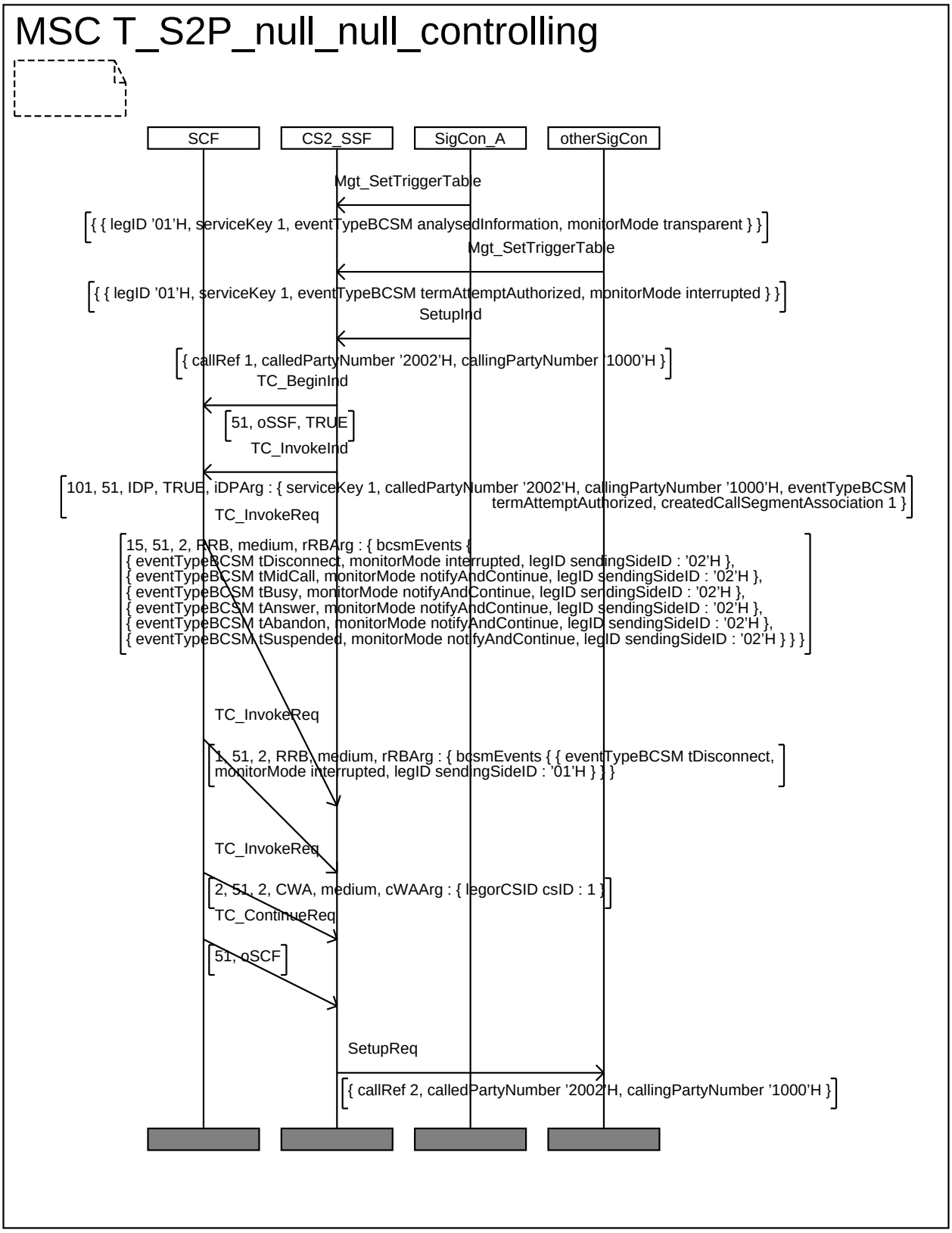
tDisconnect;

tMidCall;

tAnswer;

tCalledPartyBusy;

tSuspended.



Preamble T_null_null_S4P_controlling

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

tDisconnect;

tMidCall;

tAnswer;

tCalledPartyBusy;

tSuspended.

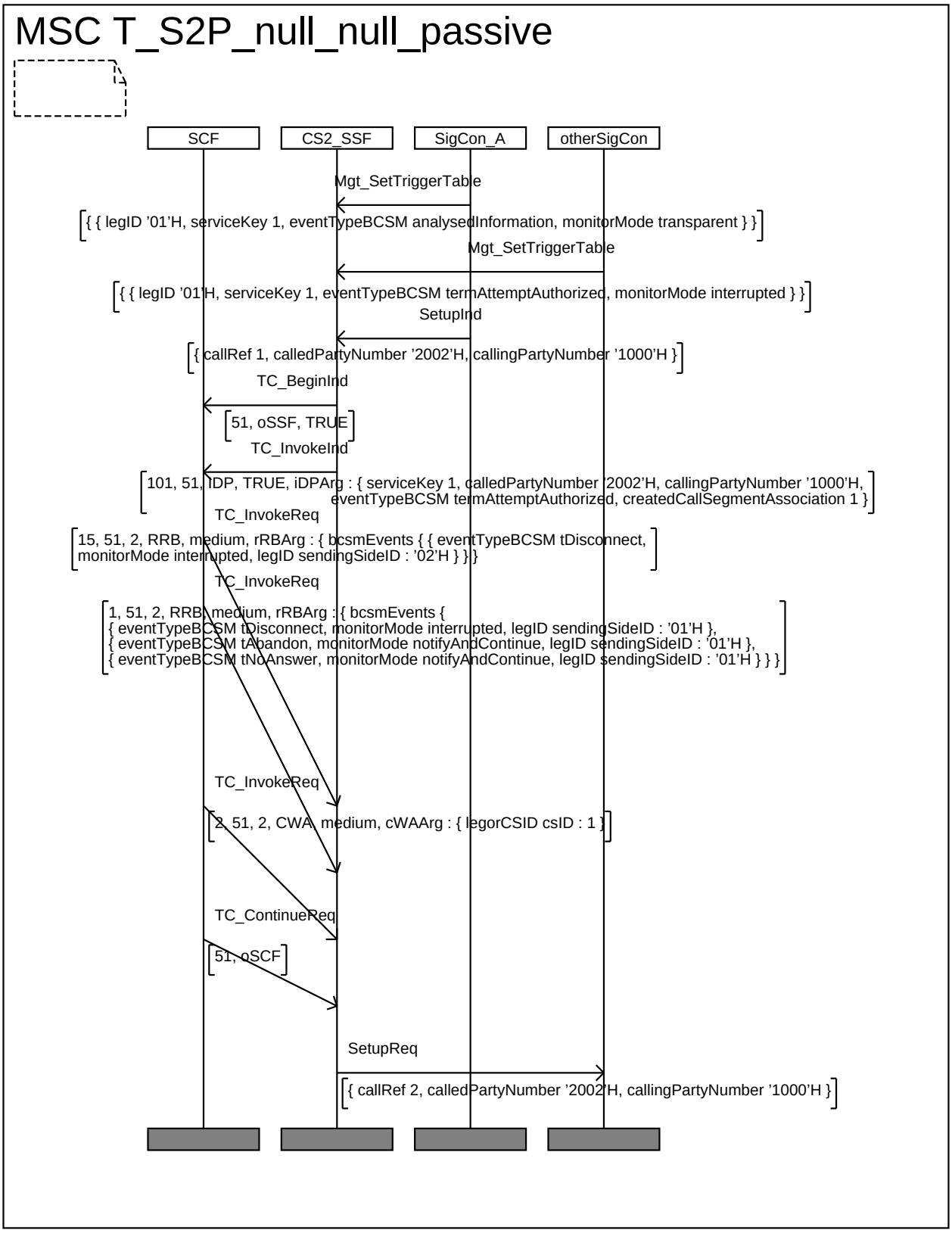
Preamble T_S2P_null_null_passive

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

tDisconnect;

tAbandon;

tAnswer.


MSC T_S2P_null_null_passive
SCF
CS2_SSF
SigCon_A
otherSigCon
Mgt_SetTriggerTable
{ { legID '01'H, serviceKey 1, eventTypeBCSM analysedInformation, monitorMode transparent } }
Mgt_SetTriggerTable
{ { legID '01'H, serviceKey 1, eventTypeBCSM termAttemptAuthorized, monitorMode interrupted } }
SetupInd
{ callRef 1, calledPartyNumber '2002'H, callingPartyNumber '1000'H }
TC_BeginInd
51, oSSF, TRUE
TC_InvokeInd
101, 51, IDP, TRUE, iDPArg : { serviceKey 1, calledPartyNumber '2002'H, callingPartyNumber '1000'H, eventTypeBCSM termAttemptAuthorized, createdCallSegmentAssociation 1 }
TC_InvokeReq
15, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM tDisconnect, monitorMode interrupted, legID sendingSideID : '02'H } } }
TC_InvokeReq
1, 51, 2, RRB, medium, rRBArg : { bcsmEvents {
{ eventTypeBCSM tDisconnect, monitorMode interrupted, legID sendingSideID : '01'H },
{ eventTypeBCSM tAbandon, monitorMode notifyAndContinue, legID sendingSideID : '01'H },
{ eventTypeBCSM tNoAnswer, monitorMode notifyAndContinue, legID sendingSideID : '01'H } } }
TC_InvokeReq
2, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 2, calledPartyNumber '2002'H, callingPartyNumber '1000'H }

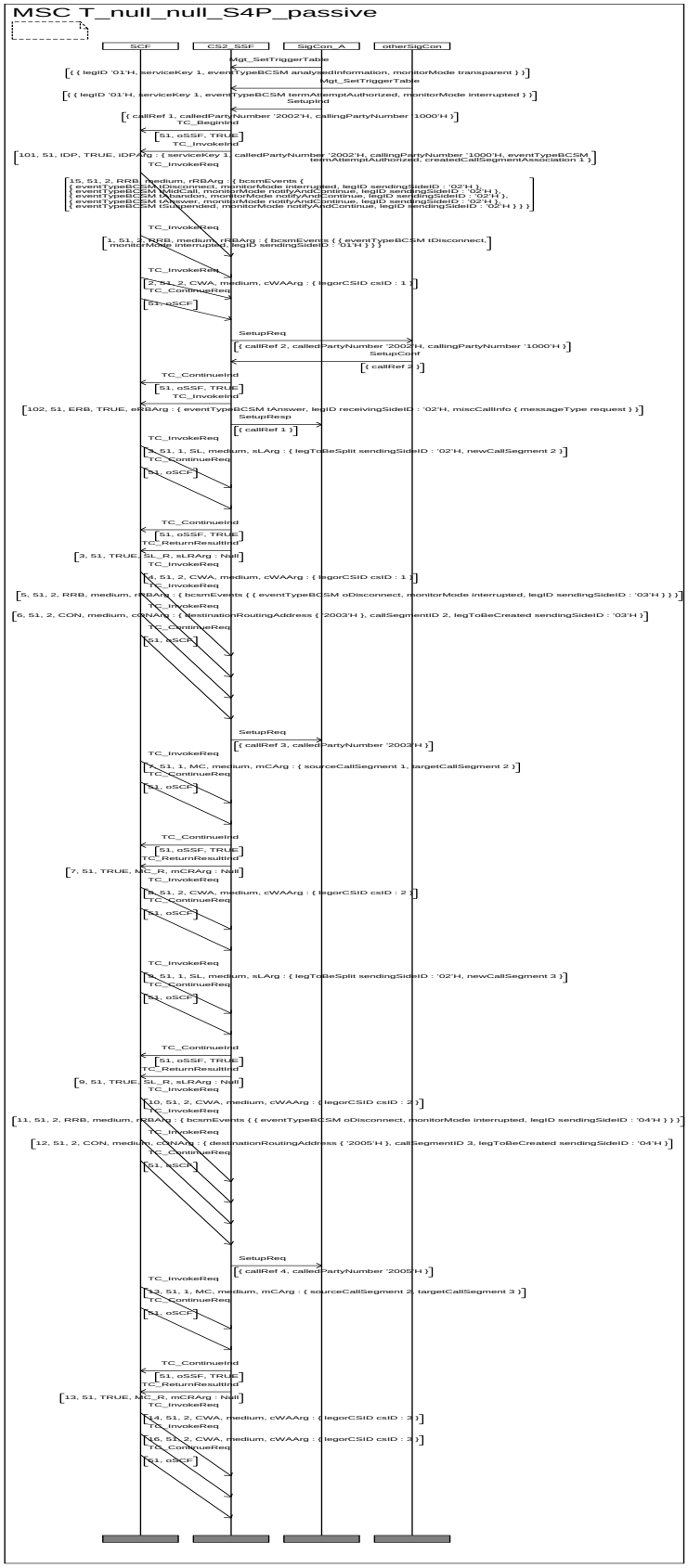
Preamble T_null_null_S4P_passive

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

t/oDisconnect;

t/oAbandon;

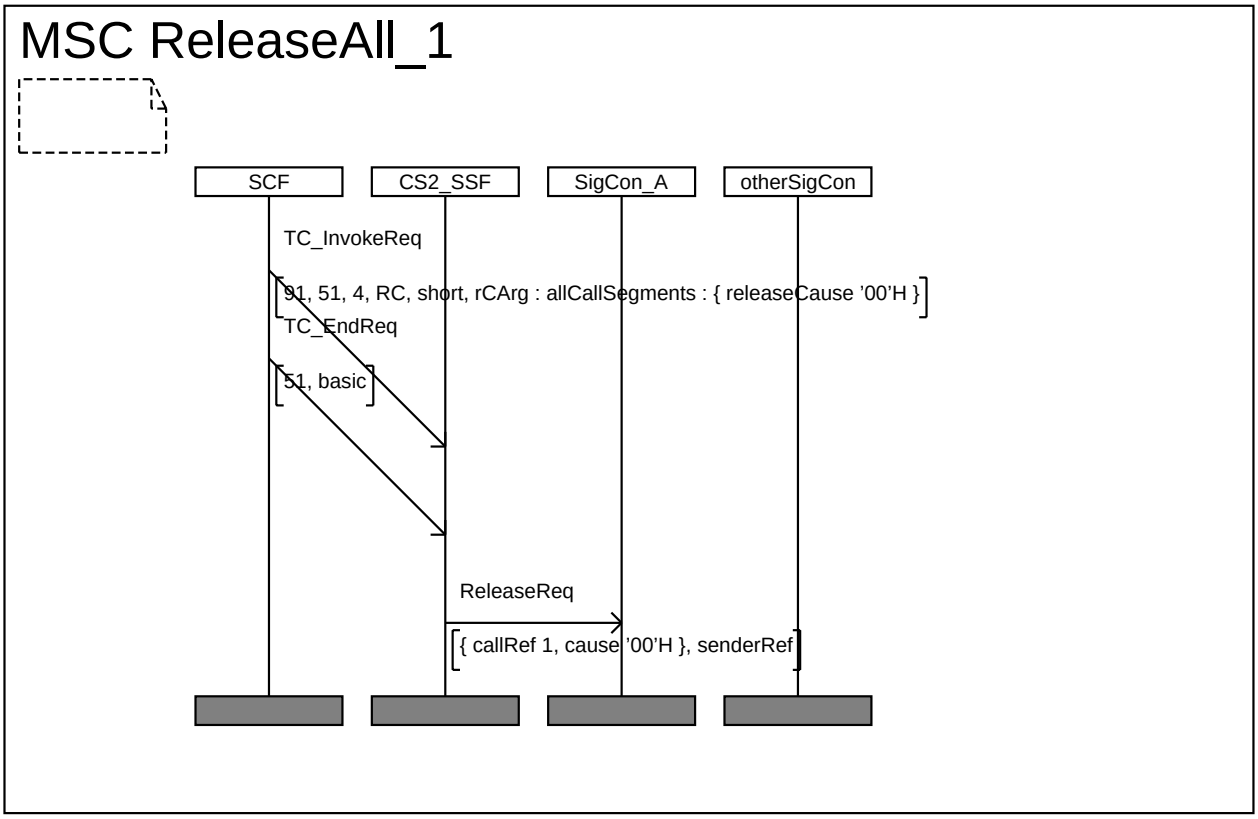
t/oAnswer.


MSC T_null_null_S4P_passive
SCF
CS2_SSF
SigCon_A
otherSigCon
Mgt_SetTriggerTable
[{ { legID '01'H, serviceKey 1, eventTypeBCSM analysedInformation, monitorMode transparent } }]
Mgt_SetTriggerTable
[{ { legID '01'H, serviceKey 1, eventTypeBCSM termAttemptAuthorized, monitorMode interrupted } }]
SetupInd
[{ callRef 1, calledPartyNumber '2002'H, callingPartyNumber '1000'H }]
TC_BeginInd
[51, oSSF, TRUE]
TC_InvokeInd
[101, 51, IDP, TRUE, iDPArg : { serviceKey 1, calledPartyNumber '2002'H, callingPartyNumber '1000'H, eventTypeBCSM termAttemptAuthorized, createdCallSegmentAssociation 1 }]
TC_InvokeReq
[15, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM tDisconnect, monitorMode interrupted, legID sendingSideID : '02'H }, { eventTypeBCSM tMidCall, monitorMode notifyAndContinue, legID sendingSideID : '02'H }, { eventTypeBCSM tAbandon, monitorMode notifyAndContinue, legID sendingSideID : '02'H }, { eventTypeBCSM tAnswer, monitorMode notifyAndContinue, legID sendingSideID : '02'H }, { eventTypeBCSM tSuspended, monitorMode notifyAndContinue, legID sendingSideID : '02'H } } }]
TC_InvokeReq
[1, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM tDisconnect, monitorMode interrupted, legID sendingSideID : '01'H } } }]
TC_InvokeReq
[2, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }]
TC_ContinueReq
[51, oSCF]
SetupReq
[{ callRef 2, calledPartyNumber '2002'H, callingPartyNumber '1000'H }]
SetupConf
[{ callRef 2 }]
TC_ContinueInd
[51, oSSF, TRUE]
TC_InvokeInd
[102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM tAnswer, legID receivingSideID : '02'H, miscCallInfo { messageType request } }]
SetupResp
[{ callRef 1 }]
TC_InvokeReq
[3, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[3, 51, TRUE, SL_R, sLRArg : Null]
TC_InvokeReq
[4, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }]
TC_InvokeReq
[5, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H } } }]
TC_InvokeReq
[6, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }]
TC_ContinueReq
[51, oSCF]
SetupReq
[{ callRef 3, calledPartyNumber '2003'H }]
TC_InvokeReq
[7, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[7, 51, TRUE, MC_R, mCRArg : Null]
TC_InvokeReq
[8, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_ContinueReq
[51, oSCF]
TC_InvokeReq
[9, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 3 }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[9, 51, TRUE, SL_R, sLRArg : Null]
TC_InvokeReq
[10, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_InvokeReq
[11, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '04'H } } }]
TC_InvokeReq
[12, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2005'H }, callSegmentID 3, legToBeCreated sendingSideID : '04'H }]
TC_ContinueReq
[51, oSCF]
SetupReq
[{ callRef 4, calledPartyNumber '2005'H }]
TC_InvokeReq
[13, 51, 1, MC, medium, mCArg : { sourceCallSegment 2, targetCallSegment 3 }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[13, 51, TRUE, MC_R, mCRArg : Null]
TC_InvokeReq
[14, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_InvokeReq
[16, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_ContinueReq
[51, oSCF]

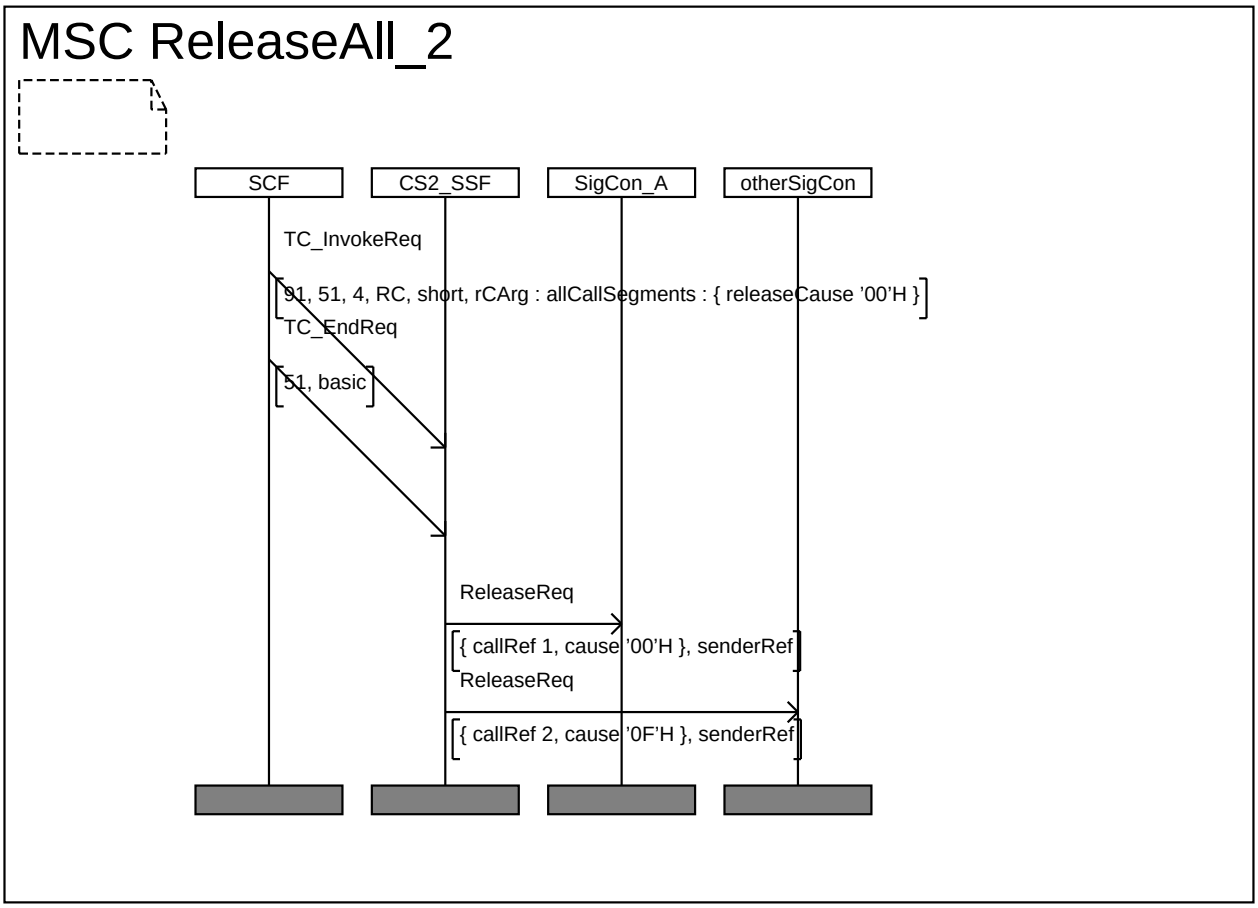
7.1.7 Postamble descriptions

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.

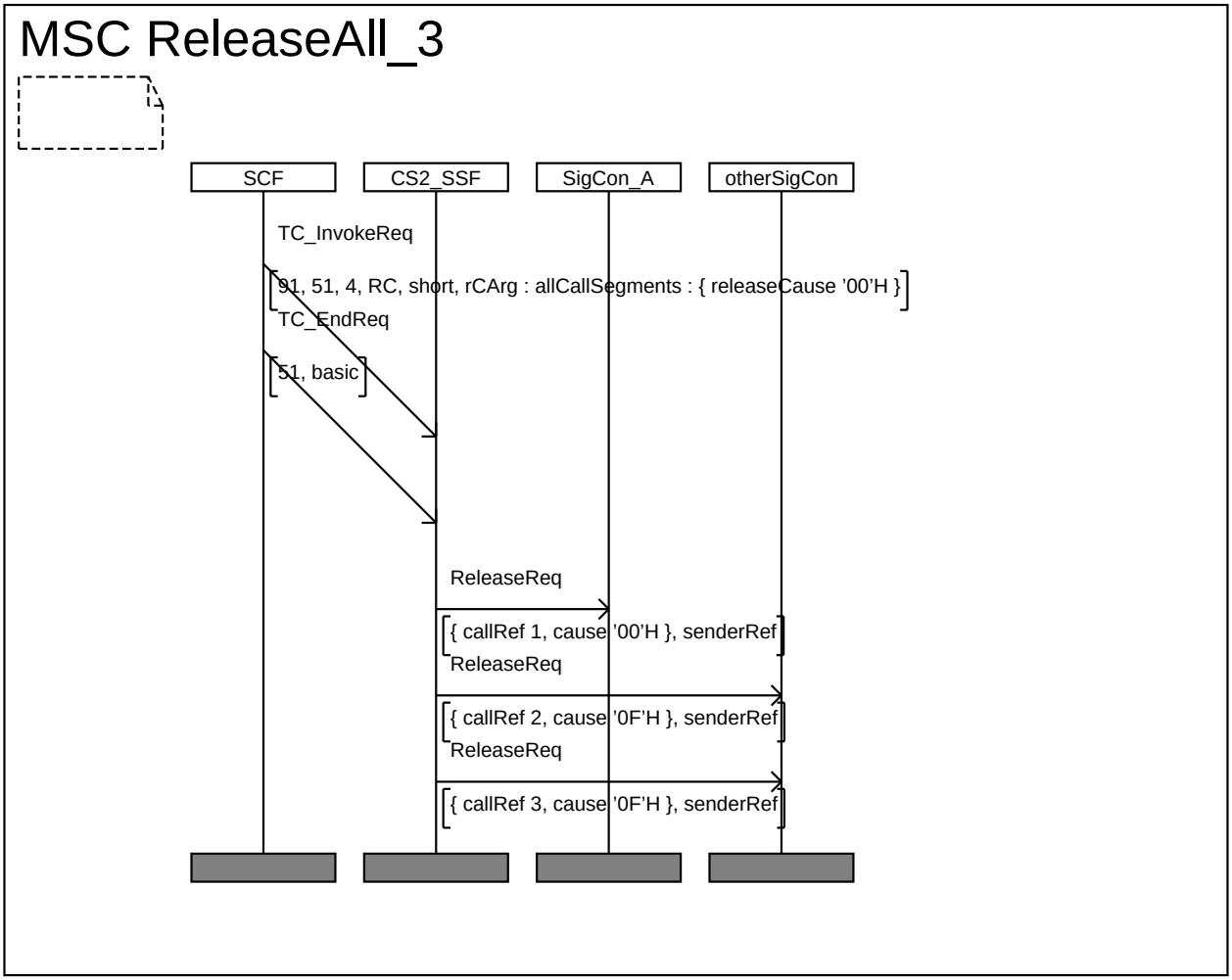
7.1.7.1 Postamble ReleaseAll_1



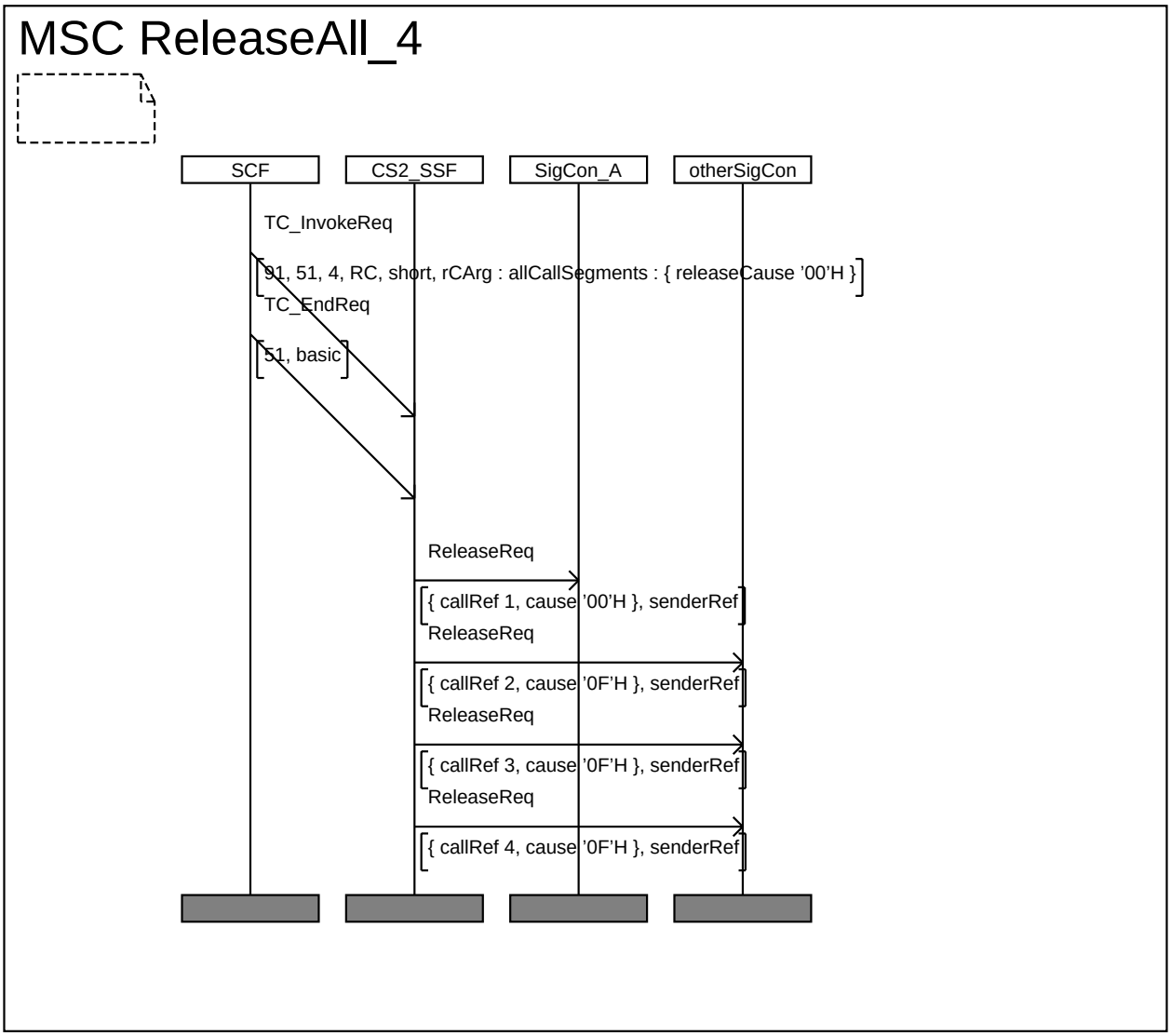
7.1.7.2 Postamble ReleaseAll_2



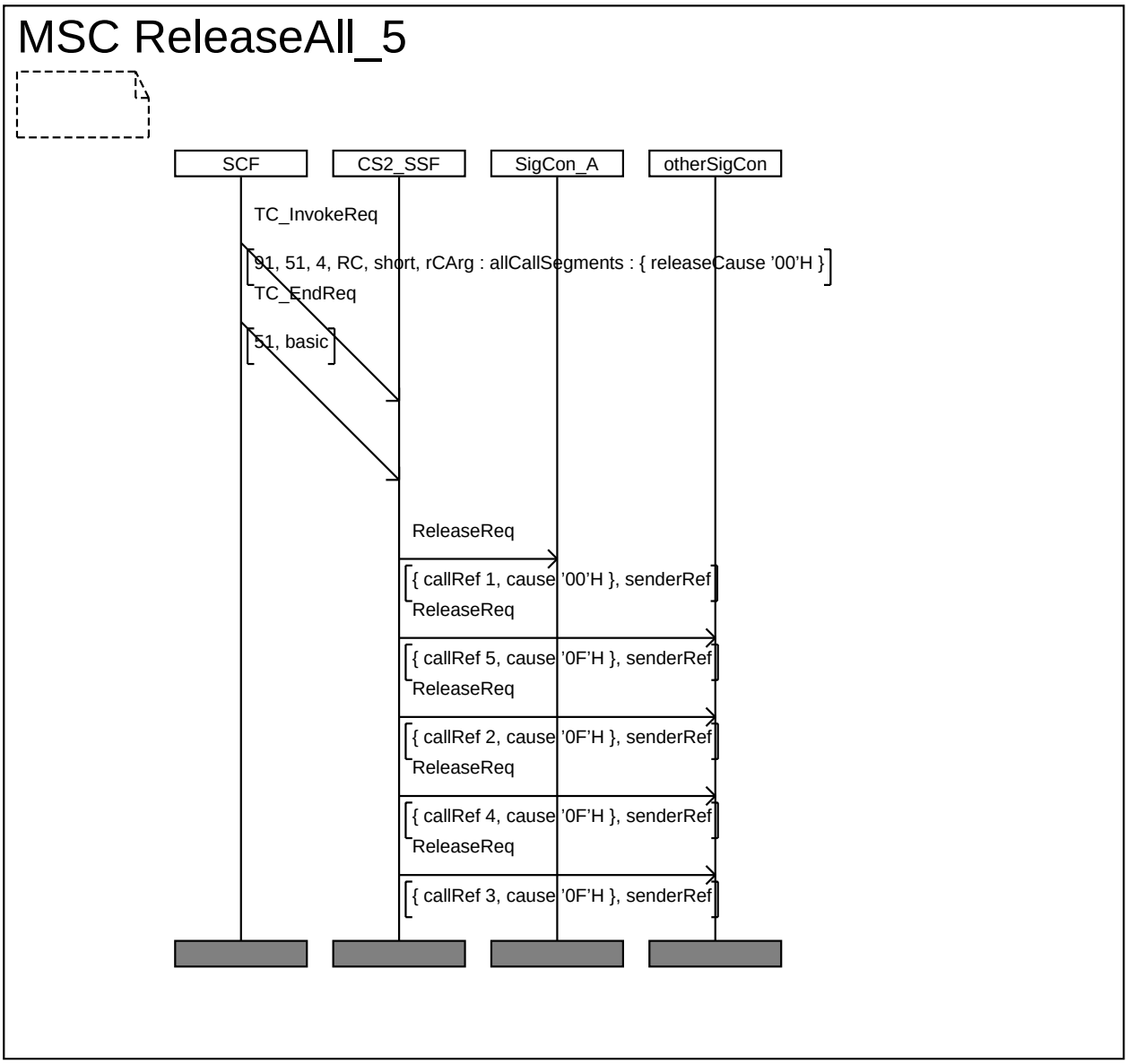
7.1.7.3 Postamble ReleaseAll_3



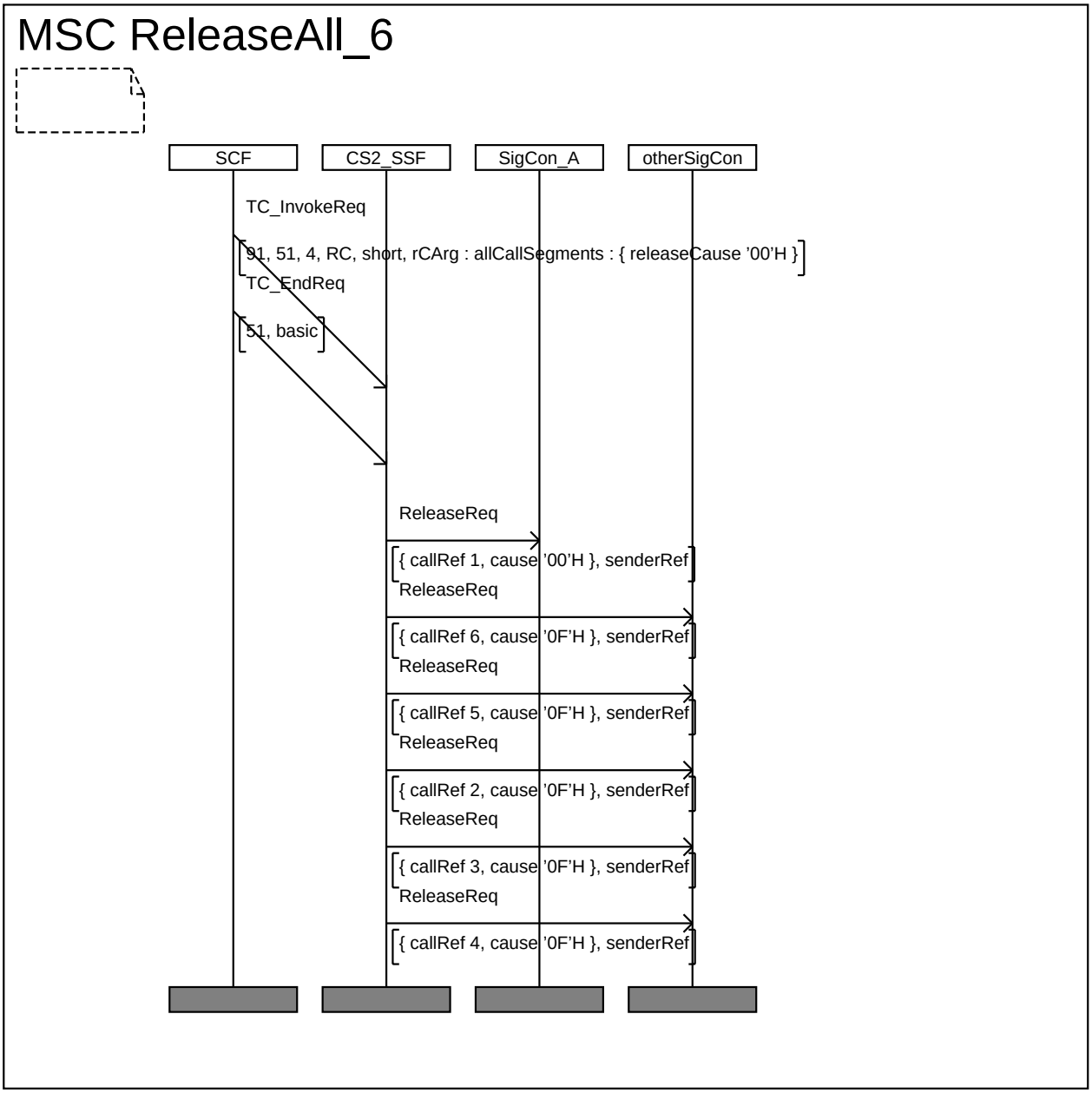
7.1.7.4 Postamble ReleaseAll_4



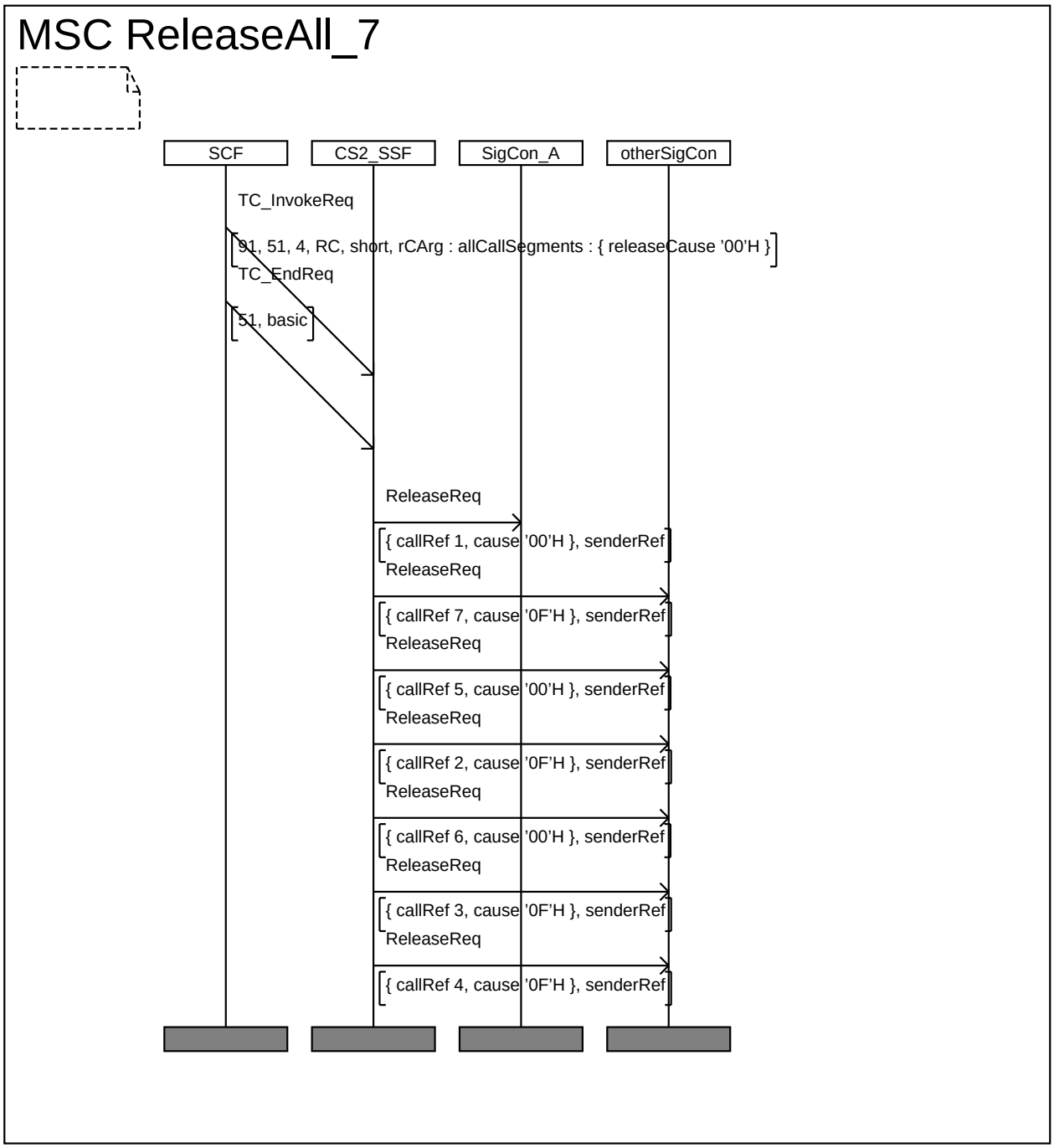
7.1.7.5 Postamble ReleaseAll_5



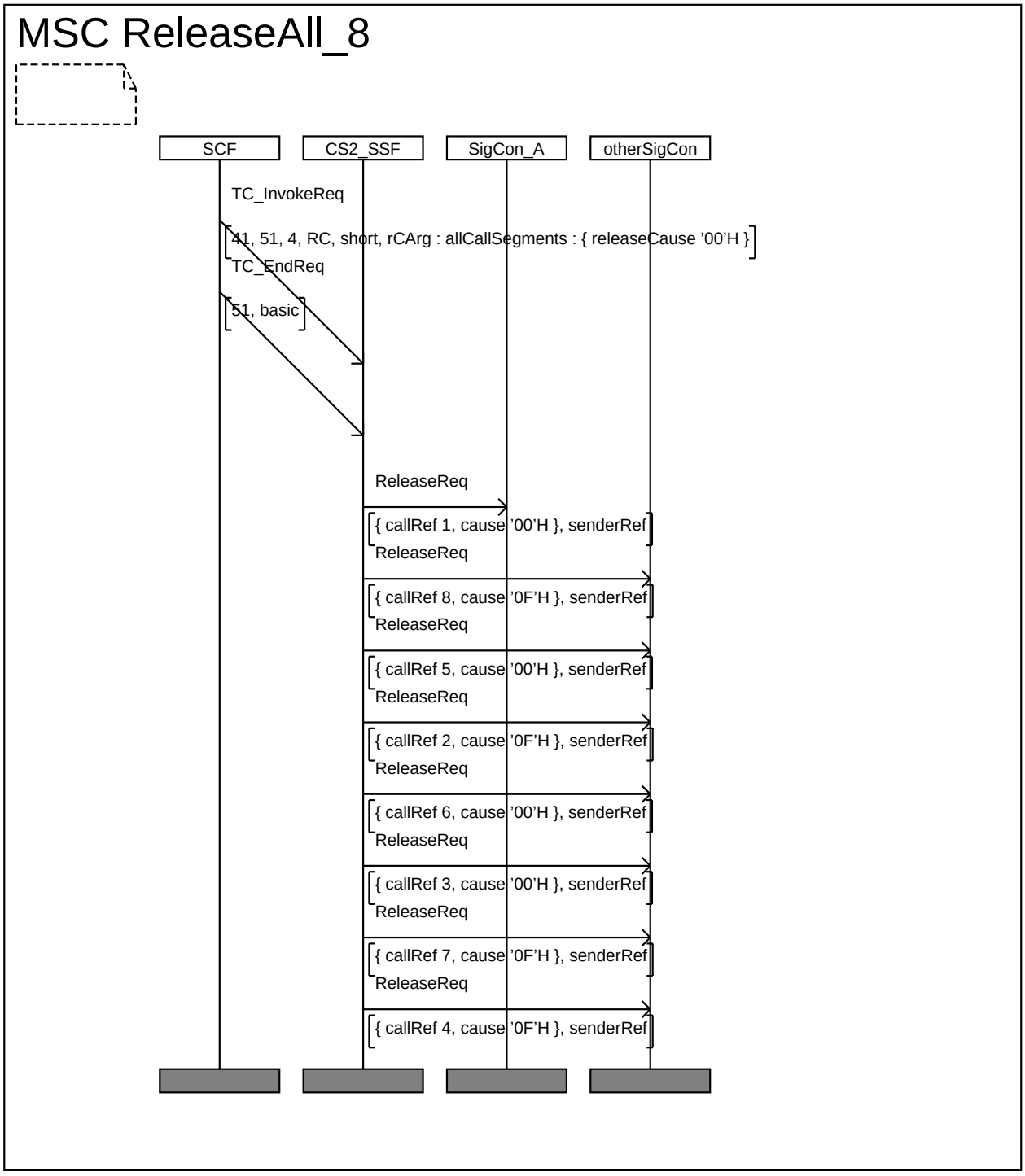
7.1.7.6 Postamble ReleaseAll_6



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	RequestReportBCSMEvent (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

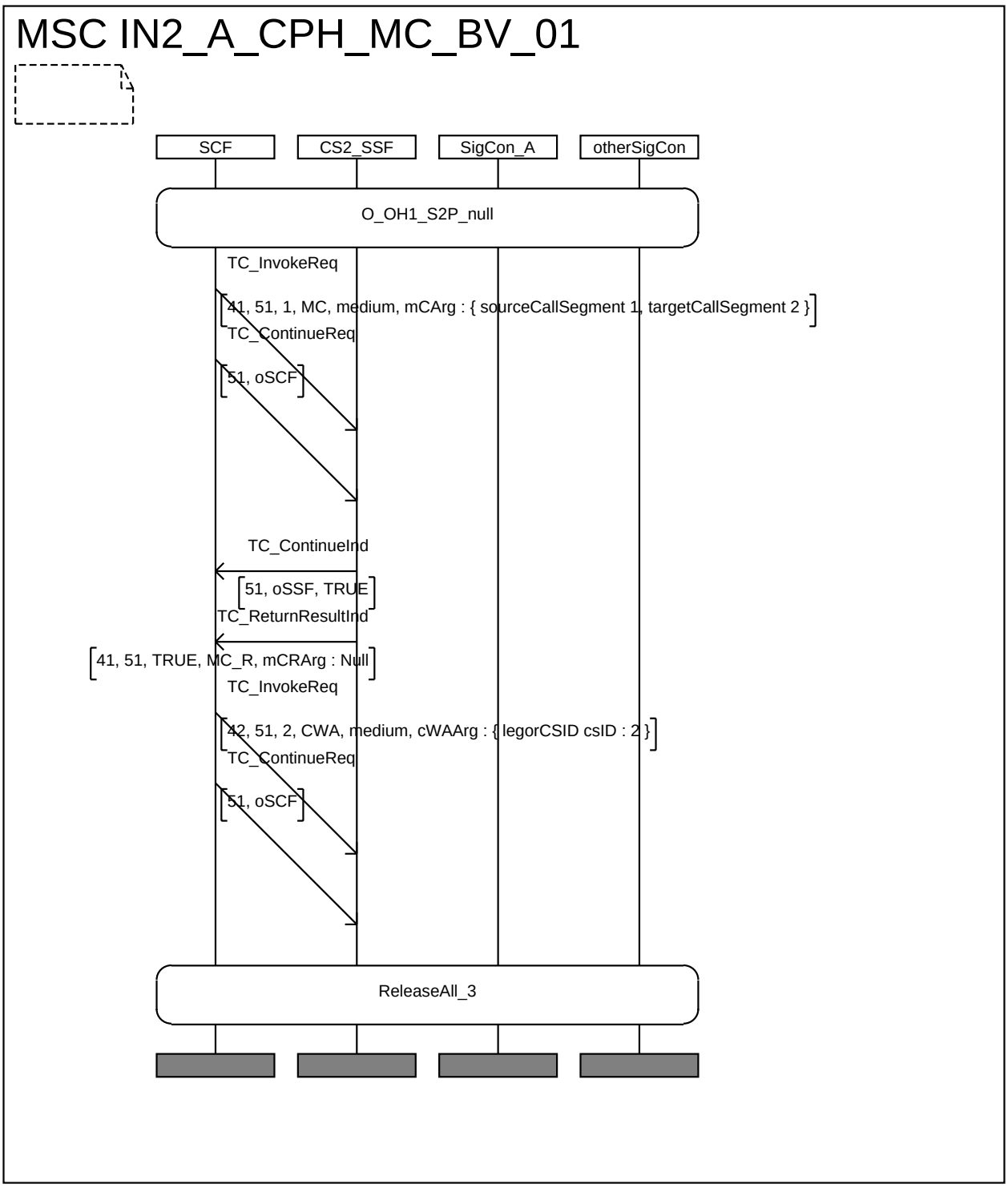
Total: 24

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

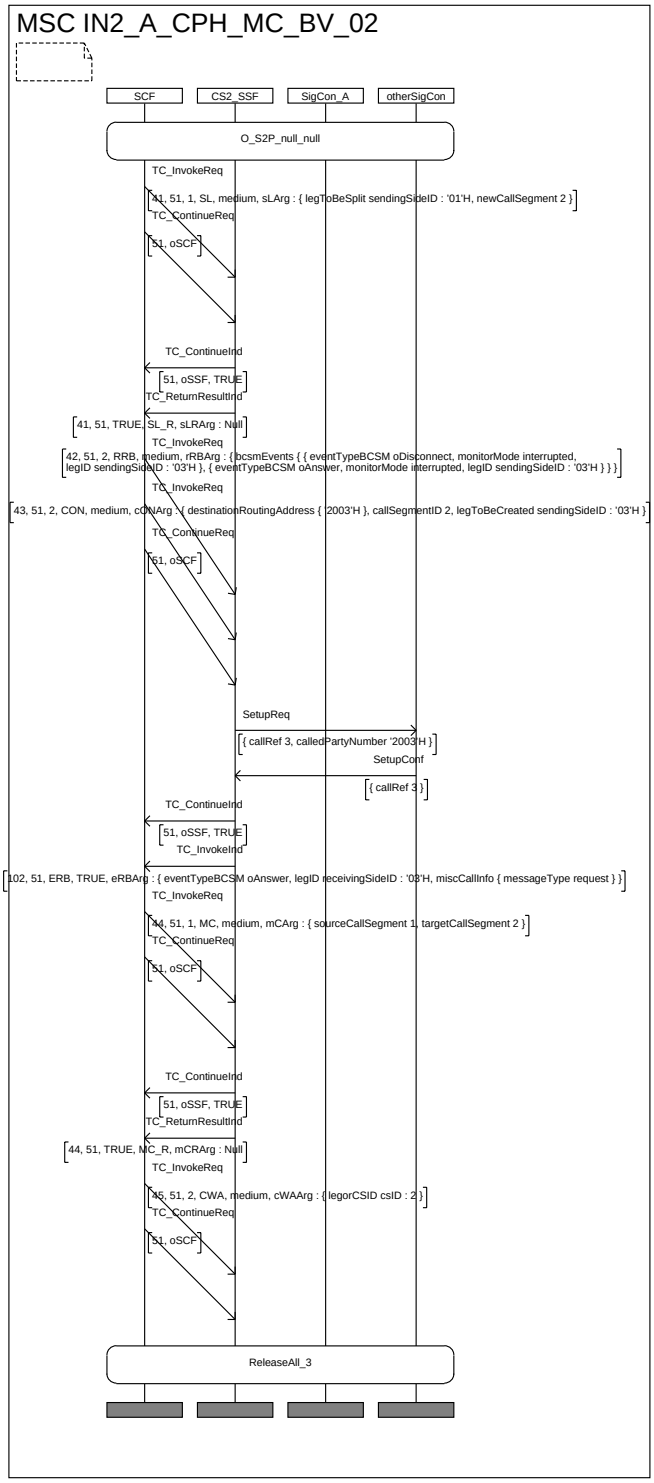
SigCon A, B, C are the signalling controls for users A, B, C, etc and the IUT is a SSF while the main lower tester is an SCF.

7.4.1 MergeCallSegment procedure (MC)

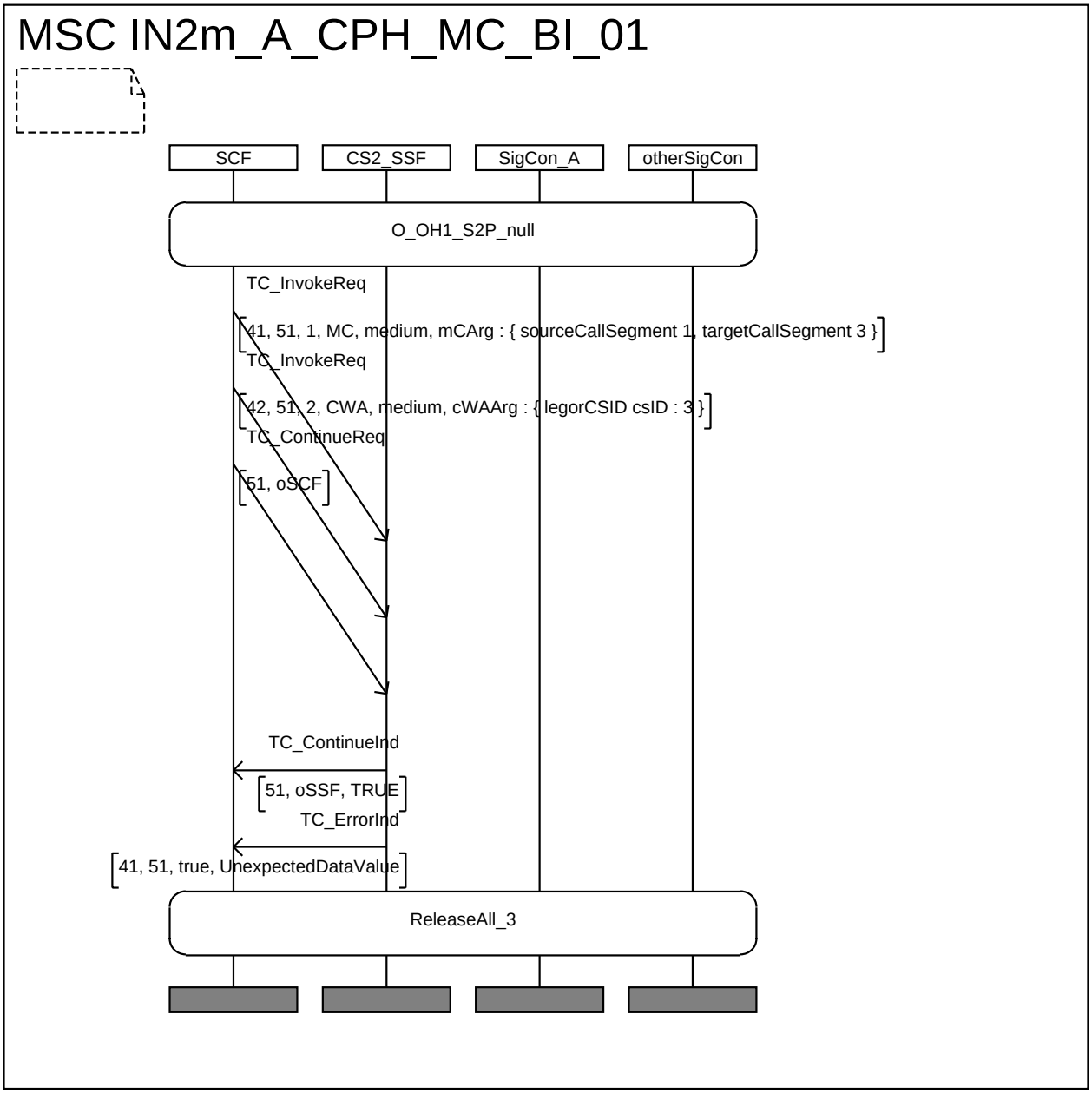
IN2_A_CPH_MC_BV_01	
Purpose:	test MergeCallSegment functionality
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **MergeCS_invoke** to SSF with the following parameters: - sourceCS=1 - targetCS=2 followed by **ContinueWithArgument** with parameter csID = 2
Pass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


MSC IN2_A_CPH_MC_BV_01
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_3

IN2_A_CPH_MC_BV_02	
Purpose:	test that armed DP are kept after MergeCS
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	Arm of the DP, then establishment of configuration: SCF sends to SSF a **SplitLeg_invoke** containing at least the parameters: - legToBeSplit=1 - newCallSegment=2 followed by **ContinueWithArgument** on csID=1 followed by **ReqRepBCSMEvent_invoke** containing at least the parameters: - eventTypeBCSM= oDisconnect - monitoringMode= interrupted - legId=3 and - eventTypeBCSM= oAnswer - monitoringMode= interrupted - legId=3 followed by **Connect_invoke** containing at least the parameter: - legToBeCreated=3 - callSegmentID=2 SSF establishes a link to SigCon C (send SetupReq, answered with SetupConf, then SetUpResp to sigconA) then SCF sends **MergeCS_invoke** with parameters: - sourceCS=1 - targetCS=2 followed by **ContinueWithArgument** on csID=2
Pass criteria	Check that SSF sends SplitLegReturnResult, and a MergeCallSegmentsReturnResult 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 **EventReportBCSM**
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_MC_BV_02
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H }, { eventTypeBCSM oAnswer, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
43, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
SetupConf
{ callRef 3 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
TC_InvokeReq
44, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
44, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
45, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_3

IN2_A_CPH_MC_BI_01	
Purpose:	test that MergeCS is rejected when wrong parameter is used
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **MergeCS_invoke** with parameters: - sourceCS=1 - targetCS=3 followed by **ContinueWithArgument** on csID=3
Pass criteria	as CS=3 does not exist, Check that SSF sends to SCF a MergeCS_err containing the parameter: - unexpected data value
Postamble:	ReleaseAll_3



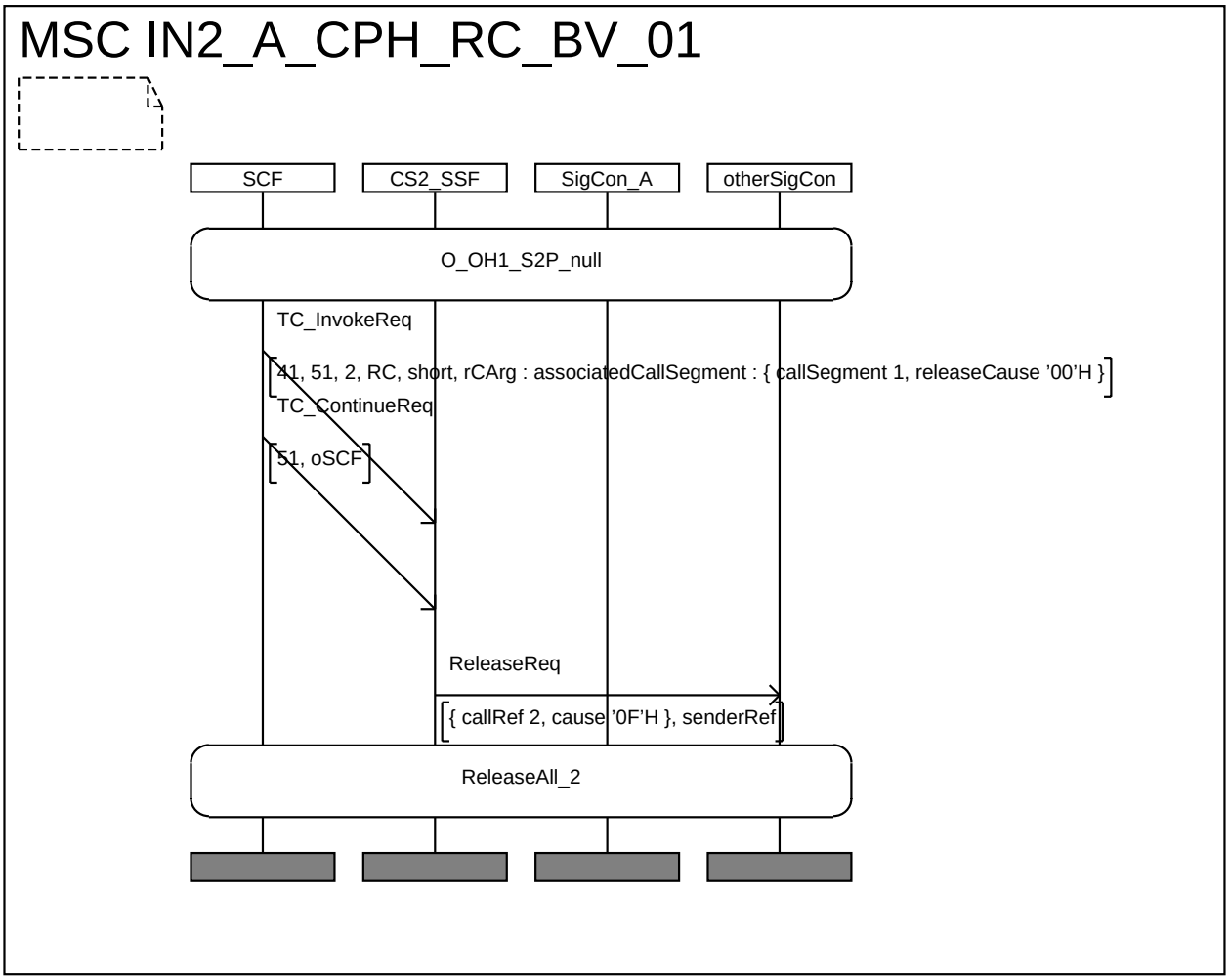
7.4.2 MoveCallSegment procedure (MCS)

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

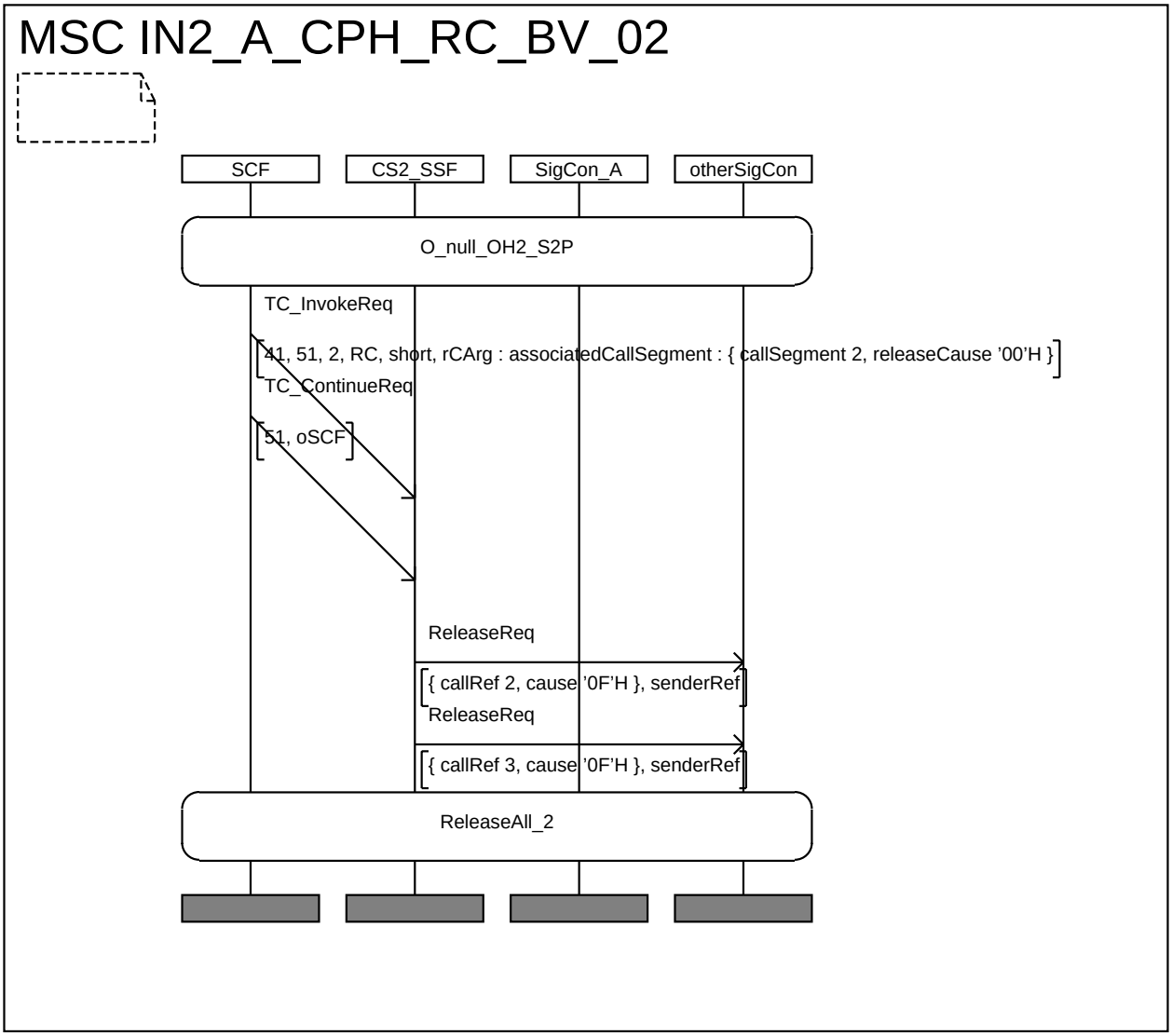
IN2_A_CPH_MCS_BV_01
IN2_A_CPH_MCS_BI_01

7.4.3 ReleaseCall (CS2 complement) procedure (RC)

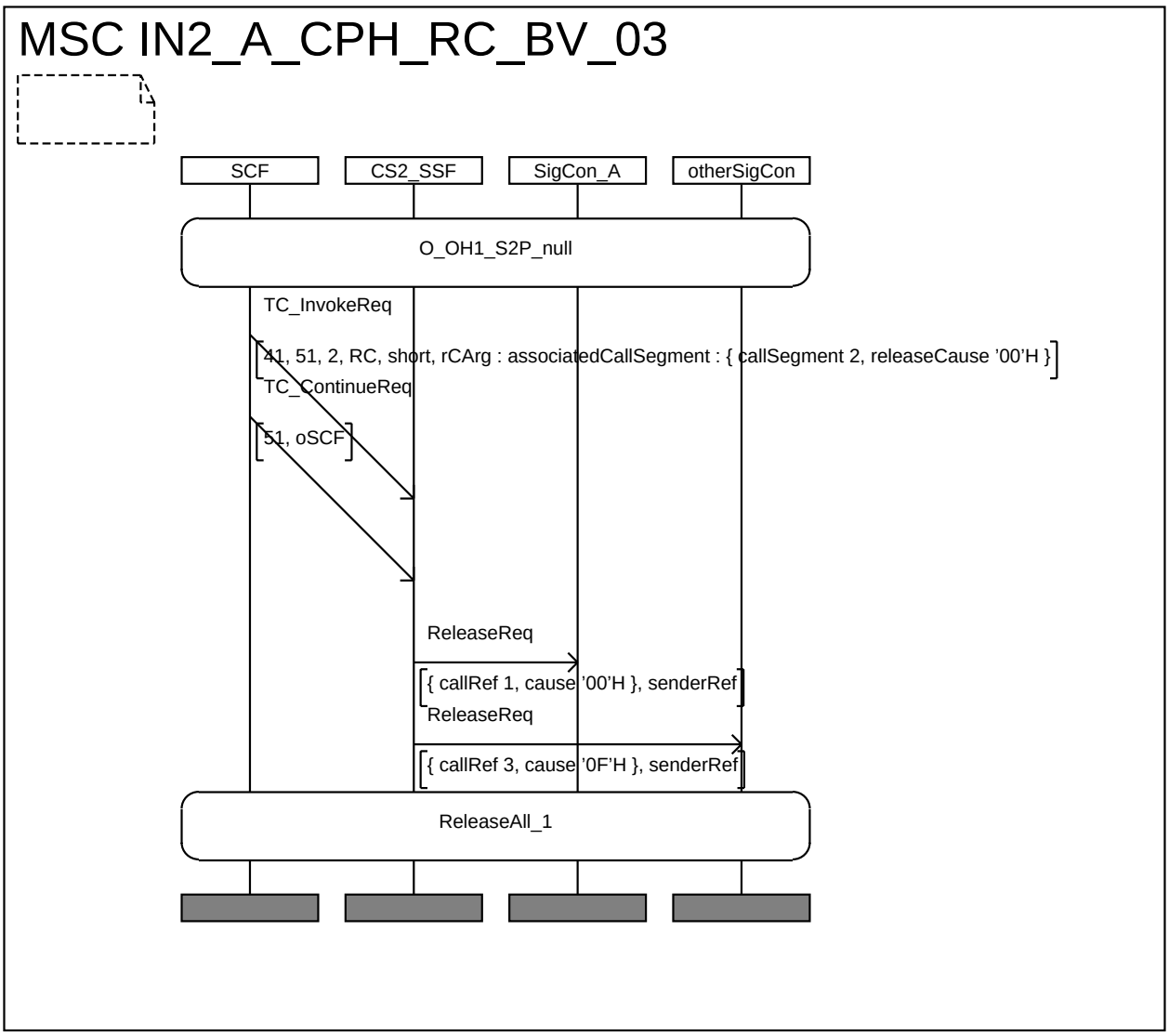
IN2_A_CPH_RC_BV_01	
Purpose:	test release of a given Csid, one leg only
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **ReleaseCall_invoke** to SSFwith the following parameters: - cause=normal unspecified - Csid=1
Pass criteria	Check that SSF sends to SigConB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



