

# ETSI EN 302 039-2 V1.1.1 (2002-04)

---

*European Standard (Telecommunications series)*

**Intelligent Network (IN);  
Intelligent Network Capability Set 4 (CS4);  
Intelligent Network Application Protocol (INAP);  
Protocol specification;  
Part 2: SSF-SCF Interface**

---



---

Reference

DEN/SPAN-120065-2

---

Keywords

CS4, IN, INAP, protocol

**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, send your comment to:

[editor@etsi.fr](mailto:editor@etsi.fr)

---

**Copyright Notification**

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

© European Telecommunications Standards Institute 2002.  
All rights reserved.

**DECT**<sup>TM</sup>, **PLUGTESTS**<sup>TM</sup> and **UMTS**<sup>TM</sup> are Trade Marks of ETSI registered for the benefit of its Members.  
**TIPHON**<sup>TM</sup> and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.  
**3GPP**<sup>TM</sup> is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

# Contents

Intellectual Property Rights .....	4
Foreword.....	4
1 Scope .....	5
2 References .....	5
3 Abbreviations .....	5
4 Operation procedures .....	5
4.1 Modified operations .....	5
4.1.1 Connect .....	6
4.1.2 ContinueWithArgument.....	6
4.1.3 InitialDP.....	6
4.1.4 InitiateCallAttempt .....	6
4.1.5 MergeCallSegments.....	6
4.1.6 MoveLeg.....	6
4.1.7 SelectFacility .....	6
4.1.8 SplitLeg .....	6
4.2 New operations.....	6
4.2.1 CallFiltering procedure .....	7
4.2.1.1 General description .....	7
4.2.1.2 Argument Parameters.....	7
4.2.1.3 Invoking entity (SCF) .....	7
4.2.1.3.1 Normal procedure.....	7
4.2.1.3.2 Error handling.....	7
4.2.1.4 Responding entity (SSF) .....	7
4.2.1.4.1 Normal procedure.....	7
4.2.1.4.2 Error handling.....	8
5 Parameter descriptions .....	8
5.1 DetachSignallingPath .....	8
5.2 DestinationIndex .....	8
5.3 ExportSignallingPath.....	8
5.4 IncomingSignallingBufferCopy .....	8
5.5 IPRelatedInformation .....	9
5.6 MergeSignallingPaths .....	9
6 ASN.1 definitions.....	9
6.1 Data Types.....	9
6.2 Classes .....	34
6.3 Operations and Arguments .....	38
6.4 Operation timers .....	60
6.5 Packages, contracts, application contexts and abstract syntaxes .....	61
6.5.1 ASN.1 modules.....	61
History .....	77

---

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

The present document is part 2 of a multi-part deliverable covering Intelligent Network (IN); Intelligence Network Capability Set 4 (CS4); Intelligent Network Application Protocol (INAP); Protocol specification, as identified below:

Part 1: "Common aspects";

**Part 2: "SSF-SCF Interface".**

The present document describes the enhancement for SSF-SCF interface.

The present document and EN 302 039-1 [1] define the Intelligent Network (IN) Application Protocol (INAP) for IN Capability Set-4 based and written as delta documents upon ETSI Core INAP CS-3 (EN 301 931-1 [3] and EN 301 931-2 [4]).

This set of documents define enhancements made on the SSF to SCF interface (the present document) as a subset of the ITU-T IN CS4 Recommendations Q.1248.1 [7], Q.1248.2 [8]. For the other interfaces, the ETSI Core INAP CS3 series of EN 301 931 apply.

<b>Proposed national transposition dates</b>	
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 specifies the protocol enhancements on the SSF-SCF interface for IN CS4.

---

## 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.
- [1] ETSI EN 302 039-1: "Intelligent Network (IN); Intelligent Network Capability Set 4 (CS4); Intelligent Network Application Protocol (INAP); Protocol specification; Part 1: Common aspects".
  - [2] ETSI EN 301 140-1: "Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 2 (CS2); Part 1: Protocol specification".
  - [3] ETSI EN 301 931-1: "Intelligent Network (IN); Intelligent Network Capability Set 3 (CS3); Intelligent Network Application Protocol (INAP); Protocol specification; Part 1: Common aspects".
  - [4] ETSI EN 301 931-2: "Intelligent Network (IN); Intelligent Network Capability Set 3 (CS3); Intelligent Network Application Protocol (INAP); Protocol specification; Part 2: SCF-SSF interface".
  - [5] ITU-T Recommendation E.410: "International network management - General information".
  - [6] ITU-T Recommendation Q.1236: "Intelligent Network Capability Set 3 - Management Information Model Requirements and Methodology".
  - [7] ITU-T Recommendation Q.1248.1: "Interface recommendation for Intelligent Network Capability Set 4 : Interface Recommendation for Intelligent Network Capability Set 4 - Common aspects".
  - [8] ITU-T Recommendation Q.1248.2: "Interface recommendation for Intelligent Network Capability Set 4: SCF-SSF Interface".
- 

## 3 Abbreviations

For the purposes of the present document, the abbreviations given in EN 301 931-1 [3] apply.

---

## 4 Operation procedures

EN 301 931-2 [4], clause 11 is applicable with the enhancements specified in the present document.

### 4.1 Modified operations

The following operations defined in EN 301 931-2 [4] are modified by the present document.

### 4.1.1 Connect

The following parameter defined in clause 5 is added to the operation argument:

- IpRelatedInformation.

Existing cug-Interlock and cug-OutgoingAcces parameters are no longer restricted for the support of CAMEL.

### 4.1.2 ContinueWithArgument

The following parameter defined in clause 5 is added to the operation argument:

- ipRelatedInformation.

### 4.1.3 InitialDP

The following parameters defined in clause 5 is added to the operation argument:

- ipRelatedInformation.

### 4.1.4 InitiateCallAttempt

The following parameters defined in clause 5 are added to the operation argument:

- incomingSignallingBufferCopy.
- ipRelatedInformation.

### 4.1.5 MergeCallSegments

The following parameter defined in clause 5 is added to the operation argument:

- mergeSignallingPaths.

### 4.1.6 MoveLeg

The following parameters defined in clause 5 are added to the operation argument:

- detachSignallingPath.
- exportSignallingPath.

### 4.1.7 SelectFacility

The following parameter defined in clause 5 is added to the operation argument:

- ipRelatedInformation.

### 4.1.8 SplitLeg

The following parameter defined in clause 5 is added to the operation argument:

- detachSignallingPath.

## 4.2 New operations

The following new operation is defined by the present document.

## 4.2.1 CallFiltering procedure

### 4.2.1.1 General description

The CallFiltering operation is used to allow the SCF to influence basic call gapping procedures based in the CCF by sending information from the SCF to the SSF. The SSF relays the received information transparently to the CCF. This way, the SCF can influence the rate at which call attempts are allowed through. The operation thus influences the filtering of calls, as opposed to service requests as is done by the Callgap operation.

### 4.2.1.2 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 13.

- destination Index  
This index is a pointer to the Destination (see ITU-T Recommendation E.410 [5]) to which calls are filtered.
- gapIndicators  
The parameter contains the gapDuration and the gapInterval.
- registratorIdentifier  
This parameter identifies the SCF and is to be used by the SSF to verify that the SCF is allowed to influence CCF-based call gapping procedures.
- extensions

### 4.2.1.3 Invoking entity (SCF)

#### 4.2.1.3.1 Normal procedure

SCF precondition:

- 1) The SCF receives an indication from the SMF an overload condition persists and callfiltering has to be initiated at the SSF.

SCF postcondition:

- 1) The SCME is in the state "idle".

If the congestion level changes new "CallFiltering" operations may be sent for active filter criteria but with a new filter interval. If the congestion situation has ended, the filtering criteria may be removed.

#### 4.2.1.3.2 Error handling

Generic error handling for the operation related error are described in clause 13 of EN 301 931-2 [4] and the TCAP services which are used for reporting operation errors as described in clause 10 of EN 301 931-1 [3].

### 4.2.1.4 Responding entity (SSF)

#### 4.2.1.4.1 Normal procedure

SSF precondition:

- 1) None.

SSF postcondition:

- 1) SSME-FSM is in the state "Idle".

The SSF relays the received information transparently to the CCF-based call filtering process. In case callfiltering to the specific destinations is already active at the CCF, then the new gapIndicator parameter overwrites the existing parameter values.

A manual initiated call filter will prevail over an automatic initiated call filter.

If a call matches several destinationIndexes, then the control corresponding with the most specific destinationIndex should be applied.

The service request gap process is stopped if the indicated duration equals zero.

#### 4.2.1.4.2 Error handling

Generic error handling for the operation related error are described in clause 13 of EN 301 931-2 [4] and the TCAP services which are used for reporting operation errors as described in clause 10 of EN 301 931-1 [3].

NOTE: In case of error (i.e. invalid registrator identifier), a TaskRefused error is returned.

---

## 5 Parameter descriptions

EN 301 931-2 [4], clause 12 is applicable with the enhancements of the new following parameters specified in the present document.

### 5.1 DetachSignallingPath

This indicator is used in the argument of CPH operations. It tells the CCF whether the signalling path between the Signalling Termination represented by an exported leg and the Signalling Terminations represented by the other legs should be broken or not. When this parameter is absent, the behaviour of the CCF is implementation dependent.

### 5.2 DestinationIndex

This parameter contains a pointer to a call destination (see ITU-T Recommendation E.410 [5]).

### 5.3 ExportSignallingPath

This indicator is used by the MoveLeg procedure. It tells the CCF whether the signalling path between the Signalling Termination represented by the exported leg and the Signalling Terminations represented by the other legs of the target CS should be impacted or not. If the parameter is absent, the behaviour of the CCF is implementation dependent. The detailed impact is outside the scope of the present document and should be specified in the present document.

### 5.4 IncomingSignallingBufferCopy

This indicator is used in the InitiateCallAttempt procedure. When present, the parameters of the setup.ind primitive sent by the CCF should be populated with the information received in the operation argument and in the setup.req primitive from the Signalling Termination associated with the joined controlling leg (if any) of the call segment association. If this indicator is absent, the parameters of the setup.ind primitive sent by the CCF will be populated with the information received in the operation argument and some locally defined information as defined in ITU-T Recommendation Q.1236 [6].



## 5.5 IPRelatedInformation

IPRelatedInformation: This parameter contains a number of sub-parameters that are specific to the interworking with IP-based networks. Currently available sub-parameters are:

- alternativeCalledPartyIds: one or more identities representing a destination in the form of a valid URL. The mapping on call signalling parameters is protocol dependent.

NOTE 1: In SIP environments, such identities are represented as SIP URLs, mapped to the "To:" field. In H.323 environments, such identities are represented as alias addresses, mapped to the destinationAddress field.

- alternativeOriginatingPartyIds: one or more identities representing an originating party in the form of a valid. The mapping on call signalling parameters is protocol dependent.

NOTE 2: In SIP environments, such identities are represented as SIP URLs mapped to the "From:" field. In H.323 environments, such identities are represented as alias addresses, mapped to the sourceAddress field.

- alternativeOriginalCalledPartyIds: one or more identities representing the original destination of a forwarded call, in the form of a valid URL. The mapping on call signalling parameters is protocol dependent.

NOTE 3: In SIP environments, such identities may be represented as SIP URLs, mapped to the "Record-route:" field. In H.323 environments, such identities are represented as alias addresses, mapped to the H.450.3 parameters.

- alternativeRedirectingPartyIds: one or more identities representing a redirecting party, in the form of a valid URL. The mapping on call signalling parameters is protocol dependent.

NOTE 4: In SIP environments, such identities may be represented as SIP URLs mapped to the "Record-route:" field. In H.323 environments, such identities are represented as alias addresses, mapped to the H.450.3 parameters.

## 5.6 MergeSignallingPaths

This indicator is used by the MergeCallSegment procedure. It tells the CCF whether the signalling path between the Signalling Termination represented by the imported legs and the Signalling Terminations represented by the other legs of the target CS should be impacted or not. If the parameter is absent, the behaviour of the CCF is implementation dependent. The detailed impact is outside the scope of the present document and should be specified in the present document.

---

# 6 ASN.1 definitions

## 6.1 Data Types

-- The Definition of SSF - SCF Data Types Follows

```
IN-SSF-SCF-datatypes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs4(40)
modules(1) in-ssf-scf-datatypes(6) version1(0)}
```

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

```
    common-classes,
    common-datatypes,
    ssf-scf-classes,
    scf-srf-classes,
    scf-srf-datatypes,
    tc-Messages
```

```
FROM IN-object-identifiers {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs4(40) modules(1) in-object-identifiers(0) version1(0)}
```

```

COMMON-BOUNDS
FROM IN-common-classes common-classes

    TRIGGER,
    SCF-SSF-BOUNDS
FROM IN-SSF-SCF-Classes ssf-scf-classes

    SCF-SRF-BOUNDS
FROM IN-CS3-scf-srf-classes scf-srf-classes

    Extensions{},
    Integer4
FROM IN-common-datatypes common-datatypes

    InformationToSend {}

FROM IN-CS3-scf-srf-datatypes scf-srf-datatypes

    AddOnChargingInformation,
    ChargingTariffInformation,
    ChargingMessageType
FROM Tariffing-DataTypes {itu-t(0) identified-organization etsi (0) 1296 version2(3)}

    ISDN-AddressString
FROM MAP-CommonDataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) gsm-Network(1)
modules(3) map-CommonDataTypes(18) version6(6)}
;-- The following three definitions are local short-hand notation for convenience.
B1::= COMMON-BOUNDS      -- defined in part 1
B2::= SCF-SSF-BOUNDS   -- defined herein.
B3::= SCF-SRF-BOUNDS   -- defined in EN 301 931-3.

AChBillingChargingCharacteristics {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minAChBillingChargingLength..b2.&maxAChBillingChargingLength))
-- The AChBillingChargingCharacteristics parameter specifies charging related information.
-- Its content is network operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.
-- CAMEL:
-- AChBillingChargingCharacteristics {PARAMETERS-BOUND: bound} ::= OCTET STRING (SIZE
--     (bound.&minAChBillingChargingLength..bound.&maxAChBillingChargingLength))
-- (CONSTRAINED BY {
--     shall be the result of the BER-encoded value of the type --
--     CAMEL-AChBillingChargingCharacteristics {bound}}}
-- The AChBillingChargingCharacteristics parameter specifies the charging related information
-- to be provided by the gsmSSF and the conditions on which this information has to be reported
-- back to the gsmSCF with the ApplyChargingReport operation. The value of the
-- AChBillingChargingCharacteristics of type OCTET STRING carries a value of
-- the ASN.1 data type: CAMEL-AChBillingChargingCharacteristics.
-- The normal encoding rules are used to encode this value.
-- The violation of the UserDefinedConstraint shall be handled as an ASN.1 syntax error.

AChChargingAddress {B2: b2} ::= CHOICE {
    legID          [2] LegID,
    srfCallSegment [50] CallSegmentID {b2},
    bNCF          [51] LegID
}

ActionIndicator ::= ENUMERATED {
    activate      (1),
    deactivate   (2),
    retrieve     (3)
}
-- indicates the action to be performed by the ManageTriggerData operation.

ActionOnProfile ::= ENUMERATED {
    activate      (0),
    deactivate   (1)
}
-- indicates the action to be performed by the SetServiceProfile operation.

```

```

ActionPerformed ::= ENUMERATED {
    activated      (1),
    deactivated    (2),
    alreadyActive  (3),
    alreadyInactive (4),
    isActive      (5),
    isInactive     (6),
    tDPunknown    (7)
}
-- indicates the result of the operation ManageTriggerData

AdditionalCallingPartyNumber {B2: b2} ::= Digits {b2}
-- Indicates the Additional Calling Party Number. Refer to ITU-T Recommendation Q.763
-- Generic Number for encoding.

AlertingPattern ::= OCTET STRING (SIZE(3))
-- Indicates a specific pattern that is used to alert a subscriber
-- (e.g. distinctive ringing, tones, etc.).
-- Only the trailing OCTET is used, the remaining OCTETS should be sent as NULL (zero)
-- The receiving side ignores the leading two OCTETS.
-- Only applies if SSF is the terminating local exchange for the subscriber.
-- Refer to the ITU-T Recommendation Q.931 Signal parameter
-- respective TS 129 002 for encoding.

AlternativeIdentities {B2:b2} ::= SEQUENCE (SIZE (1.. b2.&maxAlternativeIdentities) )
    OF AlternativeIdentity

AlternativeIdentity ::= CHOICE {
    url      [0] IA5String (SIZE(1..512))      -- any RFC1738 compliant URL (e.g.; SIP URL)
}
--Email addresses shall be represented as URLs.

AOCBeforeAnswer ::= SEQUENCE {
    aOCInitial      [0] CAI-GSM0224,
    aOCSubsequent   [1] AOCSubsequent OPTIONAL,
    ...
}
-- support for CAMEL

AOCSubsequent ::= SEQUENCE {
    cAI-GSM0224     [0] CAI-GSM0224 ,
    tariffSwitchInterval [1] INTEGER (1..86400) OPTIONAL,
    ...
}
-- tariffSwitchInterval is measured in 1 second units
-- support for CAMEL

ApplicationTimer ::= INTEGER (0..2047)
-- Used by the SCF to set a timer in the SSF. The timer is in seconds.

AssistingSSPIPRoutingAddress {B2: b2} ::= Digits {b2}
-- Indicates the destination address of the SRF for the assist procedure.
-- Refer to ITU-T Recommendation Q.763 Generic Number for encoding.

BackwardGVNS {B2: b2} ::= OCTET STRING (SIZE(
b2.&minBackwardGVNSLength..b2.&maxBackwardGVNSLength))
-- Indicates the GVNS Backward information. Refer to ITU-T Recommendation Q.735 for encoding.

BackwardServiceInteractionInd ::= SEQUENCE {
    conferenceTreatmentIndicator [1] OCTET STRING (SIZE(1)) OPTIONAL,
    -- acceptConferenceRequest    'xxxx xx01'B
    -- rejectConferenceRequest    'xxxx xx10'B
    -- network default is accept conference request,

    callCompletionTreatmentIndicator [2] OCTET STRING (SIZE(1)) OPTIONAL,
    -- acceptCallCompletionServiceRequest 'xxxx xx01'B,
    -- rejectCallCompletionServiceRequest 'xxxx xx10'B
    -- network default is accept call completion service request

    holdTreatmentIndicator [3] OCTET STRING (SIZE(1)) OPTIONAL,
    -- acceptHoldRequest    'xxxx xx01'B
    -- rejectHoldRequest    'xxxx xx10'B
    -- network default is accept hold request

    ectTreatmentIndicator [4] OCTET STRING (SIZE(1)) OPTIONAL,
    -- acceptEctRequest    'xxxx xx01'B
    -- rejectEctRequest    'xxxx xx10'B

```

```

-- network default is accept ect request
...
}

BasicGapCriteria {B2: b2} ::= CHOICE {
  calledAddressValue      [0] Digits {b2},
  gapOnService            [2] GapOnService,
  gapAllInTraffic         [3] NULL,
  calledAddressAndService [29] SEQUENCE {
    calledAddressValue      [0] Digits {b2},
    serviceKey              [1] ServiceKey,
    ...
  },
  callingAddressAndService [30] SEQUENCE {
    callingAddressValue     [0] Digits {b2},
    serviceKey              [1] ServiceKey,
    locationNumber          [2] LocationNumber {b2}      OPTIONAL,
    ...
  }
}

-- Both calledAddressValue and callingAddressValue can be
-- incomplete numbers, in the sense that a limited amount of digits can be given.
-- For encoding of the digits, refer to ITU-T Recommendation Q.763 Generic Number.

BCSMEvent {B2: b2} ::= SEQUENCE {
  eventTypeBCSM          [0] EventTypeBCSM,
  monitorMode           [1] MonitorMode,
  legID                 [2] LegID      OPTIONAL,
  dpSpecificCriteria    [30] DpSpecificCriteria {b2}  OPTIONAL,
  ...
}

-- Indicates the BCSM Event information for monitoring.

BearerCapability {B2: b2} ::= CHOICE {
  bearerCap             [0] OCTET STRING
    (SIZE(2..b2.&maxBearerCapabilityLength)),
  tmr                   [1] OCTET STRING (SIZE(1))
}

-- Indicates the type of bearer capability connection to the user.
-- For narrowband bearerCapability, either
-- DSS1 (ES 300 403-1) or the ISUP User Service Information (ITU-T Recommendation Q.763)
-- encoding can be used. Refer
-- to the ITU-T Recommendation Q.763 Transmission Medium Requirement parameter for tmr encoding.

BothwayThroughConnectionInd ::= ENUMERATED {
  bothwayPathRequired   (0),
  bothwayPathNotRequired (1)
}

-- The default is bothwayPathRequired.

CAI-GSM0224 ::= SEQUENCE {
  e1 [0] INTEGER (0..8191)      OPTIONAL,
  e2 [1] INTEGER (0..8191)      OPTIONAL,
  e3 [2] INTEGER (0..8191)      OPTIONAL,
  e4 [3] INTEGER (0..8191)      OPTIONAL,
  e5 [4] INTEGER (0..8191)      OPTIONAL,
  e6 [5] INTEGER (0..8191)      OPTIONAL,
  e7 [6] INTEGER (0..8191)      OPTIONAL,
  ...
}

-- Support for CAMEL
-- Indicates Charge Advice Information to the Mobile Station. For information regarding
-- parameter usage, refer to TS 122 024.

CalledDirectoryNumber {B2: b2} ::= OCTET STRING (SIZE
  (b2.&minCalledDirectoryNumberLength..
  b2.&maxCalledDirectoryNumberLength))

-- Indicates the Called Directory Number.
-- Refer to ITU-T Recommendation Q.763 'Called Directory Number' for encoding.

CalledPartyBCDNumber {B2: b2} ::= OCTET STRING (SIZE
  (b2.&minCalledPartyBCDNumberLength.. b2.&maxCalledPartyBCDNumberLength))

-- Indicates the Called Party Number, including service selection information.
-- Refer to TS 124 008 for encoding.

