ETSI ES 202 196-12 V1.2.1 (2005-02)

ETSI Standard

Open Service Access (OSA);
Application Programming Interface (API);
Test Suite Structure and Test Purposes (TSS&TP);
Part 12: Charging SCF
(Parlay 3)

ETSI

Reference

RES/TISPAN-06005-12-OSA

Keywords

API, OSA, TSS&TP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

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

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

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 ETSI 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://webapp.etsi.org/IPR/home.asp).

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 ETSI 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 ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 12 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API); Test Suite Structure and Test Purposes (TSS&TP) for Parlay 3, as identified below:

Part 1: "Overview";

Part 2: "Common data definitions";

Part 3: "Framework";

Part 4: "Call control SCF";

Part 5: "User interactions SCF";

Part 6: "Mobility SCF";

Part 7: "Terminal capabilities SCF";

Part 8: "Data session control SCF";

Part 9: "Generic messaging SCF";

Part 10: "Connectivity manager SCF";

Part 11: "Account management SCF";

Part 12: "Charging SCF".

To evaluate conformance of a particular implementation, it is necessary to have a set of test purposes to evaluate the dynamic behaviour of the Implementation Under Test (IUT). The specification containing those test purposes is called a Test Suite Structure and Test Purposes (TSS&TP) specification.

1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS&TP) specification for the charging SCF of the Application Programming Interface (API) for Open Service Access (OSA) defined in ES 201 915-12 [1] in compliance with the relevant requirements, and in accordance with the relevant guidance given in ISO/IEC 9646-2 [4] and ETS 300 406 [5].

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 and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1]   ETSI ES 201 915-12: "Open Service Access (OSA); Application Programming Interface (API); Part 12: Charging SCF (Parlay 3)".

[2]   ETSI ES 202 170: "Open Service Access (OSA); Application Programming Interface (API); Implementation Conformance Statement (ICS) proforma specification (Parlay 3)".

[3]   ISO/IEC 9646-1: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[4]   ISO/IEC 9646-2: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification".

[5]   ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 915-12 [1], ISO/IEC 9646-1 [3], ISO/IEC 9646-2 [4] and the following apply:

abstract test case: Refer to ISO/IEC 9646-1 [3].

Abstract Test Method (ATM): Refer to ISO/IEC 9646-1 [3].

Abstract Test Suite (ATS): Refer to ISO/IEC 9646-1 [3].

ICS proforma: Refer to ISO/IEC 9646-1 [3].

Implementation Conformance Statement (ICS): Refer to ISO/IEC 9646-1 [3].

Implementation eXtra Information for Testing (IXIT): Refer to ISO/IEC 9646-1 [3].

Implementation Under Test (IUT): Refer to ISO/IEC 9646-1 [3].

IXIT proforma: Refer to ISO/IEC 9646-1 [3].

Lower Tester (LT): Refer to ISO/IEC 9646-1 [3].

Test Purpose (TP): Refer to ISO/IEC 9646-1 [3].

3.2 Abbreviations

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

API	Application Programming Interface
ATM	Abstract Test Method
ATS	Abstract Test Suite
CH	CHarging
CM	Charging Manager
CS	Charging Session
ICS	Implementation Conformance Statement
IUT	Implementation Under Test
IXIT	Implementation eXtra Information for Testing
LT	Lower Tester
OSA	Open Service Access
TP	Test Purpose
TSS	Test Suite Structure

4 Test Suite Structure (TSS)

- Charging:

    - IpChargingManager (CM);

    - IpChargingSession (CS).

5 Test Purposes (TP)

5.1 Introduction

For each test requirement a TP is defined.

5.1.1 TP naming convention

TPs are numbered, starting at 01, within each group. Groups are organized according to the TSS. Additional references are added to identify the actual test suite (see table 1).

Table 1: TP identifier naming convention scheme

Identifier: <suite_id>_<group>_<nnn>		
<suite_id>	= SCG name:	"CH" for **CH**arging SCF
<group>	= group number:	two character field representing the group reference according to TSS
<nn>	= sequential number:	(01 to 99)

5.1.2 Source of TP definition

The TPs are based on ES 201 915-12 [1].

5.1.3   Test strategy

As the base standard ES 201 915-12 [1] contains no explicit requirements for testing, the TPs were generated as a result of an analysis of the base standard and the ICS specification ES 202 170 [2].

The TPs are only based on conformance requirements related to the externally observable behaviour of the IUT and are limited to conceivable situations to which a real implementation is likely to be faced (see ETS 300 406 [5]).

5.2   TPs for the Charging SCF

All ICS items referred to in this clause are as specified in ES 202 170 [2] unless indicated otherwise by another numbered reference.

All parameters specified in method calls are valid unless specified.

The procedures to trigger the SCF to call methods in the application are dependant on the underlying network architecture and are out of the scope of this test specification. Those method calls are preceded by the words "Triggered action".

5.2.1   Charging

5.2.1.1   IpChargingManager
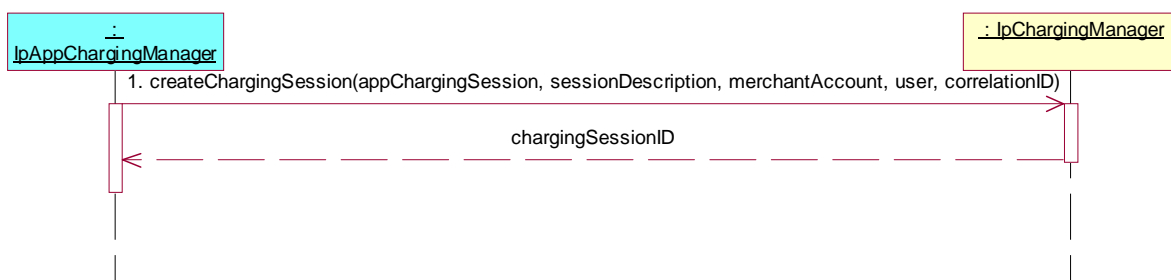
Test CH_CM_01

Summary: **IpChargingManager**, successful

Reference: ES 201 915-12 [1], clause 8.1

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1.   Method call **createChargingSession()**
      Parameters: appChargingSession, sessionDescription, merchantAccount, user, correlationID
      Check: valid value of TpChargingSessionID is returned

5.2.1.2 IpChargingSession

Test CH_CS_01

Summary: **IpChargingSession**, amount reservation and debit, successful
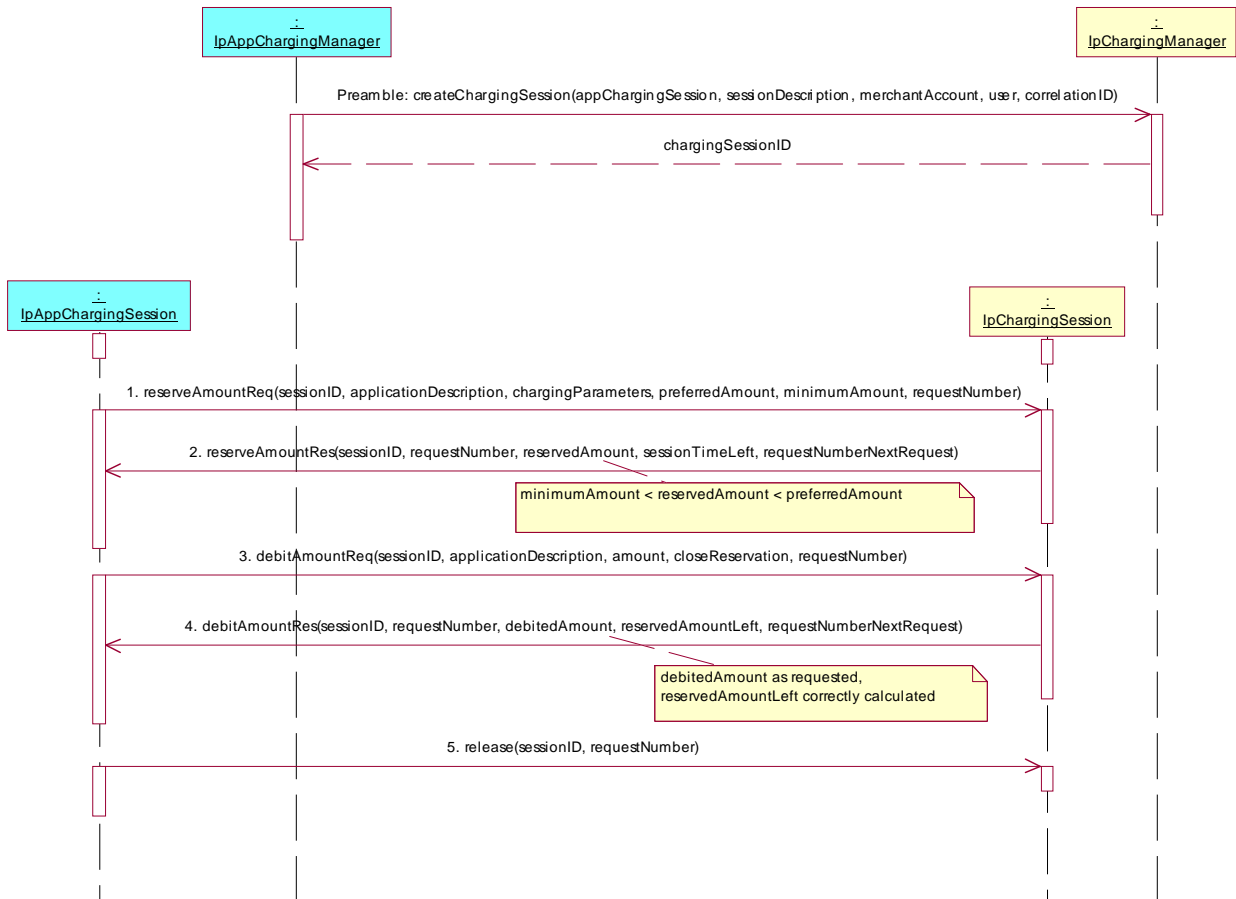
Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveAmountReq, debitAmountReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveAmountReq()**
   Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber
   Check: no exception is returned

2. Triggered action: cause IUT to call **reserveAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of an amount of money
   Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
   Check: reservedAmount has a value between the requested preferredAmount and minimumAmount

3. Method call **debitAmountReq()**
   Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber
   Check: no exception is returned

4. Triggered action: cause IUT to call **debitAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the debit request
   Parameters: sessionID, requestNumber, debitedAmount, reservedAmountLeft, requestNumberNextRequest
   Check: debitedAmount as requested, reservedAmountLeft correctly calculated

5. Method call **release()**
   Parameters: sessionID, requestNumber
   Check: no exception is returned

Test CH_CS_02

Summary: **IpChargingSession**, amount reservation and credit, successful

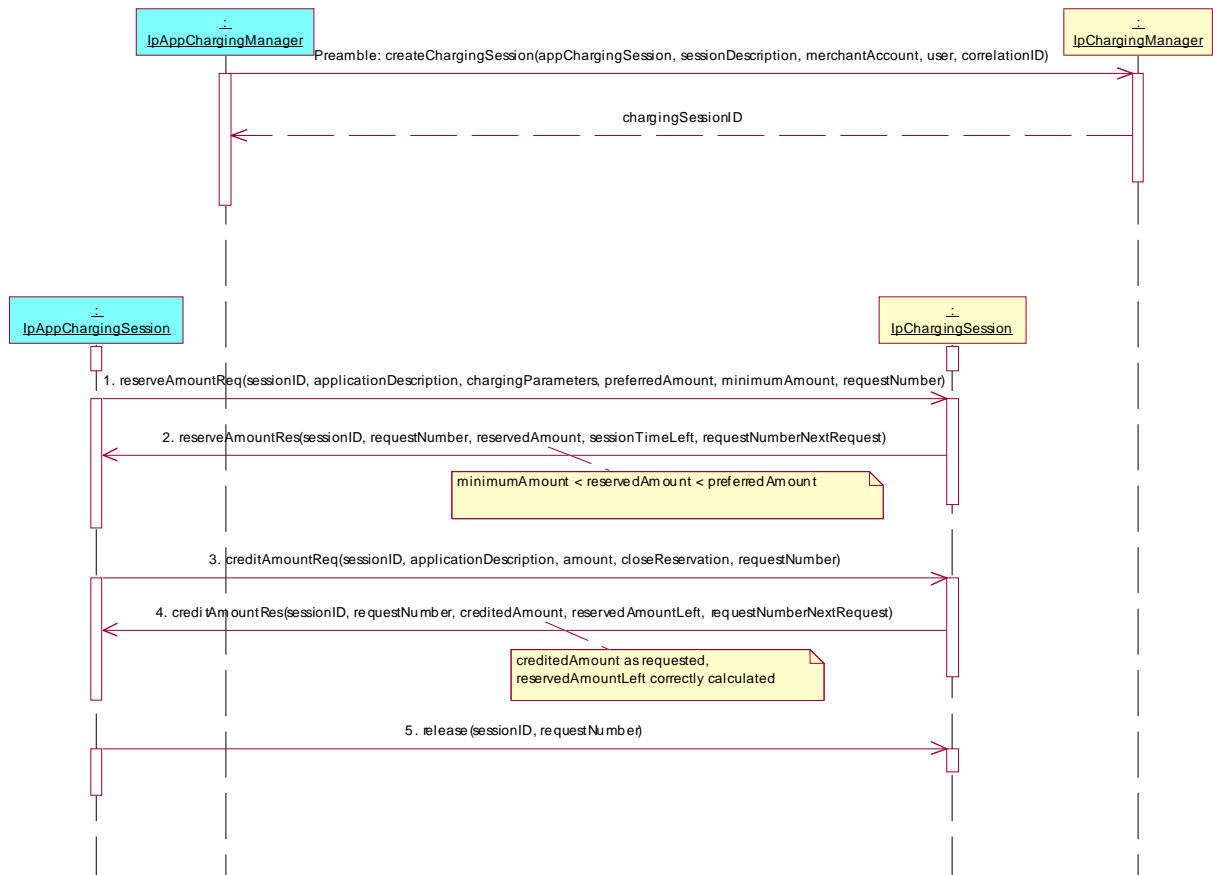Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveAmountReq, creditAmountReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of an amount of money
    Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
    Check: reservedAmount has a value between the requested preferredAmount and minimumAmount

3. Method call **creditAmountReq()**
    Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **creditAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the credit request
    Parameters: sessionID, requestNumber, creditedAmount, reservedAmountLeft, requestNumberNextRequest
    Check: creditedAmount as requested, reservedAmountLeft correctly calculated

5. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

ETSI

Test CH_CS_03

Summary: **IpChargingSession**, amount reservation and variation, successful
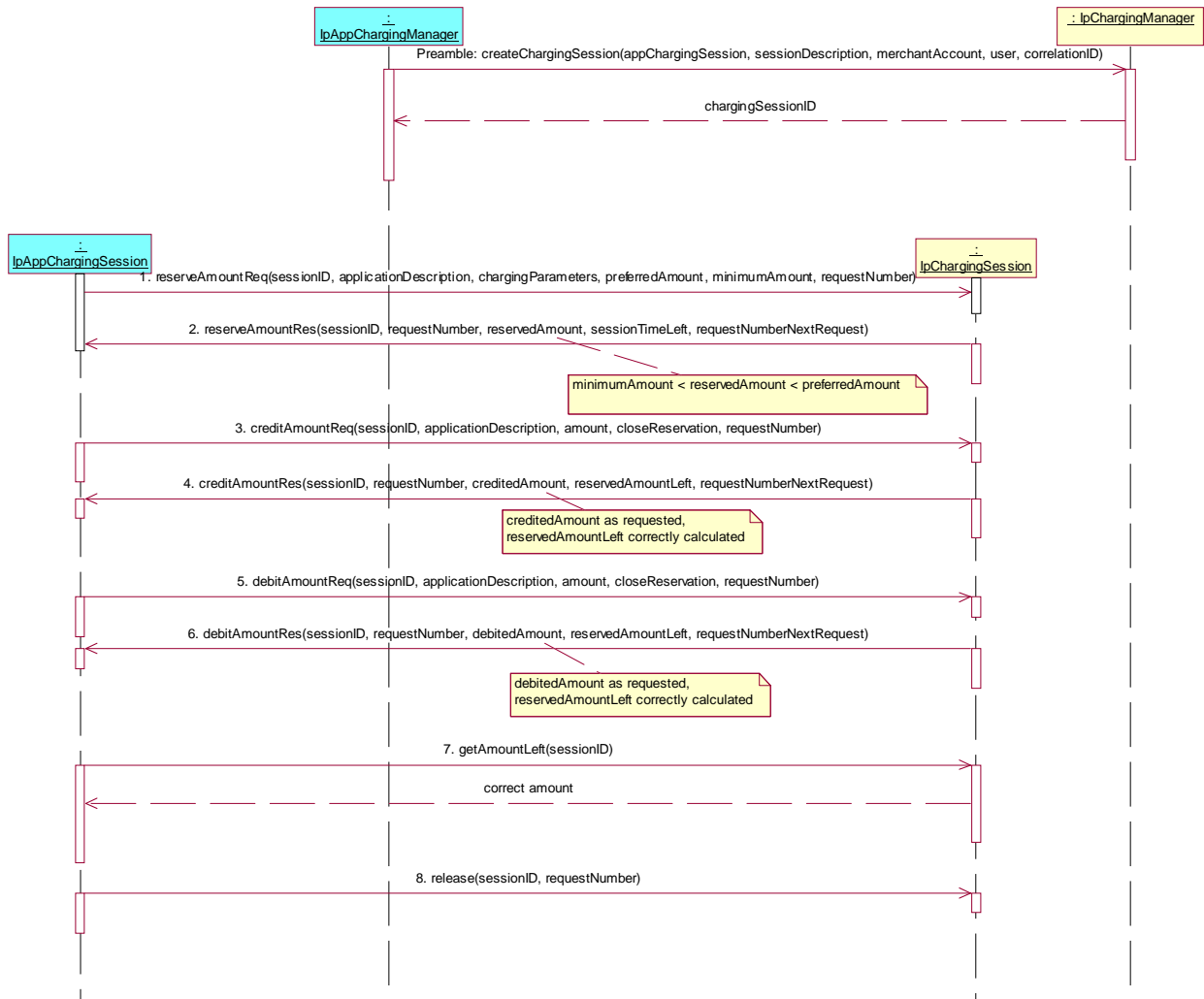
Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveAmountReq, debitAmountReq, creditAmountReq, getAmountLeft supported.