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

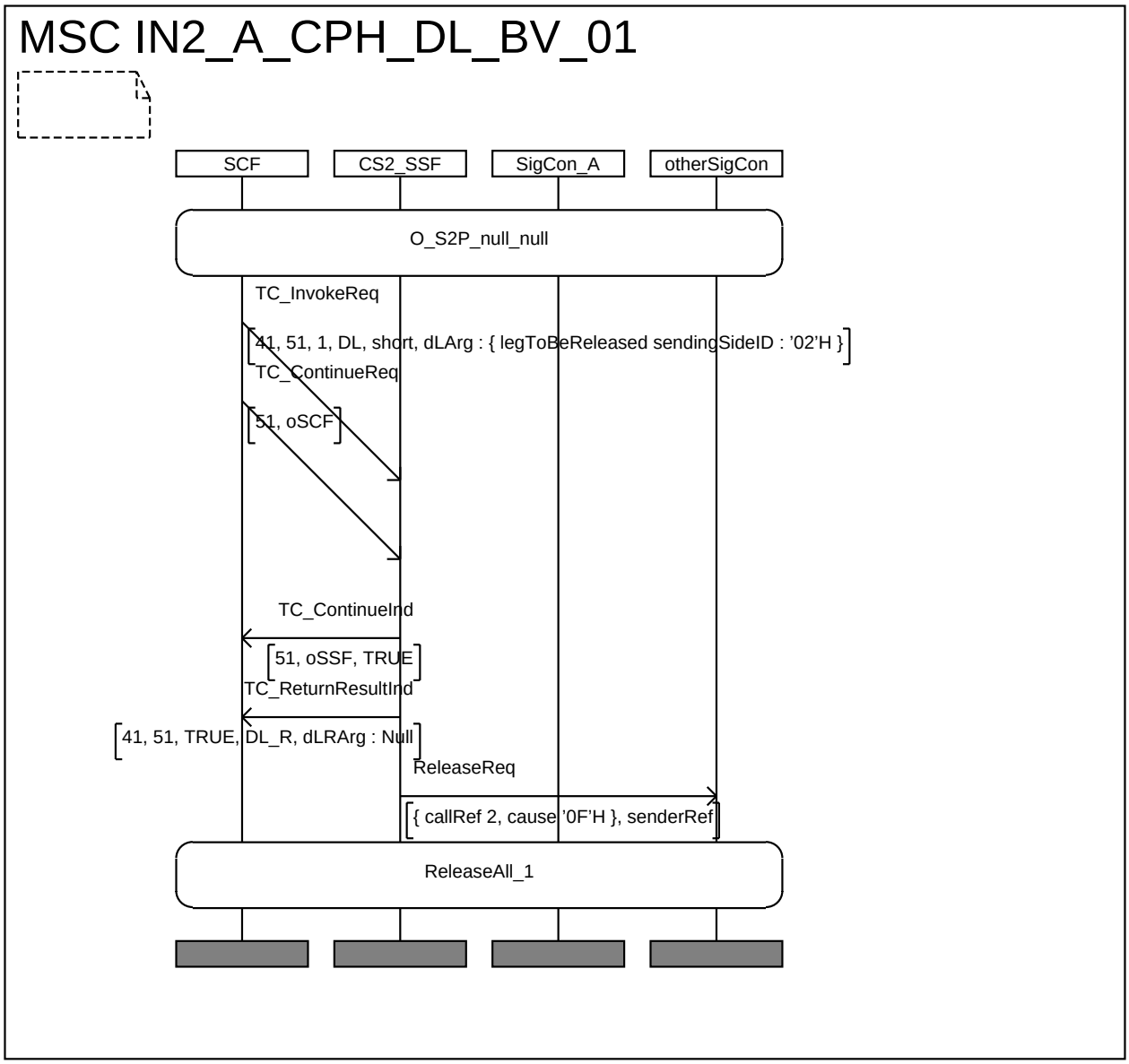


IN2_A_CPH_RC_BV_03	
Purpose:	Test of **ReleaseCall** procedure with two parties
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends to SSF **ReleaseCall** invoke, with: - associatedCallSegment callSegment(CsId=2) releaseCause(cause=00)
Pass criteria	Check that SSF releases the call (ReleaseReq received by SigConA and SigConC)
Postamble:	ReleaseAll_1

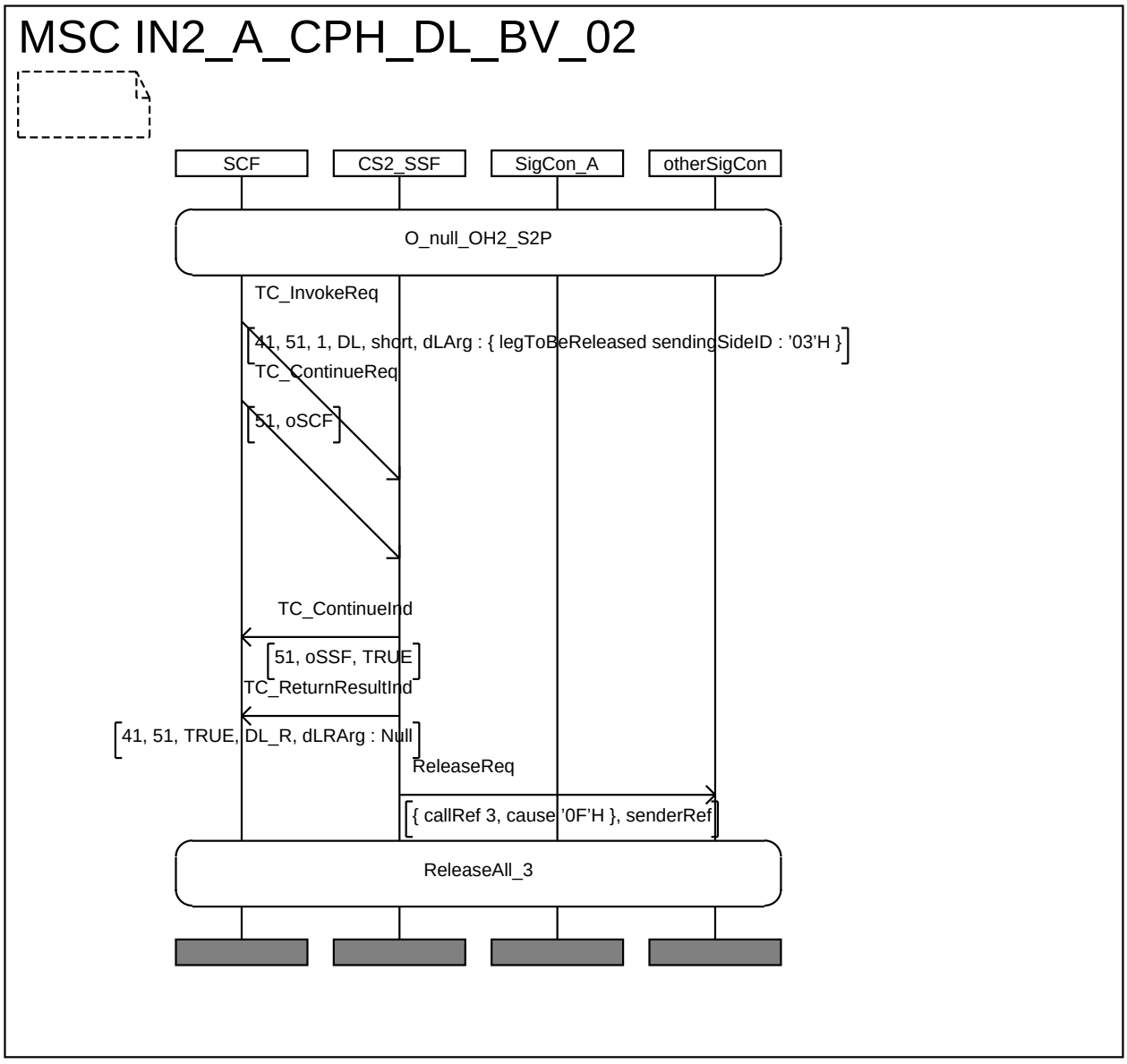


7.4.4 DisconnectLeg Procedure (DL)

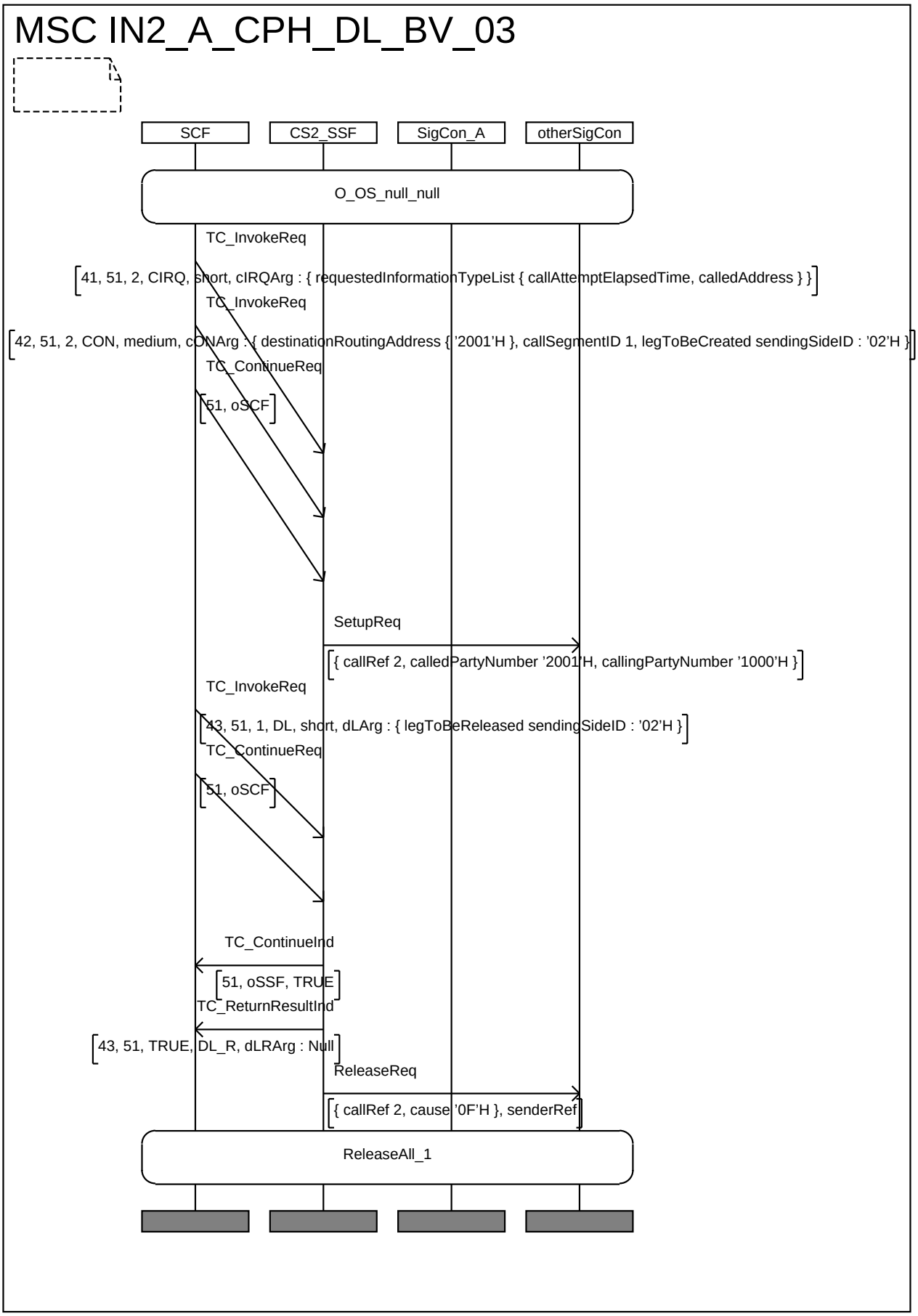
IN2_A_CPH_DL_BV_01	
Purpose:	test Disconnect leg in Csid1
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	SCF sends **DisconnectLeg_invoke** to SSF with the following parameters: - legToBeReleased=2
Pass criteria	Check SSF sends a DisconnectLegReturnResult Check that SSF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



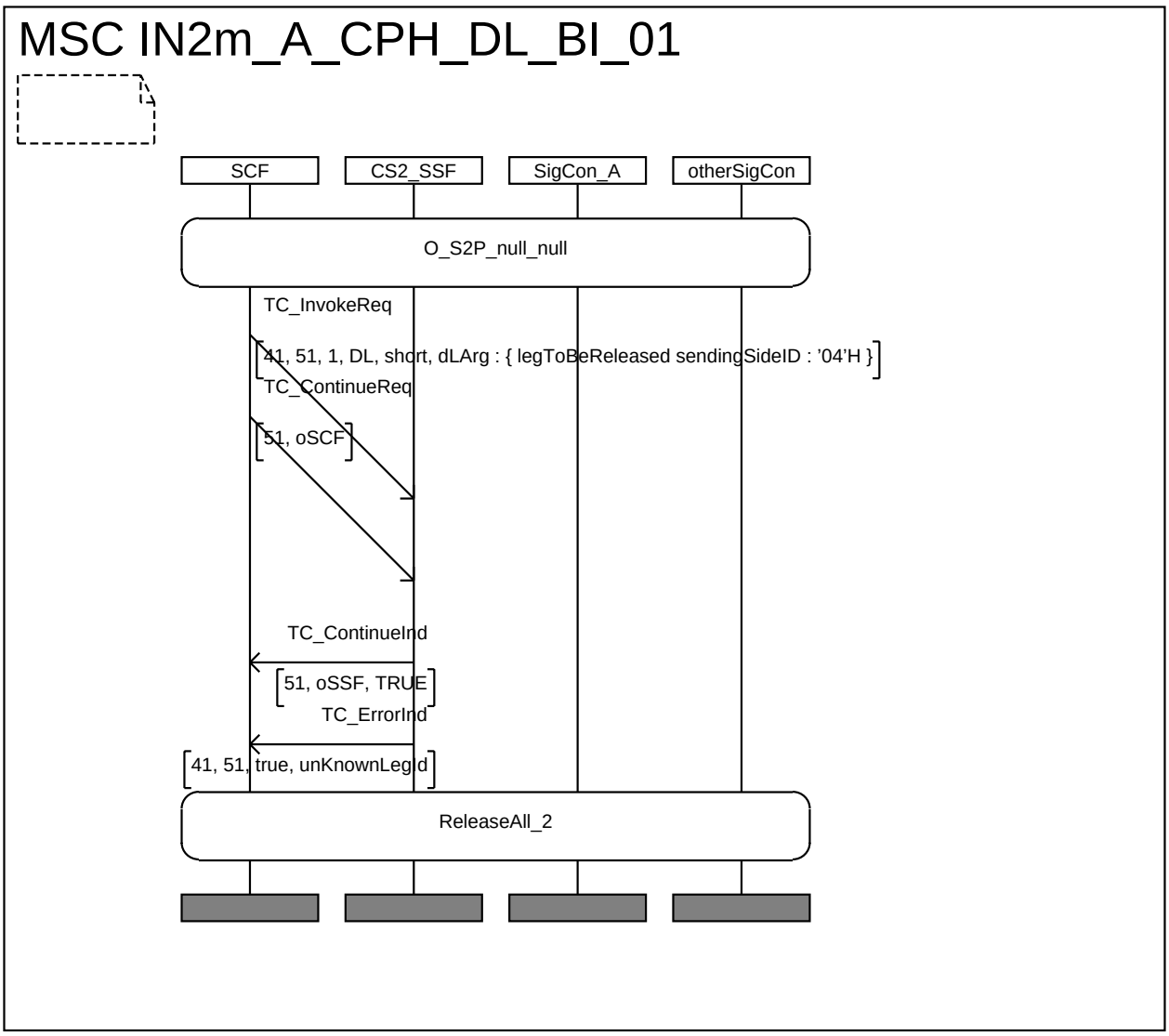
IN2_A_CPH_DL_BV_02	
Purpose:	test Disconnect one leg of a multileg Csid
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P
Test description	SCF sends **DisconnectLeg_invoke** to SSF with the following parameters: - legToBeReleased=3
Pass criteria	Check SSF sends a DisconnectLegReturnResult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



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

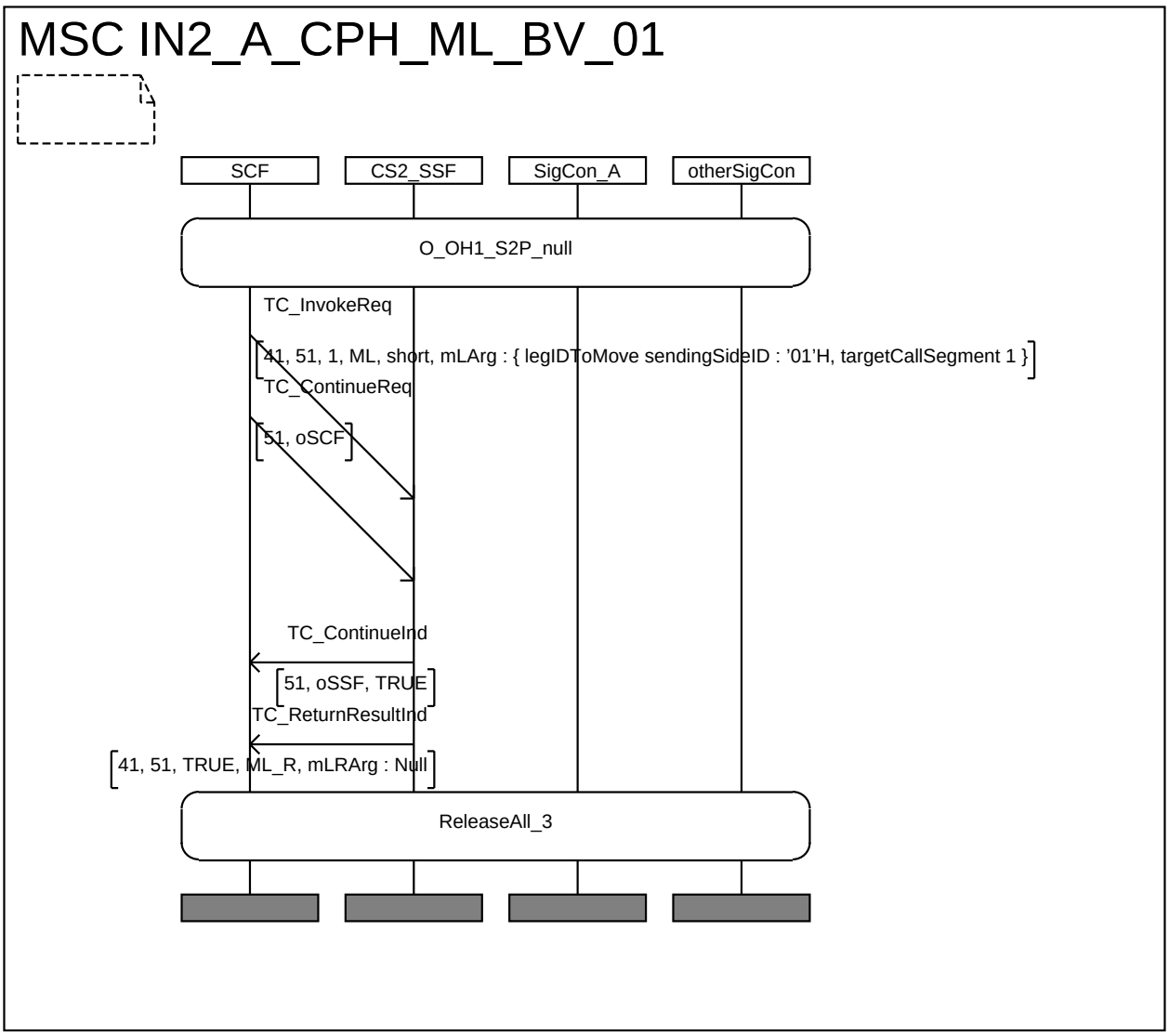

MSC IN2_A_CPH_DL_BV_03
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OS_null_null
TC_InvokeReq
41, 51, 2, CIRQ, short, cIRQArg : { requestedInformationTypeList { callAttemptElapsedTime, calledAddress } }
TC_InvokeReq
42, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2001'H }, callSegmentID 1, legToBeCreated sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1000'H }
TC_InvokeReq
43, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
43, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 2, cause '0F'H }, senderRef
ReleaseAll_1

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

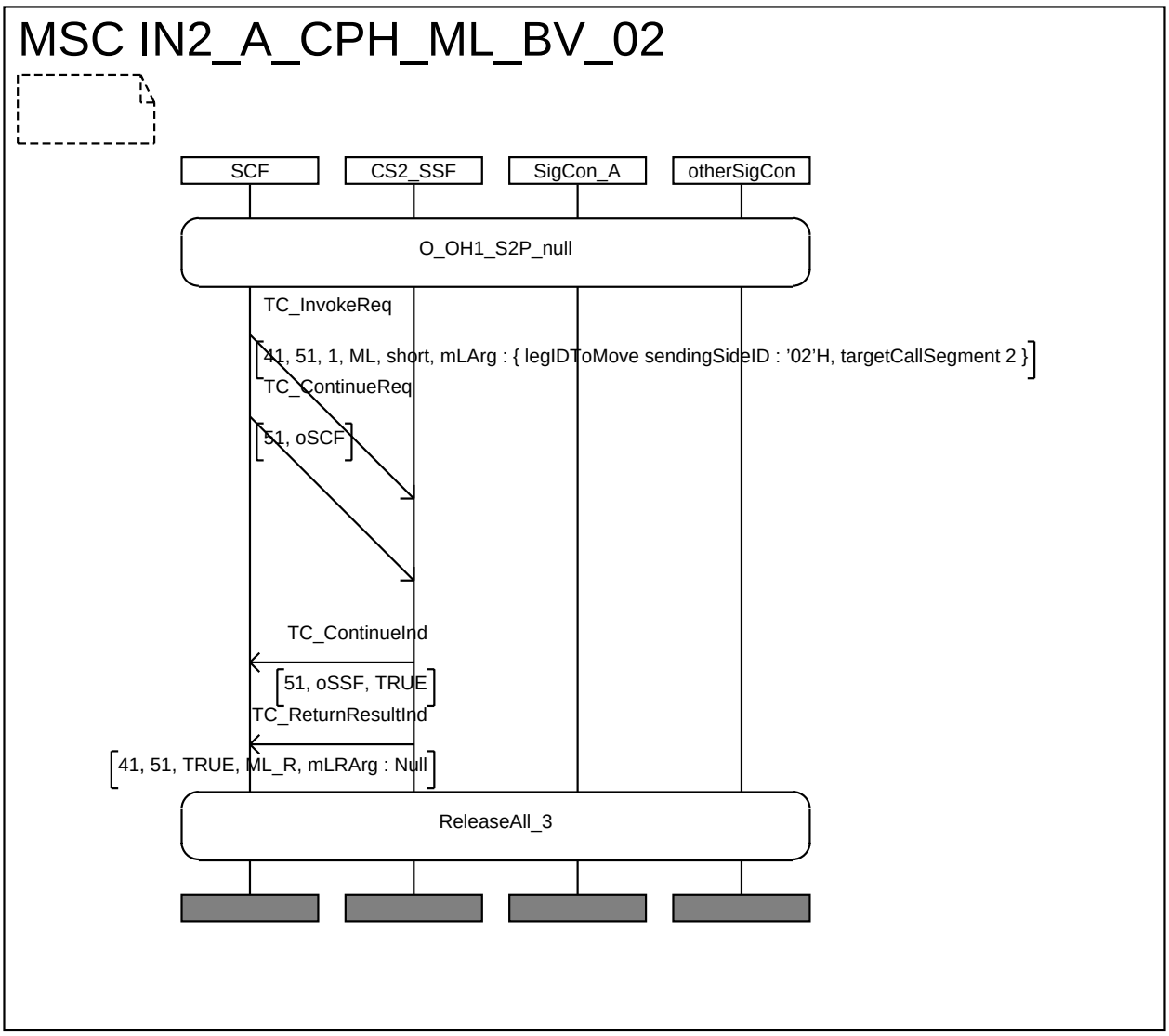


7.4.5 MoveLeg Procedure (ML)

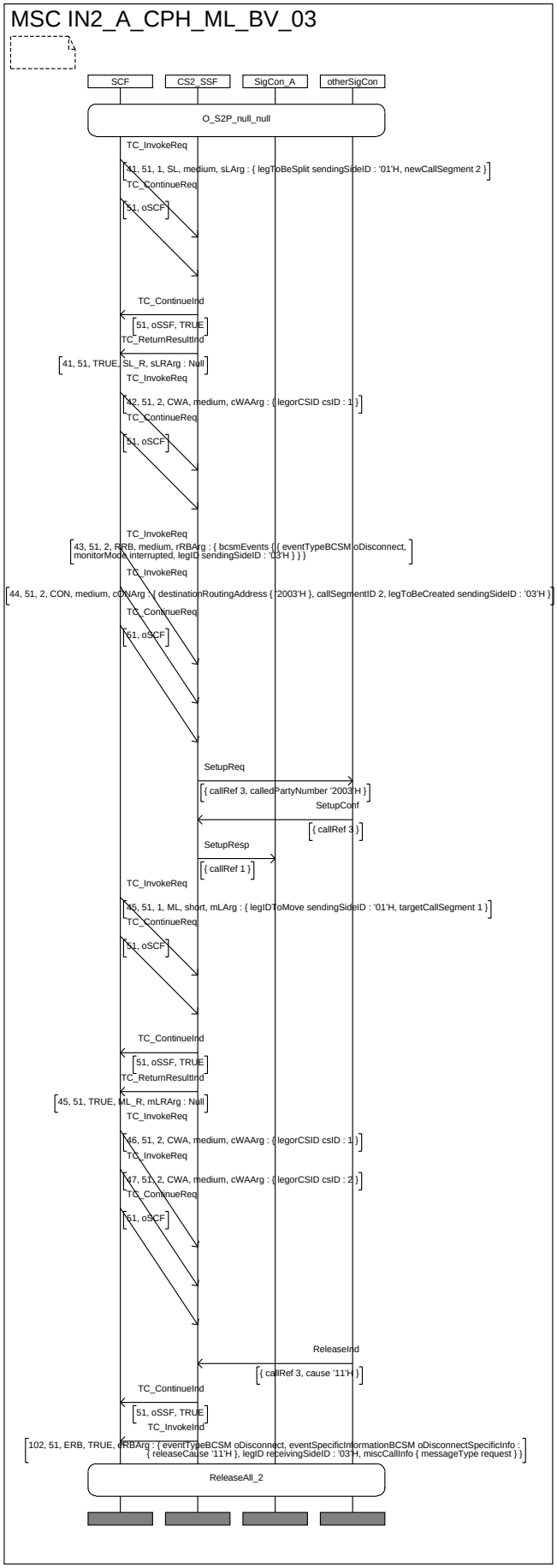
IN2_A_CPH_ML_BV_01	
Purpose:	test Moveleg functionality
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **MoveLeg_invoke** to SSFwith the following parameters: - legIdToMove=1 - targetCallSegment=1
Pass criteria	Check that the SSF sends MoveLegReturnResult
Postamble: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



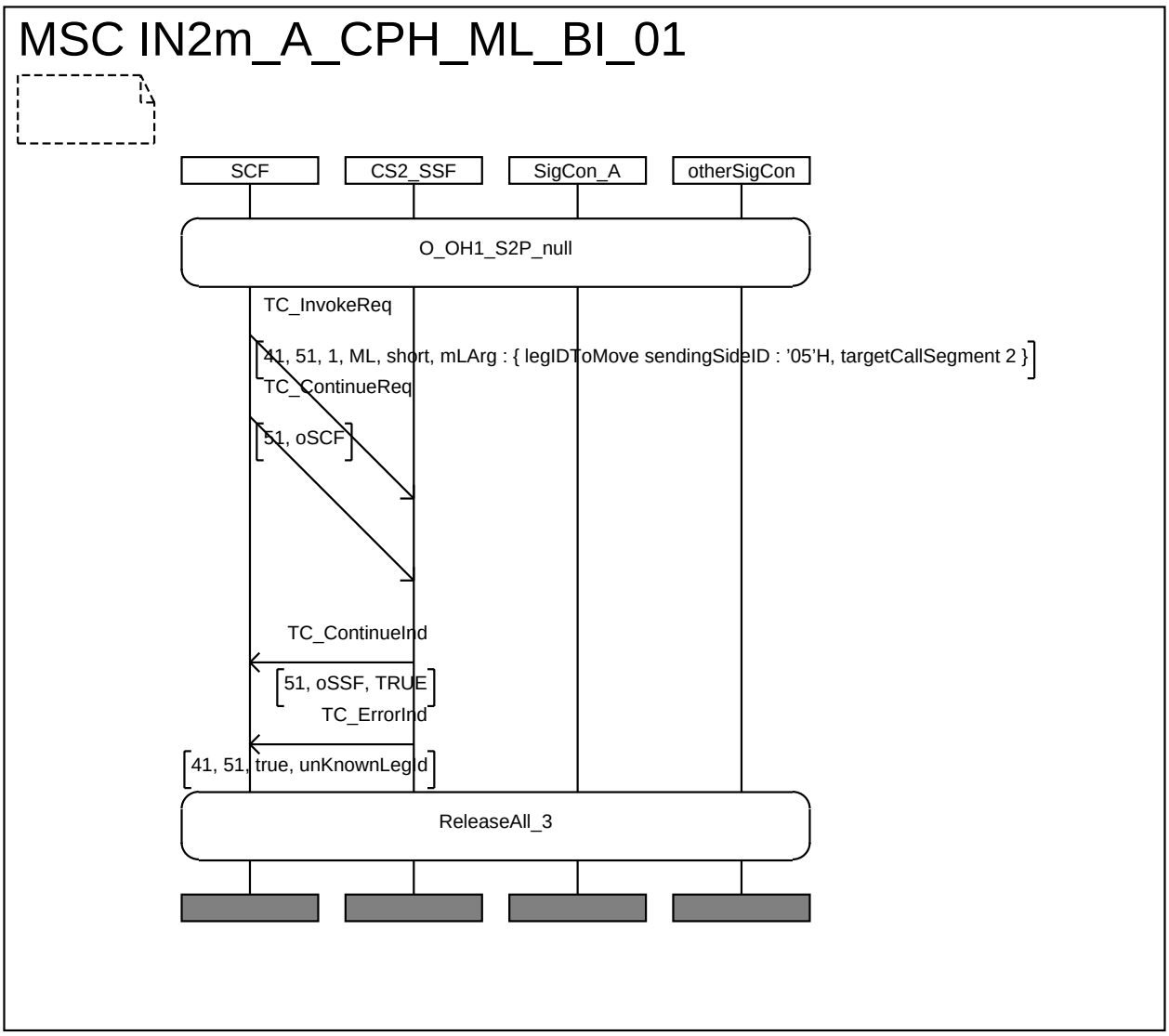
IN2_A_CPH_ML_BV_02	
Purpose:	test Moveleg functionality
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **MoveLeg_invoke** to SSFwith the following parameters: - legIdToMove=2 - targetCallSegment=2
Pass criteria	Check that the SSF sends MoveLegReturnResult
Postamble: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


MSC IN2_A_CPH_ML_BV_02
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '02'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
ReleaseAll_3

IN2_A_CPH_ML_BV_03	
Purpose:	test Event reporting rules apply after MoveLeg
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	SigConB sends SetupConf, SCF sends **SplitLeg** - legToBeSplit=1 - newCallSegment=2 followed by **ContinueWithArgument** for csID=1 SCF sends **RequestReportBCSM** (oDisconnect, legID=3), SCF send **Connect** (callSegmentID=2,legToBeCreated=3). After SetupReq is received at SigConC,which is answered with SetUpConf, SCF sends **MoveLeg_invoke** to SSFwith the following parameters: - legIdToMove=1 - targetCallSegment=1 followed by **ContinueWithArgument** for csID=1 and 2. SigConC send ReleaseInd with cause BptyDisc.
Pass criteria	Check that the SSF sends MoveLegReturnResult, Check that after ReleaseInd from SigConC, the **EventReportBCSM** is received with indication oDisconnect.
Postamble:	ReleaseAll_2

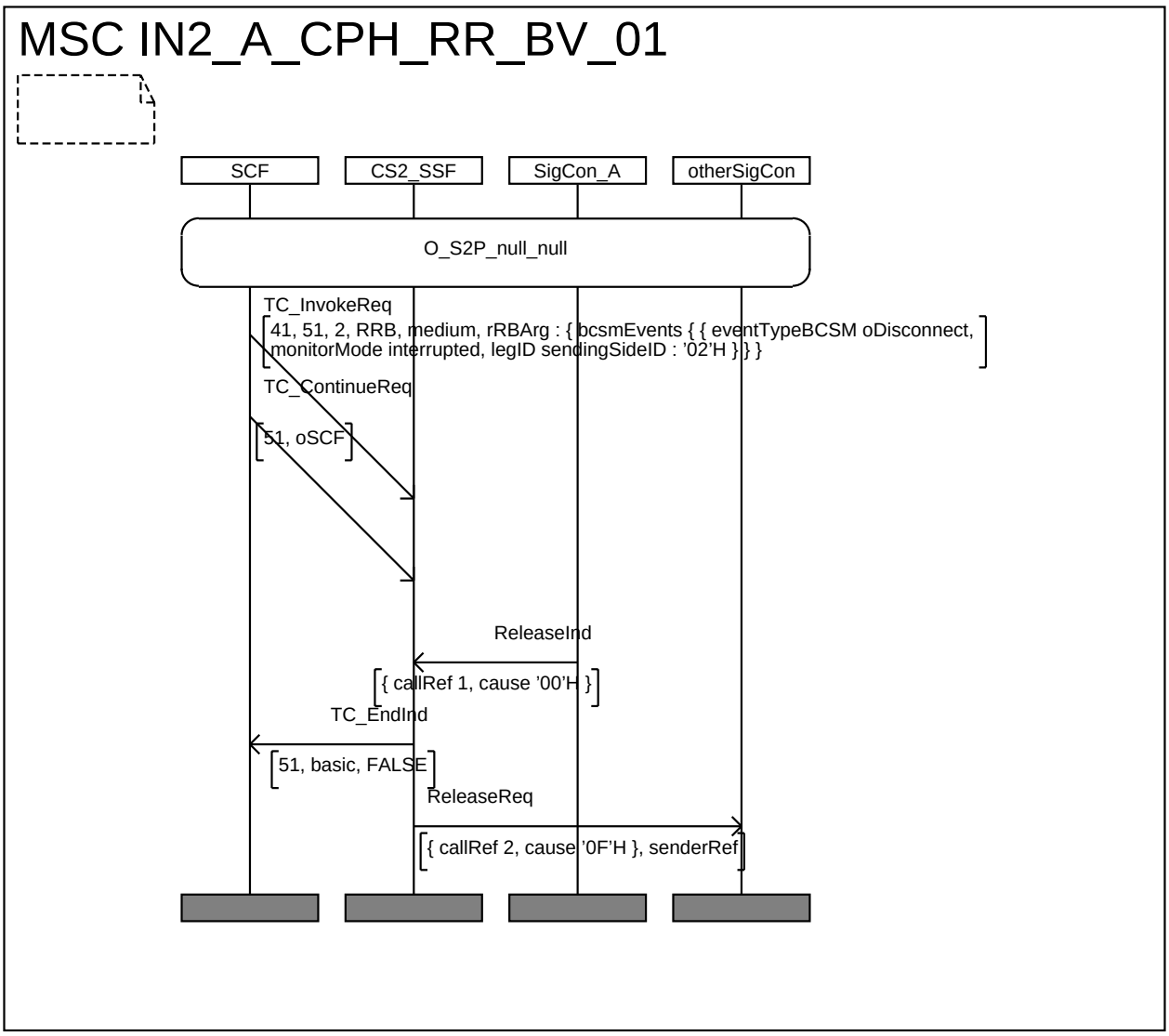

MSC IN2_A_CPH_ML_BV_03
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents {{ eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
TC_InvokeReq
45, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
45, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
46, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
47, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseInd
{ callRef 3, cause '11'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oDisconnect, eventSpecificInformationBCSM oDisconnectSpecificInfo : { releaseCause '11'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

IN2_A_CPH_ML_BI_01	
Purpose:	test that SSF sends an error after a MoveLeg with wrong parameters
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **MoveLeg_invoke** to SSF with the following parameters: - legIdToMove=5 - targetCallSegment=2
Pass criteria	as legId=5 does not exist, check that SSF sends to SCF a MoveLeg_err containing the parameter: - unknownLegId
Postamble:	ReleaseAll_3

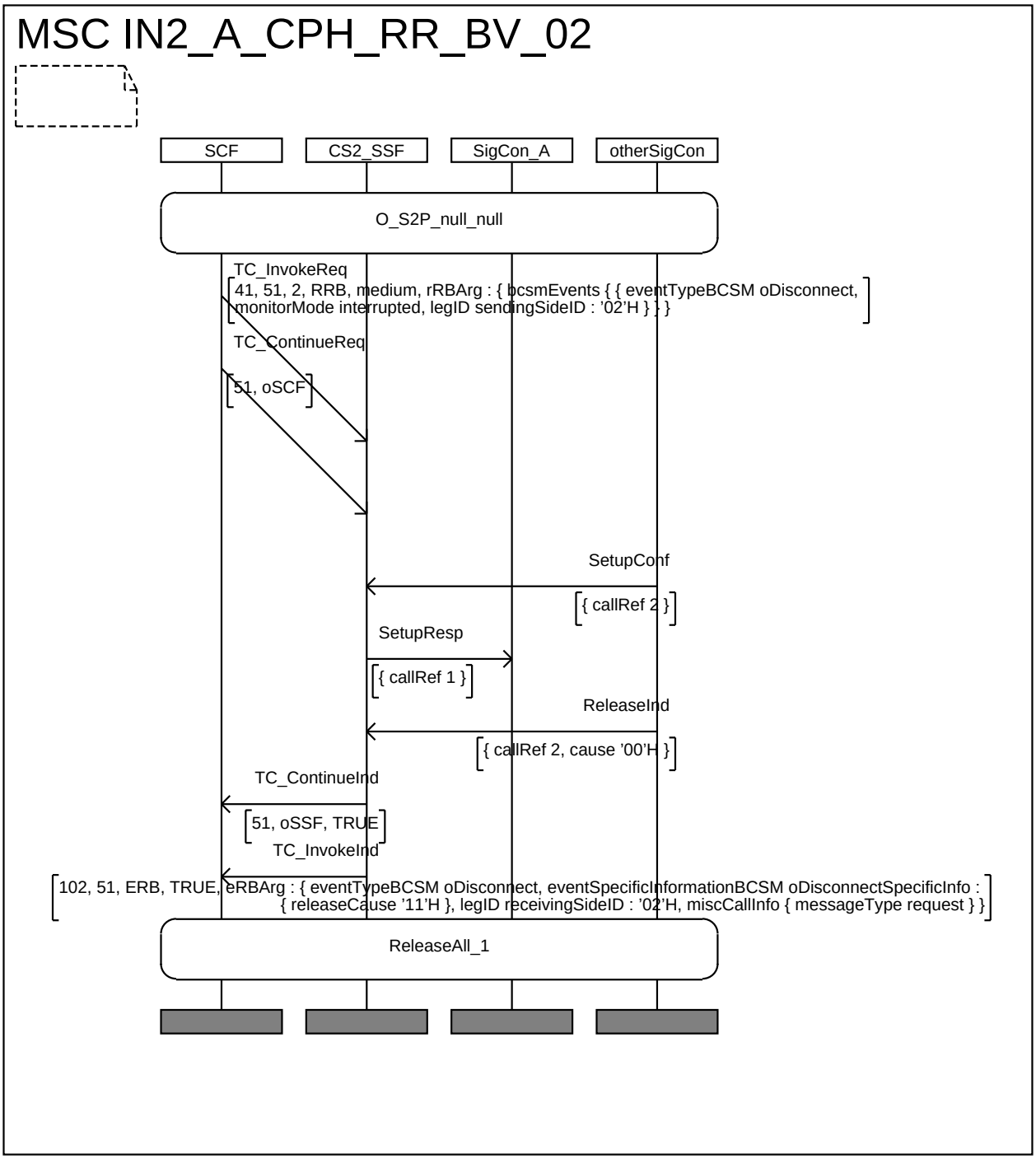


7.4.6 RequestReportBCSMEvent Procedure (CS2 additions) (RR)

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

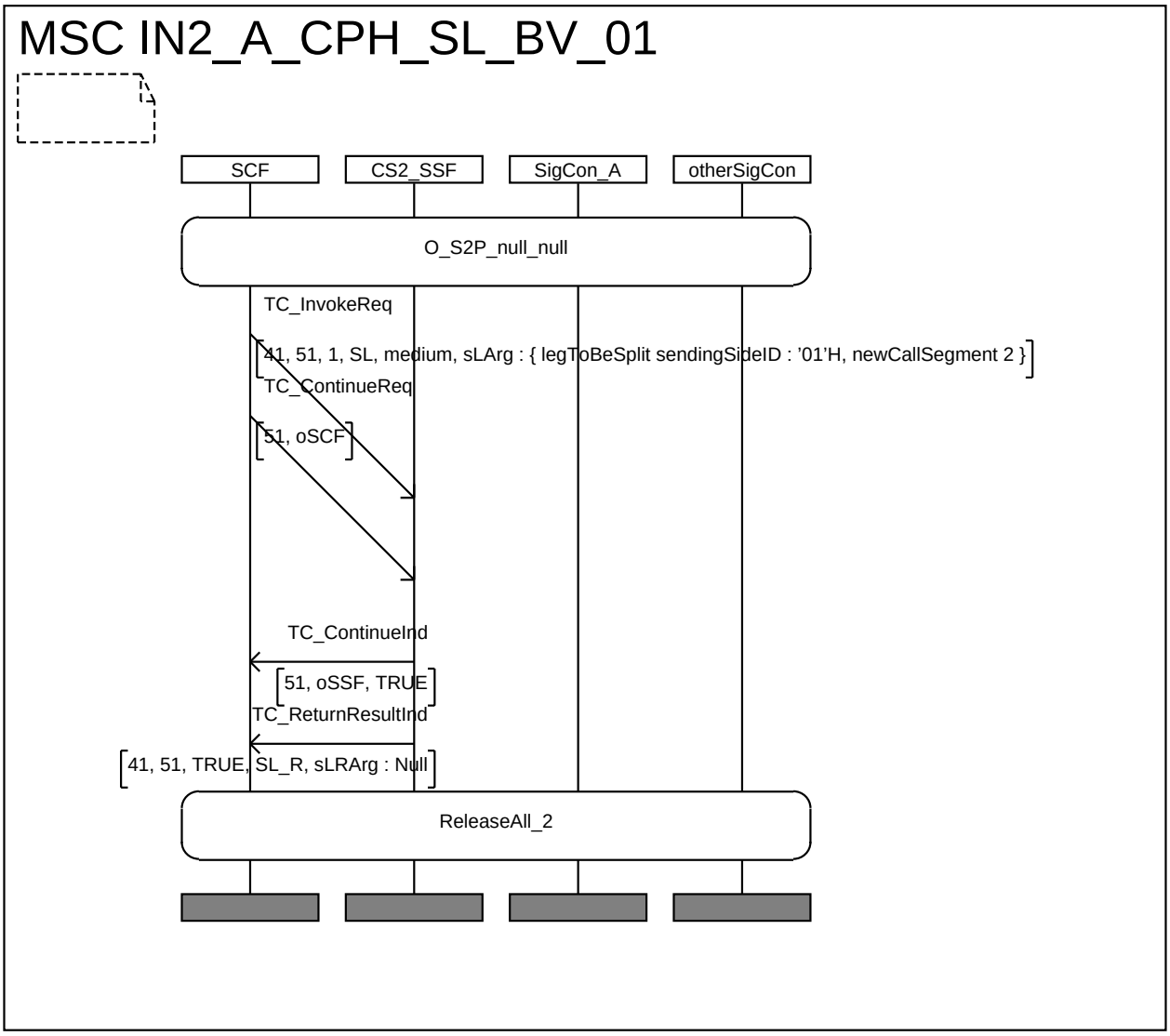


	IN2_A_CPH_RR_BV_02
Purpose:	Test of **RequestReportBCSMEvent** procedure and oDisconnect indication.
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	SCF SCF sends to SSF **RequestReportBCSMEvent** invoke containing parameters - eventTypeBCSM= oDisconnect - monitoringMode=interrupted - legid=2 Then SigCon B (called party) clears the call after it is answered (SetupConf then ReleaseInd sent)
Pass criteria	Check that SSF sends to SCF an **EventReportBCSM** with the indication of **eventTypeBCSM=** oDisconnect
Postamble:	ReleaseAll_1

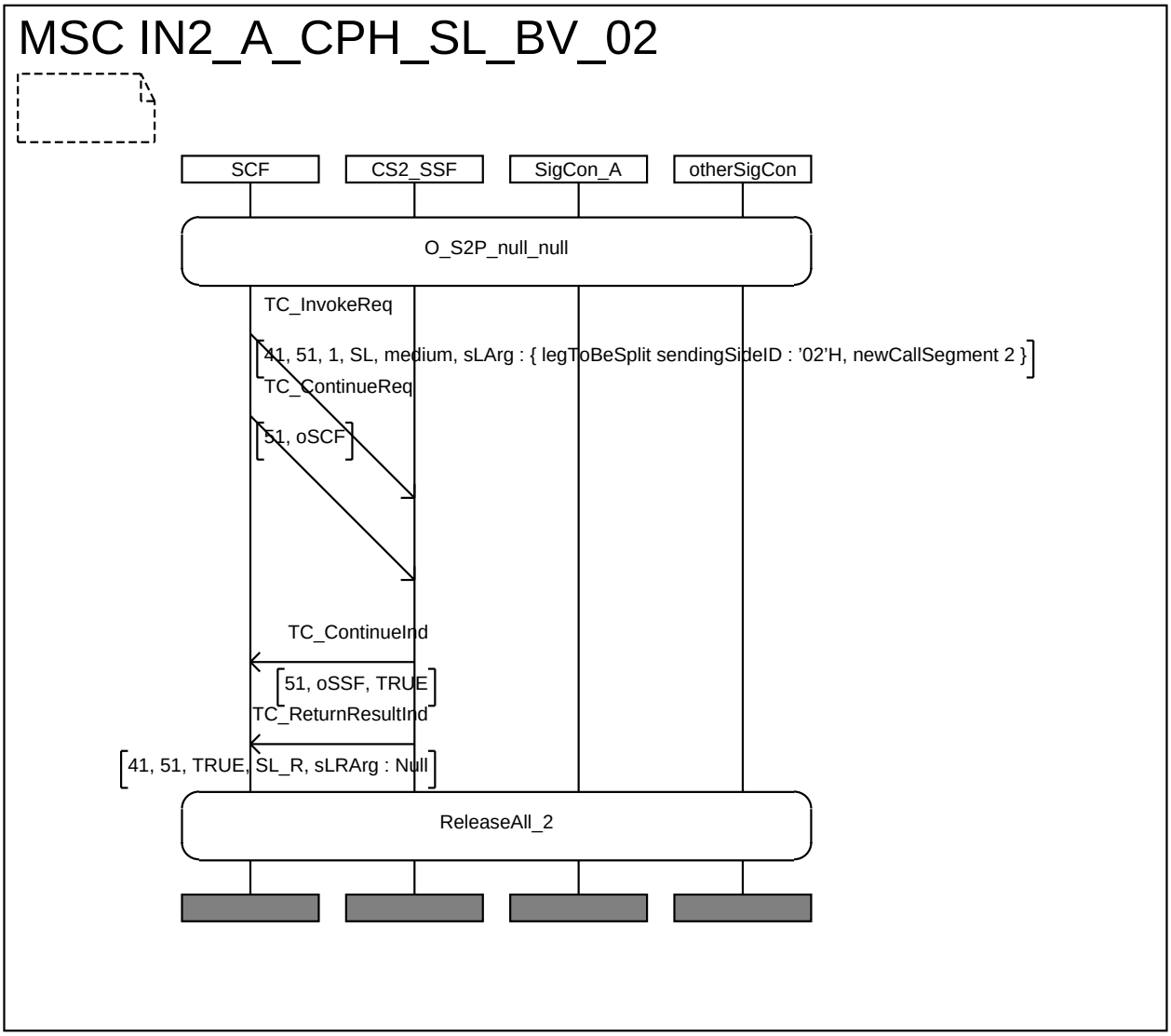

MSC IN2_A_CPH_RR_BV_02
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '02'H } } }
TC_ContinueReq
51, oSCF
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
ReleaseInd
{ callRef 2, cause '00'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oDisconnect, eventSpecificInformationBCSM oDisconnectSpecificInfo : { releaseCause '11'H }, legID receivingSideID : '02'H, miscCallInfo { messageType request } }
ReleaseAll_1

7.4.7 SplitLeg Procedure (SL)

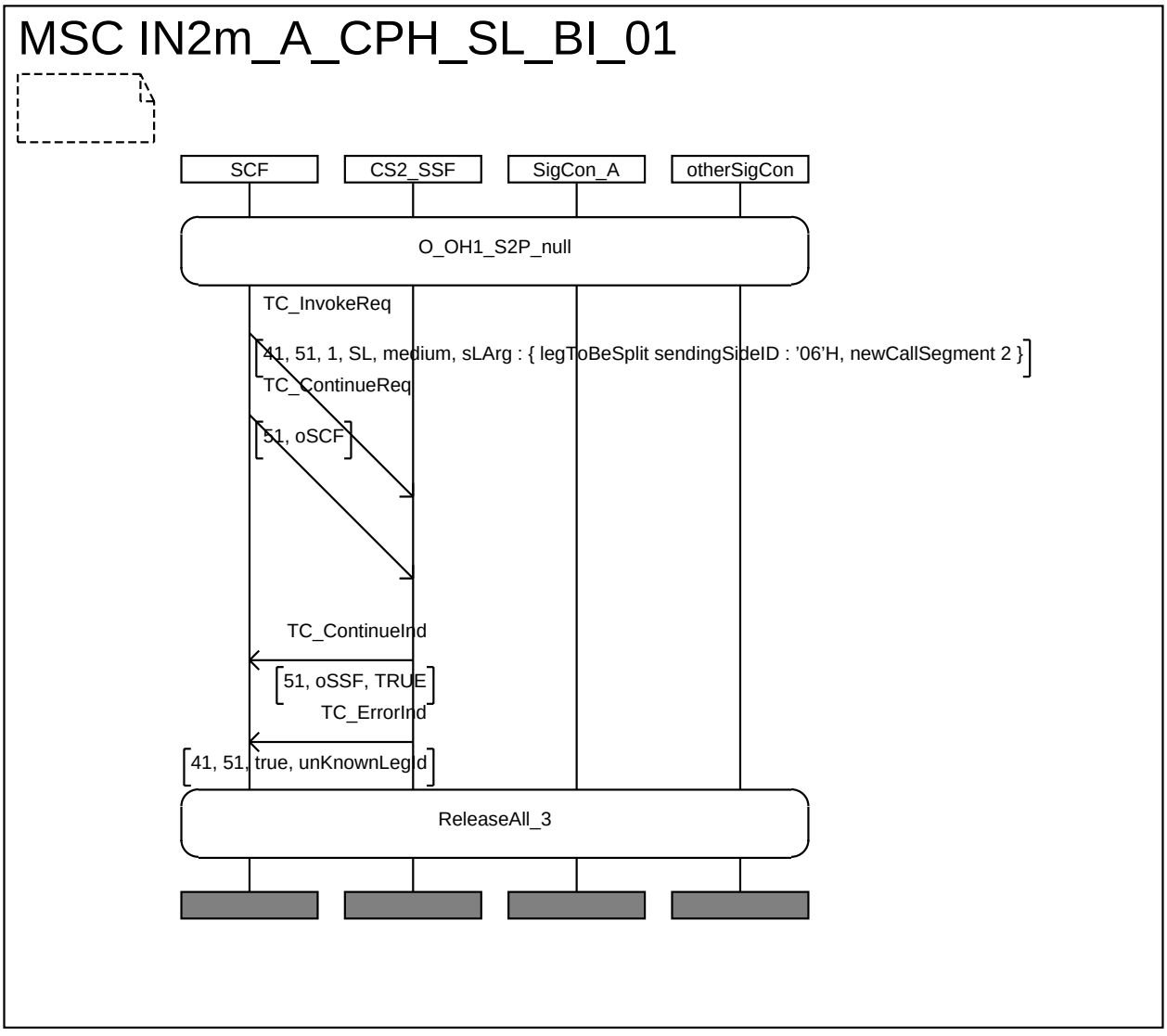
IN2_A_CPH_SL_BV_01	
Purpose:	Test of **splitLeg** procedure functionality
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	SCF sends to SSF a **SplitLeg_invoke** containing at least the parameters: - legToBeSplit=1 - newCallSegment=2
Pass criteria	Check that the SSF sends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



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

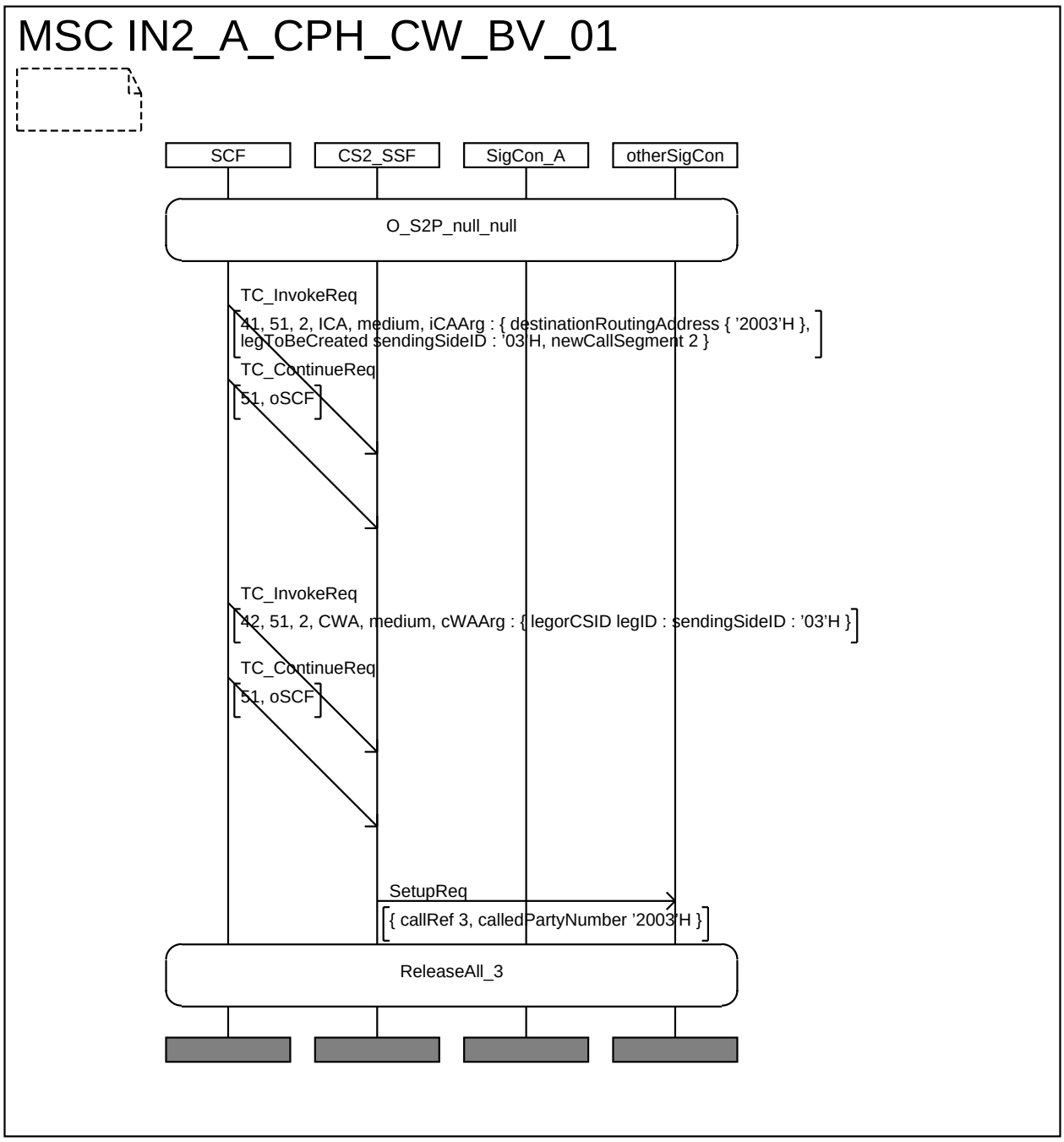


IN2_A_CPH_SL_BI_01	
Purpose:	test that SSF sends an error after a SplitLeg with wrong parameters
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends **SplitLeg_invoke** to SSF with the following parameters: - legIdToBeSplit=6 - newCallSegment=2
Pass criteria	check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

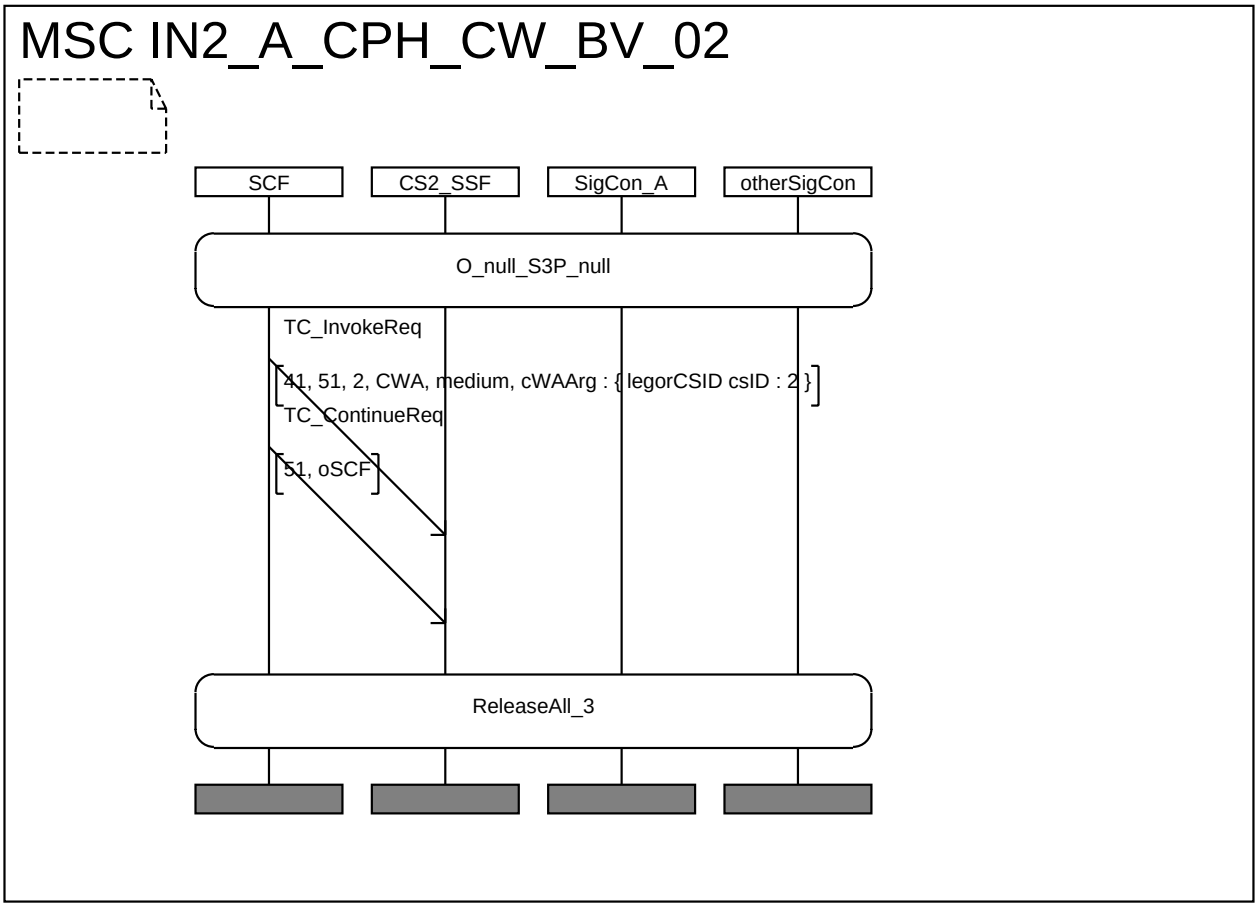


7.4.8 Continue with argument Procedure (CW)

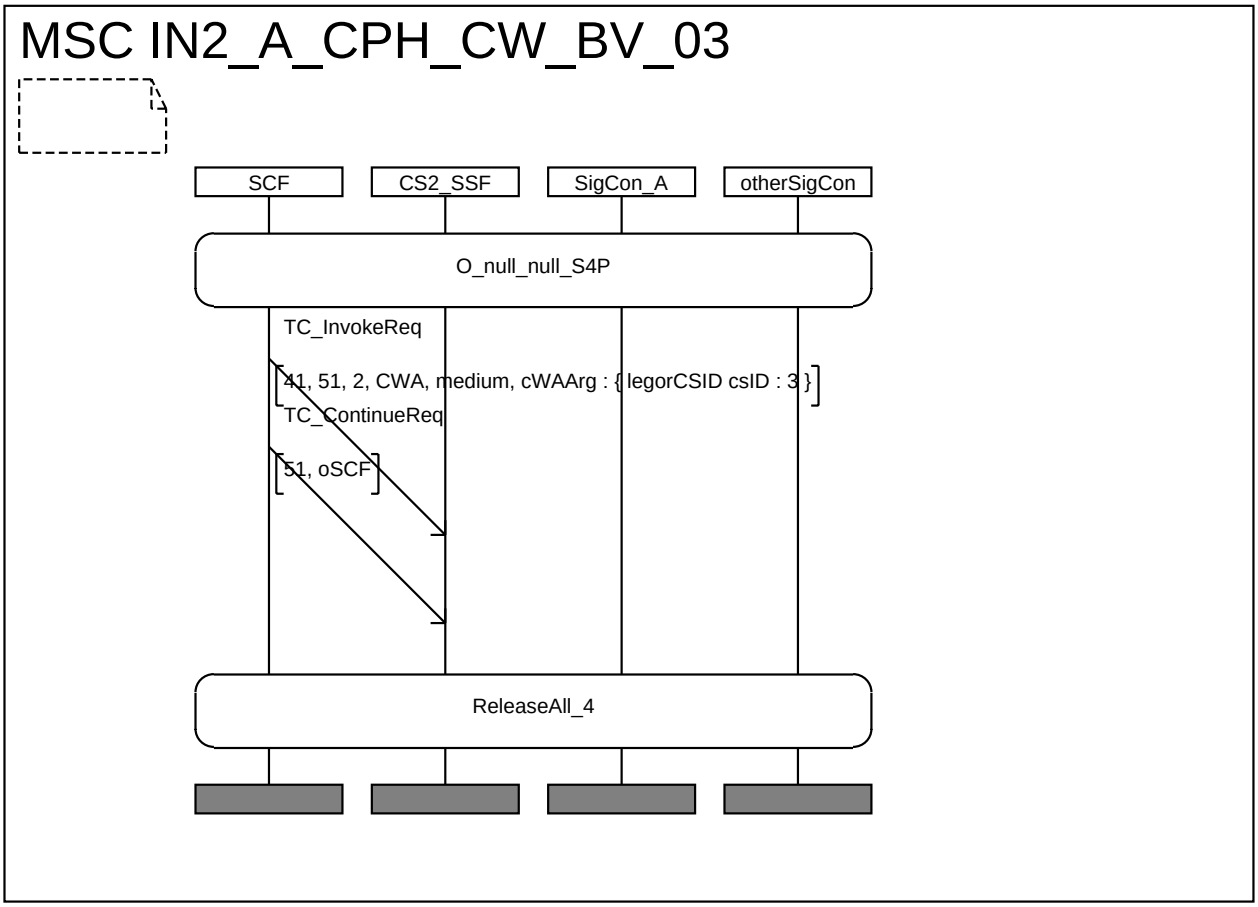
IN2_A_CPH_CW_BV_01	
Purpose:	Test of Continue with argument procedure functionality
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	SCF sends to SSF a **InitiateCallAttempt_invoke** containing at least the parameters: - legToBeCreated=3 - newCallSegment=2 followed by **ContinueWithArgument_invoke** with the following parameters: - legId=3
Pass criteria	Check that SSF sends to SigconC a SetupRequest meaning that BCSM continues
Postamble:	ReleaseAll_3