```

```
-- This data type carries only the "type of number", "numbering plan identification" and
-- "number digit" fields defined in TS 124 008;
-- it does not carry the "called party BCD number IEI" or
-- "length of called party BCD number contents".
```

```
CalledPartyNumber {B2: b2} ::= OCTET STRING (SIZE
(b2.&minCalledPartyNumberLength..
b2.&maxCalledPartyNumberLength))
```

```
-- Indicates the Called Party Number. Refer to ITU-T Recommendation Q.763 for encoding.
```

```
CallingGeodeticLocation {B2: b2} ::= OCTET STRING ( SIZE
(b2.&minCallingGeodeticLocationLength..b2.&maxCallingGeodeticLocationLength))
```

```
-- The coding of this parameter is based on the appropriate mapping with the
-- ISUP parameter Calling Geodetic Location.
-- Refer to ITU-T Recommendation Q.763 for encoding.
```

```
CallingPartyBusinessGroupID ::= OCTET STRING
```

```
-- Indicates the business group of the calling party. The value of this octet string is network
-- operator specific.
-- Its content is network operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter
-- (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.
```

```
CallingPartyNumber {B2: b2} ::= OCTET STRING (SIZE (
b2.&minCallingPartyNumberLength..
b2.&maxCallingPartyNumberLength))
```

```
-- Indicates the Calling Party Number. Refer to ITU-T Recommendation Q.763 for encoding.
```

```
CallingPartysCategory ::= OCTET STRING (SIZE(1))
```

```
-- Indicates the type of calling party (e.g. operator, payphone, ordinary subscriber).
-- Refer to ITU-T Recommendation Q.763 for encoding
```

```
CallReference {B2: b2} ::= OCTET STRING (SIZE(1..b2.&maxCallReferenceLength))
```

```
-- The coding of this parameter is network specific.
-- A possible coding is the ITU-T Recommendation Q.763 call reference
-- but other encoding schemes are possible.
```

```
CAMEL-AChBillingChargingCharacteristics {B1: b1} ::= CHOICE {
timeDurationCharging [0] SEQUENCE {
maxCallPeriodDuration [0] INTEGER (1..864000),
releaseIfdurationExceeded [1] BOOLEAN DEFAULT FALSE,
tariffSwitchInterval [2] INTEGER (1..86400) OPTIONAL,
tone [3] BOOLEAN DEFAULT FALSE,
extensions [4] Extensions {b1} OPTIONAL,
...
}
}
```

```
-- tariffSwitchInterval is measured in 1 second units.
-- maxCallPeriodDuration is measured in 100 millisecond units
```

```
CAMEL-CallResult {B1: b1} ::= CHOICE {
timeDurationChargingResult [0] SEQUENCE {
partyToCharge [0] ReceivingSideID,
timeInformation [1] TimeInformation,
callActive [2] BOOLEAN DEFAULT TRUE,
callReleasedAtTcpExpiry [3] NULL OPTIONAL,
extensions [4] Extensions {b1} OPTIONAL,
...
}
}
```

```
CAMEL-FCIBillingChargingCharacteristics {B2: b2} ::= CHOICE{
fCIBCCAMELsequence1 [0] SEQUENCE {
freeFormatData [0] OCTET STRING (SIZE
(b2.&minCamelFCIBillingChargingDataLength..
b2.&maxCamelFCIBillingChargingDataLength)),
partyToCharge [1] SendingSideID
DEFAULT sendingSideID: leg1,
...
}
}
```

```
CAMEL-SCIBillingChargingCharacteristics ::= CHOICE {
  aOCBeforeAnswer          [0] AOCBeforeAnswer,
  aOCAfterAnswer           [1] AOCSubsequent
}
```

```
GlobalCallReference {B2: b2} ::= OCTET STRING (SIZE(1..b2.&maxGlobalCallReferenceLength))
-- The coding of this parameter is defined in annex D.
```

```
CallResult {B1: b1, B2: b2} ::= CHOICE {
  callResult OCTET STRING (SIZE (b2.&minCallResultcs1Length..
                                b2.&maxCallResultcs1Length)),
  callSupervisionResult [51] SEQUENCE {
    callResult [0] OCTET STRING (SIZE (b2.&minCallResultcs1Length..
                                         b2.&maxCallResultcs1Length)) OPTIONAL,
    reportConditionInformation [1] ReportConditionInformation,
    aChChargingAddress [2] AChChargingAddress {b2} DEFAULT legID:receivingSideID:leg1,
    supervisionResult [3] CHOICE {
      cost [1] UsedCost,
      time [2] UsedTime,
      pulses [3] UsedPulses
    },
    minLimitNeeded [4] CHOICE {
      cost [1] NeededCost,
      pulses [2] NeededPulses
    },
    extensions [5] Extensions {b1} OPTIONAL,
    ...
  }
}
```

```
-- This parameter provides the SCF with the charging related information previously requested
-- using the ApplyCharging operation.
-- Examples of charging related information to be provided by the SSF may be: bulk counter values,
-- costs, tariff change and time of change, time stamps, durations, etc.
-- Examples of conditions on which the charging related information are to be reported may be:
-- threshold value reached, timer expiration, tariff change, end of connection configuration, etc.
-- CAMEL:
-- CallResult {PARAMETERS-BOUND: bound} ::= OCTET STRING (SIZE (b2.&minCallResultLength..
-- b2.&maxCallResultLength))
-- (CONSTRAINED BY {
-- CAMEL-CallResult {bound}})
-- The violation of the UserDefinedConstraint shall be handled as an ASN.1 syntax error.
-- This parameter provides the gsmSCF with the charging related information previously requested
-- using the ApplyCharging operation. This shall include the partyToCharge parameter as
-- received in the related ApplyCharging operation to correlate the result to the request
-- The value of the CallResult of type OCTET STRING carries
-- a value of the ASN.1 data type: CAMEL-CallResult.
-- The normal encoding rules are used to encode this value.
```

```
CallSegmentID {B2: b2} ::= INTEGER (1..b2.&numOfCSs)
```

```
CallSupervision {B2: b2} ::= SEQUENCE {
  supervisionMethod [1] SupervisionMethod,
  warningBeforeLimitReached [2] WarningBeforeLimitReached OPTIONAL,
  releaseWhenLimitReached [3] ReleaseWhenLimitReached {b2} OPTIONAL,
  reportConditions [4] ReportConditions OPTIONAL,
  ...
}
```

```
Carrier {B2: b2} ::= OCTET STRING (SIZE (b2.&minCarrierLength..
                                         b2.&maxCarrierLength))
```

```
-- Contains the carrier selection field (first octet) followed by either Carrier ID information
-- (option 1 (e.g. North America (na)) ), or the Transit
-- Network selection information (option 2 (e.g. Europa)), depending on the network
-- in which the switch is located..
```

```
-- In both cases, the Carrier selection is one octet and is encoded as:
```

```

-- 00000000    No indication
-- 00000001    Selected carrier identification code (CIC) pre subscribed and not
--              input by calling party
-- 00000010    Selected carrier identification code (CIC) pre subscribed and input by calling party
-- 00000011    Selected carrier identification code (CIC) pre subscribed, no indication of
--              whether input by calling party (undetermined)
-- 00000100    Selected carrier identification code (CIC) not pre subscribed and input by
--              calling party
-- 00000101
-- to          Spare
-- 11111110
-- 11111111    Reserved

-- Refer to ITU-T Recommendation Q.763 for encoding of Transit Network Selection.
-- Refer to ANSI ISUP T.113 for encoding of na carrier ID information

```

```

Cause {B2: b2} ::= OCTET STRING (SIZE (minCauseLength..
                                b2.&maxCauseLength))

```

```

-- Indicates the cause for interface related information.
-- Refer to the ITU-T Recommendation Q.763 Cause parameter for encoding
-- For the use of cause and location values refer to ITU-T Recommendation Q.850.

```

```

CCSS ::= BOOLEAN

```

```

-- Used by the SSF to indicate CCSS (Call Completion on Service Set-up) if set to "True"
-- to the SCF, i.e.
-- that the current call is due a special procedure (CCBS or CCNR).
-- The value TRUE corresponds with the ITU-T Recommendation Q.763 CCSS parameter
-- "CCSS call indicator" value 1 (CCSS call).
-- The value FALSE corresponds with the ITU-T Recommendation Q.763 CCSS parameter
-- "CCSS call indicator" value 0 (no indication).

```

```

CGEncountered ::= ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    sCPOverload (2)
}

```

```

-- Indicates the type of automatic call gapping encountered, if any.

```

```

ChargedParty ::= ENUMERATED {
    callingPartyNumber (0),
    calledINNumber (1),
    connectedNumber (2),
    translatedNumber (3),
    specialINNumber (4),
    ...
}

```

```

-- The TranslatedNumber is on the ISUP named CalledPartyNumber
-- (e.g. translated number after Connect operation).

```

```

ChargedPartyNumber {B2: b2} ::= OCTET STRING (SIZE(b2.&minChargedPartyNumberLength..
                                                b2.&maxChargedPartyNumberLength) )

```

```

-- Indicates the number that identifies the entity or party to be charged for the call.
-- The coding of this parameter is network specific.
-- A possible coding is the ITU-T Recommendation Q.763 called party number
-- but other encoding schemes are possible.

```

```

ChargeNumber {B2: b2} ::= LocationNumber {b2}

```

```

-- This parameter is only used for CAMEL. It uniquely identifies the chargeable number
-- of a call sent into a North American
-- long distance carrier. It transports the ChargeNumber Parameter Field
-- as defined in ANSI ISUP T1.113. This provides 1 octet for the nature of address indicator field,
-- plus 1 octet for a numbering plan field, plus up to 5 octets for the address signal
-- (up to 10 digits). It is used in the context of CAMEL.
-- The Charge Number in ANSI T1.113 normally contains a 10 digit national number within the North
-- American Numbering Plan (NANP); longer (e.g. international) charge numbers are not supported in
-- T1.113-
-- Information indicating the chargeable number for the call and consisting of the
-- odd/even indicator, nature of address indicator, numbering plan indicator,
-- and address signals.
-- Uses the LocationNumber format for encoding which is based on the ITU-T Recommendation Q.763
Location Number format

```

```

--ChargeUnitTimeIntervalType

```

```

ChargingAddress {B2: b2} ::= CHOICE {

```

```

leg                [2] LegID,
srfCallSegment    [50] CallSegmentID {b2}
}

ChargingControlType ::= SEQUENCE {
  tariffInfoFromSuccExchangeSCI [0] TariffInfoFromSuccExchangeSCI,
  ssfdetermination              [1] SSfdetermination DEFAULT noDetermination,
  ...
}

ChargingEvent {B2: b2} ::= SEQUENCE {
  eventTypeCharging [0] EventTypeCharging {b2},
  monitorMode       [1] MonitorMode,
  legID             [2] LegID OPTIONAL,
  eventTypeTariff   [50] EventTypeTariff OPTIONAL,
  ...
}
-- This parameter indicates the charging event type and corresponding monitor mode and LedID

CNInfo {B2: b2} ::= OCTET STRING (SIZE(1.. b2.&maxCNInfoLength))
-- Refer to ITU-T Recommendation Q.765.1 NNI specific information to be transported in the
-- Application Transport Parameter (APP).

-- The NNI specific information for the VPN application is carried within the APP.
-- Only the CNID (Corporate Telecommunications Network Identifier) indicator is significant
-- for INAP (bits 5 and 6).
-- The CNID indicator is followed by optionally (if included) the CNID length and CNID value.
-- CommunicationChargeCurrency Type

ConnectedNumberTreatmentInd ::= ENUMERATED {
  noINImpact           (0),
  presentationRestricted (1),
  presentCalledINNumber (2),
  presentCalledINNumberRestricted (3)
}
-- The default is presentCalledINNumber.

ControlType ::= ENUMERATED {
  sCPOverloaded (0),
  manuallyInitiated (1),
  destinationOverload (2)
}

CompoundCriteria {B2: b2} ::= SEQUENCE {
  basicGapCriteria [0] BasicGapCriteria {b2},
  scfID            [1] ScfID {b2} OPTIONAL,
  ...
}
-- If scfID parameter is not available the call gapping is not dedicated to a specific SCF.

CorrelationID {B2: b2} ::= Digits {b2}
-- used by SCF for correlation with a previous operation.
-- refer to ITU-T Recommendation Q.763 Generic Digits for encoding from SCF to SSF.
-- refer to ITU-T Recommendation Q.763 Generic Number for encoding from SRF/SSF to SCF.

CounterAndValue ::= SEQUENCE {
  counterID [0] CounterID,
  counterValue [1] Integer4,
  ...
}

CounterID ::= INTEGER (0..99)
-- Indicates the counters to be incremented.
-- The counterIDs are addressed by using the last digits of the dialed number.

CountersValue ::= SEQUENCE SIZE(0..numOfCounters) OF CounterAndValue

CreateOrRemoveIndicator ::= ENUMERATED {
  create (0),
  remove (1)
}
-- Indicates the creation or removal of a TDP-R.

```



```

CSAID {B2: b2} ::= INTEGER (1..b2.&numOfCSAs)
-- Indicates the SSF CSA identifier

-- Currency Type

Currency ::= ENUMERATED {
    noIndication      (0),
    australianDollar  (1),
    austrianSchilling (2),
    belgianFranc      (3),
    britishPound      (4),
    czechKoruna       (5),
    danishKrone       (6),
    dutchGuilder      (7),
    euro              (8),
    finnishMarkka     (9),
    frenchFranc       (10),
    germanMark        (11),
    greekDrachma      (12),
    hungarianForint   (13),
    irishPunt         (14),
    italianLira       (15),
    japaneseYen       (16),
    luxembourgian-Franc (17),
    norwegianKrone    (18),
    polishZloty       (19),
    portugeseEscudo   (20),
    russianRouble     (21),
    slovakKoruna      (22),
    spanishPeseta     (23),
    swedishKrone     (24),
    swissFranc        (25),
    turkishLira       (26),
    uSDollar          (27),
    ...}

-- CurrencyFactor Type

CurrencyFactor ::= INTEGER (0..999999)

-- Value 0 indicates "no charge".

CurrencyScale ::= INTEGER (-7..3)

-- The actual value for currency scale is given by 10x, where x is the value of the CurrencyScale.
-- The coding of CurrencyScale is as follows, all other values are spare:
-- -7 (249): 0,0000001
-- -6 (250): 0,000001
-- -5 (251): 0,00001
-- -4 (252): 0,0001
-- -3 (253): 0,001
-- -2 (254): 0,01
-- -1 (255): 0,1
-- 0      : 1
-- 1      : 10
-- 2      : 100
-- 3      : 1000

CutAndPaste ::= INTEGER (0..22)
-- Indicates the number of leading digits to be deleted (cut) and to paste remaining dialed digits.

DateAndTime ::= OCTET STRING (SIZE(6..7))
-- Indicates, amongst others, the start time and stop time for activate service filtering.
-- Coded as YYMMDDHHMMSS (option 1) or YYYYMMDDHHSS (option 2) with each digit coded BCD

-- OPTION 1 (Size 6):
-- The first octet contains YY and the remaining items are sequenced following
-- For example, 1998 September 30th, 12:15:01 would be encoded as:
-- Bits          HGFE      DCBA
-- leading octet    8        9
--                  9        0
--                  0        3
--                  2        1
--                  5        1
--                  1        0
-- The 2 digit value
-- representing a Year shall be interpreted as follows

```

```
-- If the two-digits value is 00 through 49 inclusive, it shall be interpreted as representing
-- year 2000 through 2049.
-- If the two-digits value is 50 through 99 inclusive, it shall be interpreted as representing
-- year 1950 through 1999.
```

```
-- Option 2 (Size 7):
-- Support for CAMEL.
-- The year digit indicating millenium occupies bits
-- 0-3 of the first octet, and the year digit indicating century occupies bits
-- 4-7 of the first octet.
-- The year digit indicating decade occupies bits 0-3 of the second octet,
-- whilst the digit indicating the year within the decade occupies bits 4-7 of
-- the second octet.
-- The most significant month digit occupies bits 0-3 of the third octet,
-- and the least significant month digit occupies bits 4-7 of the third octet.
-- The most significant day digit occupies bits 0-3 of the fourth octet,
-- and the least significant day digit occupies bits 4-7 of the fourth octet.
-- The most significant hours digit occupies bits 0-3 of the fifth octet,
-- and the least significant digit occupies bits 4-7 of the fifth octet.
-- The most significant minutes digit occupies bits 0-3 of the sixth octet,
-- and the least significant digit occupies bits 4-7 of the sixth octet.
-- The most significant seconds digit occupies bits 0-3 of the seventh octet,
-- and the least significant digit occupies bits 4-7 of the seventh octet.
-- For the encoding of digits in an octet, refer to the timeAndtimezone parameter.
```

```
DefaultFaultHandling {B1: b1, B2: b2, B3: b3} ::= SEQUENCE{
  action      [0]      ENUMERATED {
    resumeCallProcessing (0),
    releaseCall          (1),
    ...
  },
  treatment   [1] GapTreatment {b1, b2,b3} OPTIONAL,
  ...
}
```

```
-- The data type GapTreatment was reused from the existing types to avoid the definition
-- of a new one.
```

```
DestinationIndex { B2:b2 } ::= OCTET STRING
  (SIZE(b2.&minDestinationIndexLength..b2.&maxDestinationIndexLength))
```

```
-- Indicates a pointer to a destination. Its content is network operator specific
```

```
DestinationRoutingAddress {B2: b2} ::=
  SEQUENCE SIZE(1) OF CalledPartyNumber {b2}
```

```
-- Indicates the destination address for routing.
```

```
Digits {B2: b2} ::=
  OCTET STRING (SIZE (b2.&minDigitsLength..b2.&maxDigitsLength))
```

```
-- Indicates the address signalling digits.
-- Refer to the ITU-T Recommendation Q.763 Generic Number and Generic Digits parameter for
-- encoding.
```

```
-- The digits may also include the '#', '*', a, b and c digits.
-- The coding of the subfield's 'NumberQualifier' in Generic Number and 'TypeOfDigits' in
-- Generic Digits are irrelevant to the INAP, the ASN.1 tags are sufficient to
-- identify the parameter.
-- The ISUP format does not allow to exclude these subfields, therefore the value
-- is network operator specific.
-- The following parameters should use Generic Number
-- Additional Calling Party Number, CorrelationID for AssistRequestInstructions,
-- AssistingSSPIPRoutingAddress for EstablishTemporaryConnection,
-- calledAddressValue for all occurrences, callingAddressValue for all occurrences
-- The following parameters should use Generic Digits: all
-- other CorrelationID occurrences,, lineID for ResourceID type, digitResponse for
-- ReceivedInformationArg, iNServiceControlLow / iNServiceControlHigh for
-- MidCallInfoType, iNServiceControlCode for MidCallInfo.
```

```
DisplayInformation {B2: b2} ::= IA5String (SIZE (b2.&minDisplayInformationLength..
  b2.&maxDisplayInformationLength))
```

```
-- Indicates the display information
```

```
DpSpecificCriteria {B2: b2} ::= CHOICE {
  numberOfDigits      [0] NumberOfDigits,
  applicationTimer    [1] ApplicationTimer,
  midCallControlInfo [2] MidCallControlInfo {b2},
  numberOfDigitsTwo   [3] SEQUENCE {
    requestedNumberOfDigits [0] NumberOfDigits,
    minNumberOfDigits       [1] NumberOfDigits OPTIONAL,
  }
}
```

```

    ...
  }
}
-- The SCF may specify the number of digits to be collected by the SSF for the CollectedInfo event
-- When all digits are collected, the SSF reports the event to the SCF
-- The SCF may set a timer in the SSF for the No Answer event. If the user does not answer the call
-- within the allotted time, the SSF reports the event to the SCF
-- The SCF may specify the number of digits to be collected by the SSF for the
-- CollecteInfo event and hereby specify a minimum number of digits to be collected in case
-- the exact number of digits is unknown to the SCF, but a report is desired in case
-- of complete number is determined before the requested number of digits has been collected.

```

```
Duration ::= INTEGER (-2..86400)
```

```
-- Values are seconds
-- special meaning applies for -2, -1 and 0 values as described for the concerned
-- operation procedures.
```

```
Entry ::= CHOICE {
  agreements          [0] OBJECT IDENTIFIER,
  networkSpecific    [1] Integer4
}

```

```
EventSpecificInformationBCSM {B2: b2} ::= CHOICE {
  collectedInfoSpecificInfo [0] SEQUENCE {
    calledPartyNumber [0] CalledPartyNumber {b2},
    ...
  },
  analysedInfoSpecificInfo [1] SEQUENCE {
    calledPartyNumber [0] CalledPartyNumber {b2},
    ...
  },
  routeSelectFailureSpecificInfo [2] SEQUENCE {
    failureCause [0] Cause {b2} OPTIONAL,
    ...
  },
  oCalledPartyBusySpecificInfo [3] SEQUENCE {
    busyCause [0] Cause {b2} OPTIONAL,
    ...
  },
  oNoAnswerSpecificInfo [4] SEQUENCE {
    oNoAnswerCause [0] Cause {b2} OPTIONAL,
    ...
  },
  oAnswerSpecificInfo [5] SEQUENCE {
    backwardGVNS [0] BackwardGVNS {b2} OPTIONAL,
    destinationAddress [50] CalledPartyNumber {b2} OPTIONAL,
    -- CAMEL support
    or-Call [51] NULL OPTIONAL,
    -- CAMEL support
    forwardedCall [52] NULL OPTIONAL,
    ...
  },
  oMidCallSpecificInfo [6] SEQUENCE {
    connectTime [0] Integer4 OPTIONAL,
    oMidCallInfo [1] MidCallInfo {b2} OPTIONAL,
    ...
  },
  oDisconnectSpecificInfo [7] SEQUENCE {
    releaseCause [0] Cause {b2} OPTIONAL,
    connectTime [1] Integer4 OPTIONAL,
    ...
  },
  tBusySpecificInfo [8] SEQUENCE {
    busyCause [0] Cause {b2} OPTIONAL,
    callForwarded [50] NULL OPTIONAL,
    -- CAMEL support
    ...
  },
  tNoAnswerSpecificInfo [9] SEQUENCE {
    tNoAnswerCause [0] Cause {b2} OPTIONAL,
    callForwarded [50] NULL OPTIONAL,
    -- CAMEL support
    ...
  },
}

```

```

tAnswerSpecificInfo          [10] SEQUENCE {
    destinationAddress        [50]   CalledPartyNumber {b2} OPTIONAL,
-- CAMEL support
    or-Call                   [51]   NULL                OPTIONAL,
-- CAMEL support
    forwardedCall             [52]   NULL                OPTIONAL,
-- CAMEL support
    ...
},
tMidCallSpecificInfo         [11] SEQUENCE {
    connectTime               [0]   Integer4 OPTIONAL,
    tMidCallInfo              [1]   MidCallInfo {b2} OPTIONAL,
    ...
},
tDisconnectSpecificInfo     [12] SEQUENCE {
    releaseCause              [0]   Cause {b2} OPTIONAL,
    connectTime               [1]   Integer4 OPTIONAL,
    ...
},
oTermSeizedSpecificInfo     [13] SEQUENCE {
    -- no specific info defined
    ...
},
oSuspendSpecificInfo        [14] SEQUENCE {
    -- no specific info defined
    ...
},
tSuspendSpecificInfo        [15] SEQUENCE {
    -- no specific info defined
    ...
},
origAttemptAuthorizedSpecificInfo [16] SEQUENCE {
    -- no specific info defined
    ...
},
oReAnswerSpecificInfo       [17] SEQUENCE {
    -- no specific info defined
    ...
},
tReAnswerSpecificInfo       [18] SEQUENCE {
    -- no specific info defined
    ...
},
facilitySelectedAndAvailableSpecificInfo [19] SEQUENCE {
    -- no specific info defined
    ...
},
callAcceptedSpecificInfo    [20] SEQUENCE {
    -- no specific info defined
    ...
},
oAbandonSpecificInfo        [21] SEQUENCE {
    abandonCause              [0]   Cause {b2} OPTIONAL,
    ...
},
tAbandonSpecificInfo        [22] SEQUENCE {
    abandonCause              [0]   Cause {b2} OPTIONAL,
    ...
},
authorizeRouteFailureSpecificInfo [23] SEQUENCE {
    authorizeRouteFailureCause [0]   Cause {b2} OPTIONAL,
    ...
},
terminationAttemptAuthorizedSpecificInfo [24] SEQUENCE {
    -- no specific info defined
    ...
},
orinationAttemptDeniedSpecificInfo [25] SEQUENCE {
    originationDeniedCause    [0]   Cause {b2} OPTIONAL,
    ...
},

```

```

    terminationAttemptDeniedSpecificInfo      [26] SEQUENCE {
        terminationDeniedCause                [0] Cause {b2} OPTIONAL,
        ...
    }
}

-- Indicates the call related information specific to the event.
-- The unit for the connectTime is 100 milliseconds

EventSpecificInformationCharging {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minEventSpecificInformationChargingLength..
    b2.&maxEventSpecificInformationChargingLength))
-- defined by network operator.
-- Its content is network signalling/operator specific.
-- Indicates the charging related information specific to the event.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

EventTypeBCSM ::= ENUMERATED {
    origAttemptAuthorized      (1),
    collectedInfo              (2),
    analysedInformation         (3),
    routeSelectFailure         (4),
    oCalledPartyBusy          (5),
    oNoAnswer                  (6),
    oAnswer                    (7),
    oMidCall                   (8),
    oDisconnect                (9),
    oAbandon                   (10),
    termAttemptAuthorized      (12),
    tBusy                      (13),
    tNoAnswer                  (14),
    tAnswer                    (15),
    tMidCall                   (16),
    tDisconnect                (17),
    tAbandon                   (18),
    oTermSeized                (19),
    oSuspend                   (20),
    tSuspend                   (21),
    origAttempt                (22),
    termAttempt                (23),
    oReAnswer                  (24),
    tReAnswer                  (25),
    facilitySelectedAndAvailable(26),
    callAccepted               (27),
    authorizeRouteFailure       (28),
    originationAttemptDenied    (29),
    terminationAttemptDenied    (30)
}

-- Indicates the name of the BCSM detection point event.
-- The value range 100 - 127 is reserved.

EventTypeCharging {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minEventTypeChargingLength..
    b2.&maxEventTypeChargingLength))
-- This parameter indicates the charging event type..
-- Its content is network signalling / operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

EventTypeTariff ::= ENUMERATED {
    chargingTariffInformation    (0),
    addOnchargingInformation     (1),
    chargingAcknowledgementInformation (2),
    chargingAcknowledgeTimerExpired (3),
    startCharging                (4),
    stopCharging                 (5)
}

```

```

FacilityGroup ::= CHOICE {
    trunkGroupID      [0] INTEGER,
    privateFacilityID [1] INTEGER,
    huntGroup         [2] OCTET STRING,
    routeIndex        [3] OCTET STRING
}
-- Indicates the particular group of facilities to route the call.
-- huntGroup and routeIndex are encoded as
-- network operator specific.
-- The internal structure of huntGroup and routeIndex parameters can be defined using ASN.1
-- and the related Basic Encoding Rules (BER). In such a case the value of this parameter
-- (after the first tag and length information) is the BER encoding of the defined ASN.1
-- internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

FCIBillingChargingCharacteristics {B1: b1, B2: b2} ::= CHOICE {
    fCIBCCs1      OCTET STRING (SIZE (b2.&minFCIBCCcs1Length..
                                   b2.&maxFCIBCCcs1Length)),
-- Its content is network operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and
-- length information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

    fCIBCCsequence      [51] SEQUENCE {

        fCIBCC          [0] OCTET STRING (SIZE
(b2.&minFCIBCCcs2Length..b2.&maxFCIBCCcs2Length)) OPTIONAL,
-- Its content is network operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag
-- and length information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.
        chargeMessageFromSCF      [1] CHOICE {
            sCFChargingTariff      [0] ChargingTariffInformation,
            sCFAddOnCharge          [1] AddOnChargingInformation,
            sSFDeterminedTariff     [2] NULL
        }
        OPTIONAL,

        tariffFromSuccExchange     [2] TariffFromSuccExchange DEFAULT notTakenIntoAccount,
        freeFormatData             [3] OCTET STRING (SIZE
(b2.&minFormatDataLength..b2.&maxFormatDataLength)) OPTIONAL,

        chargingAddress            [4] ChargingAddress {b2}   DEFAULT leg:sendingSideID:leg2,
        partyToChargeIdentifier     [5] PartyToCharge {b2}   OPTIONAL,
        inRecordIndicators         [6] INRecordIndicators OPTIONAL,
-- The parameter should always be included.
        extensions                 [7] Extensions {b1}   OPTIONAL,
        ...
    }
}
-- This FCIBillingChargingCharacteristics parameter indicates the billing and/or
-- charging characteristics.
-- Its content is partly network operator specific.
-- CAMEL:
-- FCIBillingChargingCharacteristics {PARAMETERS-BOUND: bound} ::= OCTET STRING (SIZE
-- (bound.&minFCIBillingChargingLength..bound.&maxFCIBillingChargingLength))
-- (CONSTRAINED BY {
-- shall be the result of the BER-encoded value of type --
-- CAMEL-FCIBillingChargingCharacteristics {bound}})
-- This parameter indicates the billing and/or charging characteristics.
-- The violation of the UserDefinedConstraint shall be handled as an ASN.1 syntax error.
-- The value of the FCIBillingChargingCharacteristics of type OCTET STRING carries
-- a value of the ASN.1 data type: CAMEL-FCIBillingChargingCharacteristics.
-- The normal encoding rules are used to encode this value.