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveAmountReq()**
   Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber
   Check: no exception is returned

2. Triggered action: cause IUT to call **reserveAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of an amount of money
   Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
   Check: reservedAmount has a value between the requested preferredAmount and minimumAmount

3. Method call **creditAmountReq()**
   Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber
   Check: no exception is returned

4. Triggered action: cause IUT to call **creditAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the credit request
   Parameters: sessionID, requestNumber, creditedAmount, reservedAmountLeft, requestNumberNextRequest
   Check: creditedAmount as requested, reservedAmountLeft correctly calculated

5. Method call **debitAmountReq()**
   Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber
   Check: no exception is returned

6. Triggered action: cause IUT to call **debitAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the debit request
   Parameters: sessionID, requestNumber, debitedAmount, reservedAmountLeft, requestNumberNextRequest
   Check: debitedAmount as requested, reservedAmountLeft correctly calculated

7. Method call **getAmountLeft()**
   Parameters: sessionID
   Check: correct value of TpChargingPrice is returned

8. Method call **release()**
   Parameters: sessionID, requestNumber
   Check: no exception is returned

Test CH_CS_04

Summary: **IpChargingSession**, amount reservation and variation of life time, successful

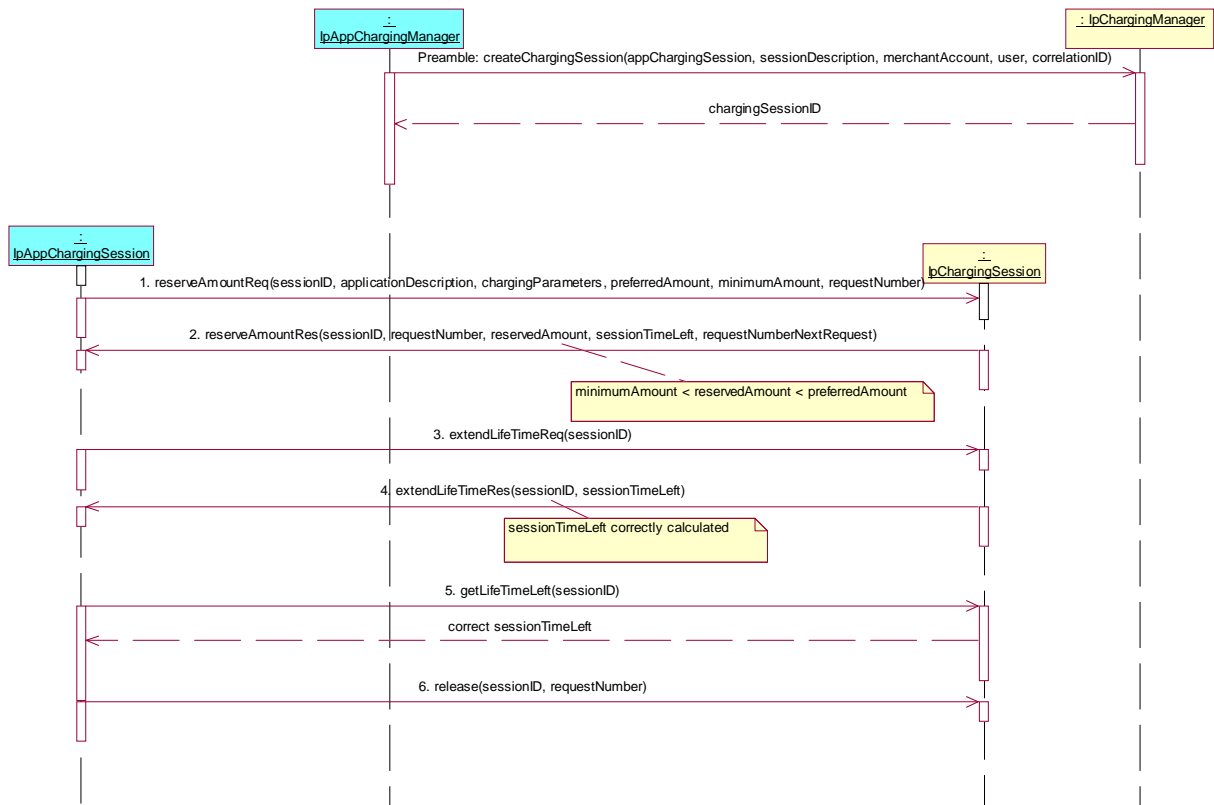Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveAmountReq, extendLifeTimeReq, getLifeTimeLeft supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                  Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of an amount of money
    Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
    Check: reservedAmount has a value between the requested preferredAmount and minimumAmount

3. Method call **extendLifeTimeReq()**
    Parameters: sessionID
    Check: no exception is returned

4. Triggered action: cause IUT to call **extendLifeTimeRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the life time extension
    Parameters: sessionID, sessionTimeLeft
    Check: sessionTimeLeft correctly calculated

5. Method call **getLifeTimeLeft()**
    Parameters: sessionID
    Check: correct value of TpInt32 is returned

6. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_05

Summary: **IpChargingSession**, unit reservation and debit, successful

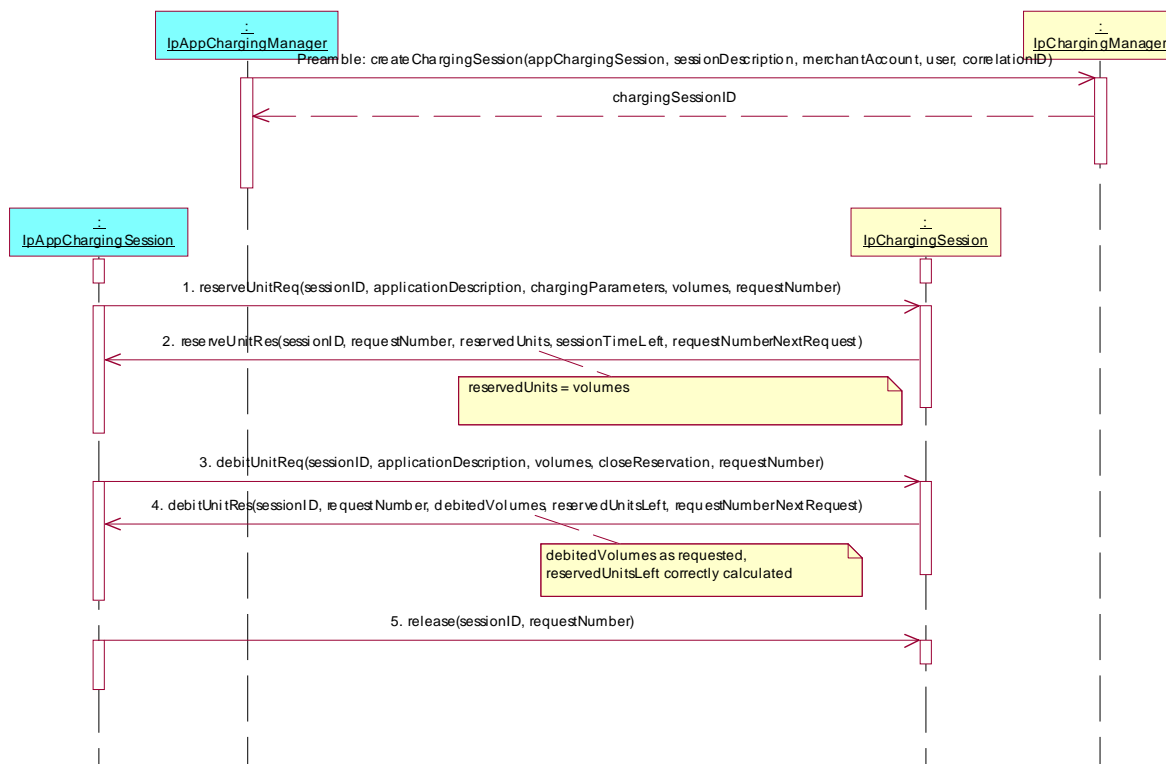Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveUnitReq, debitUnitReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveUnitRes()** method on the tester's (application's)
    IpAppChargingSession interface, i.e. accept the reservation of a number of units
    Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
    Check: reservedUnits has the requested value

3. Method call **debitUnitReq()**
    Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **debitUnitRes()** method on the tester's (application's)
    IpAppChargingSession interface, i.e. accept the debit request
    Parameters: sessionID, requestNumber, debitedVolumes, reservedUnitsLeft, requestNumberNextRequest
    Check: debitedVolumes as requested, reservedUnitsLeft correctly calculated

5. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_06

Summary: **IpChargingSession**, unit reservation and credit, successful

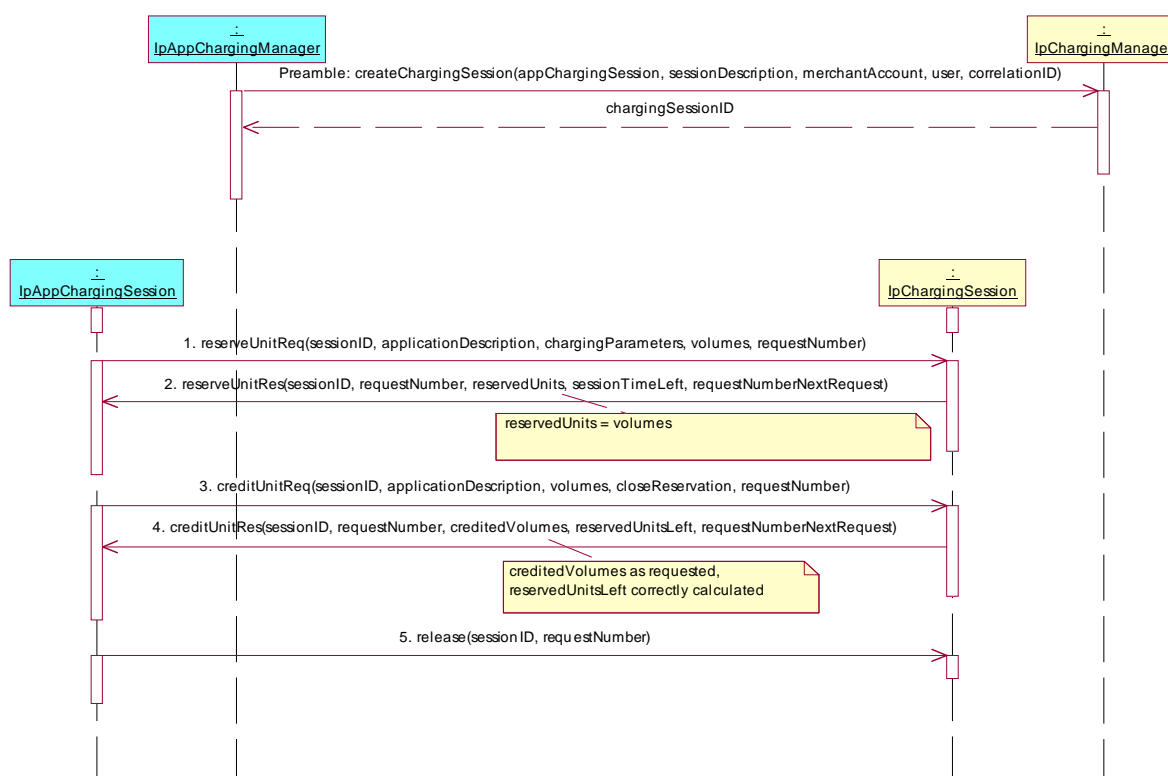Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveUnitReq, creditUnitReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveUnitReq()**
   Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
   Check: no exception is returned

2. Triggered action: cause IUT to call **reserveUnitRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of a number of units
   Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
   Check: reservedUnits has the requested value

3. Method call **creditUnitReq()**
   Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber
   Check: no exception is returned

4. Triggered action: cause IUT to call **creditUnitRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the credit request
   Parameters: sessionID, requestNumber, creditedVolumes, reservedUnitsLeft, requestNumberNextRequest
   Check: creditedVolumes as requested, reservedUnitsLeft correctly calculated

5. Method call **release()**
   Parameters: sessionID, requestNumber
   Check: no exception is returned

Test CH_CS_07

Summary: **IpChargingSession**, unit reservation and variation, successful

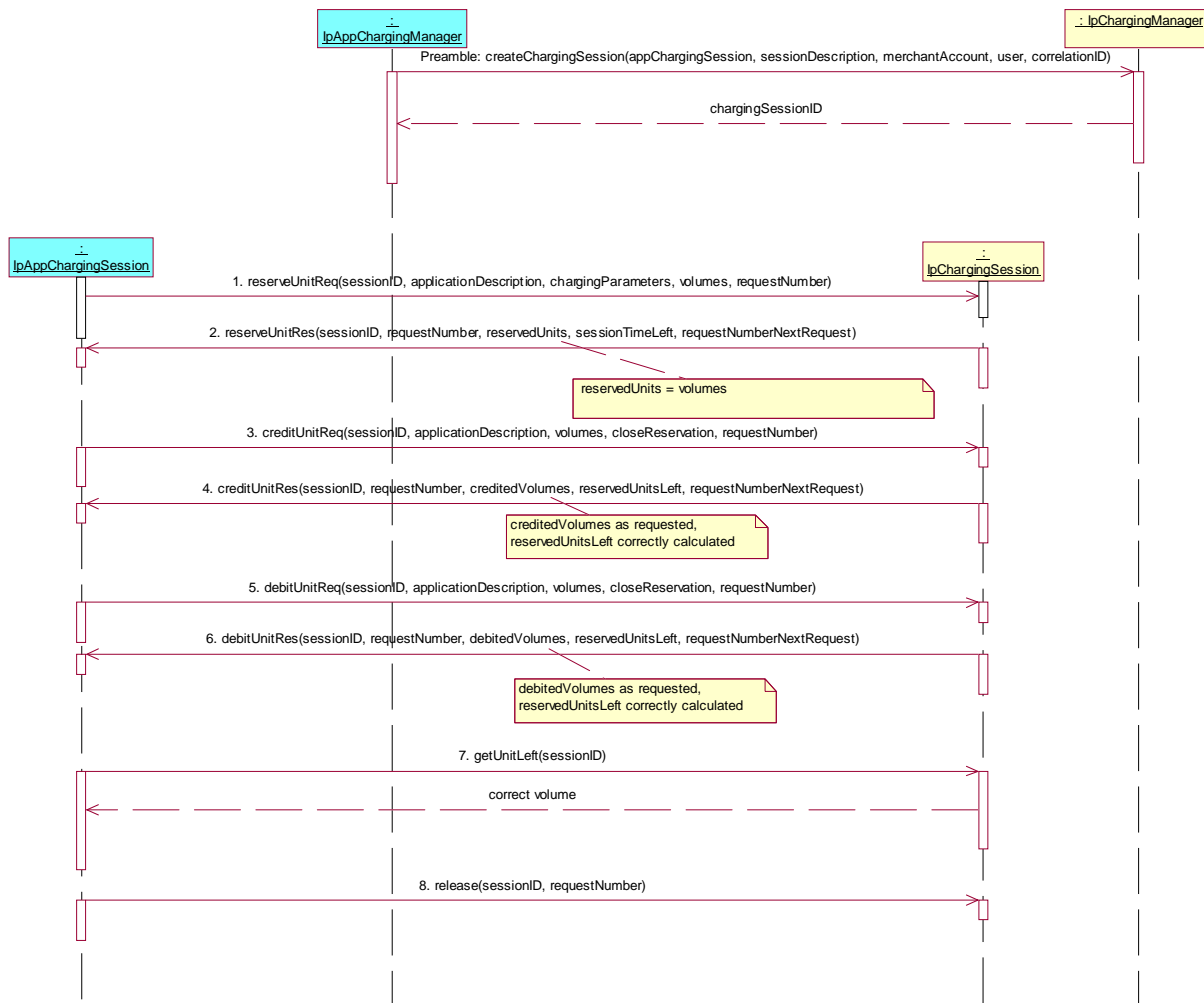Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveUnitReq, debitUnitReq, creditUnitReq, getUnitLeft supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must
                 have obtained a reference to an instance of the IpChargingManager interface through selecting that
                 service and signing the required service agreement.