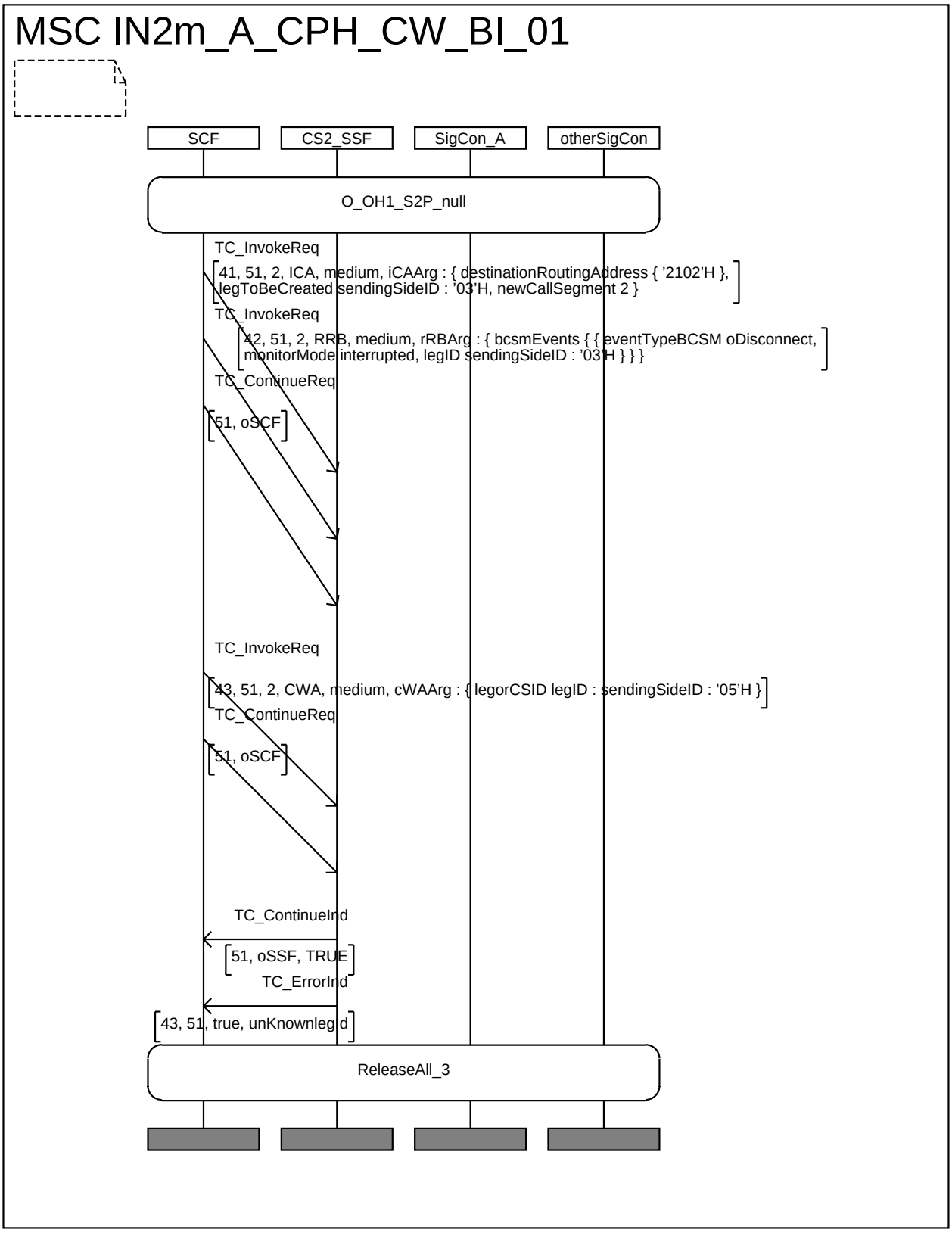
IN2_A_CPH_CW_BV_02	
Purpose:	Test of Continue with argument procedure functionality
Requirement ref	
Selection Cond.	
Preamble:	O_null_S3P_null
Test description	SCF sends **ContinueWithArgument_invoke** with the following parameters CSId=2
Pass criteria	The operation is accepted
Postamble:	ReleaseAll_3



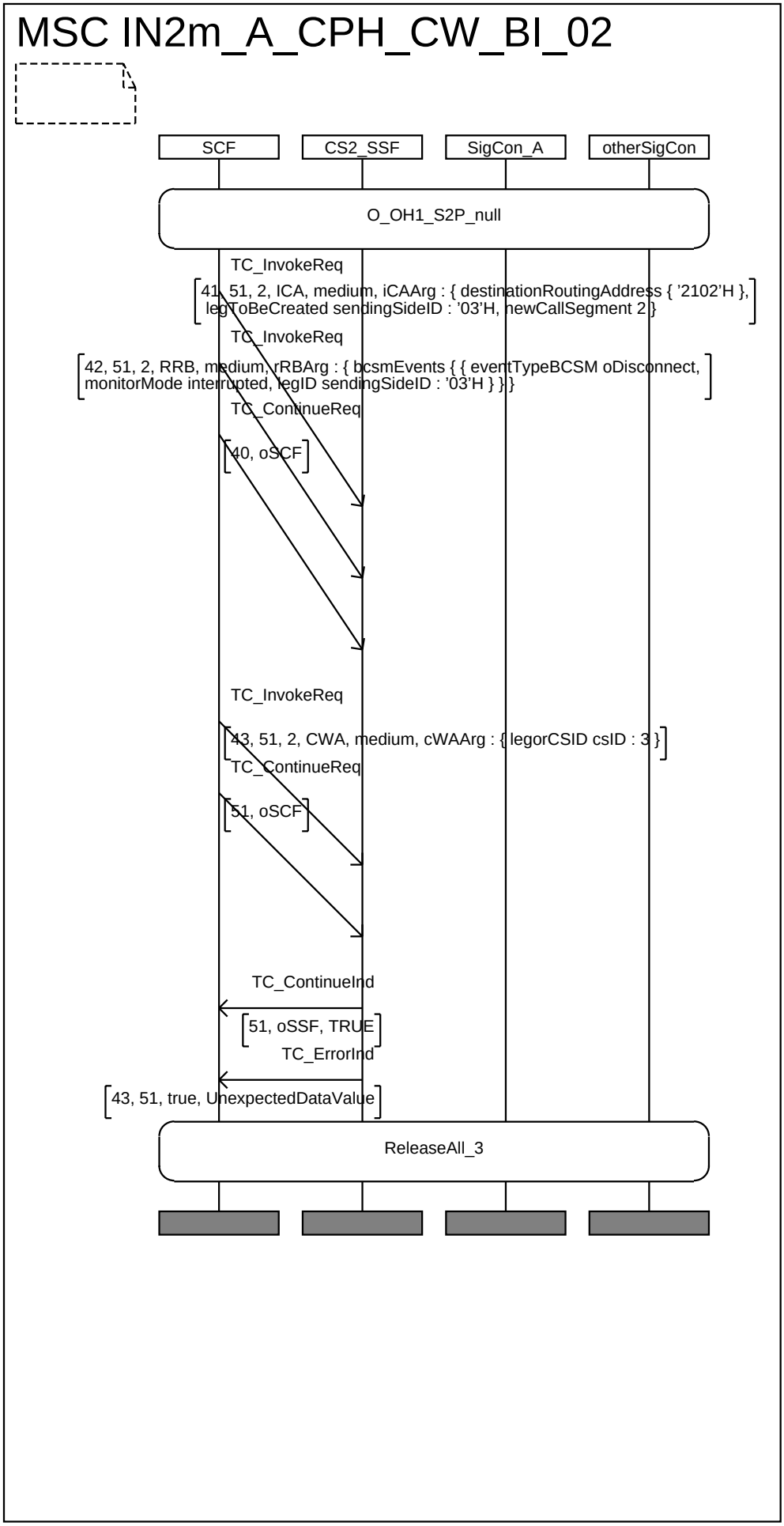
IN2_A_CPH_CW_BV_03	
Purpose:	Test of Continue with argument procedure functionality
Requirement ref	
Selection Cond.	
Preamble:	O_null_nul_S4P
Test description	SCF sends **ContinueWithArgument_invoke** with the following parameters LegId = 4, D party
Pass criteria	The operation is accepted
Postamble: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



IN2_A_CPH_CW_BI_01	
Purpose:	test that SSF sends an error after a Continue with argument with wrong parameters
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends to SSF a **InitiateCallAttempt_invoke** containing at least the parameters: - legToBeCreated=3 - newCallSegment=2 followed by **ContinueWithArgument_invoke** with the following parameters: - legId=5
Pass criteria	check that SSF sends to SCF a Continue with argument_err containing the parameter: - unknownLegId
Postamble:	ReleaseAll_3


MSC IN2m_A_CPH_CW_BI_01
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
TC_InvokeReq
41, 51, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2102'H },
legToBeCreated sendingSideID : '03'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect,
monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_ContinueReq
51, oSCF
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID legID : sendingSideID : '05'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ErrorInd
43, 51, true, unKnownlegId
ReleaseAll_3

IN2_A_CPH_CW_BI_02	
Purpose:	test that SSF sends an error after a Continue with argument with wrong parameters
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	SCF sends to SSF a **InitiateCallAttempt_invoke** containing at least the parameters: - legToBeCreated=3 - newCallSegment=2 followed by **ContinueWithArgument_invoke** with the following parameters: - CsId=3
Pass criteria	check that SSF sends to SCF a Continue with argument_err containing the parameter: - UnexpectedDataValue
Postamble:	ReleaseAll_3

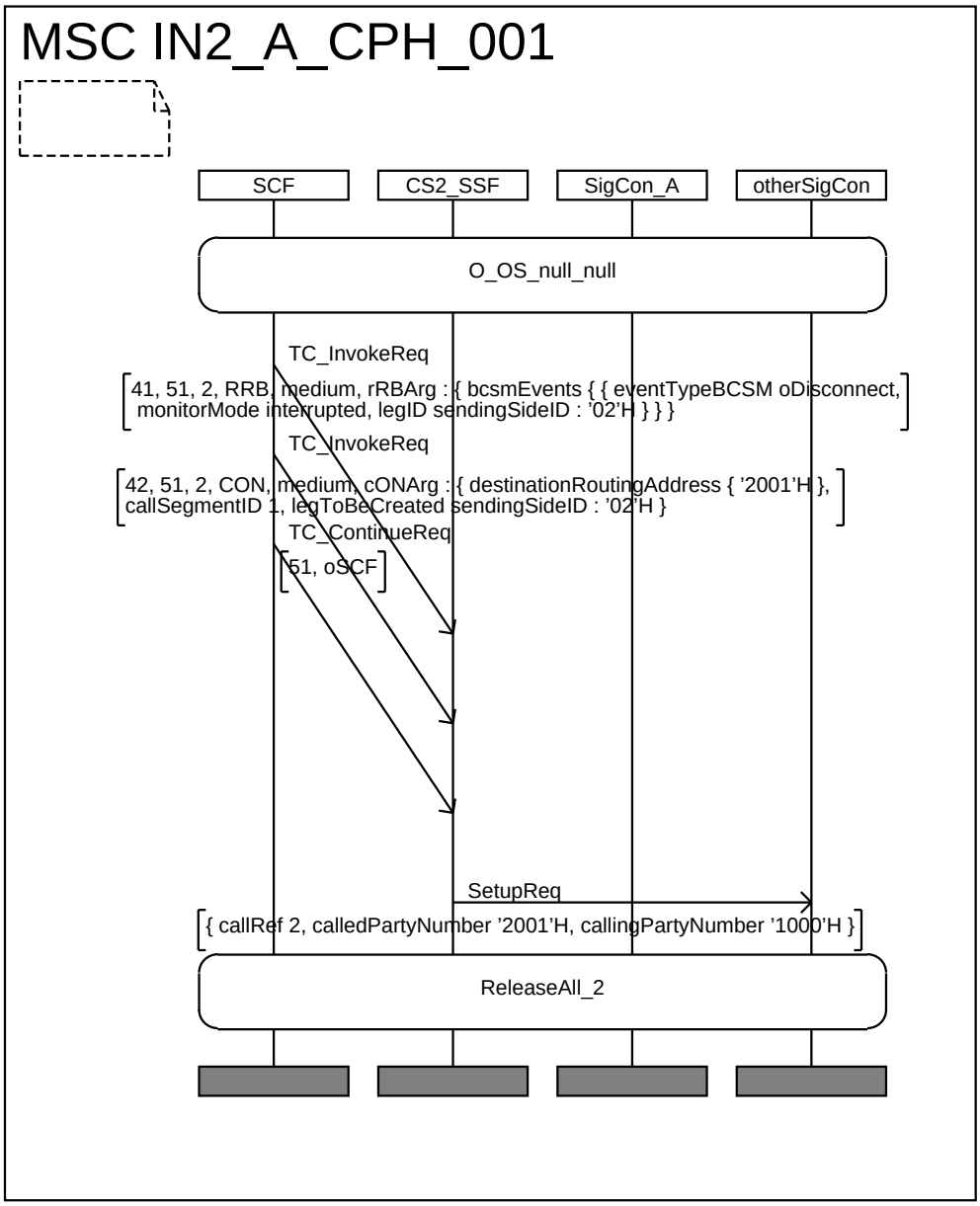

MSC IN2m_A_CPH_CW_BI_02
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
TC_InvokeReq
41, 51, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2102'H },
legToBeCreated sendingSideID : '03'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect,
monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_ContinueReq
40, oSCF
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ErrorInd
43, 51, true, UnexpectedDataValue
ReleaseAll_3

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

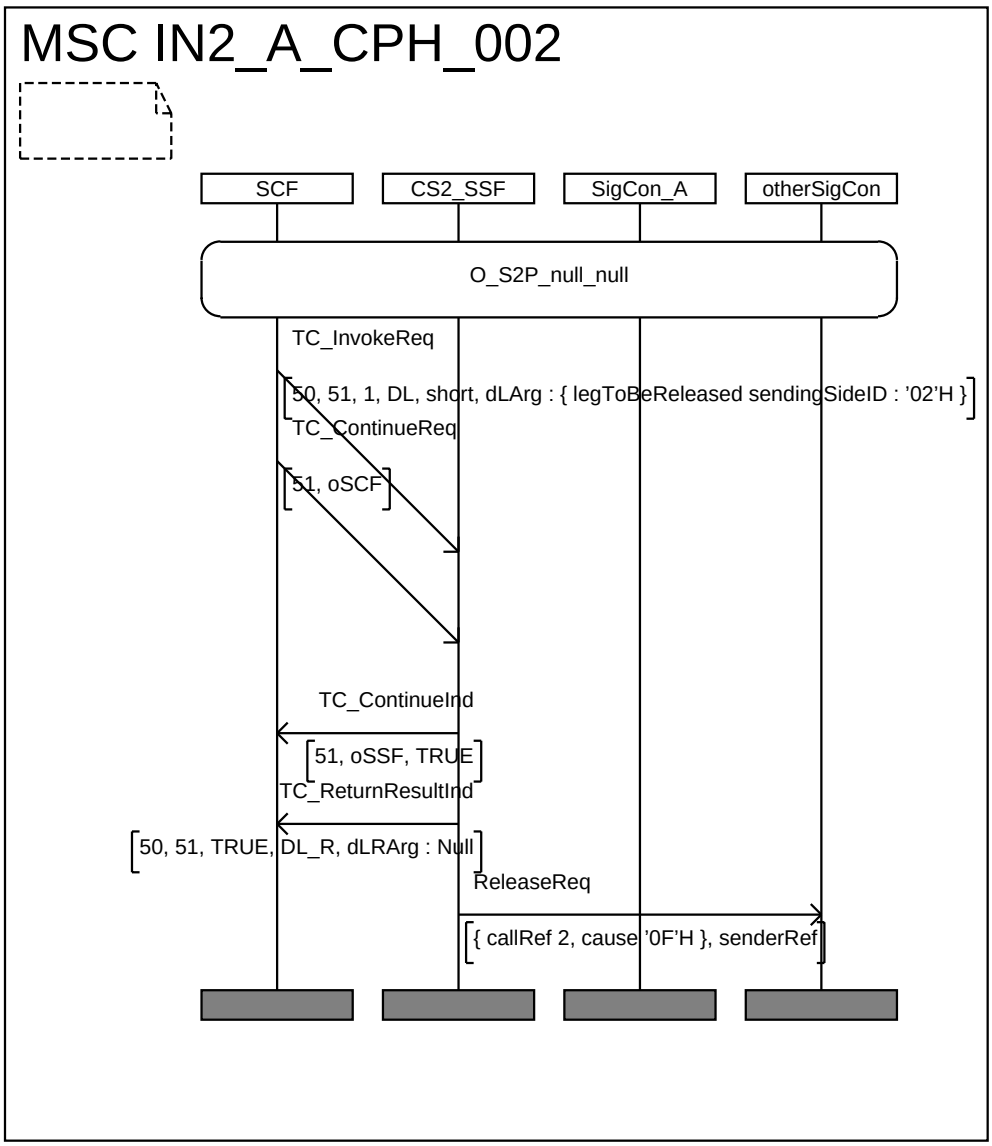
This section includes the TPs for testing the switch call handling capabilities. Due to the great number of possible transitions, the TP are restricted to a limited number of cases covering the switch functionality. Taking the preamble tree as starting point and successively starting from each preamble 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)

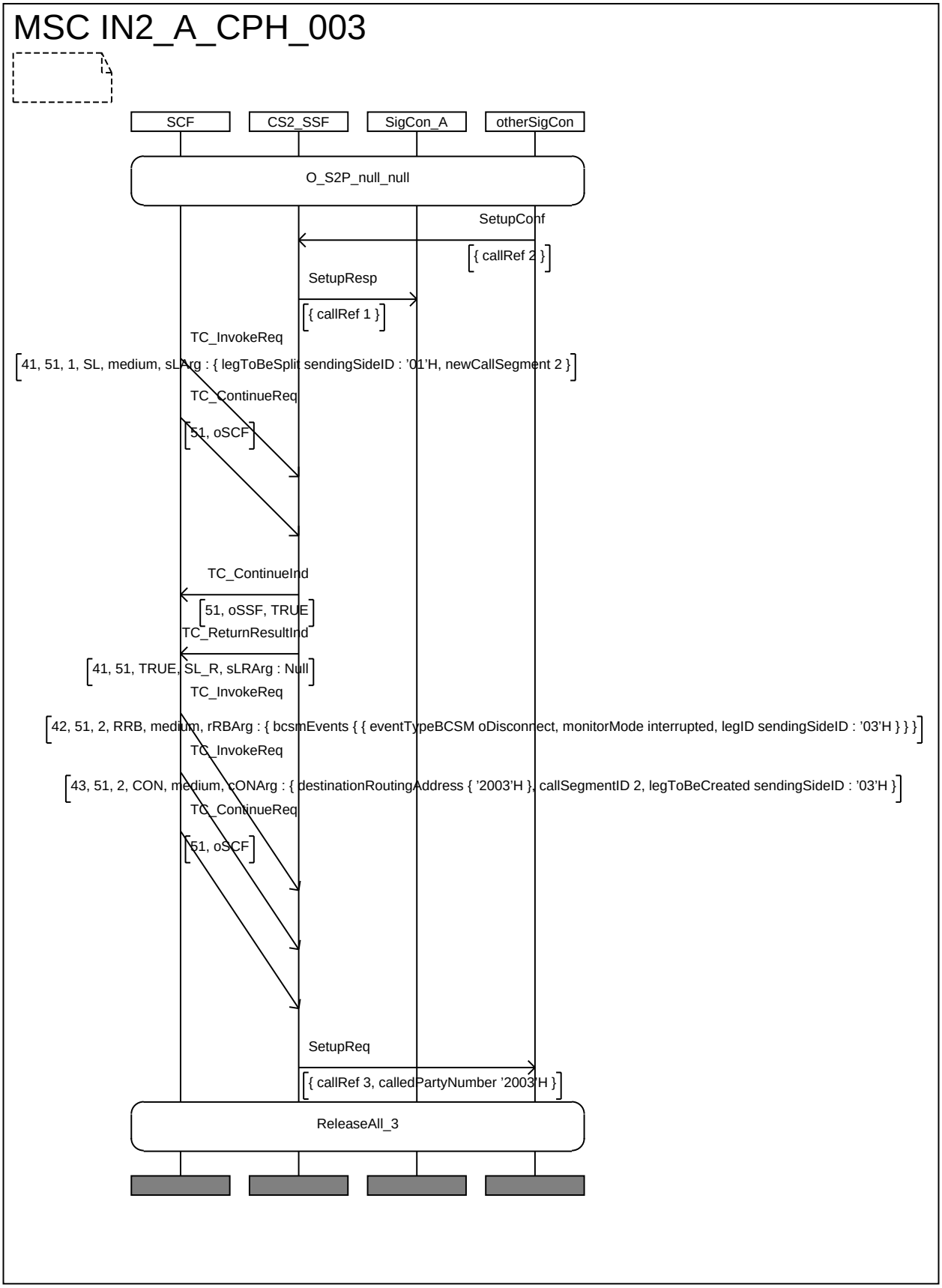
	IN2_A_CPH_001
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OS_null_null
Test description	L1! RequestReportBCSMEvent(2,oDisconnect) L1! Connect(2,1) reaching state O_S2P_null_null
Pass criteria	CP1-2? SetUpReq
Postamble:	ReleaseAll_2



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

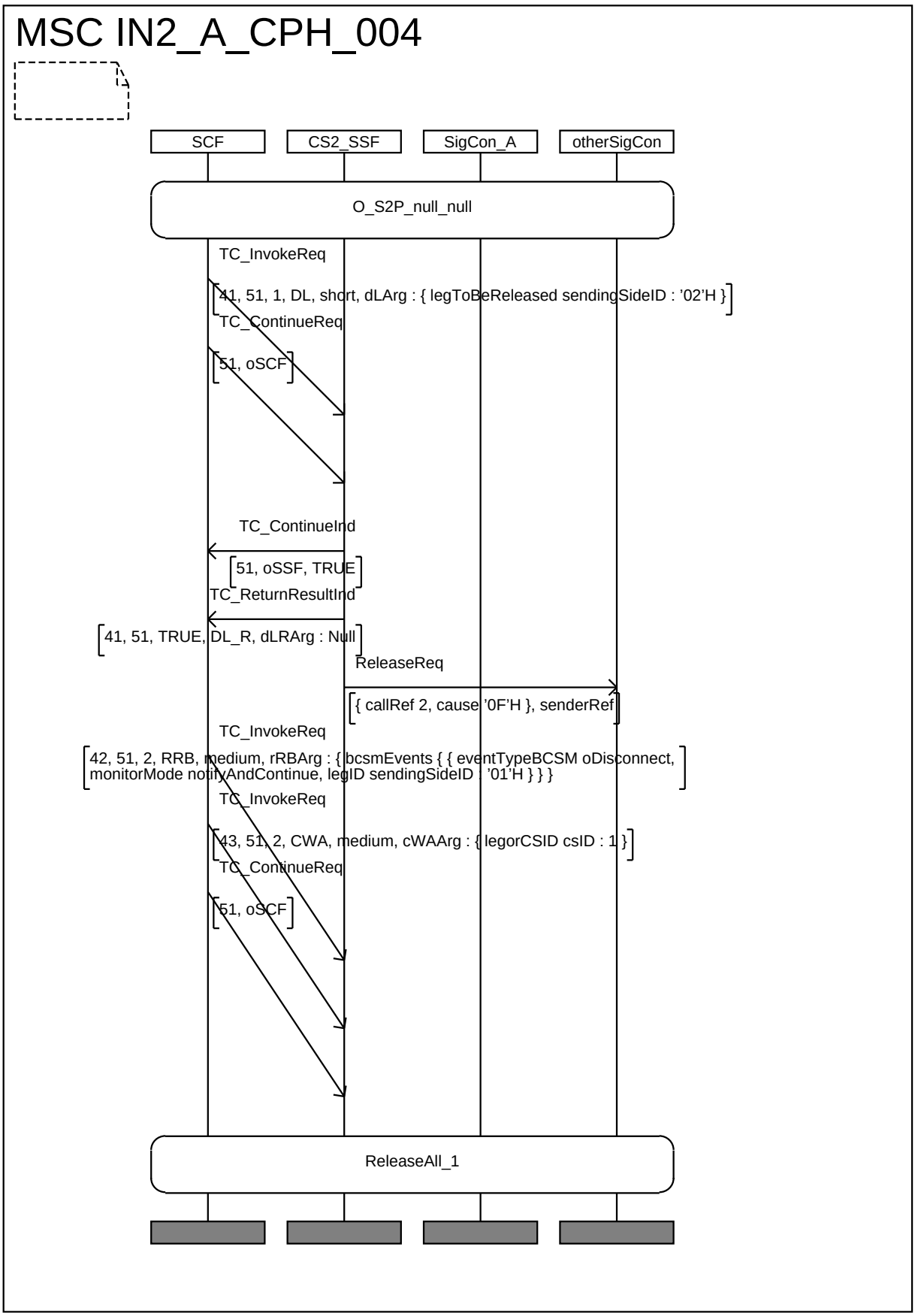


IN2_A_CPH_003	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	CP1-2! SetupConf, B party answers L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument (CsID = 1) L1! RequestReportBCSMEvent(3,oDisconnect) L1! Connect(3,2) reaching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

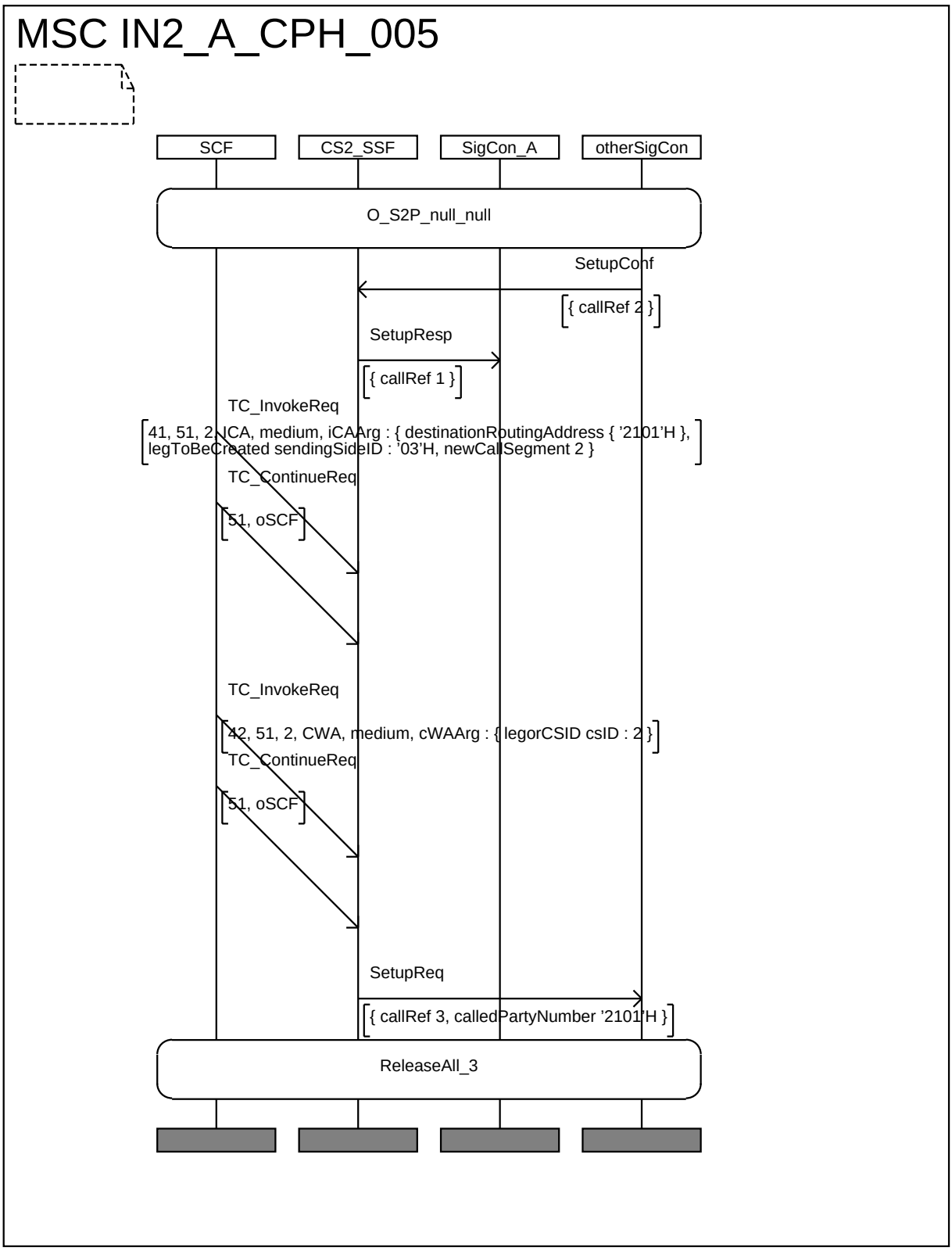


IN2_A_CPH_004	
Purpose:	Test CPH capabilities
Requirement ref	

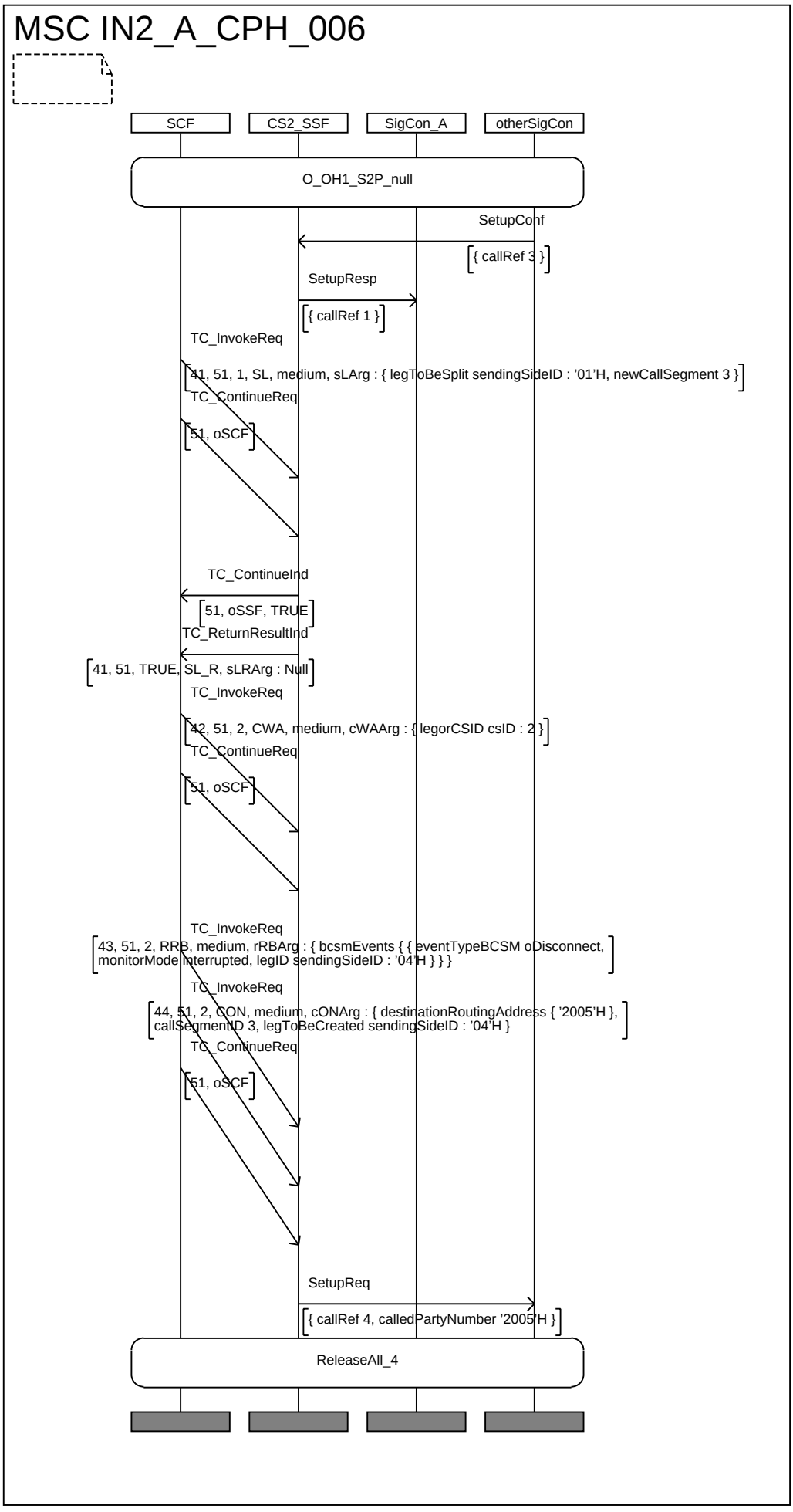
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult L1! ContinueWithArgument (CsID = 1) Reaching state O_S1P_null_null:
Pass criteria	CP1_2? ReleaseReq
Postamble:	ReleaseAll_1


MSC IN2_A_CPH_004
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 2, cause '0F'H }, senderRef
TC_InvokeReq
42, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode notifyAndContinue, legID sendingSideID : '01'H } } }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
ReleaseAll_1

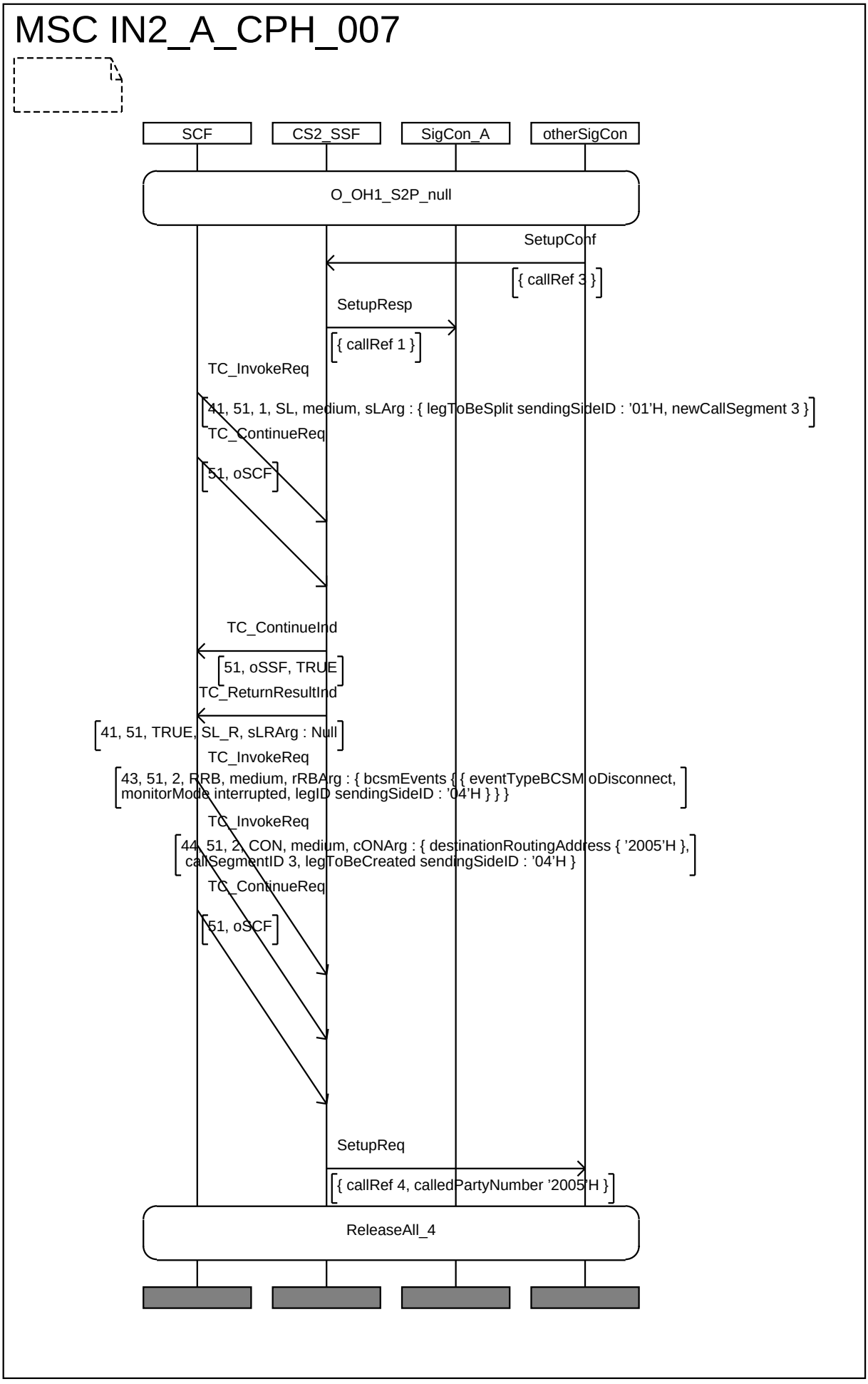
IN2_A_CPH_005	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test description	CP1-2! SetupConf, B party answers L1! InitiateCallAttempt(3,2) L1! ContinueWithArgument(LegId=3) Reaching state O_S2P_S1P_null
Pass criteria	CP1_3? SetupReq
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_005
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2101'H },
legToBeCreated sendingSideID : '03'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2101'H }
ReleaseAll_3

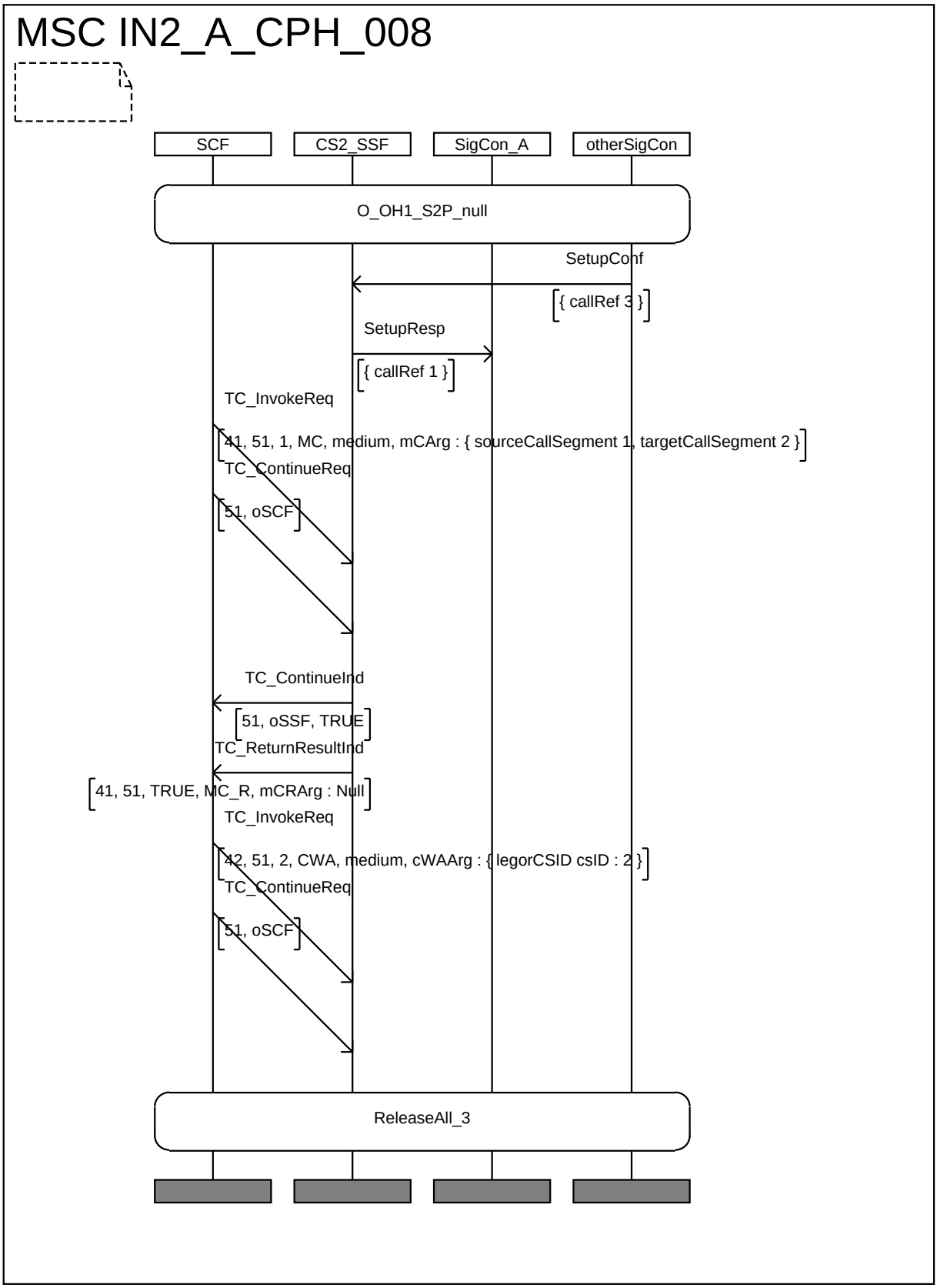
IN2_A_CPH_006	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	CP1-3! SetupConf, C party answers L1! SplitLeg(1,3) L1? SplitLegReturnResult L1! ContinueWithArgument (CsID = 2) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(4,3) Reaching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


MSC IN2_A_CPH_006
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '04'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2005'H }, callSegmentID 3, legToBeCreated sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2005'H }
ReleaseAll_4

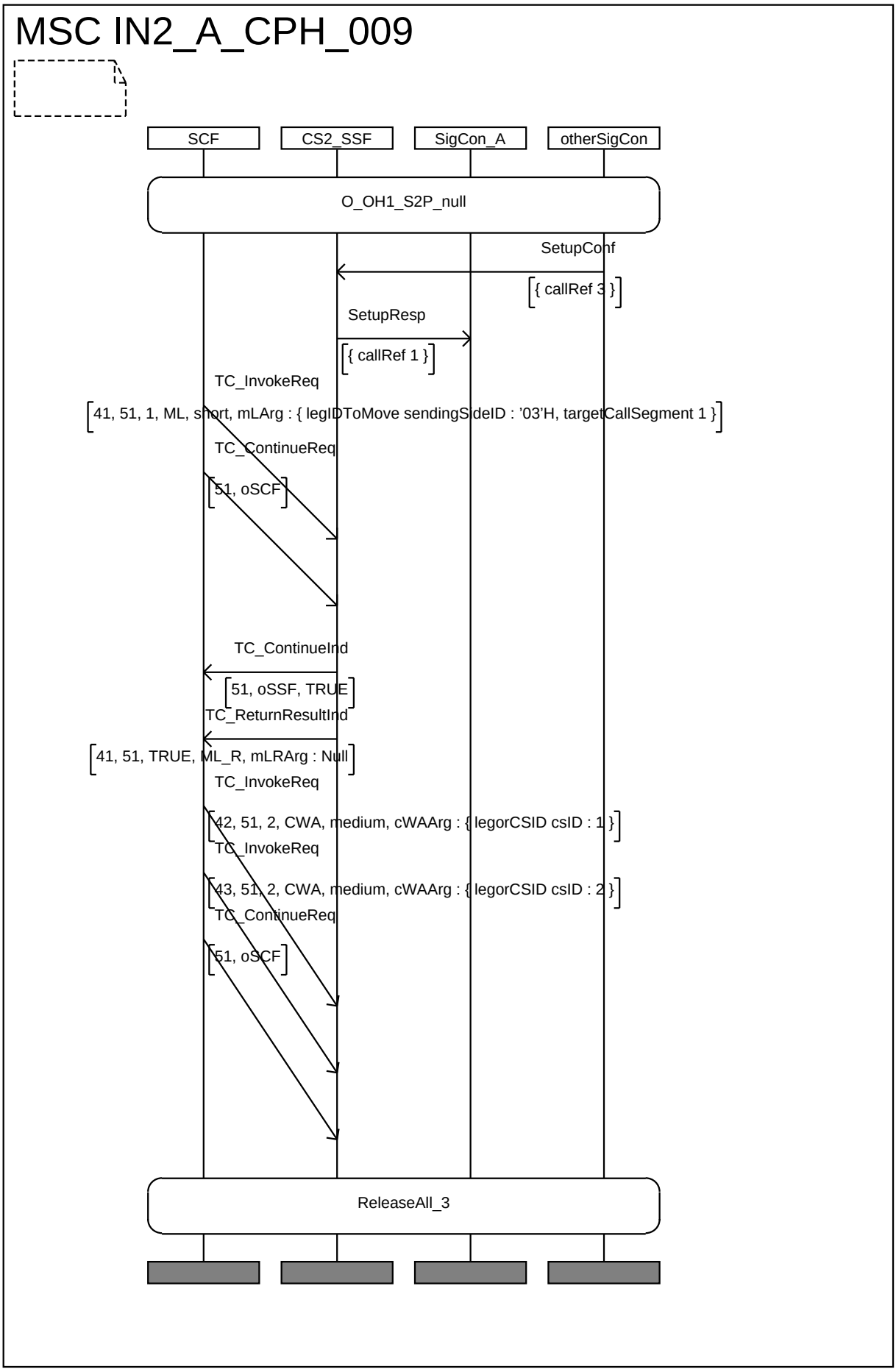
IN2_A_CPH_007	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	CP1-3! SetupConf, C party answers L1! SplitLeg(1,3) L1? SplitLegReturnResult L1! ContinueWithArgument (CsID = 2) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(4,3) Reaching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


MSC IN2_A_CPH_007
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '04'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2005'H }, callSegmentID 3, legToBeCreated sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2005'H }
ReleaseAll_4

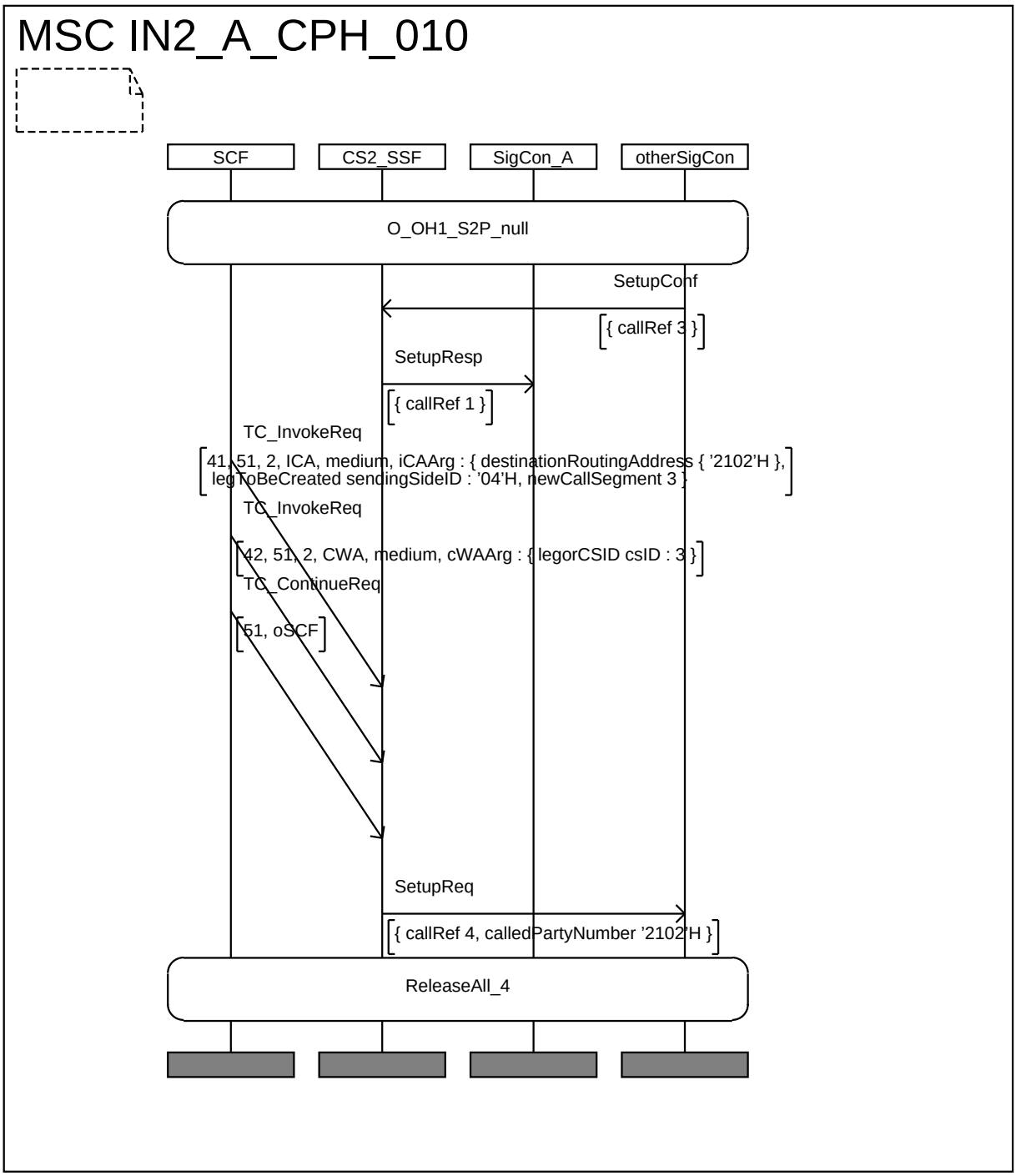
IN2_A_CPH_008	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	CP1-3! SetupConf, C party answers L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_S3P_null
Pass criteria	SSF sends MergeCallSegmentReturnResult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


MSC IN2_A_CPH_008
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_3

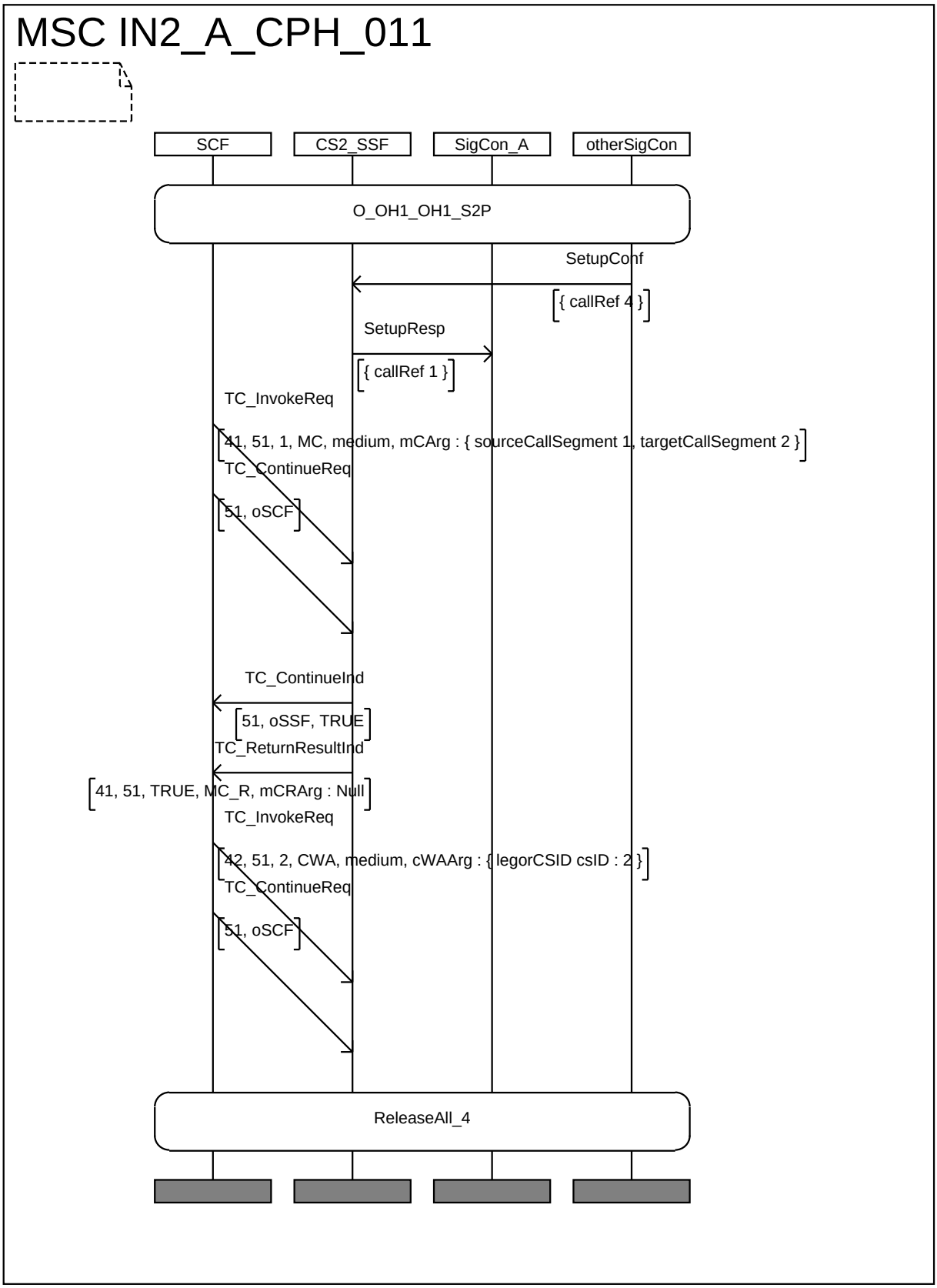
IN2_A_CPH_009	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	CP1-3! SetupConf, C party answers L1! MoveLeg(3,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=2) Reaching state O_OH(2)_1P_null
Pass criteria	SSF sends MoveLegReturnResult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


MSC IN2_A_CPH_009
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '03'H, targetCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_3

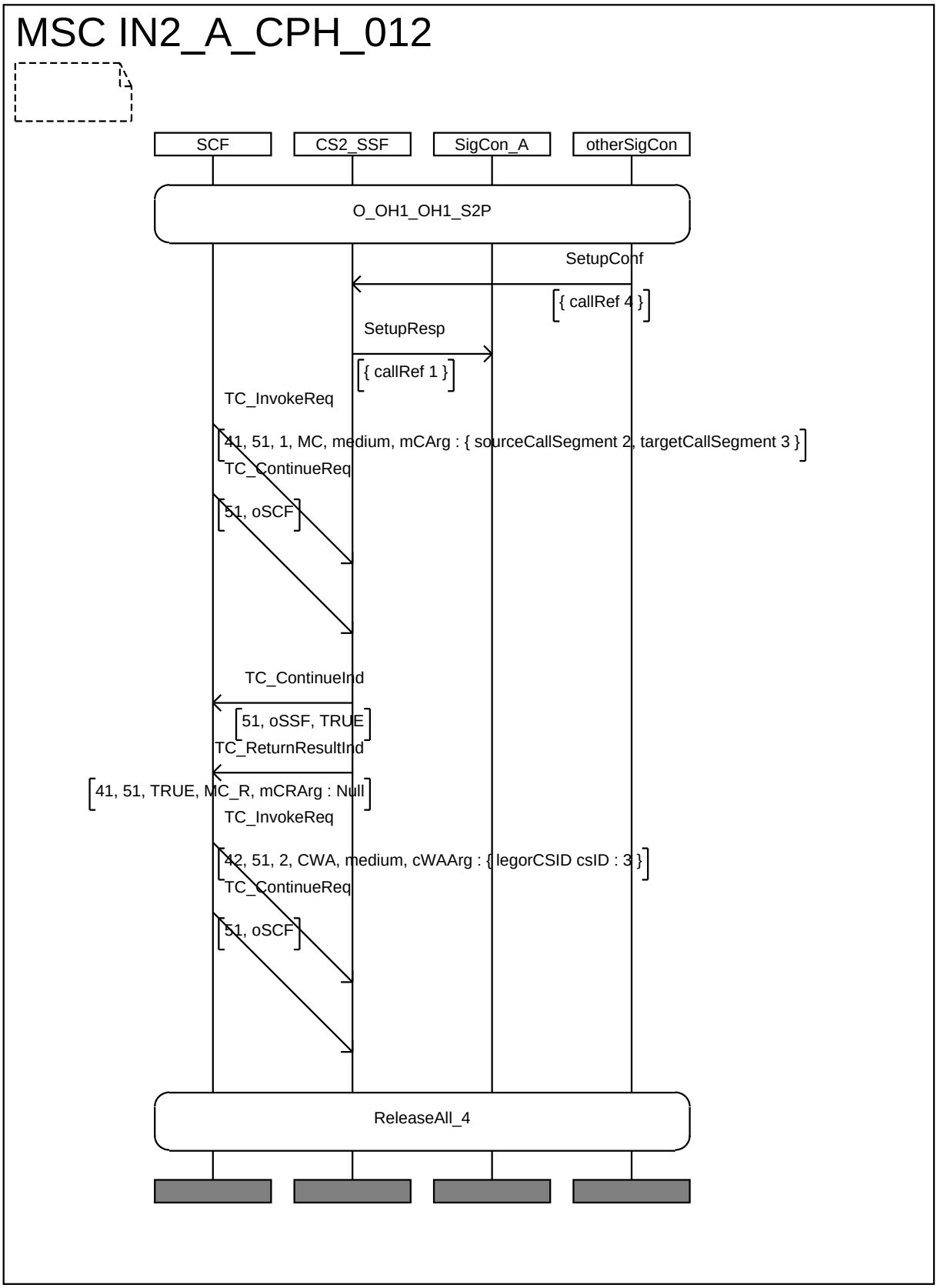
IN2_A_CPH_010	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_S2P_null
Test description	CP1-3! SetupConf, C party answers L1! InitiateCallAttempt(4,3) L1! ContinueWithArgument(LegId=4) Reaching state O_OH(1)_S2P_S1P
Pass criteria	CP1_4? SetUpReq
Postamble:	ReleaseAll_4


MSC IN2_A_CPH_010
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_S2P_null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2102'H },
legToBeCreated sendingSideID : '04'H, newCallSegment 3 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2102'H }
ReleaseAll_4

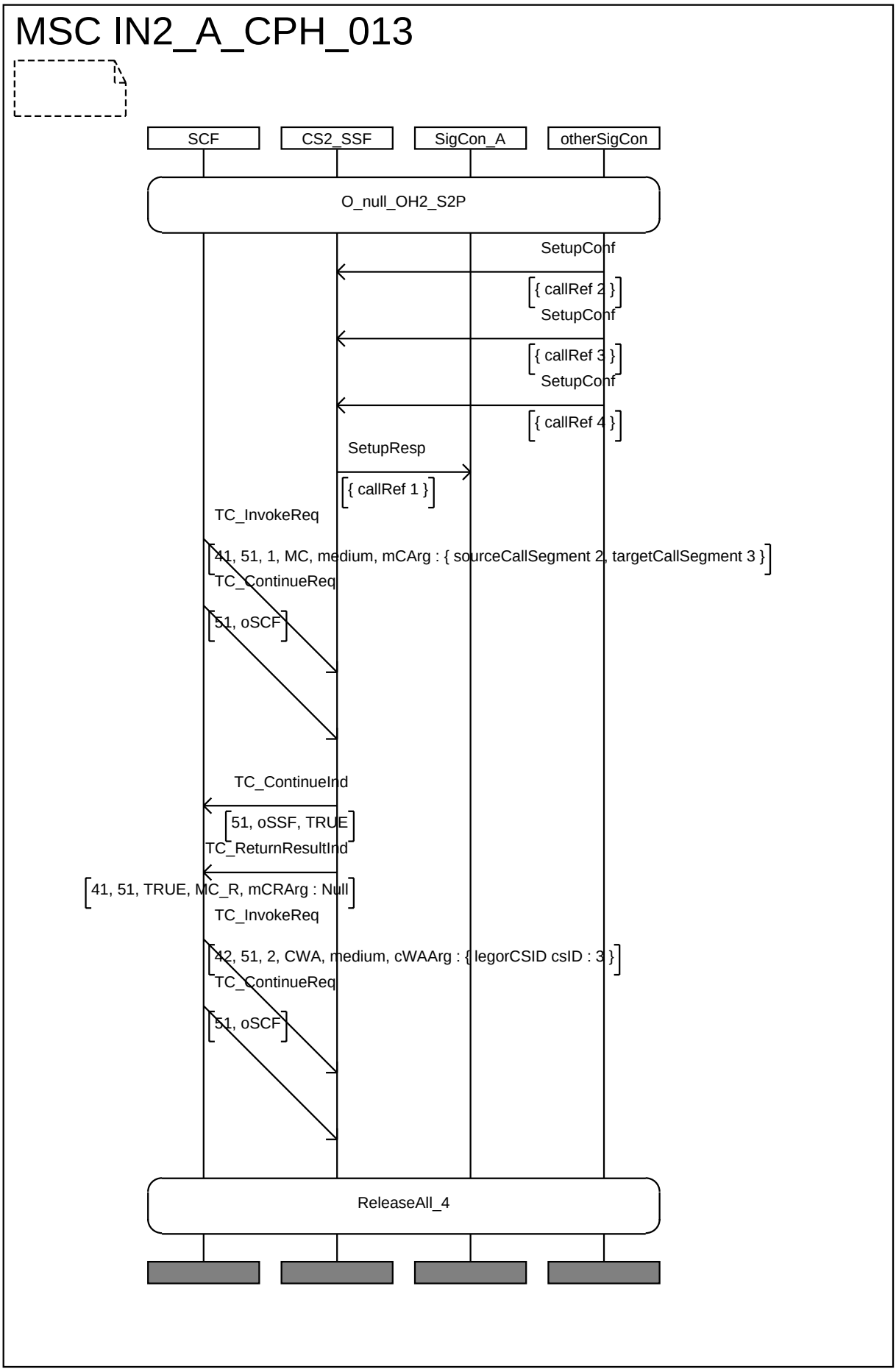
IN2_A_CPH_011	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_OH(1)_S2P
Test description	CP1-4! SetupConf L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_OH(2)_S2P
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble:	ReleaseAll_4


MSC IN2_A_CPH_011
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_OH1_S2P
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

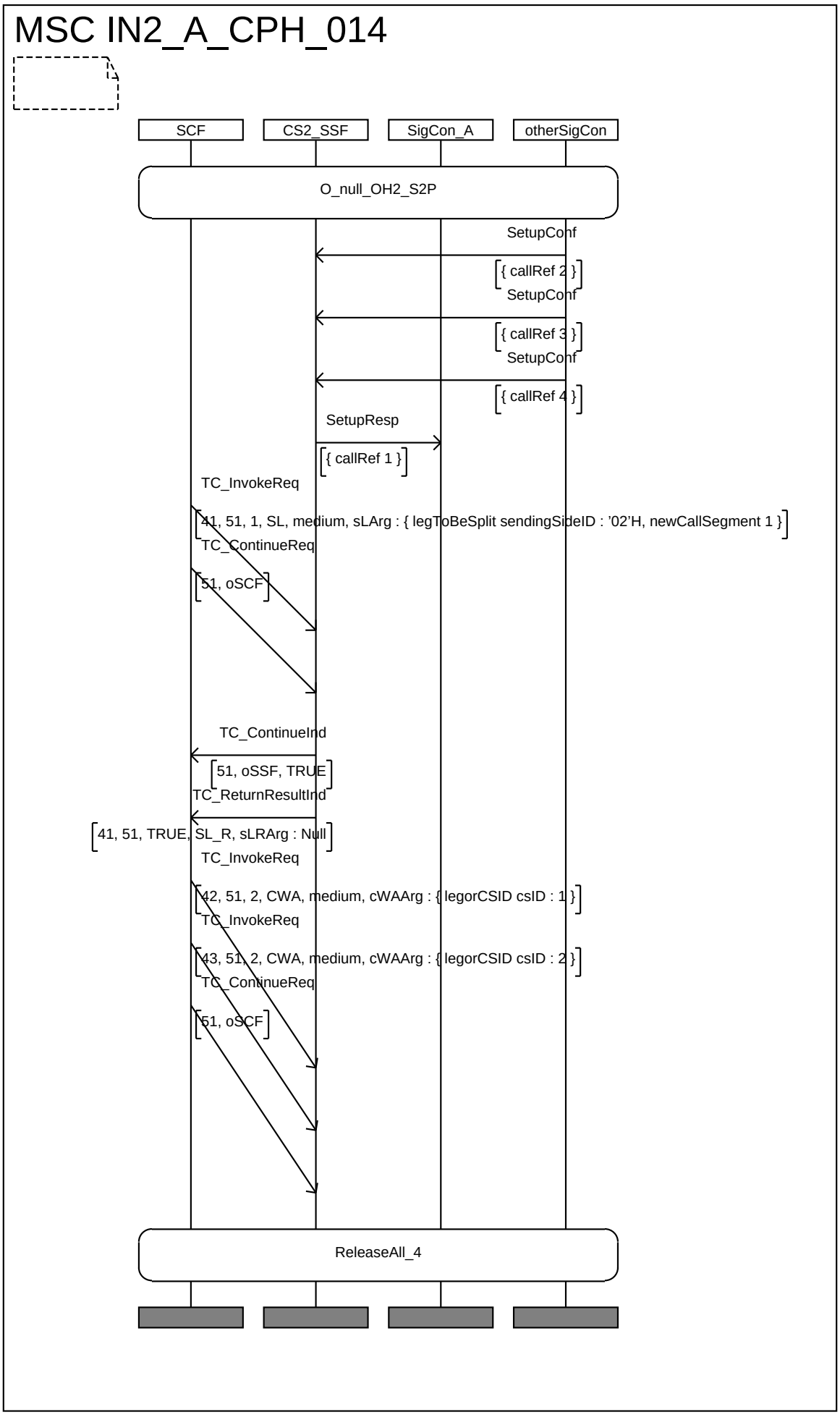
IN2_A_CPH_012	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_OH(1)_OH(1)_S2P
Test description	CP1-4! SetupConf L1! MergeCallSegments(2,3) L1! ContinueWithArgument(CsId=3) Reaching state O_OH(1)_null_S3P
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble:	ReleaseAll_4


MSC IN2_A_CPH_012
SCF
CS2_SSF
SigCon_A
otherSigCon
O_OH1_OH1_S2P
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 2, targetCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

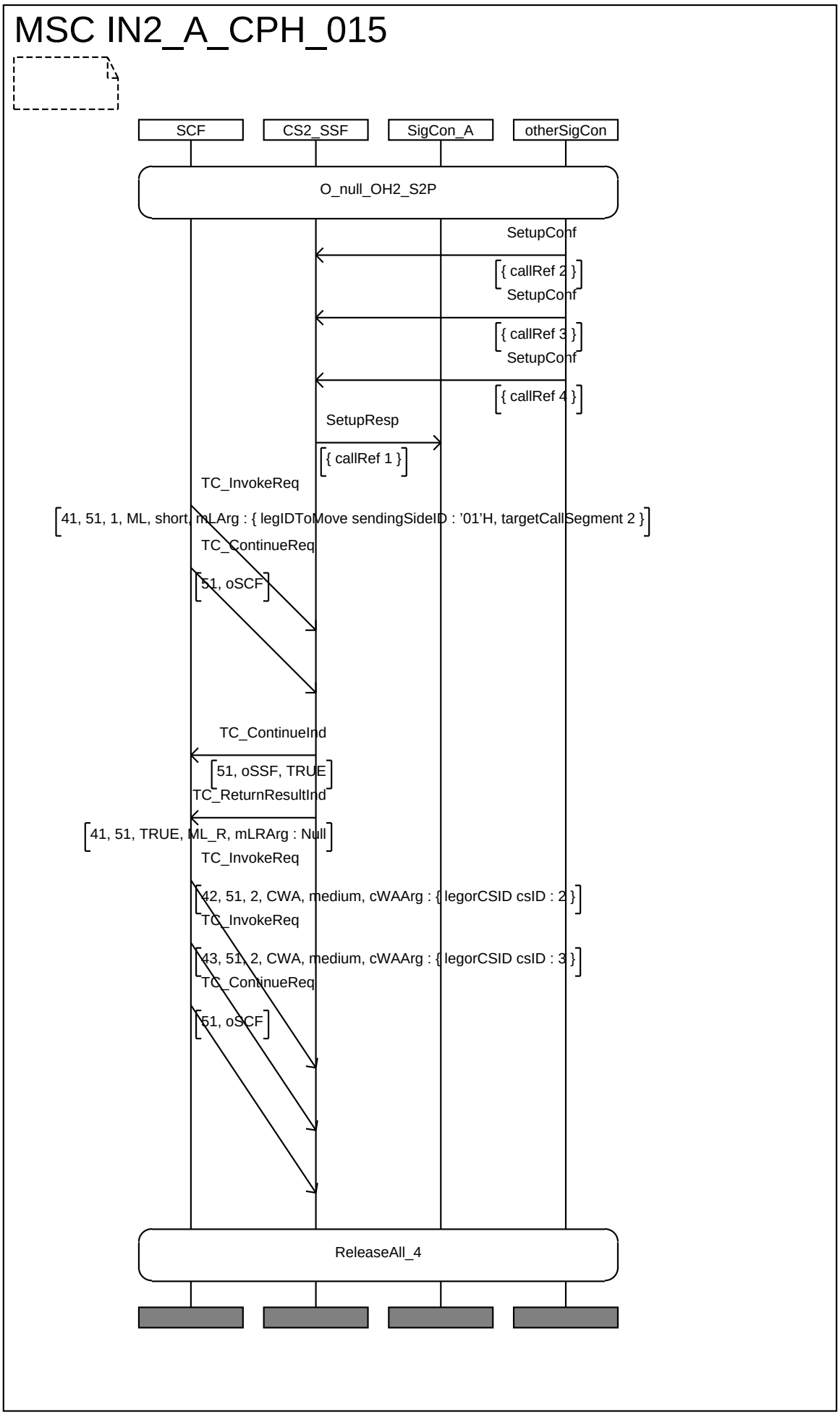
IN2_A_CPH_013	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P
Test description	CP1-2! SetupConf CP1-3! SetupConf CP1-4! SetupConf L1! MergeCallSegments(2,3) L1! ContinueWithArgument(CsId=3) Reaching state O_null_null_S4P
Pass criteria	SSF sends MergeCallSegmentReturnResult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