```

```

FilteredCallTreatment {B1: b1, B2: b2, B3: b3} ::= SEQUENCE {
    sFBillingChargingCharacteristics    [0] SFBillingChargingCharacteristics {b2} OPTIONAL,
    informationToSend                    [1] InformationToSend {b1, b2, b3} OPTIONAL,
    maximumNumberOfCounters              [2] MaximumNumberOfCounters OPTIONAL,
    releaseCause                         [3] Cause {b2} OPTIONAL,
    sFTariffMessage                      [50] CHOICE {
        crgt                             [0] ChargingTariffInformation} OPTIONAL,
    ...
}
-- If releaseCause is not present, the default value is the same as the ISUP cause value decimal 31.
-- If informationToSend is present, the call will be released after the end of the announcement
-- with the indicated or default releaseCause.
-- If maximumNumberOfCounters is not present, ServiceFilteringResponse will be sent with
-- CountersValue ::= SEQUENCE SIZE (0) OF CountersAndValue.

FilteringCharacteristics ::= CHOICE {
    interval                            [0] INTEGER (-1..32000),
    numberOfCalls                       [1] Integer4
}
-- The interval is specified in seconds.

FilteringCriteria {B2: b2} ::= CHOICE {
    serviceKey                          [2] ServiceKey,
    addressAndService                   [30] SEQUENCE {
        calledAddressValue              [0] Digits {b2},
        serviceKey                      [1] ServiceKey,
        callingAddressValue             [2] Digits {b2} OPTIONAL,
        locationNumber                  [3] LocationNumber {b2} OPTIONAL,
        ...
    }
}
-- In case calledAddressValue is specified, the numbers to be filtered are from calledAddressValue
-- up to and including calledAddressValue + maximumNumberOfCounters-1.
-- The last two digits of calledAddressvalue cannot exceed 100-maximumNumberOfCounters.
-- For encoding of the digits, refer to ITU-T Recommendation Q.763 for Generic Number.

FilteringTimeOut ::= CHOICE {
    duration                            [0] Duration,
    stopTime                            [1] DateAndTime
}
-- Indicates the maximum duration of the filtering.
-- When the timer expires, a ServiceFilteringResponse is sent to the SCF.

ForwardCallIndicators ::= OCTET STRING (SIZE(2))
-- Indicates the Forward Call Indicators. Refer to ITU-T Recommendation Q.763 for encoding

ForwardGVNS {B2: b2} ::= OCTET STRING (SIZE(
    b2.&minForwardGVNSLength..
    b2.&maxForwardGVNSLength))
-- Indicates the GVNS Forward information. Refer to ITU-T Recommendation Q.735 for encoding.

ForwardServiceInteractionInd ::= SEQUENCE {
    conferenceTreatmentIndicator        [1] OCTET STRING (SIZE(1)) OPTIONAL,
    -- acceptConferenceRequest          'xxxx xx01'B
    -- rejectConferenceRequest         'xxxx xx10'B
    -- network default is accept conference request.

    callDiversionTreatmentIndicator     [2] OCTET STRING (SIZE(1)) OPTIONAL,
    -- callDiversionAllowed            'xxxx xx01'B
    -- callDiversionNotAllowed         'xxxx xx10'B
    -- network default is Call Diversion allowed.

    callOfferingTreatmentIndicator      [3] OCTET STRING (SIZE(1)) OPTIONAL,
    -- callOfferingNotAllowed          'xxxx xx01'B,
    -- callOfferingAllowed             'xxxx xx10'B
    -- callOfferingNoINImpact         'xxxx x100'B
    -- indicates if call offering is "allowed", "not allowed" or "no impact by IN".
    -- network default is Call Offering not allowed.

    callingPartyRestrictionIndicator    [4] OCTET STRING (SIZE(1)) OPTIONAL,
    -- noINImpact                     'xxxx xx01'B,
    -- presentationRestricted          'xxxx xx10'B
    -- network default is noINImpact

```

```

callWaitingTreatmentIndicator      [5] OCTET STRING (SIZE(1)) OPTIONAL,
-- callWaitingAllowed              'xxxx xx01'B,
-- callWaitingNotAllowed          'xxxx xx10'B
-- network default is Call Waiting allowed

holdTreatmentIndicator             [6] OCTET STRING (SIZE(1)) OPTIONAL,
-- acceptHoldRequest              'xxxx xx01'B
-- rejectHoldRequest              'xxxx xx10'B
-- network default is accept hold request

ectTreatmentIndicator             [7] OCTET STRING (SIZE(1)) OPTIONAL,
-- acceptEctRequest               'xxxx xx01'B
-- rejectEctRequest               'xxxx xx10'B
-- network default is accept ect request
...
}

--The forwardServiceInteractionInd parameter is applicable to IDP, CON, CWA and ICA operations.

GapCriteria {B2: b2} ::= CHOICE {
    basicGapCriteria               BasicGapCriteria {b2},
    compoundGapCriteria            CompoundCriteria {b2}
}

GapOnService ::= SEQUENCE {
    serviceKey [0] ServiceKey,
    ...
}

GapIndicators ::= SEQUENCE {
    duration [0] Duration,
    gapInterval [1] Interval,
    ...
}

-- Indicates the gapping characteristics.

GapTreatment {B1: b1, B2: b2, B3: b3} ::= CHOICE {
    informationToSend [0] InformationToSend {b1, b2, b3},
    releaseCause [1] Cause {b2},
    both [2] SEQUENCE {
        informationToSend [0] InformationToSend {b1, b2, b3},
        releaseCause [1] Cause {b2},
        ...
    }
}

-- The default value for Cause is the same as in ISUP.

GenericName {B2: b2} ::= OCTET STRING (SIZE(
    b2.&minGenericNameLength..
    b2.&maxGenericNameLength))
-- Refer to ITU-T Recommendation Q.931 Display Information parameter for encoding.

GenericNumber {B2: b2} ::= OCTET STRING (SIZE(
    b2.&minGenericNumberLength..
    b2.&maxGenericNumberLength))
-- Refer to ITU-T Recommendation Q.763 Generic Number for encoding.

GenericNumbers {B2: b2} ::=
    SET SIZE(1..b2.&numOfGenericNumbers) OF GenericNumber {b2}

HighLayerCompatibility ::= OCTET STRING (SIZE (highLayerCompatibilityLength))
-- Indicates the teleservice. For encoding, DSS1 (ITU-T Recommendation Q.931) is used.

INRecordIndicators ::= SEQUENCE {
    inRecordAction [0] INRecordAction DEFAULT generateCallRecord,
    hotBillingRequired [1] BOOLEAN DEFAULT FALSE,
    ...
}

INRecordAction ::= ENUMERATED {
    generateCallRecord (1),
    appendDataToCallRecord (2),
    overwriteDataForCallRecord (3),
    noDataForCallRecord (4),
    ...
}

```



```

initialCallSegment INTEGER ::= 1

InitialDPArgExtension ::= SEQUENCE {
    gsmcAddress [1] ISDN-AddressString OPTIONAL,
    ...
}

INprofile {B1: b1, B2:b2} ::= SEQUENCE {
    actionOnProfile [0] ActionOnProfile,
    tDPIdentifier [1] TDPIIdentifier {b2},
    dPName [2] EventTypeBCSM OPTIONAL,
    extensions [3] Extensions {b1} OPTIONAL,
    ...
}

INServiceCompatibilityIndication {B2: b2} ::=
    SEQUENCE SIZE (1..b2.&numOfInServiceCompatibilityIndLength) OF Entry

INServiceCompatibilityResponse ::= Entry

Interval ::= INTEGER (-1..60000)
-- Units are in milliseconds. A -1 value denotes infinite.

IPAvailable {B2: b2} ::= OCTET STRING (SIZE (
    b2.&minIPAvailableLength..
    b2.&maxIPAvailableLength))
-- defined by network operator.
-- Indicates that the resource is available.
-- Its content is network operator specific

IPRelatedInformation {B2:b2} ::= SEQUENCE
{
    alternativeCalledPartyIds [0] AlternativeIdentities {b2} OPTIONAL,
    alternativeOriginatingPartyIds [1] AlternativeIdentities {b2} OPTIONAL,
    alternativeOriginalCalledPartyIds [2] AlternativeIdentities {b2} OPTIONAL,
    alternativeRedirectingPartyIds [3] AlternativeIdentities {b2} OPTIONAL,
    ...
}
-- contains information that are specific to interworking with IP-based networks

IPRoutingAddress {B2: b2} ::= CalledPartyNumber {b2}
-- Indicates the routing address for the IP.

IPSSPCapabilities {B2: b2} ::= OCTET STRING (SIZE (
    b2.&minIPSSPCapabilitiesLength..
    b2.&maxIPSSPCapabilitiesLength))
-- defined by network operator.
-- Indicates the SRF resources available at the SSP.
-- Its content is network operator specific

ISDNAccessRelatedInformation {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minISDNAccessRelatedInfoLength..
    b2.&maxISDNAccessRelatedInfoLength))
-- Indicates the destination user network interface related information. Refer to the ITU-T
Recommendation Q.763 Access
-- Transport parameter for encoding.

LegID ::= CHOICE {
    sendingSideID [0] LegType,
    receivingSideID [1] LegType
}
-- sendingSideID is used where legID is sent from the SCF to the SSF and
-- receivingSideID is used where SCF receives legID from the SSF.

LegType ::= OCTET STRING (SIZE(1))

leg1 LegType ::= '01'H

leg2 LegType ::= '02'H

LocationNumber {B2: b2} ::= OCTET STRING (SIZE (
    b2.&minLocationNumberLength..
    b2.&maxLocationNumberLength))
-- Indicates the Location Number for the calling party. Refer to ITU-T Recommendation Q.763
-- for encoding.

MaximumNumberOfCounters ::= INTEGER (1..numOfCounters)

```

```

MidCallControlInfo {B2: b2} ::= SEQUENCE SIZE (
    b2.&minMidCallControlInfoNum..
    b2.&maxMidCallControlInfoNum)
    OF SEQUENCE {
        midCallInfoType          [0] MidCallInfoType {b2},
        midCallReportType       [1] ENUMERATED {
            inMonitoringState    (0),
            inAnyState           (1)
        } DEFAULT inMonitoringState,
        ...
    }

MidCallInfo {B2: b2} ::= SEQUENCE {
    iServiceControlCode [0] Digits {b2},
    ...
}

MidCallInfoType {B2: b2} ::= SEQUENCE {
    iServiceControlCodeLow [0] Digits {b2},
    iServiceControlCodeHigh [1] Digits {b2} OPTIONAL,
    ...
}

MiscCallInfo ::= SEQUENCE {
    messageType [0] ENUMERATED {
        request      (0),
        notification (1)
    },
    ...
}

-- Indicates detection point related information.

MonitorMode ::= ENUMERATED {
    interrupted      (0),
    notifyAndContinue (1),
    transparent      (2)
}

-- Indicates the event is relayed and/or processed by the SSP.

NAOliInfo ::= OCTET STRING (SIZE (1))
-- NA Oli information is only used by Camel for NA (North America).
-- It takes the same value as defined in ANSI ISUP T1.113
-- e.g. '3D'H - Decimal value 61 - Cellular Service (Type 1)
--      '3E'H - Decimal value 62 - Cellular Service (Type 2)
--      '3F'H - Decimal value 63 - Cellular Service (roaming)

NeededCost ::= SEQUENCE {
    usedCurrencyFactor [1] CurrencyFactor,
    usedScale          [2] CurrencyScale OPTIONAL,
    reportedCurrency  [3] Currency DEFAULT noIndication,
    ...
}

NeededPulses ::= SEQUENCE {
    usedPulseUnits [1] OCTET STRING (SIZE (2)),
    ...
}

noCharge INTEGER ::= 0
noScale INTEGER ::= 0

normalRelease OCTET STRING ::= '839F'H

NumberOfDigits ::= INTEGER (1..255)
-- Indicates the number of digits to be collected

OCSIApplicable ::= NULL
-- Indicates that the Originating CAMEL Subscription Information shall be
-- applied on the outgoing call leg created with a Connect operation. For the use of this
-- parameter see TS 123 078.

OriginalCalledPartyID {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minOriginalCalledPartyIDLength..
    b2.&maxOriginalCalledPartyIDLength))
-- Indicates the original called number.
-- Refer to the ITU-T Recommendation Q.763 Original Called Number for encoding.

```

```

PartyToCharge {B2: b2} ::= SEQUENCE {
    chargedParty          [0] ChargedParty DEFAULT callingPartyNumber,
    specialINNumber      [1] ChargedPartyNumber {b2} OPTIONAL,
    ...
}

ProfileIdentifier {B2: b2} ::= CHOICE {
    access      [0] CalledPartyNumber {b2},
    group      [1] FacilityGroup
}
-- Please note that 'CalledPartyNumber' is used to address a subscriber access line.
-- The data type was reused from the existing types to avoid the definition of a new one.
-- PulseUnits Type

Reason {B2: b2} ::= OCTET STRING (SIZE(
    b2.&minReasonLength..
    b2.&maxReasonLength))
-- Its content is network operator specific
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter
-- (after the first tag and length information) is the BER encoding of the defined ASN.1
-- internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

ReceivingSideID ::= CHOICE {receivingSideID [1] LegType}
-- used to identify LegID in operations sent from gsmSSF to gsmSCF

RedirectingPartyID {B2: b2} ::= OCTET STRING (SIZE (
    b2.&minRedirectingPartyIDLength..
    b2.&maxRedirectingPartyIDLength))
-- Indicates redirecting number. Refer to the ITU-T Recommendation Q.763 Redirecting number for
encoding.

RedirectionInformation ::= OCTET STRING (SIZE(2))
-- Indicates redirection information. Refer to the ITU-T Recommendation Q.763
-- Redirection Information for encoding.

RegistrarIdentifier ::= OCTET STRING
-- Its content is network operator specific

RelayChargesFromDestination ::= ENUMERATED {
    noIndication      (0),
    relay              (1),
    noRelay            (2)
}

ReleaseWhenLimitReached {B2: b2} ::= SEQUENCE {
    releaseCause      [1] Cause {b2} DEFAULT normalRelease,
    ...
}

ReportCondition ::= ENUMERATED {
    statusReport      (0),
    timerExpired      (1),
    canceled           (2)
}
-- ReportCondition specifies the cause of sending "StatusReport" operation to the SCF.

ReportConditions ::= SEQUENCE {
    periodicReportInterval [1] INTEGER (1..32767),
    -- Values are in units of seconds.
    ...
}

ReportConditionInformation ::= ENUMERATED {
    intermediateReport (0),
    finalReportCallReleased (1),
    finalReportCallActive (2)
}

RequestedInformationList {B2: b2} ::=
SEQUENCE SIZE (1..numOfInfoItems) OF RequestedInformation {b2}

RequestedInformationTypeList ::= SEQUENCE SIZE (1..numOfInfoItems) OF RequestedInformationType

RequestedInformation {B2: b2} ::= SEQUENCE {
    requestedInformationType [0] RequestedInformationType,

```

```

    requestedInformationValue [1] RequestedInformationValue {b2},
    ...
}

RequestedInformationType ::= ENUMERATED {
    callAttemptElapsedTime (0),
    callStopTime (1),
    callConnectedElapsedTime (2),
    calledAddress (3),
    releaseCause (30)
}

RequestedInformationValue {B2: b2} ::= CHOICE {
    callAttemptElapsedTimeValue [0] INTEGER (0..255),
    callStopTimeValue [1] DateAndTime,
    callConnectedElapsedTimeValue [2] Integer4,
    calledAddressValue [3] Digits {b2},
    releaseCauseValue [30] Cause {b2}
}

-- The callAttemptElapsedTimeValue is specified in seconds.
-- The unit for the callConnectedElapsedTimeValue is 100 milliseconds

RequestedUTSI {B2: b2} ::= SEQUENCE {
    uSIServiceIndicator [0] USIServiceIndicator {b2},
    uSIMonitorMode [1] USIMonitorMode,
    ...
}

RequestedUTSIList {B2: b2} ::= SEQUENCE SIZE
    (b2.&minRequestedUTSINum..
     b2.&maxRequestedUTSINum) OF RequestedUTSI {b2}

ResourceID {B2: b2} ::= CHOICE {
    lineID [0] Digits {b2},
    facilityGroupID [1] FacilityGroup,
    facilityGroupMemberID [2] INTEGER,
    trunkGroupID [3] INTEGER
}

-- Indicates a logical identifier for the physical termination resource.

ResourceStatus ::= ENUMERATED {
    busy (0),
    idle (1)
}

ResponseCondition ::= ENUMERATED {
    intermediateResponse (0),
    lastResponse (1)
}

-- ResponseCondition is used to identify the reason why ServiceFilteringResponse operation is sent.

RouteList {B2: b2} ::=
    SEQUENCE SIZE(1..3) OF OCTET STRING (SIZE
        (b2.&minRouteListLength..
         b2.&maxRouteListLength))

-- Indicates a list of trunk groups or a route index..
-- Its content is network operator specific

RouteingNumber {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minRouteingNumberLength..
     b2.&maxRouteingNumberLength))

-- Indicates the Routeing Number.
-- Refer to ITU-T Recommendation Q.763 parameter Network Routeing Number for encoding.

ScfID {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minScfIDLength..
     b2.&maxScfIDLength))

-- defined by network operator.
-- Indicates the SCF identity.
-- Refer to ITU-T Recommendation Q.713 "calling party address" parameter for encoding.
-- Other encoding schemes are also possible as a network specific option.

SCIBillingChargingCharacteristics {B2: b2} ::= OCTET STRING (SIZE
    (b2.&minSCIBillingChargingLength..
     b2.&maxSCIBillingChargingLength))

```

```
-- This parameter indicates the billing and/or charging characteristics.
-- Its content is network signalling / operator specific
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.
-- CAMEL:
-- SCIBillingChargingCharacteristics {PARAMETERS-BOUND: bound} ::= OCTET STRING (SIZE (
-- b2.&minSCIBillingChargingLength..b2.&maxSCIBillingChargingLength))
-- (CONSTRAINED BY {
-- shall be the result of the BER-encoded value of type --
-- CAMEL-SCIBillingChargingCharacteristics})
-- Indicates AOC information to be sent to a Mobile Station
-- The violation of the UserDefinedConstraint shall be handled as an ASN.1 syntax error.
-- The value of the SCIBillingChargingCharacteristics of type OCTET STRING carries
-- a value of the ASN.1 data type: CAMEL-SCIBillingChargingCharacteristics.
-- The normal encoding rules are used to encode this value.
```

```
SDSSinformation {B2: b2} ::= OCTET STRING (SIZE
(b2.&minSDSSinformationLength..
b2.&maxSDSSinformationLength))
```

```
-- Its content is network signalling/operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
```

```
SendingSideID ::= CHOICE {sendingSideID [0] LegType}
```

```
-- used to identify LegID in operations sent from gsmSCF to gsmSSF
```

```
ServiceInteractionIndicators {B2: b2} ::= OCTET STRING (SIZE (
b2.&minServiceInteractionIndicatorsLength..
b2.&maxServiceInteractionIndicatorsLength))
```

```
-- Indicators which are exchanged between SSP and SCP to resolve interactions
-- between IN based services and network based services, respectively
-- between different IN based services.
```

```
-- Its content is network signalling/operator specific
```

```
-- Note this parameter is kept ifor backward compatibility to IN CS-1,
-- for the present document see parameter ServiceInteractionIndicatorsTwo
-- Its content is network signalling/operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.
```

```
ServiceInteractionIndicatorsTwo ::= SEQUENCE {
```

```
-- ServiceInteractionIndicatorsTwo contains Indicators which are exchanged between SSP and SCP
-- to resolve interactions between IN based services and network based services,
-- respectively between different IN based services.
```

```
    forwardServiceInteractionInd [0] ForwardServiceInteractionInd OPTIONAL,
-- applicable to operations IDP, CON, ICA, CWA.
```

```
    backwardServiceInteractionInd [1] BackwardServiceInteractionInd OPTIONAL,
--applicable to operations IDP, CON, CTR, ETC, CWA.
```

```
    bothwayThroughConnectionInd [2] BothwayThroughConnectionInd OPTIONAL,
-- applicable to operations CTR, ETC.
```

```
    suspendTimer [3] SuspendTimer OPTIONAL,
-- applicable to operations CON, ICA CWA.
```

```
    connectedNumberTreatmentInd [4] ConnectedNumberTreatmentInd OPTIONAL,
--applicable to operations CON, CTR, ETC, CWA.
```

```
    suppressCallDiversionNotification [5] BOOLEAN OPTIONAL,
-- applicable to CON, ICA, CWA
```

```
    suppressCallTransferNotification [6] BOOLEAN OPTIONAL,
-- applicable to CON, ICA, CWA
```

```
    allowCdINNoPresentationInd [7] BOOLEAN OPTIONAL,
```

```
-- applicable to CON, ICA CWA
```

```
-- indicates whether the Address Presentation restricted indicator of the ISUP
```

```

-- "called IN number" shall be set to presentation allowed (TRUE)
-- or presentation restricted (FALSE). Refer to ITU-T Recommendation Q.1601.

    userDialogueDurationInd                [8] BOOLEAN DEFAULT TRUE,
-- applicable to operations CTR, ETC.
-- applicable when interaction with the user is required during call set-up
-- The interaction TRUE means the user interaction may last longer than 90 seconds.
-- Otherwise the indicator should be set to FALSE. Used for delaying ISUP T9 timer.

    overrideLineRestrictions              [9] BOOLEAN DEFAULT FALSE,
-- only applicable to operations (e.g. Connect) which lead to a transition to a PIC before
-- the AuthorizeCallSetup PIC.
-- When set to TRUE, this parameter indicates that some facility restrictions
-- should not be checked when the authority to place a call is verified in the
-- Authorize_Call_Setup PIC.
-- Which restrictions are actually overridden is network specific.

    suppressVPNAPP                        [10] BOOLEAN DEFAULT FALSE,
-- applicable to CWA, CON, ICA.
-- indicates whether to allow or stop (suppress) the forward transmission of the
-- VPN PSS1 capability.
-- When set to TRUE, the exchange, on receipt of this parameter, will not transmit for this call
-- any ISUP Application transport parameter with Application Context Identifier set to
-- « PSS1 ASE (VPN) »
-- This indicator is populated by the SCF, where the SCF and SSF in conjunction have provided the
-- outgoing gateway PINX functionality as required by PSS1.

    calledINNumberOverriding             [11] BOOLEAN OPTIONAL,
-- applicable to CON and CWA
-- indicates whether the generation/override of the ISUP
-- "called IN number" is allowed (TRUE) or not allowed (FALSE)
-- If set to FALSE, the ISUP shall not generate a "called IN number" or override
-- an already existing "called IN number".
-- if absent, the default will be "generation/overriding allowed" (TRUE).

-- tag 12 IS RESERVED

    nonCUGCall                            [13] BOOLEAN DEFAULT FALSE,
-- applicable to CON and CWA
-- indicates whether no parameters for CUG shall be used for the call (i.e. the call shall be
-- a non-CUG call)(TRUE).
-- If set to TRUE, then neither CUG Interlock Code nor Outgoing
-- Access Indicator shall be present; if any of these parameters are present
-- then an error is returned.
-- If set to FALSE or not present, it indicates one of two things:
-- a) continue with modified CUG information (when one or more of either CUG Interlock Code
-- and Outgoing Access Indicator are present), or
-- b) continue with original CUG information (when neither CUG Interlock Code nor Outgoing
-- Access Indicator are present).
...
}

ServiceKey ::= Integer4
-- Information that allows the SCF to choose the appropriate service logic.

SFBillingChargingCharacteristics {B2: b2} ::= OCTET STRING (SIZE
(b2.&minSFBillingChargingLength..
b2.&maxSFBillingChargingLength))
-- This parameter indicates the billing and/or charging characteristics for filtered calls.
-- Its content is network signalling/operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced

SSFdetermination ::= ENUMERATED{
    noDetermination                (0),
    destinationRoutingAddress      (1),
    calledInNumber                  (2)
}
-- The calledInNumber is the calledPartyNumber in the InitialDP operation.

-- SubTariffControl Type