                 Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveUnitRes()** method on the tester's (application's)
    IpAppChargingSession interface, i.e. accept the reservation of a number of units
    Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
    Check: reservedUnits has the requested value

3. Method call **creditUnitReq()**
    Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **creditUnitRes()** method on the tester's (application's)
    IpAppChargingSession interface, i.e. accept the credit request
    Parameters: sessionID, requestNumber, creditedVolumes, reservedUnitsLeft, requestNumberNextRequest
    Check: creditedVolumes as requested, reservedUnitsLeft correctly calculated

5. Method call **debitUnitReq()**
    Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber
    Check: no exception is returned

6. Triggered action: cause IUT to call **debitUnitRes()** method on the tester's (application's)
    IpAppChargingSession interface, i.e. accept the debit request
    Parameters: sessionID, requestNumber, debitedVolumes, reservedUnitsLeft, requestNumberNextRequest
    Check: debitedVolumes as requested, reservedUnitsLeft correctly calculated

7. Method call **getUnitLeft()**
    Parameters: sessionID
    Check: correct value of TpVolumeSet is returned

8. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_08

Summary: **IpChargingSession**, unit reservation and variation of life time, successful
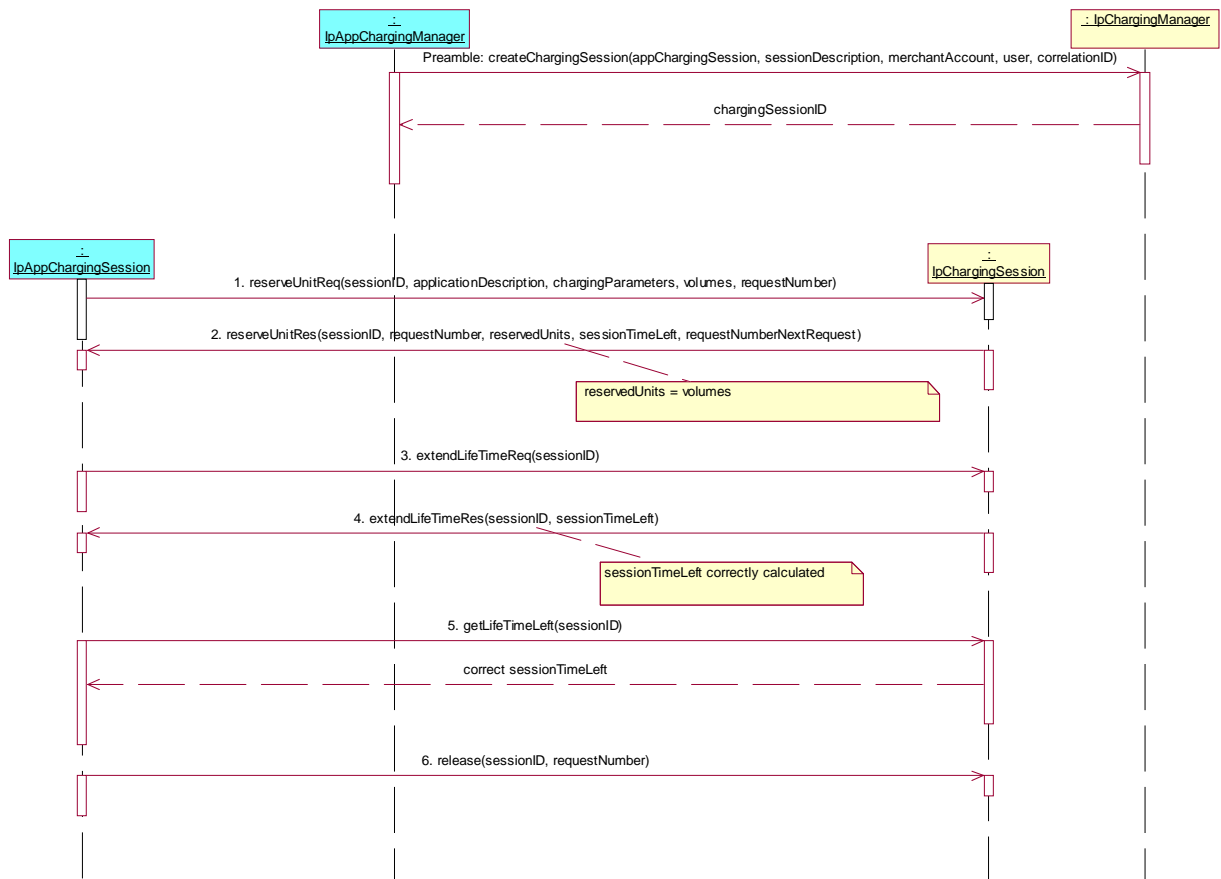
Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveUnitReq, extendLifeTimeReq, getLifeTimeLeft supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveUnitReq()**
   Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
   Check: no exception is returned

2. Triggered action: cause IUT to call **reserveUnitRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of a number of units
   Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
   Check: reservedUnits has the requested value

3. Method call **extendLifeTimeReq()**
   Parameters: sessionID
   Check: no exception is returned

4. Triggered action: cause IUT to call **extendLifeTimeRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the life time extension
   Parameters: sessionID, sessionTimeLeft
   Check: sessionTimeLeft correctly calculated

5. Method call **getLifeTimeLeft()**
   Parameters: sessionID
   Check: correct value of TpInt32 is returned

6. Method call **release()**
   Parameters: sessionID, requestNumber
   Check: no exception is returned

Test CH_CS_09

Summary: **IpChargingSession**, rateReq, successful

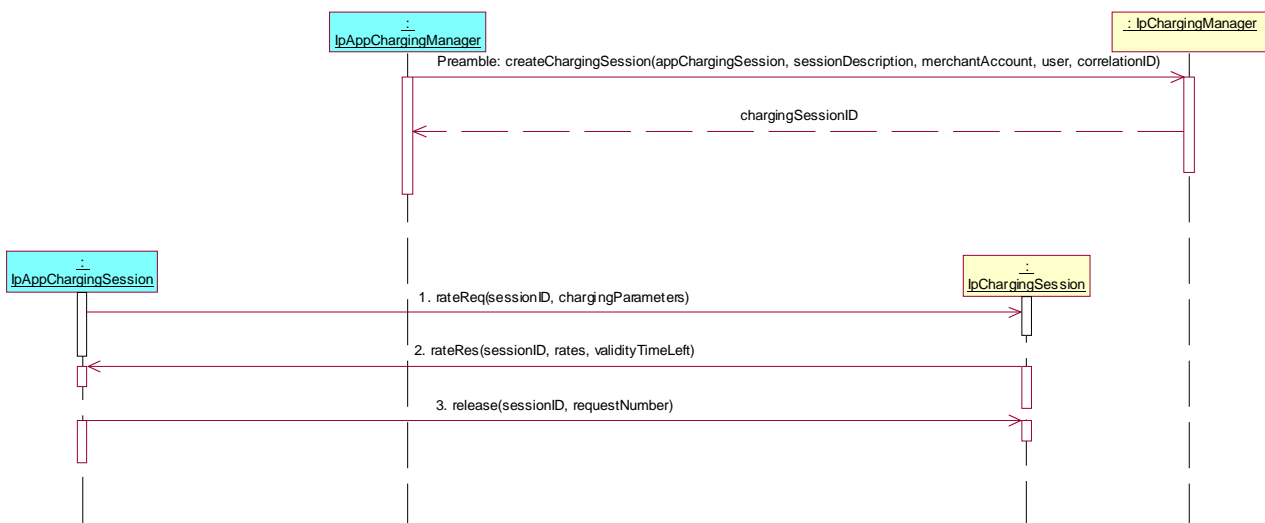Reference: ES 201 915-12 [1], clause 8.3

Precondition: rateReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **rateReq()**
   Parameters: sessionID, chargingParameters
   Check: no exception is returned

2. Triggered action: cause IUT to call **rateRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for rating information
   Parameters: sessionID, rates, validityTimeLeft

3. Method call **release()**
   Parameters: sessionID, requestNumber
   Check: no exception is returned

Test CH_CS_10

Summary: **IpChargingSession**, directCreditAmountReq, no reservation, successful

Reference: ES 201 915-12 [1], clause 8.3

Precondition: directCreditAmountReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **directCreditAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **directCreditAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct crediting
    Parameters: sessionID, requestNumber, creditedAmount, requestNumberNextRequest
    Check: creditedAmount has the requested value

3. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_11

Summary: **IpChargingSession**, directDebitAmountReq, no reservation, successful

Reference: ES 201 915-12 [1], clause 8.3

Precondition: directDebitAmountReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **directDebitAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **directDebitAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct debiting
    Parameters: sessionID, requestNumber, debitedAmount, requestNumberNextRequest
    Check: debitedAmount has the requested value

3. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_12

Summary: **IpChargingSession**, directCreditUnitReq, no reservation, successful

Reference: ES 201 915-12 [1], clause 8.3

Precondition: directCreditUnitReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

             Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **directCreditUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **directCreditUnitRes()** method on the tester's (application's)
    IpAppChargingSession interface, i.e. accept the request for direct crediting
    Parameters: sessionID, requestNumber, creditedVolumes, requestNumberNextRequest
    Check: creditedVolumes has the requested value

3. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_13

Summary: **IpChargingSession**, directDebitUnitReq, no reservation, successful

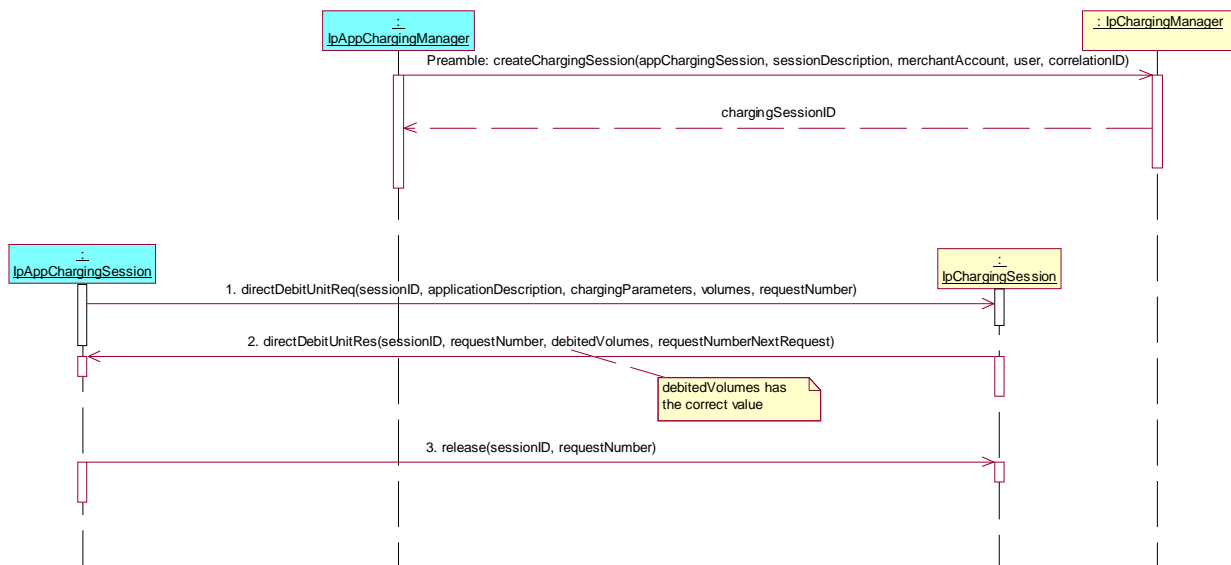Reference: ES 201 915-12 [1], clause 8.3

Precondition: directDebitUnitReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **directDebitUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **directDebitUniRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct debiting
    Parameters: sessionID, requestNumber, debitedVolumes, requestNumberNextRequest
    Check: debitedVolumes has the requested value

3. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_14

Summary: **IpChargingSession**, directCreditAmountReq, check that direct method does not affect reservation

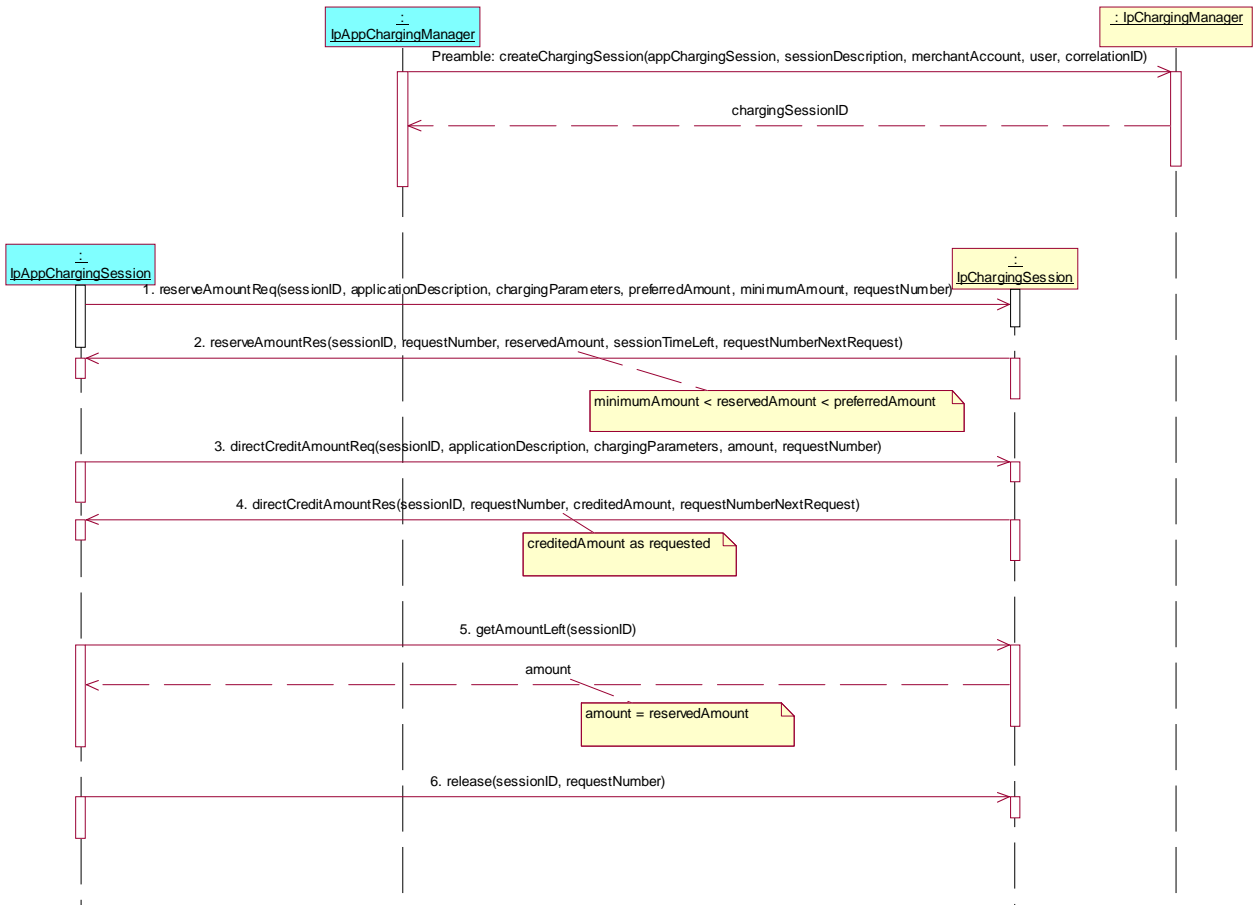Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveAmountReq, getAmountLeft, directCreditAmountReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

               Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of an amount of money
    Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
    Check: reservedAmount has a value between the requested preferredAmount and minimumAmount

3. Method call **directCreditAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **directCreditAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct crediting
    Parameters: sessionID, requestNumber, creditedAmount, requestNumberNextRequest
    Check: creditedAmount has the requested value

5. Method call **getAmountLeft()**
    Parameters: sessionID
    Check: value of TpChargingPrice equal to the reservedAmount received in 2. is returned

6. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_15

Summary: **IpChargingSession**, directDebitAmountReq, check that direct method does not affect reservation

Reference: ES 201 915-12 [1], clause 8.3
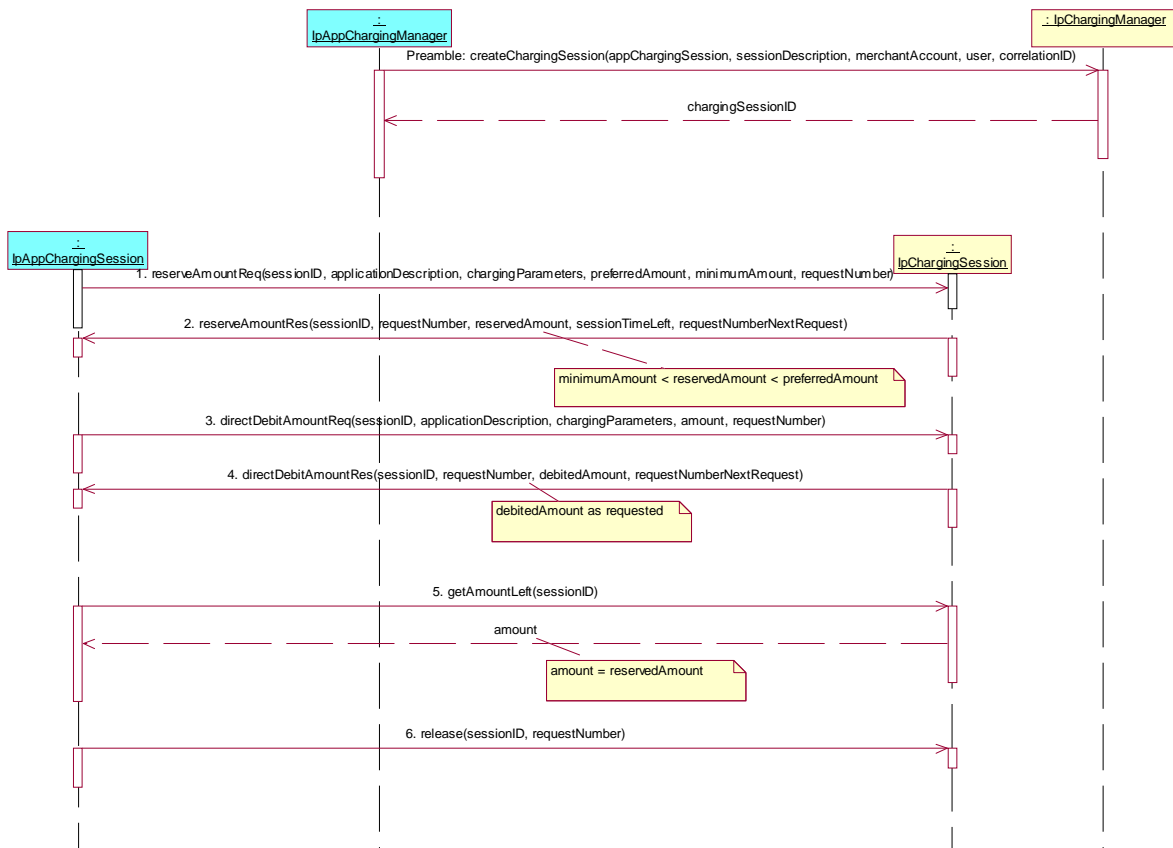
Precondition: reserveAmountReq, getAmountLeft,directDebitAmountReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of an amount of money
    Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
    Check: reservedAmount has a value between the requested preferredAmount and minimumAmount