MSC IN2_A_CPH_013
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_OH2_S2P
SetupConf
{ callRef 2 }
SetupConf
{ callRef 3 }
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 2, targetCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

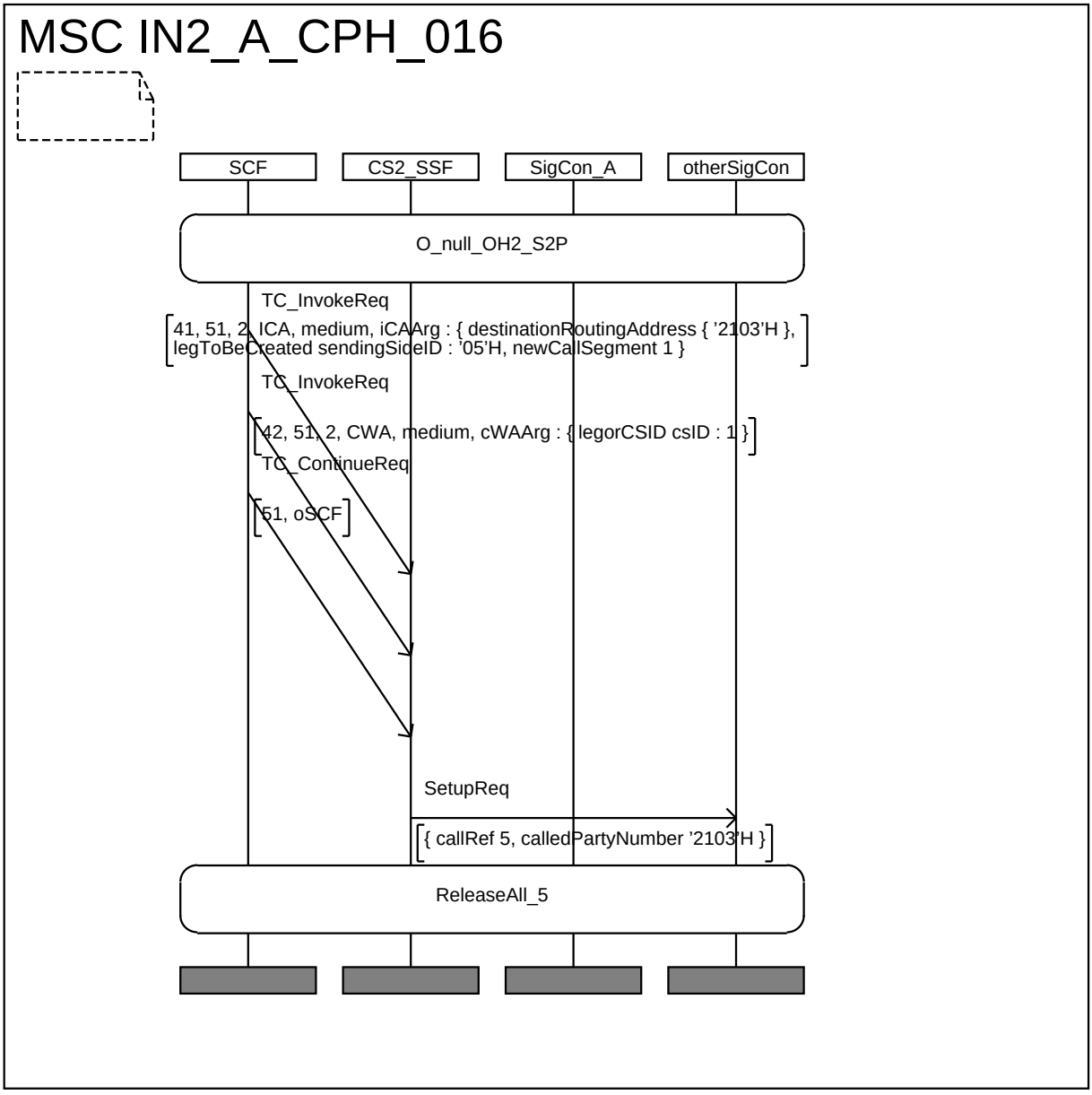
IN2_A_CPH_014	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P
Test description	CP1-2! SetupConf CP1-3! SetupConf CP1-4! SetupConf L1! SplitLeg(2,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=2) Reaching state O_OH(1)_OH(1)_S2P
Pass criteria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


MSC IN2_A_CPH_014
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_OH2_S2P
SetupConf
{ callRef 2 }
SetupConf
{ callRef 3 }
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

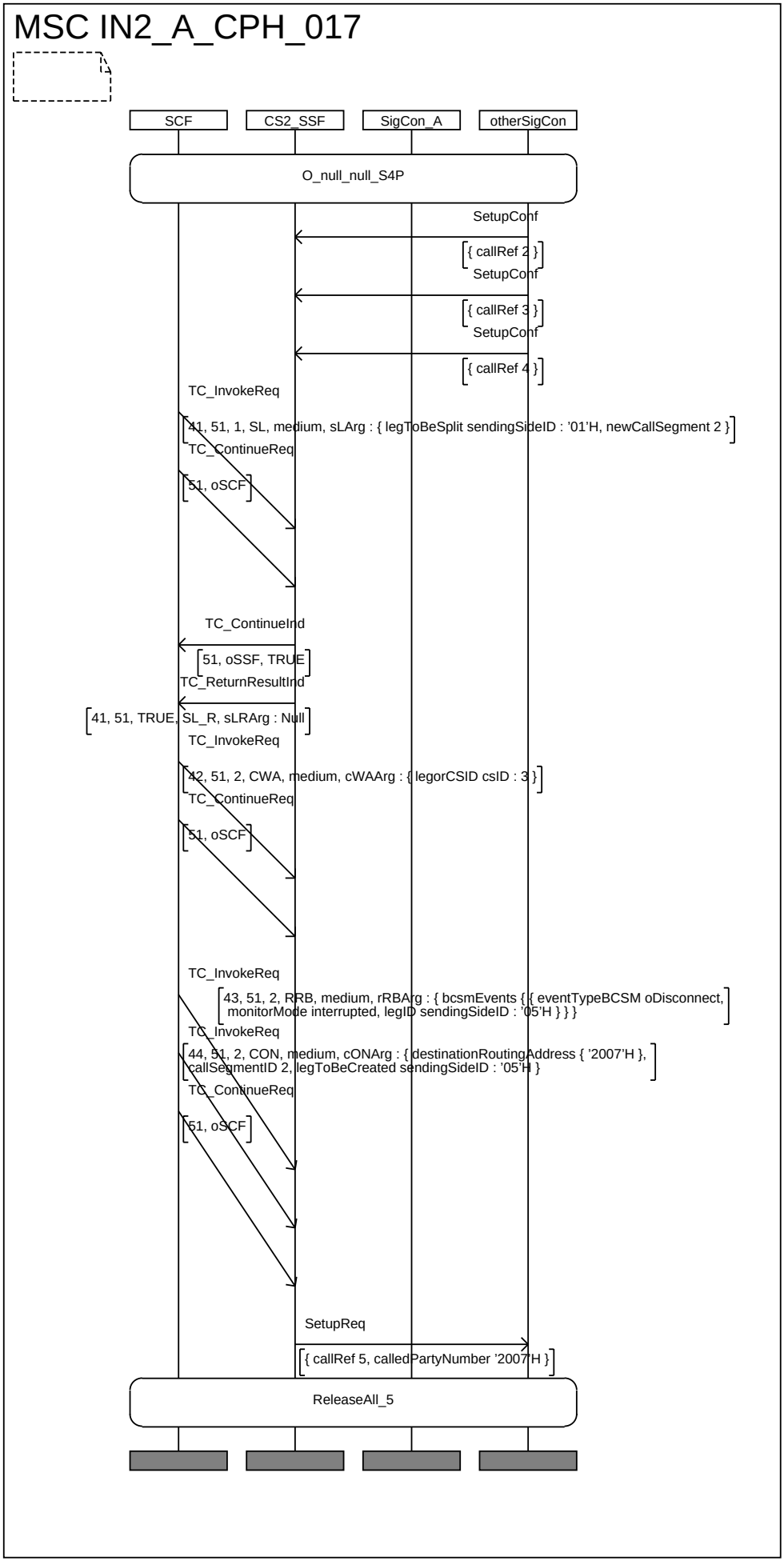
IN2_A_CPH_015	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P
Test description	CP1-2! SetupConf CP1-3! SetupConf, C party answers CP1-4! SetupConf, D party answers L1! MoveLeg(1,2) L1!ContinueWithArgument(CsId=2) L1!ContinueWithArgument(CsId=3) Reaching state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


MSC IN2_A_CPH_015
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_OH2_S2P
SetupConf
{ callRef 2 }
SetupConf
{ callRef 3 }
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

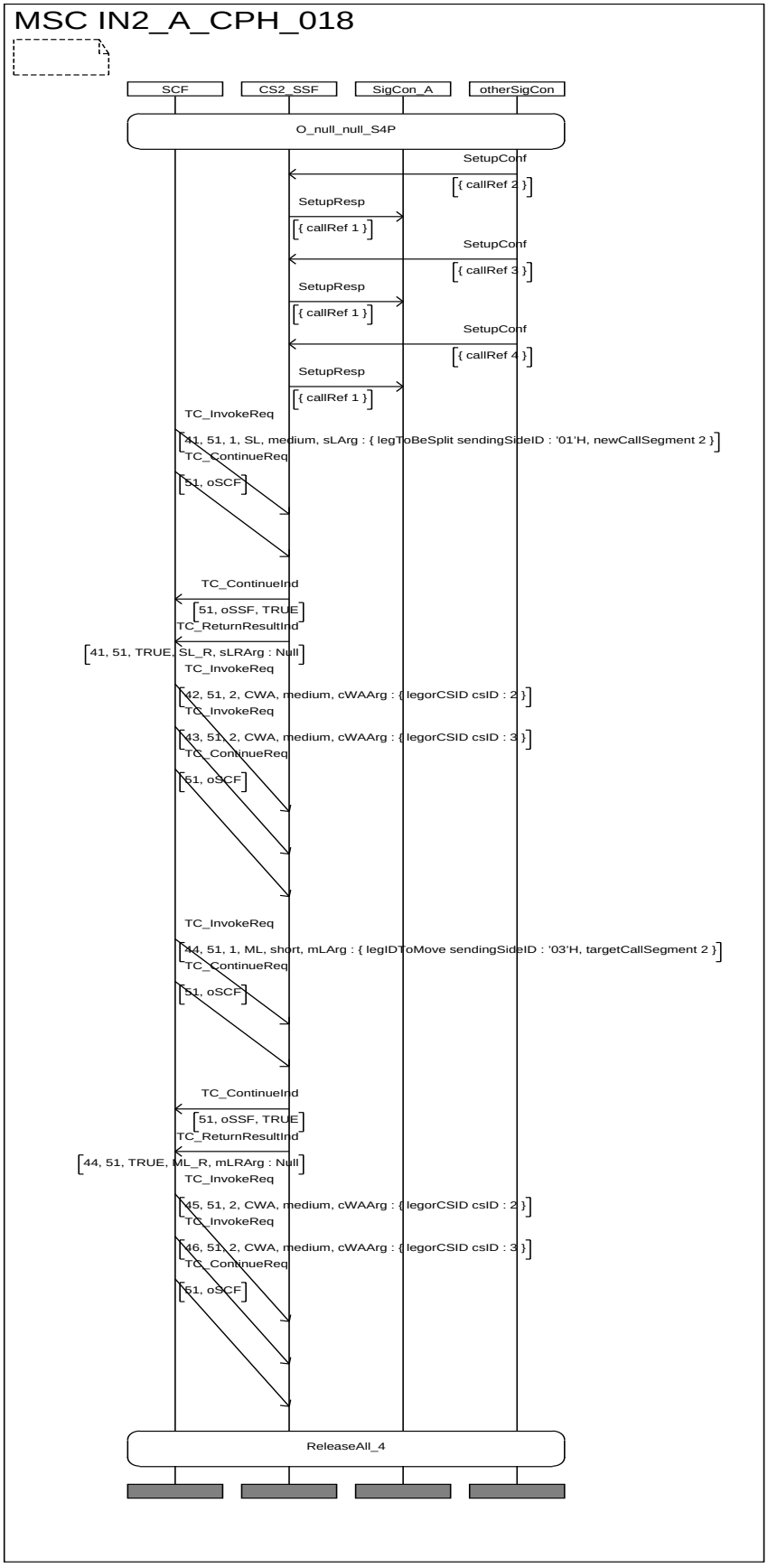
IN2_A_CPH_016	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P
Test description	L1! InitiateCallAttempt(5,1) L1! ContinueWithArgument(LegId=5) Reaching state O_S1P_OH(2)_S2P
Pass criteria	CP1_5? SetUpReq
Postamble:	ReleaseAll_5



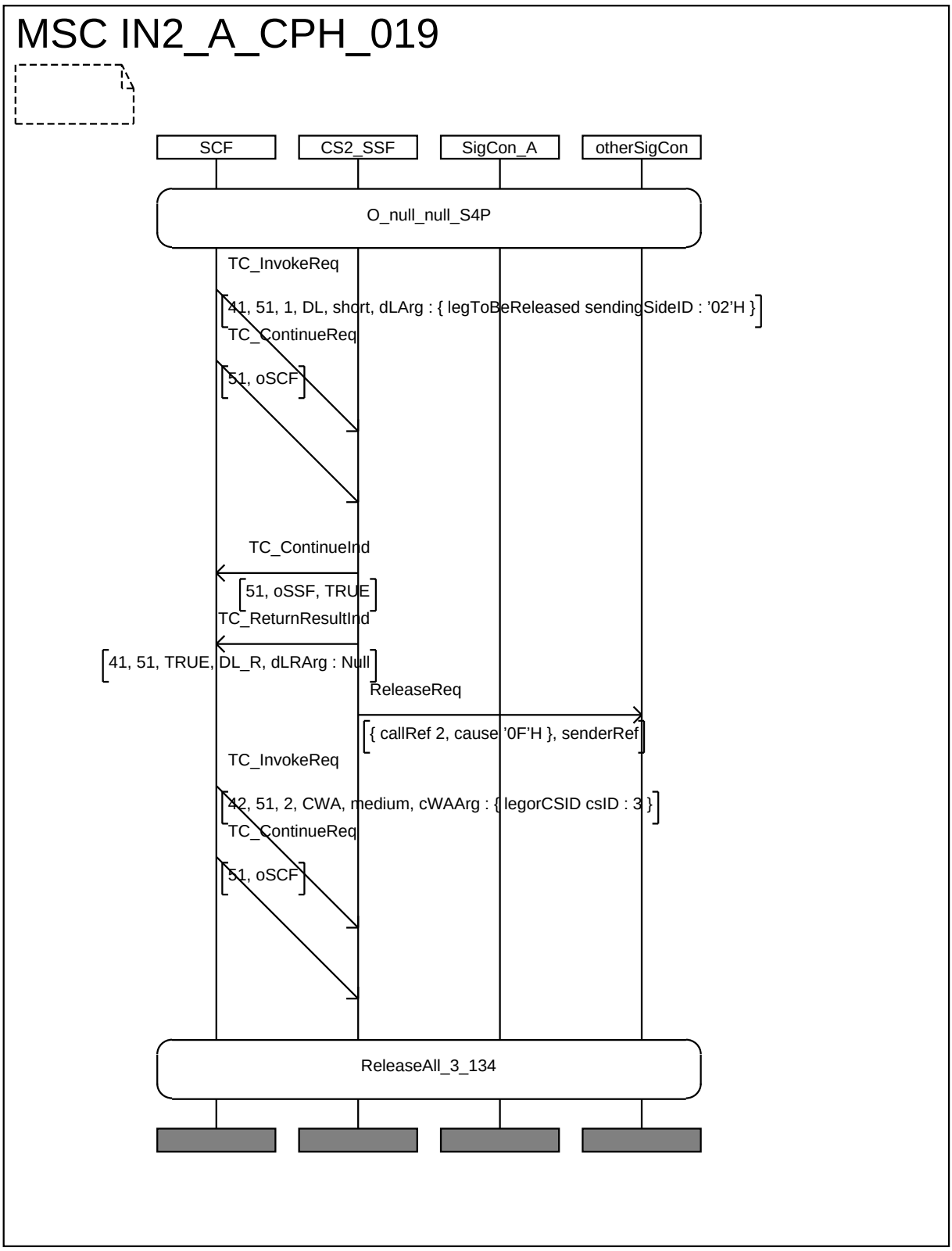
IN2_A_CPH_017	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P
Test description	CP1-2! SetupConf CP1-3! SetupConf, C party answers CP1-4! SetupConf, D party answers L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RequestReportBCSMEvent(5,oDisconnect)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


MSC IN2_A_CPH_017
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P
SetupConf
{ callRef 2 }
SetupConf
{ callRef 3 }
SetupConf
{ callRef 4 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '05'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2007'H }, callSegmentID 2, legToBeCreated sendingSideID : '05'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 5, calledPartyNumber '2007'H }
ReleaseAll_5

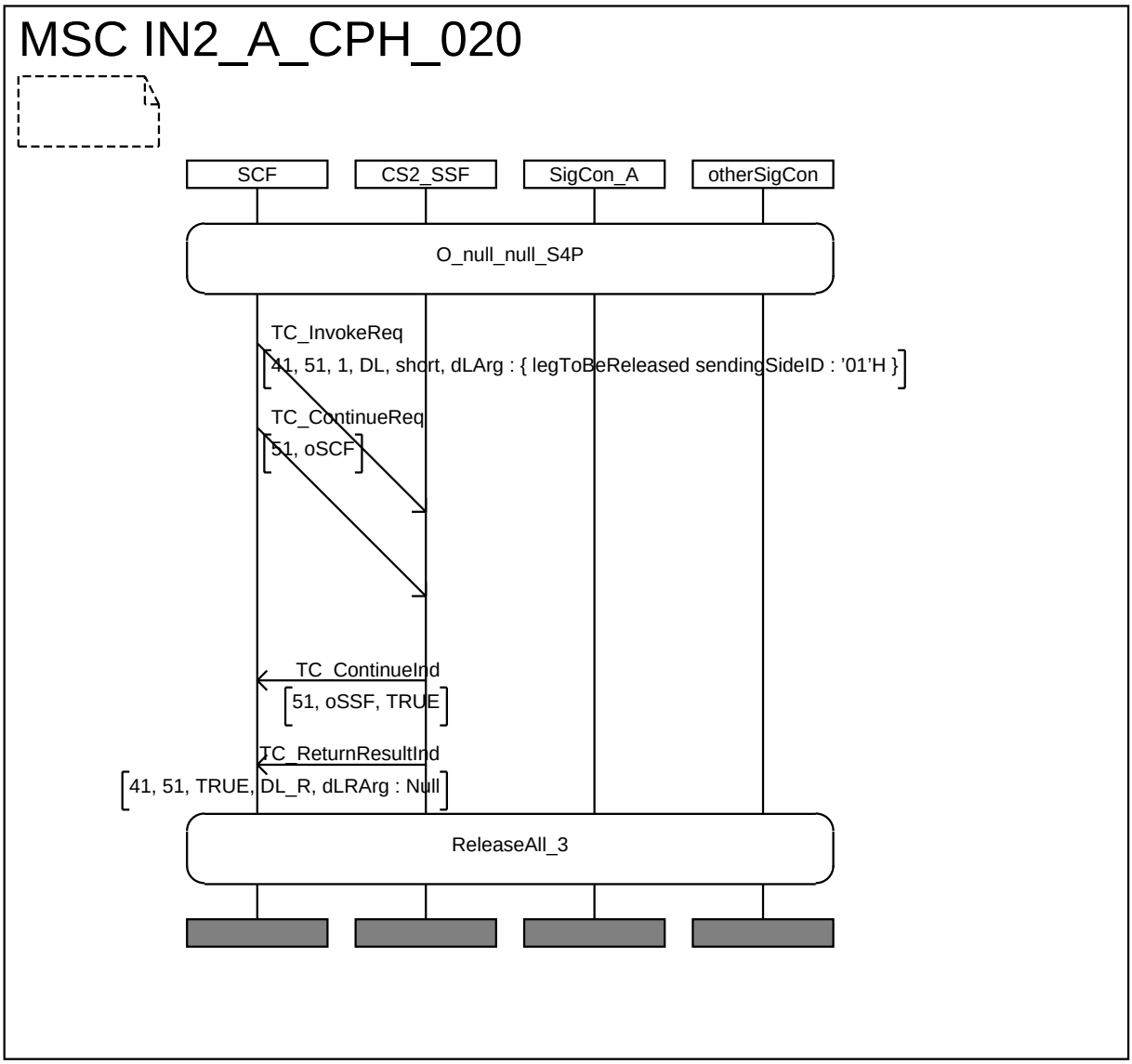
IN2_A_CPH_018	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P
Test description	CP1-2! SetupConf CP1-3! SetupConf, C party answers CP1-4! SetupConf, D party answers L1! SplitLeg(1,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) L1! MoveLeg(3,2)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


MSC IN2_A_CPH_018
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
44, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '03'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
44, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
45, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
46, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

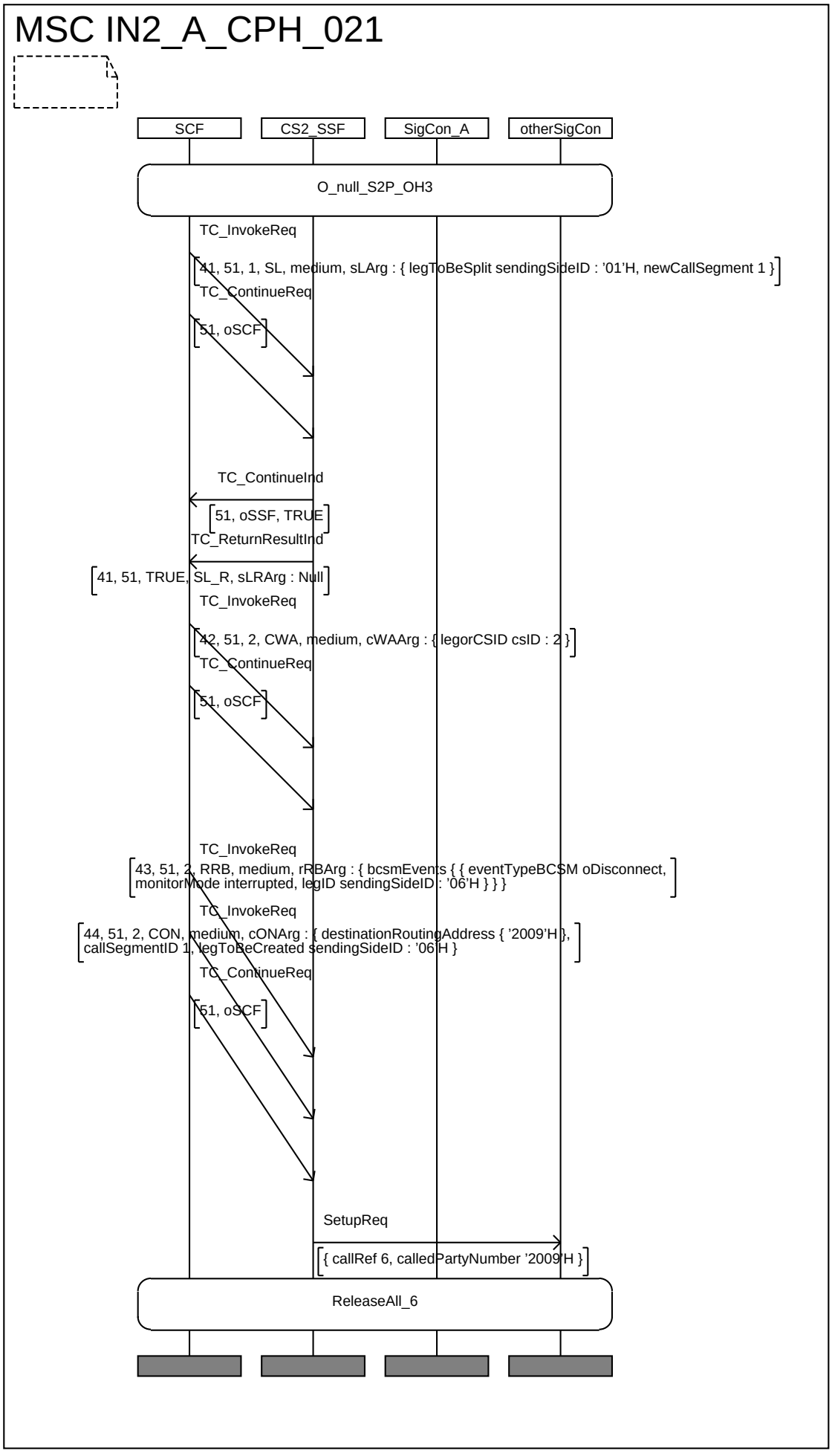
IN2_A_CPH_019	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P
Test description	L1! DisconnectLeg(2) L1!ContinueWithArgument(CsId=3) L1? DisconnectLegReturnResult Reaching state O_null_null_S3P
Pass criteria	CP1_2? ReleaseReq
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_019
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 2, cause '0F'H }, senderRef
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_3_134

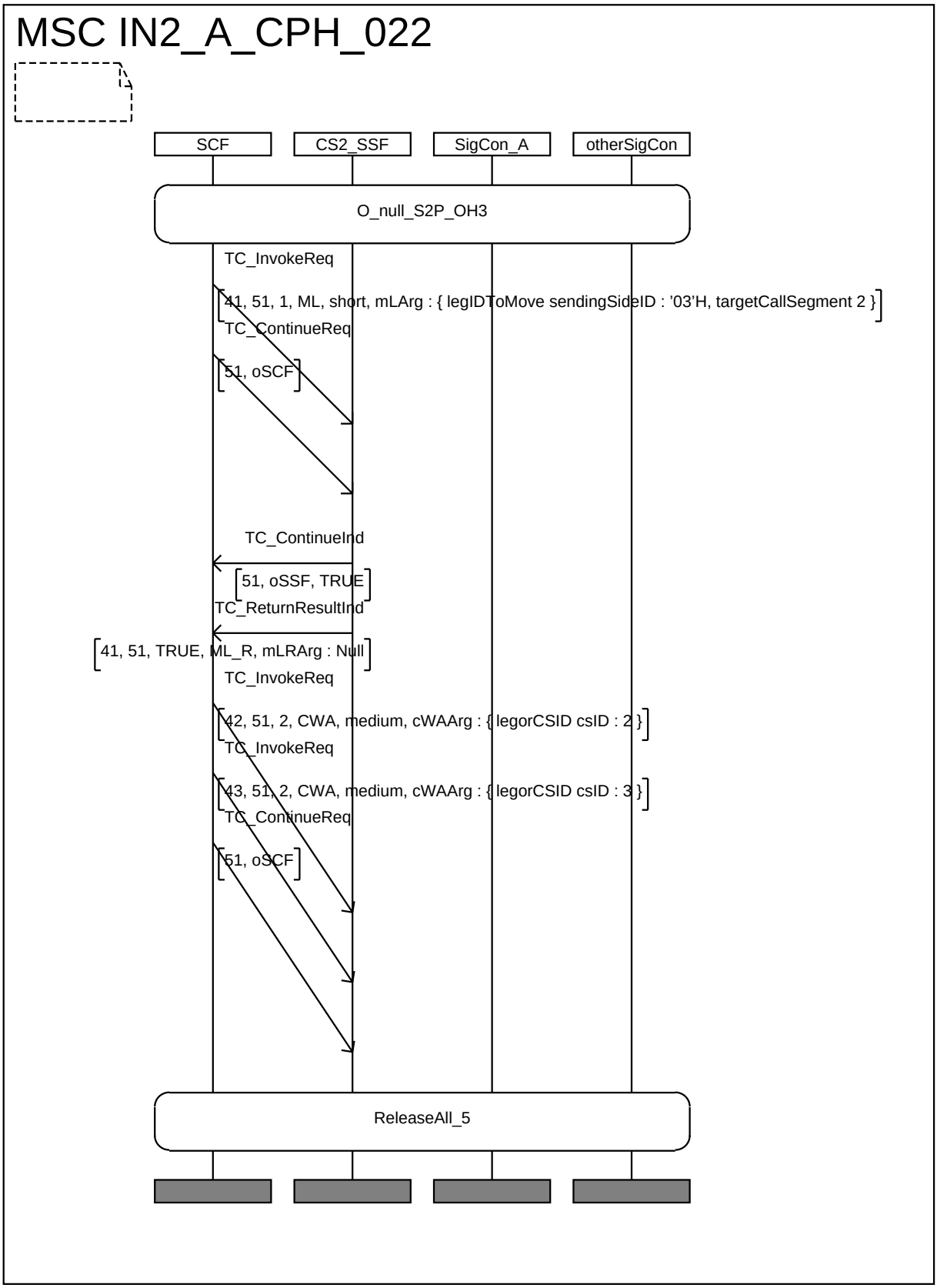
IN2_A_CPH_020	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P
Test description	L1! DisconnectLeg(1) L1!ContinueWithArgument(CsId=3) L1? DisconnectLegReturnResult Reaching state O_null_null_TF(3): (it could be reconnected afterwards)
Pass criteria	CP1_1? ReleaseReq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



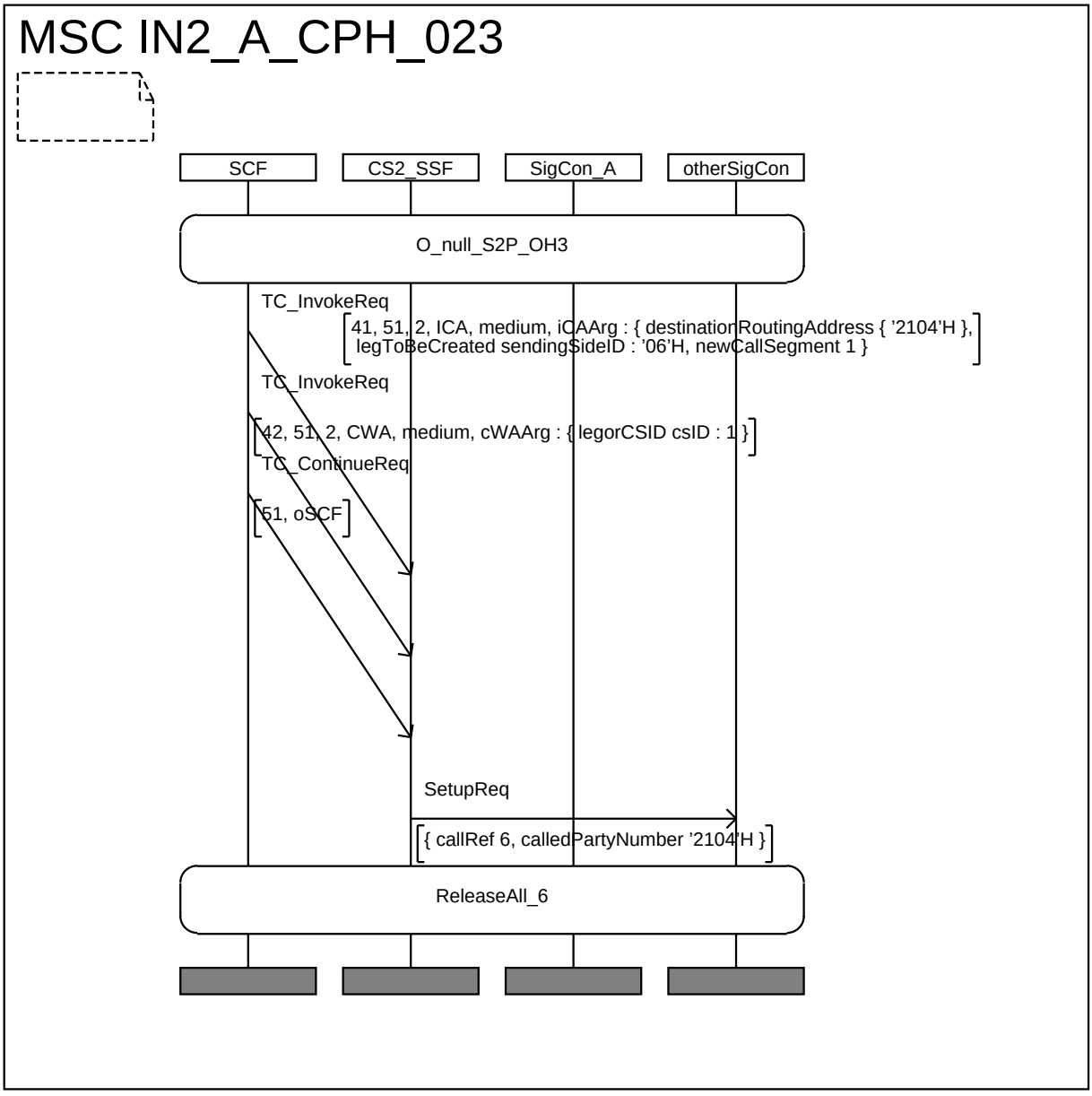
IN2_A_CPH_021	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S2P_OH(3)
Test description	L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CSId=2) L1! RequestReportBCSMEvent(6,oDisconnect) L1! Connect(6,1) Reaching state O_S2P_OH(1)_OH(3)
Pass criteria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


MSC IN2_A_CPH_021
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S2P_OH3
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '06'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2009'H }, callSegmentID 1, legToBeCreated sendingSideID : '06'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 6, calledPartyNumber '2009'H }
ReleaseAll_6

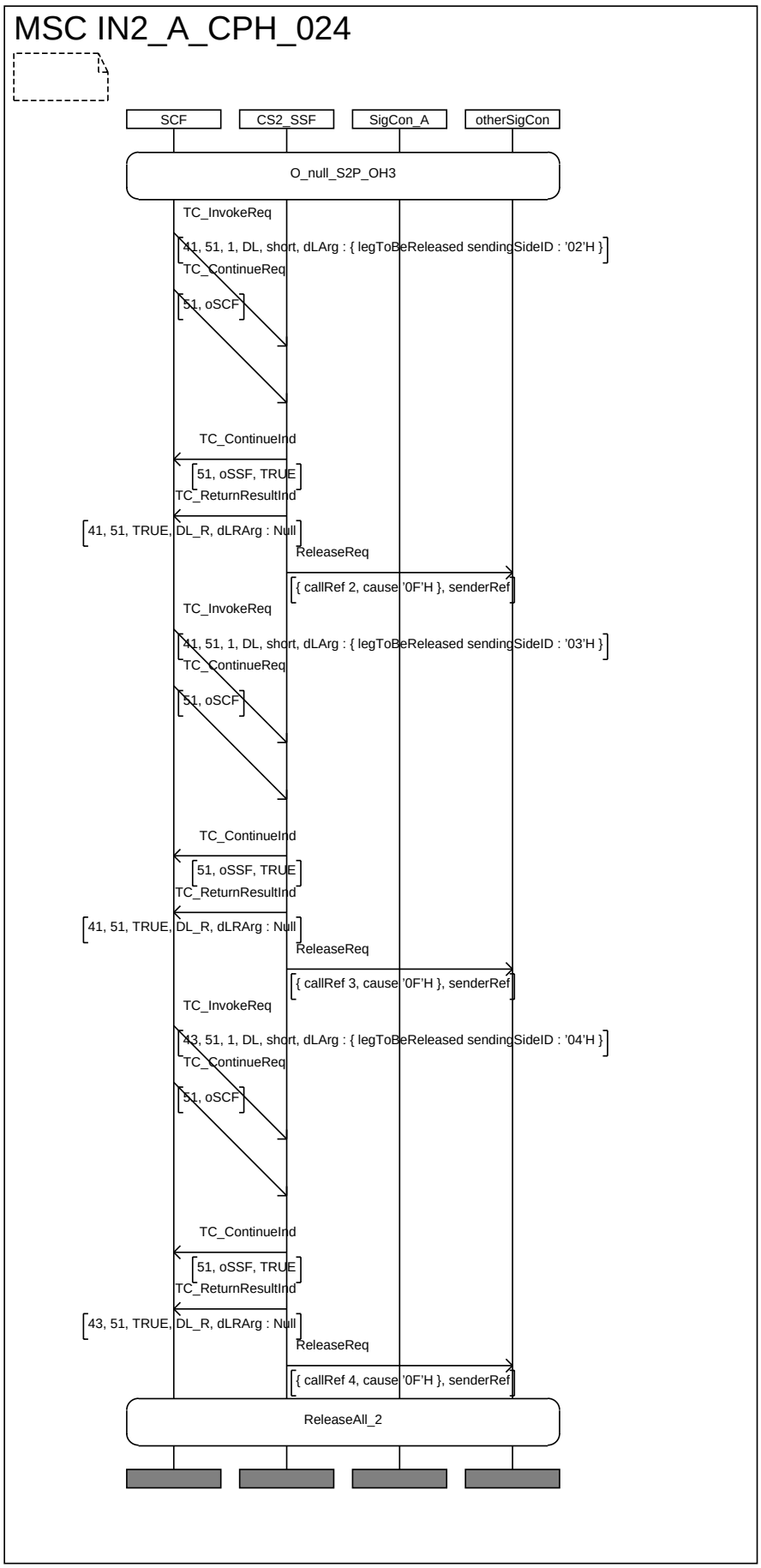
IN2_A_CPH_022	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S2P_OH(3)
Test description	L1! MoveLeg(3,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state O_null_S3P_OH(2)
Pass criteria	SSF sends MoveLegReturnResult
Postamble:	ReleaseAll_5


MSC IN2_A_CPH_022
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S2P_OH3
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '03'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_5

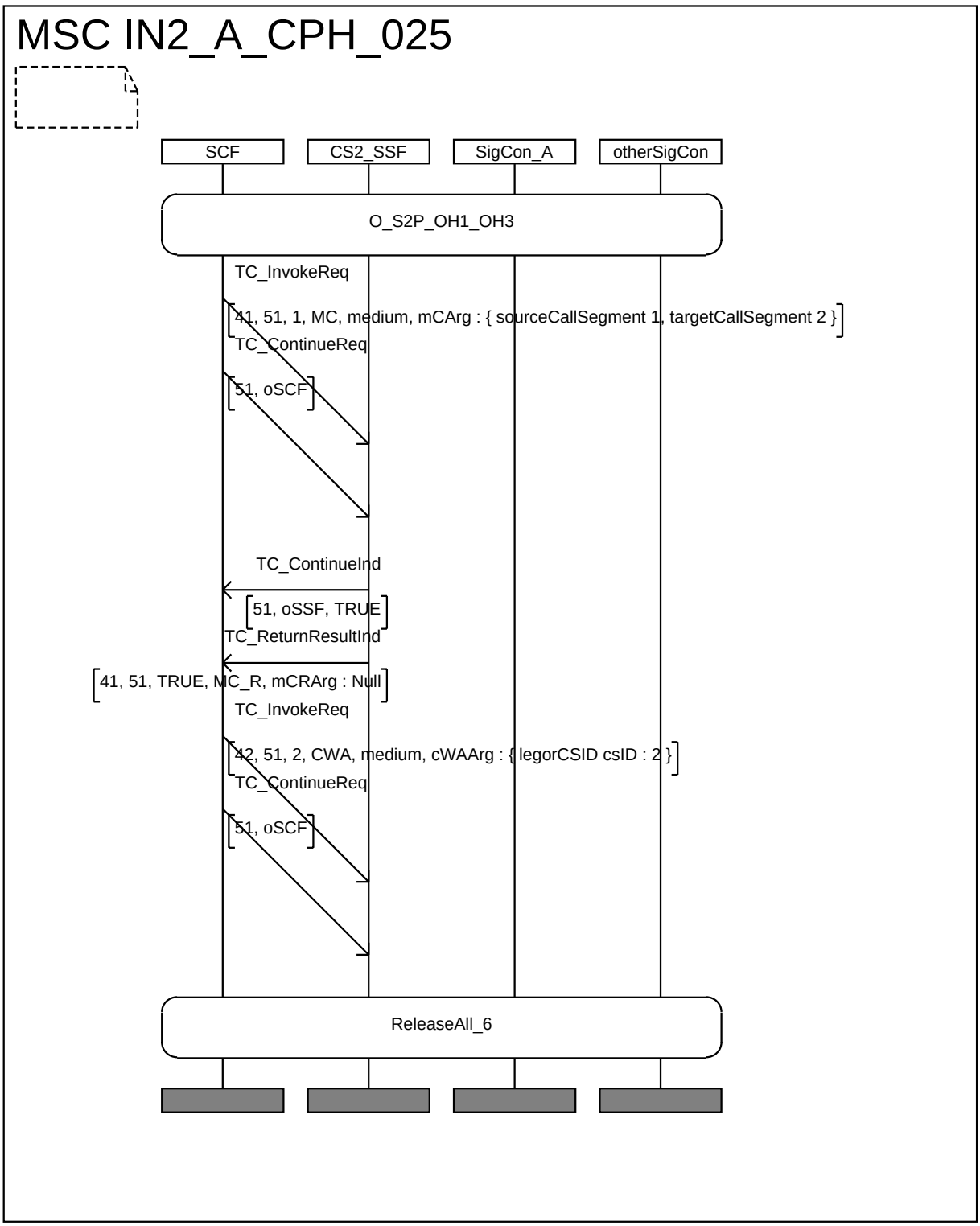
IN2_A_CPH_023	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S2P_OH(3)
Test description	L1! InitiateCallAttempt(6,1) L1! ContinueWithArgument(LegId=6) Reaching state O_S1P_S2P_OH(3)
Pass criteria	CP1_6? SetUpReq
Postamble:	ReleaseAll_6



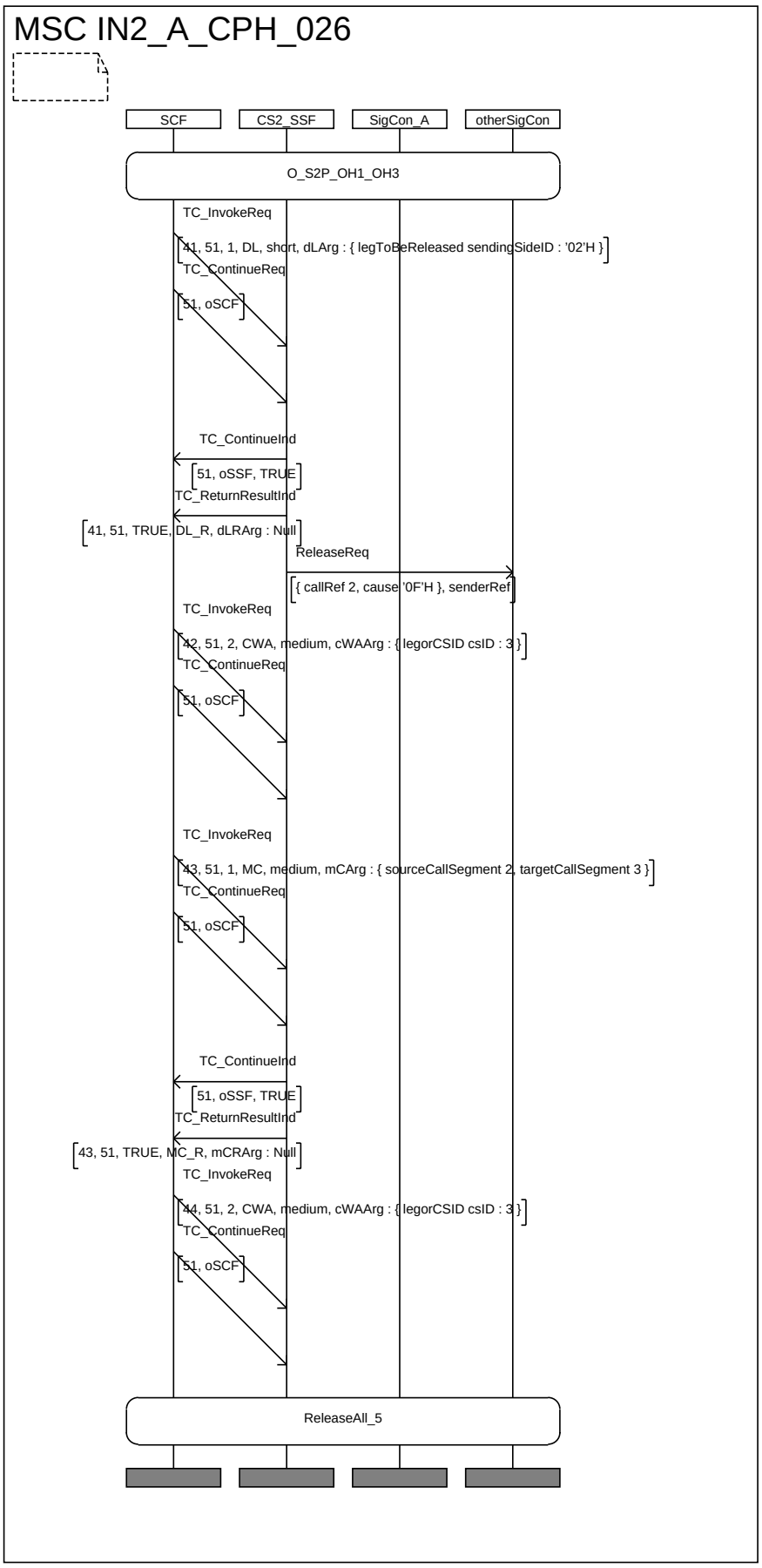
IN2_A_CPH_024	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S2P_OH(3)
Test description	L1! DisconnectLeg(2) L1! DisconnectLeg(3) L1! DisconnectLeg(4) L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult Reaching state O_null_S2P_null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


MSC IN2_A_CPH_024
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S2P_OH3
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 2, cause '0F'H }, senderRef
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 3, cause '0F'H }, senderRef
TC_InvokeReq
43, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
43, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 4, cause '0F'H }, senderRef
ReleaseAll_2

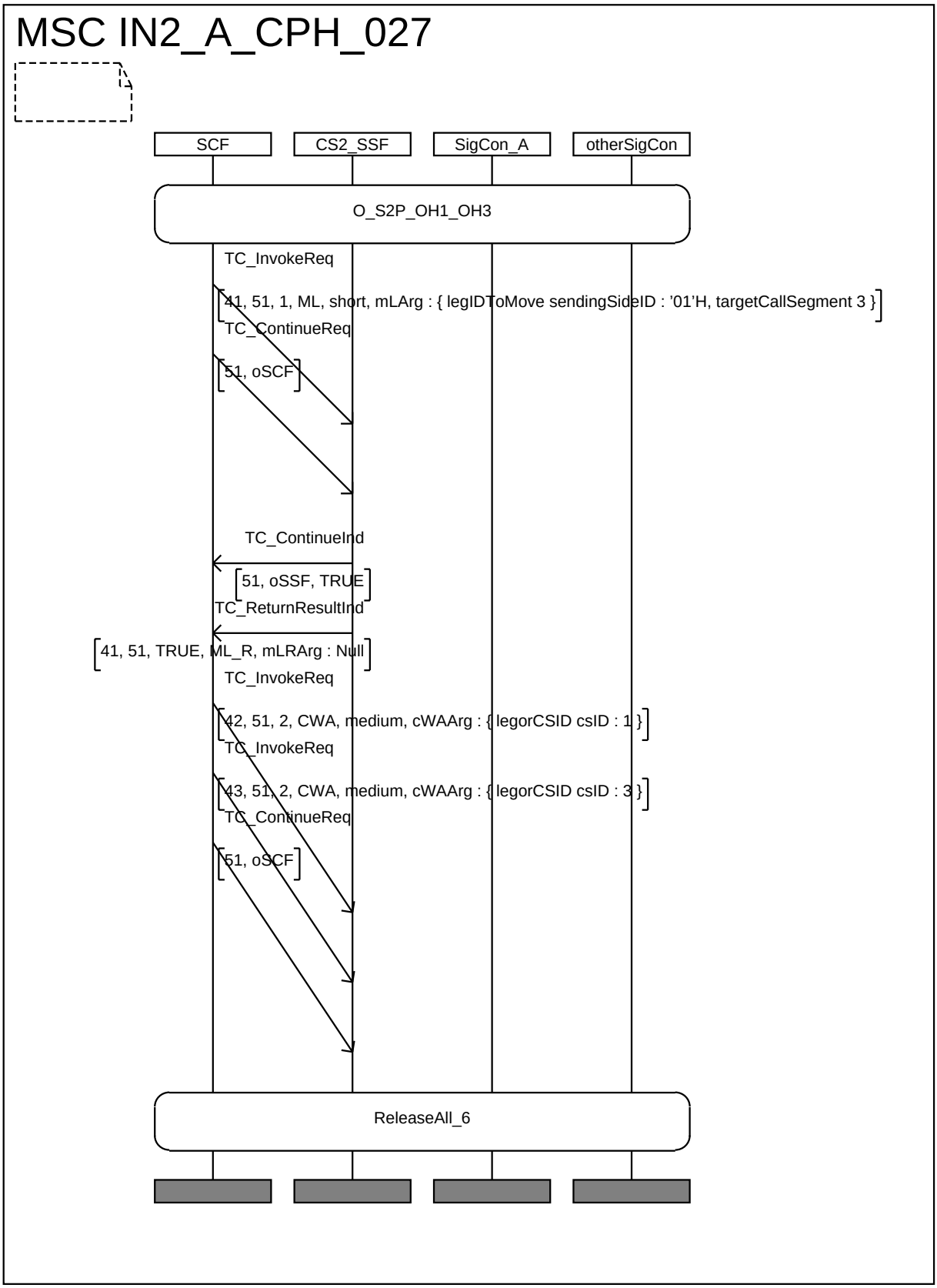
IN2_A_CPH_025	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(1)_OH(3)
Test description	L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_S3P_OH(3)
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble:	ReleaseAll_6


MSC IN2_A_CPH_025
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH1_OH3
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_6

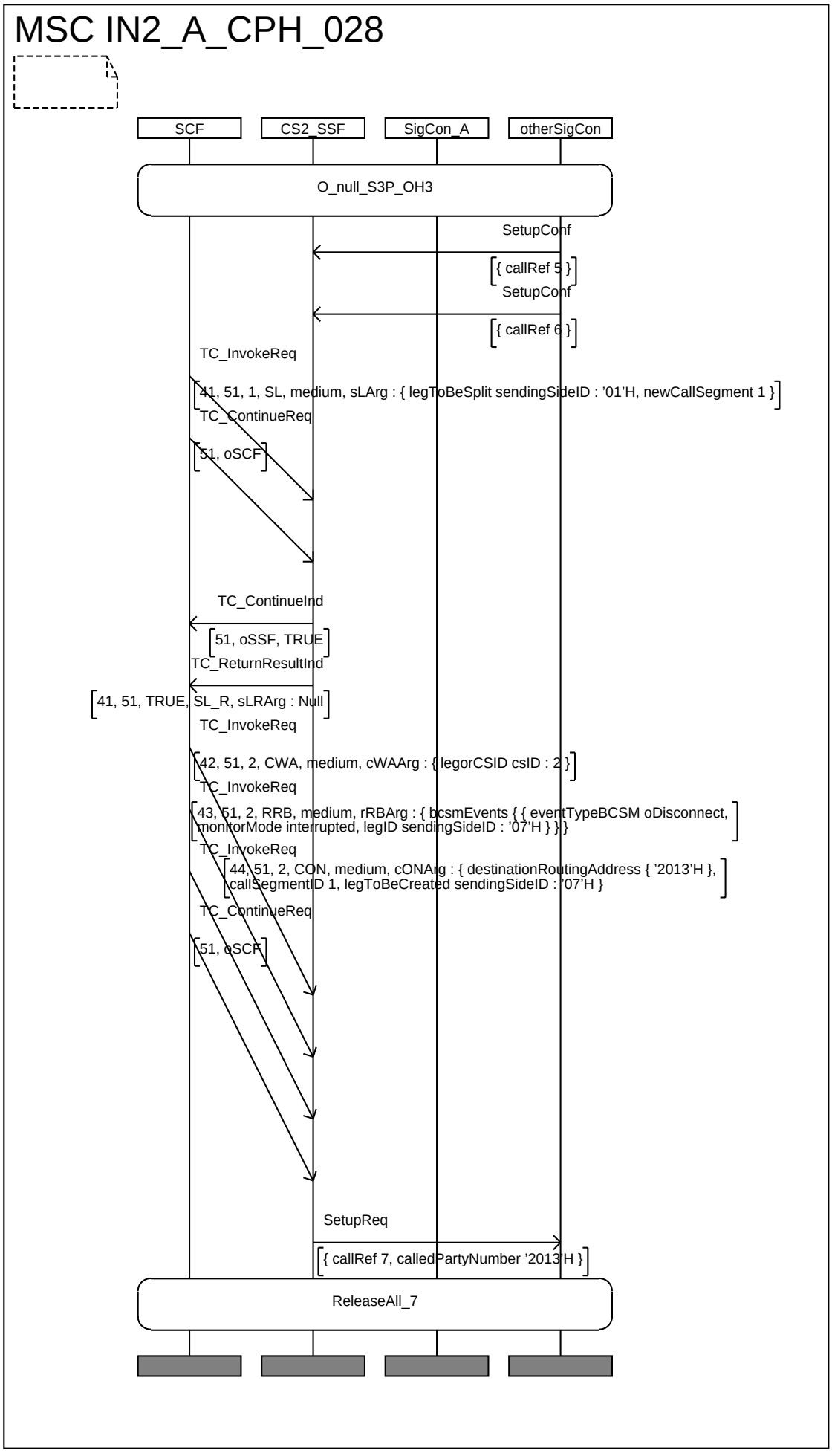
IN2_A_CPH_026	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(1)_OH(3)
Test description	L1! DisconnectLeg(2) L1! ContinueWithArgument(CsId=3) L1? DisconnectLegReturnResult CP1-2? ReleaseReq L1! MergeCallSegment(2,3) L1? MergeCallSegmentReturnResult L1! ContinueWithArgument(CsId=3)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


MSC IN2_A_CPH_026
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH1_OH3
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 2, cause '0F'H }, senderRef
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
43, 51, 1, MC, medium, mCArg : { sourceCallSegment 2, targetCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
43, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
44, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_5

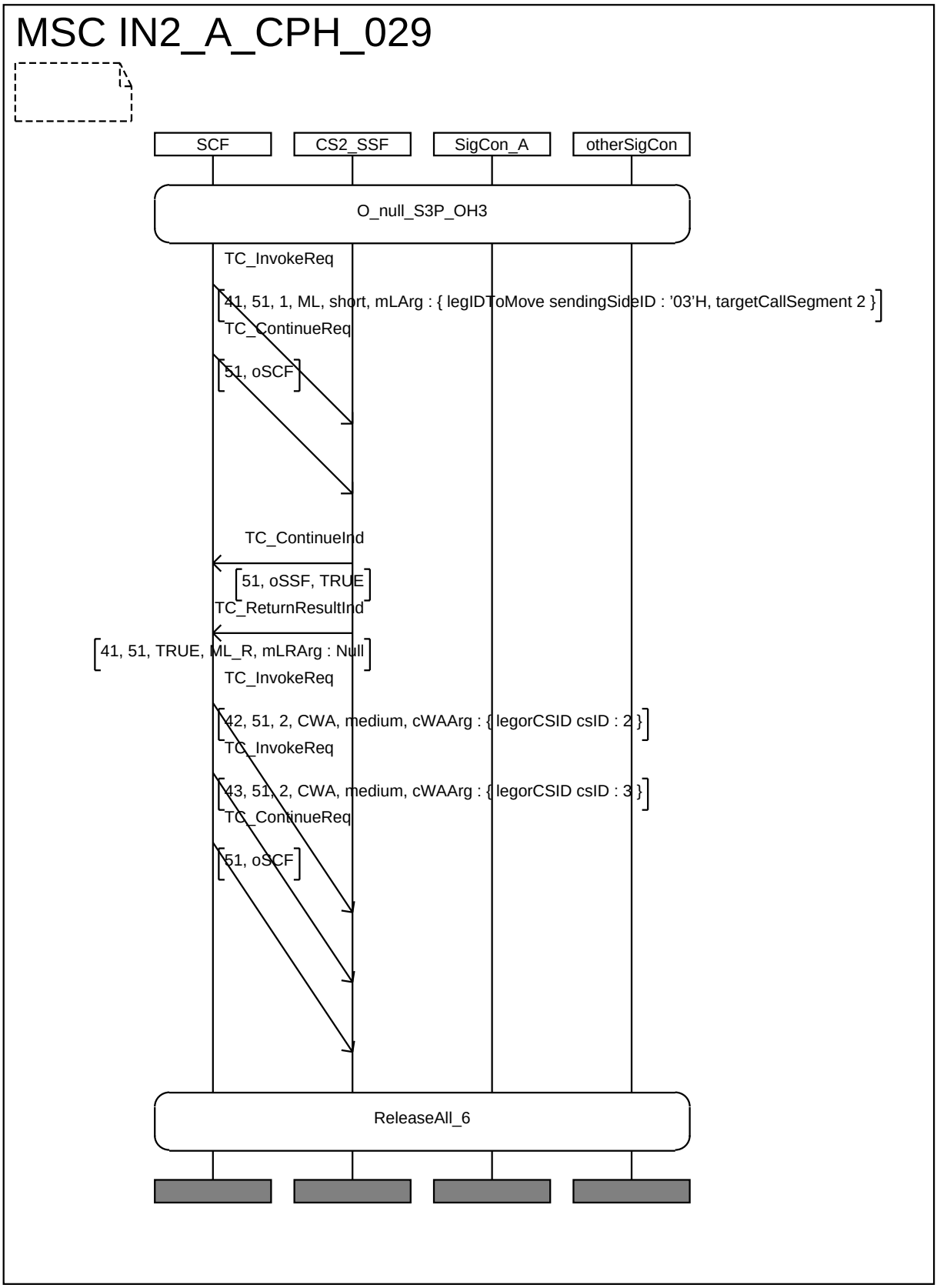
IN2_A_CPH_027	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(1)_OH(3)
Test description	L1! MoveLeg(1,3) L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=3) Reaching state O_OH(1)_OH(1)_S4P
Pass criteria	SSF sends MoveLegReturnResult
Postamble:	ReleaseAll_6


MSC IN2_A_CPH_027
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH1_OH3
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_6

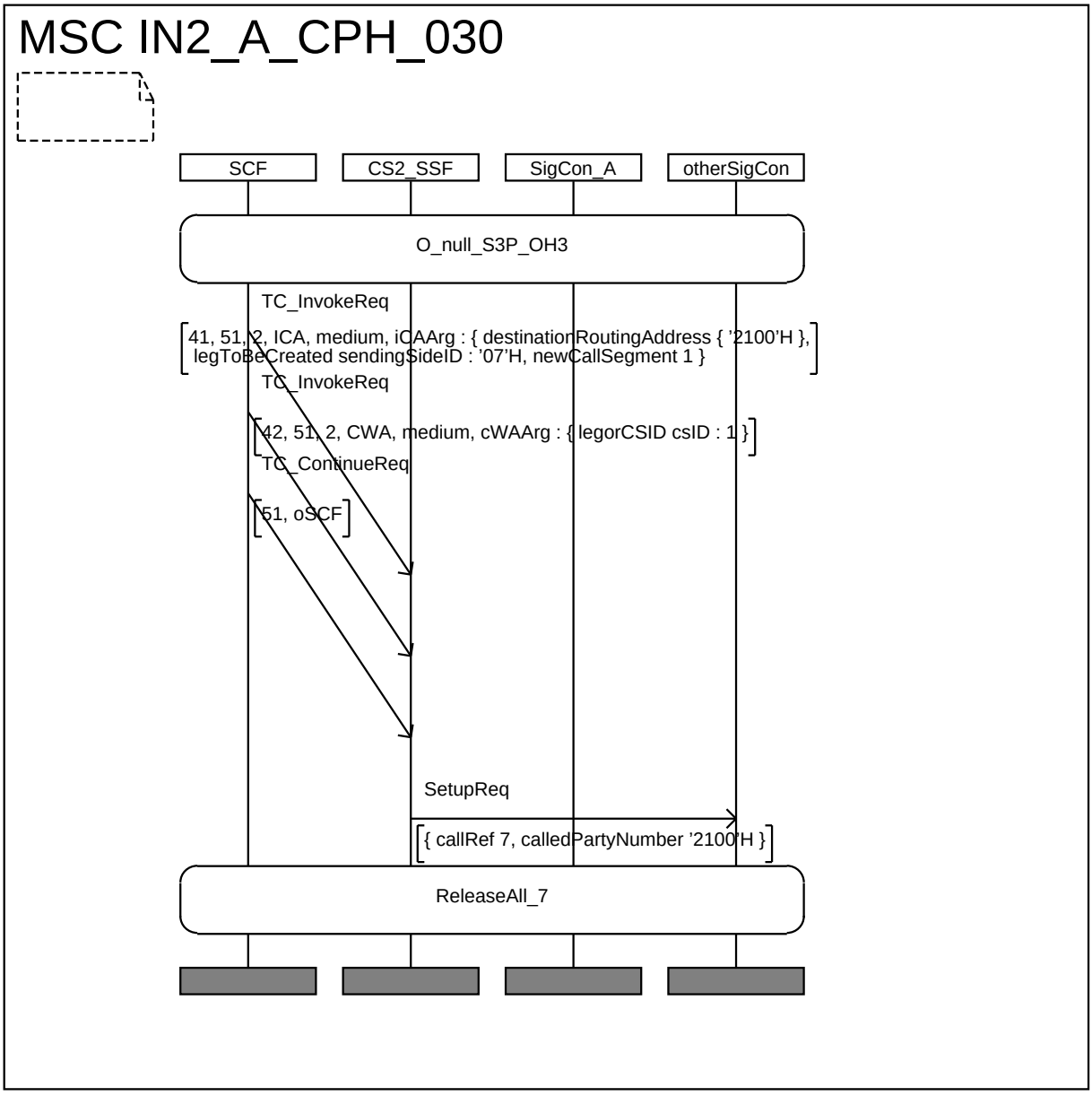
IN2_A_CPH_028	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S3P_OH(3)
Test description	CP1-5! SetupCon, E party answers CP1-6! SetupConf, F party answers L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CSId=2) L1! RequestReportBCSMEvent(7,oDisconnect)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


MSC IN2_A_CPH_028
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S3P_OH3
SetupConf
{ callRef 5 }
SetupConf
{ callRef 6 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '07'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2013'H }, callSegmentID 1, legToBeCreated sendingSideID : '07'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 7, calledPartyNumber '2013'H }
ReleaseAll_7

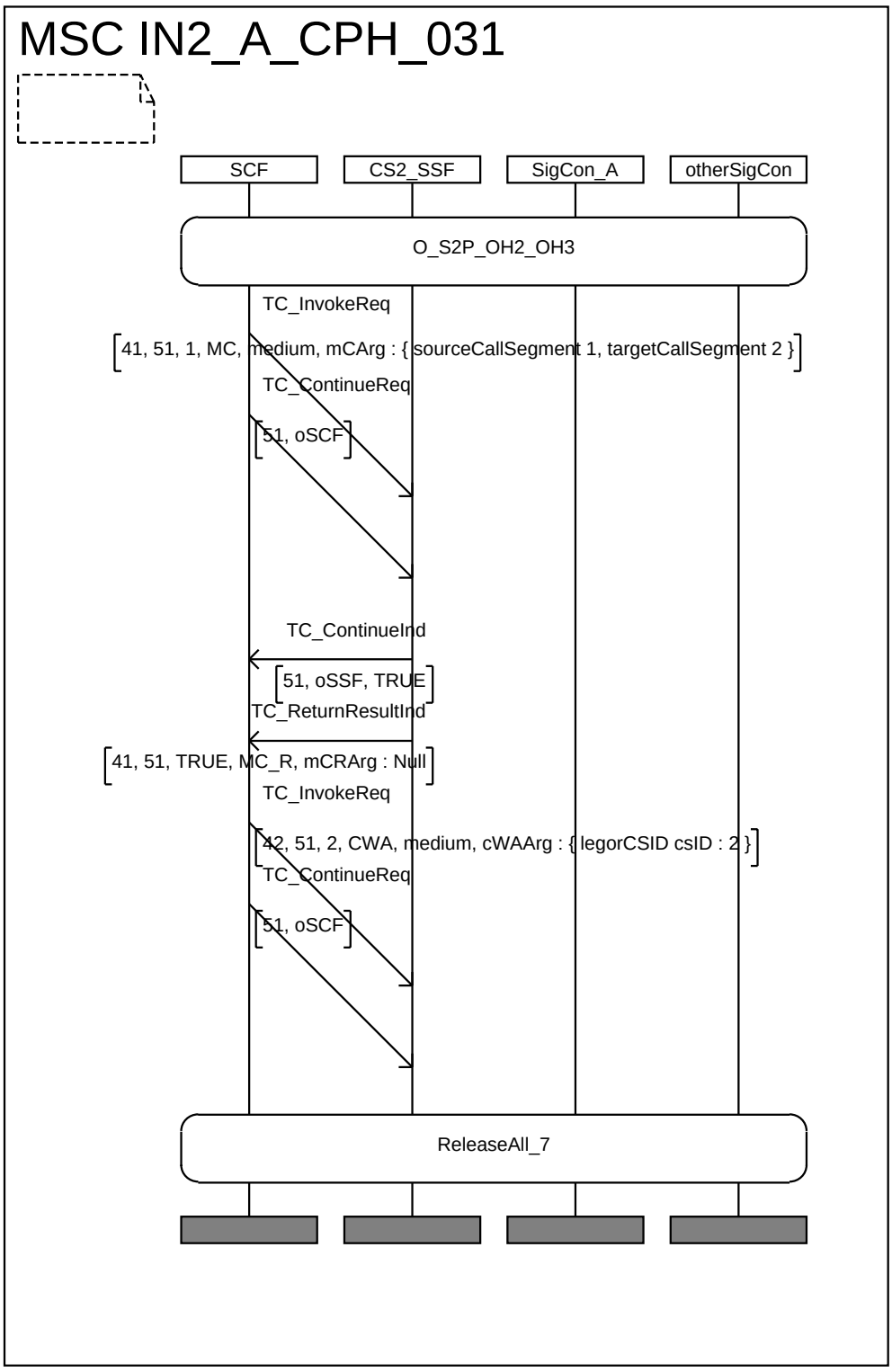
IN2_A_CPH_029	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S3P_OH(3)
Test description	L1! MoveLeg(3,2) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state O_null_S4P_OH(2)
Pass criteria	SSF sends MoveLegReturnResult
Postamble:	ReleaseAll_6


MSC IN2_A_CPH_029
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S3P_OH3
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '03'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_6