```

```

SupervisionMethod ::= CHOICE {
  maximumTariffCurrency [1] SEQUENCE {
    currencyFactor [0] INTEGER (1..999999),
    -- Values are in currency units.
    currencyScale [1] CurrencyScale DEFAULT noScale,
    -- Imported from ES 201 296.
    currency [2] Currency DEFAULT noIndication,
    -- Imported from ES 201 296.
    ...
  },
  maximumTimeDuration [2] SEQUENCE {
    maximumDurationAllowed [0] INTEGER (1..864000),
    -- Values are in units of 100 msec.
    switchOverTime [1] INTEGER (1..86400) OPTIONAL,
    -- Values are in units of seconds.
    ...
  },
  maximumPulseUnits [3] OCTET STRING (SIZE (2))
}

SupportedTriggers TRIGGER ::= {...}

SuspendTimer ::= INTEGER (-1..120)
-- value in seconds
-- The default is as specified in EN 301 070-1
-- The value -1 indicates the network specific suspend timer (T6) is to be used.

TariffFromSuccExchange ::= ENUMERATED {
  notTakenIntoAccount (0),
  takeIntoAccountNoAOC (1),
  takeIntoAccountTranslateIntoAOC (2)
}

TariffInfoFromSuccExchangeSCI ::= SEQUENCE {
  relayChargesFromDestination [0] RelayChargesFromDestination DEFAULT noRelay,
  relayAOCfromDestination [1] BOOLEAN DEFAULT FALSE,
  ...
}

TDPIdentifier {B2: b2} ::= CHOICE {
  oneTrigger INTEGER,
  triggers [1] Triggers {b2}
}

TerminalType ::= ENUMERATED {
  unknown (0),
  dialPulse (1),
  dtmf (2),
  isdn (3),
  isdnNoDtmf (4),
  spare (16)
}

TimeAndTimezone {B2: b2} ::= OCTET STRING (SIZE(b2.&minTimeAndTimezoneLength..
                                          b2.&maxTimeAndTimezoneLength))
-- Support for CAMEL
-- Indicates the time and timezone, relative to GMT. This parameter BCD encoded.
-- The year digit indicating millenium occupies bits 0-3 of the first octet, and the year
-- digit indicating century occupies bits 4-7 of the first octet.
-- The year digit indicating decade occupies bits 0-3 of the second octet, whilst the digit
-- indicating the year within the decade occupies bits 4-7 of the second octet.
-- The most significant month digit occupies bits 0-3 of the third octet, and the least
-- significant month digit occupies bits 4-7 of the third octet.
-- The most significant day digit occupies bits 0-3 of the fourth octet, and the least
-- significant day digit occupies bits 4-7 of the fourth octet.
-- The most significant hours digit occupies bits 0-3 of the fifth octet, and the least
-- significant hours digit occupies bits 4-7 of the fifth octet.
-- The most significant minutes digit occupies bits 0-3 of the sixth octet, and the least
-- significant minutes digit occupies bits 4-7 of the sixth octet.
-- The most significant seconds digit occupies bits 0-3 of the seventh octet, and the least
-- significant seconds digit occupies bits 4-7 of the seventh octet.
--
-- The timezone information occupies the eighth octet. For the encoding of Timezone refer to
-- Reference [29], TS 123 040.
--

```

```

-- The BCD digits are packed and encoded as follows:
--
-- Bit 7 6 5 4 | 3 2 1 0
--      2nd digit | 1st digit      Octet 1
--      3rd digit | 4th digit      Octet 2
--      ..        | ..            ..
--      nth digit | n-lth digit    Octet m
--
--      0000      digit 0
--      0001      digit 1
--      0010      digit 2
--      0011      digit 3
--      0100      digit 4
--      0101      digit 5
--      0110      digit 6
--      0111      digit 7
--      1000      digit 8
--      1001      digit 9
--      1010      spare
--      1011      spare
--      1100      spare
--      1101      spare
--      1110      spare
--      1101      spare
--
-- where the leftmost bit of the digit is either bit 7 or bit 3 of the octet.

TimeInformation ::= CHOICE {
    timeIfNoTariffSwitch [0] TimeIfNoTariffSwitch,
    timeIfTariffSwitch   [1] TimeIfTariffSwitch
}
-- Indicates call duration information
-- Support for CAMEL

TimeIfNoTariffSwitch ::= INTEGER(0..864000)
-- TimeIfNoTariffSwitch is measured in 100 millisecond intervals
-- Support for CAMEL

TimeIfTariffSwitch ::= SEQUENCE {
    timeSinceTariffSwitch [0] INTEGER(0..864000),
    tariffSwitchInterval [1] INTEGER(1..864000)    OPTIONAL,
    ...
}
-- timeSinceTariffSwitch and tariffSwitchInterval are measured in 100 millisecond intervals
-- Support for CAMEL

TimerID ::= ENUMERATED {
    tssf(0)
}
-- Indicates the timer to be reset.

TimerValue ::= Integer4
-- Indicates the timer value (in seconds).

Tone ::= SEQUENCE {
    toneID [0] Integer4,
    duration [1] Integer4    OPTIONAL,
    ...
}
-- The duration specifies the length of the tone in seconds, value 0 indicates infinite duration.

TriggerData ::= SEQUENCE {
    triggerId [0] TRIGGER.&id    ({SupportedTriggers}),
    triggerPar [1] TRIGGER.&Parameter ({SupportedTriggers}{@triggerId}),
    ...
}

TriggerDataIdentifier {B1: b1, B2: b2} ::= SEQUENCE {
    triggerID [0] EventTypeBCSM,
    profile [1] ProfileIdentifier {b2},
    extensions [2] Extensions {b1}    OPTIONAL,
    ...
}
-- It is outside the scope of this capability set whether all TDP types really apply

```



```

TriggerDPType ::= ENUMERATED {tdp-r(0)}

TriggerResults {B2: b2} ::=
  SEQUENCE SIZE (1..b2.&numOfTriggers) OF TriggerResult

TriggerResult ::= SEQUENCE {
  tDPIdentifer      [0] INTEGER,
  actionPerformed   [1] ActionPerformed,
  dpName            [2] EventTypeBCSM OPTIONAL,
  ...
}

Triggers {B2: b2} ::= SEQUENCE SIZE(1..b2.&numOfTriggers) OF Trigger

Trigger ::= SEQUENCE {
  tDPIdentifier     [0] INTEGER,
  dpName            [1] EventTypeBCSM OPTIONAL,
  ...
}

TriggerStatus ::= ENUMERATED {
  created           (0),
  alreadyExist     (1),
  deleted           (2),
  unknownTrigger   (3)
}

UsedCost ::= SEQUENCE {
  usedCurrencyFactor [1] CurrencyFactor,
-- imported from ES 201 296
  usedScale          [2] CurrencyScale OPTIONAL,
-- imported from ES 201 296
  reportedCurrency   [3] Currency DEFAULT noIndication,
-- imported from ES 201 296
  ...
}

UsedTime ::= SEQUENCE {
  usedDuration          [1] INTEGER (0..864000),
  -- Units are in 100 msec.
  usedDurationAfterSwitchOverTime [2] INTEGER (0..864000) OPTIONAL,
  -- Units are in 100 msec.
  ...
}

UsedPulses ::= SEQUENCE {
  usedPulseUnits [1] OCTET STRING (SIZE (2)),
  ...
}

USIInformation {B2: b2} ::= OCTET STRING (SIZE
  (b2.&minUSIInformationLength..
  b2.&maxUSIInformationLength))
-- Its content is network signalling/operator specific
-- Indicates the length of the USIInformation element, maxUSIInformationlength will depend on
-- the constraints imposed by the network signalling used to transport the USI information.
-- Its content is network signalling/operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

USIMonitorMode ::= ENUMERATED {
  monitoringActive (0),
  monitoringInactive (1)
}

-- Indicates if the relationship for the specified UTSI IE should be activated or deactivated.

USIServiceIndicator {B2: b2} ::= CHOICE {
  global OBJECT IDENTIFIER,
  local OCTET STRING (SIZE (
    b2.&minUSIServiceIndicatorLength..
    b2.&maxUSIServiceIndicatorLength))
}

```

```

}
-- In case of local its content is network signalling/operator specific
-- Its content is network signalling/operator specific.
-- The internal structure of this parameter can be defined using ASN.1 and the related Basic
-- Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length
-- information) is the BER encoding of the defined ASN.1 internal structure.
-- The tag of this parameter as defined by ETSI is never replaced.

VPNIndicator ::= BOOLEAN
-- This parameter is set to TRUE if the originating call part supports
-- VPN with PSSI information flows

WarningBeforeLimitReached ::= SEQUENCE {
    durationBeforeLimitReached [1] INTEGER (1..255),
    -- The values are in units of 1 second.
    warningToSend [2] Tone OPTIONAL,
    warningDirection [3] LegID OPTIONAL,
    ...
}

-- The Definition of range of constants Follows
highLayerCompatibilityLength INTEGER ::= 2
minCauseLength INTEGER ::= 2
numOfCounters INTEGER ::= 100
numOfInfoItems INTEGER ::= 5

END

```

## 6.2 Classes

```

IN-SSF-SCF-Classes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs4(40)
modules(1) in-ssf-scf-classes(7) version1(0)}

```

```

DEFINITIONS ::=

```

```

BEGIN

```

```

TRIGGER ::= CLASS {
    &Parameter OPTIONAL,
    &id INTEGER UNIQUE
}
WITH SYNTAX {
    [PARAMETER &Parameter]
    IDENTIFIED BY &id
}

```

```

--The &id field uniquely identifies a class of triggers.
--The &Parameter field defines the ASN.1 type for representing specific parameters
-- (e.g. criteria, scfAddress,...)
--associated with this class of triggers.

```

```

SCF-SSF-BOUNDS ::= CLASS {
    &minAchBillingChargingLength INTEGER OPTIONAL,
    &maxAchBillingChargingLength INTEGER OPTIONAL,
    &minBackwardGVNSLength INTEGER OPTIONAL,
    &maxBackwardGVNSLength INTEGER OPTIONAL,
    &maxBearerCapabilityLength INTEGER OPTIONAL,
    &minCalledDirectoryNumberLength INTEGER OPTIONAL,
    &maxCalledDirectoryNumberLength INTEGER OPTIONAL,
    &minCalledPartyBCDNumberLength INTEGER OPTIONAL,
    &maxCalledPartyBCDNumberLength INTEGER OPTIONAL,
    &minCalledPartyNumberLength INTEGER OPTIONAL,
    &maxCalledPartyNumberLength INTEGER OPTIONAL,
    &minCallingGeodeticLocationLength INTEGER OPTIONAL,
    &maxCallingGeodeticLocationLength INTEGER OPTIONAL,
    &minCallingPartyNumberLength INTEGER OPTIONAL,
    &maxCallingPartyNumberLength INTEGER OPTIONAL,
    &minCallResultcsiLength INTEGER OPTIONAL,
    &maxCallResultcsiLength INTEGER OPTIONAL,
    &maxCallReferenceLength INTEGER OPTIONAL,
    &minCarrierLength INTEGER OPTIONAL,
    &maxCarrierLength INTEGER OPTIONAL,
    &minChargedPartyNumberLength INTEGER OPTIONAL,

```

&maxChargedPartyNumberLength	INTEGER	OPTIONAL,
&maxCauseLength	INTEGER	OPTIONAL,
&minCommunicationTariffNum	INTEGER	OPTIONAL,
&maxCommunicationTariffNum	INTEGER	OPTIONAL,
&maxCNInfoLength	INTEGER	OPTIONAL,
&minDestinationIndexLength	INTEGER	OPTIONAL,
&maxDestinationIndexLength	INTEGER	OPTIONAL,
&minDigitsLength	INTEGER	OPTIONAL,
&maxDigitsLength	INTEGER	OPTIONAL,
&minDisplayInformationLength	INTEGER	OPTIONAL,
&maxDisplayInformationLength	INTEGER	OPTIONAL,
&minEventSpecificInformationChargingLength	INTEGER	OPTIONAL,
&maxEventSpecificInformationChargingLength	INTEGER	OPTIONAL,
&minEventTypeChargingLength	INTEGER	OPTIONAL,
&maxEventTypeChargingLength	INTEGER	OPTIONAL,
&minFCIBCCcs1Length	INTEGER	OPTIONAL,
&maxFCIBCCcs1Length	INTEGER	OPTIONAL,
&minFCIBCCcs2Length	INTEGER	OPTIONAL,
&maxFCIBCCcs2Length	INTEGER	OPTIONAL,
&minFCIBillingChargingDataLength	INTEGER	OPTIONAL,
&maxFCIBillingChargingDataLength	INTEGER	OPTIONAL,
&minCamelFCIBillingChargingDataLength	INTEGER	OPTIONAL,
&maxCamelFCIBillingChargingDataLength	INTEGER	OPTIONAL,
&minFCIBillingChargingLength	INTEGER	OPTIONAL,
&maxFCIBillingChargingLength	INTEGER	OPTIONAL,
&minFormatDataLength	INTEGER	OPTIONAL,
&maxFormatDataLength	INTEGER	OPTIONAL,
&minForwardGVNSLength	INTEGER	OPTIONAL,
&maxForwardGVNSLength	INTEGER	OPTIONAL,
&minGenericNameLength	INTEGER	OPTIONAL,
&maxGenericNameLength	INTEGER	OPTIONAL,
&minGenericNumberLength	INTEGER	OPTIONAL,
&maxGenericNumberLength	INTEGER	OPTIONAL,
&maxGlobalCallReferenceLength	INTEGER	OPTIONAL,
&maxInitialTimeInterval	INTEGER	OPTIONAL,
&maxINServiceCompatibilityIndLength	INTEGER	OPTIONAL,
&minIPAvailableLength	INTEGER	OPTIONAL,
&maxIPAvailableLength	INTEGER	OPTIONAL,
&minIPSSPCapabilitiesLength	INTEGER	OPTIONAL,
&maxIPSSPCapabilitiesLength	INTEGER	OPTIONAL,
&minISDNAccessRelatedInfoLength	INTEGER	OPTIONAL,
&maxISDNAccessRelatedInfoLength	INTEGER	OPTIONAL,
&minLocationNumberLength	INTEGER	OPTIONAL,
&maxLocationNumberLength	INTEGER	OPTIONAL,
&minMidCallControlInfoNum	INTEGER	OPTIONAL,
&maxMidCallControlInfoNum	INTEGER	OPTIONAL,
&minOriginalCalledPartyIDLength	INTEGER	OPTIONAL,
&maxOriginalCalledPartyIDLength	INTEGER	OPTIONAL,
&minReasonLength	INTEGER	OPTIONAL,
&maxReasonLength	INTEGER	OPTIONAL,
&minRedirectingPartyIDLength	INTEGER	OPTIONAL,
&maxRedirectingPartyIDLength	INTEGER	OPTIONAL,
&minRequestedUTSINum	INTEGER	OPTIONAL,
&maxRequestedUTSINum	INTEGER	OPTIONAL,
&minRouteListLength	INTEGER	OPTIONAL,
&maxRouteListLength	INTEGER	OPTIONAL,
&minRouteingNumberLength	INTEGER	OPTIONAL,
&maxRouteingNumberLength	INTEGER	OPTIONAL,
&minScfIDLength	INTEGER	OPTIONAL,
&maxScfIDLength	INTEGER	OPTIONAL,
&minSCIBillingChargingLength	INTEGER	OPTIONAL,
&maxSCIBillingChargingLength	INTEGER	OPTIONAL,
&minSDSSinformationLength	INTEGER	OPTIONAL,
&maxSDSSinformationLength	INTEGER	OPTIONAL,
&minServiceInteractionIndicatorsLength	INTEGER	OPTIONAL,
&maxServiceInteractionIndicatorsLength	INTEGER	OPTIONAL,
&minSFBillingChargingLength	INTEGER	OPTIONAL,
&maxSFBillingChargingLength	INTEGER	OPTIONAL,
&minSubTariffControlLen	INTEGER	OPTIONAL,
&maxSubTariffControlLen	INTEGER	OPTIONAL,
&minTimeAndTimezoneLength	INTEGER	OPTIONAL,
&maxTimeAndTimezoneLength	INTEGER	OPTIONAL,
&minTariffIndicatorsLen	INTEGER	OPTIONAL,
&maxTariffIndicatorsLen	INTEGER	OPTIONAL,
&minUSIInformationLength	INTEGER	OPTIONAL,
&maxUSIInformationLength	INTEGER	OPTIONAL,
&minUSIServiceIndicatorLength	INTEGER	OPTIONAL,
&maxUSIServiceIndicatorLength	INTEGER	OPTIONAL,

&numOfBCSMEvents	INTEGER	OPTIONAL,
&numOfBCUSMEvents	INTEGER	OPTIONAL,
&numOfChargingEvents	INTEGER	OPTIONAL,
&numOfCSAs	INTEGER	OPTIONAL,
&numOfCSs	INTEGER	OPTIONAL,
&numOfGenericNumbers	INTEGER	OPTIONAL,
&numOfINProfile	INTEGER	OPTIONAL,
&numOfTriggers	INTEGER	OPTIONAL,
&numOfInServiceCompatibilityIndLength	INTEGER	OPTIONAL,
&numOfLegs	INTEGER	OPTIONAL,
&numOfMessageIDs	INTEGER	OPTIONAL,
&maxAmount	INTEGER	OPTIONAL,
&maxInitialUnitIncrement	INTEGER	OPTIONAL,
&maxScalingFactor	INTEGER	OPTIONAL,
&maxSegmentsPerDataInterval	INTEGER	OPTIONAL,
&ub-nbCall	INTEGER	OPTIONAL,
&numOfAddresses	INTEGER	OPTIONAL,
&maxAlternativeIdentities	INTEGER	OPTIONAL
}		
WITH SYNTAX		
{		
[MINIMUM-FOR-ACH-BILLING-CHARGING	&minAchBillingChargingLength]	
[MAXIMUM-FOR-ACH-BILLING-CHARGING	&maxAchBillingChargingLength]	
[MINIMUM-FOR-BACKWARD-GVNS	&minBackwardGVNSLength]	
[MAXIMUM-FOR-BACKWARD-GVNS	&maxBackwardGVNSLength]	
[MAXIMUM-FOR-BEARER-CAPABILITY	&maxBearerCapabilityLength]	
[MINIMUM-FOR-CALLED-DIRECTORY-NUMBER	&minCalledDirectoryNumberLength]	
[MAXIMUM-FOR-CALLED-DIRECTORY-NUMBER	&maxCalledDirectoryNumberLength]	
[MINIMUM-FOR-CALLED-PARTY-BCD-NUMBER	&minCalledPartyBCDNumberLength]	
[MAXIMUM-FOR-CALLED-PARTY-BCD-NUMBER	&maxCalledPartyBCDNumberLength]	
[MINIMUM-FOR-CALLED-PARTY-NUMBER	&minCalledPartyNumberLength]	
[MAXIMUM-FOR-CALLED-PARTY-NUMBER	&maxCalledPartyNumberLength]	
[MINIMUM-FOR-CALLING-GEODETIC-LOCATION	&minCallingGeodeticLocationLength]	
[MAXIMUM-FOR-CALLING-GEODETIC-LOCATION	&maxCallingGeodeticLocationLength]	
[MINIMUM-FOR-CALLING-PARTY-NUMBER	&minCallingPartyNumberLength]	
[MAXIMUM-FOR-CALLING-PARTY-NUMBER	&maxCallingPartyNumberLength]	
[MINIMUM-FOR-CALL-RESULT-CSONE	&minCallResultcs1Length]	
[MAXIMUM-FOR-CALL-RESULT-CSONE	&maxCallResultcs1Length]	
[MAXIMUM-FOR-CALL-REFERENCE	&maxCallReferenceLength]	
[MINIMUM-FOR-CARRIER	&minCarrierLength]	
[MAXIMUM-FOR-CARRIER	&maxCarrierLength]	
[MINIMUM-FOR-CHARGED-PARTYNUMBER	&minChargedPartyNumberLength]	
[MAXIMUM-FOR-CHARGED-PARTYNUMBER	&maxChargedPartyNumberLength]	
[MAXIMUM-FOR-CAUSE	&maxCauseLength]	
[MINIMUM-FOR-COMMUNICATION-TARIFF-NUM	&minCommunicationTariffNum]	
[MAXIMUM-FOR-COMMUNICATION-TARIFF-NUM	&maxCommunicationTariffNum]	
[MAXIMUM-FOR-CNINFO	&maxCNInfoLength]	
[MINIMUM-FOR-DESTINATION	&minDestinationIndexLength]	
[MAXIMUM-FOR-DESTINATION	&maxDestinationIndexLength]	
[MINIMUM-FOR-DIGITS	&minDigitsLength]	
[MAXIMUM-FOR-DIGITS	&maxDigitsLength]	
[MINIMUM-FOR-DISPLAY	&minDisplayInformationLength]	
[MAXIMUM-FOR-DISPLAY	&maxDisplayInformationLength]	
[MINIMUM-FOR-EVENT-SPECIFIC-CHARGING	&minEventSpecificInformationChargingLength]	
[MAXIMUM-FOR-EVENT-SPECIFIC-CHARGING	&maxEventSpecificInformationChargingLength]	
[MINIMUM-FOR-EVENT-TYPE-CHARGING	&minEventTypeChargingLength]	
[MAXIMUM-FOR-EVENT-TYPE-CHARGING	&maxEventTypeChargingLength]	
[MINIMUM-FOR-FCI-CSONE-BILLING-CHARGING	&minFCIBCCcs1Length]	
[MAXIMUM-FOR-FCI-CSONE-BILLING-CHARGING	&maxFCIBCCcs1Length]	
[MINIMUM-FOR-FCI-CSTWO-BILLING-CHARGING	&minFCIBCCcs2Length]	
[MAXIMUM-FOR-FCI-CSTWO-BILLING-CHARGING	&maxFCIBCCcs2Length]	
[MINIMUM-FOR-FCI-BILLING-CHARGING-DATA	&minFCIBillingChargingDataLength]	
[MAXIMUM-FOR-FCI-BILLING-CHARGING-DATA	&maxFCIBillingChargingDataLength]	
[MINIMUM-FOR-CAMELFCI-BILLING-CHARGING-DATA	&minCamelFCIBillingChargingDataLength]	
[MAXIMUM-FOR-CAMELFCI-BILLING-CHARGING-DATA	&maxCamelFCIBillingChargingDataLength]	
[MINIMUM-FOR-FCI-BILLING-CHARGING	&minFCIBillingChargingLength]	
[MAXIMUM-FOR-FCI-BILLING-CHARGING	&maxFCIBillingChargingLength]	
[MINIMUM-FOR-FORMAT-DATA-LENGTH	&minFormatDataLength]	
[MAXIMUM-FOR-FORMAT-DATA-LENGTH	&maxFormatDataLength]	
[MINIMUM-FOR-FORWARD-GVNS	&minForwardGVNSLength]	
[MAXIMUM-FOR-FORWARD-GVNS	&maxForwardGVNSLength]	
[MINIMUM-FOR-GENERIC-NAME	&minGenericNameLength]	
[MAXIMUM-FOR-GENERIC-NAME	&maxGenericNameLength]	
[MINIMUM-FOR-GENERIC-NUMBER	&minGenericNumberLength]	
[MAXIMUM-FOR-GENERIC-NUMBER	&maxGenericNumberLength]	
[MAXIMUM-FOR-GLOBAL-CALLREF	&maxGlobalCallReferenceLength]	
[MAXIMUM-FOR-INITIAL-TIME-INTERVAL	&maxInitialTimeInterval]	

```

[ MAXIMUM-FOR-IN-SERVICE-COMPATIBILITY                &maxINServiceCompatibilityIndLength]
[ MINIMUM-FOR-IP-AVAILABLE                             &minIPAvailableLength]
[ MAXIMUM-FOR-IP-AVAILABLE                             &maxIPAvailableLength]
[ MINIMUM-FOR-IP-SSP-CAPABILITIES                     &minIPSSPCapabilitiesLength]
[ MAXIMUM-FOR-IP-SSP-CAPABILITIES                     &maxIPSSPCapabilitiesLength]
[ MINIMUM-FOR-ISDN-ACCESS-RELATED-INFO                &minISDNAccessRelatedInfoLength]
[ MAXIMUM-FOR-ISDN-ACCESS-RELATED-INFO                &maxISDNAccessRelatedInfoLength]
[ MINIMUM-FOR-LOCATION-NUMBER                           &minLocationNumberLength]
[ MAXIMUM-FOR-LOCATION-NUMBER                           &maxLocationNumberLength]
[ MINIMUM-FOR-MID-CALL-CONTROL-INFO                   &minMidCallControlInfoNum]
[ MAXIMUM-FOR-MID-CALL-CONTROL-INFO                   &maxMidCallControlInfoNum]
[ MINIMUM-FOR-ORIGINAL-CALLED-PARTY-ID                &minOriginalCalledPartyIDLength]
[ MAXIMUM-FOR-ORIGINAL-CALLED-PARTY-ID                &maxOriginalCalledPartyIDLength]
[ MINIMUM-FOR-REASON                                   &minReasonLength]
[ MAXIMUM-FOR-REASON                                   &maxReasonLength]
[ MINIMUM-FOR-REDIRECTING-ID                           &minRedirectingPartyIDLength]
[ MAXIMUM-FOR-REDIRECTING-ID                           &maxRedirectingPartyIDLength]
[ MINIMUM-FOR-REQUESTED-UTSI-NUM                     &minRequestedUTSINum]
[ MAXIMUM-FOR-REQUESTED-UTSI-NUM                     &maxRequestedUTSINum]
[ MINIMUM-FOR-ROUTE-LIST                               &minRouteListLength]
[ MAXIMUM-FOR-ROUTE-LIST                               &maxRouteListLength]
[ MINIMUM-FOR-ROUTING-NUMBER                           &minRoutingNumberLength]
[ MAXIMUM-FOR-ROUTING-NUMBER                           &maxRoutingNumberLength]
[ MINIMUM-FOR-SCF-ID                                   &minScfIDLength]
[ MAXIMUM-FOR-SCF-ID                                   &maxScfIDLength]
[ MINIMUM-FOR-SCI-BILLING-CHARGING                    &minSCIBillingChargingLength]
[ MAXIMUM-FOR-SCI-BILLING-CHARGING                    &maxSCIBillingChargingLength]
[ MINIMUM-FOR-SDSS-INFORMATION                         &minSDSSinformationLength]
[ MAXIMUM-FOR-SDSS-INFORMATION                         &maxSDSSinformationLength]
[ MINIMUM-FOR-SII                                       &minServiceInteractionIndicatorsLength]
[ MAXIMUM-FOR-SII                                       &maxServiceInteractionIndicatorsLength]
[ MINIMUM-FOR-SF-BILLING-CHARGING                     &minSFBillingChargingLength]
[ MAXIMUM-FOR-SF-BILLING-CHARGING                     &maxSFBillingChargingLength]
[ MINIMUM-FOR-SUB-TARIFF-CONTROL-LENGTH               &minSubTariffControlLen]
[ MAXIMUM-FOR-SUB-TARIFF-CONTROL-LENGTH               &maxSubTariffControlLen]
[ MINIMUM-FOR-TIME-AND-TIMEZONE-LENGTH                &minTimeAndTimezoneLength]
[ MAXIMUM-FOR-TIME-AND-TIMEZONE-LENGTH                &maxTimeAndTimezoneLength]
[ MINIMUM-FOR-TARIFF-INDICATORS-LENGTH                &minTariffIndicatorsLen]
[ MAXIMUM-FOR-TARIFF-INDICATORS-LENGTH                &maxTariffIndicatorsLen]
[ MINIMUM-FOR-USI-INFORMATION                          &minUSIInformationLength]
[ MAXIMUM-FOR-USI-INFORMATION                          &maxUSIInformationLength]
[ MINIMUM-FOR-USI-SERVICE-INDICATOR                   &minUSIServiceIndicatorLength]
[ MAXIMUM-FOR-USI-SERVICE-INDICATOR                   &maxUSIServiceIndicatorLength]
[ NUM-OF-BCSM-EVENT                                    &numOfBCSMEvents]
[ NUM-OF-BCUSM-EVENT                                    &numOfBCUSMEvents]
[ NUM-OF-CHARGING-EVENT                                &numOfChargingEvents]
[ NUM-OF-CSAS                                          &numOfCSAs]
[ NUM-OF-CSS                                           &numOfCSSs]
[ NUM-OF-GENERIC-NUMBERS                               &numOfGenericNumbers]
[ NUM-OF-INPROFILE                                     &numOfINProfile]
[ NUM-OF-SEVERALTRIGGER                                &numOfTriggers]
[ NUM-OF-IN-SERVICE-COMPATIBILITY-ID                  &numOfInServiceCompatibilityIndLength]
[ NUM-OF-LEGS                                          &numOfLegs]
[ NUM-OF-MESSAGE-IDS                                   &numOfMessageIDs]
[ MAXIMUM-FOR-AMOUNT                                    &maxAmount]
[ MAXIMUM-FOR-INITIAL-UNIT-INCREMENT                  &maxInitialUnitIncrement]
[ MAXIMUM-FOR-SCALING-FACTOR                           &maxScalingFactor]
[ MAXIMUM-FOR-SEGMENTS-PER-DATA-INTERVAL              &maxSegmentsPerDataInterval]
[ MAXIMUM-FOR-UB-NB-CALL                               &ub-nbCall]
[ NUM-OF-ADDRESSES                                    &numOfAddresses]
[ MAXIMUM-NB-OF-ALTERNATIVE-IDENTITIES                &maxAlternativeIdentities]

}