3. Method call **directDebitAmountReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **directDebitAmountRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct debiting
    Parameters: sessionID, requestNumber, debitedAmount, requestNumberNextRequest
    Check: debitedAmount has the requested value

5. Method call **getAmountLeft()**
    Parameters: sessionID
    Check: value of TpChargingPrice equal to the reservedAmount received in 2. is returned

6. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_16

Summary: **IpChargingSession**, directCreditUnitReq, check that direct method does not affect reservation

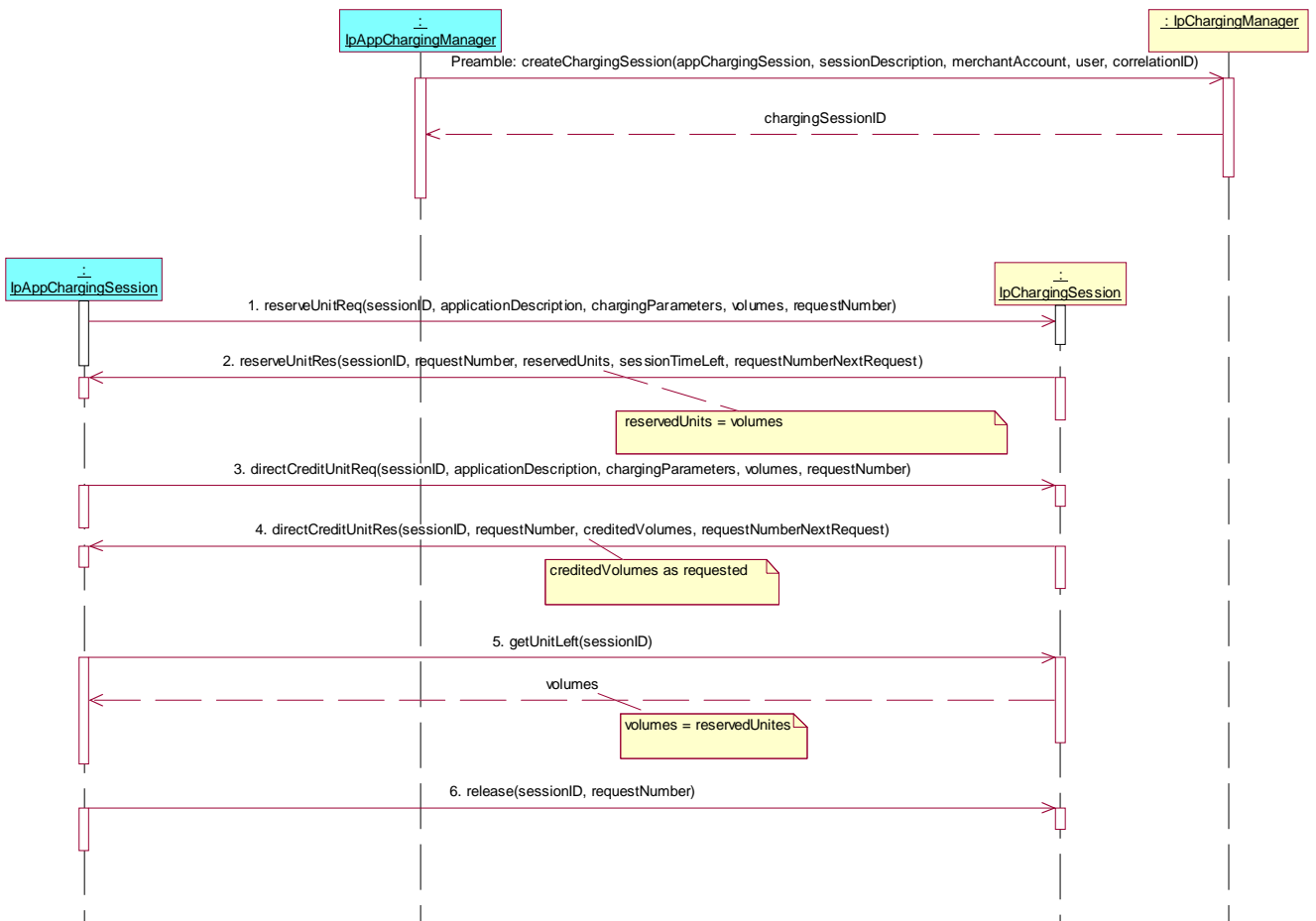Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveUnitReq, getUnitLeft,directCreditUnitReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveUnitRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of a number of units
    Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
    Check: reservedUnits has the requested value

3. Method call **directCreditUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **directCreditUnitRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct crediting
    Parameters: sessionID, requestNumber, creditedVolumes, requestNumberNextRequest
    Check: creditedVolumes has the requested value

5. Method call **getUnitLeft()**
    Parameters: sessionID
    Check: value of TpVolumeSet equal to the reservedUnits received in 2. is returned

6. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

Test CH_CS_17

Summary: **IpChargingSession**, directDebitUnitReq, check that direct method does not affect reservation

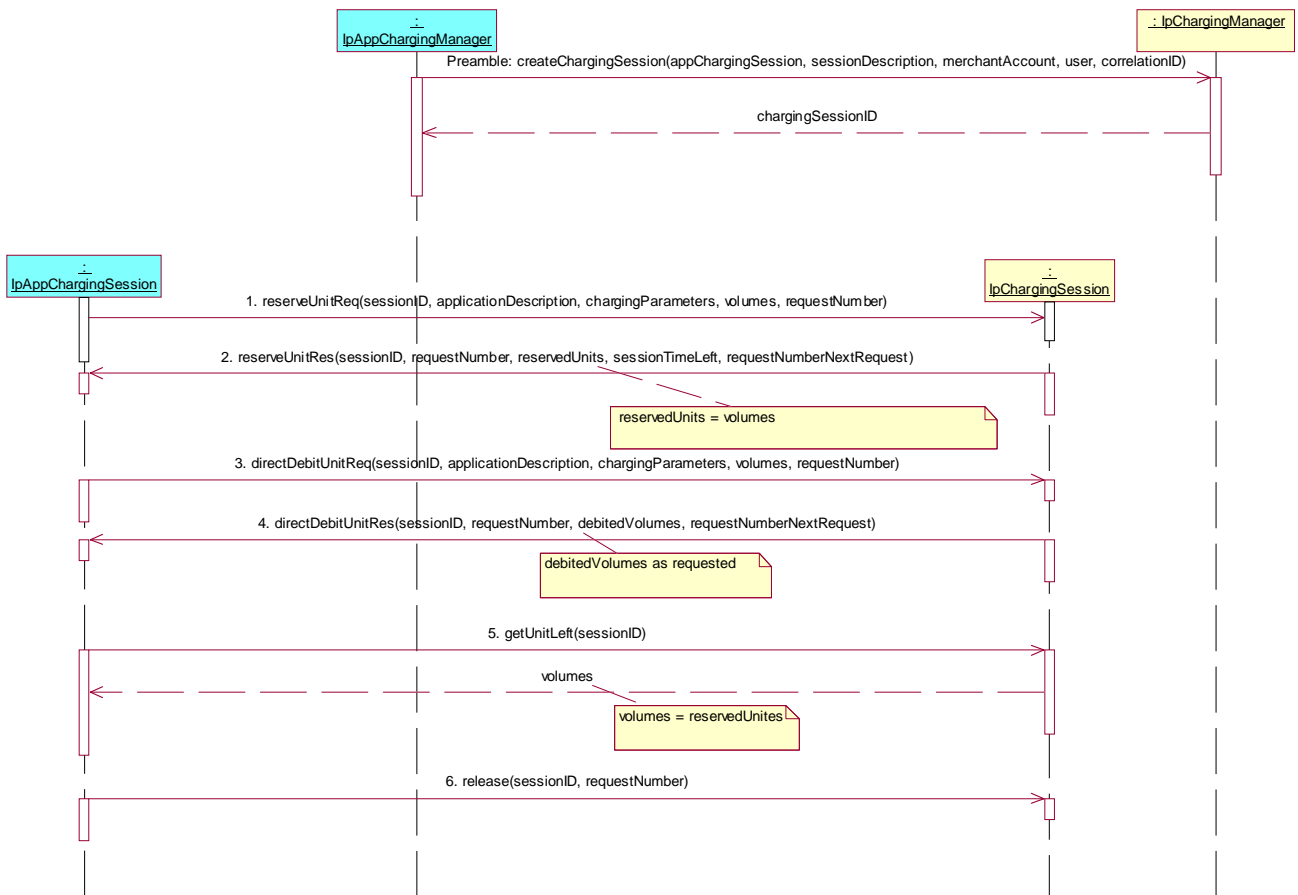Reference: ES 201 915-12 [1], clause 8.3

Precondition: reserveUnitReq, getUnitLeft,directDebitUnitReq supported

Preamble: Registration of the IUT (Charging SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpChargingManager interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Method call **reserveUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

2. Triggered action: cause IUT to call **reserveUnitRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the reservation of a number of units
    Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
    Check: reservedUnits has the requested value

3. Method call **directDebitUnitReq()**
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber
    Check: no exception is returned

4. Triggered action: cause IUT to call **directDebitUniRes()** method on the tester's (application's) **IpAppChargingSession** interface, i.e. accept the request for direct debiting
    Parameters: sessionID, requestNumber, debitedVolumes, requestNumberNextRequest
    Check: debitedVolumes has the requested value

5. Method call **getUnitLeft()**
    Parameters: sessionID
    Check: value of TpVolumeSet equal to the reservedUnits received in 2. is returned

6. Method call **release()**
    Parameters: sessionID, requestNumber
    Check: no exception is returned

5.3 TPs for the application using the Charging SCF

All ICS items referred to in this clause are as specified in ES 202 170 [2] unless indicated otherwise by another numbered reference.

All parameters specified in method calls are valid unless specified.

The procedures to trigger the application to call methods in the SCF are dependant on the underlying network architecture and are out of the scope of this test specification. Those method calls are preceded by the words "Triggered action".

5.3.1 Charging

5.3.1.1 IpAppChargingManager

Test CH_APP_CM_01

Summary: Create charging session

Reference: ES 201 915-12 [1], clause 8.1

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework.

Test Sequence:

1. Triggered action: cause IUT to call **createChargingSession()** method on the tester's (SCF) **IpChargingManager** interface.
   Parameters: appChargingSession, sessionDescription, merchantAccount, user, correlationID
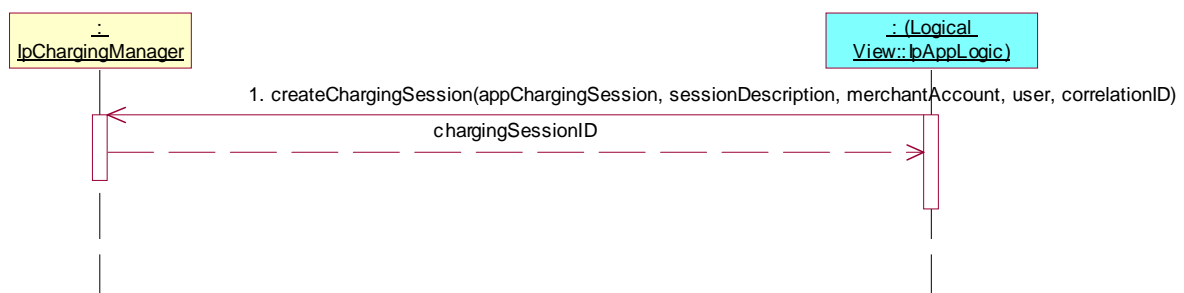
Test CH_APP_CM_02

Summary: Create charging session and accept abort

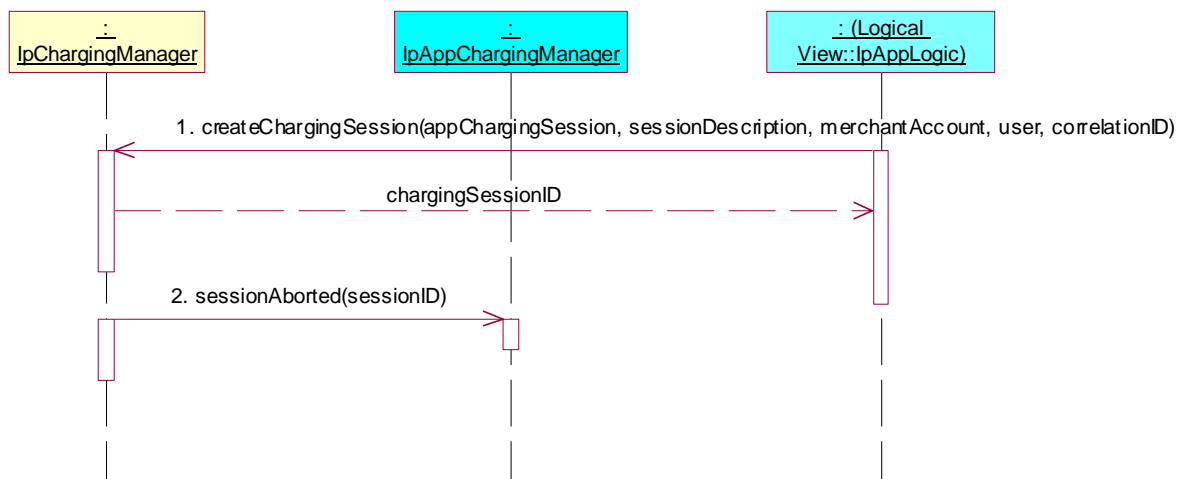Reference: ES 201 915-12 [1], clause 8.1

Precondition: **IpAppChargingManager** interfaceimplemented

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingManager interface through selecting that
                service and signing the required service agreement.