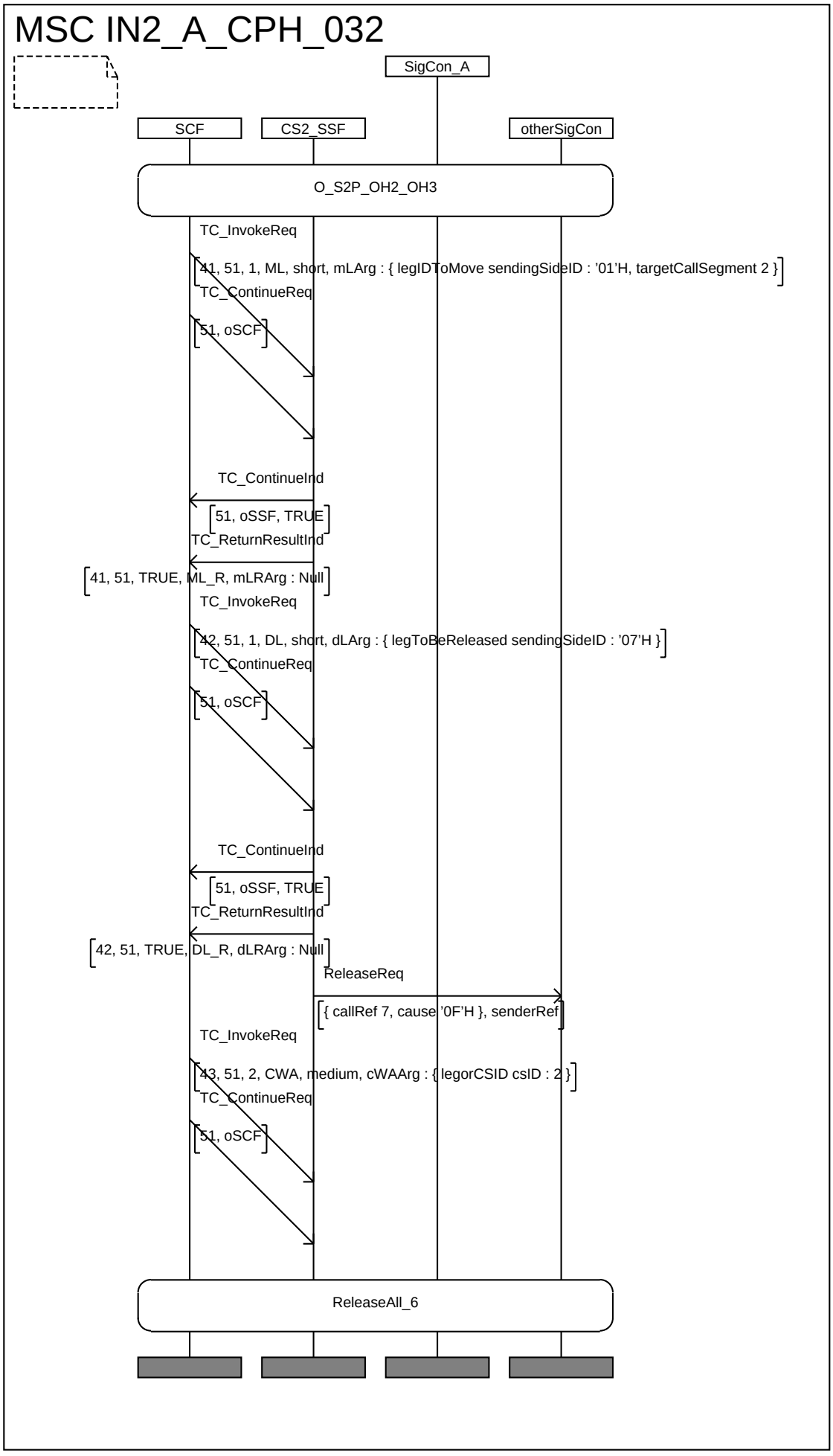
IN2_A_CPH_030	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S3P_OH(3)
Test description	L1! InitiateCallAttempt(7,1) L1! ContinueWithArgument(7) Reaching state O_S1P_S3P_OH(3)
Pass criteria	CP1_7? SetUpReq
Postamble:	ReleaseAll_7



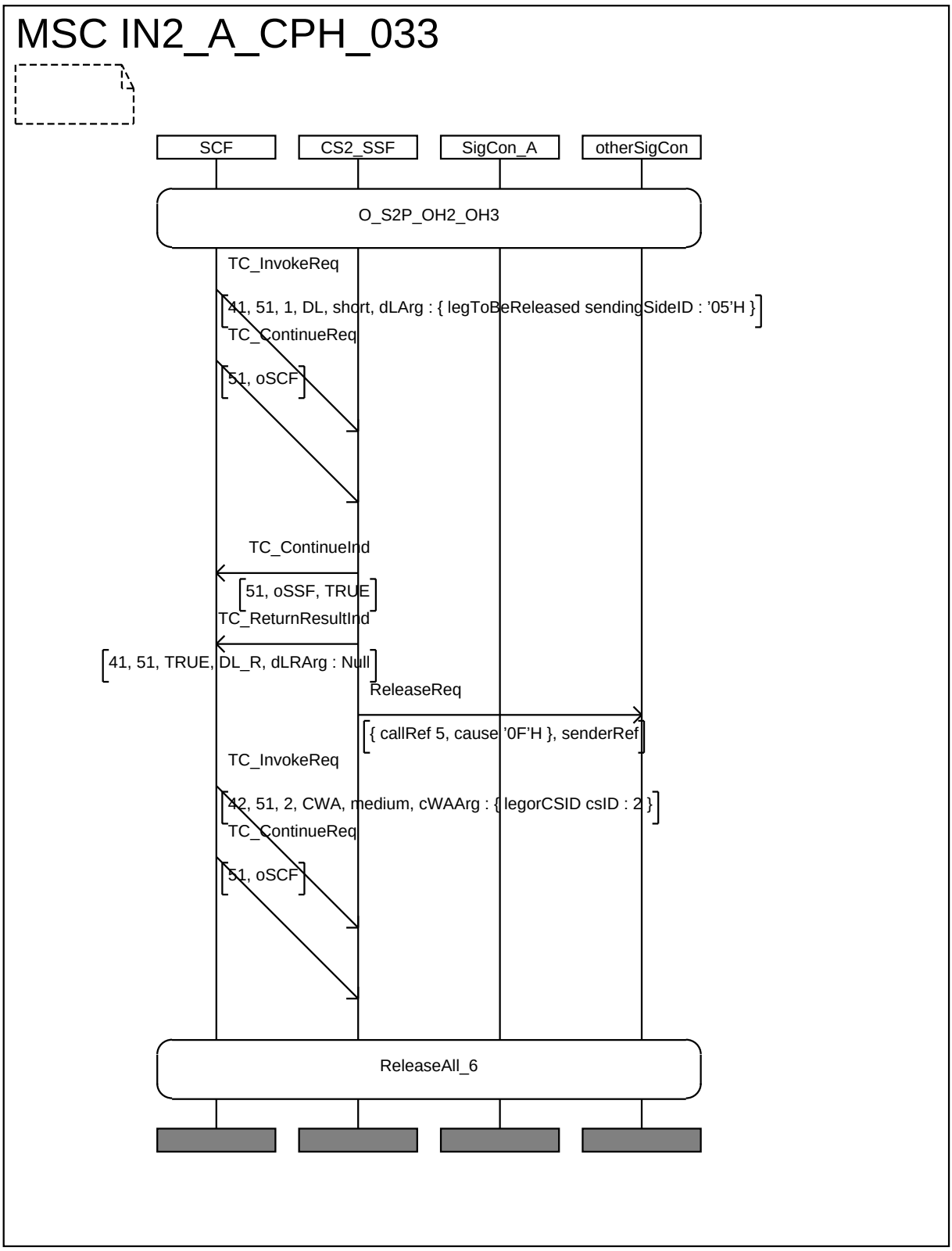
IN2_A_CPH_031	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(2)_OH(3)
Test description	L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2) Reaching state O_null_S4P_OH(3)
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble:	ReleaseAll_7


MSC IN2_A_CPH_031
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH2_OH3
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_7

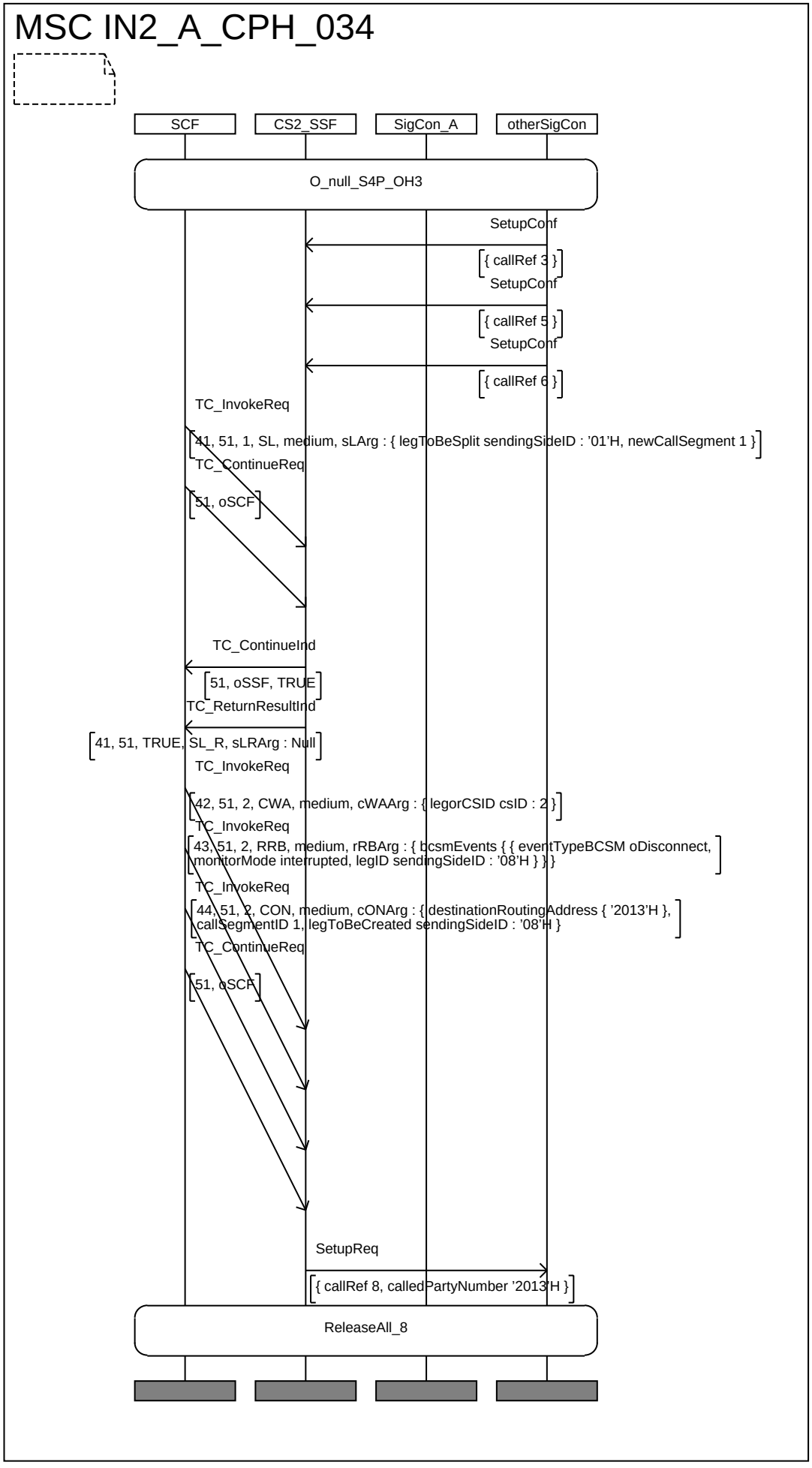
IN2_A_CPH_032	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(2)_OH(3)
Test description	L1! MoveLeg(1,2) L1? MoveLegReturnResult L1! DisconnectLeg(7) L1! ContinueWithArgument(CsId=2) L1? DisconnectLegReturnResult Reaching state O_null_S3P_OH(3)
Pass criteria	CP1_7?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


MSC IN2_A_CPH_032
SigCon_A
SCF
CS2_SSF
otherSigCon
O_S2P_OH2_OH3
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '07'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
42, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 7, cause '0F'H }, senderRef
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_6

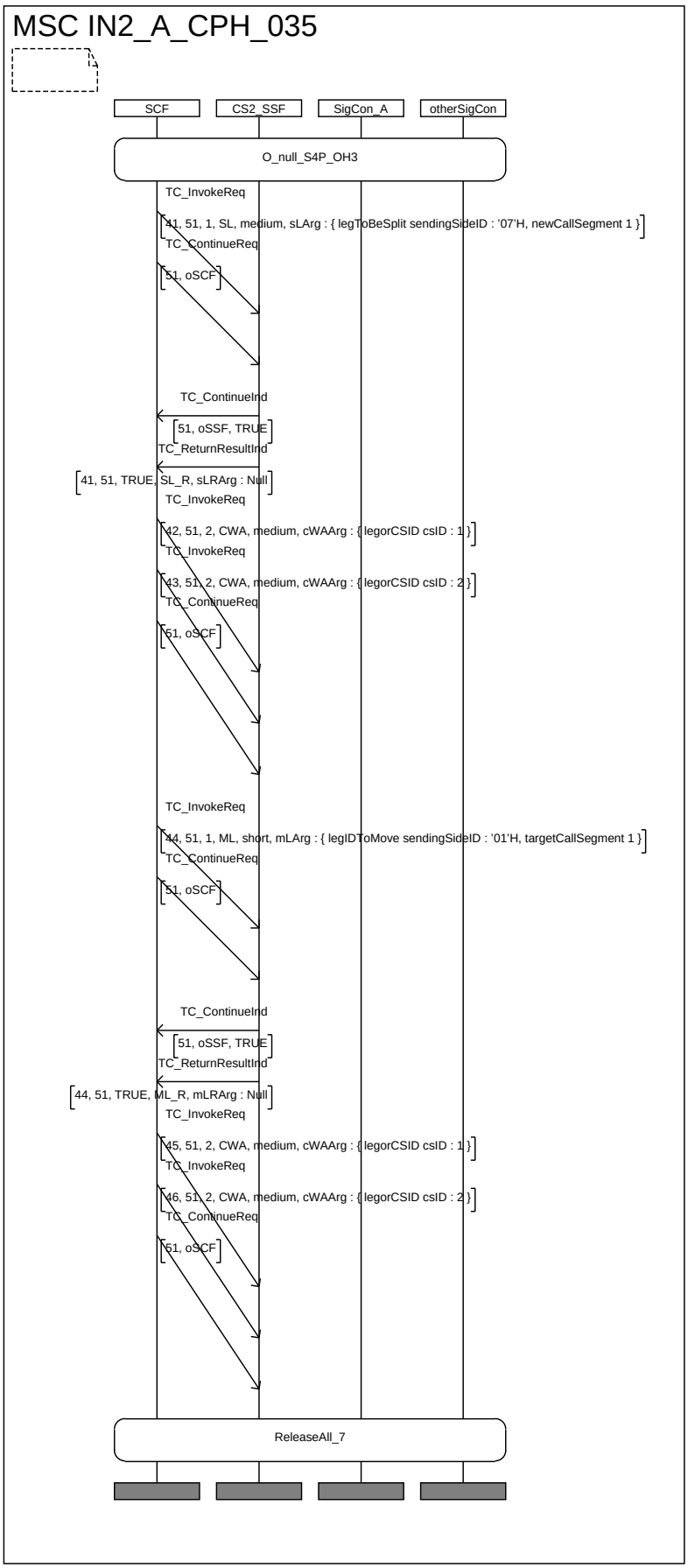
IN2_A_CPH_033	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(2)_OH(3)
Test description	L1! DisconnectLeg(5) L1! ContinueWithArgument(CsId=2) L1? DisconnectLegReturnResult Reaching state O_S2P_OH(1)_OH(3)
Pass criteria	CP1_5? ReleaseReq
Postamble:	ReleaseAll_6


MSC IN2_A_CPH_033
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH2_OH3
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '05'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 5, cause '0F'H }, senderRef
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_6

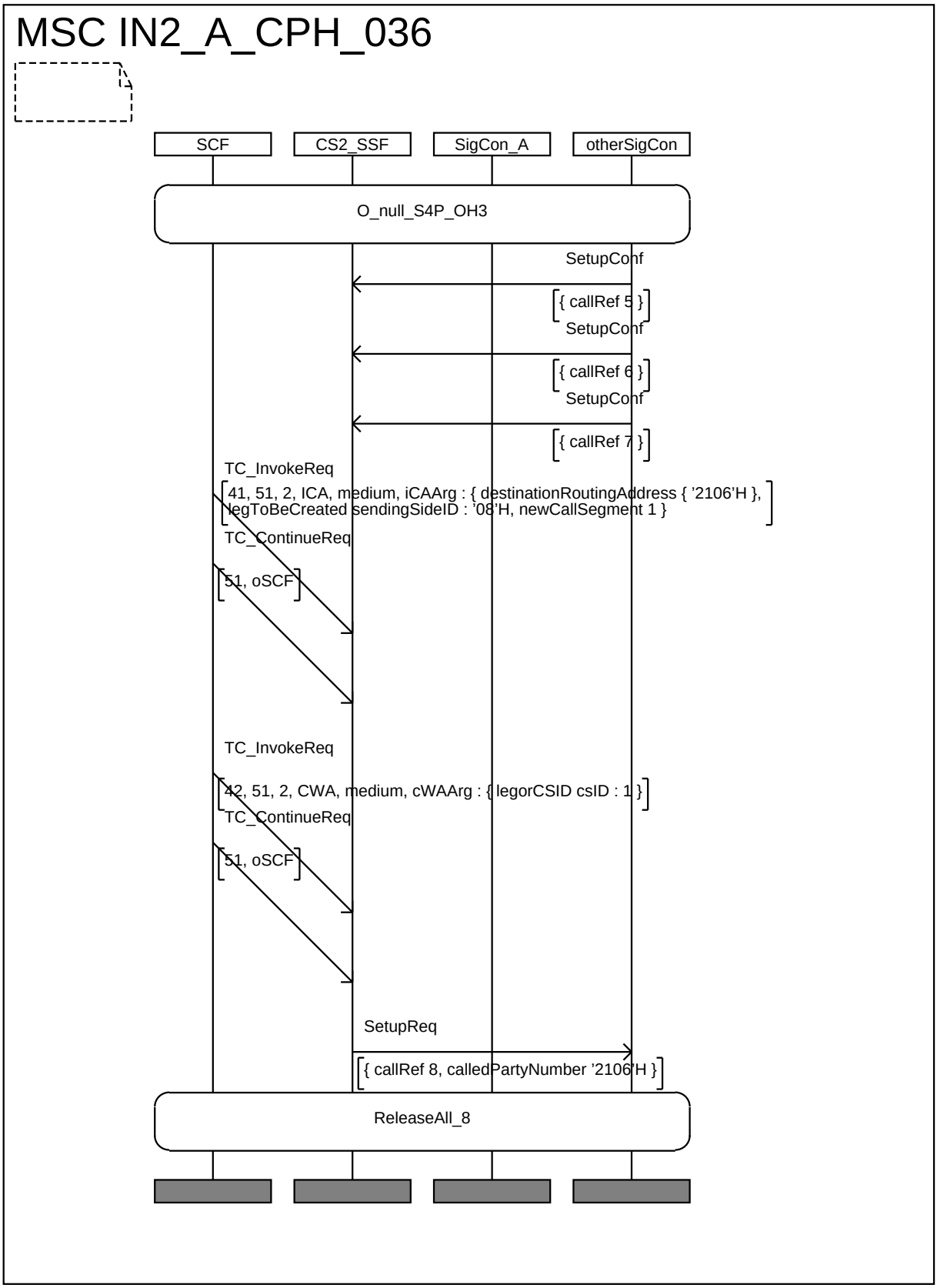
IN2_A_CPH_034	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S4P_OH(3)
Test description	CP1-3! SetupConf, C party answers CP1-5! SetupConf, E party answers CP1-6! SetupConf, F party answers L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1)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


MSC IN2_A_CPH_034
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S4P_OH3
SetupConf
{ callRef 3 }
SetupConf
{ callRef 5 }
SetupConf
{ callRef 6 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '08'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2013'H }, callSegmentID 1, legToBeCreated sendingSideID : '08'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 8, calledPartyNumber '2013'H }
ReleaseAll_8

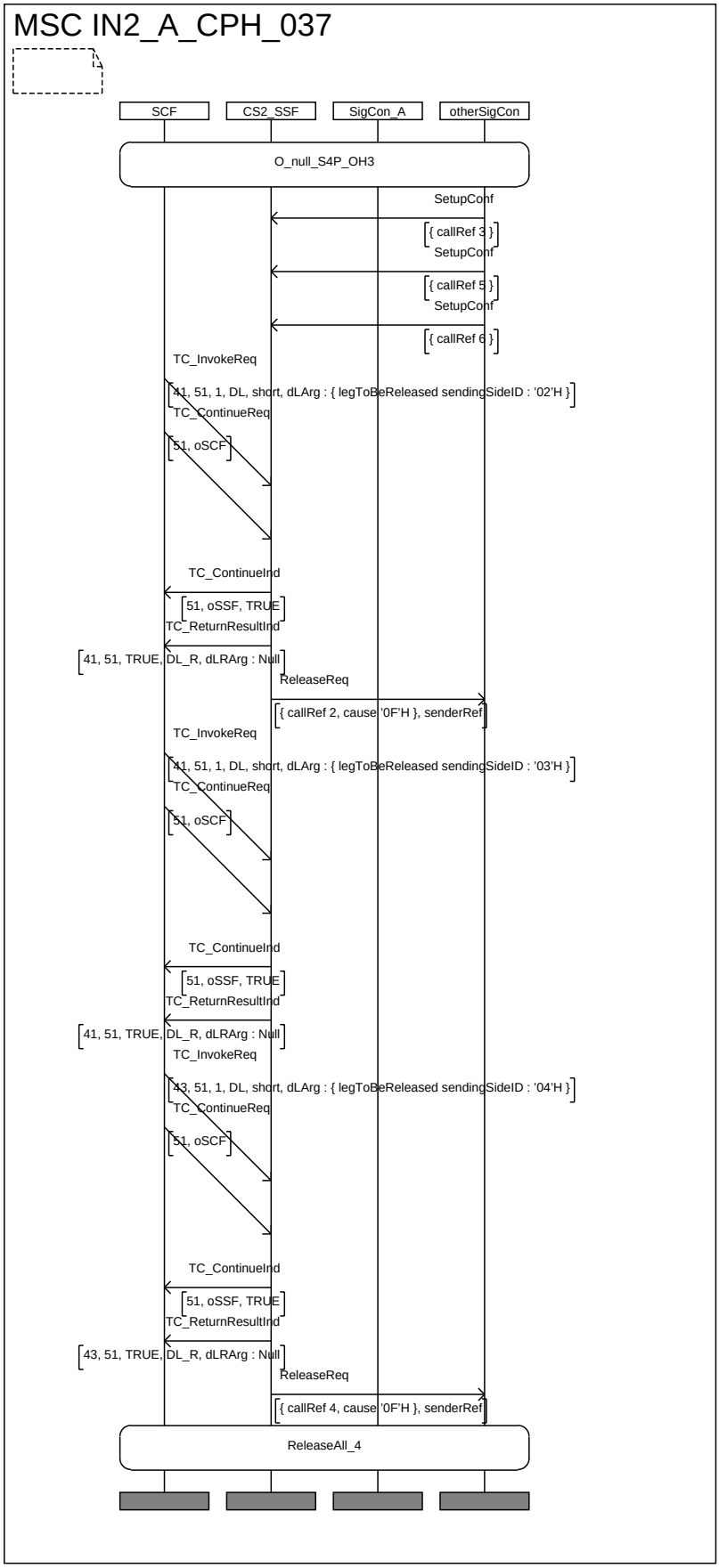
IN2_A_CPH_035	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S4P_OH(3)
Test description	L1! SplitLeg(7,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithAgument(CsId=2) L1! MoveLeg(1,1) L1! ContinueWithArgument(CsId=1) L1! ContinueWithAgument(CsId=2) Reaching state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


MSC IN2_A_CPH_035
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S4P_OH3
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '07'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
44, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
44, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
45, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
46, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_7

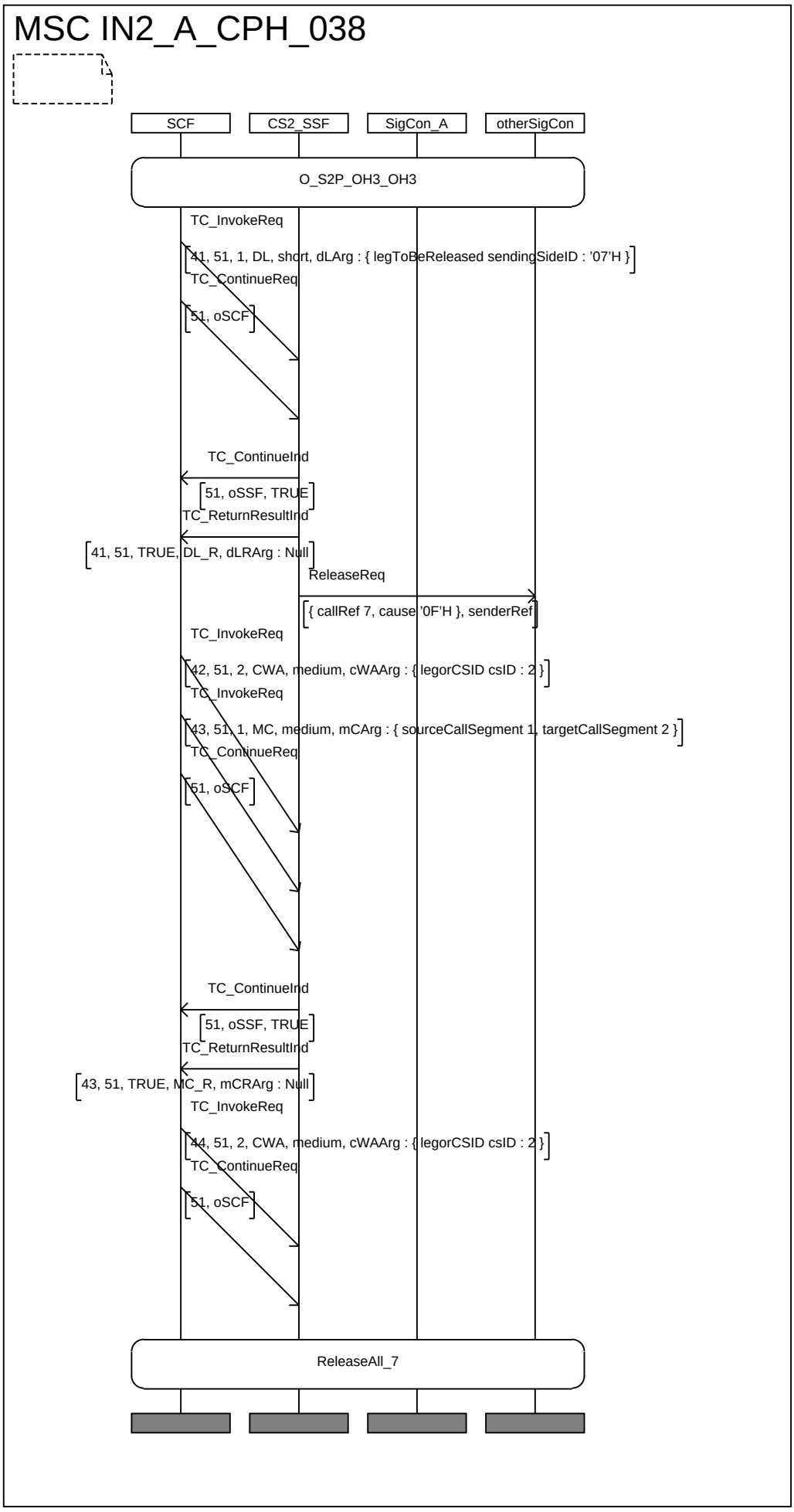
IN2_A_CPH_036	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S4P_OH(3)
Test description	CP1-3! SetupConf, C party answers CP1-5! SetupConf, E party answers CP1-6! SetupConf, F party answers L1! InitiateCallAttempt(8,1) L1! ContinueWithArgument(8) Reaching state O_S1P_S4P_OH(3)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


MSC IN2_A_CPH_036
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S4P_OH3
SetupConf
{ callRef 5 }
SetupConf
{ callRef 6 }
SetupConf
{ callRef 7 }
TC_InvokeReq
41, 51, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2106'H },
legToBeCreated sendingSideID : '08'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 8, calledPartyNumber '2106'H }
ReleaseAll_8

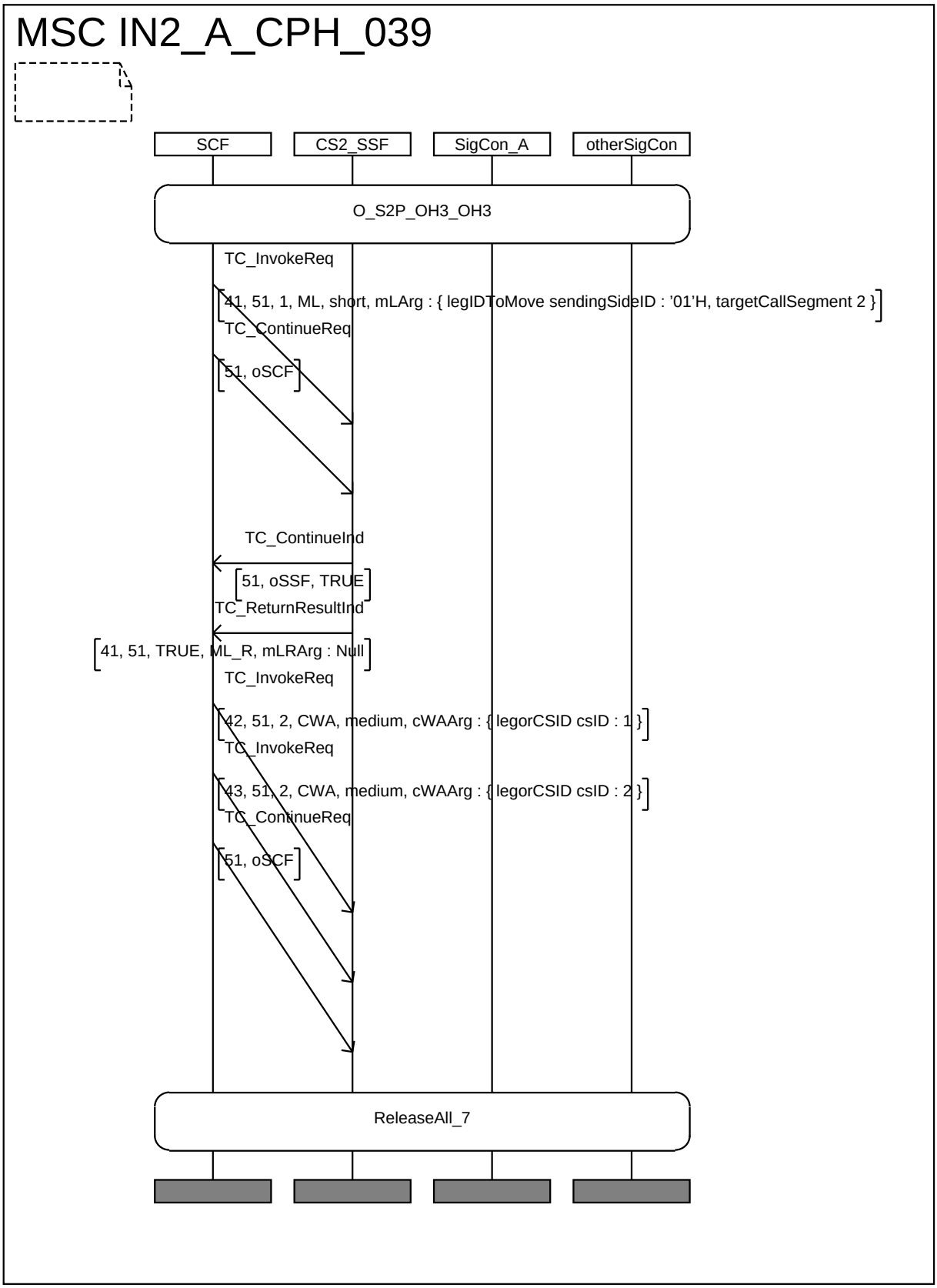
IN2_A_CPH_037	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_null_S4P_OH(3)
Test description	CP1-3! SetupConf, C party answers CP1-5! SetupConf, E party answers CP1-6! SetupConf, F party answers L1! DisconnectLeg(2) L1! DisconnectLeg(3) L1! DisconnectLeg(4) L1? DisconnectLegReturnResult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


MSC IN2_A_CPH_037
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_S4P_OH3
SetupConf
{ callRef 3 }
SetupConf
{ callRef 5 }
SetupConf
{ callRef 6 }
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 2, cause '0F'H }, senderRef
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
TC_InvokeReq
43, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
43, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 4, cause '0F'H }, senderRef
ReleaseAll_4

IN2_A_CPH_038	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_OH(3)_OH(3)
Test description	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)
Pass criteria	CP1_7? ReleaseReq
Postamble:	ReleaseAll_7

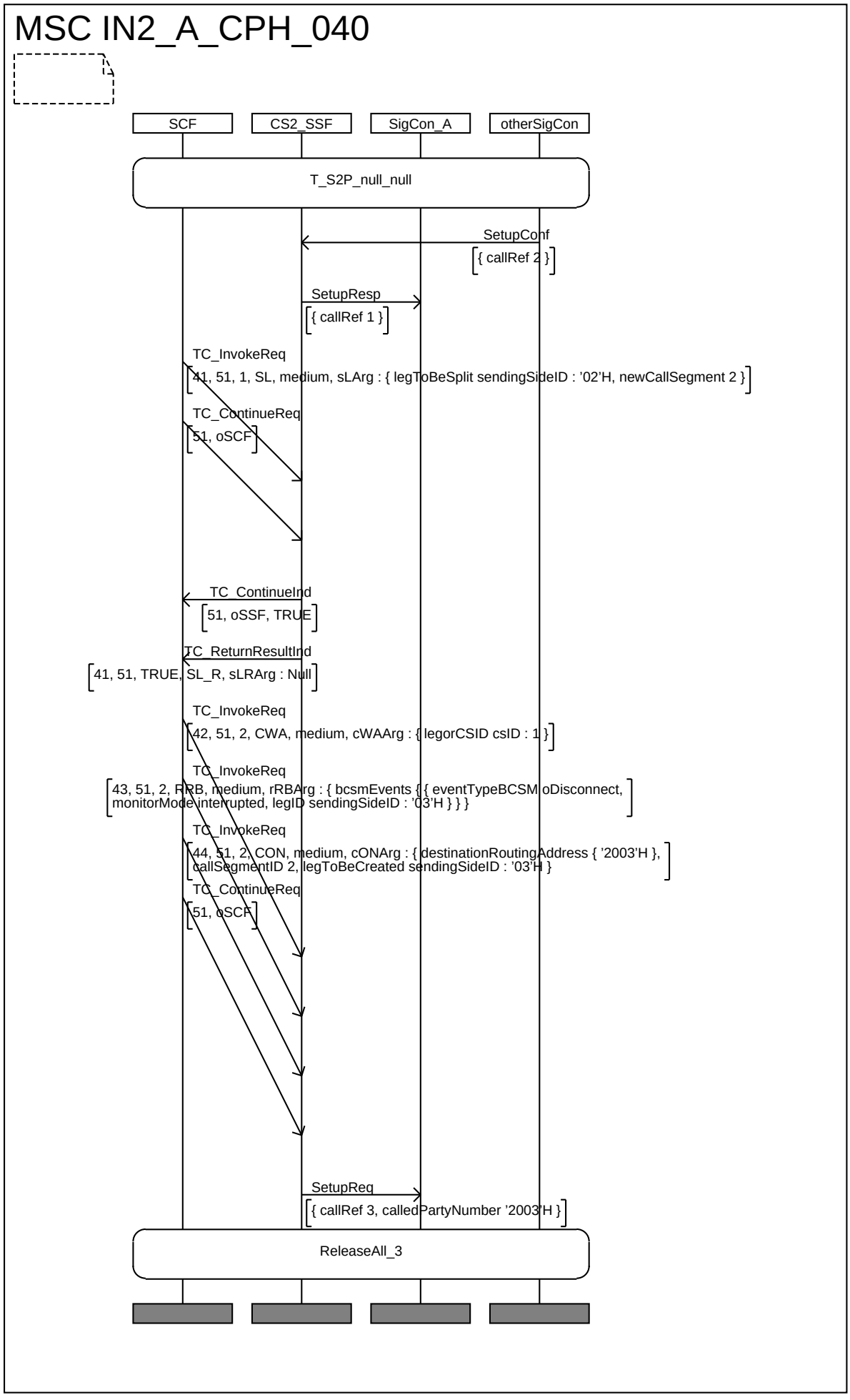

MSC IN2_A_CPH_038
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH3_OH3
TC_InvokeReq
[41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '07'H }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[41, 51, TRUE, DL_R, dLRArg : Null]
ReleaseReq
[{ callRef 7, cause '0F'H }, senderRef]
TC_InvokeReq
[42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_InvokeReq
[43, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[43, 51, TRUE, MC_R, mCRArg : Null]
TC_InvokeReq
[44, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_ContinueReq
[51, oSCF]
ReleaseAll_7

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

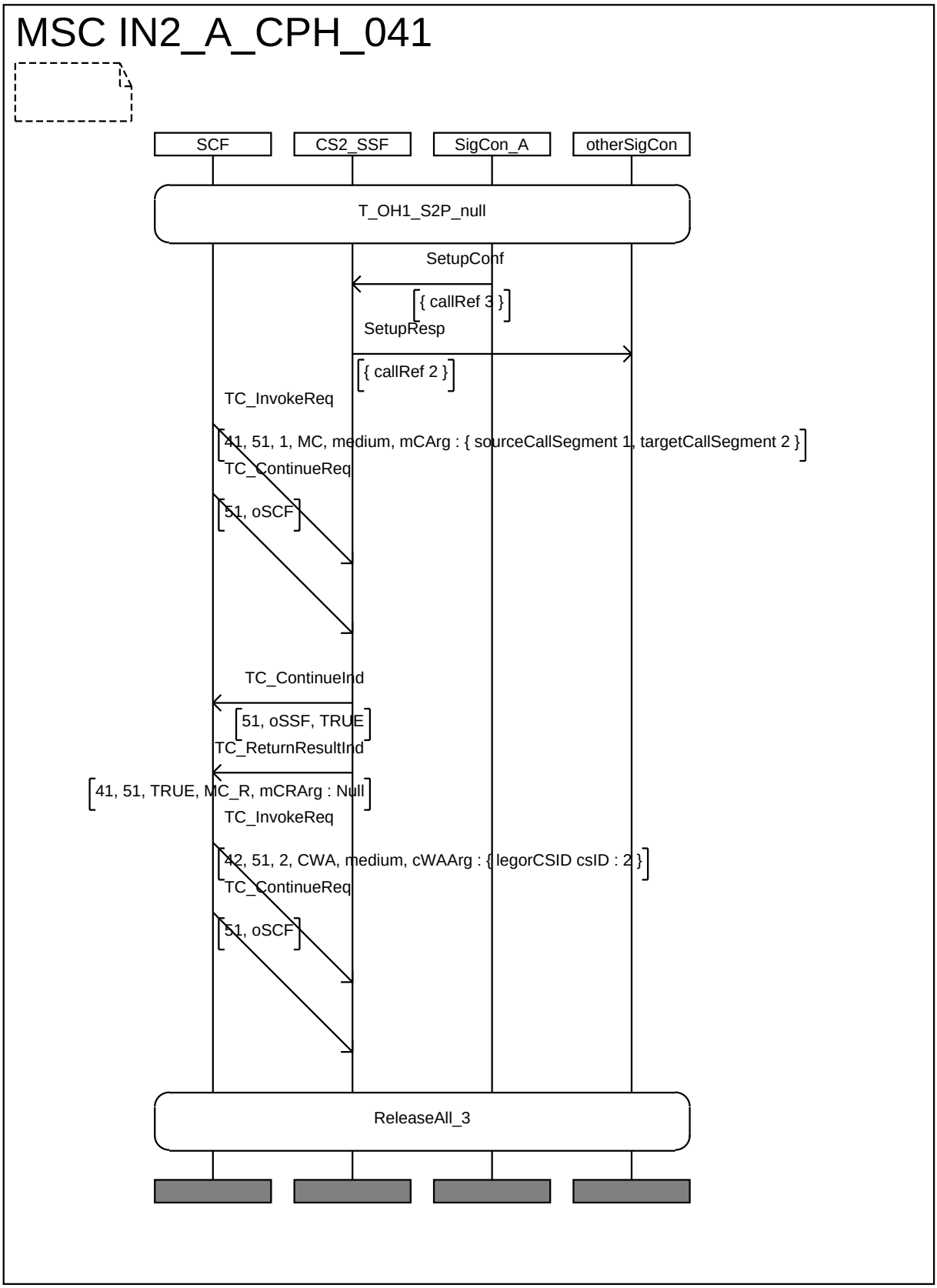

MSC IN2_A_CPH_039
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_OH3_OH3
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_7

7.5.2 Terminating (T_BCSM) trigger (controlling legId = 2)

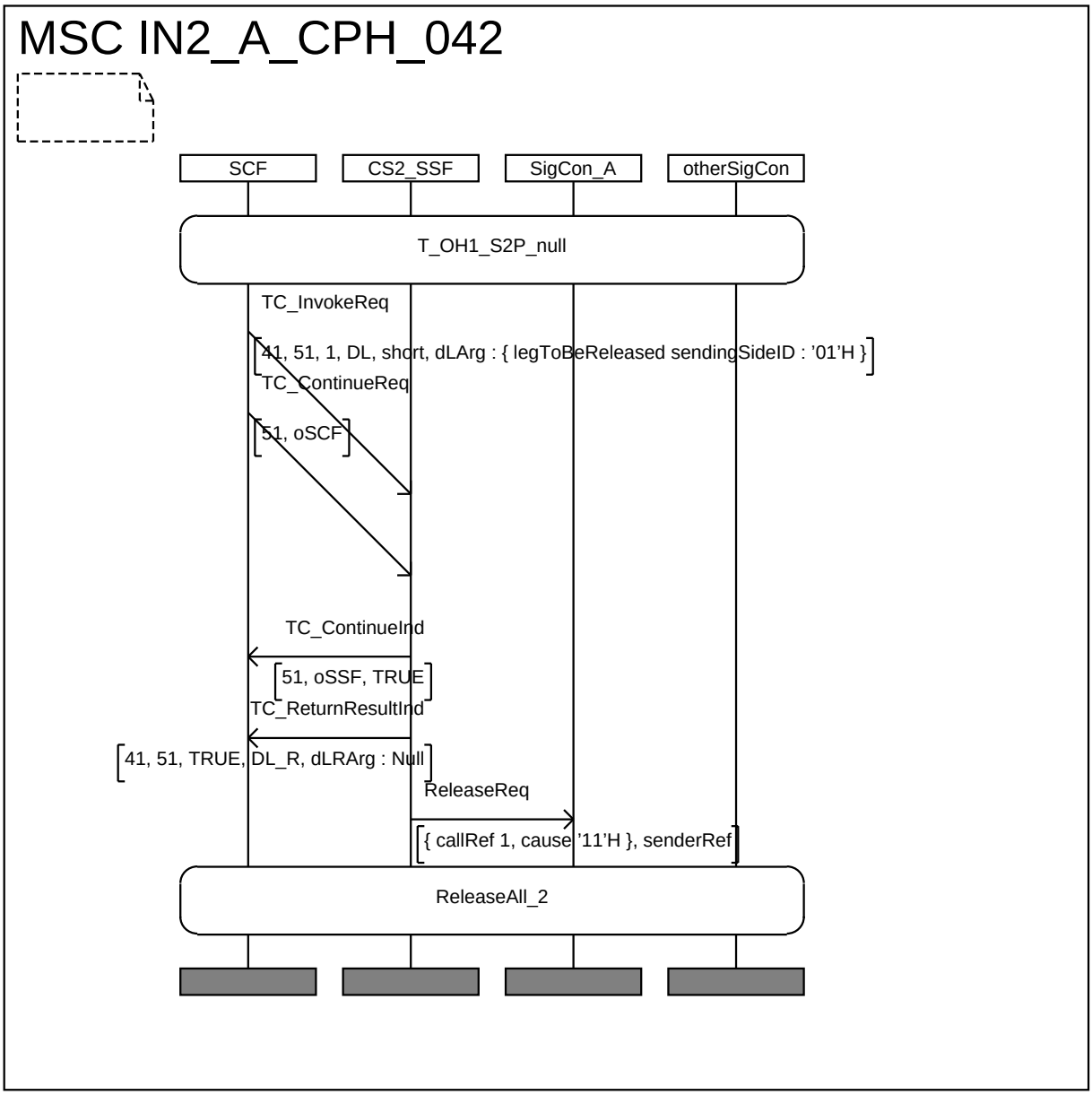
IN2_A_CPH_040	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null
Test description	CP1-2! SetupConf L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CSId=1) L1! RequestReportBCSMEvent(3,oDisconnect) L1! Connect(3,2) Reaching state T_OH(1)_S2P_null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


MSC IN2_A_CPH_040
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect,
monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H },
callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseAll_3

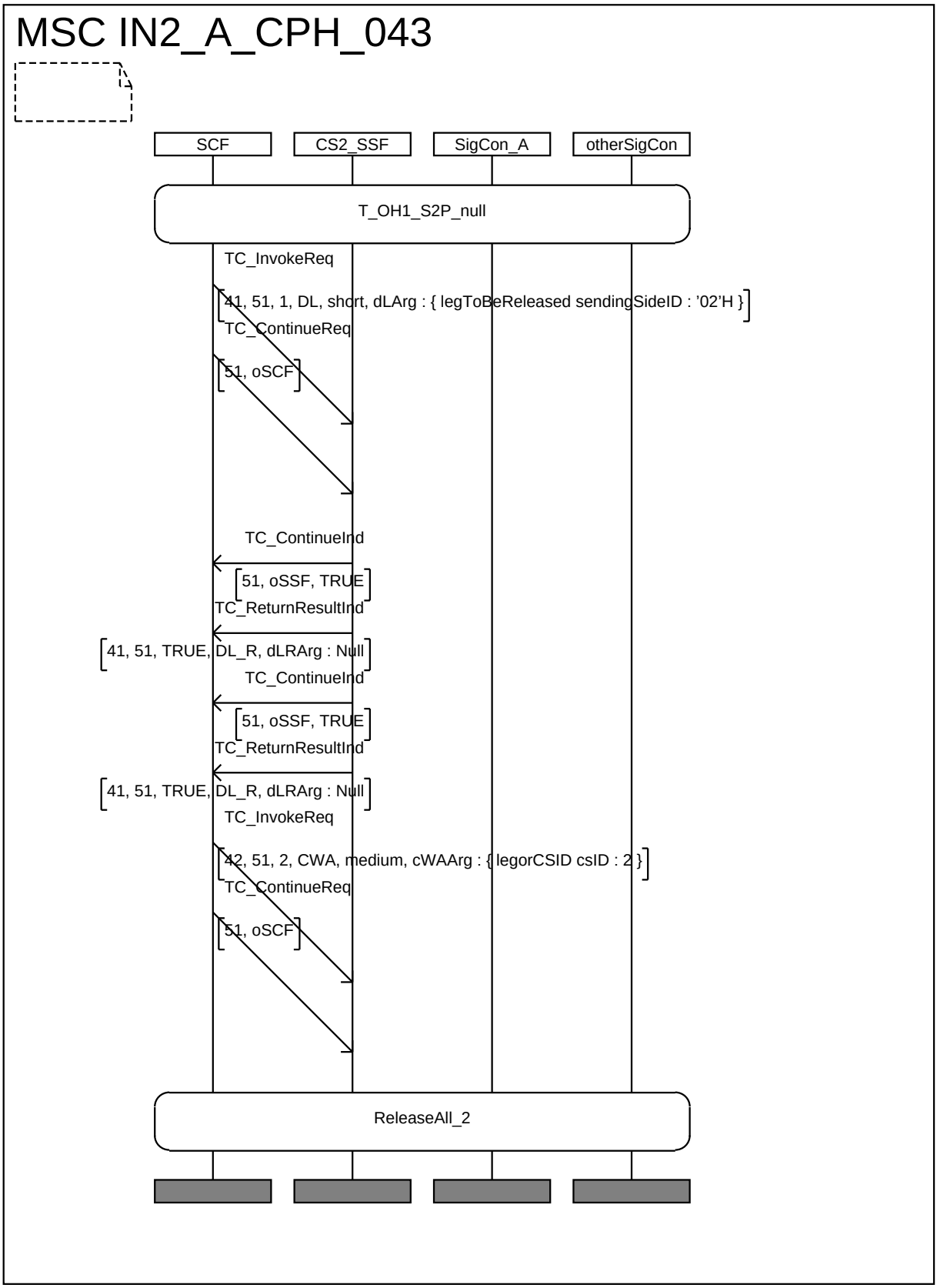
IN2_A_CPH_041	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_OH(1)_S2P_null
Test description	CP1-3! SetupConf, C party answer L1! MergeCallSegment(1,2) L1! ContinueWithArgument(CsId=2 Reaching state T_null_S3P_null
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble: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


MSC IN2_A_CPH_041
SCF
CS2_SSF
SigCon_A
otherSigCon
T_OH1_S2P_null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 2 }
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_3

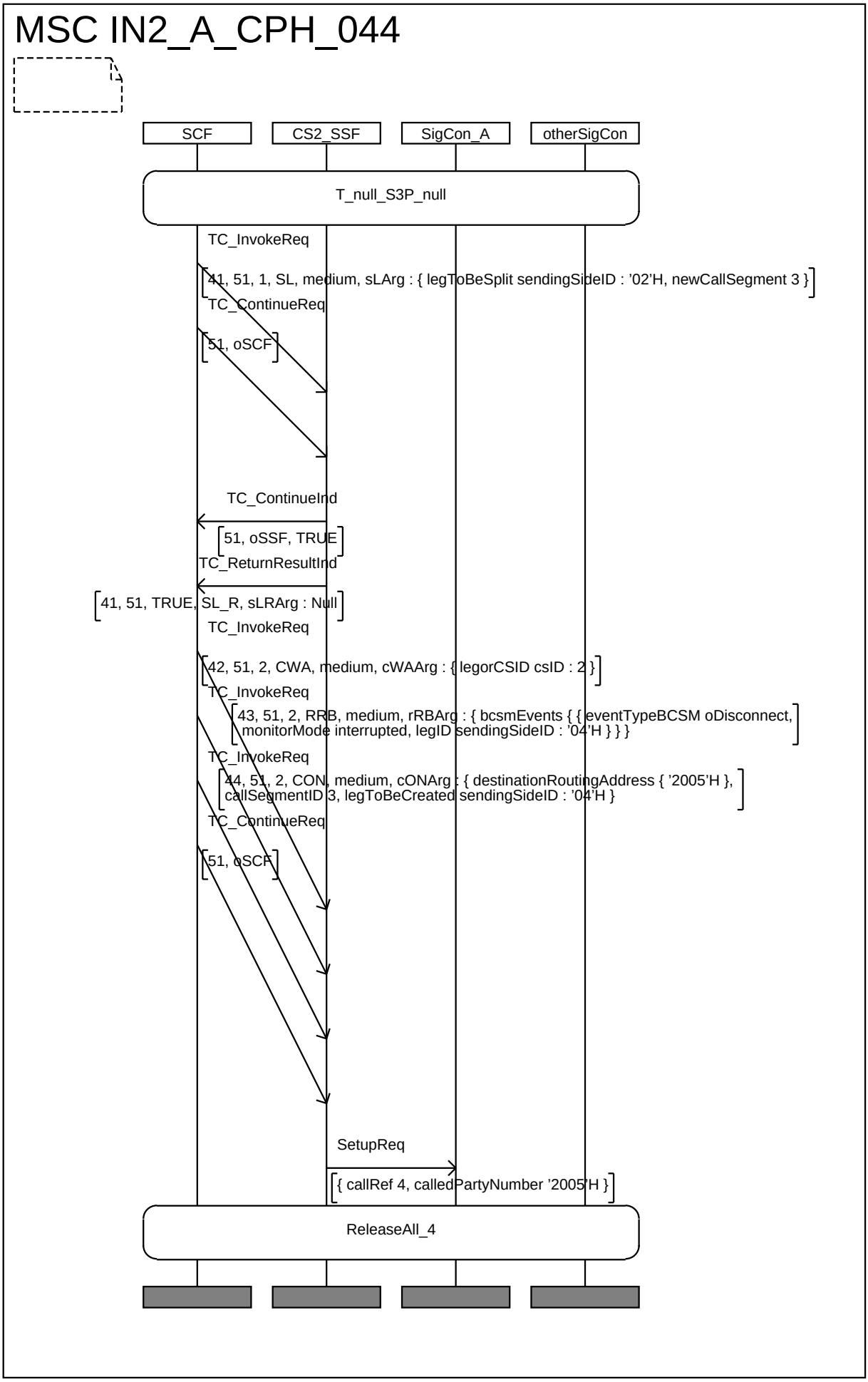
IN2_A_CPH_042	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_OH(1)_S2P_null
Test description	L1! DisconnectLeg(1) L1? DisconnectLegReturnResult Reaching state T_null_S2P_null
Pass criteria	CP1_1? ReleaseReq
Postamble:	ReleaseAll_2



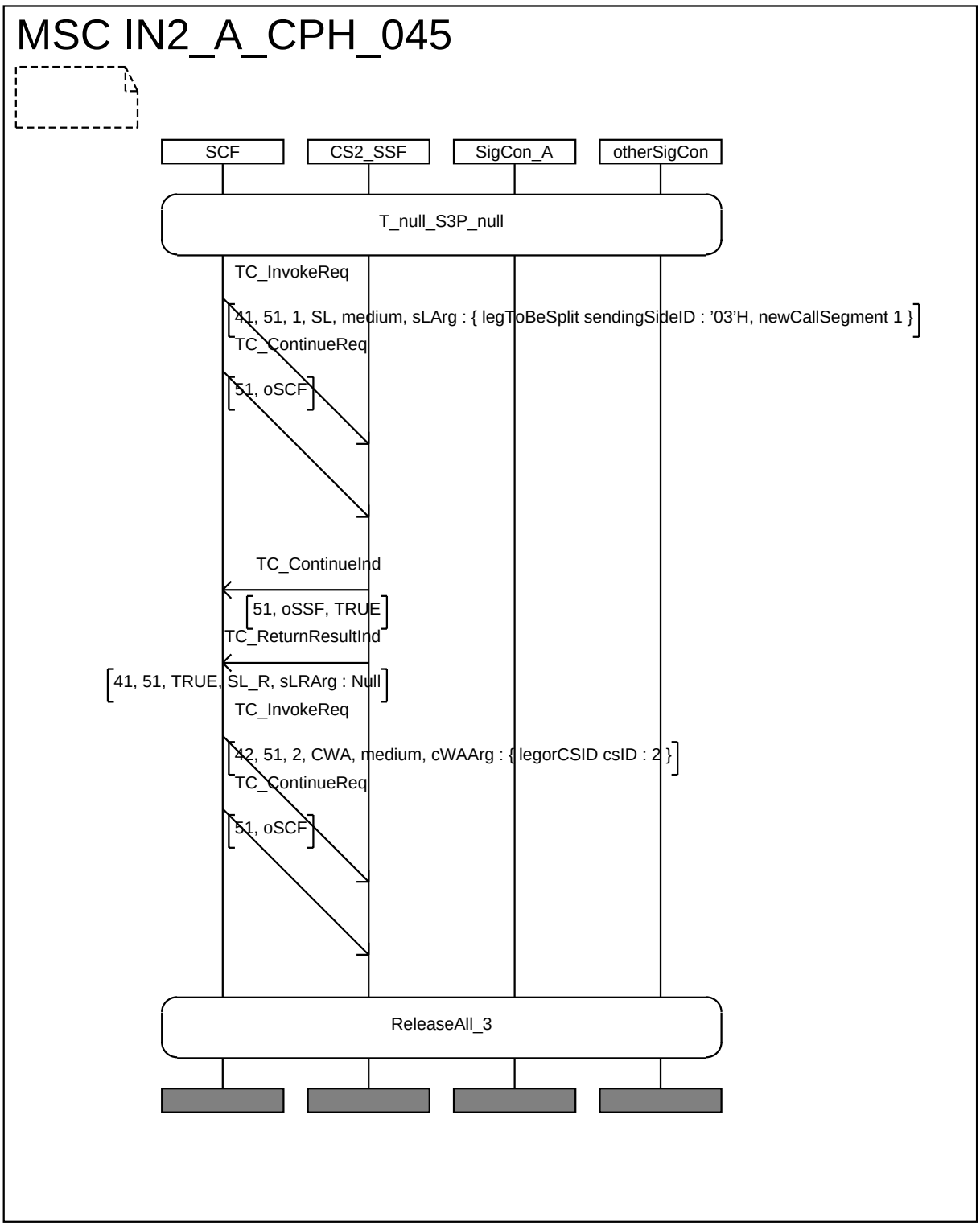
IN2_A_CPH_043	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_OH(1)_S2P_null
Test description	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult L1! ContinueWithArgument(CsId=2) Reaching state T_OH(1)_OH(1)_null
Pass criteria	CP1_2? ReleaseReq
Postamble:	ReleaseAll_2


MSC IN2_A_CPH_043
SCF
CS2_SSF
SigCon_A
otherSigCon
T_OH1_S2P_null
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '02'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_2

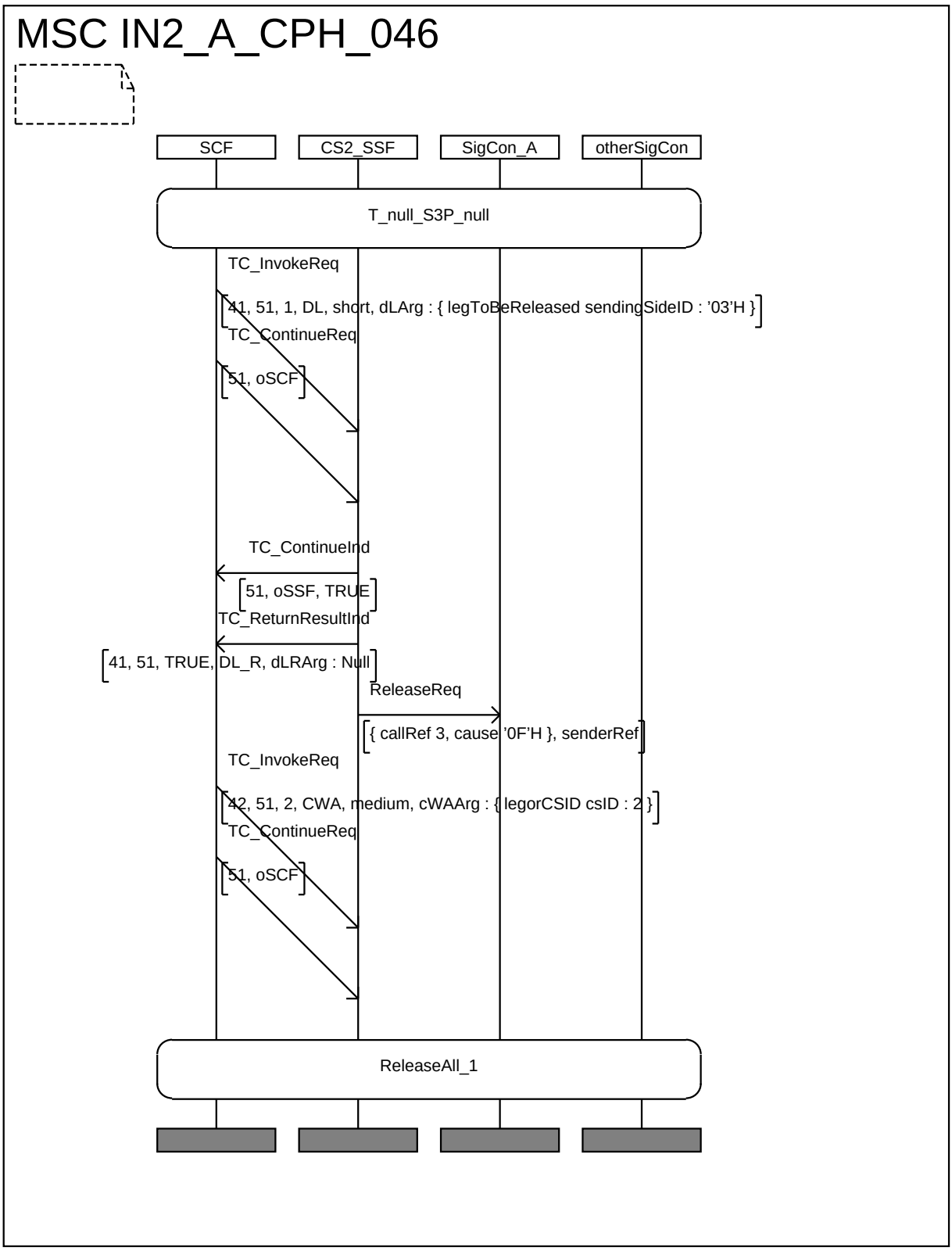
IN2_A_CPH_044	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_S3P_null
Test description	L1! SplitLeg(2,3) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=2) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(4,3) Reaching state T_null_OH(2)_S2P
Pass criteria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


MSC IN2_A_CPH_044
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_S3P_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '04'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2005'H }, callSegmentID 3, legToBeCreated sendingSideID : '04'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2005'H }
ReleaseAll_4

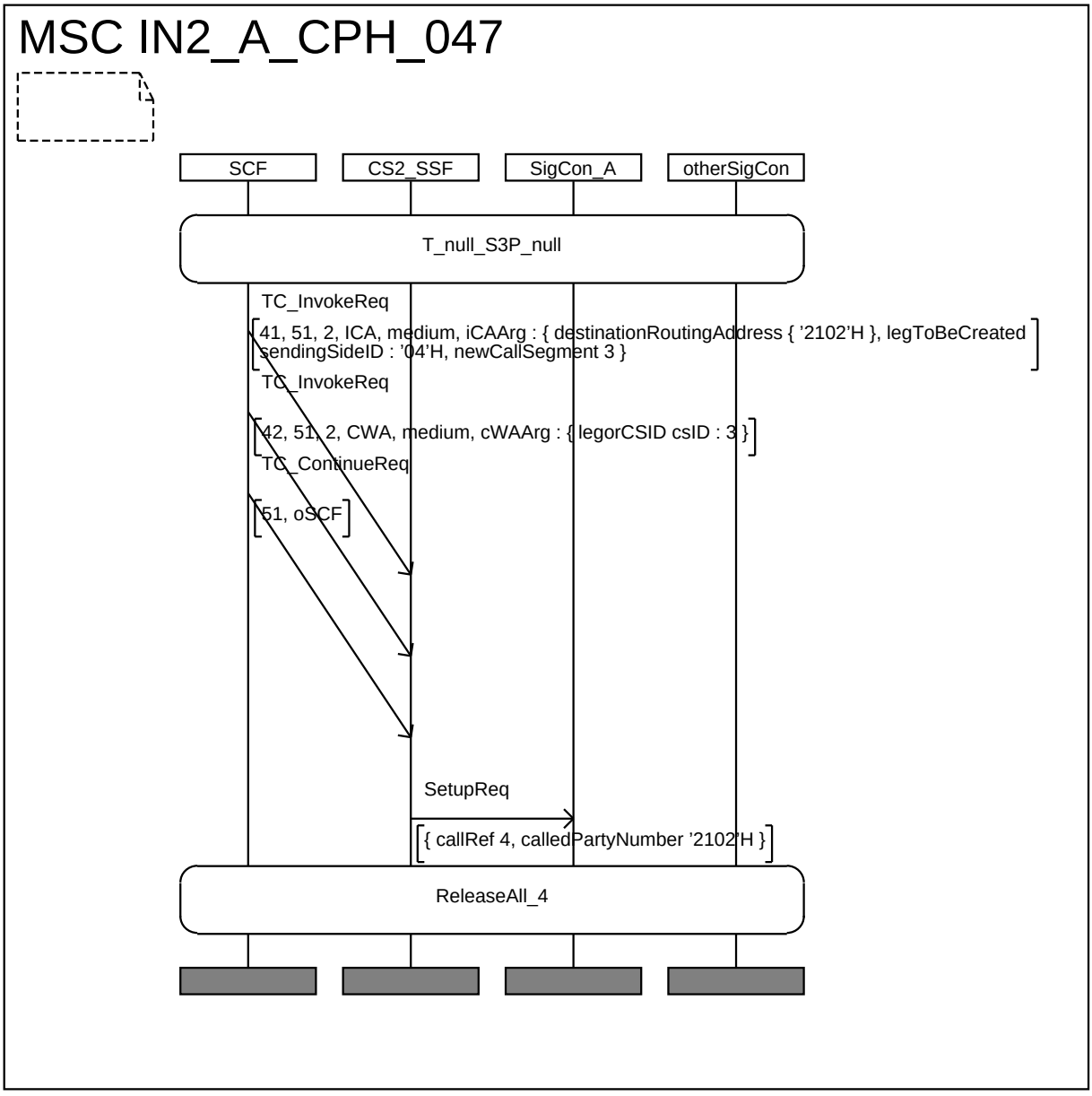
IN2_A_CPH_045	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_S3P_null
Test description	L1! SplitLeg(3,1) L1! ContinueWithArgument(CsId=2) Reaching state T_OH(1)_S2P_null
Pass criteria	SSF sends SplitLegReturnResult
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_045
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_S3P_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '03'H, newCallSegment 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_3

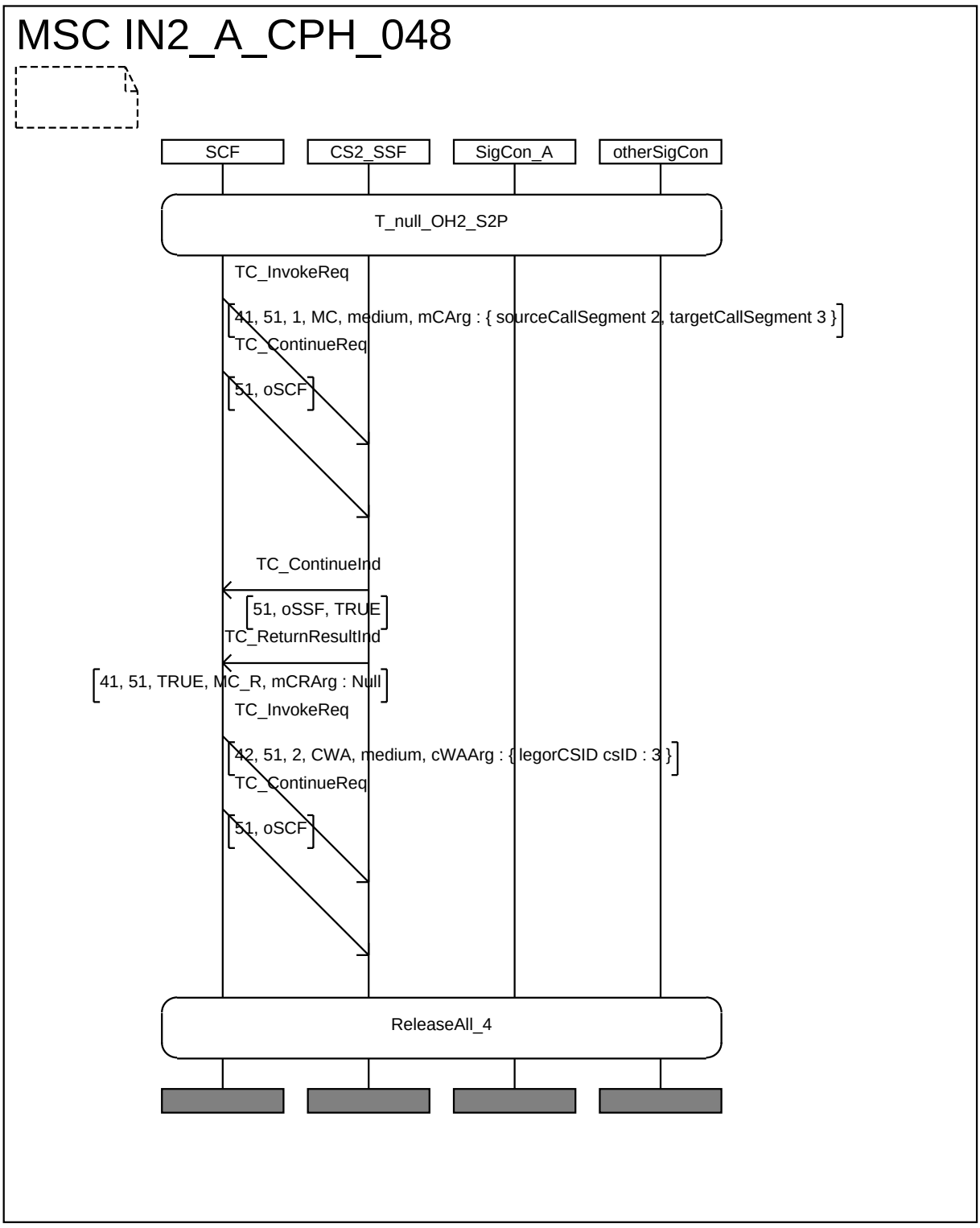
IN2_A_CPH_046	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_S3P_null
Test description	L1! DisconnectLeg(3) L1! ContinueWithArgument(CsId=2) L1?DisconnectLegReturnResult Reaching state T_null_S2P_null
Pass criteria	CP1_3? ReleaseReq
Postamble:	ReleaseAll_1