```

END

## 6.3 Operations and Arguments

```
IN-SSF-SCF-ops-args {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs4(40)
modules(1) in-ssf-scf-ops-args(8) version1(0)}
```

```
DEFINITIONS IMPLICIT TAGS ::=
```

```
BEGIN
```

```
IMPORTS
```

```
    common-classes,
    common-datatypes,
    errorTypes,
    scf-srf-classes,
    scf-srf-datatypes,
    ssf-scf-classes,
    ssf-scf-datatypes,
    operationcodes,
    ros-InformationObjects,
    tc-Messages
```

```
FROM IN-object-identifiers
    {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs4(40) modules(1) in-
object-identifiers(0) version1(0)}
```

```
    OPERATION
```

```
FROM Remote-Operations-Information-Objects ros-InformationObjects
```

```
    AddOnChargingInformation,
    ChargingTariffInformation,
    ChargingMessageType
```

```
FROM Tariffing-DataTypes {itu-t(0) identified-organization etsi (0) 1296 version2(3)}
```

```
    ISDN-AddressString,
    IMSI,
    Ext-BasicServiceCode
```

```
-- Refer to TS 129 002 for encoding.
```

```
FROM MAP-CommonDataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-CommonDataTypes(18) version6(6)}
```

```
    CUG-Index,
    CUG-Interlock,
    LocationInformation,
    SubscriberState
```

```
-- Refer to TS 129 002 for encoding.
```

```
FROM MAP-MS-DataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-MS-DataTypes(11) version6(6)}
```

```
    CallReferenceNumber,
```

```
-- Refer to TS 129 002 for encoding.
```

```
    SuppressionOfAnnouncement
```

```
FROM MAP-CH-DataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-CH-DataTypes(13) version6(6)}
```

```
    COMMON-BOUNDS
```

```
FROM IN-common-classes common-classes
```

```
    SCF-SSF-BOUNDS
```

```
FROM IN-SSF-SCF-Classes ssf-scf-classes
```

```
    SCF-SRF-BOUNDS
```

```
FROM IN-CS3-scf-srf-classes scf-srf-classes
```

```
    opcode-activateServiceFiltering,
    opcode-activityTest,
    opcode-applyCharging,
    opcode-applyChargingReport,
    opcode-assistRequestInstructions,
    opcode-callFiltering,
```

```

opcode-callGap,
opcode-callInformationReport,
opcode-callInformationRequest,
opcode-cancel,
opcode-cancelStatusReportRequest,
opcode-collectInformation,
opcode-connect,
opcode-connectToResource,
opcode-continue,
opcode-continueWithArgument,
opcode-createCallSegmentAssociation,
opcode-createOrRemoveTriggerData,
opcode-disconnectForwardConnection,
opcode-dFCWithArgument,
opcode-disconnectLeg,
opcode-entityReleased,
opcode-establishTemporaryConnection,
opcode-eventNotificationCharging,
opcode-eventReportBCSM,
opcode-furnishChargingInformation,
opcode-initialDP,
opcode-initiateCallAttempt,
opcode-manageTriggerData,
opcode-mergeCallSegments,
opcode-moveCallSegments,
opcode-moveLeg,
opcode-releaseCall,
opcode-reportUTSI,
opcode-requestCurrentStatusReport,
opcode-requestEveryStatusChangeReport,
opcode-requestFirstStatusMatchReport,
opcode-requestNotificationChargingEvent,
opcode-requestReportBCSMEvent,
opcode-requestReportUTSI,
opcode-resetTimer,
opcode-selectFacility,
opcode-sendChargingInformation,
opcode-sendSTUI,
opcode-serviceFilteringResponse,
opcode-setServiceProfile,
opcode-splitLeg,
opcode-statusReport

```

FROM IN-operationcodes operationcodes

```

Extensions {},
Integer4,
InvokeID

```

FROM IN-common-datatypes common-datatypes

```

ActionIndicator,
ActionPerformed,
AChBillingChargingCharacteristics {},
AChChargingAddress {},
AdditionalCallingPartyNumber {},
AlertingPattern,
ApplicationTimer,
AssistingSSPIPRoutingAddress {},
BackwardGVNS {},
BCSMEvent {},
BearerCapability {},
CalledDirectoryNumber {},
CalledPartyNumber {},
CalledPartyBCDNumber {},
CallingGeodeticLocation {},
CallingPartyBusinessGroupID,
CallingPartyNumber {},
CallingPartysCategory,
CallReference {},
CallResult {},
CallSegmentID {},
CallSupervision {},
Carrier {},
Cause {},
CCSS,
CGEncountered,

```

ChargeNumber {},  
 ChargingControlType,  
 ChargingEvent {},  
 CNInfo {},  
 ControlType,  
 CorrelationID {},  
 CountersValue,  
 CreateOrRemoveIndicator,  
 CSAID {},  
 CutAndPaste,  
 DateAndTime,  
 DefaultFaultHandling {},  
 DestinationIndex {},  
 DestinationRoutingAddress {},  
 Digits {},  
 DisplayInformation {},  
 Duration,  
 EventSpecificInformationBCSM {},  
 EventSpecificInformationCharging {},  
 EventTypeBCSM,  
 EventTypeCharging {},  
 EventTypeTariff,  
 FCIBillingChargingCharacteristics {},  
 FilteredCallTreatment {},  
 FilteringCharacteristics,  
 FilteringCriteria {},  
 FilteringTimeOut,  
 ForwardCallIndicators,  
 ForwardGVNS {},  
 GapCriteria {},  
 GapIndicators,  
 GapTreatment {},  
 GenericName {},  
 GenericNumbers {},  
 GlobalCallReference {},  
 HighLayerCompatibility,  
 initialCallSegment,  
 InitialDPArgExtension,  
 INprofile {},  
 INServiceCompatibilityIndication {},  
 INServiceCompatibilityResponse,  
 IPAvailable {},  
 IPRelatedInformation {},  
 IPRoutingAddress {},  
 IPSSPCapabilities {},  
 ISDNAccessRelatedInformation {},  
 LegID,  
 leg1,  
 LocationNumber {},  
 MiscCallInfo,  
 MonitorMode,  
 NAOLIInfo,  
 OCSTApplicable,  
 OriginalCalledPartyID {},  
 ProfileIdentifier {},  
 Reason {},  
 RedirectingPartyID {},  
 RedirectionInformation,  
 RegistrarIdentifier,  
 ReportCondition,  
 RouteingNumber {},  
 RequestedInformationList {},  
 RequestedInformationTypeList,  
 RequestedUTSILList {},  
 ResourceID {},  
 ResourceStatus,  
 ResponseCondition,  
 RouteList {},  
 ScfID {},  
 SCIBillingChargingCharacteristics {},  
 SDSSinformation {},  
 ServiceInteractionIndicators {},  
 ServiceInteractionIndicatorsTwo,  
 ServiceKey,  
 TDPIdentifier,  
 TerminalType,  
 TimeAndTimezone {},  
 TimerID,  
 TimerValue,



```

    TriggerData,
    TriggerDataIdentifier {},
    TriggerDPTYPE,
    Triggers {},
    TriggerStatus,
    TriggerResults {},
    USIInformation {},
    USIServiceIndicator {},
    VPNIndicator
FROM IN-SSF-SCF-datatypes ssf-scf-datatypes

    InformationToSend {}
FROM IN-CS3-scf-srf-datatypes scf-srf-datatypes

    cancelFailed,
    eTCFailed,
    improperCallerResponse,
    missingCustomerRecord,
    missingParameter,
    parameterOutOfRange,
    requestedInfoError,
    systemFailure,
    taskRefused,
    unavailableResource,
    unexpectedComponentSequence,
    unexpectedDataValue,
    unexpectedParameter,
    unknownLegID,
    unknownResource
FROM IN-errorTypes errorTypes
;
-- The following three definitions are local short-hand notation for convenience.
B1::= COMMON-BOUNDS      -- defined in part 1
B2::= SCF-SSF-BOUNDS    -- defined in this part
B3::= SCF-SRF-BOUNDS    -- defined in EN 301 931-3

-- Operations and Arguments:

activateServiceFiltering {B1: b1, B2: b2, B3: b3} OPERATION::= {
    ARGUMENT    ActivateServiceFilteringArg {b1, b2, b3}
    RETURN RESULT TRUE
    ERRORS      {missingParameter |
                 parameterOutOfRange |
                 systemFailure |
                 taskRefused |
                 unexpectedComponentSequence |
                 unexpectedParameter
                }
    CODE         opcode-activateServiceFiltering
}
-- Direction: SCF -> SSF, Timer: Tasf
-- When receiving this operation, the SSF handles calls to destination in a specified manner
-- without sending queries for every detected call. It is used for example for providing
-- televoting or mass calling services. Simple registration functionality (counters) and
-- announcement control may be located at the SSF. The operation initializes the specified
-- counters in the SSF.

ActivateServiceFilteringArg {B1: b1, B2: b2, B3: b3} ::= SEQUENCE {
    filteredCallTreatment      [0] FilteredCallTreatment {b1, b2, b3},
    filteringCharacteristics    [1] FilteringCharacteristics,
    filteringTimeOut           [2] FilteringTimeOut,
    filteringCriteria           [3] FilteringCriteria {b2},
    startTime                  [4] DateAndTime OPTIONAL,
    extensions                  [5] Extensions {b1} OPTIONAL,
    ...
}

activityTest OPERATION::= {
    RETURN RESULT TRUE
    ALWAYS RESPONDS FALSE
    CODE         opcode-activityTest
}
-- Direction: SCF -> SSF or SSF-> SCF, Timer: Tat
-- This operation is used to check for the continued existence of a relationship between the SCF
-- and SSF. If the relationship is still in existence, then the SSF will respond. If no reply is

```

-- received, then the SCF will assume that the SSF has failed in some way and will take the appropriate action.. As an option, this operation may be used in the reverse direction by the SSF to check for the continued existence of a relationship with the SCF.

```
applyCharging {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    ApplyChargingArg {b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingParameter |
               unexpectedComponentSequence |
               unexpectedParameter |
               unexpectedDataValue |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unknownLegID}
  ALWAYS RESPONDS FALSE    CODE    opcode-applyCharging
}
```

-- Direction: SCF -> SSF, Timer: T<sub>ac</sub>

-- This operation is used for interacting from the SCF with the SSF charging mechanisms.  
 -- The ApplyChargingReport operation provides the feedback from the SSF to the SCF.  
 -- This operation is can also be used to instruct the SSF to release the call regarding  
 -- some condition.

```
ApplyChargingArg {B1: b1, B2: b2} ::= SEQUENCE {
  aChBillingChargingCharacteristics [0] AChBillingChargingCharacteristics {b2} OPTIONAL,
  -- parameter is made optional to allow backward compatibility and to avoid a mandatory use
  -- of a CS1 defined operator specific Octet string.
  sendCalculationToSCPIndication [1] BOOLEAN OPTIONAL,
  -- This parameter, if present, shall be set to TRUE.
  -- The parameter is ignored if CS2 or CS3 is used.
  aChChargingAddress AChChargingAddress {b2} DEFAULT legID:sendingSideID:leg1,
  extensions [3] Extensions {b1} OPTIONAL,
  callSupervision [52] CallSupervision {b2} OPTIONAL,
  ...
}
```

```
applyChargingReport {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    ApplyChargingReportArg {b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingParameter |
               unexpectedComponentSequence |
               unexpectedParameter |
               unexpectedDataValue |
               parameterOutOfRange |
               systemFailure |
               taskRefused}
  ALWAYS RESPONDS FALSE
  CODE    opcode-applyChargingReport
}
```

-- Direction: SSF -> SCF, Timer: T<sub>acr</sub>

-- This operation is used by the SSF to report to the SCF the occurrence of a specific charging event as requested by the SCF using the ApplyCharging operation.

```
ApplyChargingReportArg { B1: b1, B2: b2} ::= CallResult {b1, b2}
```

```
assistRequestInstructions {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    AssistRequestInstructionsArg {b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingCustomerRecord |
               missingParameter |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter}
  ALWAYS RESPONDS FALSE
  CODE    opcode-assistRequestInstructions
}
```

-- Direction: SSF -> SCF or SRF -> SCF, Timer: T<sub>ari</sub>

-- This operation is used when there is an assist or a hand-off procedure and may be sent by the SSF or SRF to the SCF. This operation is sent by the assisting SSF to SCF, when the initiating SSF has set up a connection to the SRF or to the assisting SSF as a result of receiving an EstablishTemporaryConnection or Connect operation (in the case of hand-off) from the SCF.

```

AssistRequestInstructionsArg {B1: b1, B2: b2} ::= SEQUENCE {
    correlationID      [0] CorrelationID {b2},
    iPAvailable        [1] IPAavailable {b2}      OPTIONAL,
    iPSSPCapabilities [2] IPSSPCapabilities {b2}  OPTIONAL,
    extensions         [3] Extensions {b1}       OPTIONAL,
    ...
}
-- OPTIONAL denotes network operator specific use. The value of the correlationID may be the
-- Called Party Number supplied by the initiating SSF.

callFiltering {B1:b1,B2:b2} OPERATION ::= {
    ARGUMENT      CallFilteringArg {b1,b2}
    RETURN RESULT FALSE
    ERRORS        {missingParameter |
                  parameterOutOfRange |
                  systemFailure |
                  taskRefused |
                  unexpectedComponentSequence |
                  unexpectedDataValue |
                  unexpectedParameter
                 }
    ALWAYS RESPONDS FALSE
    CODE          opcode-callFiltering
}
-- Direction: SCF -> SSF, Timer: Tcf
-- This operation is used to allow the SCF to influence basic call gapping procedures based in the
-- CCF by sending information from the SCF to the SSF. The SSF relays the received information
-- transparently to the CCF. This way, the SCF can influence the rate at which call attempts are
-- allowed through. The operation thus influences the filtering of calls, as opposed to service
-- requests as is done by the Callgap operation.

CallFilteringArg { B1:b1,B2:b2 } ::= SEQUENCE {
    destinationIndex [0] DestinationIndex {b2},
    gapIndicators     [1] GapIndicators,
    registrarIdentifier [2] RegistrarIdentifier OPTIONAL,
    ...,
    ...,
    extensions        [3] Extensions {b1}     OPTIONAL
}
-- OPTIONAL denotes network operator optional.

callGap { B1: b1, B2: b2, B3: b3 } OPERATION ::= {
    ARGUMENT      CallGapArg {b1, b2, b3}
    RETURN RESULT FALSE
    ALWAYS RESPONDS FALSE
    CODE          opcode-callGap
}
-- Direction: SCF -> SSF, Timer: Tcg
-- This operation is used to request the SSF to reduce the rate at which specific
-- service requests are sent to the SCF.
-- Use of this operation by the SCF to gap queries and updates at the SDF is outside
-- the scope of this capability set.

CallGapArg {B1: b1, B2: b2, B3: b3} ::= SEQUENCE {
    gapCriteria      [0] GapCriteria {b2},
    gapIndicators    [1] GapIndicators,
    controlType      [2] ControlType OPTIONAL,
    gapTreatment     [3] GapTreatment {b1, b2, b3} OPTIONAL,
    extensions       [4] Extensions {b1}     OPTIONAL,
    ...
}
-- OPTIONAL denotes network operator optional. If gapTreatment is not present, the SSF will use
-- a default treatment depending on network operator implementation.

callInformationReport { B1: b1, B2: b2 } OPERATION ::= {
    ARGUMENT      CallInformationReportArg { b1, b2}
    RETURN RESULT FALSE
    ALWAYS RESPONDS FALSE
    CODE          opcode-callInformationReport
}
-- Direction: SSF -> SCF, Timer: Tcirp
-- This operation is used to send specific call information for a single call to
-- the SCF as requested by the SCF
-- in a previous CallInformationRequest.

```

```

CallInformationReportArg {B1: b1, B2: b2} ::= SEQUENCE {
    requestedInformationList [0] RequestedInformationList {b2},
    extensions               [2] Extensions {b1}          OPTIONAL,
    legID                    [3] LegID                    OPTIONAL,
    ...
}
-- OPTIONAL denotes network operator optional.

callInformationRequest {B1: b1} OPERATION ::= {
    ARGUMENT    CallInformationRequestArg {b1}
    RETURN RESULT FALSE
    ERRORS      {missingParameter |
                 parameterOutOfRange |
                 requestedInfoError |
                 systemFailure |
                 taskRefused |
                 unexpectedComponentSequence |
                 unexpectedDataValue |
                 unexpectedParameter |
                 unknownLegID}
    ALWAYS RESPONDS FALSE
    CODE        opcode-callInformationRequest
}
-- Direction: SCF -> SSF, Timer: T_cirq
-- This operation is used to request the SSF to record specific information about a
-- single call and report it to
-- the SCF (with a CallInformationReport operation).

CallInformationRequestArg {B1: b1} ::= SEQUENCE {
    requestedInformationTypeList [0] RequestedInformationTypeList,
    extensions                   [2] Extensions {b1}          OPTIONAL,
    legID                        [3] LegID                    OPTIONAL,
    ...
}
-- OPTIONAL denotes network operator optional.

cancel {B2: b2} OPERATION ::= {
    ARGUMENT    CancelArg {b2}
    RETURN RESULT FALSE
    ERRORS      {cancelFailed |
                 missingParameter |
                 taskRefused}
    ALWAYS RESPONDS FALSE
    CODE        opcode-cancel
}
-- Direction: SCF -> SSF, or SCF -> SRF, Timer: T_can
-- This operation cancels the correlated previous operation or all previous requests
-- This operation can also be used to cancel all outstanding requests and enable the
-- state machine (SSF) to go to idle.
-- In this case the Cancel operation does not specify any specific operation to be cancelled.
-- For the SCF-SRF operations that can be cancelled, refer to Part 3 of ITU-T Recommendation Q.1238

CancelArg {B2: b2} ::= CHOICE {
    invokeID [0] InvokeID,
    allRequests [1] NULL,
    callSegmentToCancel [2] SEQUENCE {
        invokeID [0] InvokeID,
        callSegmentID [1] CallSegmentID {b2},
        ...
    },
    allRequestsForCallSegment [3] CallSegmentID {b2},
    ...
}
-- The InvokeID has the same value as that which was used for the SCF-SRF operation,
-- i.e. is used to identify the correlated previous SCF-SRF operation to be cancelled.

cancelStatusReportRequest { B1: b1, B2: b2 } OPERATION ::= {
    ARGUMENT    CancelStatusReportRequestArg {b1, b2}
    RETURN RESULT FALSE
    ERRORS      {cancelFailed |
                 missingParameter |
                 taskRefused}
    ALWAYS RESPONDS FALSE
    CODE        opcode-cancelStatusReportRequest
}
-- Direction: SCF -> SSF, Timer: T_csr
-- This operation cancels the following processes: RequestFirstStatusMatchReport and
-- RequestEveryStatusChangeReport.