Test Sequence:

  1.   Triggered action: cause IUT to call **createChargingSession()** method on the tester's (SCF)
       IpChargingManager interface.
       Parameters: appChargingSession, sessionDescription, merchantAccount, user, correlationID

  2.   Method call **sessionAborted()**
       Parameters: sessionId
       Check: no exception is returned

5.3.1.2   IpAppChargingSession

Test CH_APP_CS_01

Summary: **IpAppChargingSession**, reserve amount, successful

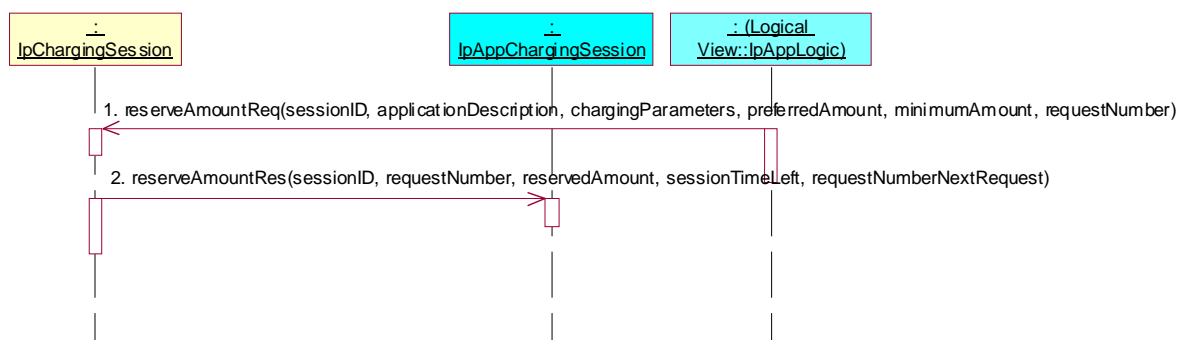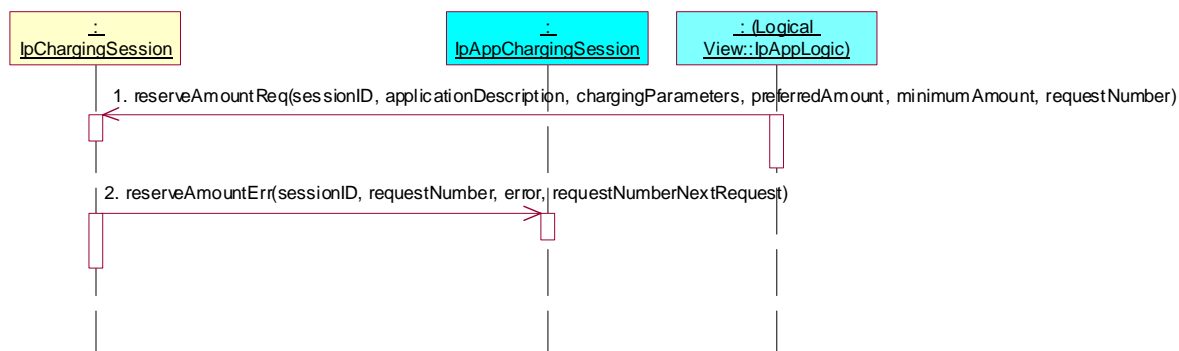Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession**
     interface.
     Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount,
                    requestNumber

2.   Method call **reserveAmountRes()**
     Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
     Check: no exception is returned

Test CH_APP_CS_02

Summary: **IpAppChargingSession**, reserve amount, unsuccessful

Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2. Method call **reserveAmountErr()**
   Parameters: sessionID, requestNumber, error, requestNumberNextRequest
   Check: no exception is returned

Test CH_APP_CS_03

Summary: **IpAppChargingSession**, reserve amount and credit it, successful

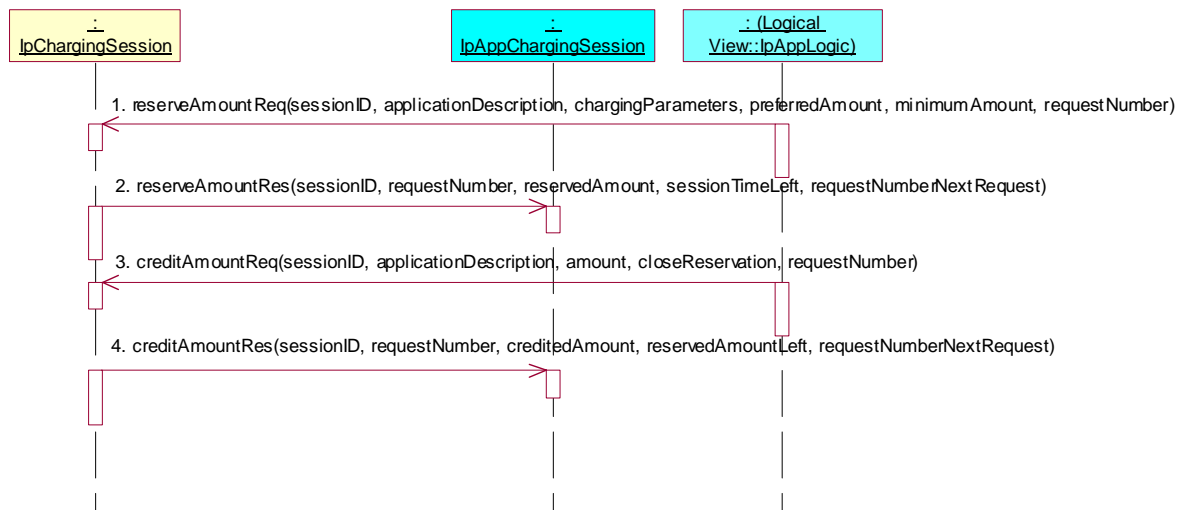Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **creditAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

   1.   Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession**
        interface.
        Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount,
                       requestNumber

   2.   Method call **reserveAmountRes()**
        Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
        Check: no exception is returned

   3.   Triggered action: cause IUT to call **creditAmountReq()** method on the tester's (SCF) **IpChargingSession**
        interface.
        Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber

   4.   Method call **creditAmountRes()**
        Parameters: sessionID, requestNumber, creditedAmount, reservedAmountLeft, requestNumberNextRequest
        Check: no exception is returned

Test CH_APP_CS_04

Summary: **IpAppChargingSession**, reserve amount and credit it, unsuccessful

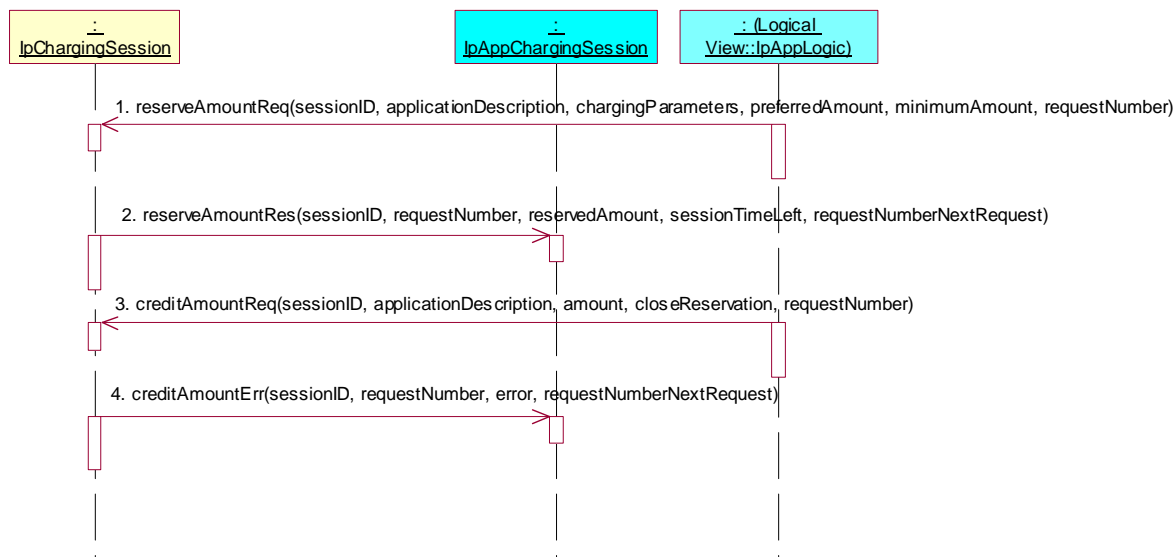Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **creditAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2.   Method call **reserveAmountRes()**
     Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
     Check: no exception is returned

3.   Triggered action: cause IUT to call **creditAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber

4.   Method call **creditAmountErr()**
     Parameters: sessionID, requestNumber, error, requestNumberNextRequest
     Check: no exception is returned

Test CH_APP_CS_05

Summary: **IpAppChargingSession**, reserve amount and debit it, successful

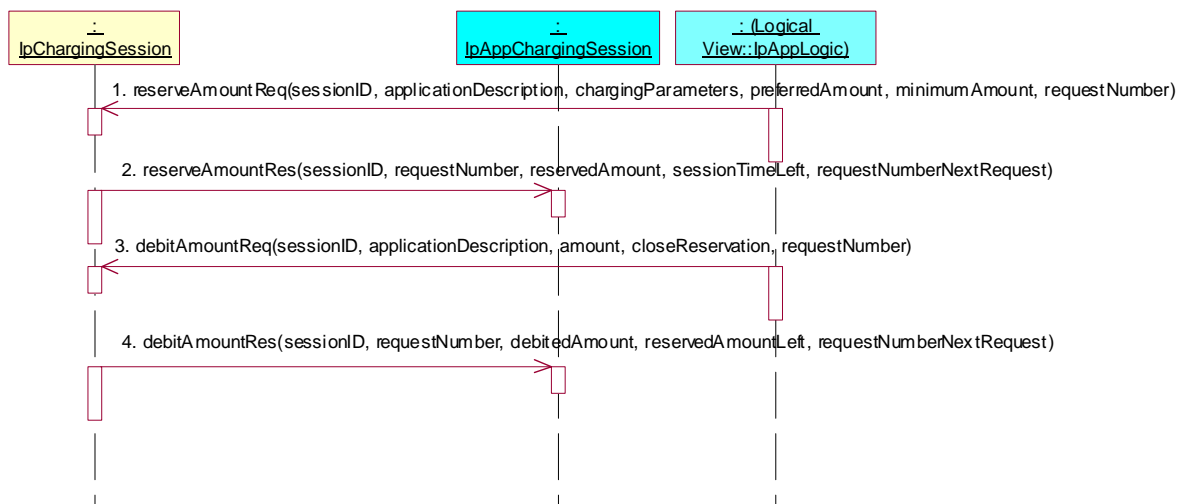Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **debitAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

   1.   Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession**
        interface.
        Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount,
                        requestNumber

   2.   Method call **reserveAmountRes()**
        Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
        Check: no exception is returned

   3.   Triggered action: cause IUT to call **debitAmountReq()** method on the tester's (SCF) **IpChargingSession**
        interface.
        Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber

   4.   Method call **debitAmountRes()**
        Parameters: sessionID, requestNumber, creditedAmount, reservedAmountLeft, requestNumberNextRequest
        Check: no exception is returned

Test CH_APP_CS_06

Summary: **IpAppChargingSession**, reserve amount and debit it, unsuccessful

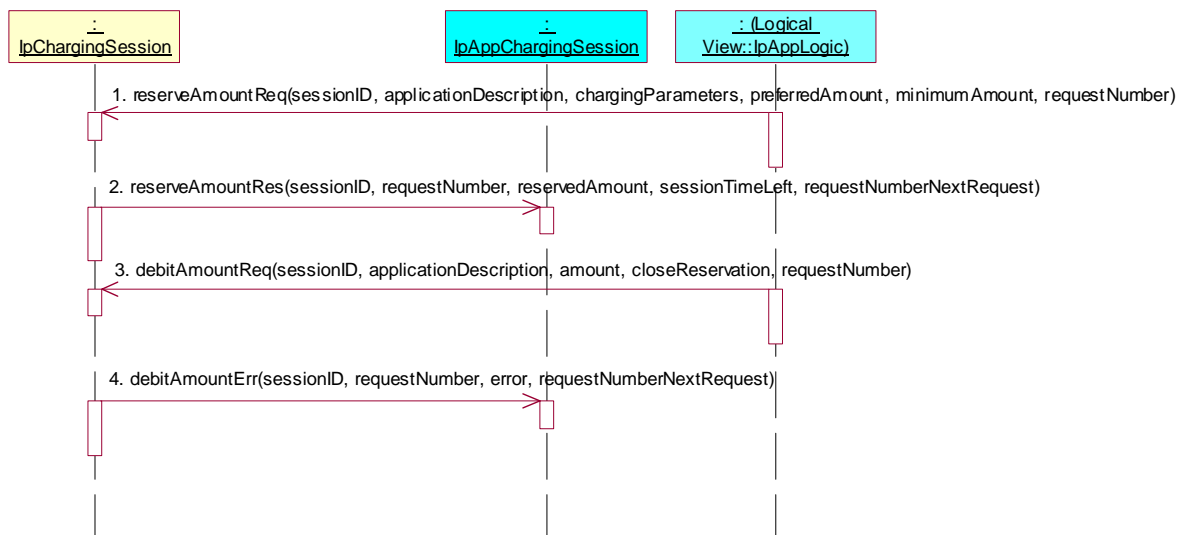Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **debitAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2. Method call **reserveAmountRes()**
   Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
   Check: no exception is returned

3. Triggered action: cause IUT to call **debitAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, amount, closeReservation, requestNumber

4. Method call **debitAmountErr()**
   Parameters: sessionID, requestNumber, error, requestNumberNextRequest
   Check: no exception is returned

ETSI

Test CH_APP_CS_07

Summary: **IpAppChargingSession**, reserve amount and extend lifetime of the reservation, successful

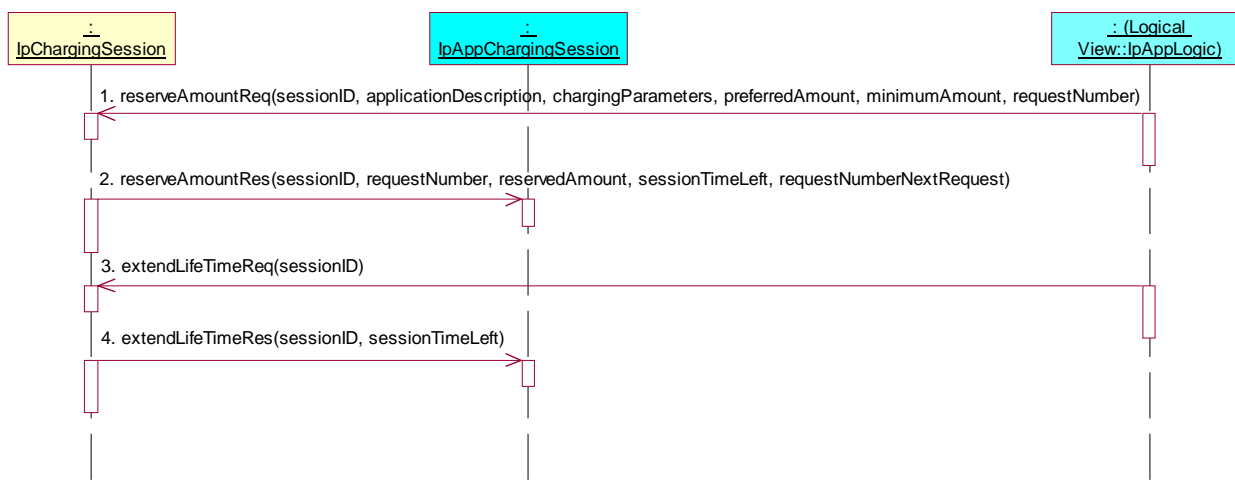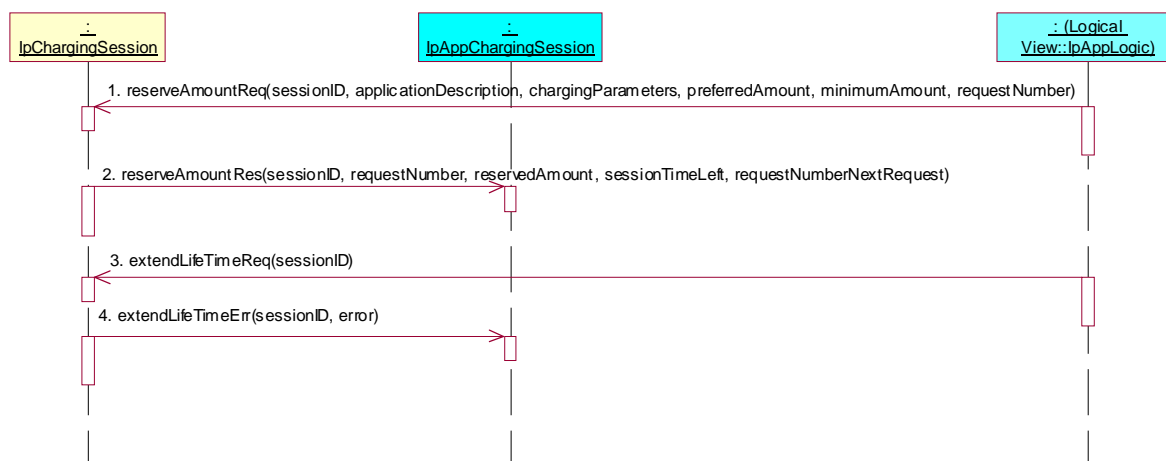Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **extendLifeTimeReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                 have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                 service and signing the required service agreement.