MSC IN2_A_CPH_046
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_S3P_null
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 3, cause '0F'H }, senderRef
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
ReleaseAll_1

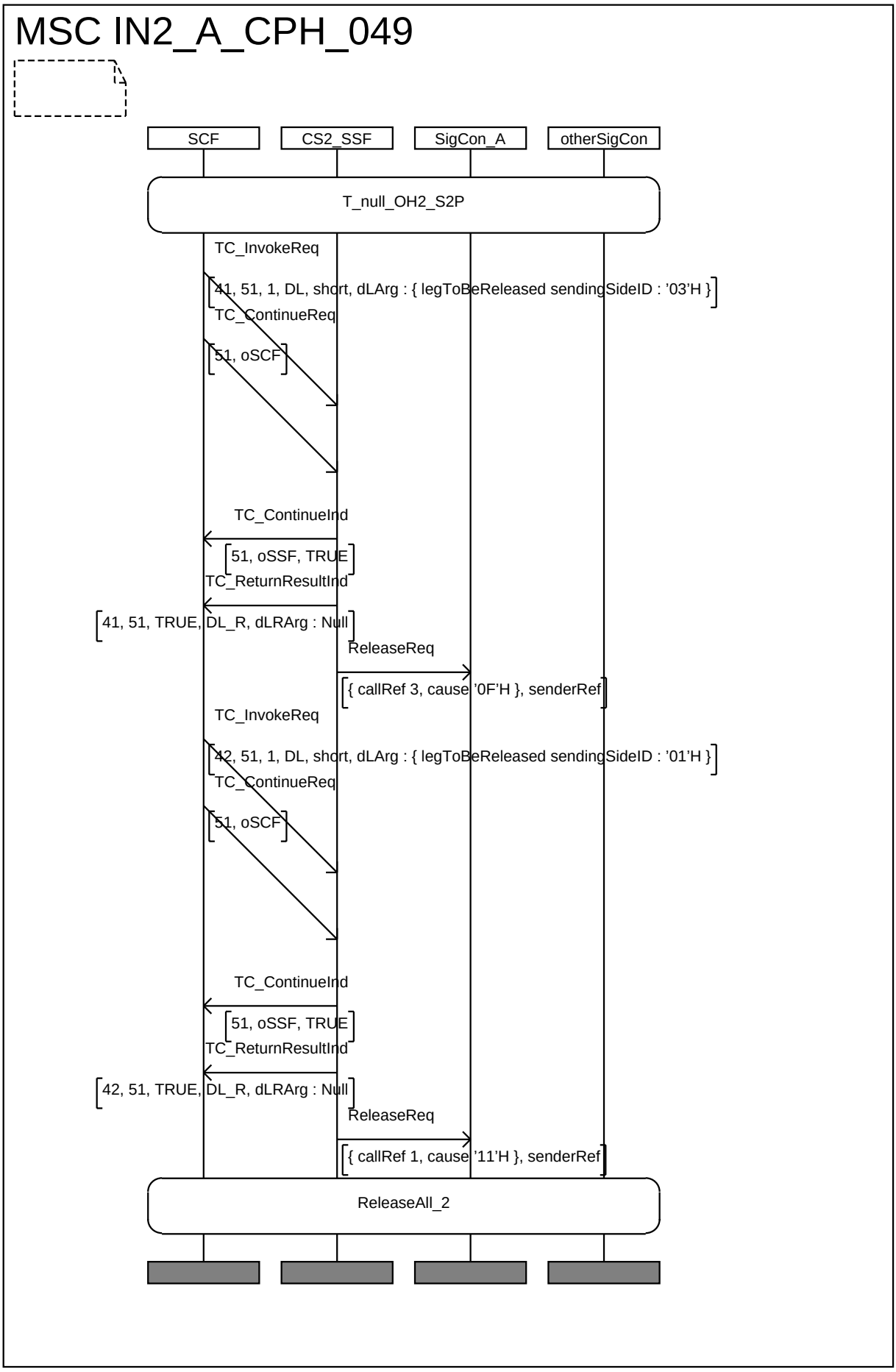
IN2_A_CPH_047	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_S3P_null
Test description	L1! InitiateCallAttempt(4,3) L1! ContinueWithArgument(CsId=3) Reaching state T_null_S3P_S1P
Pass criteria	CP1_4? SetUpReq
Postamble:	ReleaseAll_4



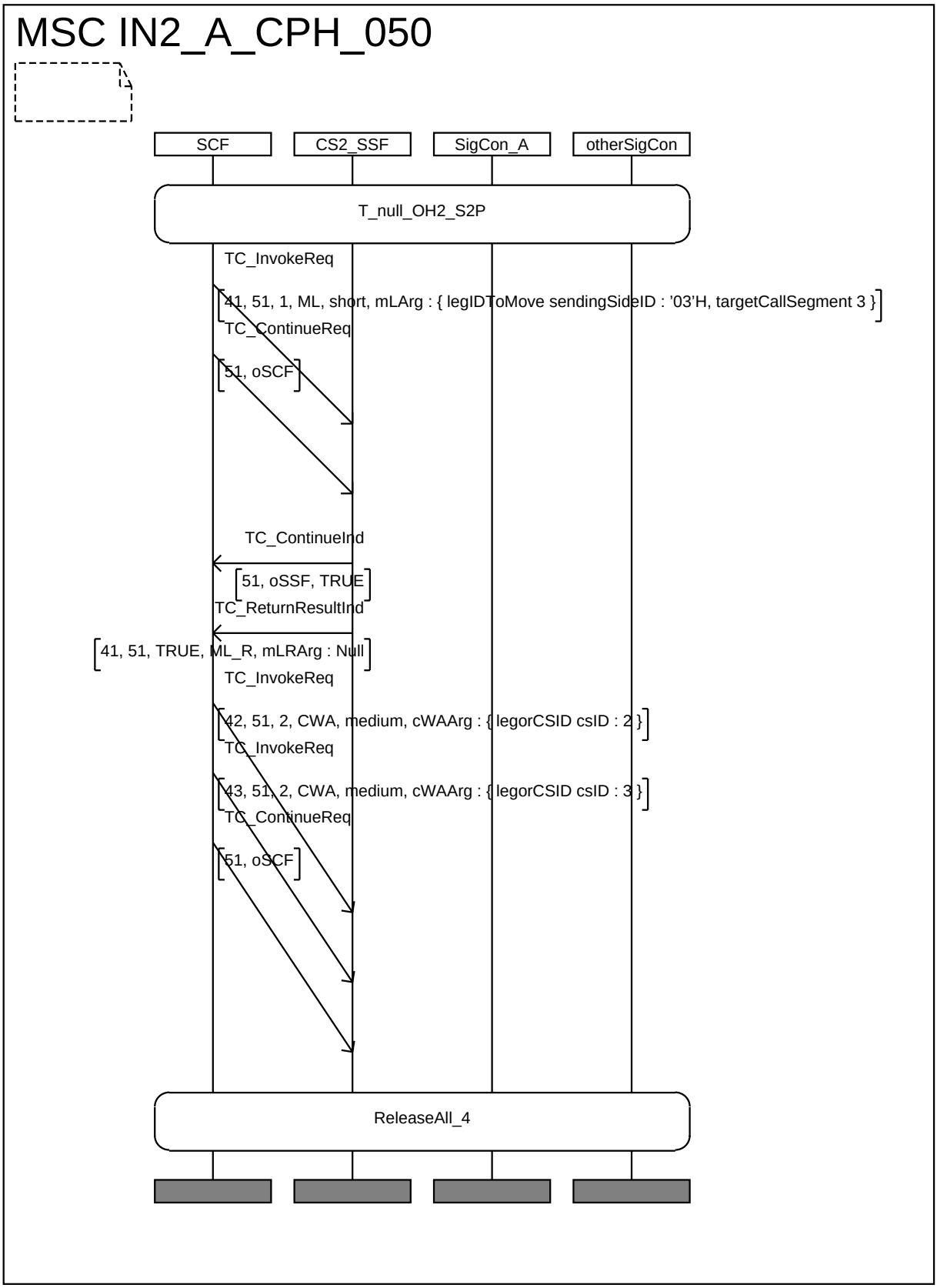
IN2_A_CPH_048	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_OH(2)_S2P
Test description	L1! MergeCallSegments(2,3) L1! ContinueWithArgument(CsId=3) Reaching state T_null_null_S4P
Pass criteria	SSF send MergeCallSegmentsReturnResult
Postamble:	ReleaseAll_4


MSC IN2_A_CPH_048
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_OH2_S2P
TC_InvokeReq
41, 51, 1, MC, medium, mCArg : { sourceCallSegment 2, targetCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

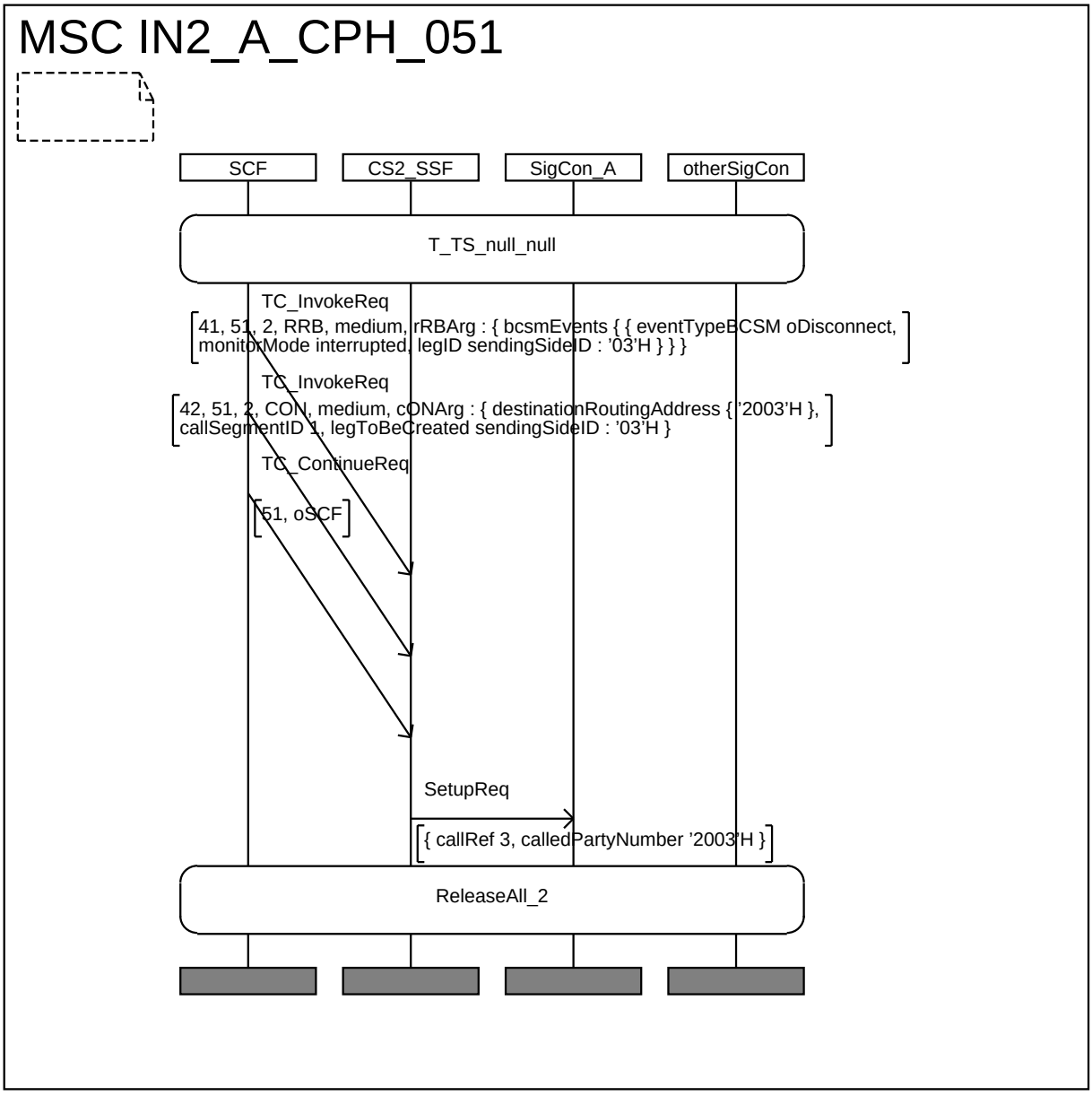
IN2_A_CPH_049	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_OH(2)_S2P
Test description	L1! DisconnectLeg(3) L1! DisconnectLeg(1) L1? DisconnectLegReturnResult L1? DisconnectLegReturnResult Reaching state T_null_null_S2P
Pass criteria	CP1_3? ReleaseReq CP1_1? ReleaseReq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


MSC IN2_A_CPH_049
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_OH2_S2P
TC_InvokeReq
41, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 3, cause '0F'H }, senderRef
TC_InvokeReq
42, 51, 1, DL, short, dLArg : { legToBeReleased sendingSideID : '01'H }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
42, 51, TRUE, DL_R, dLRArg : Null
ReleaseReq
{ callRef 1, cause '11'H }, senderRef
ReleaseAll_2

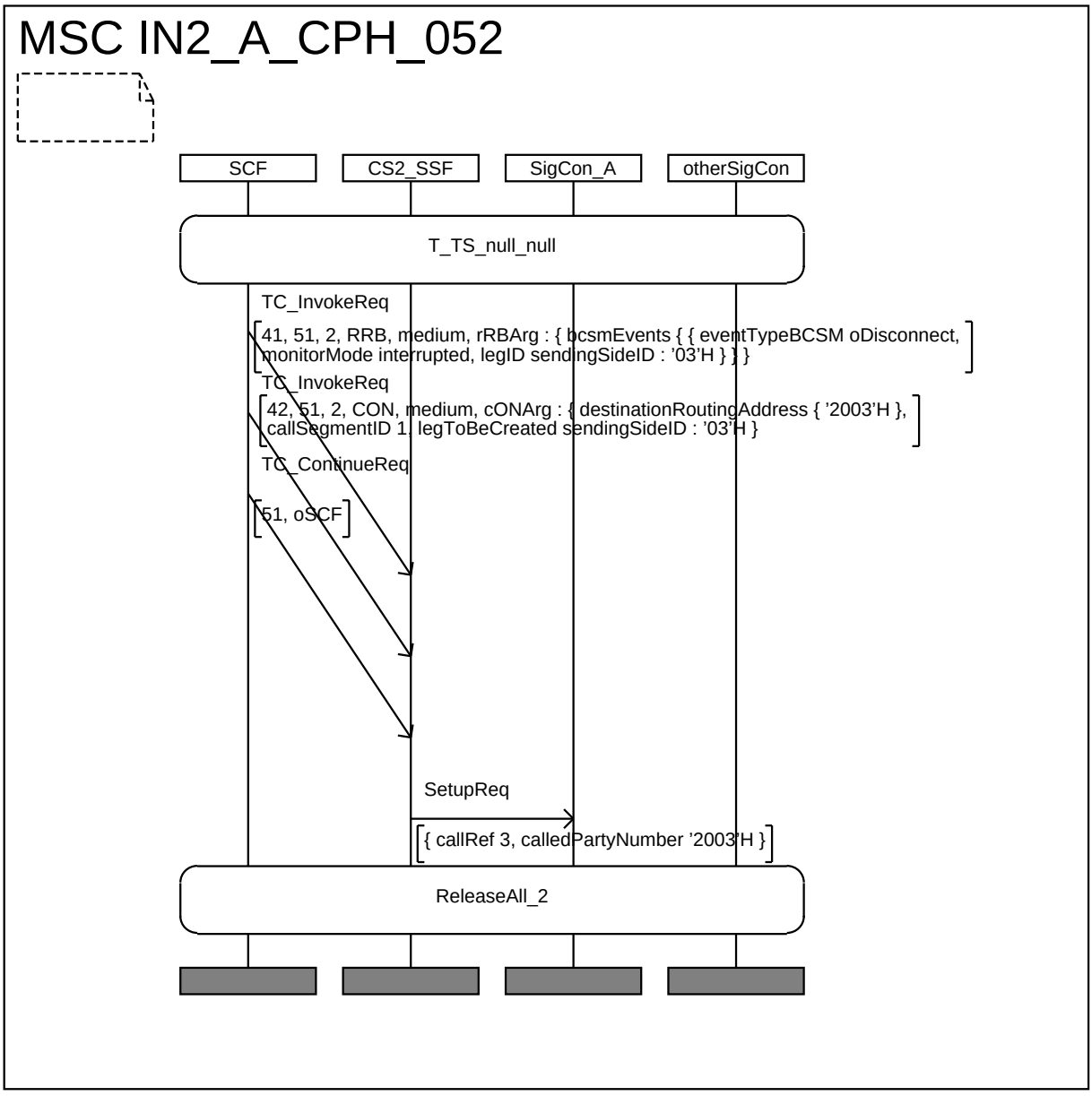
IN2_A_CPH_050	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_null_OH(2)_S2P
Test description	L1! MoveLeg(3,3) L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3) Reaching state T_null_OH(1)_S3P
Pass criteria	SSF sends MoveLegReturnResult
Postamble:	ReleaseAll_4


MSC IN2_A_CPH_050
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_OH2_S2P
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '03'H, targetCallSegment 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
ReleaseAll_4

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

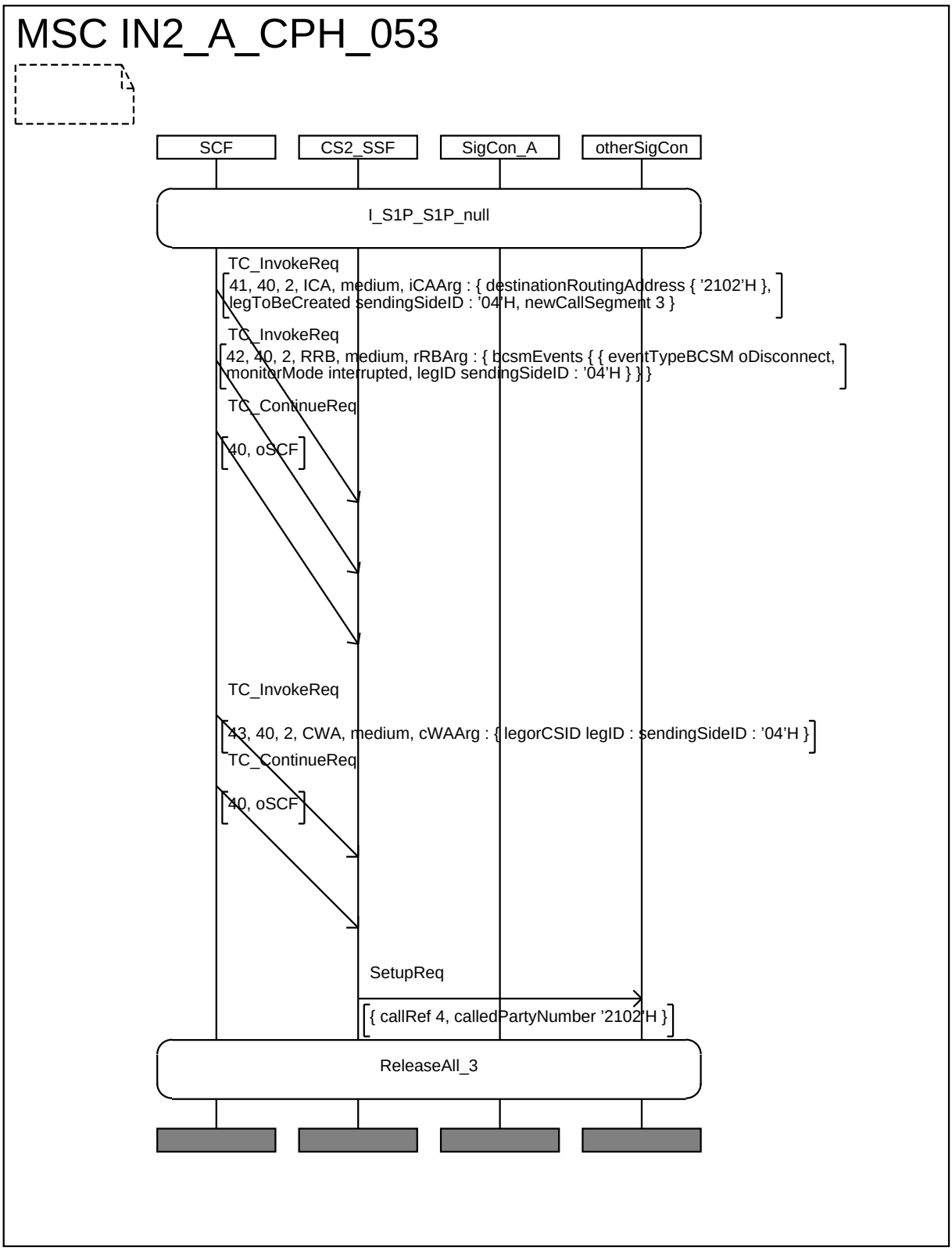


IN2_A_CPH_052	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	T_TS_null_null
Test description	L1! RequestReportBCSMEvent(3,oDisconnect) L1! RequestReportBCSMEvent(4,oDisconnect) L1! Connect(3,1) L1! Connect(4,1) Reaching state T_TF(3)_null_null
Pass criteria	CP1_3? SetUpReq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

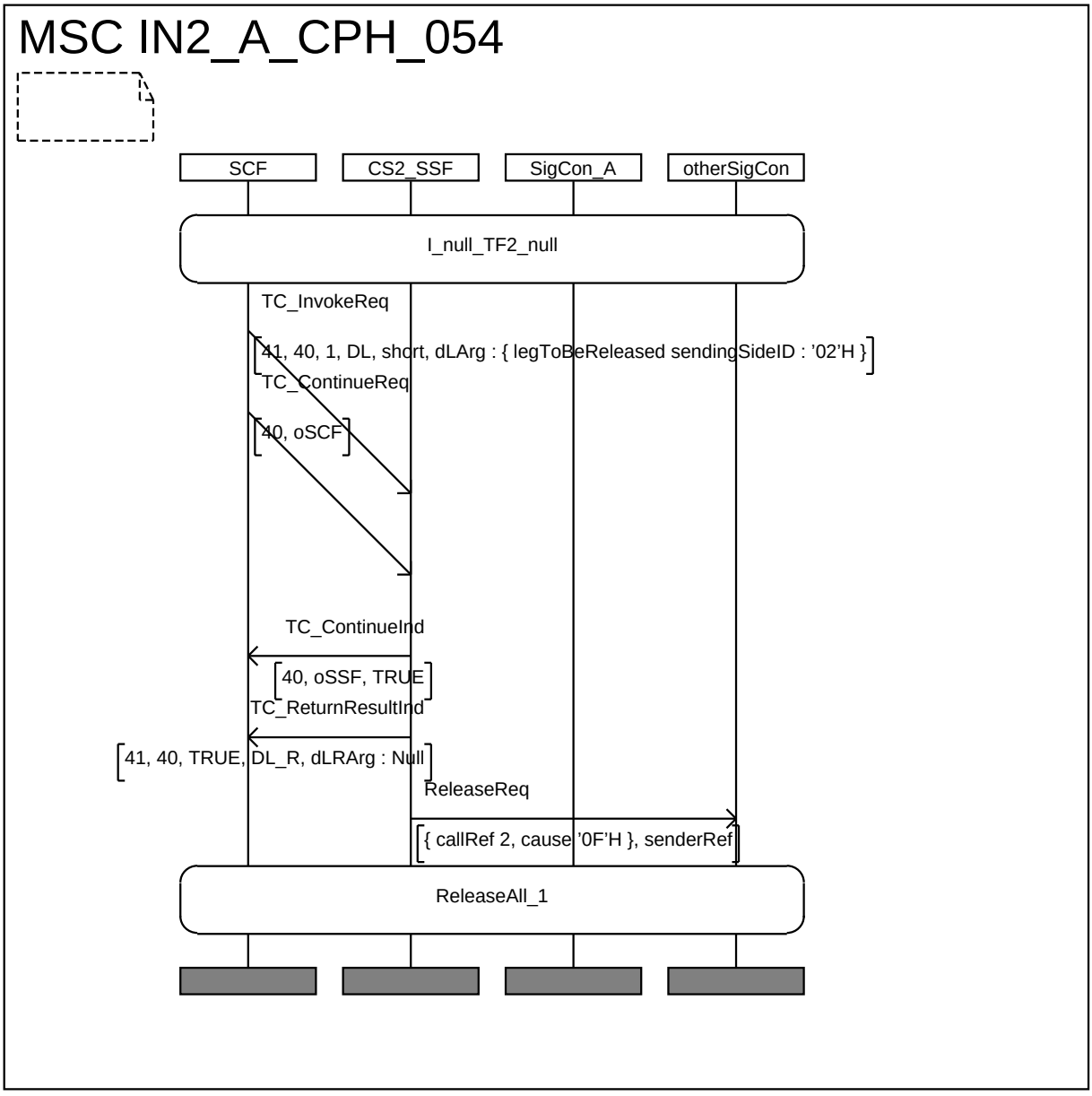

MSC IN2_A_CPH_052
SCF
CS2_SSF
SigCon_A
otherSigCon
T_TS_null_null
TC_InvokeReq
41, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
42, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 1, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseAll_2

7.5.3 Network initiated

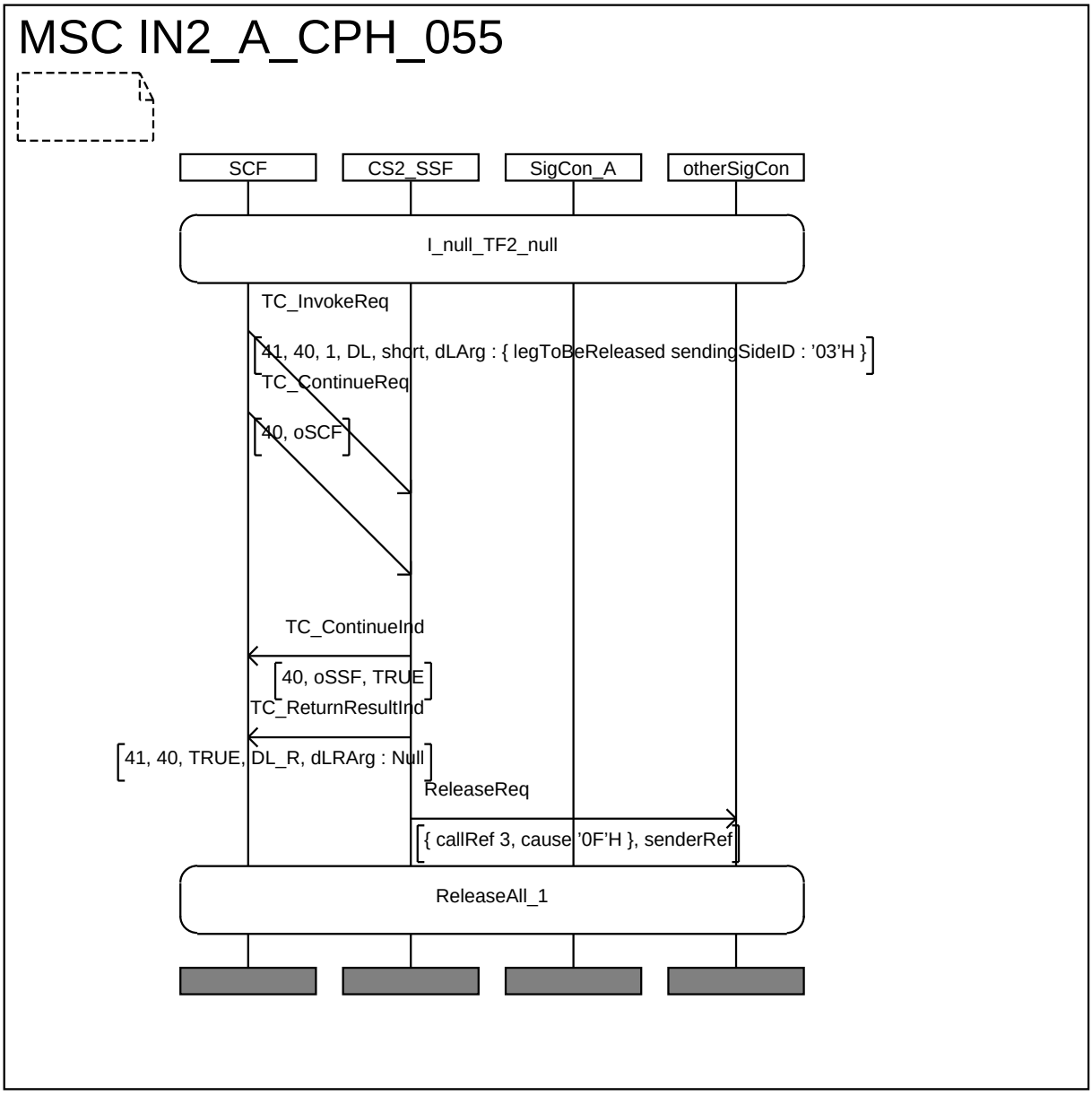
IN2_A_CPH_053	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	I_S1P_S1P_null
Test description	L1! InitiateCallAttempt(4,3) L1! RequestReportBCSMEvent(4,oDisconnect) L1! ContinueWithArgument(LegId=4) Reaching state I_S1P_S1P_S1P
Pass criteria	CP1_4? SetUpInd
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_053
SCF
CS2_SSF
SigCon_A
otherSigCon
I_S1P_S1P_null
TC_InvokeReq
41, 40, 2, ICA, medium, iCAArg : { destinationRoutingAddress { '2102'H },
legToBeCreated sendingSideID : '04'H, newCallSegment 3 }
TC_InvokeReq
42, 40, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oDisconnect,
monitorMode interrupted, legID sendingSideID : '04'H } } }
TC_ContinueReq
40, oSCF
TC_InvokeReq
43, 40, 2, CWA, medium, cWAArg : { legorCSID legID : sendingSideID : '04'H }
TC_ContinueReq
40, oSCF
SetupReq
{ callRef 4, calledPartyNumber '2102'H }
ReleaseAll_3

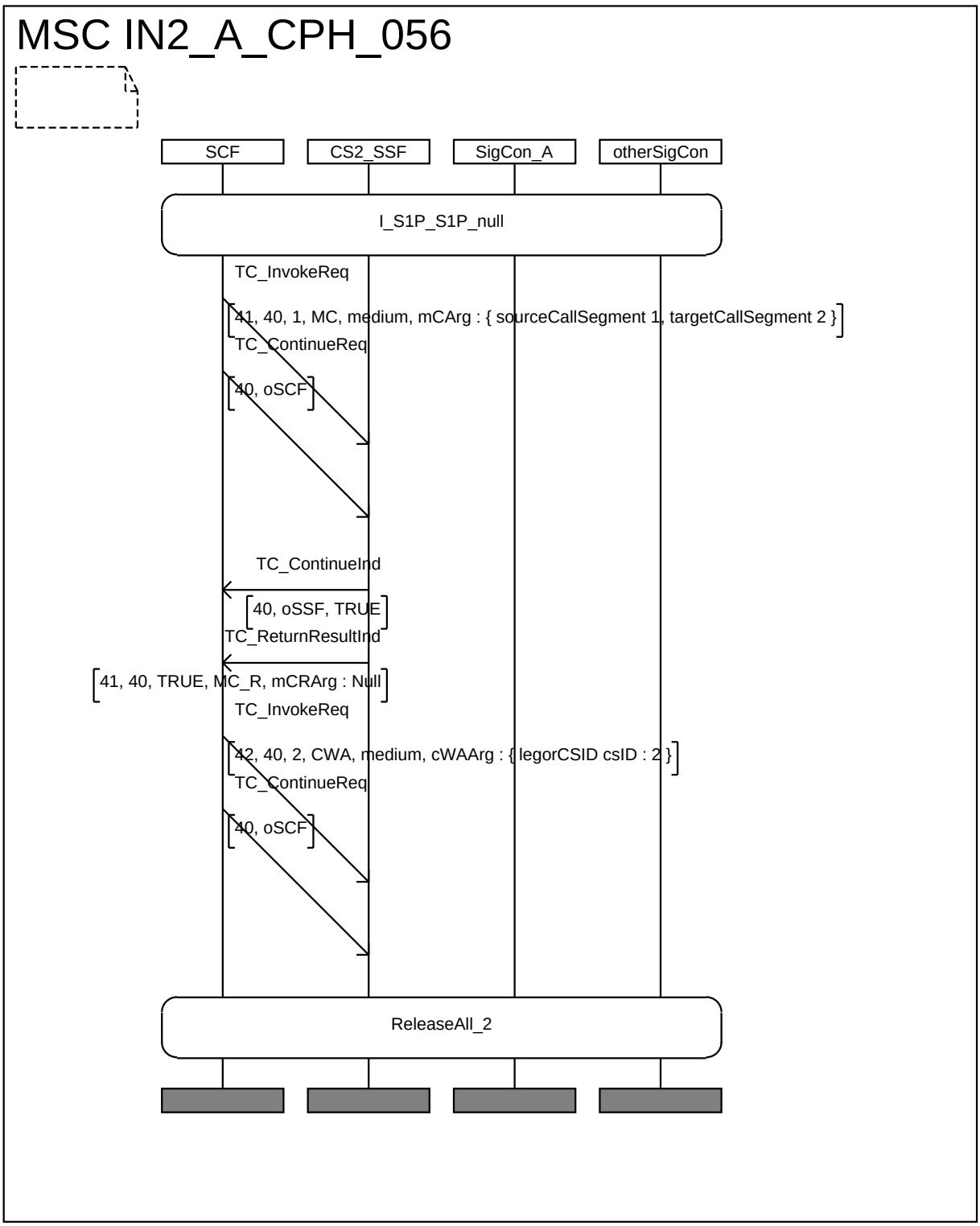
IN2_A_CPH_054	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	I_null_TF2_null
Test description	L1! DisconnectLeg(2) L1? DisconnectLegReturnResult Reaching state I_null_S1P_null
Pass criteria	CP1_2? ReleaseReq
Postamble:	ReleaseAll_1



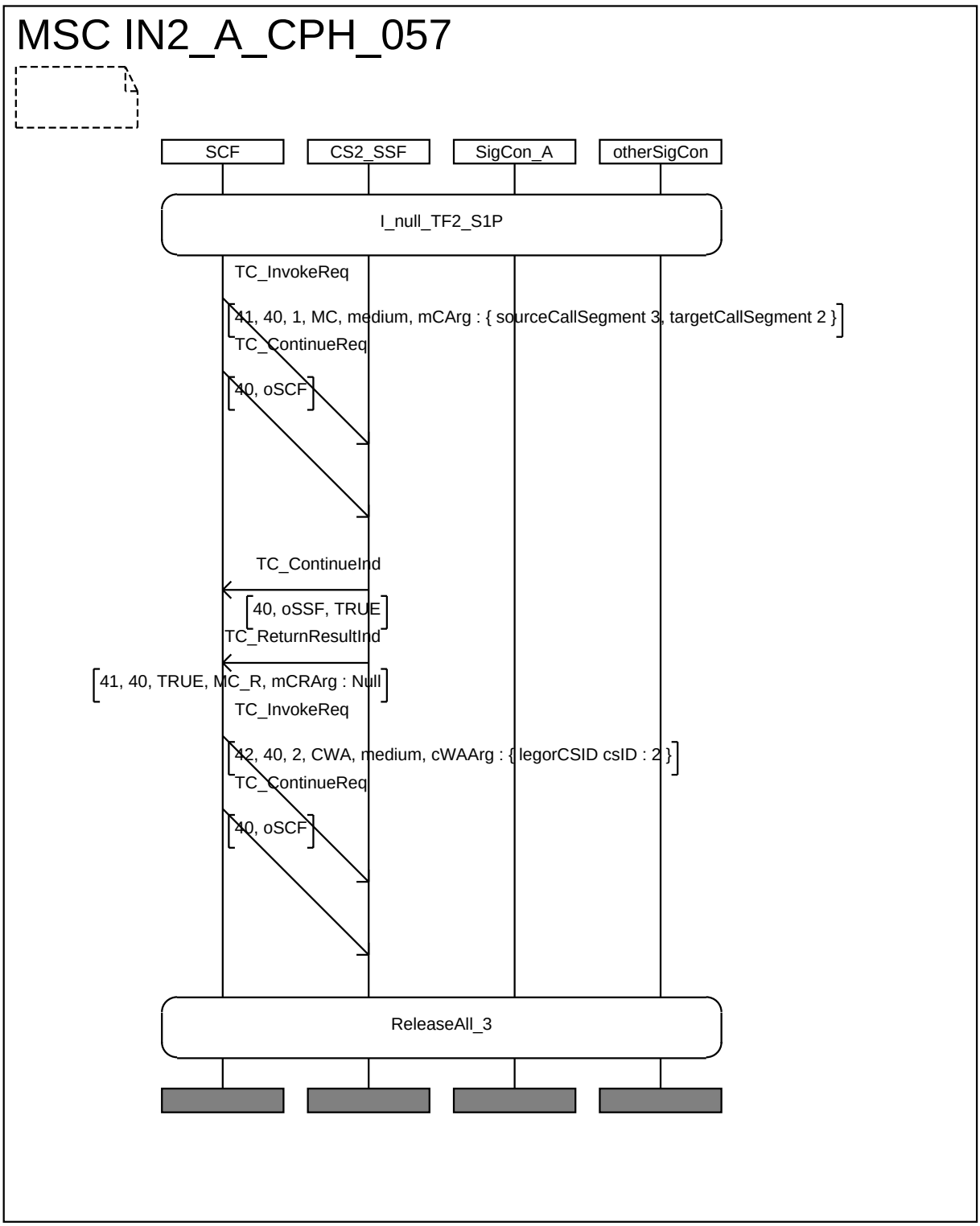
IN2_A_CPH_055	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	I_null_TF2_null
Test description	L1! DisconnectLeg(3) L1? DisconnectLegReturnResult Reaching state I_null_S1P_null
Pass criteria	CP1_3? ReleaseReq
Postamble:	ReleaseAll_1



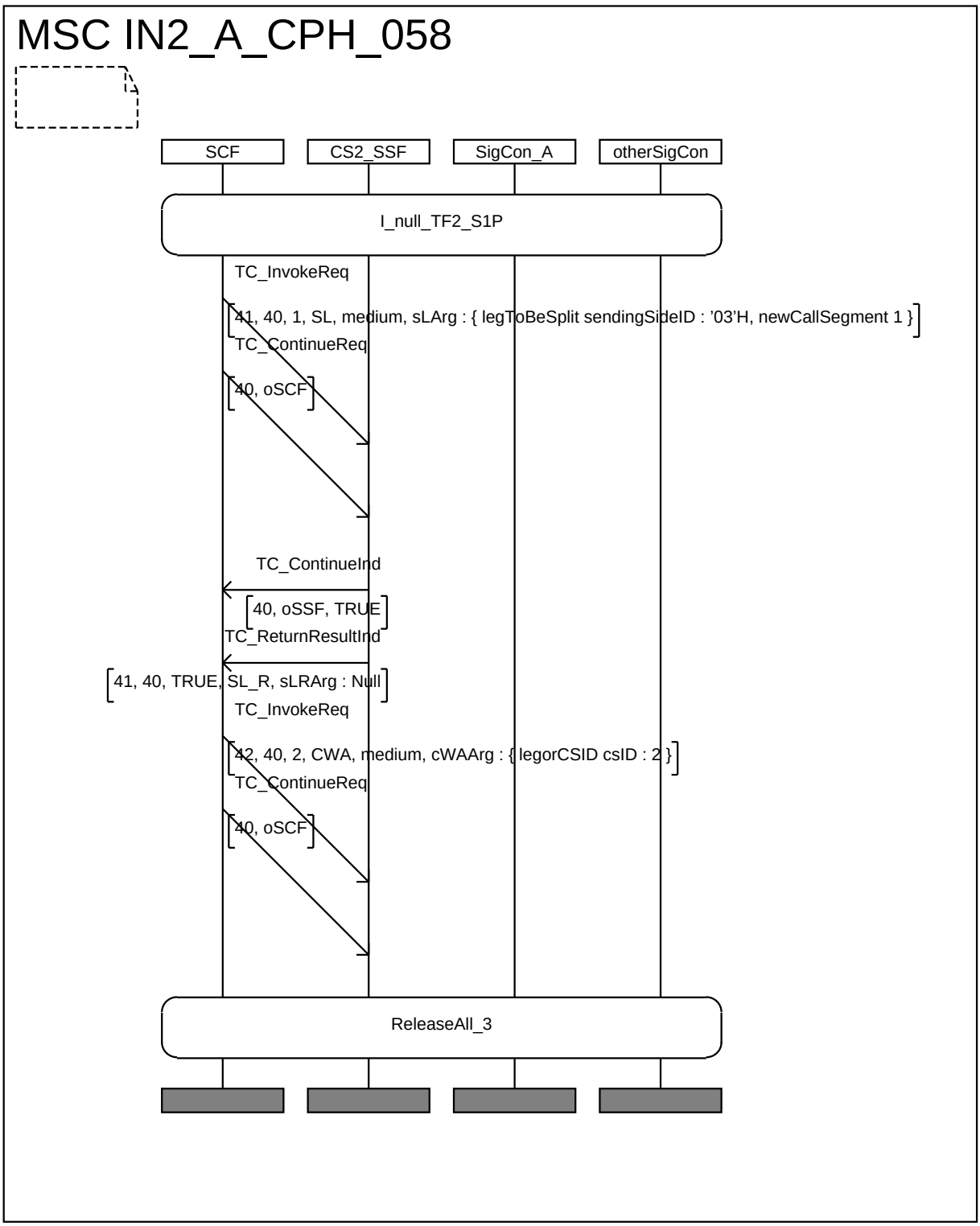
IN2_A_CPH_056	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	I_S1P_S1P_null
Test description	L1! MergeCallSegments(1,2) L1! ContinueWithArgument(CsId=2) Reaching state I_null_TF(2)_null
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble:	ReleaseAll_2


MSC IN2_A_CPH_056
SCF
CS2_SSF
SigCon_A
otherSigCon
I_S1P_S1P_null
TC_InvokeReq
41, 40, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_ContinueReq
40, oSCF
TC_ContinueInd
40, oSSF, TRUE
TC_ReturnResultInd
41, 40, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 40, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
40, oSCF
ReleaseAll_2

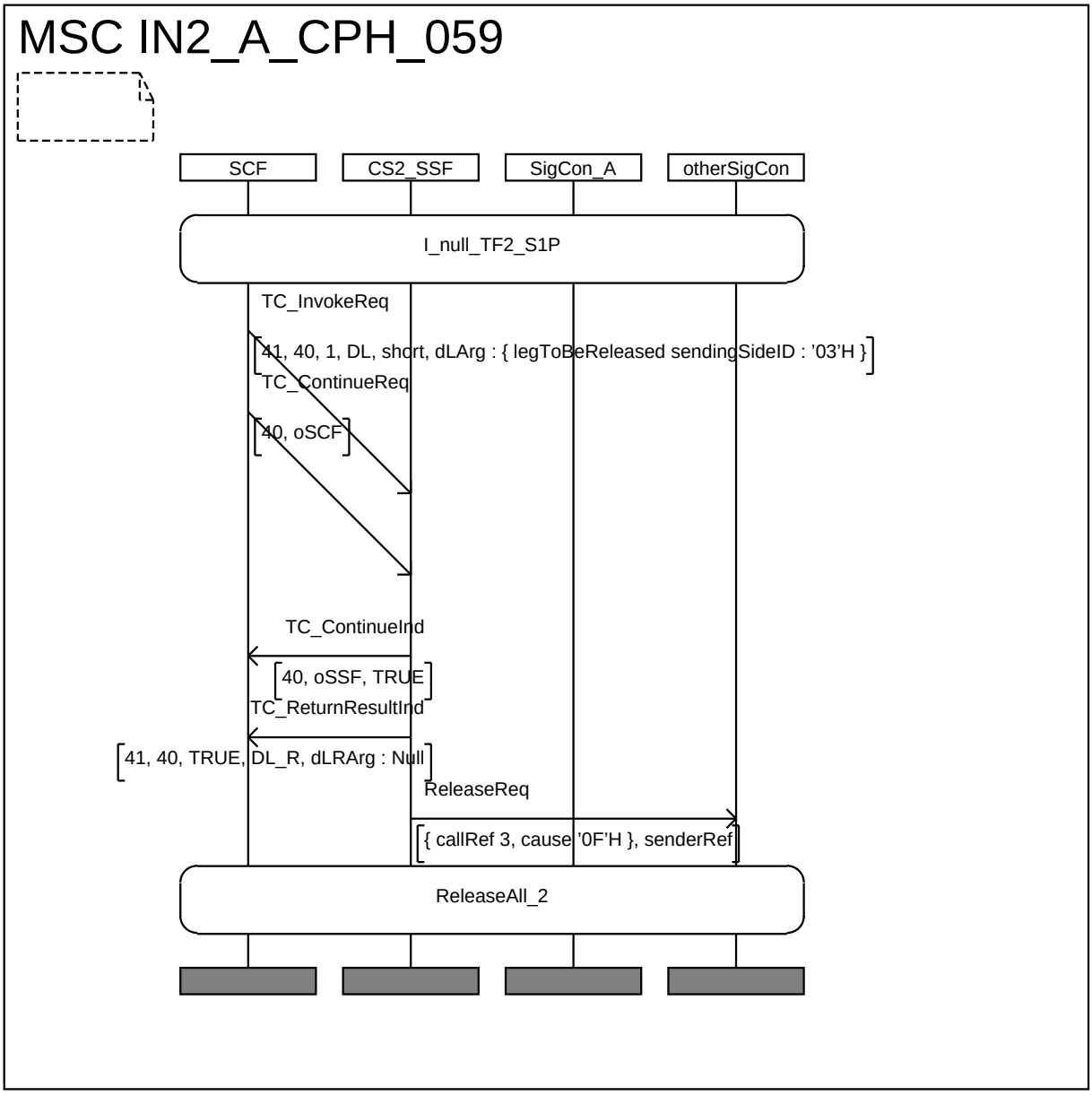
IN2_A_CPH_057	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	I_null_TF(2)_S1P
Test description	L1! MergeCallSegments(3,2) L1! ContinueWithArgument(CsId=2) Reaching state I_null_TF(3)_null
Pass criteria	SSF sends MergeCallSegmentReturnResult
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_057
SCF
CS2_SSF
SigCon_A
otherSigCon
I_null_TF2_S1P
TC_InvokeReq
41, 40, 1, MC, medium, mCArg : { sourceCallSegment 3, targetCallSegment 2 }
TC_ContinueReq
40, oSCF
TC_ContinueInd
40, oSSF, TRUE
TC_ReturnResultInd
41, 40, TRUE, MC_R, mCRArg : Null
TC_InvokeReq
42, 40, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
40, oSCF
ReleaseAll_3

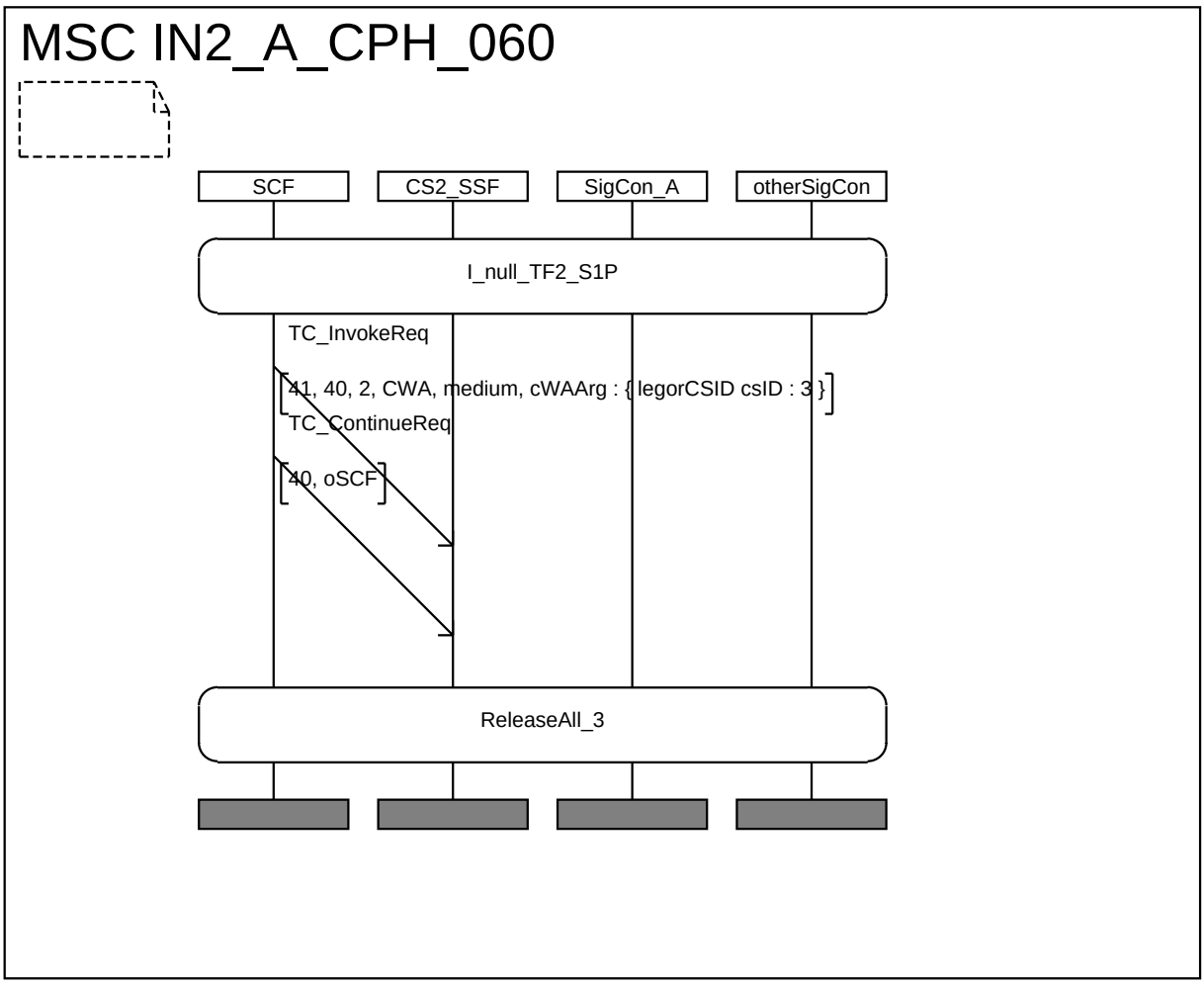
IN2_A_CPH_058	
Purpose:	Test CPH capabilities
Requirement ref	
Selection Cond.	
Preamble:	I_null_TF(2)_S1P
Test description	L1! SplitLeg(3,1) L1! ContinueWithArgument(CsId=2) Reaching state I_S1P_S1P_S1P
Pass criteria	SSF sends SplitLegReturnResult
Postamble:	ReleaseAll_3


MSC IN2_A_CPH_058
SCF
CS2_SSF
SigCon_A
otherSigCon
I_null_TF2_S1P
TC_InvokeReq
41, 40, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '03'H, newCallSegment 1 }
TC_ContinueReq
40, oSCF
TC_ContinueInd
40, oSSF, TRUE
TC_ReturnResultInd
41, 40, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
42, 40, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
40, oSCF
ReleaseAll_3

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



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

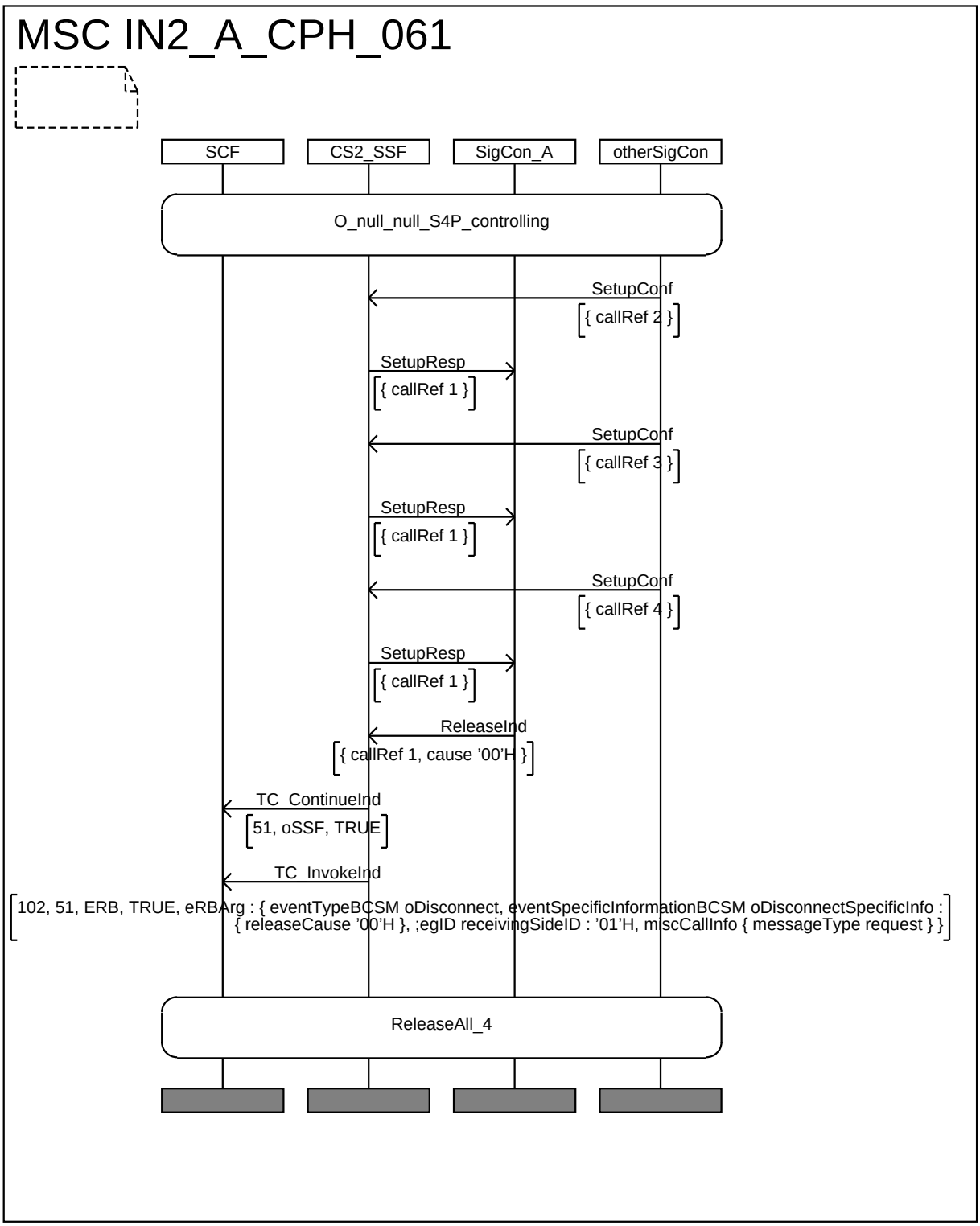
In these scenarios, the passive legs detection points are armed for signalling events coming from the controlling leg, this means that filtering rules should apply.

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

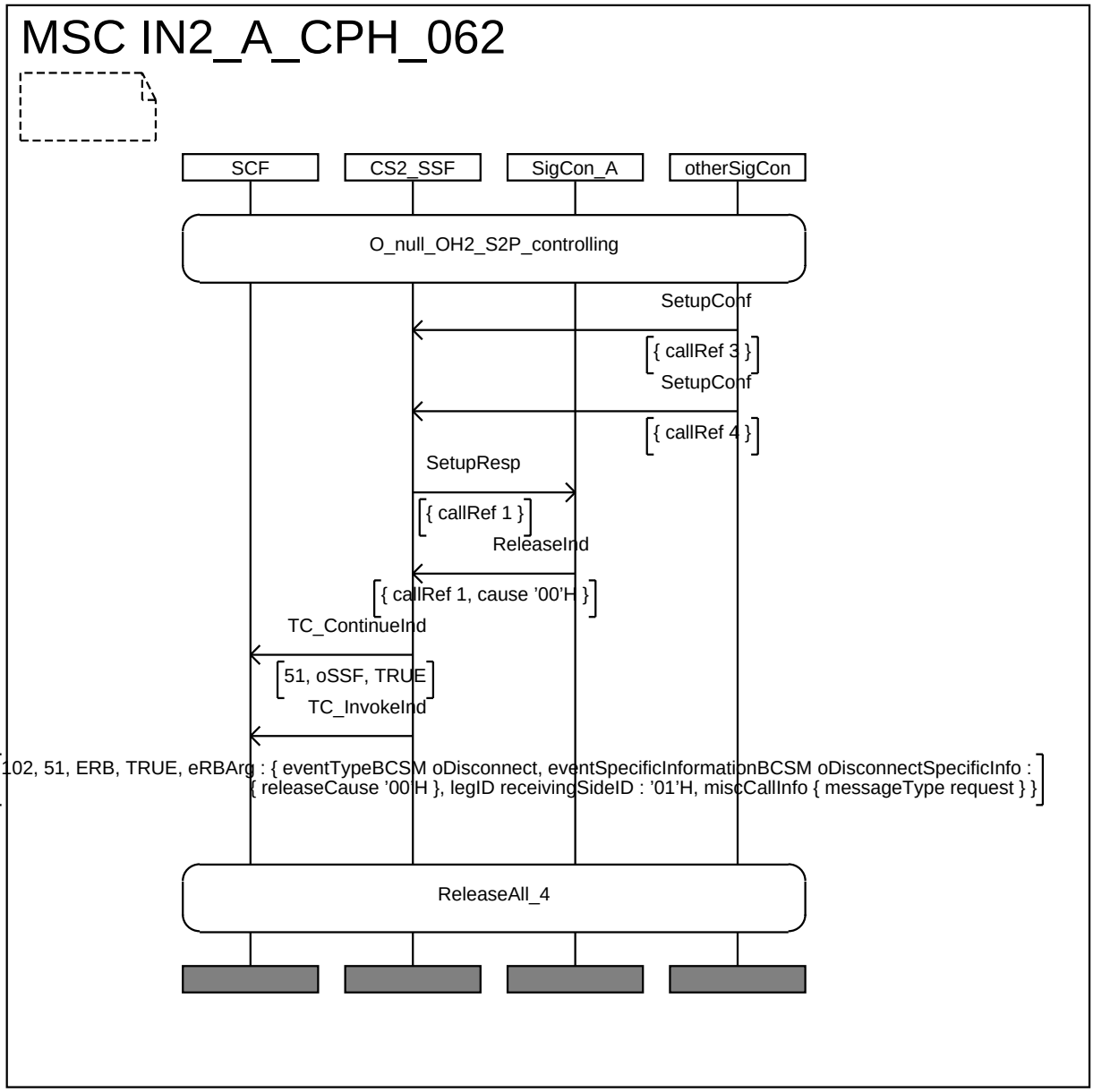
7.6.1 Originating (O) trigger

7.6.1.1 O_1 Events coming from the controlling leg (legId=1)

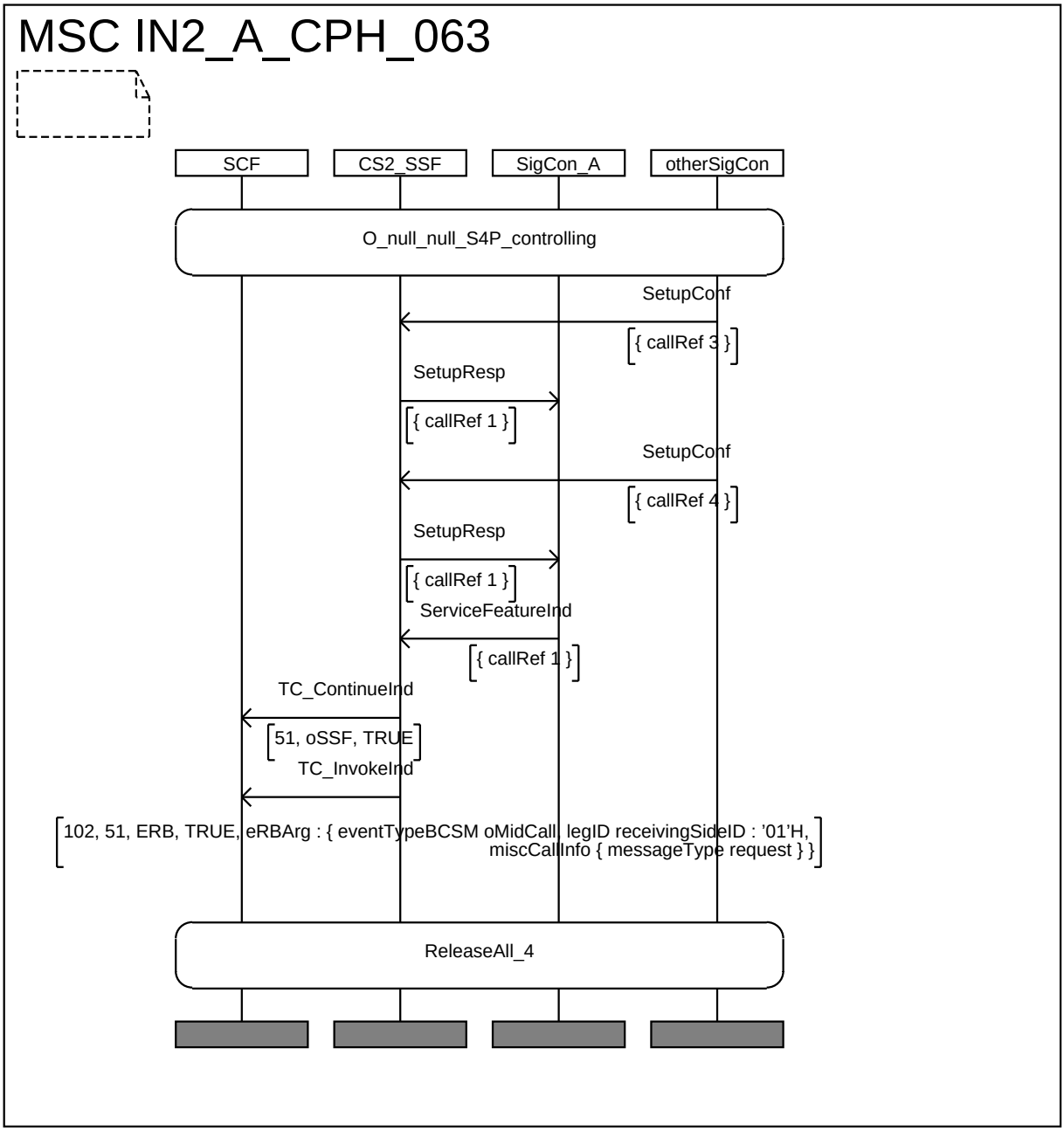
	IN2_A_CPH_061
Purpose:	Check that event filtering rules are applied eventType = oDisconnect
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P_controlling
Test description	CP1_2! SetUpConf CP1_3! SetUpConf CP1_4! SetUpConf CP1_1! ReleaseInd
Pass criteria	L1?: ERB (oDisconnect) is received only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


MSC IN2_A_CPH_061
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P_controlling
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
ReleaseInd
{ callRef 1, cause '00'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oDisconnect, eventSpecificInformationBCSM oDisconnectSpecificInfo :
{ releaseCause '00'H }, ;egID receivingSideID : '01'H, miscCallInfo { messageType request } }
ReleaseAll_4

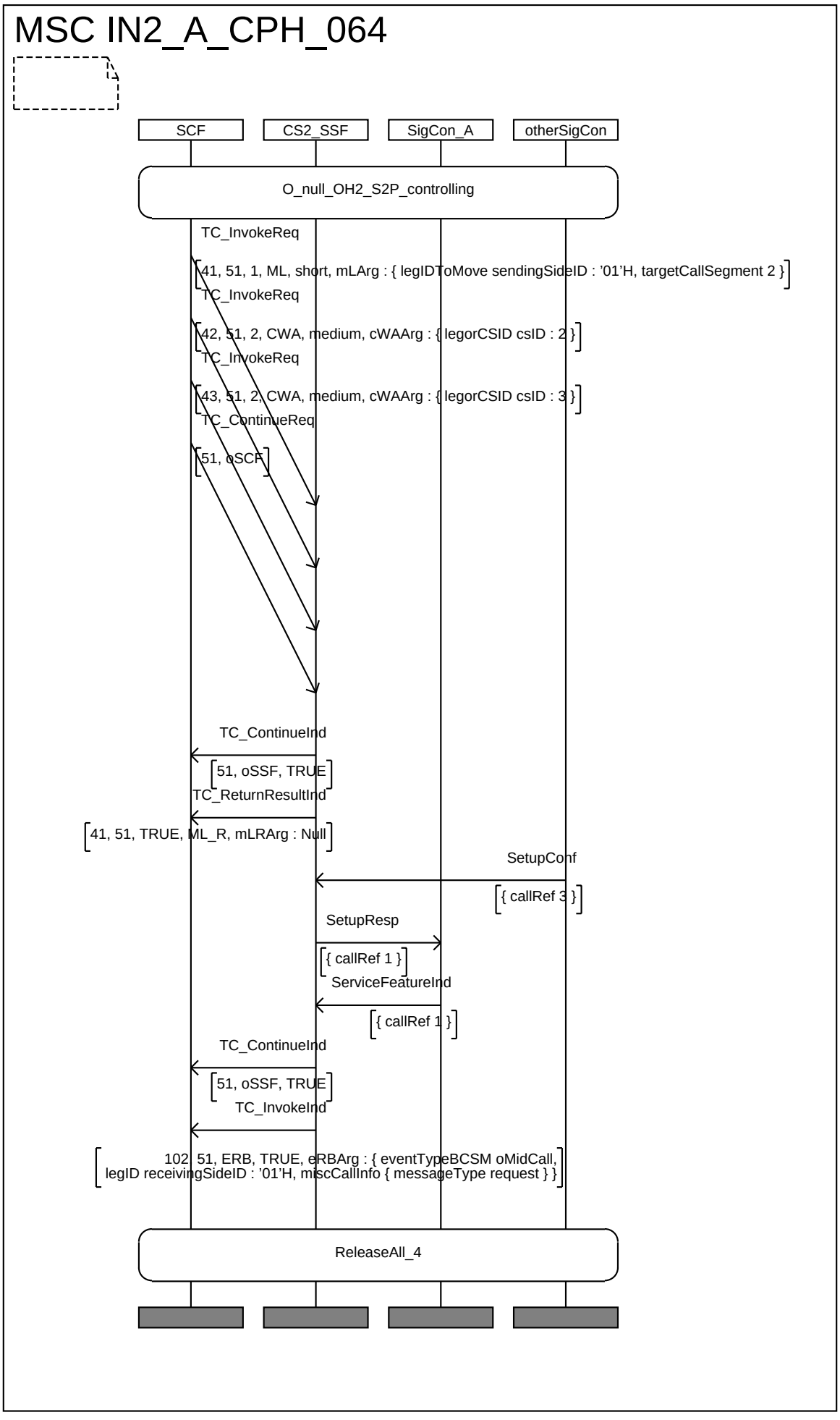
IN2_A_CPH_062	
Purpose:	Check that event filtering rules are applied for multiple callsegments eventType = oDisconnect
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P_controlling
Test description	CP1_3! SetUpConf CP1_4! SetUpConf CP1_1! ReleaseInd
Pass criteria	L1?: ERB (oDisconnect) is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