```

```

CancelStatusReportRequestArg {B1: b1, B2: b2} ::= SEQUENCE {
    resourceID [0] ResourceID {b2} OPTIONAL,
    extensions [1] Extensions {b1} OPTIONAL,
    ...
}

collectInformation {B1: b1} OPERATION ::= {
    ARGUMENT CollectInformationArg { b1} OPTIONAL TRUE
    RETURN RESULT FALSE
    ERRORS {missingParameter |
        parameterOutOfRange |
        systemFailure |
        taskRefused |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter}
    ALWAYS RESPONDS FALSE
    CODE opcode-collectInformation
}
-- Direction: SCF -> SSF, Timer: Tci
-- This operation is used to request the SSF to perform the originating basic call
-- processing actions to prompt a calling party for destination information,
-- then collect destination information according to a specified
-- numbering plan (e.g. for virtual private networks).

CollectInformationArg {B1: b1} ::= SEQUENCE {
    extensions [4] Extensions {b1} OPTIONAL,
    ...
}

connect {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT ConnectArg {b1, b2}
    RETURN RESULT FALSE
    ERRORS {missingParameter |
        parameterOutOfRange |
        systemFailure |
        taskRefused |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter}
    ALWAYS RESPONDS FALSE
    CODE opcode-connect
}
-- Direction: SCF -> SSF, Timer: Tcon
-- This operation is used to request the SSF to perform the call processing actions to route
-- or forward a call to a specified destination. To do so, the SSF may or may not use
-- destination information from the calling party (e.g. dialed digits) and existing call
-- setup information (e.g. route index to a list of trunk groups), depending on
-- the information provided by the SCF.
-- When address information is only included in the Connect operation, call processing
-- resumes at PIC Analyse_Information in the O-BCSM.
-- When address information and routing information is included, call processing resumes at PIC
-- Select_Route.

ConnectArg {B1: b1, B2: b2} ::= SEQUENCE {
    destinationRoutingAddress [0] DestinationRoutingAddress { b2},
    alertingPattern [1] AlertingPattern OPTIONAL,
    correlationID [2] CorrelationID { b2} OPTIONAL,
    cutAndPaste [3] CutAndPaste OPTIONAL,
    isDNAccessRelatedInformation [5] ISDNAccessRelatedInformation {b2} OPTIONAL,
    originalCalledPartyID [6] OriginalCalledPartyID { b2} OPTIONAL,
    routeList [7] RouteList { b2} OPTIONAL,
    scfID [8] ScfID { b2} OPTIONAL,
    extensions [10] Extensions {b1} OPTIONAL,
    carrier [11] Carrier {b2} OPTIONAL,
    serviceInteractionIndicators [26] ServiceInteractionIndicators { b2} OPTIONAL,
    callingPartyNumber [27] CallingPartyNumber { b2} OPTIONAL,
    callingPartysCategory [28] CallingPartysCategory OPTIONAL,
    redirectingPartyID [29] RedirectingPartyID { b2} OPTIONAL,
    redirectionInformation [30] RedirectionInformation OPTIONAL,
    displayInformation [12] DisplayInformation { b2} OPTIONAL,
    forwardCallIndicators [13] ForwardCallIndicators OPTIONAL,
    genericNumbers [14] GenericNumbers { b2} OPTIONAL,
    serviceInteractionIndicatorsTwo [15] ServiceInteractionIndicatorsTwo OPTIONAL,
    iNServiceCompatibilityResponse [16] INServiceCompatibilityResponse OPTIONAL,
    forwardGVNS [17] ForwardGVNS { b2} OPTIONAL,
    backwardGVNS [18] BackwardGVNS { b2} OPTIONAL,
}

```

```

chargeNumber          [19] ChargeNumber { b2}          OPTIONAL,
callSegmentID         [20] CallSegmentID {b2}         OPTIONAL,
legToBeCreated        [21] LegID                      OPTIONAL,
sDSSinformation       [22] SDSSinformation { b2}       OPTIONAL,
calledDirectoryNumber [23] CalledDirectoryNumber { b2} OPTIONAL,
bearerCapability      [24] BearerCapability {b2}      OPTIONAL,
cug-Interlock         [31] CUG-Interlock              OPTIONAL,
cug-OutgoingAccess    [32] NULL                      OPTIONAL,
suppressionOfAnnouncement [55] SuppressionOfAnnouncement OPTIONAL,
oCSIApplicable        [56] OCSIApplicable             OPTIONAL,
na-OliInfo            [57] NAOliInfo                  OPTIONAL,
...,
ipRelatedInformation  [33] IPRelatedInformation {b2}  OPTIONAL
}
-- na-OliInfo is included at the discretion of the gsmSCF operator.
-- OPTIONAL parameters are only provided if modifications desired to basic call
-- processing values
-- TAG 4, 9, 50 and 51 are reserved and shall not be used.

```

```

connectToResource {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT    ConnectToResourceArg { b1, b2}
  RETURN RESULT FALSE
  ERRORS      {missingParameter |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter |
               unknownLegID}
  ALWAYS RESPONDS FALSE
  CODE        opcode-connectToResource
}
-- Direction: SCF -> SSF, Timer: Tctr
-- This operation is used to connect a call from the SSP to the physical entity containing the SRF.

```

```

ConnectToResourceArg {B1: b1, B2: b2} ::= SEQUENCE {
  resourceAddress CHOICE {
    ipRoutingAddress      [0] IPRoutingAddress { b2},
    legID                 [1] LegID,
    ipAddressAndLegID     [2] SEQUENCE {
      ipRoutingAddress    [0] IPRoutingAddress {b2},
      legID               [1] LegID,
      ...
    },
    none                  [3] NULL,
    callSegmentID         [5] CallSegmentID { b2},
    ipAddressAndCallSegment [6] SEQUENCE {
      ipRoutingAddress    [0] IPRoutingAddress {b2},
      callSegmentID       [1] CallSegmentID { b2},
      ...
    },
    extensions            [4] Extensions {b1}          OPTIONAL,
    serviceInteractionIndicators [30] ServiceInteractionIndicators { b2} OPTIONAL,
    serviceInteractionIndicatorsTwo [7] ServiceInteractionIndicatorsTwo OPTIONAL,
    uSIServiceIndicator    [35] USIServiceIndicator { b2} OPTIONAL,
    uSIIInformation        [36] USIIInformation { b2}   OPTIONAL,
    ...
  },
  ...
}

```

```

continue OPERATION ::= {
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE        opcode-continue
}
-- Direction: SCF -> SSF, Timer: Tcue
-- This operation is used to request the SSF to proceed with call processing at the DP at which it
-- previously suspended call processing to await SCF instructions (i.e. proceed to the next point
-- in call in the BCSM). The SSF continues call processing without substituting new data from SCF.
-- This operation is not valid for a single call segment CSA with more than 2 legs or a
-- multi call segment CSA.

```

```

continueWithArgument {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT    ContinueWithArgumentArg { b1, b2}
  RETURN RESULT FALSE
  ERRORS      {missingParameter |
               parameterOutOfRange |

```

```

        unexpectedComponentSequence |
        unexpectedParameter |
        unexpectedDataValue |
        unknownLegID
    }

    ALWAYS RESPONDS FALSE
    CODE opcode-continueWithArgument
}

-- Direction: SCF -> SSF, Timer: T_cwa
-- This operation is used to request the SSF to proceed with call processing at the DP at
-- which it previously suspended call processing to await SCF instructions.
-- It is also used to provide additional service related information to a
-- User (Called Party or Calling Party) whilst the call processing proceeds.

ContinueWithArgumentArg {B1: b1, B2: b2} ::= SEQUENCE {
    legorCSID CHOICE{
        legID [0] LegID,
        csID [9] CallSegmentID {b2}} DEFAULT legID:sendingSideID:leg1,
    alertingPattern [1] AlertingPattern OPTIONAL,
    genericName [2] GenericName { b2} OPTIONAL,
    iNServiceCompatibilityResponse [3] INServiceCompatibilityResponse OPTIONAL,
    forwardGVNS [4] ForwardGVNS { b2} OPTIONAL,
    backwardGVNS [5] BackwardGVNS { b2} OPTIONAL,
    extensions [6] Extensions {b1} OPTIONAL,
    serviceInteractionIndicatorsTwo [7] ServiceInteractionIndicatorsTwo OPTIONAL,
    sDSSinformation [8] SDSSinformation { b2} OPTIONAL,
    iSDNAccessRelatedInformation [19] ISDNAccessRelatedInformation{b2} OPTIONAL,
    originalCalledPartyID [10] OriginalCalledPartyID {b2} OPTIONAL,
    callingPartyNumber [11] CallingPartyNumber {b2} OPTIONAL,
    callingPartysCategory [12] CallingPartysCategory OPTIONAL,
    redirectingPartyID [13] RedirectingPartyID {b2} OPTIONAL,
    redirectionInformation [14] RedirectionInformation OPTIONAL,
    forwardCallIndicators [15] ForwardCallIndicators OPTIONAL,
    genericNumbers [16] GenericNumbers {b2} OPTIONAL,
    cug-Interlock [17] CUG-Interlock OPTIONAL,
    cug-OutgoingAccess [18] NULL OPTIONAL,
    chargeNumber [50] ChargeNumber { b2} OPTIONAL,
    carrier [52] Carrier { b2} OPTIONAL,
    suppressionOfAnnouncement [55] SuppressionOfAnnouncement OPTIONAL,
    na-OliInfo [56] NAOliInfo OPTIONAL,
    ...,
    ipRelatedInformation [33] IPRelatedInformation {b2} OPTIONAL
}

-- OPTIONAL parameters are only provided if modifications desired to basic call processing values

-- Tag value 51 is reserved.

createCallSegmentAssociation {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT CreateCallSegmentAssociationArg { b1} OPTIONAL TRUE
    RESULT CreateCallSegmentAssociationResultArg { b1, b2}
    ERRORS {missingParameter |
        systemFailure |
        taskRefused |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter
    }
    CODE opcode-createCallSegmentAssociation
}

-- Direction SCF -> SSF, Timer T_Csa
-- This operation is used to create a new CSA. The new CSA will not contain any Call Segments
-- after creation.
-- The SSF is responsible for specifying a new CSA identifier for the created CSA which is
-- unique within the SSF.

CreateCallSegmentAssociationArg {B1: b1} ::= SEQUENCE {
    extensions [0] Extensions {b1} OPTIONAL,
    ...
}

CreateCallSegmentAssociationResultArg {B1:b1, B2: b2} ::= SEQUENCE {
    newCallSegmentAssociation [0] CSAID { b2},
    extensions [1] Extensions {b1} OPTIONAL,
    ...
}

createOrRemoveTriggerData {B1: b1, B2: b2, B3: b3} OPERATION ::= {

```

```

ARGUMENT    CreateOrRemoveTriggerDataArg {b1, b2, b3}
RESULT      CreateOrRemoveTriggerDataResultArg {b1, b2}
ERRORS      {missingParameter |
             missingCustomerRecord |
             parameterOutOfRange |
             systemFailure |
             taskRefused |
             unexpectedComponentSequence |
             unexpectedDataValue |
             unexpectedParameter
            }
CODE        opcode-createOrRemoveTriggerData
}
-- Direction: SCF -> SSF, Class 1, Timer: Tcrt
-- This trigger management operation is used by the SCF outside the context of a call
-- to create a new trigger detection point in the CCF/SSF by downloading trigger data
-- (e.g. triggering criteria, ServiceKey, SCF address,...)
-- or to remove an existing trigger.

CreateOrRemoveTriggerDataArg {B1: b1, B2: b2, B3: b3} ::= SEQUENCE {
  createOrRemove          [0] CreateOrRemoveIndicator DEFAULT create,
  dPName                  [1] EventTypeBCSM                                OPTIONAL,
  triggerDPType           [2] TriggerDPType    DEFAULT tdp-r,
  serviceKey              [3] ServiceKey                                OPTIONAL,
  profile                  [4] ProfileIdentifier{b2}                    OPTIONAL,
  triggerData              [5] TriggerData                                OPTIONAL,
  defaultFaultHandling    [6] DefaultFaultHandling {b1, b2, b3}      OPTIONAL,
  tDPIdentifier            [7] TDPIIdentifier {b2}                      OPTIONAL,
  ...,
  ...,
  extensions               [30] Extensions {b1}                        OPTIONAL
}

CreateOrRemoveTriggerDataResultArg {B1: b1, B2: b2} ::= SEQUENCE {
  triggerStatus            [0] TriggerStatus,
  tDPIdentifier            [1] TDPIIdentifier {b2},
  registratorIdentifier    [2] RegistratorIdentifier                    OPTIONAL,
  ...,
  ...,
  extensions               [30] Extensions {b1}                        OPTIONAL
}

disconnectForwardConnection OPERATION ::= {
  RETURN RESULT FALSE
  ERRORS {systemFailure |
         taskRefused |
         unexpectedComponentSequence }
  ALWAYS RESPONDS FALSE
  CODE   opcode-disconnectForwardConnection
}
-- Direction: SCF -> SSF, Timer: Tdfc
-- This operation is used to disconnect a forward temporary connection or a
-- connection to a resource.

disconnectForwardConnectionWithArgument {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT    DisconnectForwardConnectionWithArgumentArg {b1, b2}
  RETURN RESULT FALSE
  ERRORS      {missingParameter |
             systemFailure |
             taskRefused |
             unexpectedComponentSequence |
             unexpectedDataValue |
             unexpectedParameter |
             unknownLegID}
  ALWAYS RESPONDS FALSE
  CODE        opcode-dFCWithArgument
}

-- Direction: SCF -> SSF, Timer: Tdfcwa
-- This operation is used to disconnect a forward temporary connection or a
-- connection to a resource.

DisconnectForwardConnectionWithArgumentArg {B1: b1, B2: b2} ::= SEQUENCE {
  partyToDisconnect CHOICE {
    legID              [0] LegID,
    callSegmentID      [1] CallSegmentID { b2}
  },

```



```

    extensions                [2] Extensions {b1}                OPTIONAL,
    uSIServiceIndicator       [3] USIServiceIndicator { b2}     OPTIONAL,
    uSIInformation            [4] USIInformation { b2}           OPTIONAL,
    ...
  }

disconnectLeg {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT    DisconnectLegArg { b1, b2}
  RETURN RESULT TRUE
  ERRORS      {missingParameter|
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter|
               unknownLegID}
  CODE        opcode-disconnectLeg
}
-- Direction: SCF -> SSF, Timer: T d1
-- This operation is issued by the SCF to release a specific leg associated with the call
-- and retain any other legs not specified in the DisconnectLeg. Any leg may be disconnected,
-- including the controlling leg, without completely releasing all legs.

DisconnectLegArg {B1: b1, B2: b2} ::= SEQUENCE {
  legToBeReleased    [0] LegID,
  releaseCause       [1] Cause { b2}                OPTIONAL,
  extensions         [2] Extensions {b1}            OPTIONAL,
  ...
}

entityReleased {B2: b2} OPERATION ::= {
  ARGUMENT    EntityReleasedArg { b2}
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE        opcode-entityReleased
}
-- Direction SSF -> SCF, Timer: Ter
-- This operation is used by SSF to inform the SCF of an error/exception

EntityReleasedArg {B2: b2} ::= CHOICE {
  csFailure [0] SEQUENCE{
    callSegmentID [0] CallSegmentID { b2},
    reason        [1] Reason { b2}                OPTIONAL,
    cause         [2] Cause { b2}                OPTIONAL,
    ...
  },
  bCSMFailure [1] SEQUENCE{
    legID        [0] LegID,
    reason       [1] Reason { b2}                OPTIONAL,
    cause        [2] Cause { b2}                OPTIONAL,
    ...
  }
}

establishTemporaryConnection {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT    EstablishTemporaryConnectionArg { b1, b2}
  RETURN RESULT FALSE
  ERRORS      {eTCFailed |
               missingParameter |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter|
               unknownLegID}
  ALWAYS RESPONDS FALSE
  CODE        opcode-establishTemporaryConnection
}
-- Direction: SCF -> SSF, Timer: Tetc
-- This operation is used to create a connection to a resource for a limited period of time
-- (e.g. to play an announcement, to collect user information); it implies the use of the assist
-- procedure.

```

```

EstablishTemporaryConnectionArg {B1: b1, B2: b2} ::= SEQUENCE {
  assistingSSPIPRoutingAddress [0] AssistingSSPIPRoutingAddress { b2},
  correlationID [1] CorrelationID { b2} OPTIONAL,
  partyToConnect CHOICE {
    legID [2] LegID,
    callSegmentID [7] CallSegmentID { b2}
  } OPTIONAL,
  scfID [3] ScfID { b2} OPTIONAL,
  extensions [4] Extensions {b1} OPTIONAL,
  carrier [5] Carrier {b2} OPTIONAL,
  serviceInteractionIndicators [30] ServiceInteractionIndicators { b2} OPTIONAL,
  serviceInteractionIndicatorsTwo [6] ServiceInteractionIndicatorsTwo OPTIONAL,
  na-OliInfo [50] NAOliInfo OPTIONAL,
  chargeNumber [51] ChargeNumber {b2} OPTIONAL,
  ...
}
-- OPTIONAL parameters are only provided if modifications desired to basic call processing values

eventNotificationCharging {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT EventNotificationChargingArg { b1, b2}
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE opcode-eventNotificationCharging
}
-- Direction: SSF -> SCF, Timer: Tenc
-- This operation is used by the SSF to report to the SCF the occurrence of a specific
-- charging event
-- type as previously requested by the SCF in a
-- RequestNotificationChargingEvent operation.

EventNotificationChargingArg {B1: b1, B2: b2} ::= SEQUENCE {
  eventTypeCharging [0] EventTypeCharging { b2} OPTIONAL,
  eventSpecificInformationCharging [1] EventSpecificInformationCharging { b2} OPTIONAL,
  legID [2] LegID OPTIONAL,
  extensions [3] Extensions {b1} OPTIONAL,
  monitorMode [30] MonitorMode DEFAULT notifyAndContinue,
  eventTypeTariff [50] EventTypeTariff OPTIONAL,
  eventSpecificInformationTariff [51] ChargingMessageType OPTIONAL,
  ...
}
-- OPTIONAL denotes network operator specific use.

eventReportBCSM {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT EventReportBCSMArg { b1, b2}
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE opcode-eventReportBCSM
}
-- Direction: SSF -> SCF, Timer: Terb
-- This operation is used to notify the SCF of a call-related event (e.g. BCSM events such as
-- busy or no answer) previously requested by the SCF in a RequestReportBCSMEvent operation.

EventReportBCSMArg {B1: b1, B2: b2} ::= SEQUENCE {
  eventTypeBCSM [0] EventTypeBCSM,
  eventSpecificInformationBCSM [2] EventSpecificInformationBCSM { b2} OPTIONAL,
  legID [3] LegID OPTIONAL,
  miscCallInfo [4] MiscCallInfo DEFAULT {messageType request},
  extensions [5] Extensions {b1} OPTIONAL,
  ...
}

furnishChargingInformation {B1:b1, B2: b2} OPERATION ::= {
  ARGUMENT FurnishChargingInformationArg { b1, b2}
  RETURN RESULT FALSE
  ERRORS {missingParameter |
    taskRefused |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter}
  ALWAYS RESPONDS FALSE
  CODE opcode-furnishChargingInformation
}
-- Direction: SCF -> SSF, Timer: Tfci
-- This operation is used to request the SSF to generate, register a call record or to
-- include some information in the default call record. The registered call record is intended
-- for off line charging of the call.

```

```

FurnishChargingInformationArg {B1: b1, B2: b2} ::= FCIBillingChargingCharacteristics {b1, b2}

initialDP {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT    InitialDPArg { b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingCustomerRecord |
               missingParameter |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter
              }
  ALWAYS RESPONDS FALSE
  CODE      opcode-initialDP
}

-- Direction: SSF -> SCF, Timer: Tidp
-- This operation is used after a TDP to indicate request for service.

InitialDPArg {B1: b1, B2: b2} ::= SEQUENCE {
  serviceKey                [0] ServiceKey,
  calledPartyNumber         [2] CalledPartyNumber { b2}           OPTIONAL,
  callingPartyNumber        [3] CallingPartyNumber { b2}         OPTIONAL,
  callingPartyBusinessGroupID [4] CallingPartyBusinessGroupID   OPTIONAL,
  callingPartysCategory     [5] CallingPartysCategory           OPTIONAL,
  cGEncountered            [7] CGEncountered                   OPTIONAL,
  iPSSPCapabilities        [8] IPSSPCapabilities { b2}          OPTIONAL,
  iPAavailable              [9] IPAavailable { b2}              OPTIONAL,
  locationNumber           [10] LocationNumber { b2}            OPTIONAL,
  originalCalledPartyID    [12] OriginalCalledPartyID {b2}      OPTIONAL,
  terminalType             [14] TerminalType                   OPTIONAL,
  extensions               [15] Extensions {b1}               OPTIONAL,
  highLayerCompatibility   [23] HighLayerCompatibility         OPTIONAL,
  serviceInteractionIndicators [24] ServiceInteractionIndicators { b2} OPTIONAL,
  additionalCallingPartyNumber [25] AdditionalCallingPartyNumber { b2} OPTIONAL,
  forwardCallIndicators    [26] ForwardCallIndicators          OPTIONAL,
  bearerCapability         [27] BearerCapability { b2}         OPTIONAL,
  eventTypeBCSM           [28] EventTypeBCSM                  OPTIONAL,
  redirectingPartyID      [29] RedirectingPartyID { b2}        OPTIONAL,
  redirectionInformation   [30] RedirectionInformation         OPTIONAL,
  cause                   [17] Cause { b2}                   OPTIONAL,
  iSDNAccessRelatedInformation [21] ISDNAccessRelatedInformation { b2} OPTIONAL,
  iNServiceCompatibilityIndication [22] INServiceCompatibilityIndication { b2} OPTIONAL,

  genericNumbers          [31] GenericNumbers { b2}           OPTIONAL,
  serviceInteractionIndicatorsTwo [32] ServiceInteractionIndicatorsTwo OPTIONAL,
  forwardGVNS            [33] ForwardGVNS { b2}             OPTIONAL,
  createdCallSegmentAssociation [34] CSAID { b2}            OPTIONAL,
  uSIServiceIndicator    [35] USIServiceIndicator { b2}     OPTIONAL,
  uSIInformation         [36] USIInformation { b2}          OPTIONAL,
  carrier                [37] Carrier {b2}                 OPTIONAL,
  cCSS                  [38] CCSS                           OPTIONAL,
  vPNIndicator          [39] VPNIndicator                   OPTIONAL,
  cNInfo               [40] CNInfo { b2}                   OPTIONAL,
  callReference         [41] CallReference{ b2}            OPTIONAL,
  routeingNumber       [42] RouteingNumber { b2}          OPTIONAL,
  callingGeodeticLocation [43] CallingGeodeticLocation { b2} OPTIONAL,
  globalCallReference  [44] GlobalCallReference {b2}      OPTIONAL,
  cug-Index            [45] CUG-Index                     OPTIONAL,
  cug-Interlock        [46] CUG-Interlock                 OPTIONAL,
  cug-OutgoingAccess   [47] NULL                          OPTIONAL,
  iMSI                 [50] IMSI                          OPTIONAL,
  subscriberState      [51] SubscriberState               OPTIONAL,
  locationInformation  [52] LocationInformation           OPTIONAL,
  ext-basicServiceCode [53] Ext-BasicServiceCode         OPTIONAL,
  callReferenceNumber  [54] CallReferenceNumber          OPTIONAL,
  mscAddress           [55] ISDN-AddressString           OPTIONAL,
  calledPartyBCDNumber [56] CalledPartyBCDNumber {b2}     OPTIONAL,
  timeAndTimezone     [57] TimeAndTimezone {b2}         OPTIONAL,
  gsm-ForwardingPending [58] NULL                        OPTIONAL,
  initialDPArgExtension [59] InitialDPArgExtension       OPTIONAL,
  ...,
  ipRelatedInformation [33] IPRelatedInformation {b2}     OPTIONAL
}