                 Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession**
    interface.
    Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount,
                    requestNumber

2. Method call **reserveAmountRes()**
    Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
    Check: no exception is returned

3. Triggered action: cause IUT to call **extendLifeTimeReq()** method on the tester's (SCF) **IpChargingSession**
    interface.
    Parameters: sessionID

4. Method call **extendLifeTimeRes()**
    Parameters: sessionID, sessionTimeLeft
    Check: no exception is returned

Test CH_APP_CS_08

Summary: **IpAppChargingSession**, reserve amount and extend lifetime of the reservation, unsuccessful

Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **extendLifeTimeReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                 Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2.   Method call **reserveAmountRes()**
     Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
     Check: no exception is returned

3.   Triggered action: cause IUT to call **extendLifeTimeReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID

4.   Method call **extendLifeTimeErr()**
     Parameters: sessionID, error
     Check: no exception is returned

Test CH_APP_CS_09

Summary: **IpAppChargingSession**, reserve amount and request amount left
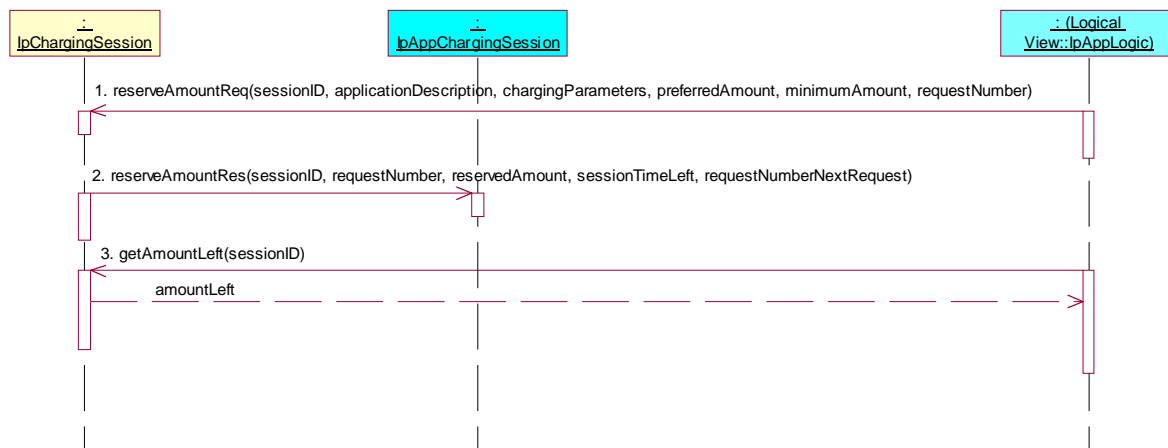
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **getAmountLeft()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2.   Method call **reserveAmountRes()**
     Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
     Check: no exception is returned

3.   Triggered action: cause IUT to call **getAmountLeft()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID

Test CH_APP_CS_10

Summary: **IpAppChargingSession**, reserve amount and request lifetime left
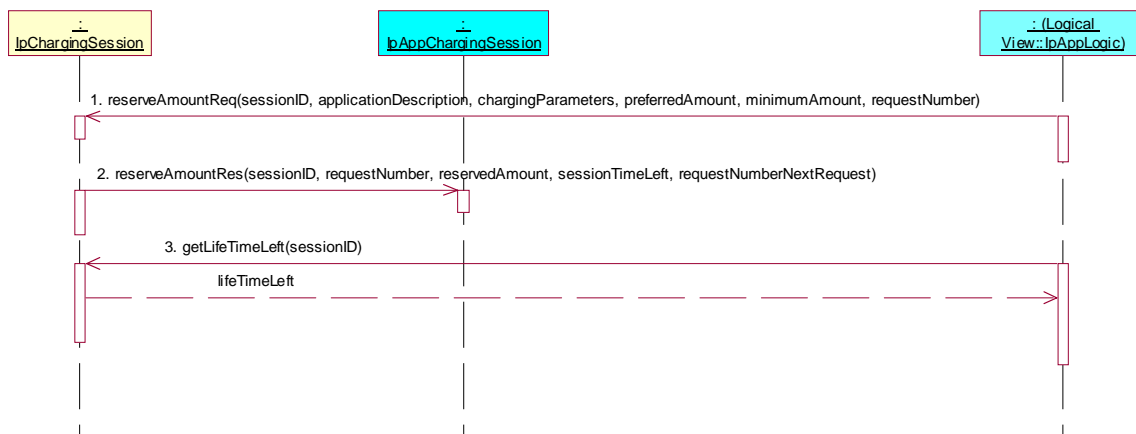
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **getLifeTimeLeft()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2. Method call **reserveAmountRes()**
   Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
   Check: no exception is returned

3. Triggered action: cause IUT to call **getLifeTimeLeft()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID

Test CH_APP_CS_11

Summary: **IpAppChargingSession**, reserve amount and release charging session

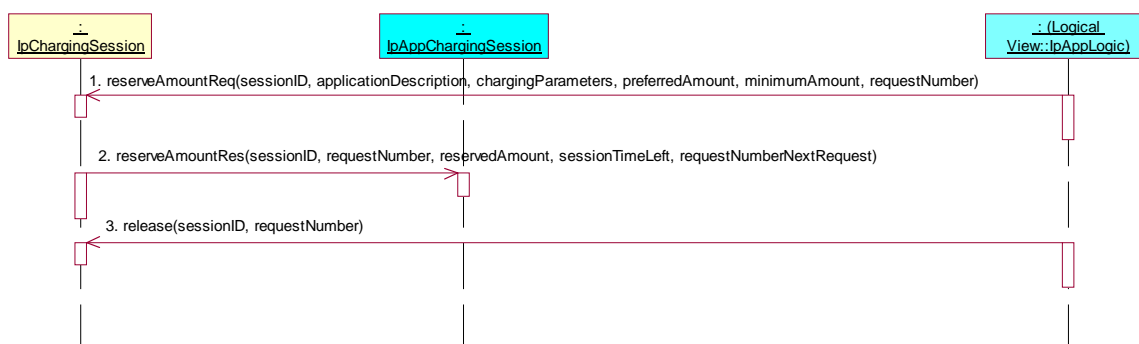Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveAmountReq()** and **release()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2. Method call **reserveAmountRes()**
    Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
    Check: no exception is returned

3. Triggered action: cause IUT to call **release()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID, requestNumber

Test CH_APP_CS_12

Summary: **IpAppChargingSession**, reserve amount and accept charging session abort

Reference: ES 201 915-12 [1], clause 8.1
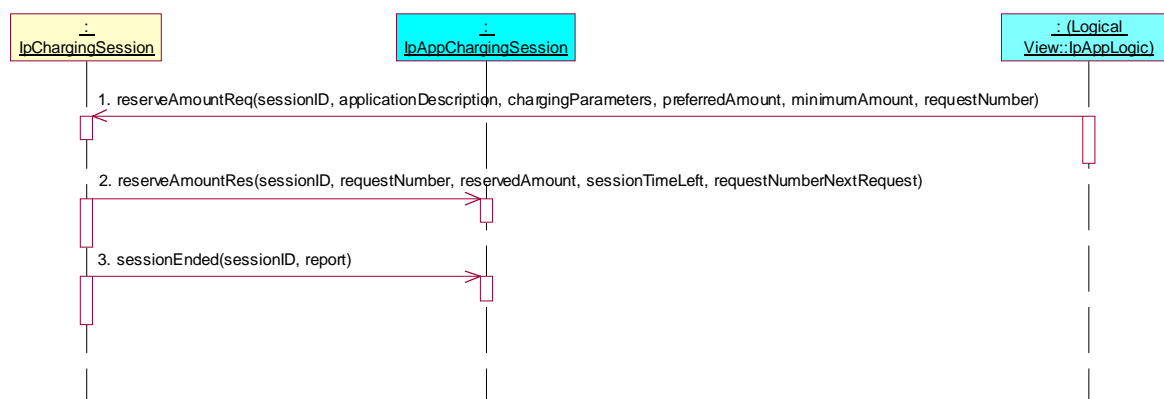
Precondition: IUT capable of invoking **reserveAmountReq()**, **sessionEnded** implemented

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                  Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, preferredAmount, minimumAmount, requestNumber

2. Method call **reserveAmountRes()**
   Parameters: sessionID, requestNumber, reservedAmount, sessionTimeLeft, requestNumberNextRequest
   Check: no exception is returned

3. Method call **sessionEnded()**
   Parameters: sessionID, report
   Check: no exception is returned

Test CH_APP_CS_13

Summary: **IpAppChargingSession**, reserve units, successful

Reference: ES 201 915-12 [1], clause 8.1
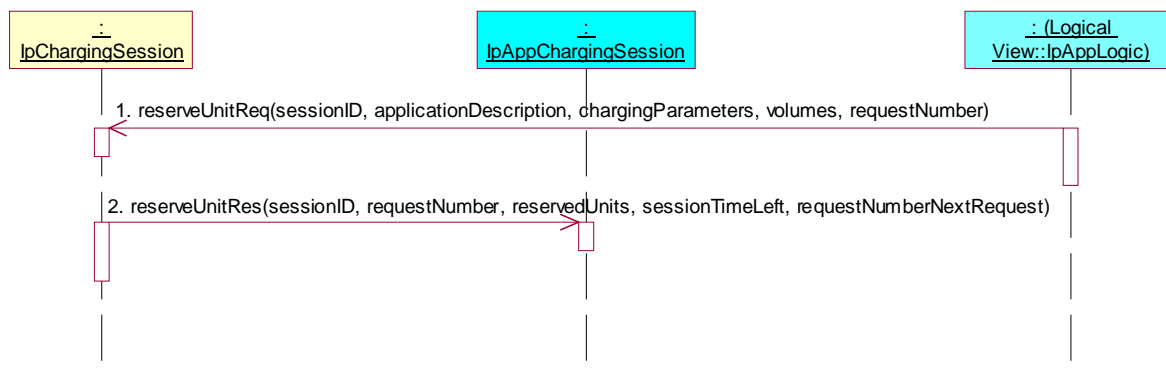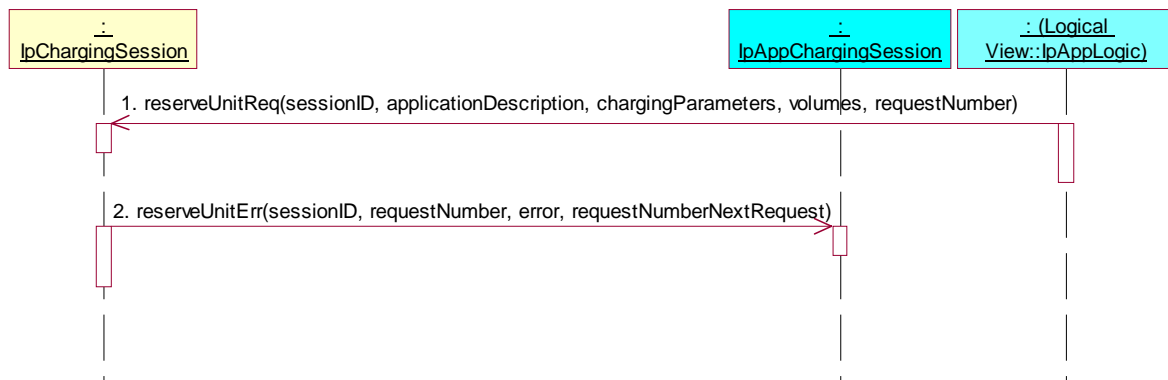
Precondition: IUT capable of invoking **reserveUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession**
      interface.
      Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **reserveUnitRes()**
      Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
      Check: no exception is returned

Test CH_APP_CS_14

Summary: **IpAppChargingSession**, reserve units, unsuccessful

Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession**
    interface.
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2. Method call **reserveUnitErr()**
    Parameters: sessionID, requestNumber, error, requestNumberNextRequest
    Check: no exception is returned

Test CH_APP_CS_15

Summary: **IpAppChargingSession**, reserve and credit units, successful

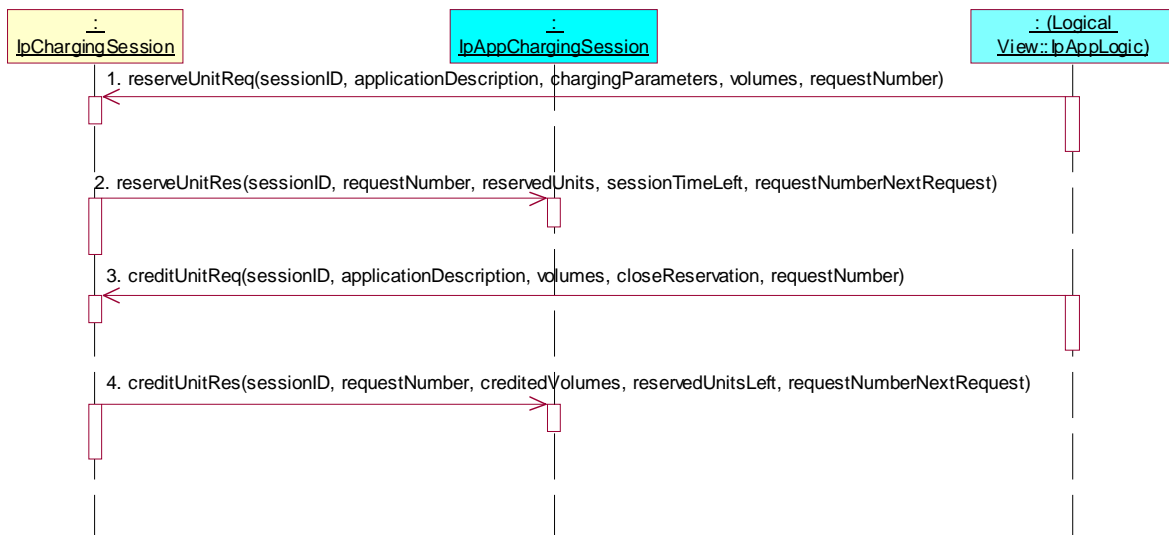Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **creditUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

   1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession**
        interface.
        Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

   2.   Method call **reserveUnitRes()**
        Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
        Check: no exception is returned

   3.   Triggered action: cause IUT to call **creditUnitReq()** method on the tester's (SCF) **IpChargingSession**
        interface.
        Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber

   4.   Method call **creditUnitRes()**
        Parameters: sessionID, requestNumber, creditedVolumes, reservedUnitsLeft, requestNumberNextRequest
        Check: no exception is returned

Test CH_APP_CS_16

Summary: **IpAppChargingSession**, reserve and credit units, unsuccessful

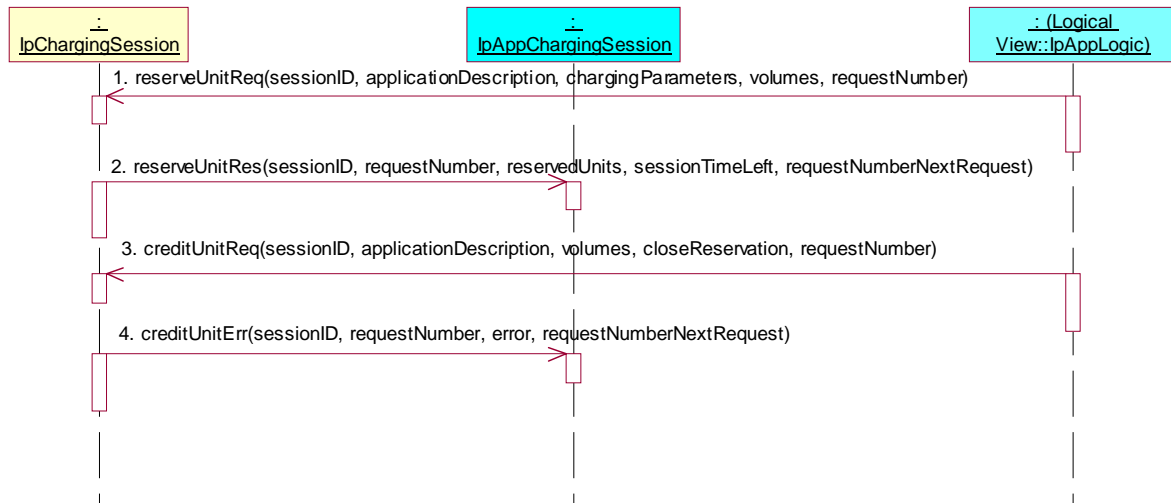Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **creditUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2. Method call **reserveUnitRes()**
   Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
   Check: no exception is returned

3. Triggered action: cause IUT to call **creditUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber

4. Method call **creditUnitErr()**
   Parameters: sessionID, requestNumber, error, requestNumberNextRequest
   Check: no exception is returned

Test CH_APP_CS_17

Summary: **IpAppChargingSession**, reserve and debit units, successful

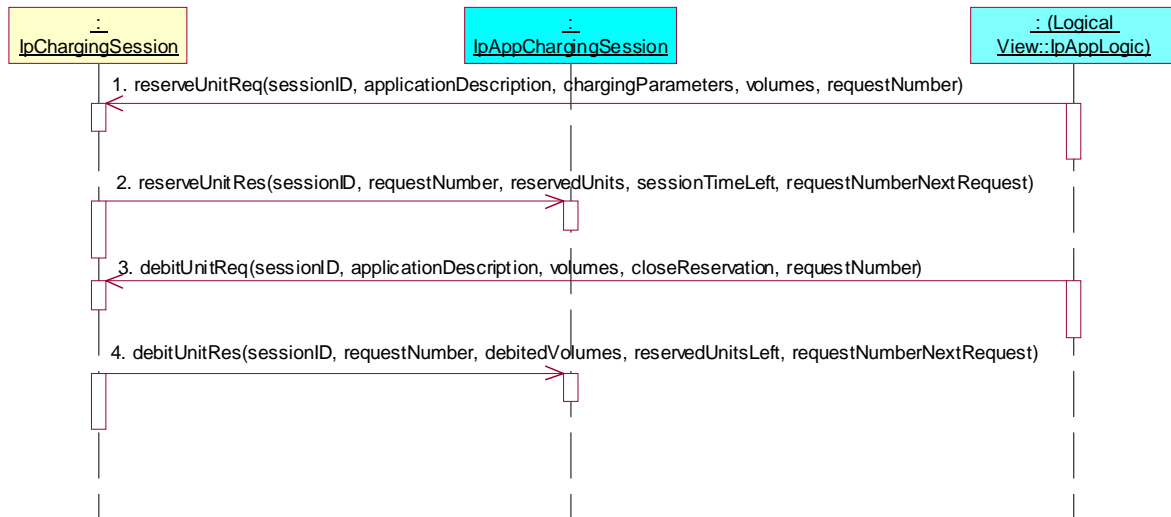Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **debitUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **reserveUnitRes()**
     Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
     Check: no exception is returned

3.   Triggered action: cause IUT to call **debitUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber

4.   Method call **debitUnitRes()**
     Parameters: sessionID, requestNumber, debitedVolumes, reservedUnitsLeft, requestNumberNextRequest
     Check: no exception is returned