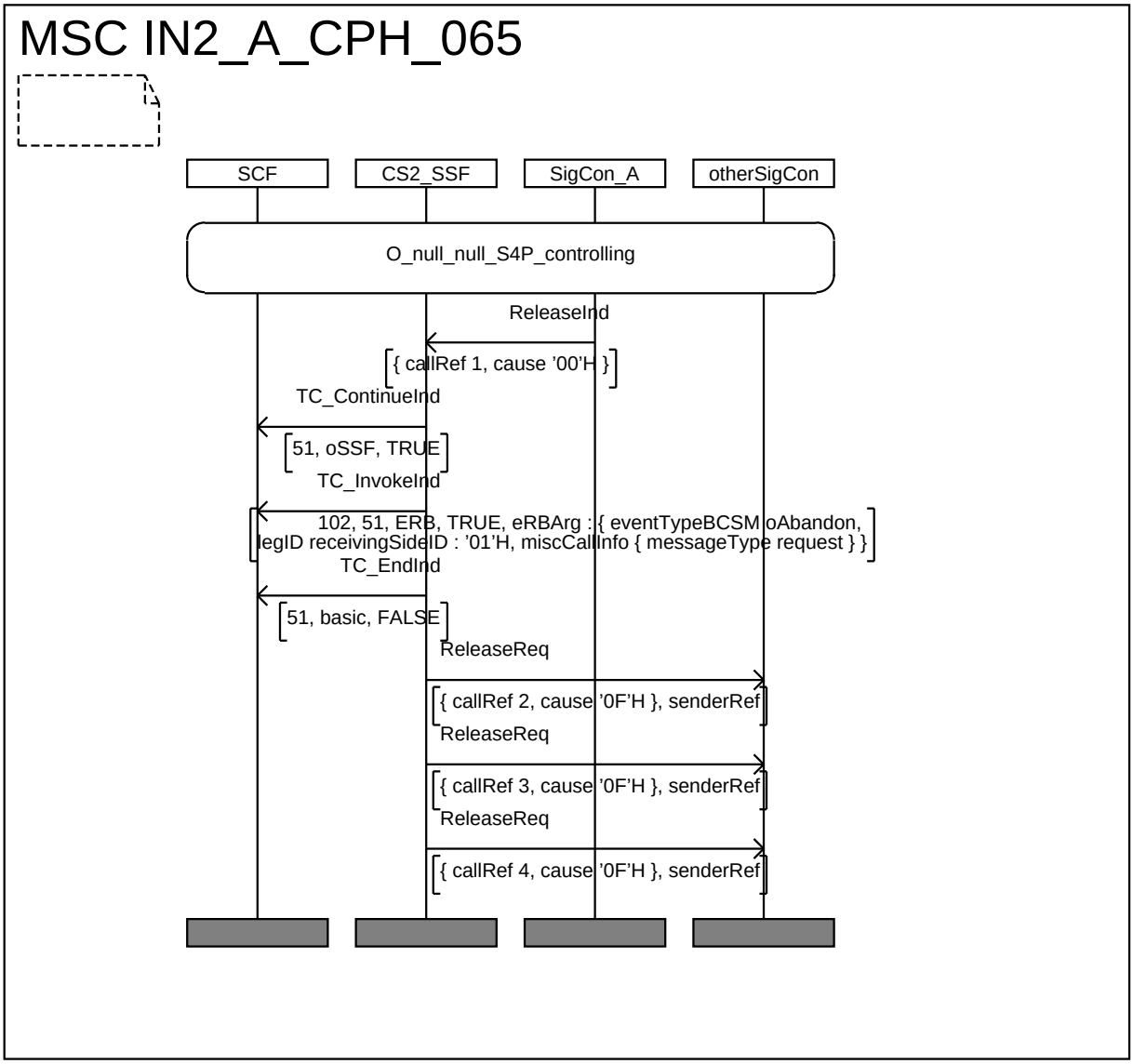
IN2_A_CPH_063	
Purpose:	Check that event filtering rules are applied eventType = oMidcall
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P_controlling
Test description	CP1_3! SetUpConf CP1_4! SetUpConf CP1_1! ServiceFeatureIndication
Pass criteria	L1?: ERB (oMidCall) is received only once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


MSC IN2_A_CPH_063
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P_controlling
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
SetupConf
{ callRef 4 }
SetupResp
{ callRef 1 }
ServiceFeatureInd
{ callRef 1 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oMidCall legID receivingSideID : '01'H,
miscCallInfo { messageType request } }
ReleaseAll_4

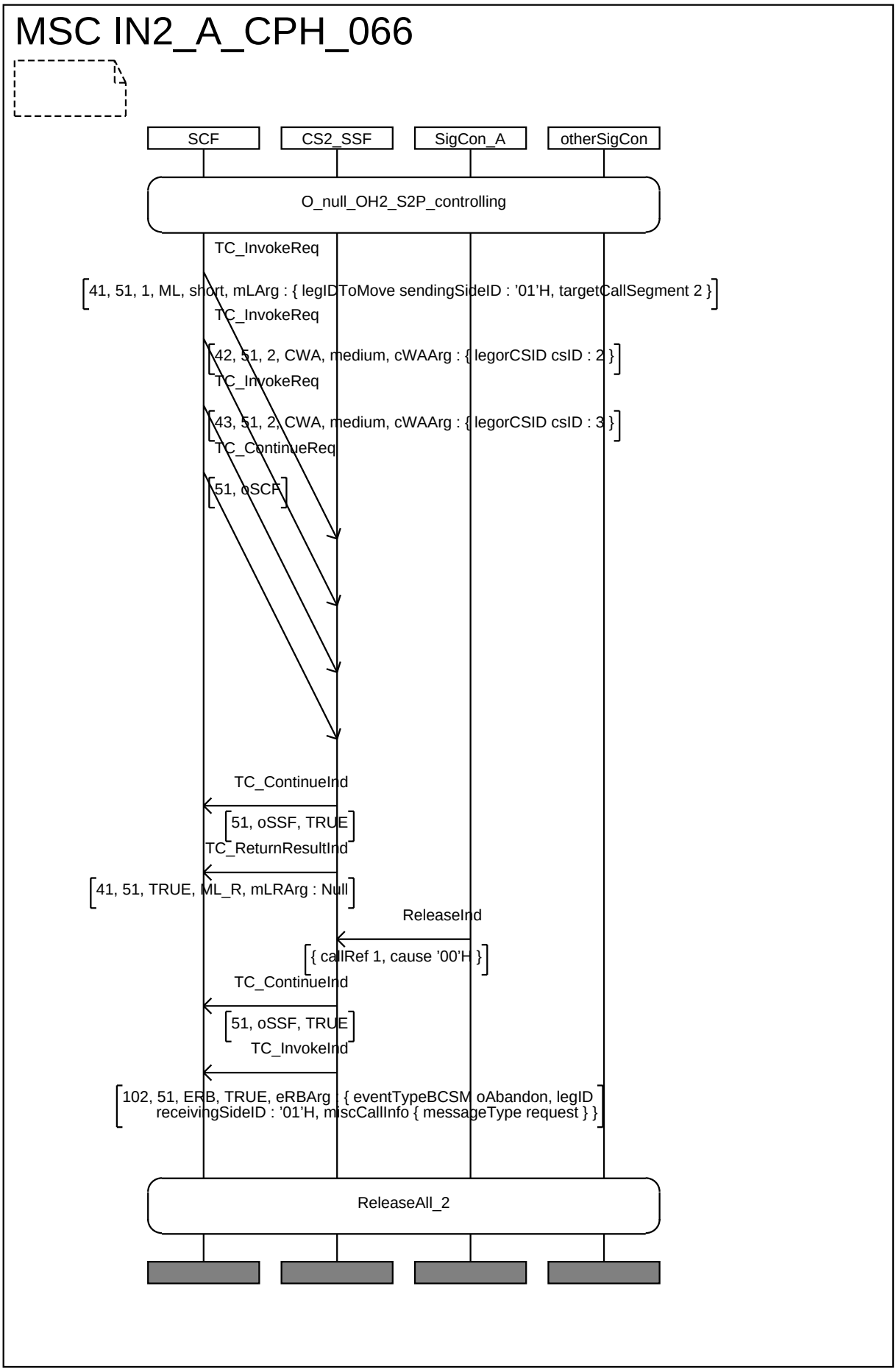
IN2_A_CPH_064	
Purpose:	Check that event filtering rules are applied for multiple callsegments eventType = oMidcall
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P_controlling
Test description	L1! MoveLeg(1,2) L1? MoveLegReturnResult L1! ContinueWithArgument(CsId=2) L1! ContinueWithArgument(CsId=3)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


MSC IN2_A_CPH_064
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_OH2_S2P_controlling
TC_InvokeReq
41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, ML_R, mLRArg : Null
SetupConf
{ callRef 3 }
SetupResp
{ callRef 1 }
ServiceFeatureInd
{ callRef 1 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oMidCall, legID receivingSideID : '01'H, miscCallInfo { messageType request } }
ReleaseAll_4

IN2_A_CPH_065	
Purpose:	Check that event filtering rules are applied eventType = oAbandon
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P_controlling
Test description	CP1_1! ReleaseInd
Pass criteria	L1?: ERB (oAbandon) is received only once (filtering rules apply)
Postamble:	none

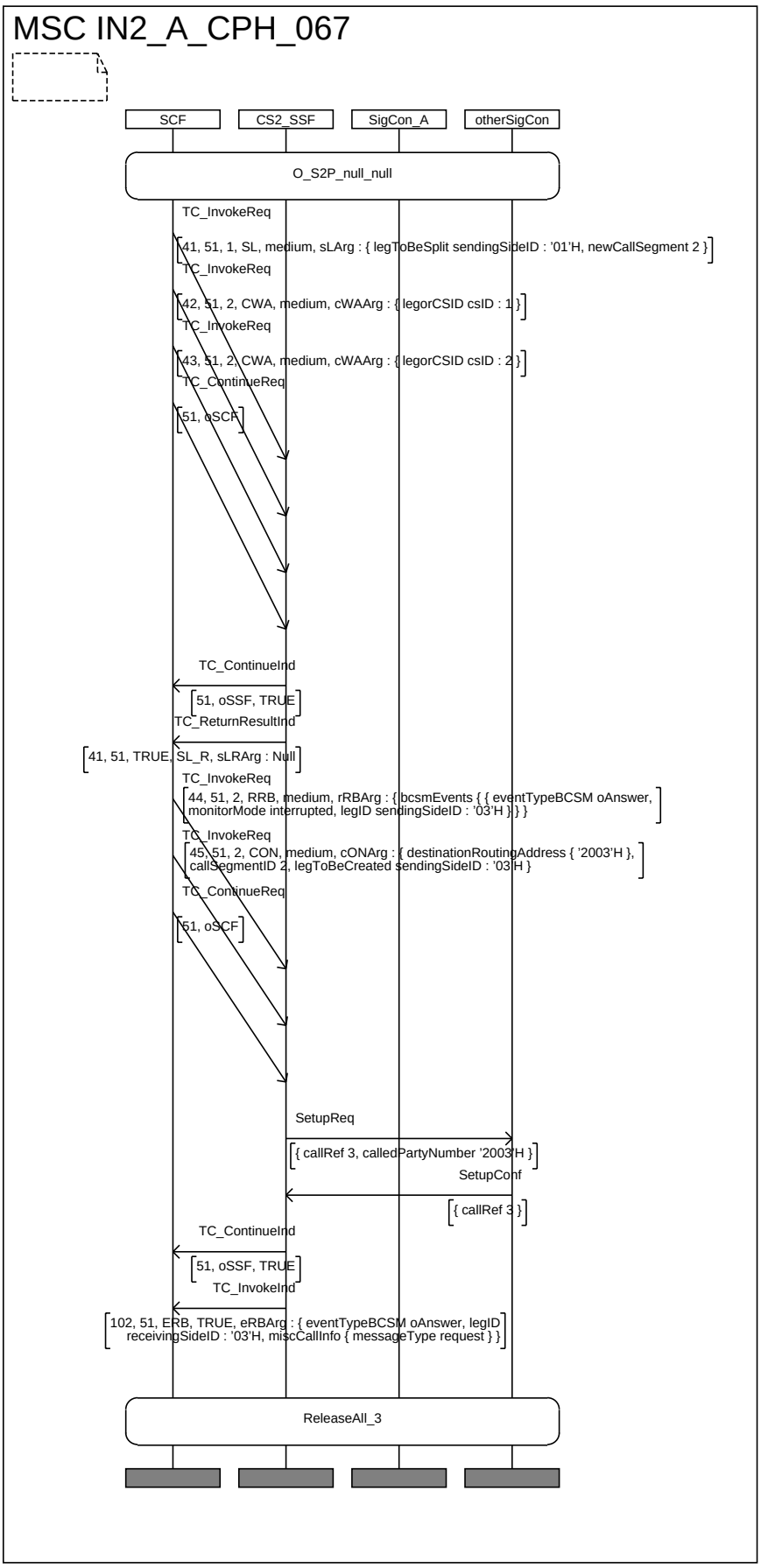


IN2_A_CPH_066	
Purpose:	Check that event filtering rules are applied for multiple callsegments eventType = oAbandon
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P_controlling
Test	L1! MoveLeg(1,2) L1? MoveLegReturnResult L1! ContinueWithArgumentCsId=2) L1! ContinueWithArgumentCsId=3) CP1_1!:ReleaseInd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

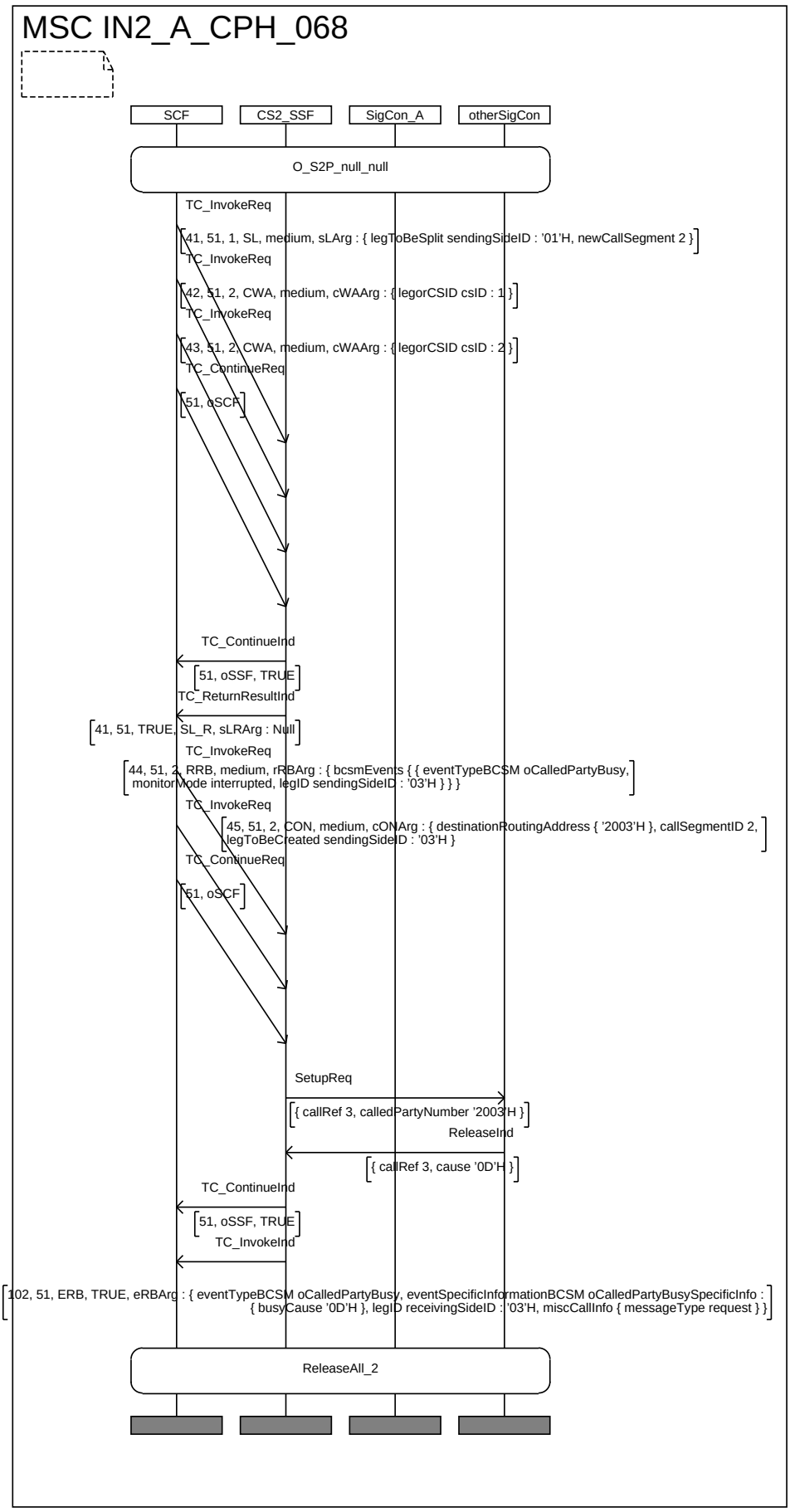

MSC IN2_A_CPH_066
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_OH2_S2P_controlling
TC_InvokeReq
[41, 51, 1, ML, short, mLArg : { legIDToMove sendingSideID : '01'H, targetCallSegment 2 }]
TC_InvokeReq
[42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }]
TC_InvokeReq
[43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }]
TC_ContinueReq
[51, oSCF]
TC_ContinueInd
[51, oSSF, TRUE]
TC_ReturnResultInd
[41, 51, TRUE, ML_R, mLRArg : Null]
ReleaseInd
[{ callRef 1, cause '00'H }]
TC_ContinueInd
[51, oSSF, TRUE]
TC_InvokeInd
[102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAbandon, legID receivingSideID : '01'H, miscCallInfo { messageType request } }]
ReleaseAll_2

7.6.1.2 O_2 Events coming from passive legs (legId= 2,3...)

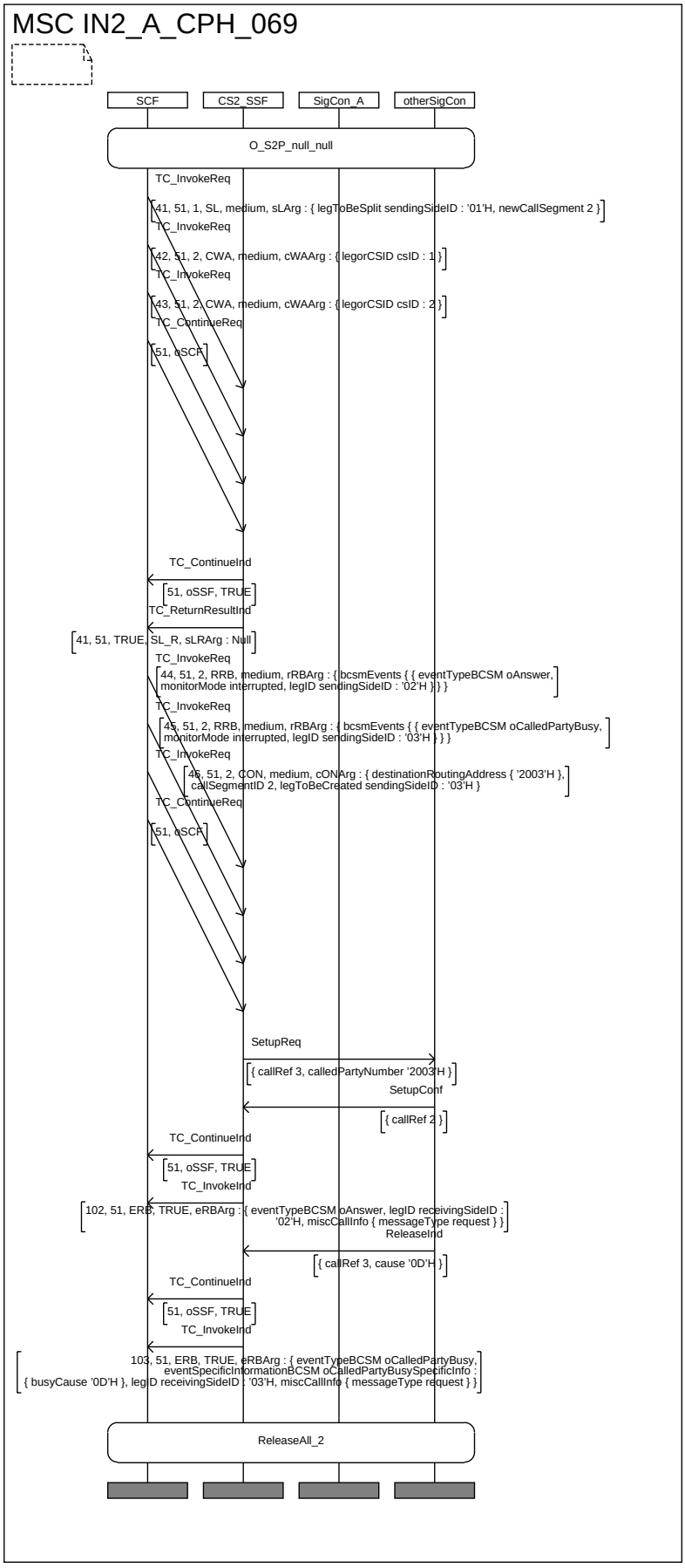
	IN2_A_CPH_067
Purpose:	Check that a single event oAnswer is detected on a passive leg
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test	L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oAnswer) L1! Connect(3,2) CP1_3? SetUpReq CP1_3! SetupConf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


MSC IN2_A_CPH_067
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
44, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oAnswer, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
45, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
SetupConf
{ callRef 3 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_3

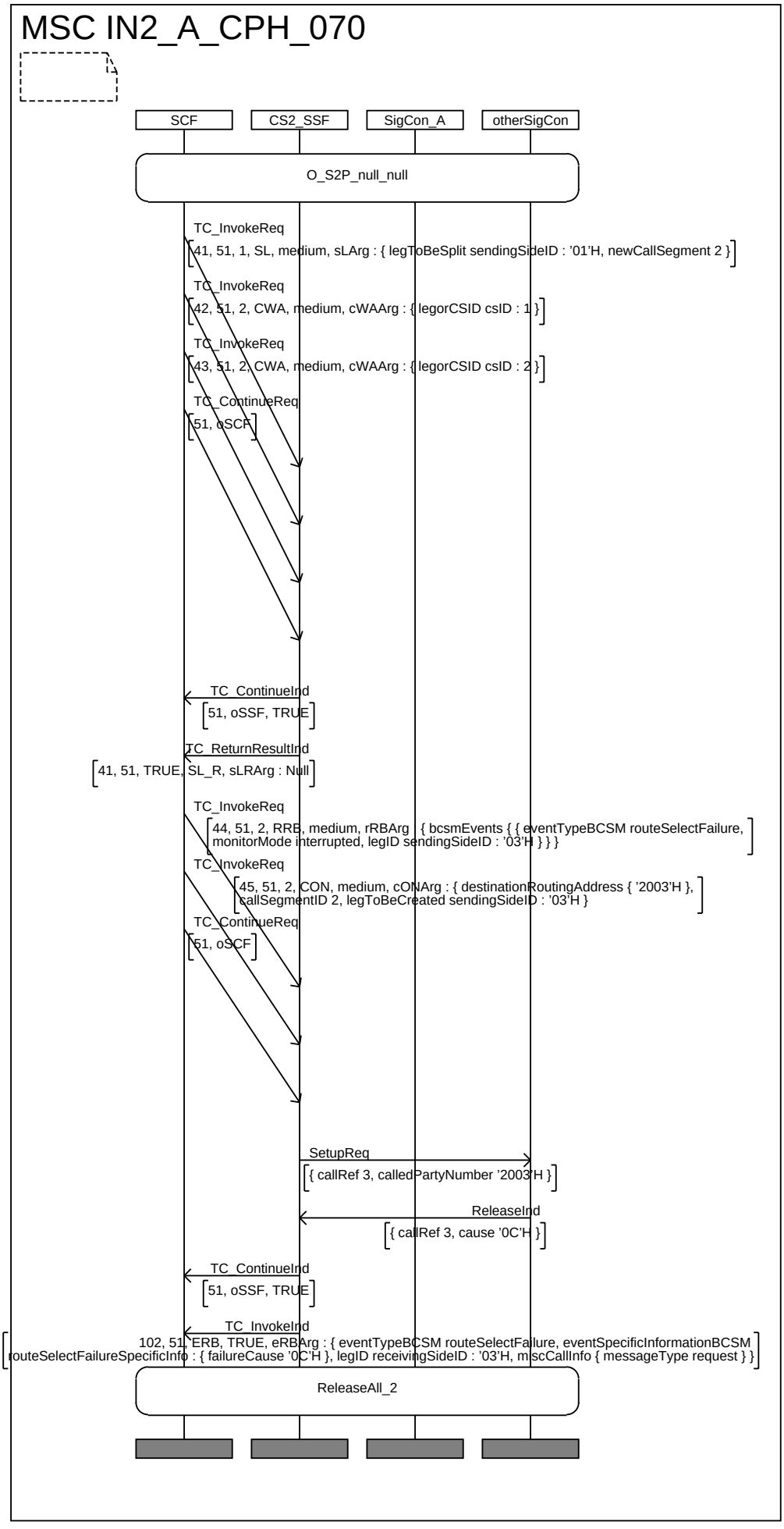
IN2_A_CPH_068	
Purpose:	Check that a single event oCalledPartyBusy is detected on a passive leg
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test	L1! SplitLeg(1,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oBusy, notify, oAnswer) L1! Connect(3,2)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


MSC IN2_A_CPH_068
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
44, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oCalledPartyBusy, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
45, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseInd
{ callRef 3, cause '0D'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oCalledPartyBusy, eventSpecificInformationBCSM oCalledPartyBusySpecificInfo : { busyCause '0D'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

IN2_A_CPH_069	
Purpose:	Check after two events are armed, they are reported
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test	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))
Pass criteria	L1? ERB(2,interrupted,oAnswer) L1? ERB(3,interrupted,oBusy)
Postamble:	ReleaseAll_2


MSC IN2_A_CPH_069
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
44, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oAnswer, monitorMode interrupted, legID sendingSideID : '02'H } } }
TC_InvokeReq
45, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oCalledPartyBusy, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
46, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
SetupConf
{ callRef 2 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '02'H, miscCallInfo { messageType request } }
ReleaseInd
{ callRef 3, cause '0D'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
103, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oCalledPartyBusy, eventSpecificInformationBCSM oCalledPartyBusySpecificInfo : { busyCause '0D'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

IN2_A_CPH_070	
Purpose:	Check that a single event routeSelectFailure is detected on a passive leg
Requirement ref	
Selection Cond.	
Preamble:	O_S2P_null_null
Test	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)
Pass criteria	L1? ERB(3,interrupted, oRouteSelectFailure)
Postamble:	ReleaseAll_2


MSC IN2_A_CPH_070
SCF
CS2_SSF
SigCon_A
otherSigCon
O_S2P_null_null
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
44, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM routeSelectFailure, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
45, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseInd
{ callRef 3, cause '0C'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM routeSelectFailure, eventSpecificInformationBCSM routeSelectFailureSpecificInfo : { failureCause '0C'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

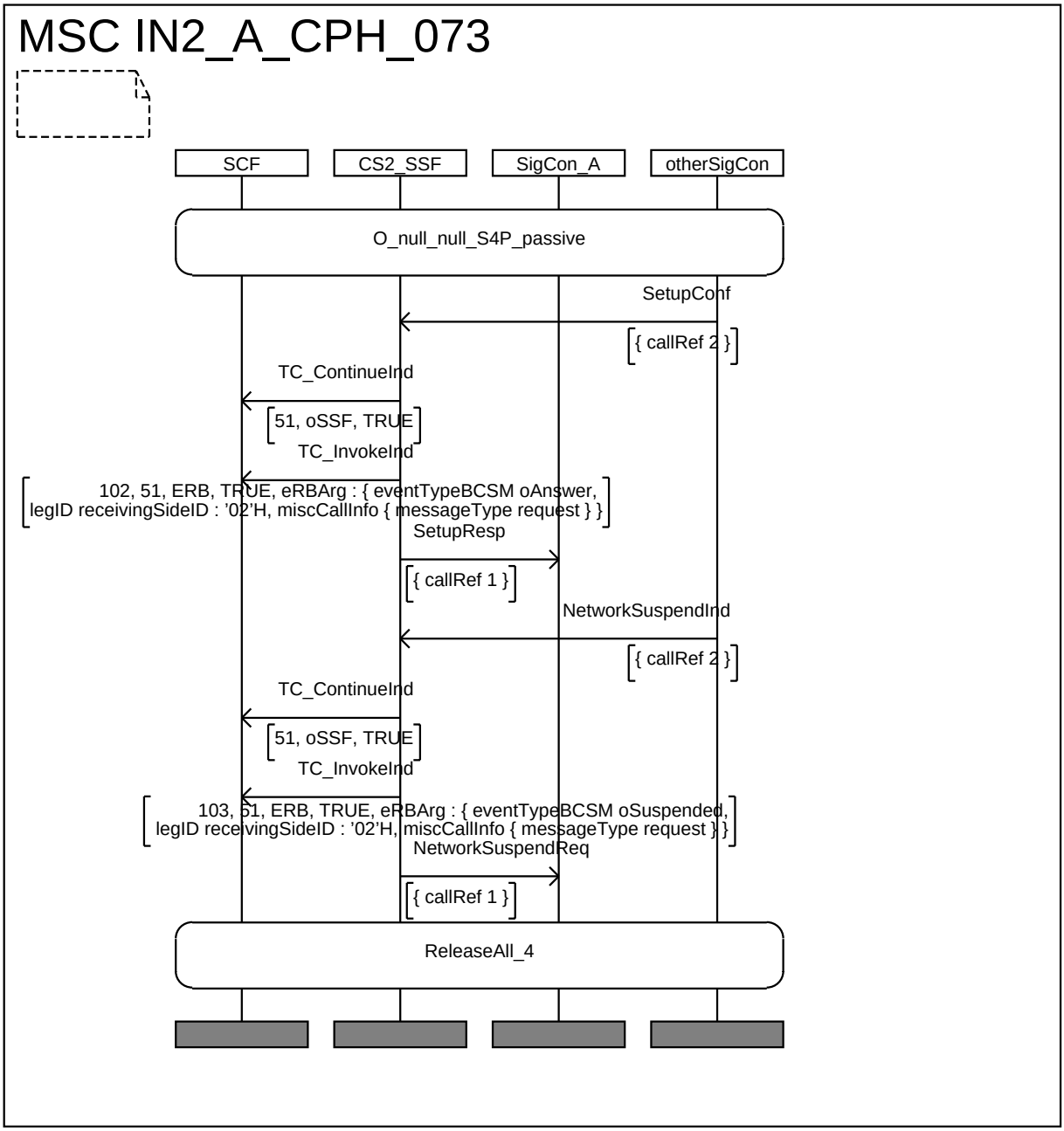
IN2_A_CPH_071

This test purpose is not included.

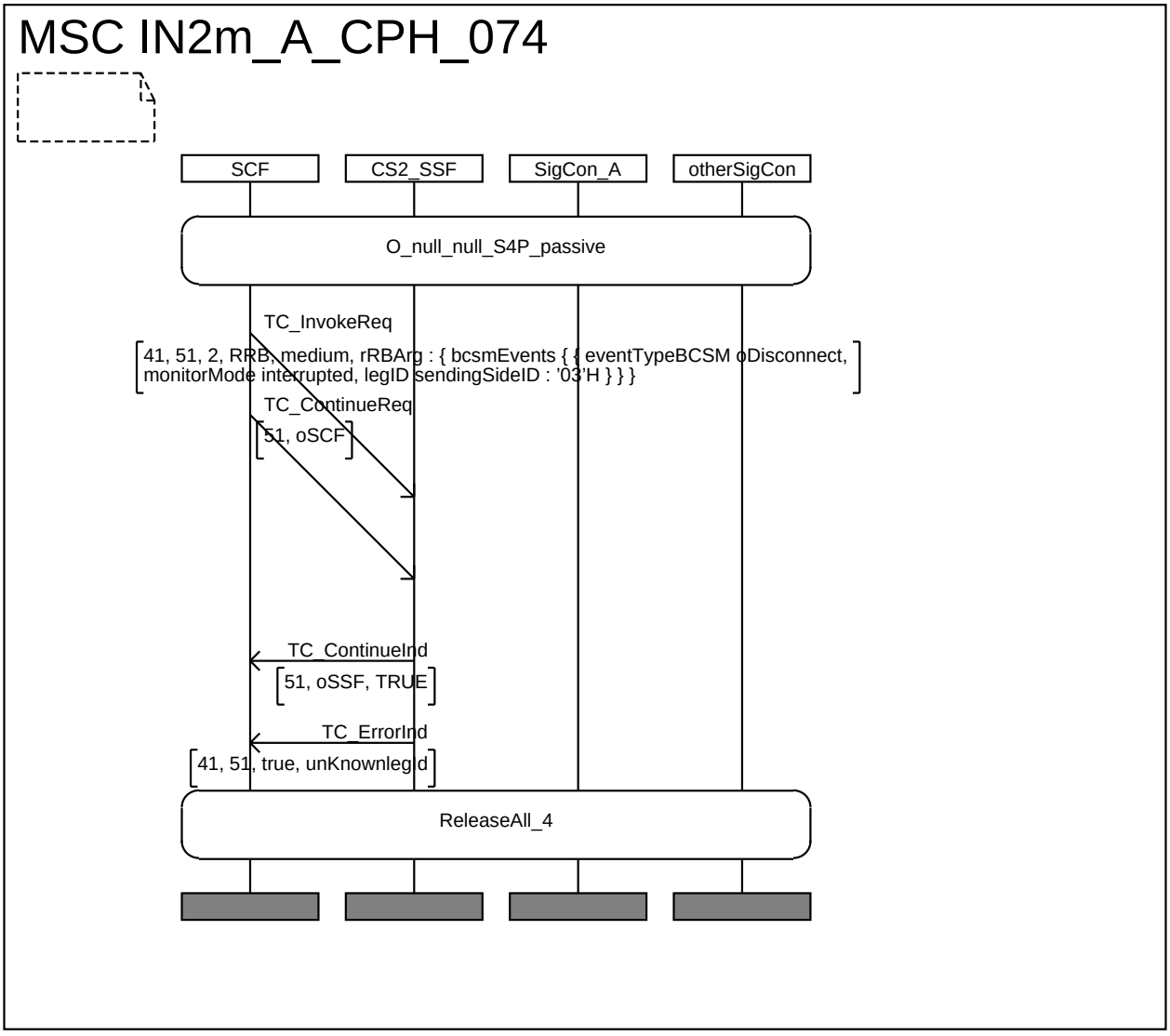
IN2_A_CPH_072

This test purpose is not included.

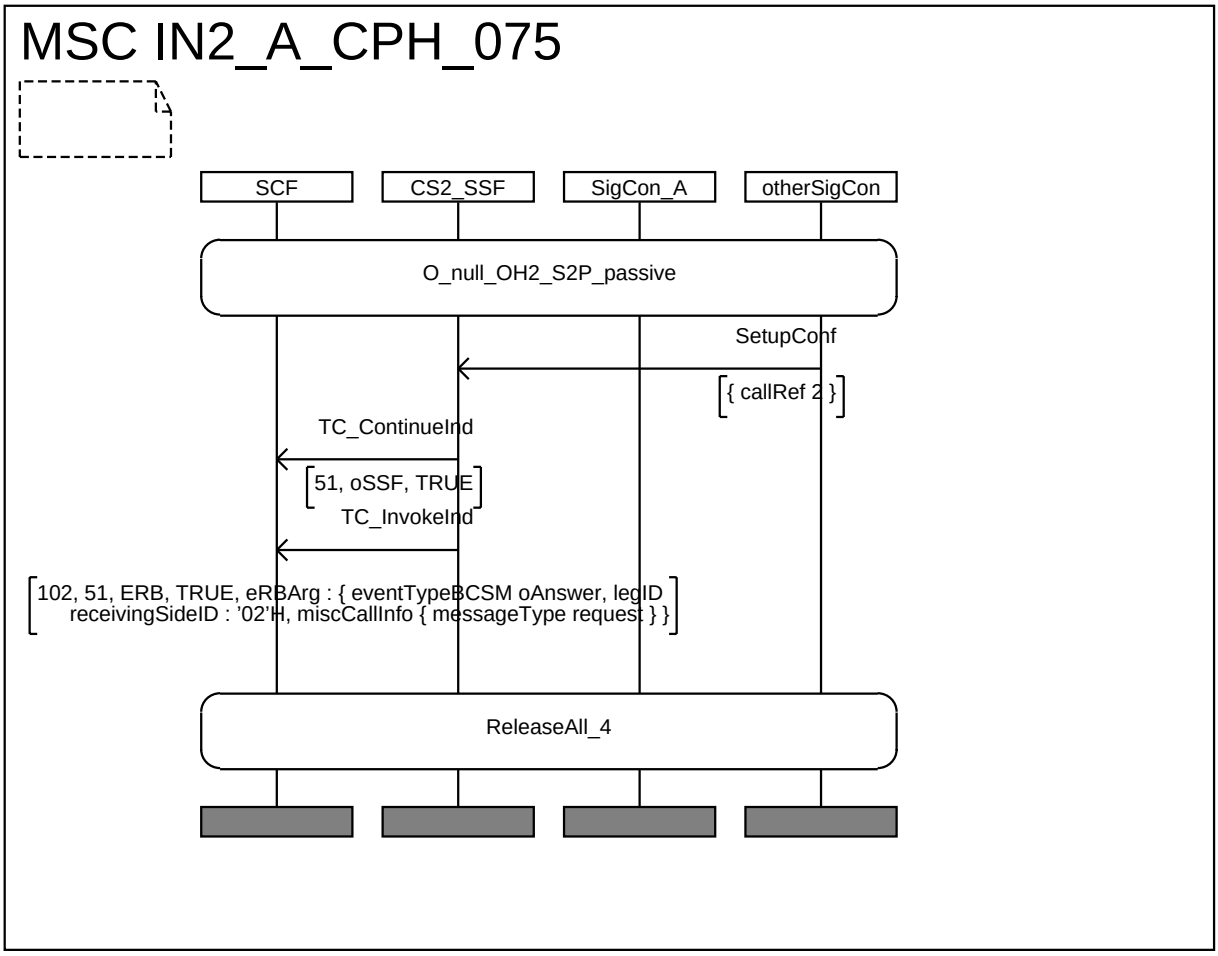
IN2_A_CPH_073	
Purpose:	Check that after two events are armed, then both are reported
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P_passive
Test description	CP1_2! SetupConf CP1_2! NetworkSuspendInd
Pass criteria	L1? ERB(2,notify, oAnswer) L1? ERB(2,notify, oSuspended)
Postamble:	ReleaseAll_4


MSC IN2_A_CPH_073
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P_passive
SetupConf
{ callRef 2 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer,
legID receivingSideID : '02'H, miscCallInfo { messageType request } }
SetupResp
{ callRef 1 }
NetworkSuspendInd
{ callRef 2 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
103, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oSuspended,
legID receivingSideID : '02'H, miscCallInfo { messageType request } }
NetworkSuspendReq
{ callRef 1 }
ReleaseAll_4

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



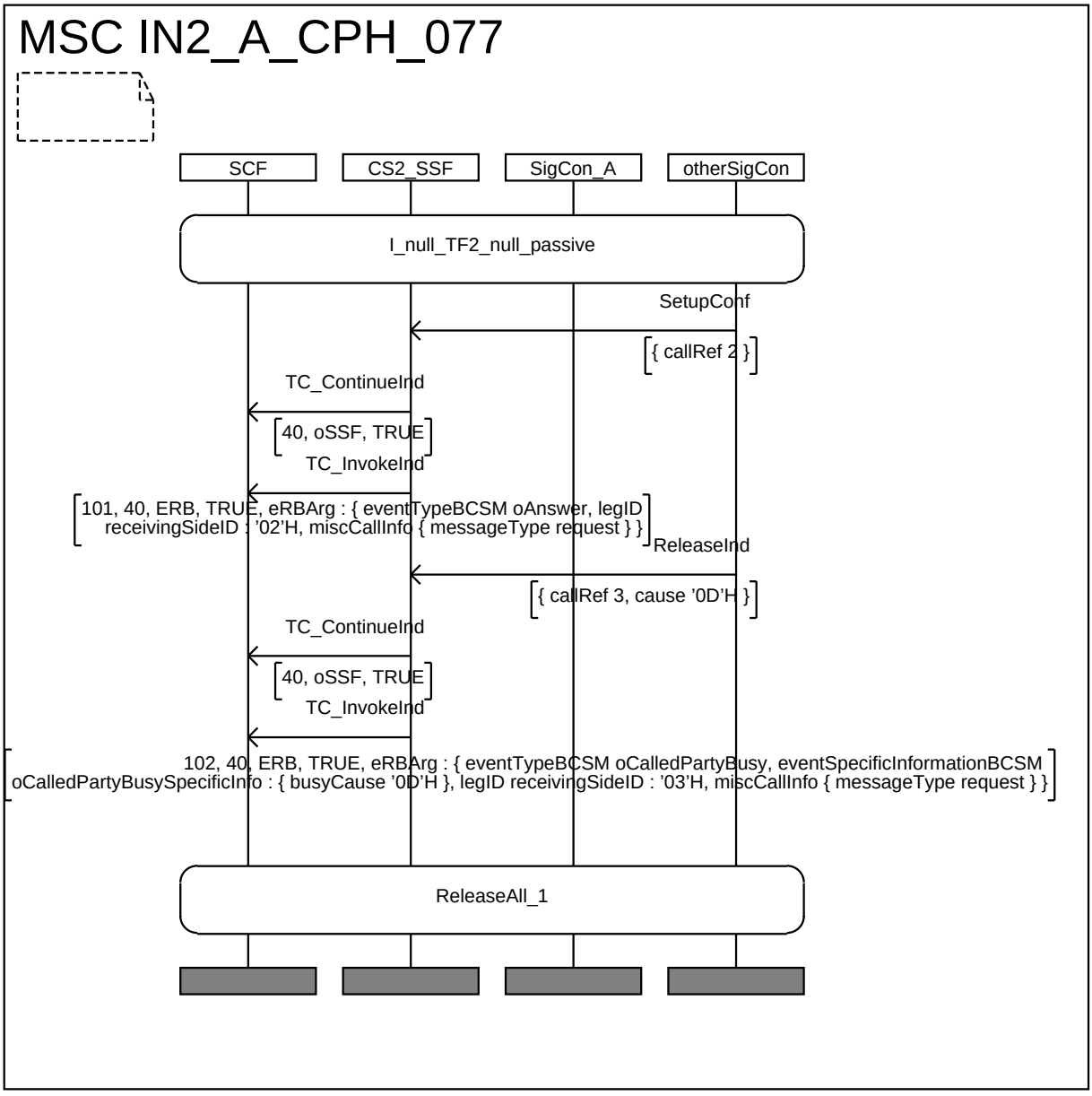
IN2_A_CPH_075	
Purpose:	Check that a single event oAnswer is detected on a passive leg
Requirement ref	
Selection Cond.	
Preamble:	O_null_OH(2)_S2P_passive
Test description	CP1_2! SetupConf
Pass criteria	L1? ERB(2, notify, oAnswer)
Postamble:	ReleaseAll_4



IN2_A_CPH_076

This test purpose is not included.

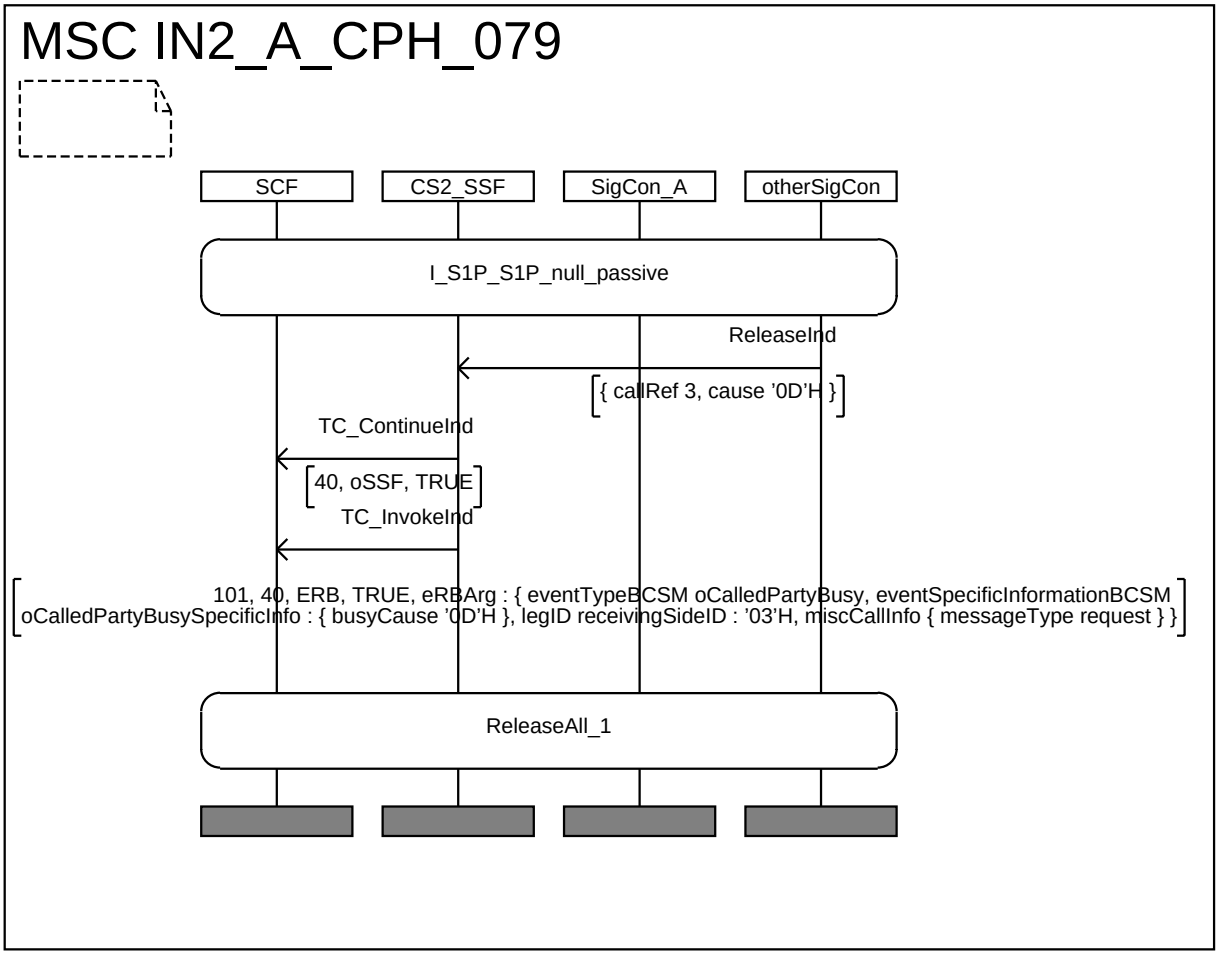
IN2_A_CPH_077	
Purpose:	Check that after two events are armed, then both are reported
Requirement ref	
Selection Cond.	
Preamble:	I_null_TF(2)_null_passive
Test description	CP1_2! SetUpConf CP1_3! ReleaseInd(Busy cause)
Pass criteria	L1? ERB(2,notify, oAnswer) L1? ERB(3,notify, oBusy)
Postamble:	ReleaseAll_1



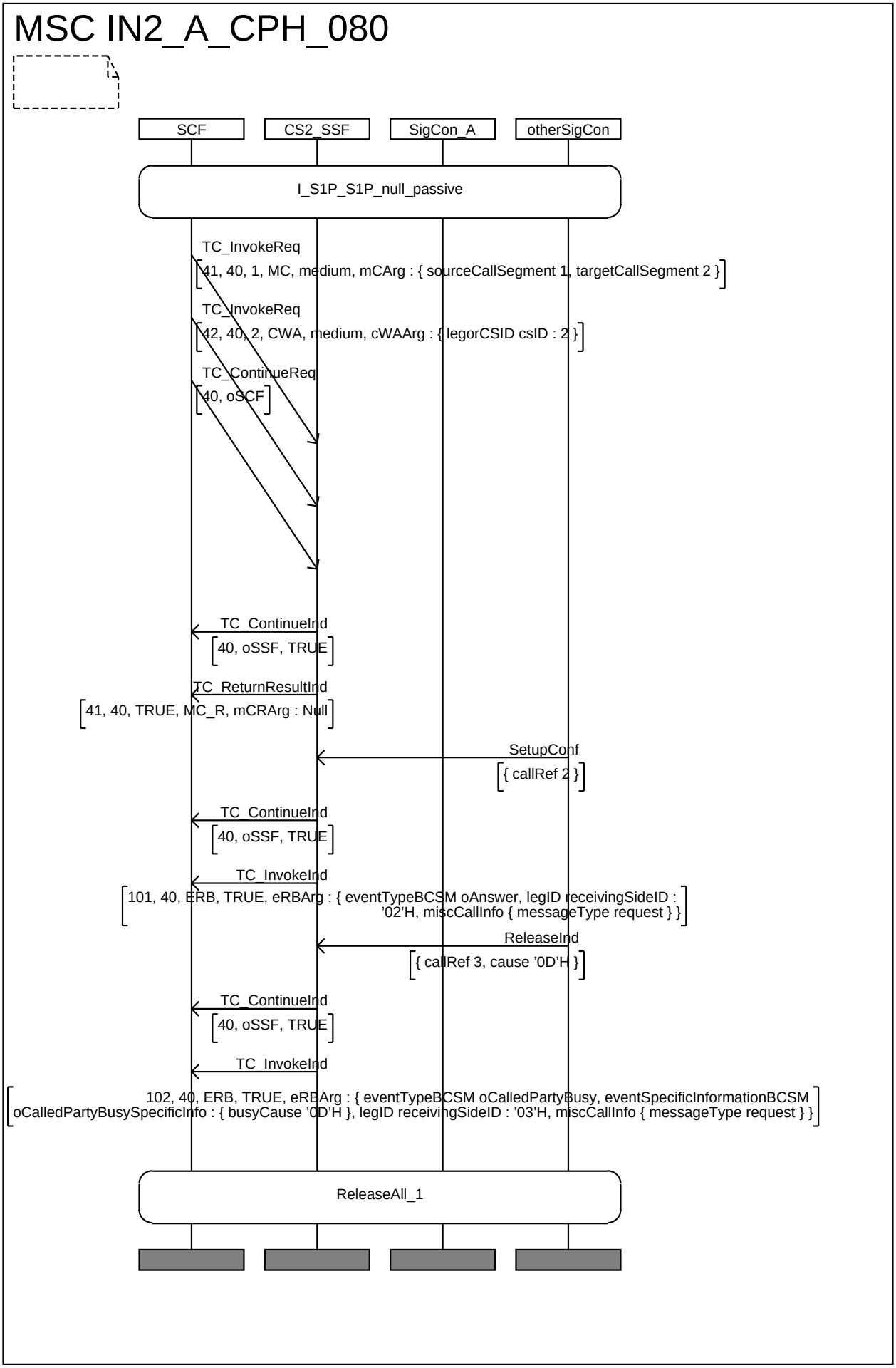
IN2_A_CPH_078

This test purpose is not included.

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



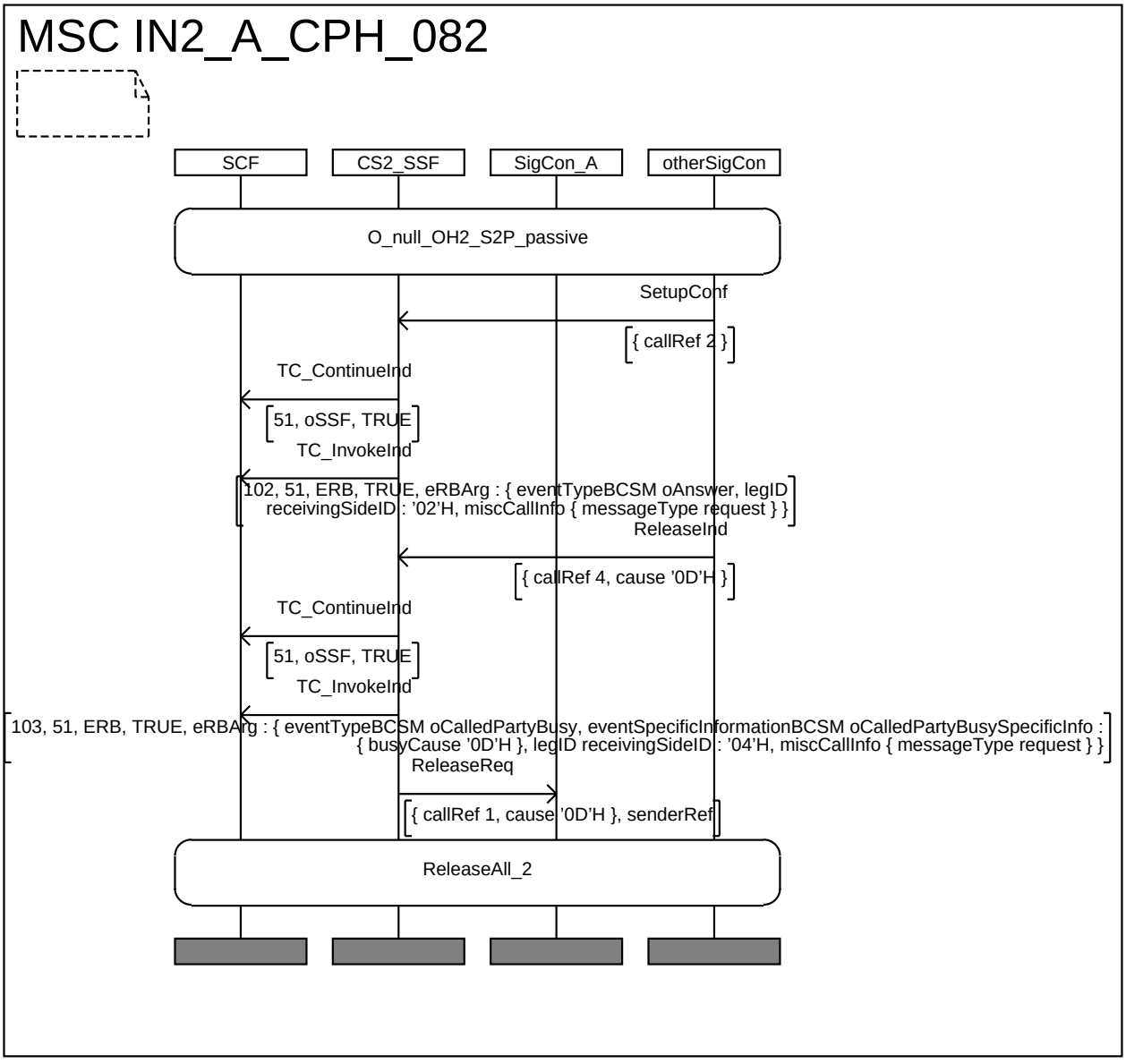
IN2_A_CPH_080	
Purpose:	Test that after a Merge Call segments operation the events armed remain the same for the passive legs.
Requirement ref	
Selection Cond.	
Preamble:	I_S1P_S1P_null_passive
Test description	L1! MergeCallSegments(1,2) L1? MergeCallSegmentReturnResult L1! ContinueWithArgument(CsId=2)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


MSC IN2_A_CPH_080
SCF
CS2_SSF
SigCon_A
otherSigCon
I_S1P_S1P_null_passive
TC_InvokeReq
41, 40, 1, MC, medium, mCArg : { sourceCallSegment 1, targetCallSegment 2 }
TC_InvokeReq
42, 40, 2, CWA, medium, cWAArg : { legorCSID csID : 2 }
TC_ContinueReq
40, oSCF
TC_ContinueInd
40, oSSF, TRUE
TC_ReturnResultInd
41, 40, TRUE, MC_R, mCRArg : Null
SetupConf
{ callRef 2 }
TC_ContinueInd
40, oSSF, TRUE
TC_InvokeInd
101, 40, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '02'H, miscCallInfo { messageType request } }
ReleaseInd
{ callRef 3, cause '0D'H }
TC_ContinueInd
40, oSSF, TRUE
TC_InvokeInd
102, 40, ERB, TRUE, eRBArg : { eventTypeBCSM oCalledPartyBusy, eventSpecificInformationBCSM oCalledPartyBusySpecificInfo : { busyCause '0D'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_1

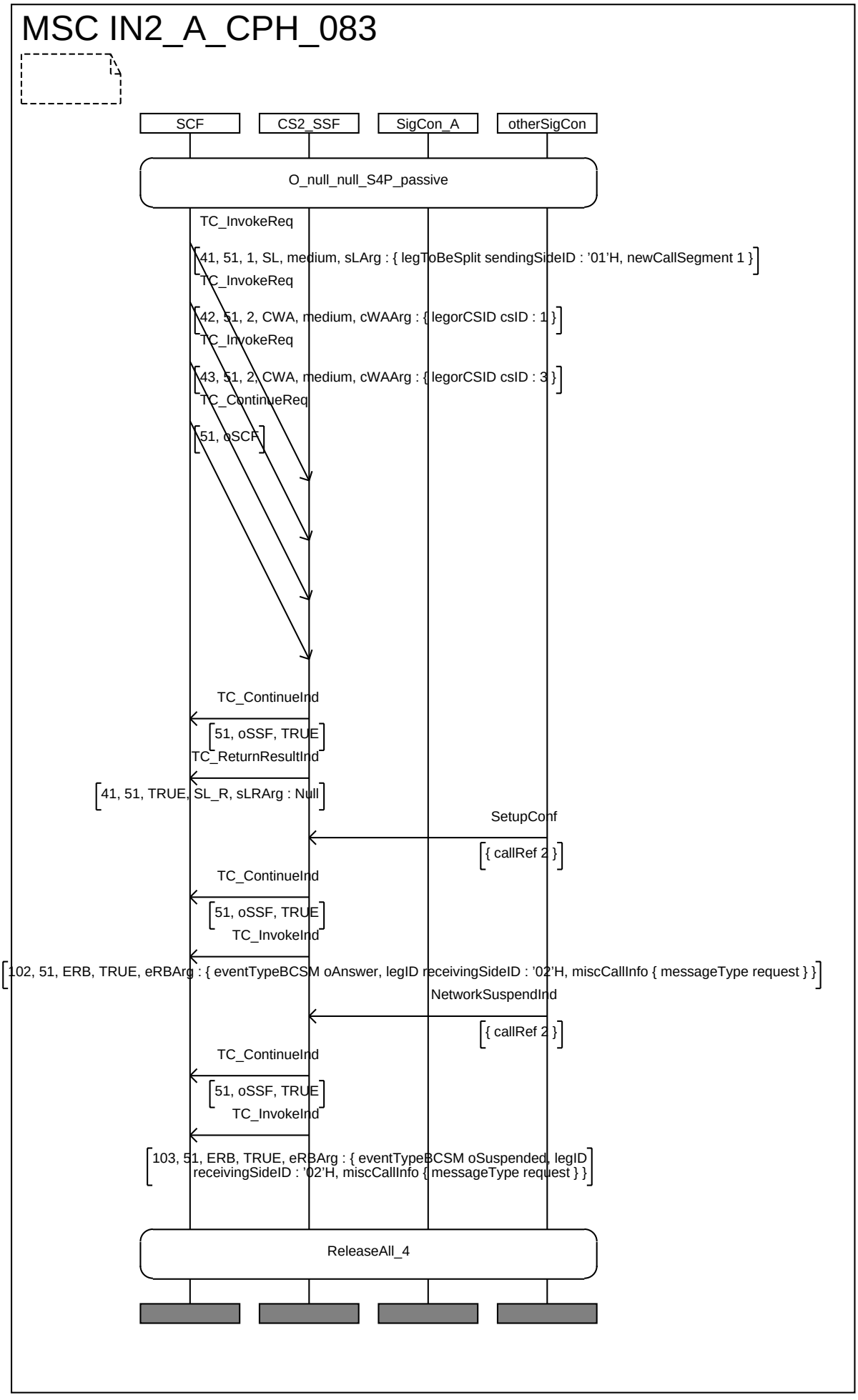
IN2_A_CPH_081

This test purpose is not included.