```

```

-- for mscAddress for encoding see TS 129 002
-- OPTIONAL for iPSSPCapabilities, iPAvailable, cGEncountered, and miscCallInfo denotes network
-- operator specific use.
-- OPTIONAL for terminalType indicates that this parameter applies only at originating
-- or terminating
-- local exchanges if the SSF has this information.
-- Tag values 1, 6, 11, 13, 16, 18, 19, 20 are reserved and shall not be used

initiateCallAttempt {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    InitiateCallAttemptArg { b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingParameter |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter
              }
  ALWAYS RESPONDS FALSE

  CODE      opcode-initiateCallAttempt
}

-- Direction: SCF -> SSF, Timer: Tica
-- This operation is used to request the SSF to create a new call to one call party using address
-- information provided by the SCF.
InitiateCallAttemptArg {B1: b1, B2: b2} ::= SEQUENCE {
  destinationRoutingAddress [0] DestinationRoutingAddress { b2},
  alertingPattern [1] AlertingPattern OPTIONAL,
  iSDNAccessRelatedInformation [2] ISDNAccessRelatedInformation {b2} OPTIONAL,
  extensions [4] Extensions {b1} OPTIONAL,
  serviceInteractionIndicators [29] ServiceInteractionIndicators { b2} OPTIONAL,
  callingPartyNumber [30] CallingPartyNumber { b2} OPTIONAL,
  legToBeCreated [5] LegID DEFAULT sendingSideID:leg1,
  newCallSegment [6] CallSegmentID { b2} DEFAULT initialCallSegment,
  iNServiceCompatibilityResponse [7] INServiceCompatibilityResponse OPTIONAL,
  serviceInteractionIndicatorsTwo [8] ServiceInteractionIndicatorsTwo OPTIONAL,
  carrier [9] Carrier {b2} OPTIONAL,
  correlationID [10] CorrelationID { b2} OPTIONAL,
  scfID [11] ScfID { b2} OPTIONAL,
  callReference [12] CallReference { b2} OPTIONAL,
  calledDirectoryNumber [13] CalledDirectoryNumber { b2} OPTIONAL,
  originalCalledPartyID [14] OriginalCalledPartyID {b2} OPTIONAL,
  callingPartysCategory [15] CallingPartysCategory OPTIONAL,
  redirectingPartyID [16] RedirectingPartyID {b2} OPTIONAL,
  redirectionInformation [17] RedirectionInformation OPTIONAL,
  displayInformation [18] DisplayInformation {b2} OPTIONAL,
  forwardCallIndicators [19] ForwardCallIndicators OPTIONAL,
  genericNumbers [20] GenericNumbers {b2} OPTIONAL,
  forwardGVNS [21] ForwardGVNS {b2} OPTIONAL,
  bearerCapability [60] BearerCapability { b2} OPTIONAL,
  globalCallReference [23] GlobalCallReference { b2} OPTIONAL,
  cug-Interlock [24] CUG-Interlock OPTIONAL,
  cug-OutgoingAccess [25] NULL OPTIONAL,
  ...,
  incomingSignallingBufferCopy [26] BOOLEAN DEFAULT FALSE,
  ipRelatedInformation [27] IPRelatedInformation {b2} OPTIONAL
}

-- Tag values 3, 50, 51, 52 are reserved and shall not be used
-- OPTIONAL parameters are only provided if modifications desired to basic call processing values

manageTriggerData {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    ManageTriggerDataArg { b1, b2}
  RESULT      ManageTriggerDataResultArg { b1, b2}
  ERRORS      {missingParameter |
               missingCustomerRecord |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter
              }
  CODE      opcode-manageTriggerData
}

```

```
-- Direction: SCF -> SSF, Class 1, Timer: Tmtd
-- This trigger management operation is used outside the context of a call to activate,
-- deactivate or retrieve
-- the status of one or several trigger detection point linked to a subscriber profile known at
-- the switch, e.g. related to an access line ( i.e. an individual trigger).
```

```
ManageTriggerDataArg {B1: b1, B2: b2} ::= SEQUENCE {
  actionIndicator          [0] ActionIndicator,
  triggerDataIdentifier   CHOICE {
    profileAndDP          [1] TriggerDataIdentifier {b1, b2},
    -- one trigger
    profile                [5] ProfileIdentifier {b2}
  },
  registratorIdentifier   [2] RegistratorIdentifier          OPTIONAL,
  extensions              [3] Extensions {b1}                OPTIONAL,
  tDPIdentifier           [4] TDPIIdentifier {b2}             OPTIONAL,
  ...
}
```

```
ManageTriggerDataResultArg {B1: b1, B2: b2} ::= CHOICE
{
  oneTriggerResult        SEQUENCE {
    actionPerformed       [0] ActionPerformed,
    extensions            [1] Extensions {b1}          OPTIONAL,
    ... },
  severalTriggerResult    [1] SEQUENCE {
    results                [0] TriggerResults {b2},
    extensions             [1] Extensions {b1}          OPTIONAL,
    ... }
}
```

```
mergeCallSegments {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT MergeCallSegmentsArg { b1, b2}
  RETURN RESULT TRUE
  ERRORS {missingParameter |
    systemFailure |
    taskRefused |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter
  }
  CODE opcode-mergeCallSegments
}
```

```
-- Direction: SCF -> SSF. Timer: Tmc
```

```
-- This operation is issued by the SCF to merge two associated CSs, into one CS.
```

```
MergeCallSegmentsArg {B1: b1, B2: b2} ::= SEQUENCE {
  sourceCallSegment       [0] CallSegmentID {b2},
  targetCallSegment       [1] CallSegmentID {b2} DEFAULT initialCallSegment,
  extensions               [2] Extensions {b1}          OPTIONAL,
  ...,
  mergeSignallingPaths    [3] NULL OPTIONAL
}
```

```
moveCallSegments {B1: b1, B2: b2} OPERATION ::= {
  ARGUMENT MoveCallSegmentsArg { b1, b2}
  RETURN RESULT TRUE
  ERRORS {missingParameter |
    systemFailure |
    taskRefused |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter |
    unknownLegID
  }
  CODE opcode-moveCallSegments
}
```

```
-- Direction: SCF -> SSF, Timer Tmcs
```

```
-- This operation moves a CS from the source CSA to the target CSA
```

```
MoveCallSegmentsArg {B1: b1, B2: b2} ::= SEQUENCE {
  targetCallSegmentAssociation [0] CSAID { b2},
  callSegments                 [1] SEQUENCE SIZE (1..b2.&numOfCSs)
  OF SEQUENCE {
    sourceCallSegment          [0] CallSegmentID { b2}    DEFAULT initialCallSegment,
```

```

        newCallSegment          [1] CallSegmentID { b2},
        ...
    },
    legs                        [2] SEQUENCE SIZE (1..b2.&numOfLegs)
OF SEQUENCE {
    sourceLeg                   [0] LegID,
    newLeg                      [1] LegID,
    ...
},
    extensions                  [3] Extensions {b1}          OPTIONAL,
    ...
}

```

```

moveLeg {B1: b1, B2: b2} OPERATION::= {
    ARGUMENT    MoveLegArg { b1, b2}
    RETURN RESULT TRUE
    ERRORS     {missingParameter |
                systemFailure |
                taskRefused |
                unexpectedComponentSequence |
                unexpectedDataValue |
                unexpectedParameter |
                unknownLegID
                }
    CODE       opcode-moveLeg
}

```

-- Direction: SCF ->SSF, Timer: T<sub>m1</sub>

-- This operation is issued by the SCF to move a leg from one CS to another with which it is associated.

```

MoveLegArg {B1: b1, B2: b2} ::= SEQUENCE {
    legIDToMove          [0] LegID,
    targetCallSegment    [1] CallSegmentID { b2} DEFAULT 1,
    extensions           [2] Extensions {b1}          OPTIONAL,
    ...,
    detachSignallingPath [3] NULL OPTIONAL, -- action on source
    exportSignallingPath [4] NULL OPTIONAL -- action on target
}

```

```

releaseCall { B2: b2} OPERATION::= {
    ARGUMENT    ReleaseCallArg { b2}
    RETURN RESULT FALSE
    ALWAYS RESPONDS FALSE
    CODE       opcode-releaseCall
}

```

-- Direction: SCF -> SSF, Timer: T<sub>rc</sub>

-- This operation is used by the SCF to tear down an existing call segment at any phase of the call for all parties involved in the call segment or to tear down all existing call segments within a Call Segment Association.

```

ReleaseCallArg {B2: b2} ::= CHOICE {
    initialCallSegment          Cause { b2},
    callSegmentToRelease        [1] SEQUENCE {
        callSegment              [0] INTEGER (1..b2.&numOfCSs),
        releaseCause             [1] Cause { b2}          OPTIONAL,
        forcedRelease            [2] BOOLEAN DEFAULT FALSE,
        ...
    },
    allCallSegments            [2] SEQUENCE {
        releaseCause             [0] Cause { b2}          OPTIONAL,
        timeToRelease            [1] TimerValue           OPTIONAL,
        forcedRelease            [2] BOOLEAN DEFAULT FALSE,
        ...
    },
    ...
}

```

-- A default cause value of decimal 31 (normal unspecified) should be coded appropriately.

-- If timeToRelease parameter is omitted, the default shall be no timed disconnect requested

-- If forcedRelease parameter is omitted (default value "FALSE") the default shall be no

-- forced release requested.

```

reportUTSI {B1: b1, B2: b2} OPERATION::= {
    ARGUMENT    ReportUTSIArg { b1, b2}
    RETURN RESULT FALSE
    ALWAYS RESPONDS FALSE
    CODE       opcode-reportUTSI
}

```

```

}
-- Direction: SSF -> SCF. Timer: Tru
-- This operation is issued by the SSF in the context of the USI feature.
-- It is used to report the receipt of a User to Service Information (USI) to the SCF.

```

```

ReportUTSIArg {B1: b1, B2: b2} ::= SEQUENCE {
    uSIServiceIndicator [0] USIServiceIndicator { b2},
    legID                [1] LegID DEFAULT receivingSideID:leg1,
    uSIInformation       [2] USIInformation { b2},
    extensions           [3] Extensions {b1} OPTIONAL,
    ...
}

```

```

requestCurrentStatusReport {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT RequestCurrentStatusReportArg { b2}
    RESULT RequestCurrentStatusReportResultArg { b1, b2}
    ERRORS {missingParameter |
            parameterOutOfRange |
            systemFailure |
            taskRefused |
            unexpectedComponentSequence |
            unexpectedDataValue |
            unexpectedParameter |
            unknownResource
           }
    CODE opcode-requestCurrentStatusReport
}

```

```

-- Direction: SCF -> SSF, Timer: TrCS
-- This operation is used to request the SSF to report immediately the busy/idle status
-- of a physical termination resource.

```

```

RequestCurrentStatusReportArg {B2: b2} ::= ResourceID { b2}

```

```

RequestCurrentStatusReportResultArg {B1: b1, B2: b2} ::= SEQUENCE {
    resourceStatus [0] ResourceStatus,
    resourceID     [1] ResourceID { b2} OPTIONAL,
    extensions     [2] Extensions {b1} OPTIONAL,
    ...
}

```

```

requestEveryStatusChangeReport {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT RequestEveryStatusChangeReportArg { b1, b2}
    RETURN RESULT TRUE
    ERRORS {missingParameter |
            parameterOutOfRange |
            systemFailure |
            taskRefused |
            unexpectedComponentSequence |
            unexpectedDataValue |
            unexpectedParameter |
            unknownResource
           }
    CODE opcode-requestEveryStatusChangeReport
}

```

```

-- Direction: SCF -> SSF, Timer: Tres
-- This operation is used to request the SSF to report every change of busy/idle status
-- of a physical termination resource.

```

```

RequestEveryStatusChangeReportArg {B1: b1, B2: b2} ::= SEQUENCE {
    resourceID [0] ResourceID { b2},
    correlationID [1] CorrelationID { b2} OPTIONAL,
    monitorDuration [2] Duration OPTIONAL,
    extensions [3] Extensions {b1} OPTIONAL,
    ...
}

```

```

-- For correlationID OPTIONAL denotes network operator optional.
-- monitorDuration is required if outside the context of a call.
-- It is not expected if we are in the context of a call, because in that case
-- the end of the call implicitly means the end of the monitoring.

```

```

requestFirstStatusMatchReport {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT RequestFirstStatusMatchReportArg { b1, b2}
    RETURN RESULT TRUE
    ERRORS {missingParameter |
            parameterOutOfRange |
            systemFailure |
            taskRefused |
            unexpectedComponentSequence |

```

```

        unexpectedDataValue |
        unexpectedParameter |
        unknownResource
    }
    CODE opcode-requestFirstStatusMatchReport
}
-- Direction: SCF -> SSF, Timer: Trfs
-- This operation is used to request the SSF to report the first change busy/idle to
-- the specified status of a physical termination resource.

RequestFirstStatusMatchReportArg {B1: b1, B2: b2} ::= SEQUENCE {
    resourceID          [0] ResourceID { b2}          OPTIONAL,
    resourceStatus      [1] ResourceStatus OPTIONAL,
    correlationID       [2] CorrelationID { b2}        OPTIONAL,
    monitorDuration     [3] Duration                  OPTIONAL,
    extensions          [4] Extensions {b1}           OPTIONAL,
    bearerCapability    [5] BearerCapability { b2}     OPTIONAL,
    ...
}
-- For correlationID OPTIONAL denotes network operator optional.
-- monitorDuration is required if outside the context of a call. It is not expected if we are in
-- the context of a call, because in that case the end of the call implicitly means the end of
-- the monitoring.

requestNotificationChargingEvent {B2: b2} OPERATION ::= {
    ARGUMENT RequestNotificationChargingEventArg { b2}
    RETURN RESULT FALSE
    ERRORS {missingParameter |
        parameterOutOfRange |
        systemFailure |
        taskRefused |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter |
        unknownLegID
    }
    ALWAYS RESPONDS FALSE
    CODE opcode-requestNotificationChargingEvent
}
-- Direction: SCF -> SSF, Timer: Trnc
-- This operation is used by the SCF to instruct the SSF on how to manage the charging events
-- which are received from other FE's and not under control of the service logic instance.

RequestNotificationChargingEventArg {B2: b2} ::= SEQUENCE SIZE(1..b2.&numOfChargingEvents) OF
    ChargingEvent {b2}

requestReportBCSMEvent {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT RequestReportBCSMEventArg { b1, b2}
    RETURN RESULT FALSE
    ERRORS {missingParameter |
        parameterOutOfRange |
        systemFailure |
        taskRefused |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter |
        unknownLegID
    }
    ALWAYS RESPONDS FALSE

    CODE opcode-requestReportBCSMEvent
}
-- Direction: SCF -> SSF, Timer: Trrb

RequestReportBCSMEventArg {B1: b1, B2: b2} ::= SEQUENCE {
    bcsmEvents          [0] SEQUENCE SIZE(1..b2.&numOfBCSMEvents) OF
        BCSMEvent {b2},
    extensions          [2] Extensions {b1} OPTIONAL,
    ...
}
-- Indicates the BCSM related events for notification.

```



```

requestReportUTSI {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT RequestReportUTSIArg { b1, b2}
  RETURN RESULT FALSE
  ERRORS {
    missingParameter |
    parameterOutOfRange |
    systemFailure |
    taskRefused |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter |
    unknownLegID
  }
  ALWAYS RESPONDS FALSE
  CODE opcode-requestReportUTSI
}

-- Direction: SCF -> SSF. Timer: Trru
-- This operation is issued by the SCF in the context of the USI feature to request the
-- SSF to monitor for
-- a User to Service Information (UTSI) information element, which are received from a user.

RequestReportUTSIArg {B1: b1, B2: b2} ::= SEQUENCE {
  requestedUTSIList [0] RequestedUTSIList { b2},
  extensions [1] Extensions {b1} OPTIONAL,
  legID [2] LegID DEFAULT sendingSideID:leg1 ,
  ...
}

resetTimer {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT ResetTimerArg { b1, b2}
  RETURN RESULT FALSE
  ERRORS {missingParameter |
    parameterOutOfRange |
    taskRefused |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter
  }
  ALWAYS RESPONDS FALSE
  CODE opcode-resetTimer
}

-- Direction: SCF -> SSF, Timer: Trt
-- This operation is used to request the SSF to refresh an application timer in the SSF.

ResetTimerArg {B1: b1, B2: b2} ::= SEQUENCE {
  timerID [0] TimerID DEFAULT tssf,
  timervalue [1] TimerValue,
  extensions [2] Extensions {b1} OPTIONAL,
  callSegmentID [3] CallSegmentID { b2} OPTIONAL,
  ...
}

selectFacility {B1: b1, B2: b2 } OPERATION::= {
  ARGUMENT SelectFacilityArg { b1, b2 }
  RETURN RESULT FALSE
  ERRORS {missingParameter |
    parameterOutOfRange |
    systemFailure |
    taskRefused |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter
  }
  ALWAYS RESPONDS FALSE
  CODE opcode-selectFacility
}

-- Direction: SCF -> SSF, Timer: Tsf
-- This operation is used to request the SSF to perform the terminating basic call processing
-- actions to select the terminating line if it is idle,. If no idle line, the SSF
-- determines that the terminating facility is busy.

SelectFacilityArg {B1: b1, B2: b2} ::= SEQUENCE {
  alertingPattern [0] AlertingPattern OPTIONAL,
  extensions [6] Extensions {b1} OPTIONAL,
  serviceInteractionIndicatorsTwo [12] ServiceInteractionIndicatorsTwo OPTIONAL,
  legID [16] LegID OPTIONAL,
}

```

```

    ....
    ipRelatedInformation          [17] IPRelatedInformation {b2} OPTIONAL
  }
-- OPTIONAL parameters are only provided if modifications desired to basic call processing values.

sendChargingInformation {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    SendChargingInformationArg { b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingParameter |
               unexpectedComponentSequence |
               unexpectedParameter |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unexpectedDataValue |
               unknownLegID
              }
  ALWAYS RESPONDS FALSE
  CODE        opcode-sendChargingInformation
}
-- Direction: SCF -> SSF, Timer: Tsci
-- This operation is used to instruct the SSF on the charging information to send by the SSF.
-- The charging information can either be sent back by means of signalling or internal
-- if the SSF is located in the local exchange. In the local exchange
-- this information may be used to update the charge meter or to create a standard call record.

SendChargingInformationArg {B1: b1, B2: b2} ::= SEQUENCE {
  sCIBillingChargingCharacteristics [0] SCIBillingChargingCharacteristics {b2} OPTIONAL,
  partyToCharge                      [1] LegID,
  extensions                          [2] Extensions {b1}
  OPTIONAL,
  tariffMessage                      [50] ChargingMessageType OPTIONAL,
  chargingControl                    [51] ChargingControlType DEFAULT
  {tariffInfoFromSuccExchangeSCI { }},
  ...
}

sendSTUI {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    SendSTUIArg { b1, b2}
  RETURN RESULT  FALSE
  ERRORS      {missingParameter |
               parameterOutOfRange |
               unexpectedComponentSequence |
               unexpectedParameter |
               unexpectedDataValue |
               systemFailure |
               taskRefused |
               unknownLegID
              }
  ALWAYS RESPONDS FALSE
  CODE        opcode-sendSTUI
}
-- Direction: SCF -> SSF. Timer: Tss
-- This operation is issued by the SCF in the context of the USI feature.
-- It is used to request the SSF to send a Service to User Information (USI information)
-- data element to the indicated user.

SendSTUIArg {B1: b1, B2: b2} ::= SEQUENCE {
  uSIServiceIndicator [0] USIServiceIndicator { b2},
  legID               [1] LegID DEFAULT sendingSideID:leg1,
  uSIInformation      [2] USIInformation { b2},
  extensions          [3] Extensions {b1} OPTIONAL,
  ...
}

serviceFilteringResponse {B1: b1, B2: b2} OPERATION::= {
  ARGUMENT    ServiceFilteringResponseArg { b1, b2}
  RETURN RESULT  FALSE
  ALWAYS RESPONDS FALSE
  CODE        opcode-serviceFilteringResponse
}
-- Direction: SSF -> SCF, Timer: Tsfr
-- This operation is used to send back to the SCF the values of counters specified in a previous
-- ActivateServiceFiltering operation

```

```

ServiceFilteringResponseArg {B1: b1, B2: b2} ::= SEQUENCE {
    countersValue      [0] CountersValue,
    filteringCriteria  [1] FilteringCriteria { b2},
    extensions         [2] Extensions {b1}           OPTIONAL,
    responseCondition [3] ResponseCondition         OPTIONAL,
    ...
}

setServiceProfile {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT      SetServiceProfileArg {b1,b2}
    RETURN RESULT FALSE
    ERRORS       {missingParameter |
                  parameterOutOfRange |
                  taskRefused |
                  unexpectedComponentSequence |
                  unexpectedDataValue |
                  unexpectedParameter }
    ALWAYS RESPONDS FALSE
    CODE         opcode-setServiceProfile
}
-- Direction SCF -> SSF, Timer Tsep
-- This operation is used within the context of a call to request the SSF to
-- activate/de-activate a list of trigger for one of the parties in the call.

SetServiceProfileArg {B1: b1, B2: b2} ::= SEQUENCE {
    inProfiles [0] SEQUENCE SIZE (1..b2.&numOfINProfile)
                  OF INProfile {b1, b2},
    ...,
    ...,
    extensions [30] Extensions {b1}           OPTIONAL
}

splitLeg {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT      SplitLegArg { b1, b2}
    RETURN RESULT TRUE
    ERRORS       {missingParameter |
                  unexpectedComponentSequence |
                  unexpectedParameter |
                  unexpectedDataValue |
                  systemFailure |
                  taskRefused |
                  unknownLegID
                  }
    CODE         opcode-splitLeg
}
-- Direction: SCF -> SSF. Timer: TS1
-- This operation is issued by the SCF to separate one joined leg from a multi-way connection
-- or a single 2 party Call segment.

SplitLegArg {B1: b1, B2: b2} ::= SEQUENCE {
    legToBeSplit      [0] LegID,
    newCallSegment    [1] INTEGER (2..b2.&numOfCSs),
    extensions         [2] Extensions {b1}           OPTIONAL,
    ...,
    detachSignallingPath [3] NULL OPTIONAL
}

statusReport {B1: b1, B2: b2} OPERATION ::= {
    ARGUMENT      StatusReportArg { b1, b2}
    RETURN RESULT FALSE
    ALWAYS RESPONDS FALSE
    CODE         opcode-statusReport
}
-- Direction: SSF -> SCF, Timer: TSrp
-- This operation is used as a response to RequestFirstStatusMatchReport or
-- RequestEveryStatusChangeReport operations.

```

```

StatusReportArg {B1: b1, B2: b2} ::= SEQUENCE {
    resourceStatus      [0] ResourceStatus      OPTIONAL,
    correlationID       [1] CorrelationID { b2}  OPTIONAL,
    resourceID          [2] ResourceID { b2}    OPTIONAL,
    extensions          [3] Extensions {b1}     OPTIONAL,
    reportCondition     [4] ReportCondition     OPTIONAL,
    ...
}
-- For correlationID, OPTIONAL denotes network operator optional.
-- resourceID is required when the SSF sends a report as an answer to a previous request when the
-- correlationID was present.

END

```

## 6.4 Operation timers

The following value ranges do apply for operation specific timers in INAP:

short: 1 - 10 seconds

medium: 1 - 60 seconds

long: 1 second - 30 minutes

Table 1 lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network specific and has to be defined by the network operator.