Test CH_APP_CS_18

Summary: **IpAppChargingSession**, reserve and debit units, unsuccessful

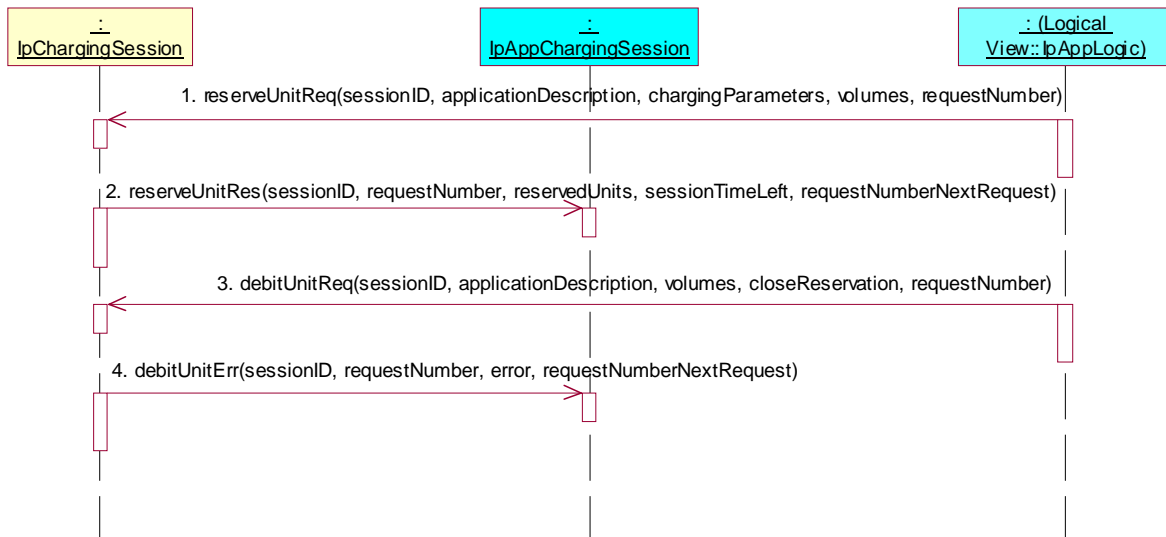Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **debitUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **reserveUnitRes()**
     Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
     Check: no exception is returned

3.   Triggered action: cause IUT to call **debitUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
     Parameters: sessionID, applicationDescription, volumes, closeReservation, requestNumber

4.   Method call **debitUnitErr()**
     Parameters: sessionID, requestNumber, error, requestNumberNextRequest
     Check: no exception is returned

Test CH_APP_CS_19

Summary: **IpAppChargingSession**, reserve units and extend lifetime of the reservation, successful

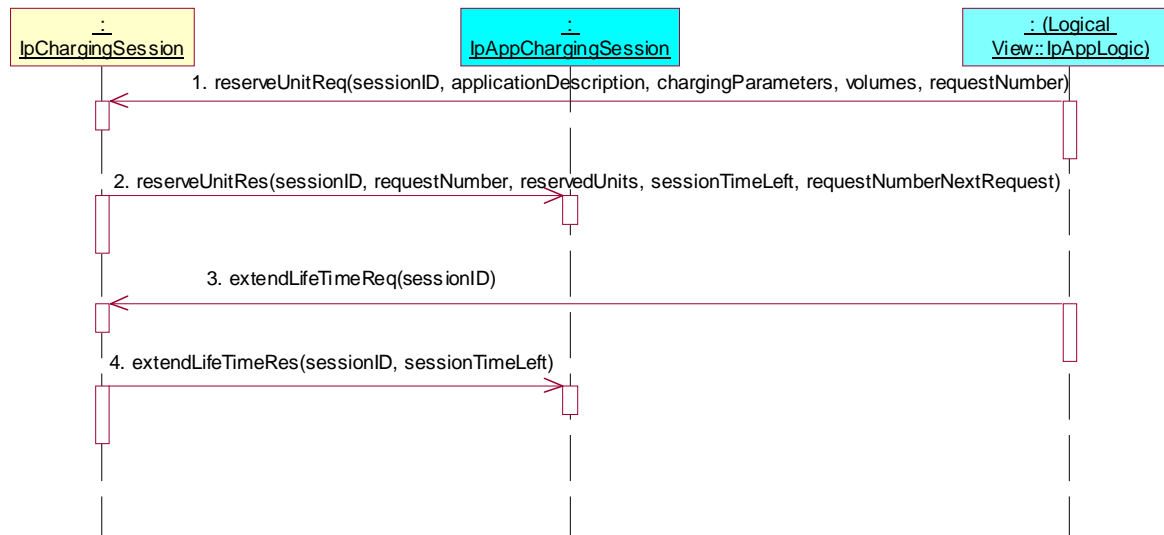Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **extendLifeTimeReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
      Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **reserveUnitRes()**
      Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
      Check: no exception is returned

3.   Triggered action: cause IUT to call **extendLifeTimeReq()** method on the tester's (SCF) **IpChargingSession** interface.
      Parameters: sessionID

4.   Method call **extendLifeTimeRes()**
      Parameters: sessionID, sessionTimeLeft
      Check: no exception is returned

Test CH_APP_CS_20

Summary: **IpAppChargingSession**, reserve units and extend lifetime of the reservation, unsuccessful

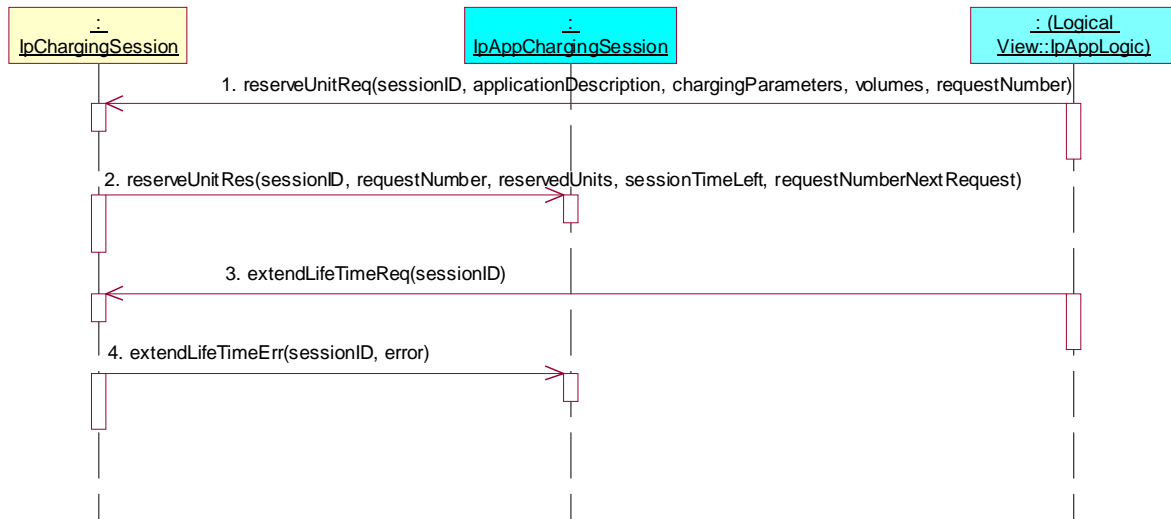Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **extendLifeTimeReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2. Method call **reserveUnitRes()**
    Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
    Check: no exception is returned

3. Triggered action: cause IUT to call **extendLifeTimeReq()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID

4. Method call **extendLifeTimeErr()**
    Parameters: sessionID, error
    Check: no exception is returned

Test CH_APP_CS_21

Summary: **IpAppChargingSession**, reserve units and request units left

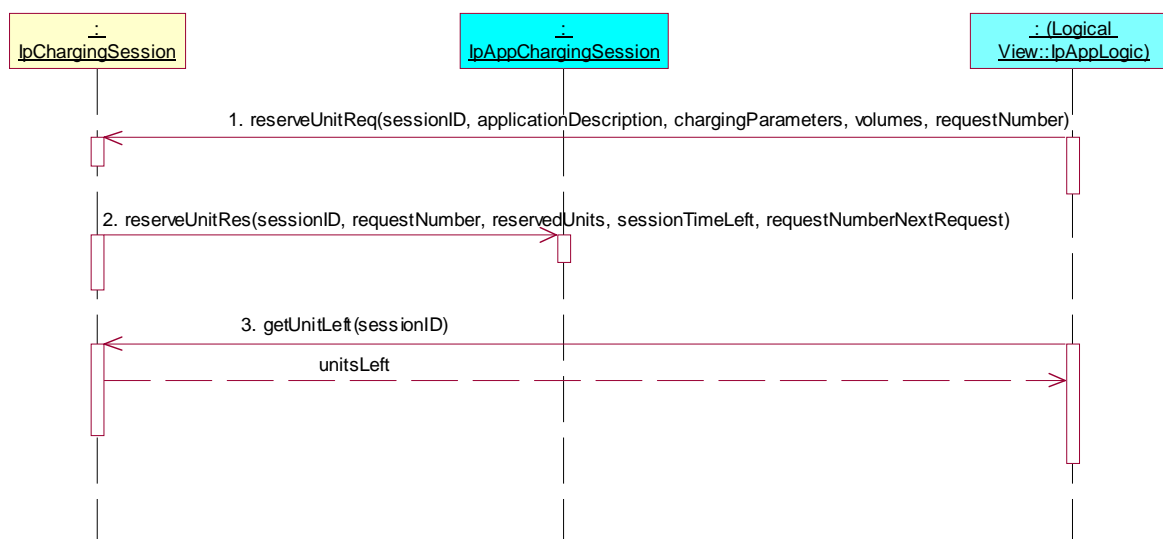Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **getUnitLeft()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                 have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                 service and signing the required service agreement.

                 Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession**
      interface.
      Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **reserveUnitRes()**
      Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
      Check: no exception is returned

3.   Triggered action: cause IUT to call **getUnitLeft()** method on the tester's (SCF) **IpChargingSession** interface.
      Parameters: sessionID

Test CH_APP_CS_22

Summary: **IpAppChargingSession**, reserve units and request lifetime left
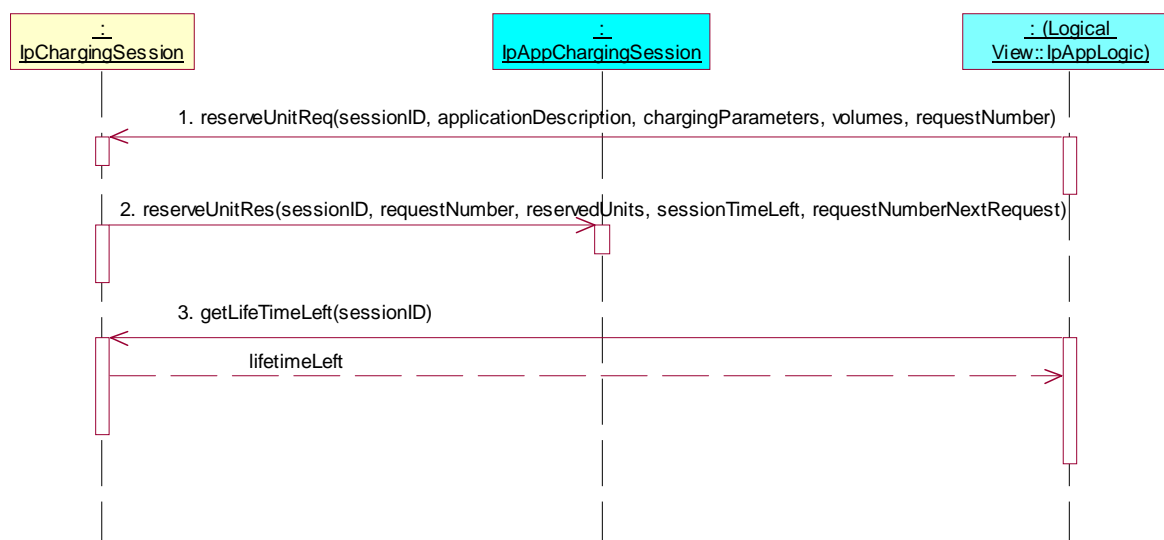
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **getLifeTimeLeft()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

   1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
        Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

   2.   Method call **reserveUnitRes()**
        Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
        Check: no exception is returned

   3.   Triggered action: cause IUT to call **getLifeTimeLeft()** method on the tester's (SCF) **IpChargingSession** interface.
        Parameters: sessionID

Test CH_APP_CS_23

Summary: **IpAppChargingSession**, reserve units and release charging session

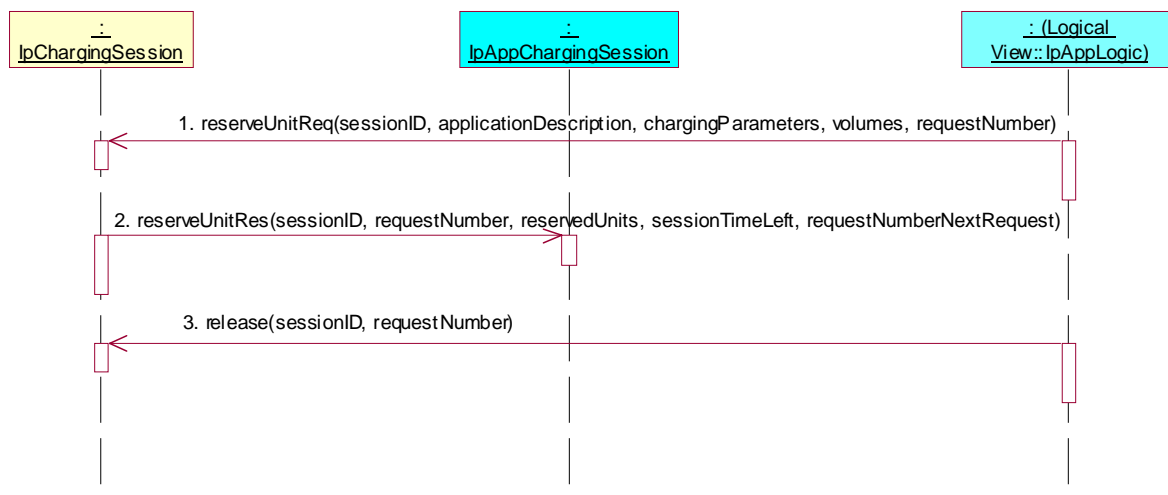Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()** and **release()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
      Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **reserveUnitRes()**
      Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
      Check: no exception is returned

3.   Triggered action: cause IUT to call **release()** method on the tester's (SCF) **IpChargingSession** interface.
      Parameters: sessionID, requestNumber

Test CH_APP_CS_24

Summary: **IpAppChargingSession**, reserve units and accept charging session abort
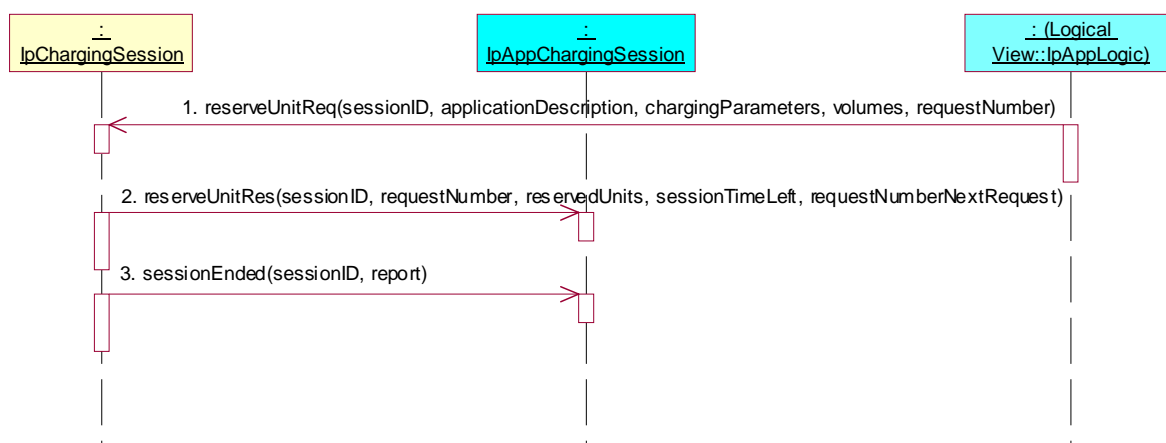
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **reserveUnitReq()**, **sessionEnded** implemented

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **reserveUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2. Method call **reserveUnitRes()**
   Parameters: sessionID, requestNumber, reservedUnits, sessionTimeLeft, requestNumberNextRequest
   Check: no exception is returned

3. Method call **sessionEnded()**
   Parameters: sessionID, report
   Check: no exception is returned

Test CH_APP_CS_25

Summary: **IpAppChargingSession**, credit amount directly, successful
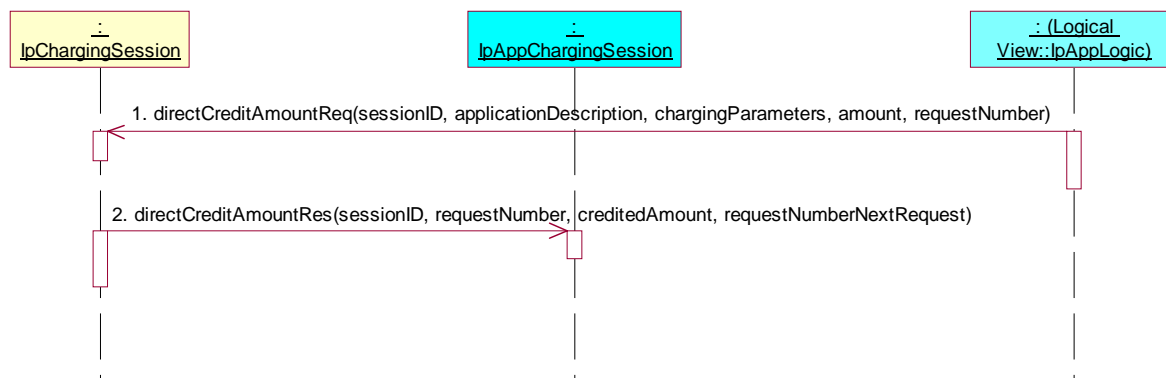
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directCreditAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **directCreditAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber

2. Method call **directCreditAmountRes()**
    Parameters: sessionID, requestNumber, creditedAmount, requestNumberNextRequest
    Check: no exception is returned

Test CH_APP_CS_26

Summary: **IpAppChargingSession**, credit amount directly, unsuccessful
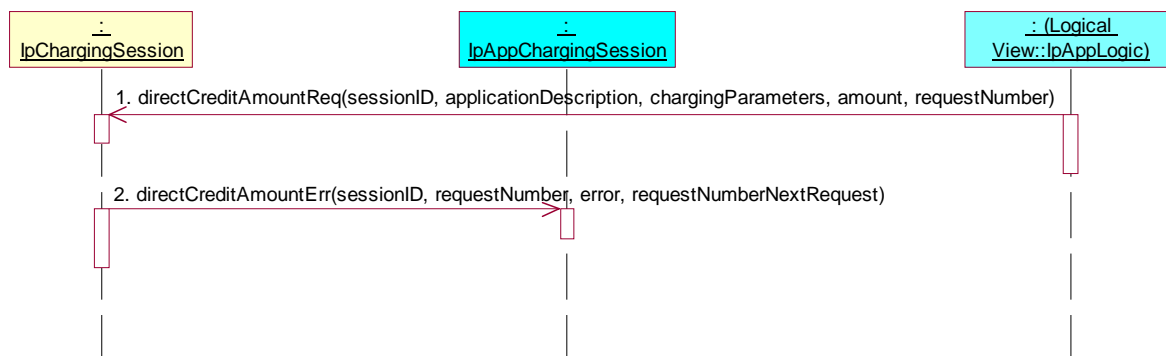
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directCreditAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **directCreditAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber

2. Method call **directCreditAmountErr()**
   Parameters: sessionID, requestNumber, error
   Check: no exception is returned

Test CH_APP_CS_27

Summary: **IpAppChargingSession**, debit amount directly, successful

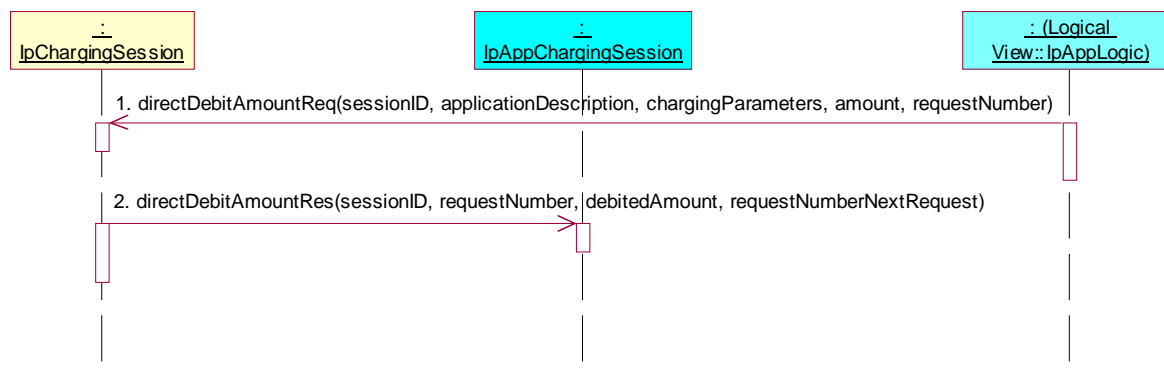Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directDebitAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
               have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
               service and signing the required service agreement.

               Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **directDebitAmountReq()** method on the tester's (SCF)
     IpChargingSession interface.
     Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber

2.   Method call **directDebitAmountRes()**
     Parameters: sessionID, requestNumber, debitedAmount, requestNumberNextRequest
     Check: no exception is returned

Test CH_APP_CS_28

Summary: **IpAppChargingSession**, debit amount directly, unsuccessful
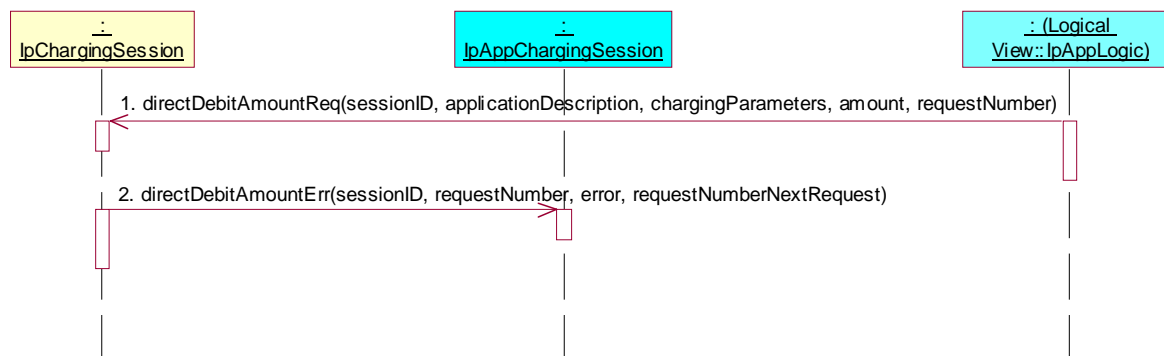
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directDebitAmountReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **directDebitAmountReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, applicationDescription, chargingParameters, amount, requestNumber

2. Method call **directDebitAmountErr()**
   Parameters: sessionID, requestNumber, error
   Check: no exception is returned

Test CH_APP_CS_29

Summary: **IpAppChargingSession**, credit units directly, successful

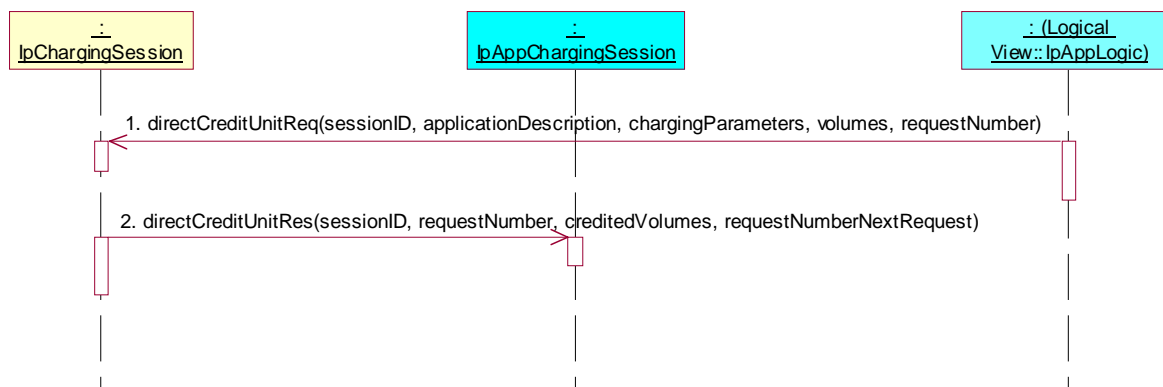Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directCreditUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **directCreditUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2. Method call **directCreditUnitRes()**
    Parameters: sessionID, requestNumber, creditedVolumes, requestNumberNextRequest
    Check: no exception is returned

Test CH_APP_CS_30

Summary: **IpAppChargingSession**, credit units directly, unsuccessful

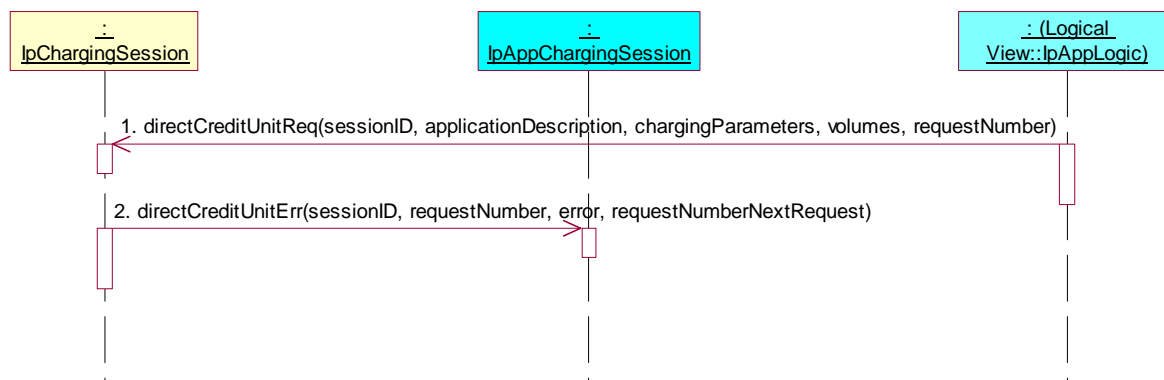Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directCreditUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **directCreditUnitReq()** method on the tester's (SCF) **IpChargingSession**
     interface.
     Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **directCreditUnitErr()**
     Parameters: sessionID, requestNumber, error
     Check: no exception is returned

Test CH_APP_CS_31

Summary: **IpAppChargingSession**, debit units directly, successful

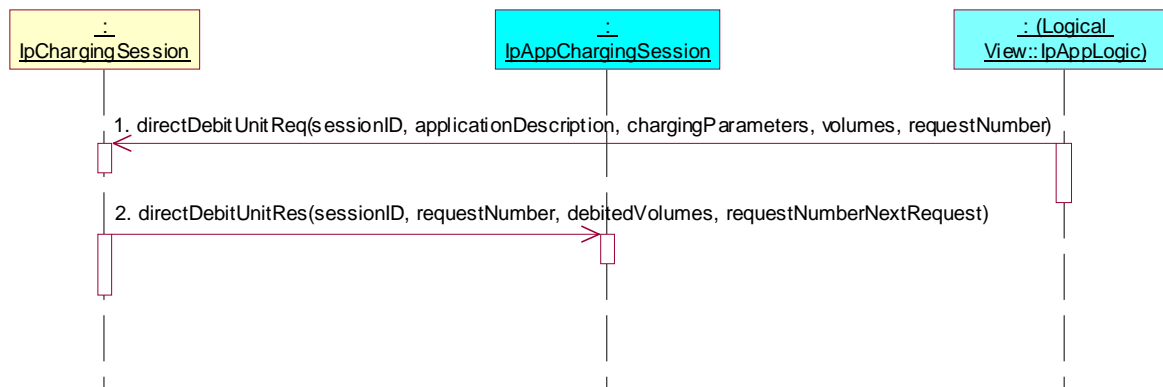Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directDebitUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

                  Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **directDebitUnitReq()** method on the tester's (SCF) **IpChargingSession** interface.
    Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2. Method call **directDebitUnitRes()**
    Parameters: sessionID, requestNumber, debitedVolumes, requestNumberNextRequest
    Check: no exception is returned

Test CH_APP_CS_32

Summary: **IpAppChargingSession**, debit units directly, unsuccessful

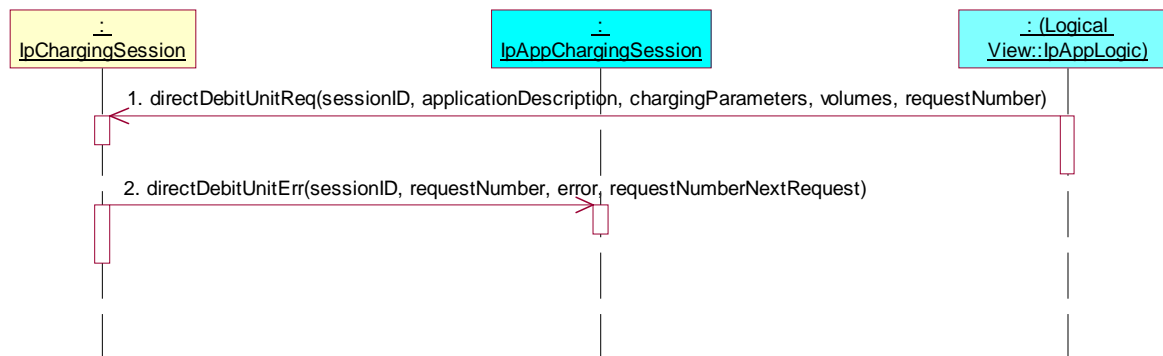Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **directDebitUnitReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must
                have obtained a reference to an instance of the IpAppChargingSession interface through selecting that
                service and signing the required service agreement.

                Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **directDebitUnitReq()** method on the tester's (SCF) **IpChargingSession**
     interface.
     Parameters: sessionID, applicationDescription, chargingParameters, volumes, requestNumber

2.   Method call **directDebitUnitErr()**
     Parameters: sessionID, requestNumber, error
     Check: no exception is returned

Test CH_APP_CS_33

Summary: **IpAppChargingSession**, request rate, successful
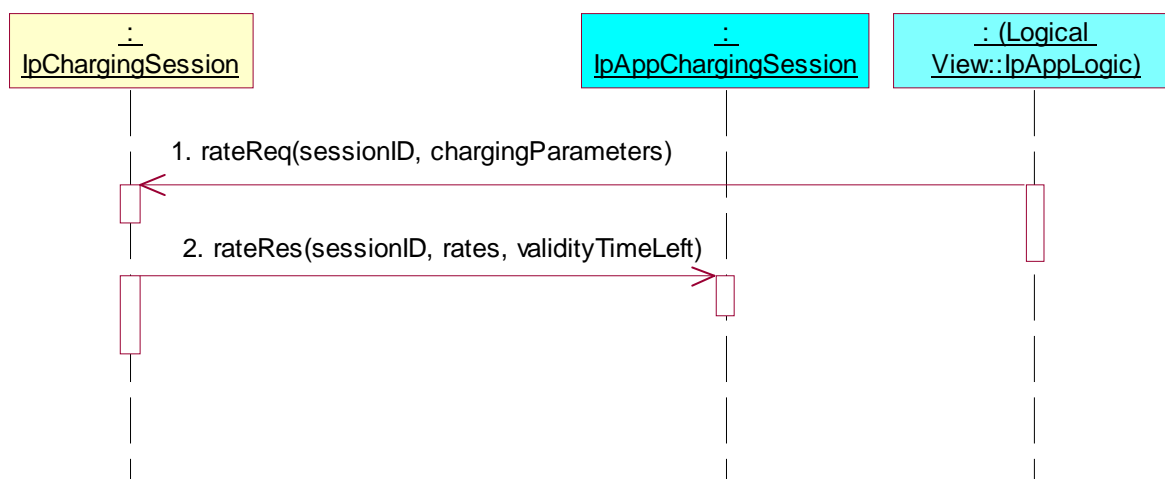
Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **rateReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1. Triggered action: cause IUT to call **rateReq()** method on the tester's (SCF) **IpChargingSession** interface.
   Parameters: sessionID, chargingParameters

2. Method call **rateRes()**
   Parameters: sessionID, rates, validityTimeLeft
   Check: no exception is returned

Test CH_APP_CS_34

Summary: **IpAppChargingSession**, request rate, unsuccessful

Reference: ES 201 915-12 [1], clause 8.1

Precondition: IUT capable of invoking **rateReq()**

Preamble: Registration of the IUT (application) and the tester (Charging SCF) to the framework. The tester must have obtained a reference to an instance of the IpAppChargingSession interface through selecting that service and signing the required service agreement.

Successful call of the createChargingSession() method on the IpChargingManager interface.

Test Sequence:

1.   Triggered action: cause IUT to call **rateReq()** method on the tester's (SCF) **IpChargingSession** interface.
      Parameters: sessionID, chargingParameters

2.   Method call **rateErr()**
      Parameters: sessionID, error
      Check: no exception is returned
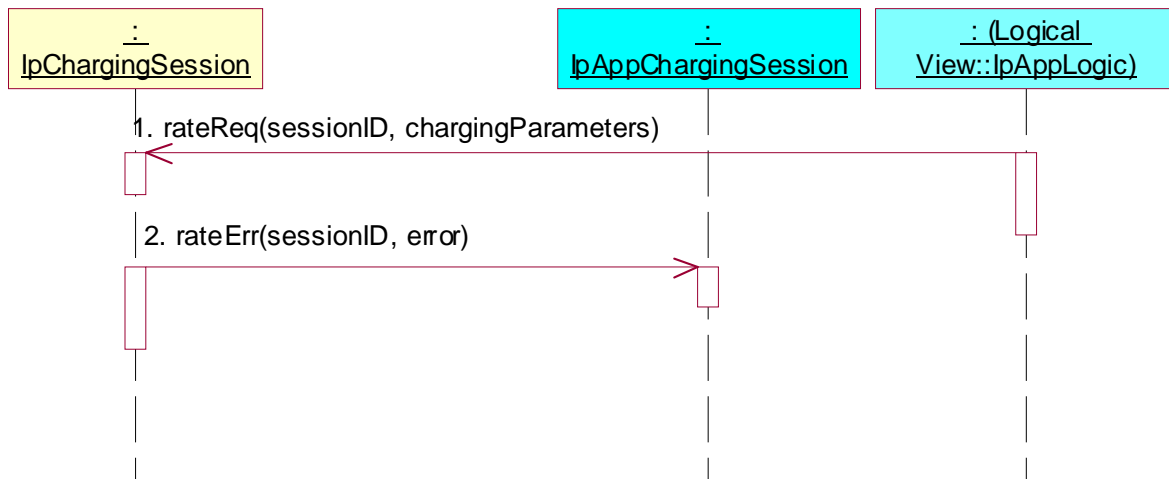
History

Document history		
V1.1.1	August 2003	Publication
V1.2.1	December 2004	Membership Approval Procedure MV 20050211: 2004-12-14 to 2005-02-11
V1.2.1	February 2005	Publication