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



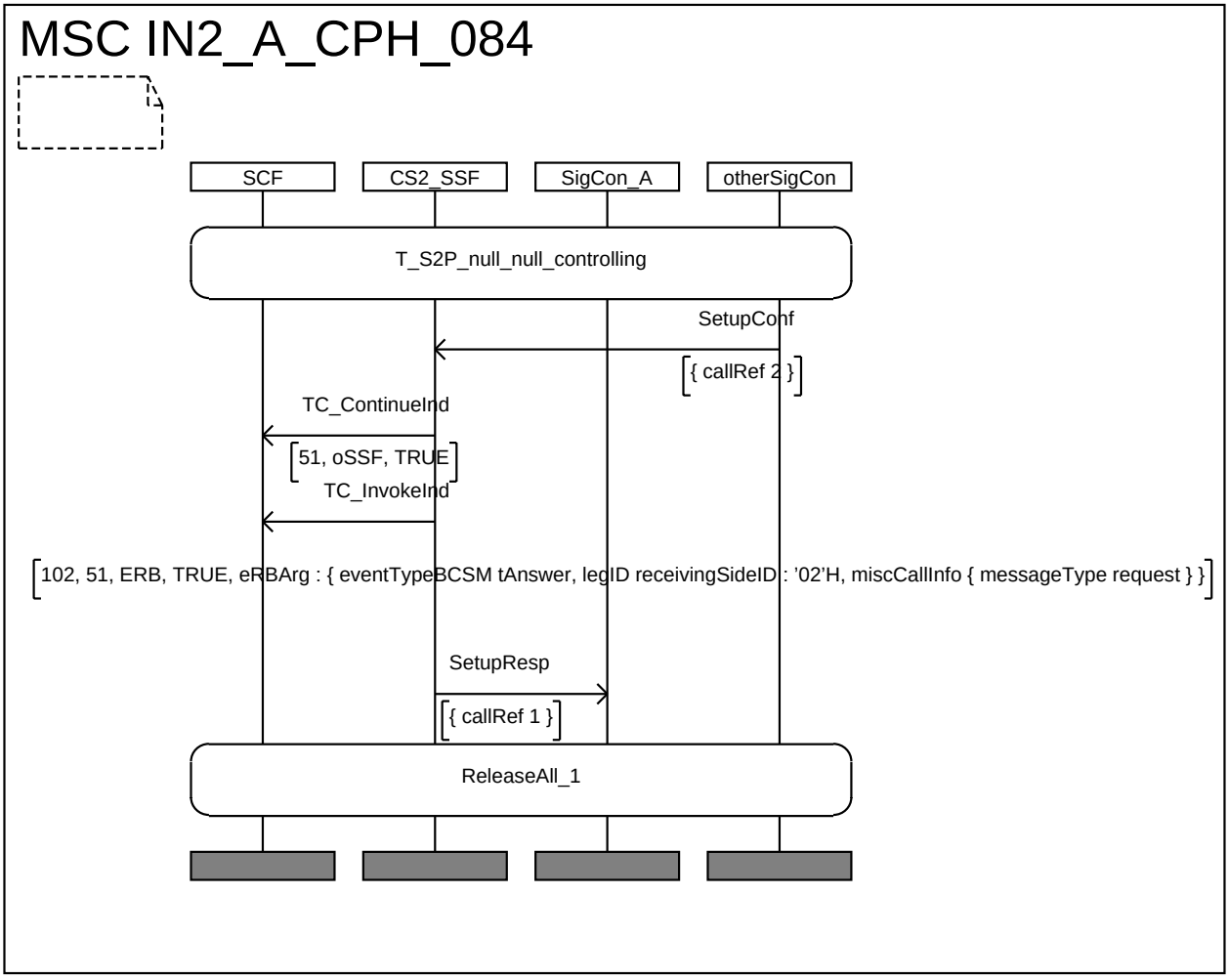
IN2_A_CPH_083	
Purpose:	Test that after a Split leg operation is sent the armed events remain the same
Requirement ref	
Selection Cond.	
Preamble:	O_null_null_S4P_passive
Test description	L1! SplitLeg(1,1) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! ContinueWithArgument(CsId=3) CP1_2!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


MSC IN2_A_CPH_083
SCF
CS2_SSF
SigCon_A
otherSigCon
O_null_null_S4P_passive
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '01'H, newCallSegment 1 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_InvokeReq
43, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
SetupConf
{ callRef 2 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '02'H, miscCallInfo { messageType request } }
NetworkSuspendInd
{ callRef 2 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
103, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oSuspended, legID receivingSideID : '02'H, miscCallInfo { messageType request } }
ReleaseAll_4

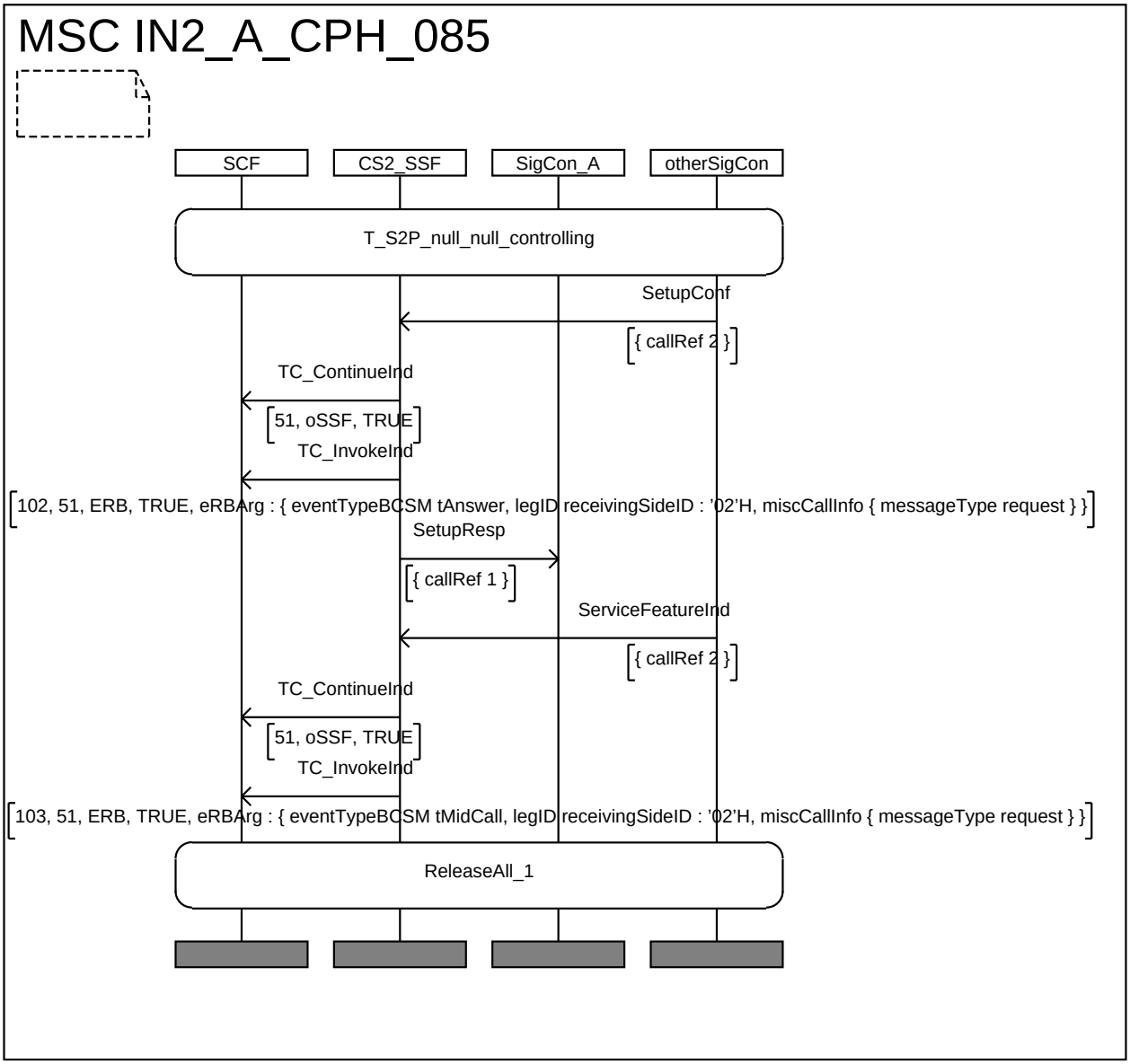
7.6.2 Terminating (T) trigger

7.6.2.1 T_1 Events coming from the controlling leg (legId=2),

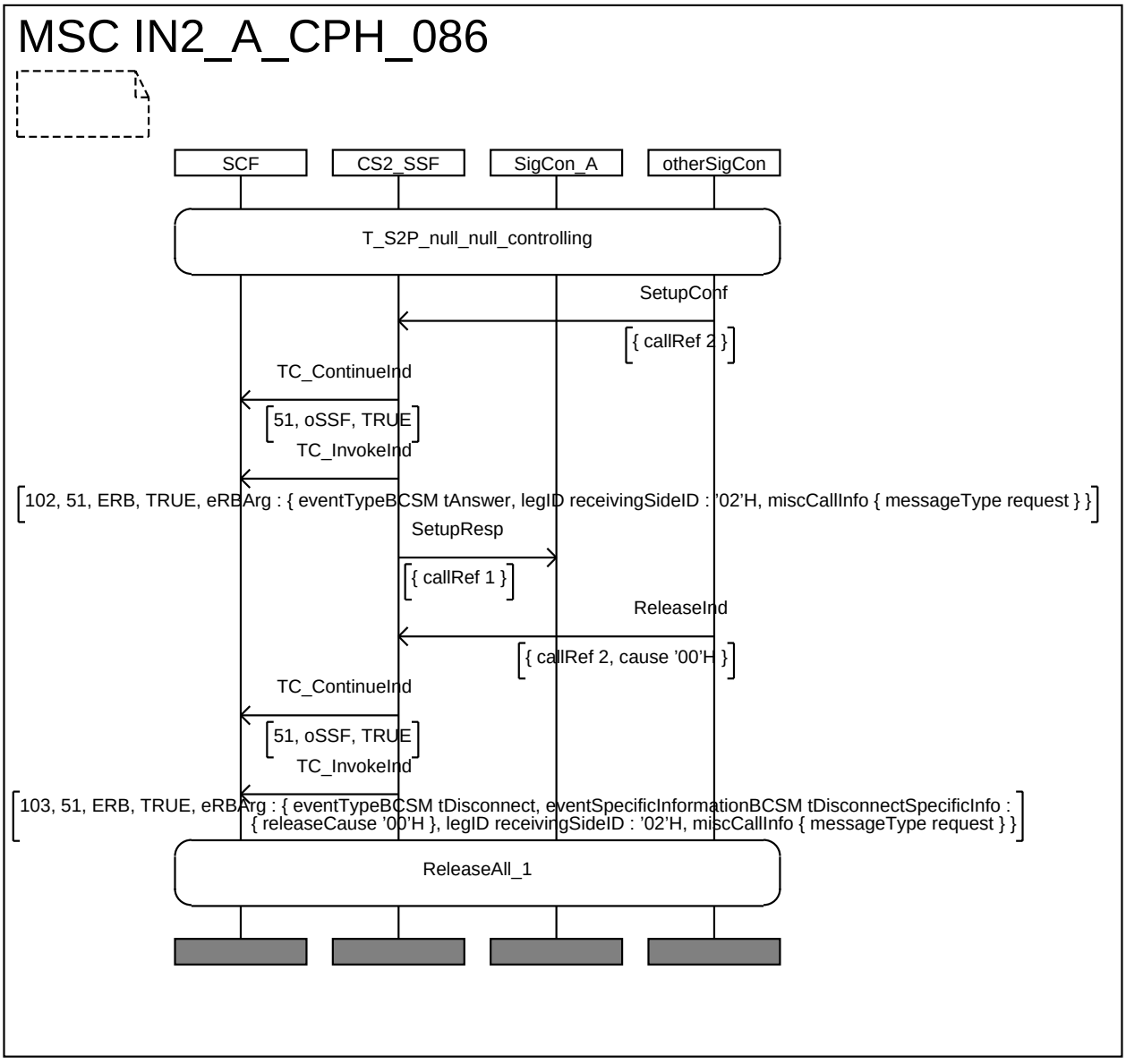
	IN2_A_CPH_084
Purpose:	Check that an event coming from the controlling leg is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_controlling
Test description	CP1_2! SetupConf
Pass criteria	L1?: ERB (tAnswer) is received only once (filtering rules apply)
Postamble:	ReleaseAll_1



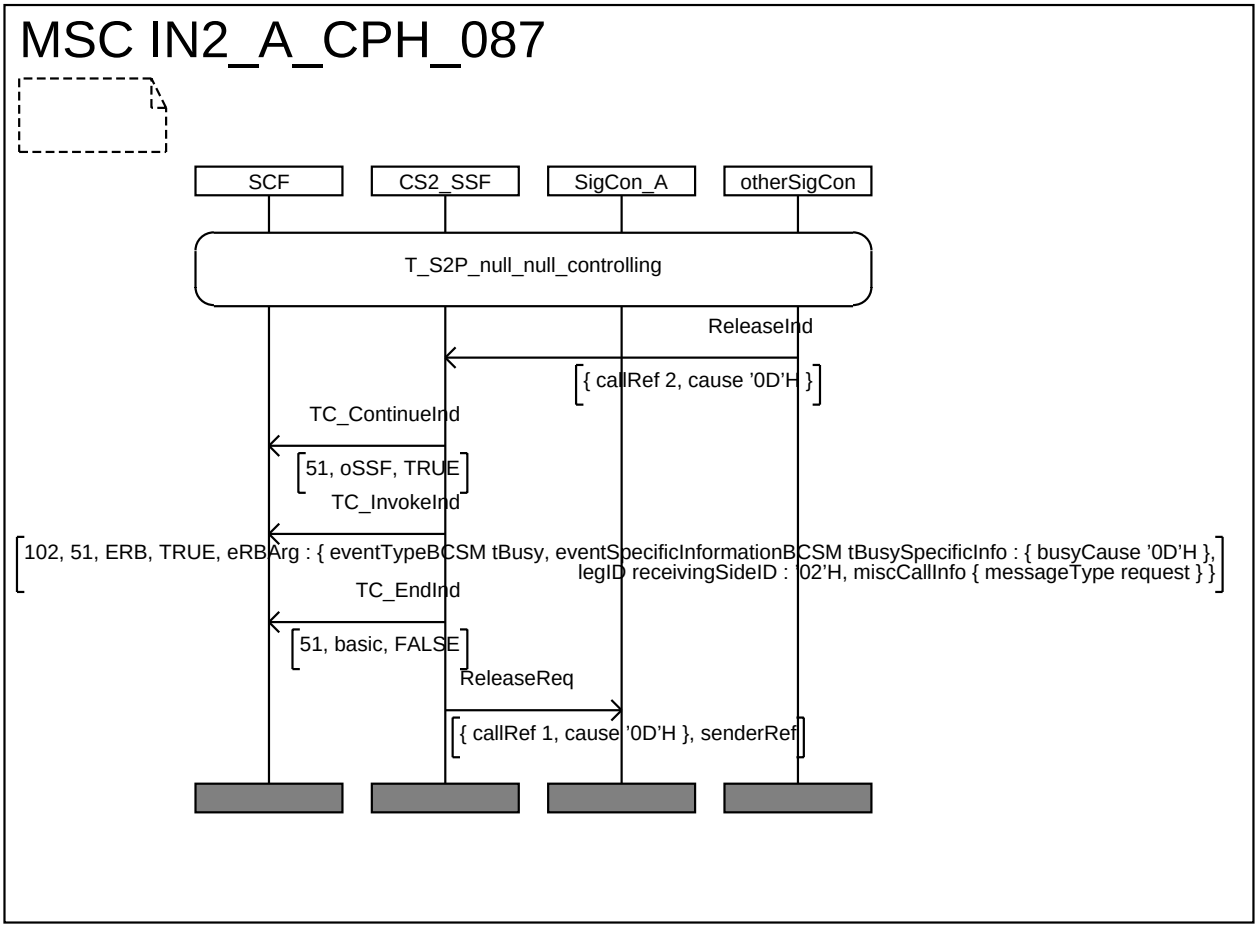
IN2_A_CPH_085	
Purpose:	Check that an event coming from the controlling leg is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_controlling
Test description	CP1_2!:SetupConf CP1_2!:ServiceFeatureIndication
Pass criteria	L1?: ERB (tAnswer**)** L1?: ERB (tMidCall)
Postamble:	ReleaseAll_1



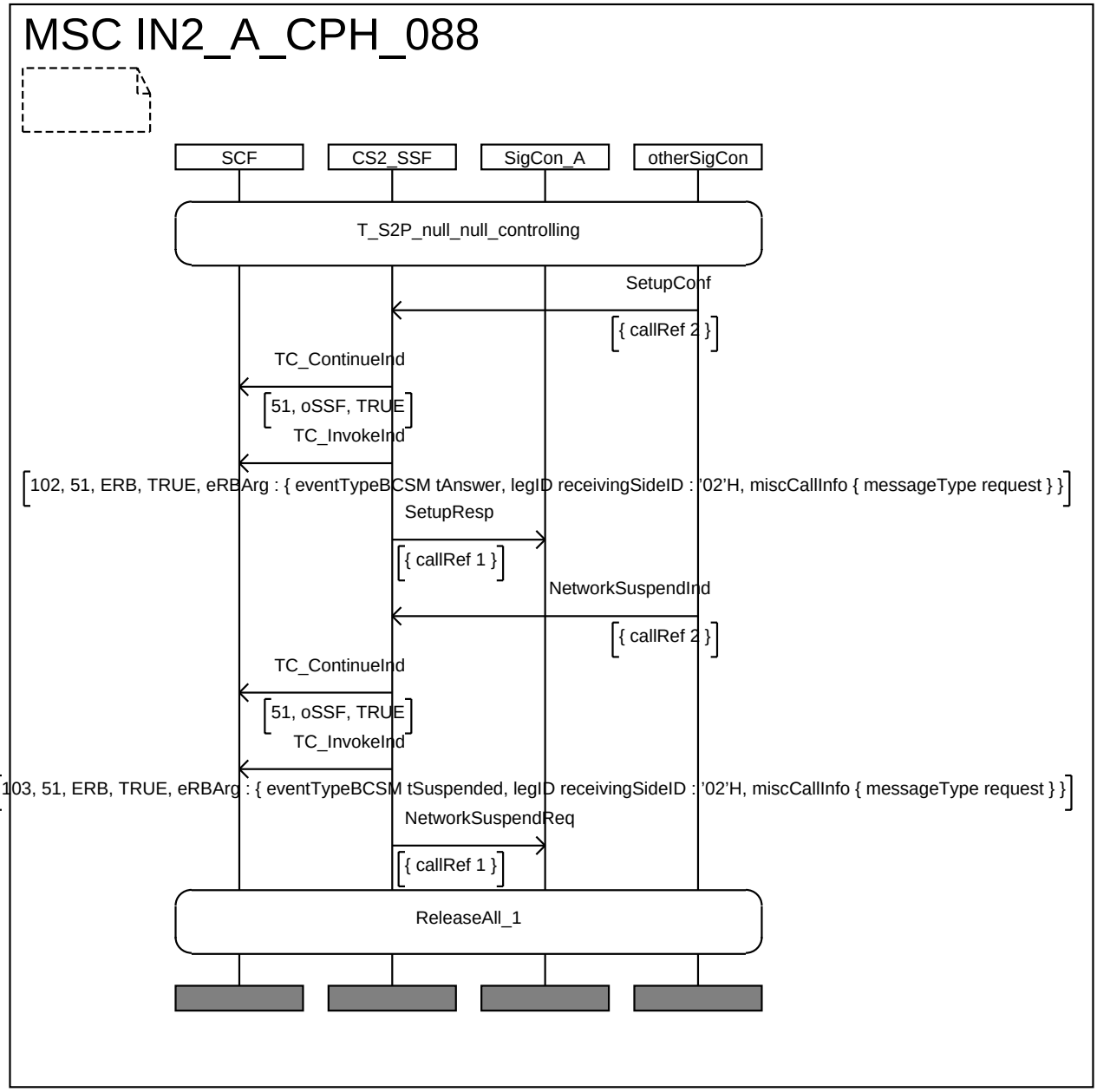
IN2_A_CPH_086	
Purpose:	Check that an event coming from the controlling leg is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_controlling
Test description	CP1_2!: SetupConf CP1_2: ReleaseInd
Pass criteria	L1?: ERB (tAnswer) L1?: ERB(tDisconnect)
Postamble:	ReleaseAll_1



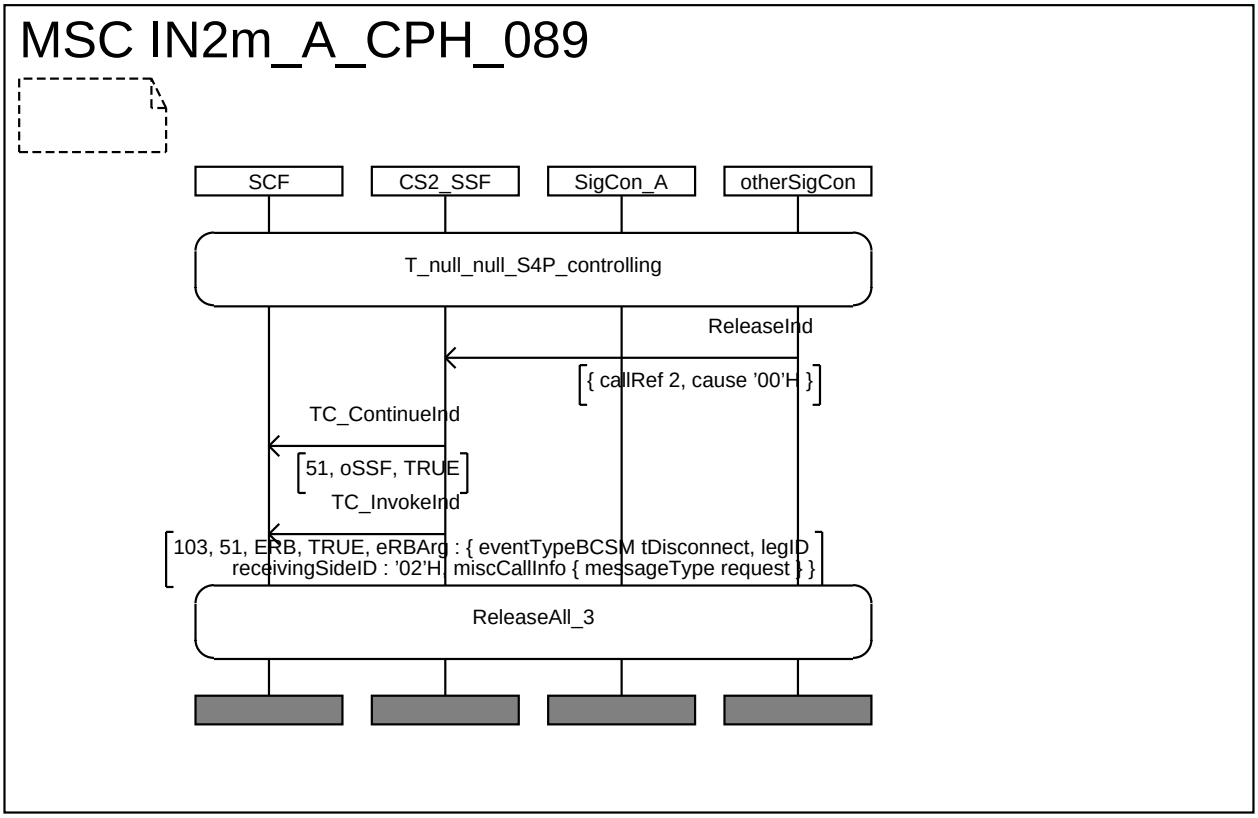
IN2_A_CPH_087	
Purpose:	Check that an event coming from the controlling leg is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_controlling
Test description	CP1_2!: ReleaseIndication(busy)
Pass criteria	L1?: ERB (tBusy))
Postamble:	none



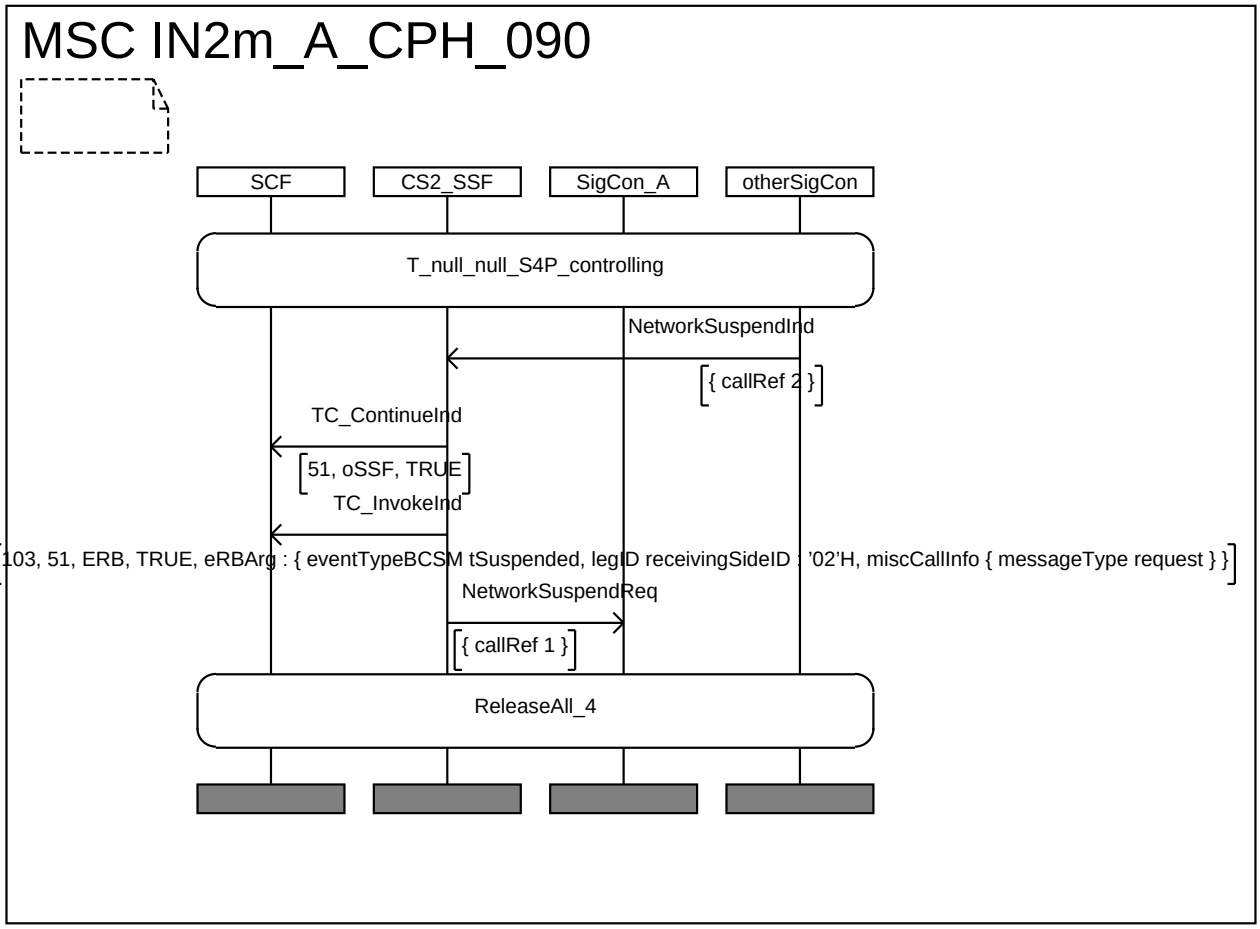
IN2_A_CPH_088	
Purpose:	Check that an event coming from the controlling leg is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_controlling
Test description	CP1_2!: SetupConf CP1_2!: SuspendIndication
Pass criteria	L1?: ERB (tAnswer) L1?: ERB(tSuspended)
Postamble:	ReleaseAll_1



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



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

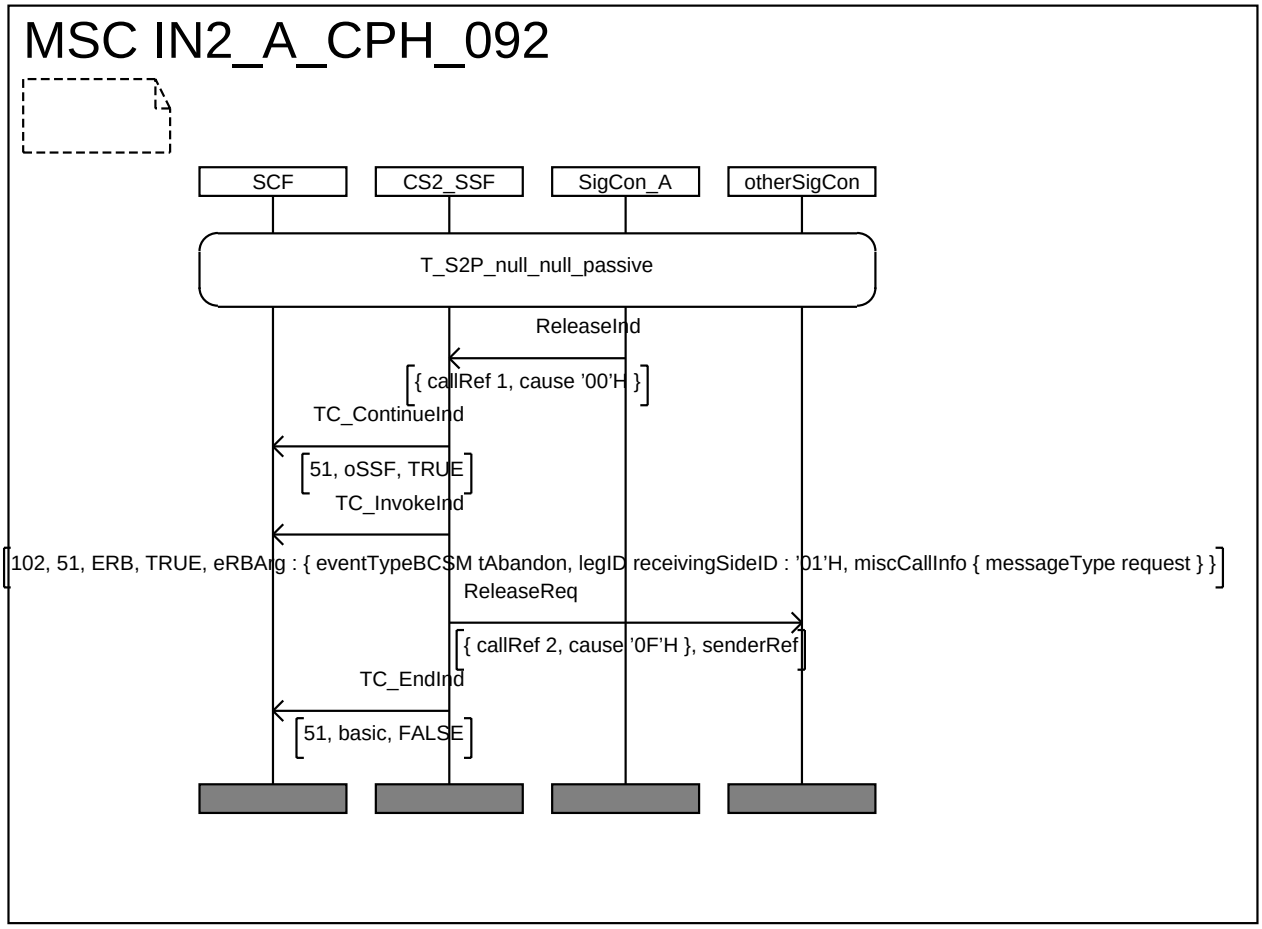


IN2_A_CPH_091

This test purpose is not included.

7.6.2.2 T_2 Events coming from passive legs (legId= 2,3...)

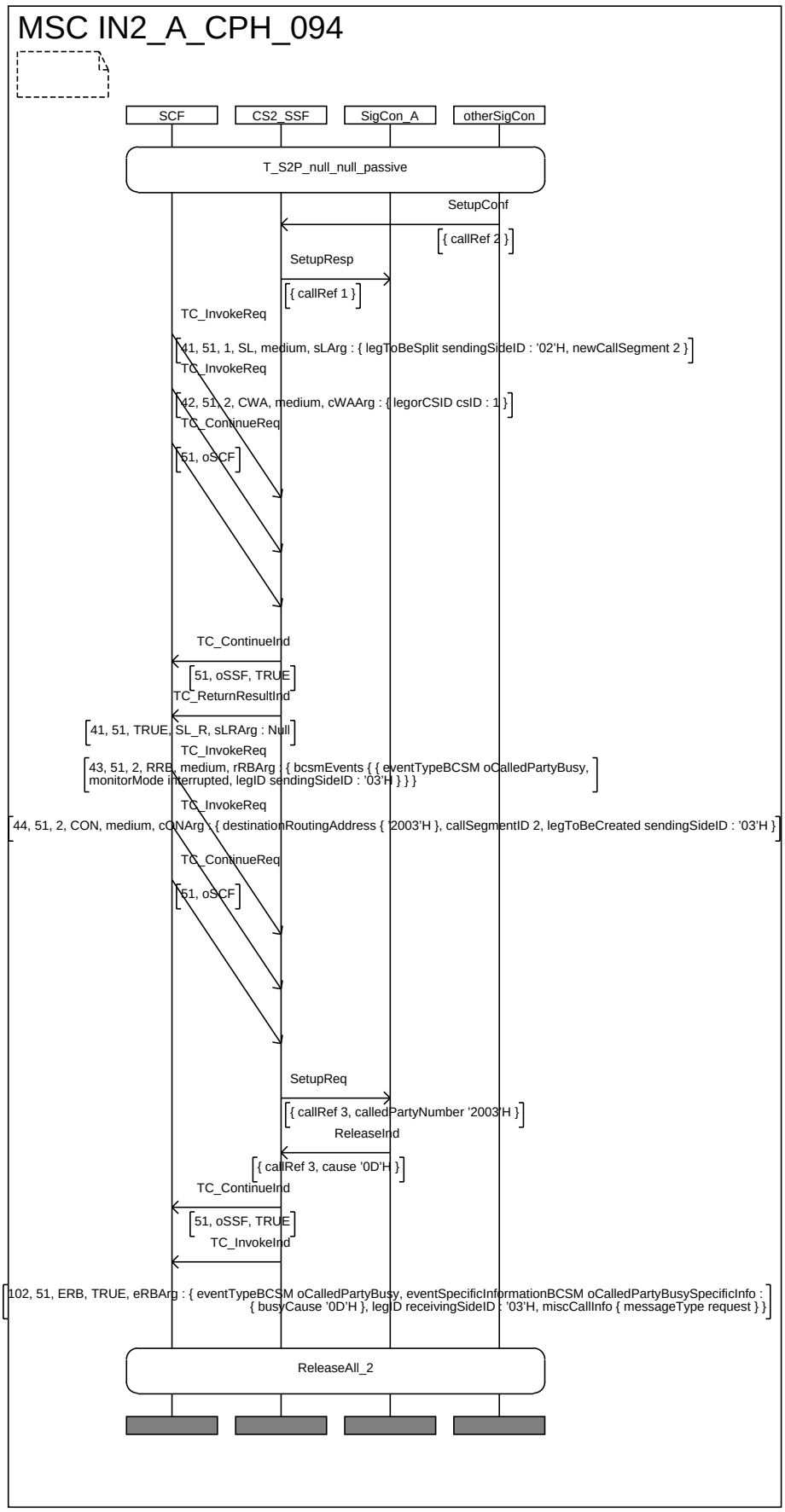
IN2_A_CPH_092	
Purpose:	Check that a single event tAbandon is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_passive
Test description	CP1_1! ReleaseInd
Pass criteria	L1? ERB(1,interrupted, tAbandon)
Postamble:	none



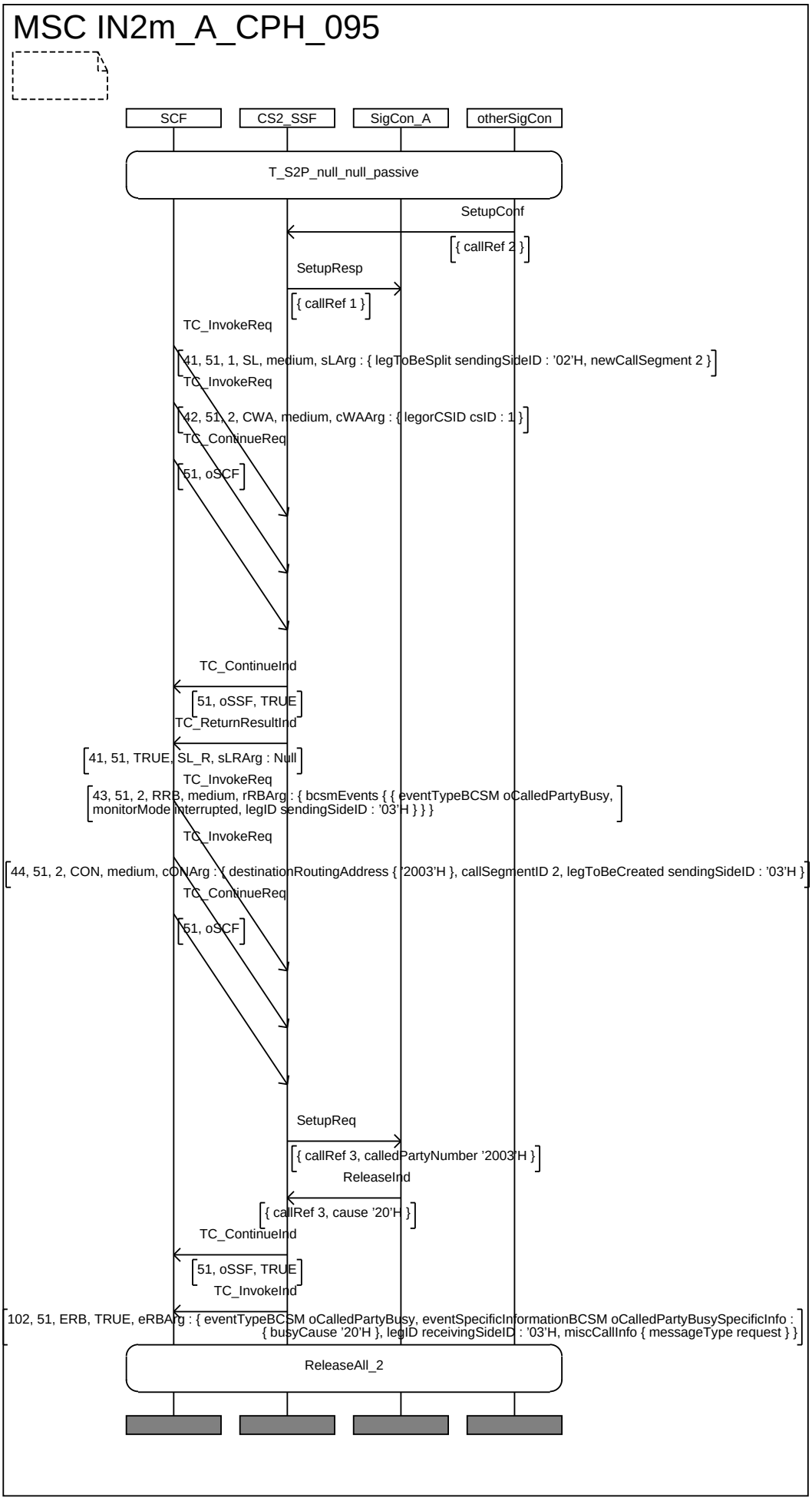
IN2_A_CPH_093

This test purpose is not included.

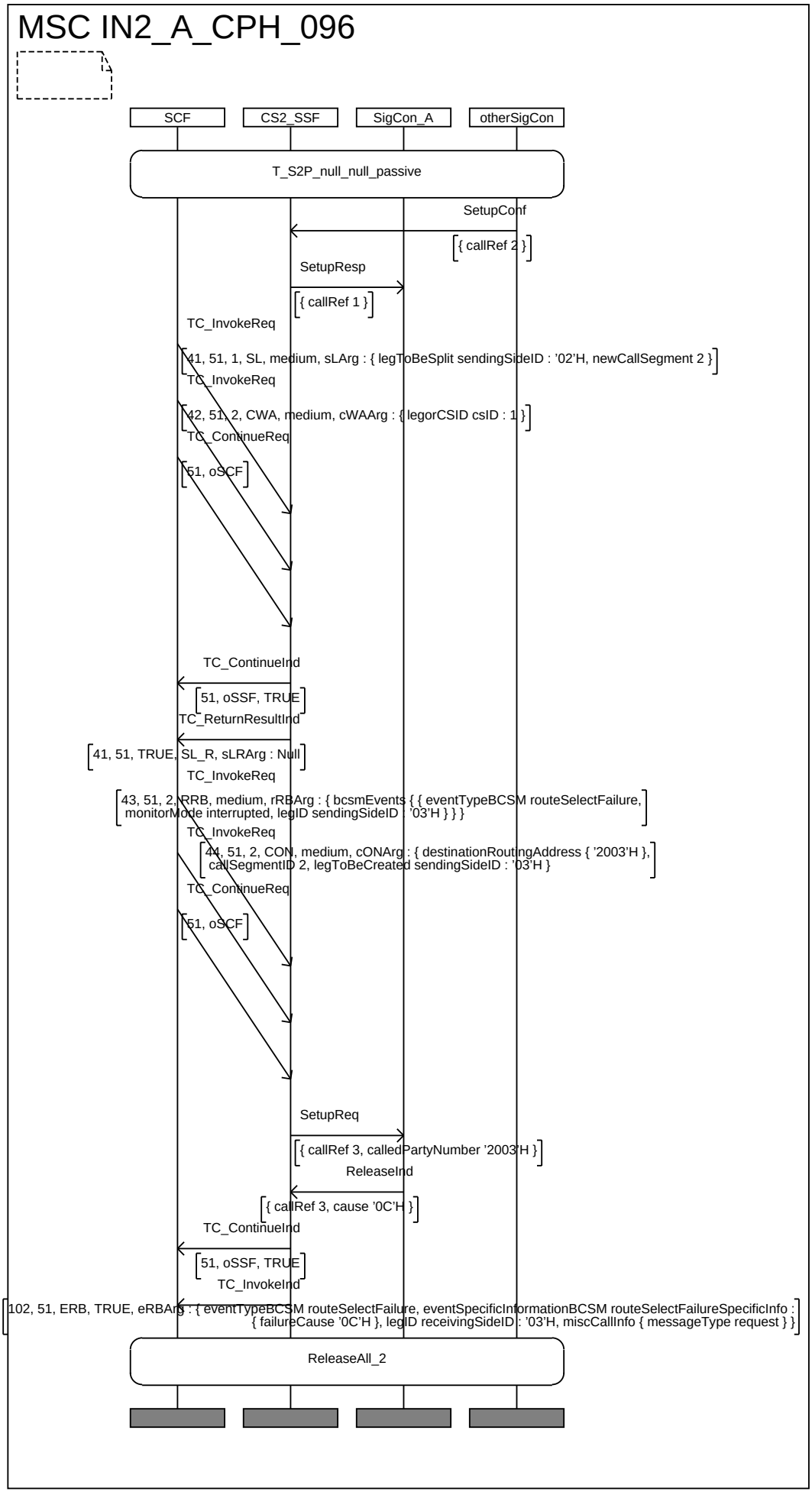
IN2_A_CPH_094	
Purpose:	Check that a single event oCalledpartyBusy is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_passive
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oCalledPartyBusy) L1! Connect(3,2) CP1_3?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


MSC IN2_A_CPH_094
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null_passive
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oCalledPartyBusy, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseInd
{ callRef 3, cause '0D'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oCalledPartyBusy, eventSpecificInformationBCSM oCalledPartyBusySpecificInfo : { busyCause '0D'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

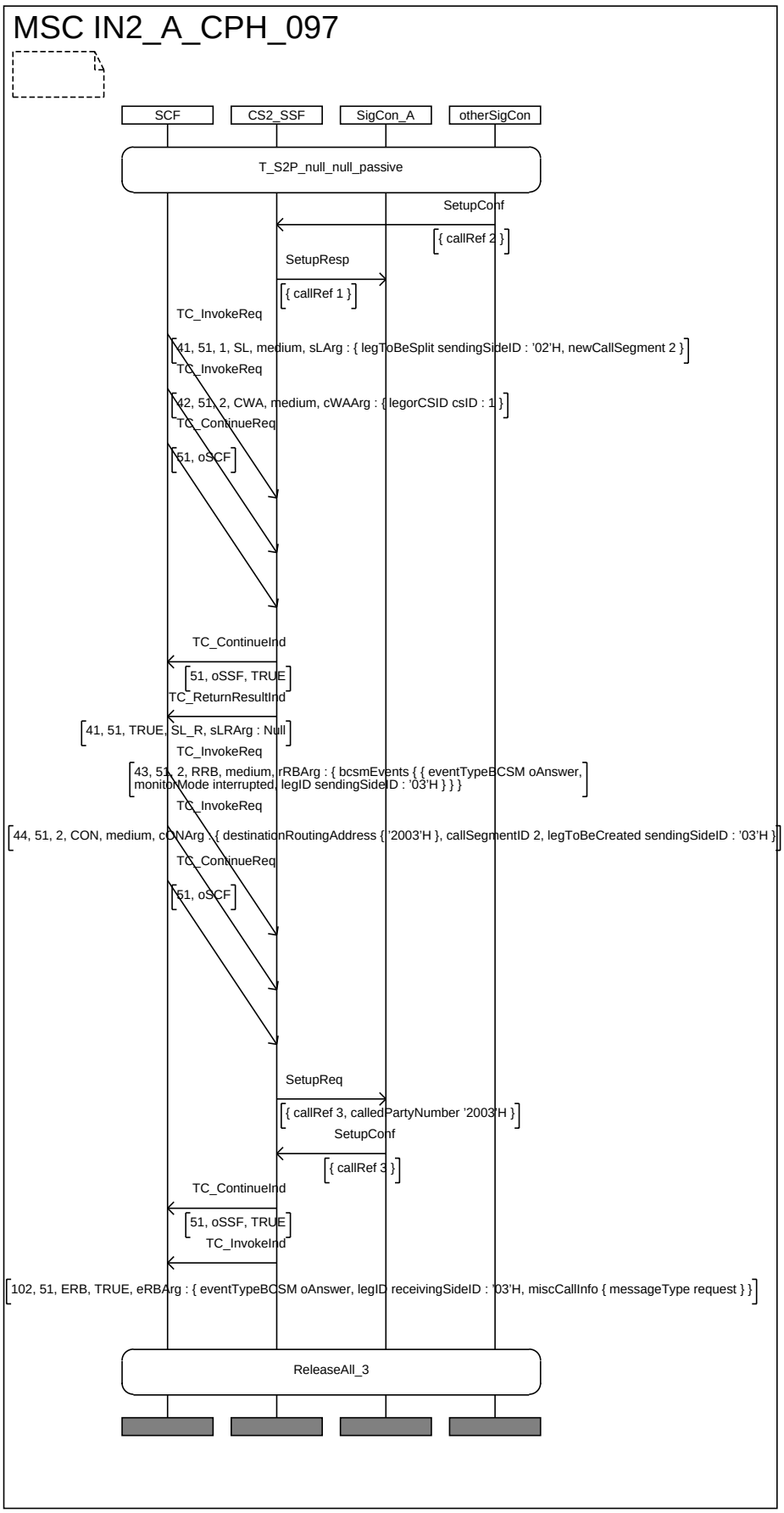
IN2_A_CPH_095	
Purpose:	Check that a single event oCalledPartyBusy is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_passive
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oCalledPartyBusy) L1! Connect(3,2) CP1_3?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


MSC IN2m_A_CPH_095
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null_passive
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oCalledPartyBusy, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseInd
{ callRef 3, cause '20'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oCalledPartyBusy, eventSpecificInformationBCSM oCalledPartyBusySpecificInfo : { busyCause '20'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

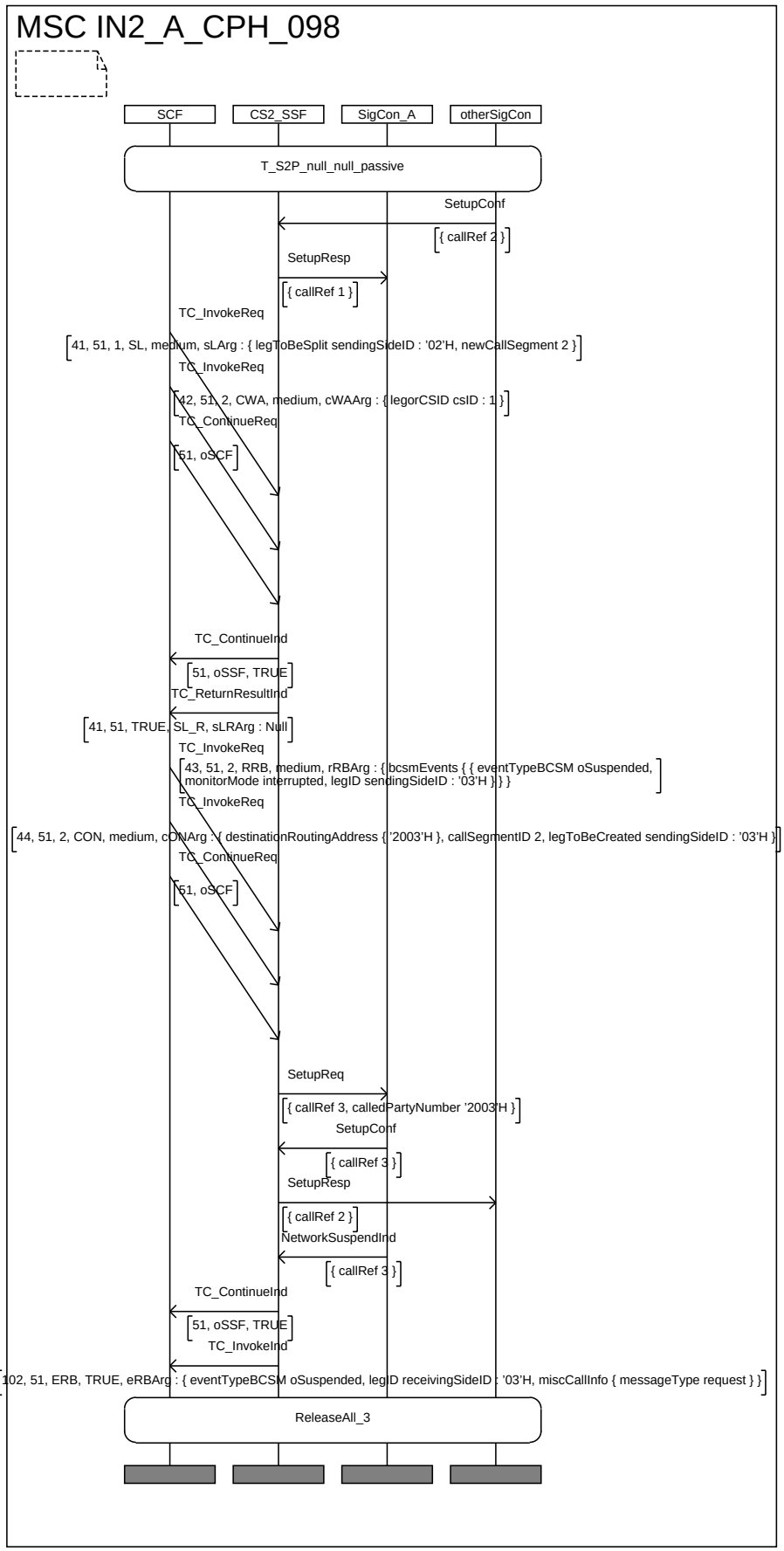
IN2_A_CPH_096	
Purpose:	Check that a single event routeSelectFailure is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_passive
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oRouteSelectFailure) L1! Connect(3,2)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


MSC IN2_A_CPH_096
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null_passive
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM routeSelectFailure, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
ReleaseInd
{ callRef 3, cause '0C'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM routeSelectFailure, eventSpecificInformationBCSM routeSelectFailureSpecificInfo : { failureCause '0C'H }, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_2

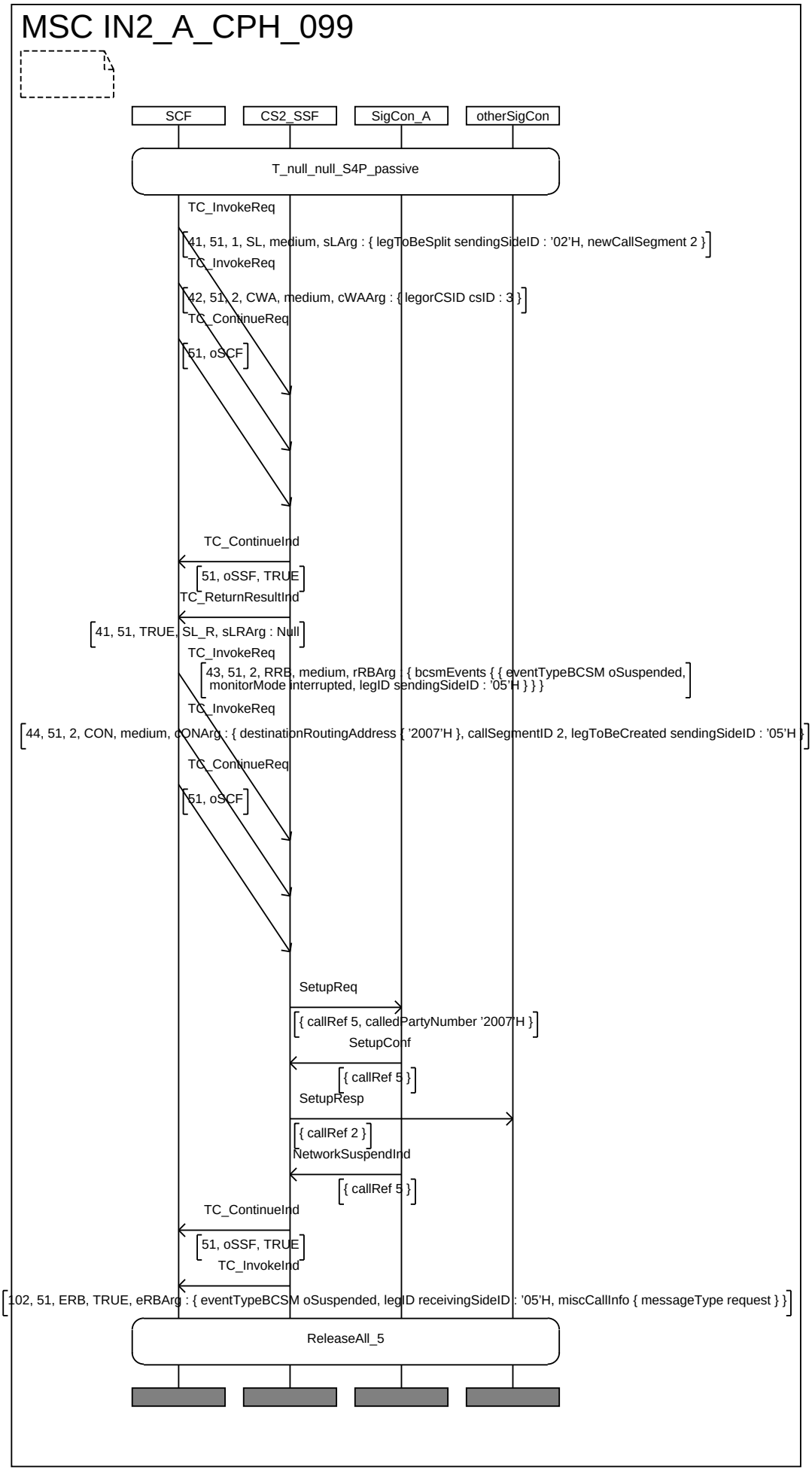
IN2_A_CPH_097	
Purpose:	Check that a single event oAnswer is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_passive
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oAnswer) L1! Connect(3,2) CP1_3? SetUpReq CP1_3!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


MSC IN2_A_CPH_097
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null_passive
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oAnswer, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
SetupConf
{ callRef 3 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_3

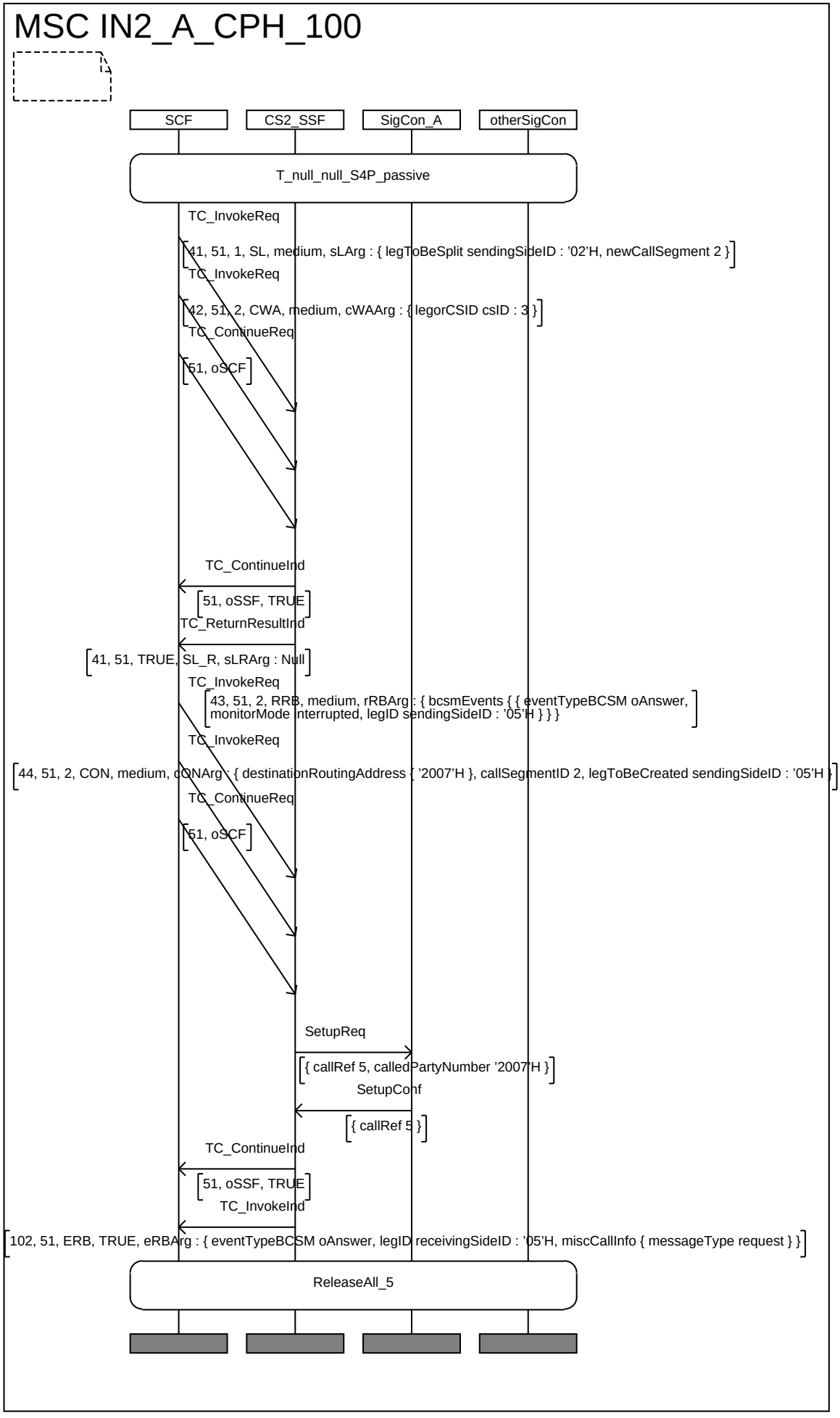
IN2_A_CPH_098	
Purpose:	Check that a single event oSuspended is reported
Requirement ref	
Selection Cond.	
Preamble:	T_S2P_null_null_passive
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=1) L1! RRB(3, interrupted, oSuspend) L1! Connect(3,2) CP1_3? SetUpReq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


MSC IN2_A_CPH_098
SCF
CS2_SSF
SigCon_A
otherSigCon
T_S2P_null_null_passive
SetupConf
{ callRef 2 }
SetupResp
{ callRef 1 }
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 1 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oSuspended, monitorMode interrupted, legID sendingSideID : '03'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress {'2003'H }, callSegmentID 2, legToBeCreated sendingSideID : '03'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 3, calledPartyNumber '2003'H }
SetupConf
{ callRef 3 }
SetupResp
{ callRef 2 }
NetworkSuspendInd
{ callRef 3 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oSuspended, legID receivingSideID : '03'H, miscCallInfo { messageType request } }
ReleaseAll_3

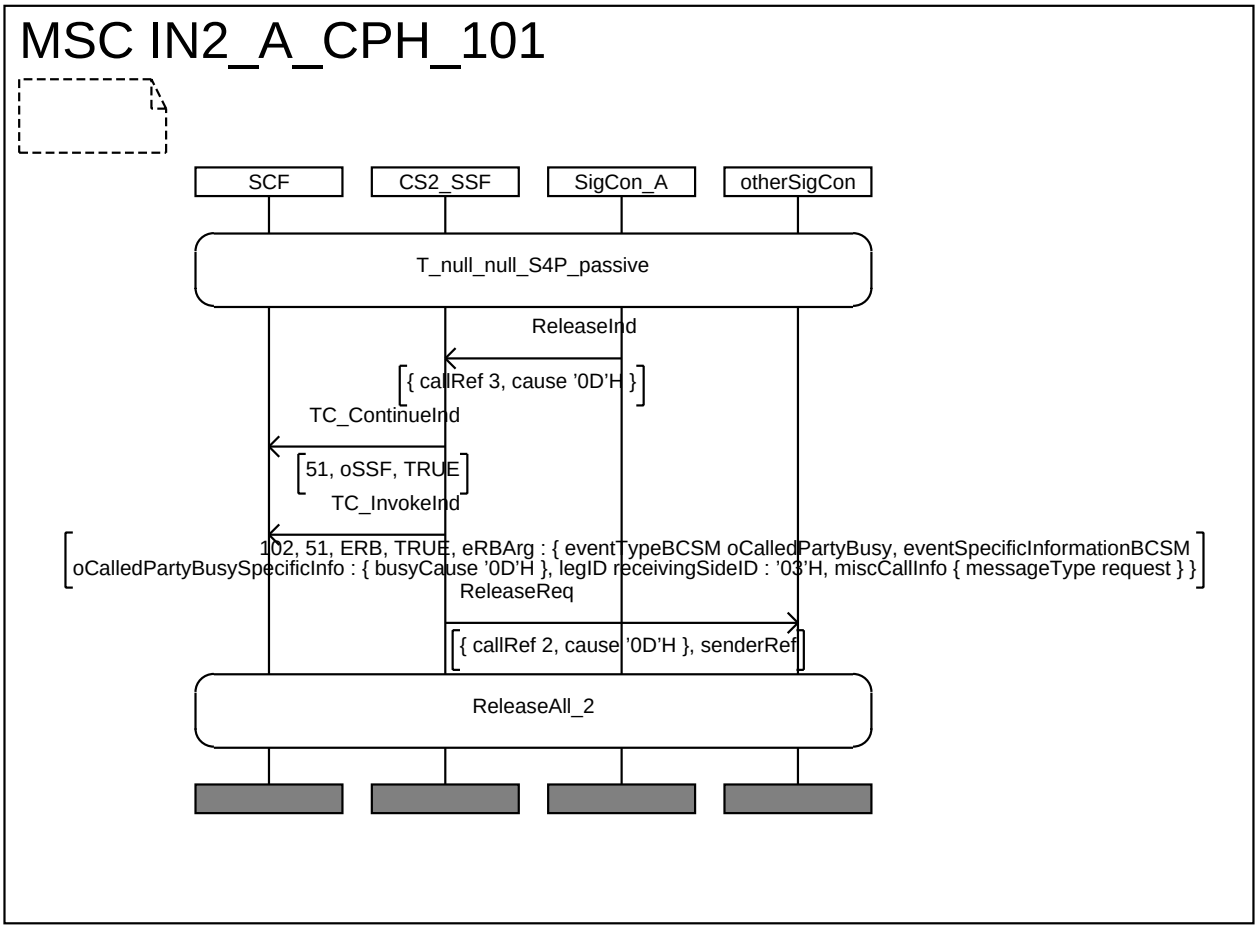
IN2_A_CPH_099	
Purpose:	Check that a single event oSuspended is reported
Requirement ref	
Selection Cond.	
Preamble:	T_null_null_S4P
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RRB(5, interrupted, oSuspend) L1! Connect(5,2) CP1_5? SetUpReq CP1_5! SetUpConf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


MSC IN2_A_CPH_099
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_null_S4P_passive
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oSuspended, monitorMode interrupted, legID sendingSideID : '05'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2007'H }, callSegmentID 2, legToBeCreated sendingSideID : '05'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 5, calledPartyNumber '2007'H }
SetupConf
{ callRef 5 }
SetupResp
{ callRef 2 }
NetworkSuspendInd
{ callRef 5 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oSuspended, legID receivingSideID : '05'H, miscCallInfo { messageType request } }
ReleaseAll_5

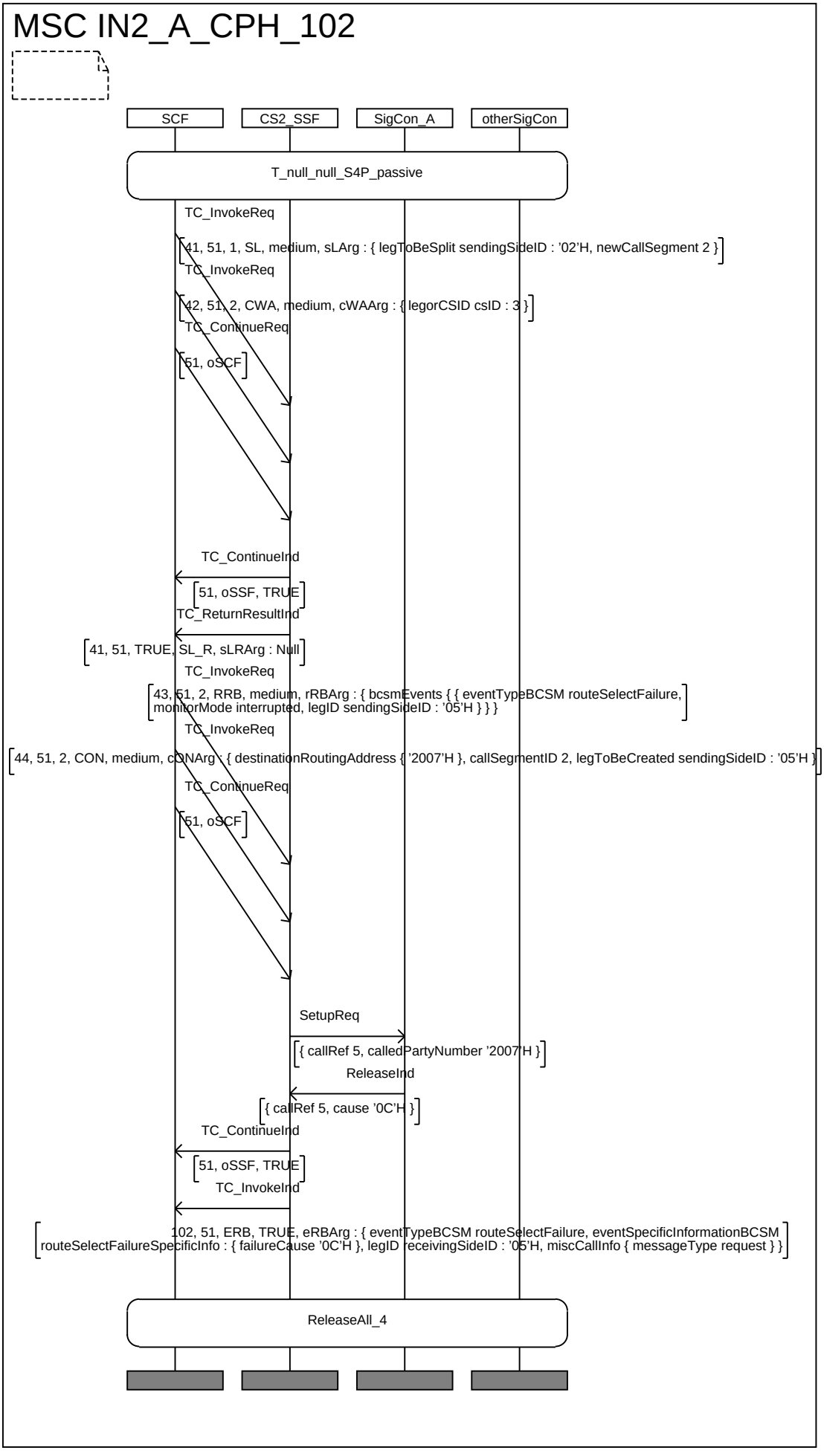
IN2_A_CPH_100	
Purpose:	Check that a single event oAnswer is reported
Requirement ref	
Selection Cond.	
Preamble:	T_null_null_S4P
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RRB(5, interrupted, oAnswer) L1! Connect(5,2) CP1_5? SetUpReq CP1_5! SetUpConf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


MSC IN2_A_CPH_100
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_null_S4P_passive
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM oAnswer, monitorMode interrupted, legID sendingSideID : '05'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2007'H }, callSegmentID 2, legToBeCreated sendingSideID : '05'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 5, calledPartyNumber '2007'H }
SetupConf
{ callRef 5 }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM oAnswer, legID receivingSideID : '05'H, miscCallInfo { messageType request } }
ReleaseAll_5

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

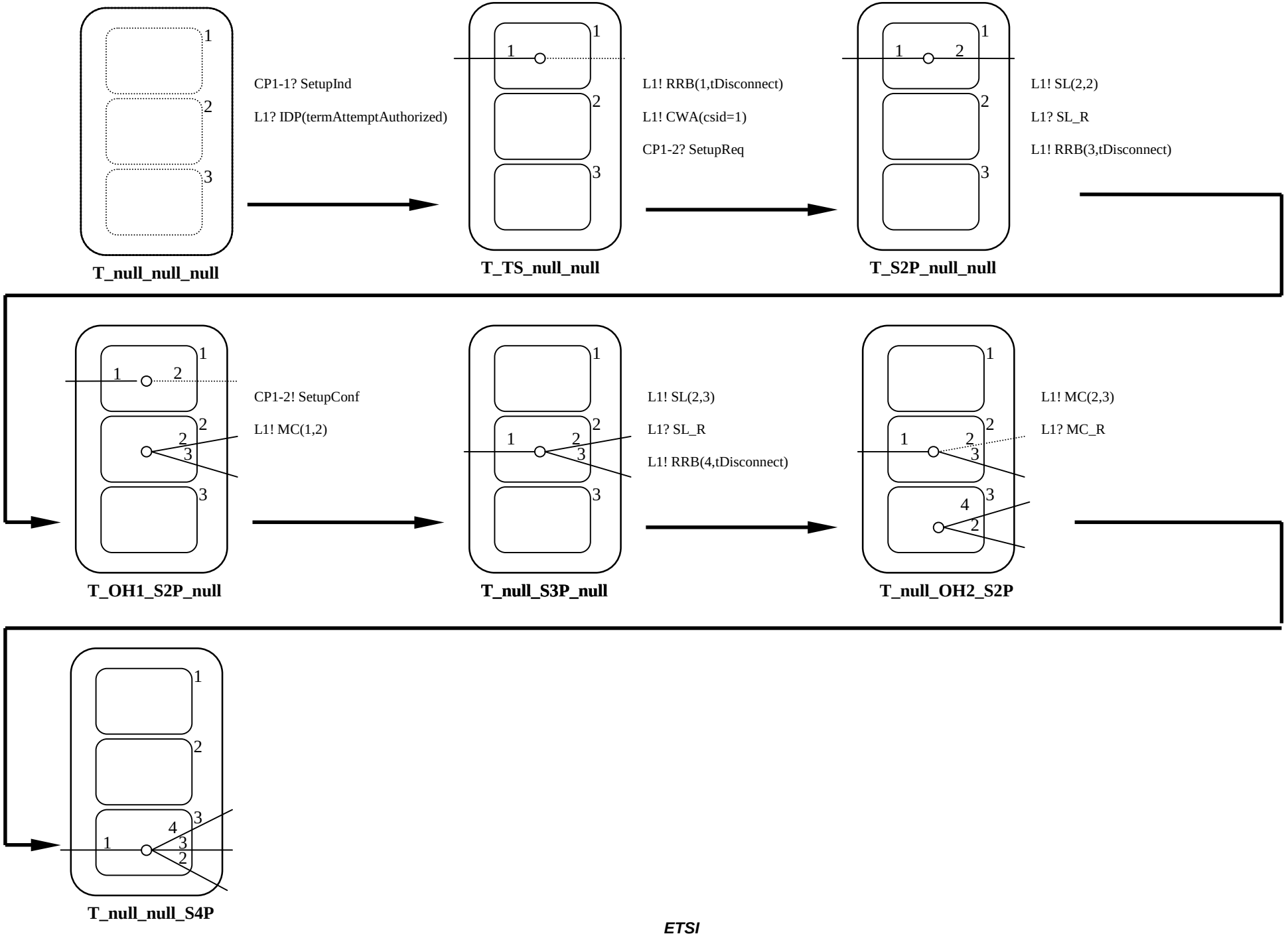


IN2_A_CPH_102	
Purpose:	Check that a single event routeSelectFailure is reported
Requirement ref	
Selection Cond.	
Preamble:	T_null_null_S4P
Test description	L1! SplitLeg(2,2) L1? SplitLegReturnResult L1! ContinueWithArgument(CsId=3) L1! RRB(5, interrupted, routeSelectFailure) L1! Connect(5,2) CP1_5?SetupReq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

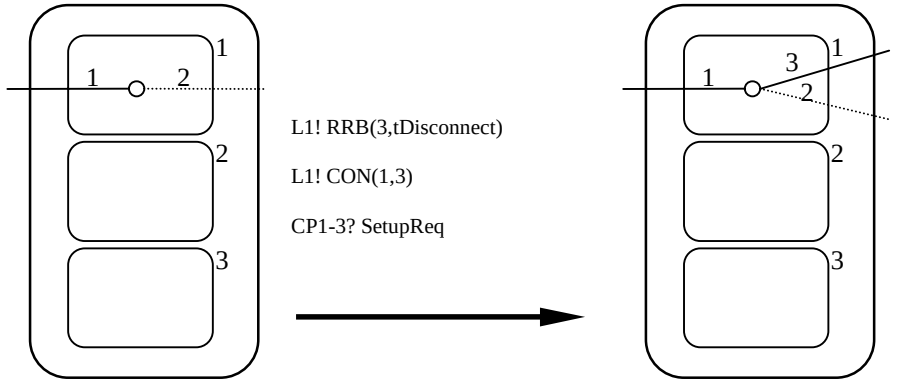

MSC IN2_A_CPH_102
SCF
CS2_SSF
SigCon_A
otherSigCon
T_null_null_S4P_passive
TC_InvokeReq
41, 51, 1, SL, medium, sLArg : { legToBeSplit sendingSideID : '02'H, newCallSegment 2 }
TC_InvokeReq
42, 51, 2, CWA, medium, cWAArg : { legorCSID csID : 3 }
TC_ContinueReq
51, oSCF
TC_ContinueInd
51, oSSF, TRUE
TC_ReturnResultInd
41, 51, TRUE, SL_R, sLRArg : Null
TC_InvokeReq
43, 51, 2, RRB, medium, rRBArg : { bcsmEvents { { eventTypeBCSM routeSelectFailure, monitorMode interrupted, legID sendingSideID : '05'H } } }
TC_InvokeReq
44, 51, 2, CON, medium, cONArg : { destinationRoutingAddress { '2007'H }, callSegmentID 2, legToBeCreated sendingSideID : '05'H }
TC_ContinueReq
51, oSCF
SetupReq
{ callRef 5, calledPartyNumber '2007'H }
ReleaseInd
{ callRef 5, cause '0C'H }
TC_ContinueInd
51, oSSF, TRUE
TC_InvokeInd
102, 51, ERB, TRUE, eRBArg : { eventTypeBCSM routeSelectFailure, eventSpecificInformationBCSM routeSelectFailureSpecificInfo : { failureCause '0C'H }, legID receivingSideID : '05'H, miscCallInfo { messageType request } }
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)



T_TS_null_null **T_TF2_null_nul**

Annex B (normative): TCAP Parameter values

Table B.1: Parameter values

Item	Parameter	Parameter type	Explanation	Value
	PIX_InvokeId	InvokeIDType	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	"xxx"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_ScfID	ScfID	"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

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