**Table 1: Operation timers and their value range**

Operation Name	Timer	Value range
ActivateServiceFiltering	Tasf	medium
ActivityTest	Tat	short
ApplyCharging	Tac	short
ApplyChargingReport	Tacr	short
AssistRequestInstructions	Tari	short
CallFiltering	Tcf	Short
CallGap	Tcg	short
CallInformationReport	Tcirp	short
CallInformationRequest	Tcirq	short
Cancel	Tcan	short
CancelStatusReportRequest	Tcsr	short
CollectInformation	Tci	medium
Connect	Tcon	short
ConnectToResource	Tctr	short
Continue	Tcue	short
ContinueWithArgument	Tcwa	short
CreateCallSegmentAssociation	Tcsa	short
CreateOrRemoveTriggerData	Tcrt	medium
DisconnectForwardConnection	Tdfc	short
DisconnectForwardConnectionWithArgument	Tdfcwa	short
DisconnectLeg	Tdl	short
EntityRelease	Ter	short
EstablishTemporaryConnection	Tetc	medium
EventNotificationCharging	Tenc	short
EventReportBCSM	Terb	short
FurnishChargingInformation	Tfci	short
InitialDP	Tidp	short
InitiateCallAttempt	Tica	short
ManageTriggerData	Tmtd	medium
MergeCallSegments	Tmc	short
MoveCallSegments	Tmcs	short
MoveLeg	Tml	short
ReleaseCall	Trc	short
ReportUTSI	Tru	short
RequestCurrentStatusReport	Trcs	short
RequestEveryStatusChangeReport	Tres	short

Operation Name	Timer	Value range
RequestFirstStatusMatchReport	Trfs	short
RequestNotificationChargingEvent	Trnc	short
RequestReportBCSMEvent	Trrb	short
RequestReportUTSI	Trru	short
ResetTimer	T <sub>rt</sub>	short
SelectFacility	T <sub>sf</sub>	short
SendChargingInformation	Tsci	short
SendSTUI	T <sub>ss</sub>	short
ServiceFilteringResponse	Tsfr	short
SetServiceProfile	Tsep	short
SplitLeg	T <sub>sl</sub>	short
StatusReport	Tsrp	short

## 6.5 Packages, contracts, application contexts and abstract syntaxes

### 6.5.1 ASN.1 modules

```
IN-SSF-SCF-pkgs-contracts-acs {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs4(40) modules(1) in-ssf-scf-pkgs-contracts-acs(9) version1(0)}
```

```
DEFINITIONS ::=
```

```
BEGIN
```

```
-- This module describes the operation-packages, contracts and application-contexts used
-- over the SSF-SCF interface.
```

```
IMPORTS
```

```
    IMSI,
    ISDN-AddressString,
    Ext-BasicServiceCode
```

```
FROM MAP-CommonDataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) gsm-Network(1)
modules(3) map-CommonDataTypes(18) version6(6)}
```

```
    LocationInformation,
    SubscriberState
```

```
FROM MAP-MS-DataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) gsm-Network(1)
modules(3) map-MS-DataTypes(11) version6(6)}
```

```
    CallReferenceNumber,
    SuppressionOfAnnouncement
```

```
FROM MAP-CH-DataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) gsm-Network(1)
modules(3) map-CH-DataTypes(13) version6(6)}
```

```
    id-ac-ssf-scfGenericAC,
    id-ac-ssf-scfAssistHandoffAC,
    id-ac-ssf-scfServiceManagementAC,
    id-ac-scf-ssfGenericAC,
    id-ac-scf-ssfTrafficManagementAC,
    id-ac-scf-ssfINTRafficManagementAC,
    id-ac-scf-ssfServiceManagementAC,
    id-ac-scf-ssfStatusReportingAC,
    id-ac-scf-ssfTriggerManagementAC,
```

```
    id-inSsfToScfGeneric,
    id-inAssistHandoffSsfToScf,
    id-inScfToSsfGeneric,
    id-inScfToSsfTrafficManagement,
    id-inScfToSsfINTRafficManagement,
    id-inScfToSsfServiceManagement,
    id-inSsfToScfServiceManagement,
```

```

id-inScfToSsfStatusReporting,
id-inScfToSsfTriggerManagement,

id-as-ssf-scfGenericAS,
id-as-assistHandoff-ssf-scfAS,
id-as-scf-ssfGenericAS,
id-as-scf-ssfTrafficManagementAS,
id-as-scf-ssfINTrafficManagementAS,
id-as-scf-ssfServiceManagementAS,
id-as-ssf-scfServiceManagementAS,
id-as-scf-ssfStatusReportingAS,
id-as-scf-ssfTriggerManagementAS,
id-package-scfActivation,
id-package-srf-scfActivationOfAssist,
id-package-assistConnectionEstablishment,
id-package-genericDisconnectResource,
id-package-nonAssistedConnectionEstablishment,
id-package-connect,
id-package-callHandling,
id-package-bcsmEventHandling,
id-package-chargingEventHandling,
id-package-ssfCallProcessing,
id-package-scfCallInitiation,
id-package-timer,
id-package-billing,
id-package-charging,
id-package-trafficManagement,
id-package-inTrafficManagement,
id-package-serviceManagementActivate,
id-package-serviceManagementResponse,
id-package-callReport,
id-package-signallingControl,
id-package-activityTest,
id-package-statusReporting,
id-package-cancel,
id-package-cphResponse,
id-package-entityReleased,
id-package-triggerManagement,
id-package-uSIHandling,
    id-package-triggerCallManagement,

ros-InformationObjects,
common-classes,
scf-srf-classes,
scf-srf-Operations,
scf-srf-Protocol,
ssf-scf-classes,
ssf-scf-Operations,
tc-Messages,
tc-NotationExtensions
FROM IN-object-identifiers
    {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs4(40) modules(1) in-
object-identifiers(0) version1(0)}

COMMON-BOUNDS,
    EmptyReturnable
FROM IN-Common-Classes common-classes

SCF-SSF-BOUNDS

FROM IN-SSF-SCF-Classes scf-scf-classes

SCF-SRF-BOUNDS

FROM IN-CS3-scf-srf-classes scf-srf-classes

CONTRACT,
OPERATION-PACKAGE,
OPERATION,
ROS-OBJECT-CLASS

FROM Remote-Operations-Information-Objects ros-InformationObjects

TCMessage {}

FROM TCAPMessages tc-Messages

APPLICATION-CONTEXT, dialogue-abstract-syntax

```

FROM TC-Notation-Extensions tc-NotationExtensions

```

activateServiceFiltering {},
activityTest,
applyCharging {},
applyChargingReport {},
assistRequestInstructions {},
callFiltering {},
callGap {},
callInformationReport {},
callInformationRequest {},
cancel {},
cancelStatusReportRequest {},
collectInformation {},
connect {},
connectToResource {},
continue,
continueWithArgument {},
createCallSegmentAssociation {},
createOrRemoveTriggerData {},
disconnectForwardConnection,
disconnectForwardConnectionWithArgument {},
disconnectLeg {},
entityReleased {},
establishTemporaryConnection {},
eventNotificationCharging {},
eventReportBCSM {},
furnishChargingInformation {},
initialDP {},
initiateCallAttempt {},
manageTriggerData {},
mergeCallSegments {},
moveCallSegments {},
moveLeg {},
releaseCall {},
reportUTSI {},
requestCurrentStatusReport {},
requestEveryStatusChangeReport {},
requestFirstStatusMatchReport {},
requestNotificationChargingEvent {},
requestReportBCSMEvent {},
requestReportUTSI {},
resetTimer {},
selectFacility {},
sendChargingInformation {},
sendSTUI {},
serviceFilteringResponse {},
setServiceProfile {},
splitLeg {},
statusReport {}

```

FROM IN-SSF-SCF-ops-args ssf-scf-Operations

```

playAnnouncement {},
promptAndCollectUserInformation {},
promptAndReceiveMessage {},
scriptClose {},
scriptEvent {},
scriptInformation {},
scriptRun {},
specializedResourceReport

```

FROM IN-CS3-scf-srf-ops-args scf-srf-Operations

```

specializedResourceControlPackage {},
scriptControlPackage {},
messageControlPackage {}

```

FROM IN-CS3-SCF-SRF-pkgs-contracts-acs scf-srf-Protocol

```

;
-- The following three definitions are local short-hand notation for convenience.
B1::= COMMON-BOUNDS      -- defined in part 1
B2::= SCF-SSF-BOUNDS     -- defined in this part
B3::= SCF-SRF-BOUNDS     -- defined in EN 301 931-3
networkSpecificB1 COMMON-BOUNDS::=
{  NUM-OF-EXTENSIONS  1}

```

-- The following instance of the parameter bound is just an example

```

networkSpecificB2 SCF-SSF-BOUNDS::=
{
  MINIMUM-FOR-ACH-BILLING-CHARGING 1 -- example value
  MAXIMUM-FOR-ACH-BILLING-CHARGING 5 -- example value
  MINIMUM-FOR-ACH-BILLING-GVNS 1 -- example value
  MAXIMUM-FOR-ACH-BILLING-GVNS 5 -- example value
  MINIMUM-FOR-CALLED-DIRECTORY-NUMBER 1 -- example value
  MAXIMUM-FOR-CALLED-DIRECTORY-NUMBER 5 -- example value
  MINIMUM-FOR-CALLED-PARTY-BCD-NUMBER 1 -- example value
  MAXIMUM-FOR-CALLED-PARTY-BCD-NUMBER 5 -- example value
  MINIMUM-FOR-CALLED-PARTY-NUMBER 1 -- example value
  MAXIMUM-FOR-CALLED-PARTY-NUMBER 5 -- example value
  MINIMUM-FOR-CALLING-GEODETTIC-LOCATION 1 -- example value
  MAXIMUM-FOR-CALLING-GEODETTIC-LOCATION 10 -- example value
  MINIMUM-FOR-CALLING-PARTY-NUMBER 1 -- example value
  MAXIMUM-FOR-CALLING-PARTY-NUMBER 5 -- example value
  MINIMUM-FOR-CALL-RESULT-CSONE 1 -- example value
  MAXIMUM-FOR-CALL-RESULT-CSONE 5 -- example value
  MAXIMUM-FOR-CALL-REFERENCE 5 -- example value
  MINIMUM-FOR-CARRIER 3 -- example value
  MAXIMUM-FOR-CARRIER 10 -- example value
  MINIMUM-FOR-CHARGED-PARTYNUMBER 1 -- example value
  MAXIMUM-FOR-CHARGED-PARTYNUMBER 14 -- example value
  MAXIMUM-FOR-CAUSE 4 -- example value
  MINIMUM-FOR-COMMUNICATION-TARIFF-NUM 1 -- example value
  MAXIMUM-FOR-COMMUNICATION-TARIFF-NUM 4 -- example value
  MAXIMUM-FOR-CNINFO 13 -- example value
  MINIMUM-FOR-DESTINATION 1 -- example value
  MAXIMUM-FOR-DESTINATION 10 -- example value
  MINIMUM-FOR-DIGITS 1 -- example value
  MAXIMUM-FOR-DIGITS 5 -- example value
  MINIMUM-FOR-DISPLAY 1 -- example value
  MAXIMUM-FOR-DISPLAY 5 -- example value
  MINIMUM-FOR-EVENT-SPECIFIC-CHARGING 1 -- example value
  MAXIMUM-FOR-EVENT-SPECIFIC-CHARGING 5 -- example value
  MINIMUM-FOR-EVENT-TYPE-CHARGING 1 -- example value
  MAXIMUM-FOR-EVENT-TYPE-CHARGING 5 -- example value
  MINIMUM-FOR-FCI-CSONE-BILLING-CHARGING 1 -- example value
  MAXIMUM-FOR-FCI-CSONE-BILLING-CHARGING 5 -- example value
  MINIMUM-FOR-FCI-CSTWO-BILLING-CHARGING 1 -- example value
  MAXIMUM-FOR-FCI-CSTWO-BILLING-CHARGING 5 -- example value
  MINIMUM-FOR-FCI-BILLING-CHARGING-DATA 1 -- example value
  MAXIMUM-FOR-FCI-BILLING-CHARGING-DATA 160 -- example value
  MINIMUM-FOR-CAMELFCI-BILLING-CHARGING-DATA 1 -- example value
  MAXIMUM-FOR-CAMELFCI-BILLING-CHARGING-DATA 5 -- example value
  MINIMUM-FOR-FCI-BILLING-CHARGING 5 -- example value
  MAXIMUM-FOR-FCI-BILLING-CHARGING 172 -- example value
  MINIMUM-FOR-FORMAT-DATA-LENGTH 1 -- example value
  MAXIMUM-FOR-FORMAT-DATA-LENGTH 5 -- example value
  MINIMUM-FOR-FORWARD-GVNS 1 -- example value
  MAXIMUM-FOR-FORWARD-GVNS 5 -- example value
  MINIMUM-FOR-GENERIC-NAME 1 -- example value
  MAXIMUM-FOR-GENERIC-NAME 5 -- example value
  MINIMUM-FOR-GENERIC-NUMBER 1 -- example value
  MAXIMUM-FOR-GENERIC-NUMBER 5 -- example value
  MAXIMUM-FOR-GLOBAL-CALLREF 10 -- example value
  MAXIMUM-FOR-INITIAL-TIME-INTERVAL 5 -- example value
  MAXIMUM-FOR-IN-SERVICE-COMPATIBILITY 5 -- example value
  MINIMUM-FOR-IP-AVAILABLE 1 -- example value
  MAXIMUM-FOR-IP-AVAILABLE 5 -- example value
  MINIMUM-FOR-IP-SSP-CAPABILITIES 1 -- example value
  MAXIMUM-FOR-IP-SSP-CAPABILITIES 5 -- example value
  MINIMUM-FOR-ISDN-ACCESS-RELATED-INFO 1 -- example value
  MAXIMUM-FOR-ISDN-ACCESS-RELATED-INFO 10 -- example value
  MINIMUM-FOR-LOCATION-NUMBER 1 -- example value
  MAXIMUM-FOR-LOCATION-NUMBER 5 -- example value
  MINIMUM-FOR-MID-CALL-CONTROL-INFO 1 -- example value
  MAXIMUM-FOR-MID-CALL-CONTROL-INFO 5 -- example value
  MINIMUM-FOR-ORIGINAL-CALLED-PARTY-ID 1 -- example value
  MAXIMUM-FOR-ORIGINAL-CALLED-PARTY-ID 5 -- example value
  MINIMUM-FOR-REASON 1 -- example value
  MAXIMUM-FOR-REASON 5 -- example value
  MINIMUM-FOR-REDIRECTING-ID 1 -- example value
  MAXIMUM-FOR-REDIRECTING-ID 5 -- example value
  MINIMUM-FOR-REQUESTED-UTSI-NUM 1 -- example value
}

```



```

MAXIMUM-FOR-REQUESTED-UTSI-NUM      5  -- example value
MINIMUM-FOR-ROUTE-LIST               1  -- example value
MAXIMUM-FOR-ROUTE-LIST               5  -- example value
MINIMUM-FOR-ROUTING-NUMBER           1  -- example value
MAXIMUM-FOR-ROUTING-NUMBER           5  -- example value
MINIMUM-FOR-SCF-ID                   1  -- example value
MAXIMUM-FOR-SCF-ID                   5  -- example value
MINIMUM-FOR-SCI-BILLING-CHARGING      1  -- example value
MAXIMUM-FOR-SCI-BILLING-CHARGING      5  -- example value
MINIMUM-FOR-SDSS-INFORMATION          1  -- example value
MAXIMUM-FOR-SDSS-INFORMATION          5  -- example value
MINIMUM-FOR-SII                       1  -- example value
MAXIMUM-FOR-SII                       5  -- example value
MINIMUM-FOR-SF-BILLING-CHARGING       1  -- example value
MAXIMUM-FOR-SF-BILLING-CHARGING       5  -- example value
MINIMUM-FOR-SUB-TARRIF-CONTROL-LENGTH 1  -- example value
MAXIMUM-FOR-SUB-TARRIF-CONTROL-LENGTH 5  -- example value
MINIMUM-FOR-TIME-AND-TIMEZONE-LENGTH  1  -- example value
MAXIMUM-FOR-TIME-AND-TIMEZONE-LENGTH  8  -- example value
MINIMUM-FOR-TARIFF-INDICATORS-LENGTH  1  -- example value
MAXIMUM-FOR-TARIFF-INDICATORS-LENGTH  8  -- example value
MINIMUM-FOR-USI-INFORMATION           1  -- example value
MAXIMUM-FOR-USI-INFORMATION           5  -- example value
MINIMUM-FOR-USI-SERVICE-INDICATOR     1  -- example value
MAXIMUM-FOR-USI-SERVICE-INDICATOR     5  -- example value
NUM-OF-BCSM-EVENT                     4  -- example value
NUM-OF-BCUSM-EVENT                     4  -- example value
NUM-OF-CHARGING-EVENT                  4  -- example value
NUM-OF-CSAS                             2  -- example value
NUM-OF-CSS                              2  -- example value
NUM-OF-GENERIC-NUMBERS                 2  -- example value
NUM-OF-INPROFILE                       1  -- example value
NUM-OF-SEVERALTRIGGER                  5  -- example value
NUM-OF-IN-SERVICE-COMPATIBILITY-ID     2  -- example value
NUM-OF-LEGS                             2  -- example value
NUM-OF-MESSAGE-IDS                     2  -- example value
MAXIMUM-FOR-AMOUNT                     2  -- example value
MAXIMUM-FOR-INITIAL-UNIT-INCREMENT     2  -- example value
MAXIMUM-FOR-SCALING-FACTOR             2  -- example value
MAXIMUM-FOR-SEGMENTS-PER-DATA-INTERVAL 5  -- example value
MAXIMUM-FOR-UB-NB-CALL                 5  -- example value
NUM-OF-ADDRESSES                       5  -- example value
MAXIMUM-NB-OF-ALTERNATIVE-IDENTITIES  1  -- example value
}

```

```
networkSpecificB3 SCF-SRF-BOUNDS::=
```

```

{
  MINIMUM-FOR-ATTRIBUTES               1  -- example value
  MAXIMUM-FOR-ATTRIBUTES               5  -- example value
  MINIMUM-FOR-MAIL-BOX-ID              1  -- example value
  MAXIMUM-FOR-MAIL-BOX-ID              5  -- example value
  MINIMUM-FOR-MESSAGE-CONTENT          1  -- example value
  MAXIMUM-FOR-MESSAGE-CONTENT          5  -- example value
  MINIMUM-FOR-RECEIVED-INFORMATION     1  -- example value
  MAXIMUM-FOR-RECEIVED-INFORMATION     5  -- example value
  MAXIMUM-FOR-RECORDING-TIME           5  -- example value
  NUM-OF-MESSAGE-IDS                   2  -- example value
  MAXIMUM-FOR-RECORDED-MESSAGE-UNITS   5  -- example value
  NUM-OF-VARIABLE-PARTS                 5  -- must be 5 or
--   greater.
}

```

```
-- Application Contexts --
```

```
ssf-scfGenericAC APPLICATION-CONTEXT::= {
  CONTRACT      inSsfToScfGeneric
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    ssf-scfGenericAbstractSyntax}
  APPLICATION CONTEXT NAME      id-ac-ssf-scfGenericAC}

```

```
ssf-scfAssistHandoffAC APPLICATION-CONTEXT::= {
  CONTRACT      inAssistHandoffSsfToScf
  DIALOGUE MODE structured

```

```

ABSTRACT SYNTAXES {dialogue-abstract-syntax |
  assistHandoff-ssf-scfAbstractSyntax}
APPLICATION CONTEXT NAME id-ac-ssf-scfAssistHandoffAC}

ssf-scfServiceManagementAC APPLICATION-CONTEXT::= {
  CONTRACT inSsfToScfServiceManagement
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    ssf-scfServiceManagementAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-ssf-scfServiceManagementAC}

scf-ssfGenericAC APPLICATION-CONTEXT::= {
  CONTRACT inScfToSsfGeneric
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    ssf-scfGenericAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-scf-ssfGenericAC}

scf-ssfTrafficManagementAC APPLICATION-CONTEXT::= {
  CONTRACT inScfToSsfTrafficManagement
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    scf-ssfTrafficManagementAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-scf-ssfTrafficManagementAC}

scf-ssfINTrafficManagementAC APPLICATION-CONTEXT::= {
  CONTRACT inScfToSsfINTrafficManagement
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    scf-ssfINTrafficManagementAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-scf-ssfINTrafficManagementAC}

scf-ssfServiceManagementAC APPLICATION-CONTEXT::= {
  CONTRACT inScfToSsfServiceManagement
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    scf-ssfServiceManagementAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-scf-ssfServiceManagementAC}

scf-ssfStatusReportingAC APPLICATION-CONTEXT::= {
  CONTRACT inScfToSsfStatusReporting
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    scf-ssfStatusReportingAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-scf-ssfStatusReportingAC}

scf-ssfTriggerManagementAC APPLICATION-CONTEXT::= {
  CONTRACT inScfToSsfTriggerManagement
  DIALOGUE MODE structured
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    scf-ssfTriggerManagementAbstractSyntax}
  APPLICATION CONTEXT NAME id-ac-scf-ssfTriggerManagementAC}

-- Contracts --

inSsfToScfGeneric CONTRACT::= {
-- dialogue initiated by SSF with InitialDP Operation
--The inSsfToScfGeneric contract expresses the form of the service in which the SSF,
--a ROS object of class ssf, initiates the generic triggering approach contract.
--A ROS-object of class scf, responds to this contract.

  INITIATOR CONSUMER OF { activityTestPackage |
    exceptionInformPackage {networkSpecificB2} |
    scfActivationPackage {networkSpecificB1, networkSpecificB2} }
  RESPONDER CONSUMER OF {activityTestPackage |
    assistConnectionEstablishmentPackage {networkSpecificB1, networkSpecificB2} |
    bcsmEventHandlingPackage {networkSpecificB1, networkSpecificB2} |
    billingPackage {networkSpecificB1, networkSpecificB2} |
    callHandlingPackage { networkSpecificB2} |
    callReportPackage {networkSpecificB1, networkSpecificB2} |
    cancelPackage {networkSpecificB2} |
    chargingEventHandlingPackage {networkSpecificB1, networkSpecificB2} |
    chargingPackage {networkSpecificB1, networkSpecificB2} |

```

```

connectPackage {networkSpecificB1, networkSpecificB2} |
cphResponsePackage {networkSpecificB1, networkSpecificB2} |
genericDisconnectResourcePackage {networkSpecificB1, networkSpecificB2} |
nonAssistedConnectionEstablishmentPackage

{networkSpecificB1, networkSpecificB2} |
  signallingControlPackage {networkSpecificB1, networkSpecificB2} |
  specializedResourceControlPackage {networkSpecificB1, networkSpecificB2,
networkSpecificB3} |
  scriptControlPackage {networkSpecificB1, networkSpecificB2} |
  messageControlPackage {networkSpecificB1, networkSpecificB2, networkSpecificB3} |
  ssfCallProcessingPackage {networkSpecificB1, networkSpecificB2} |
  statusReportingPackage {networkSpecificB1, networkSpecificB2} |
  timerPackage {networkSpecificB1, networkSpecificB2} |
  trafficManagementPackage {networkSpecificB1, networkSpecificB2} |
  usiHandlingPackage {networkSpecificB1, networkSpecificB2} |
  scfCallInitiationPackage {networkSpecificB1, networkSpecificB2}|
  triggerCallManagementPackage {networkSpecificB1, networkSpecificB2}
}
ID      id-inSsfToScfGeneric
}

```

```

inAssistHandoffSsfToScf CONTRACT ::= {
-- dialogue initiated by SSF with AssistRequestInstructions
--The inAssistHandoffSsfToScf contract expresses the form of the service in which the SSF,
-- a ROS-object of class ssf, initiates the Assist or Hand-off contract.
--A ROS-object of class scf, responds to this contract.

```

```

INITIATOR CONSUMER OF { activityTestPackage|
  srf-scfActivationOfAssistPackage {networkSpecificB1, networkSpecificB2} }
RESPONDER CONSUMER OF {activityTestPackage|
  billingPackage {networkSpecificB1, networkSpecificB2} |
  callHandlingPackage { networkSpecificB2} |
  cancelPackage {networkSpecificB2} |
  chargingPackage {networkSpecificB1, networkSpecificB2} |
  signallingControlPackage {networkSpecificB1, networkSpecificB2} |
  genericDisconnectResourcePackage {networkSpecificB1, networkSpecificB2} |
  nonAssistedConnectionEstablishmentPackage {networkSpecificB1, networkSpecificB2} |
  specializedResourceControlPackage {networkSpecificB1, networkSpecificB2,
networkSpecificB3} |
  scriptControlPackage {networkSpecificB1, networkSpecificB2} |
  messageControlPackage {networkSpecificB1, networkSpecificB2, networkSpecificB3} |
  statusReportingPackage {networkSpecificB1, networkSpecificB2} |
  timerPackage {networkSpecificB1, networkSpecificB2}
}
ID      id-inAssistHandoffSsfToScf
}

```

```

inScfToSsfGeneric CONTRACT ::= {
--dialogue initiated by SCF with InitiateCallAttempt or CreateCallSegmentAssociation, Generic Case
--The inScfToSsfGeneric contract expresses the form of the service in which the SCF,
--a ROS-object of class scf, initiates the generic messaging approach for the SCF
--Initiate Call Attempt contract. A ROS-object of class ssf, responds to this contract.

```

```

INITIATOR CONSUMER OF {activityTestPackage|
  assistConnectionEstablishmentPackage {networkSpecificB1, networkSpecificB2} |
  bcsmeventHandlingPackage {networkSpecificB1, networkSpecificB2} |
  billingPackage {networkSpecificB1, networkSpecificB2} |
  callHandlingPackage { networkSpecificB2} |
  callReportPackage {networkSpecificB1, networkSpecificB2} |
  cancelPackage { networkSpecificB2} |
  chargingEventHandlingPackage {networkSpecificB1, networkSpecificB2} |
  chargingPackage {networkSpecificB1, networkSpecificB2} |
  connectPackage {networkSpecificB1, networkSpecificB2} |
  cphResponsePackage {networkSpecificB1, networkSpecificB2} |
  genericDisconnectResourcePackage {networkSpecificB1, networkSpecificB2} |
  nonAssistedConnectionEstablishmentPackage {networkSpecificB1, networkSpecificB2} |
  scfCallInitiationPackage {networkSpecificB1, networkSpecificB2} |
  signallingControlPackage {networkSpecificB1, networkSpecificB2} |
  specializedResourceControlPackage {networkSpecificB1, networkSpecificB2,
networkSpecificB3} |
  scriptControlPackage {networkSpecificB1, networkSpecificB2} |
  messageControlPackage {networkSpecificB1, networkSpecificB2, networkSpecificB3} |
  ssfCallProcessingPackage {networkSpecificB1, networkSpecificB2} |
  statusReportingPackage {networkSpecificB1, networkSpecificB2} |

```

```

    timerPackage {networkSpecificB1, networkSpecificB2} |
    usiHandlingPackage {networkSpecificB1, networkSpecificB2} |
    scfCallInitiationPackage {networkSpecificB1, networkSpecificB2} |
    triggerCallManagementPackage {networkSpecificB1, networkSpecificB2}
  }

  RESPONDER CONSUMER OF { activityTestPackage |
    exceptionInformPackage {networkSpecificB2} }
  ID      id-inScfToSsfGeneric
}

inScfToSsfTrafficManagement CONTRACT::= {
-- dialogue initiated by SCF with CallGap
-- The inScfToSsfTrafficManagement contract expresses the form of the service in
-- which the SCF, a ROS-object of class scf, initiates the Traffic Management related contract.
-- A ROS-object of class ssf, responds to this contract.

  INITIATOR CONSUMER OF {trafficManagementPackage {networkSpecificB1, networkSpecificB2}
    }
  ID      id-inScfToSsfTrafficManagement
}

inScfToSsfINTrafficManagement CONTRACT::= {
-- dialogue initiated by SCF with CallGap
-- The inScfToSsfINTrafficManagement contract expresses the form of the service in
-- which the SCF, a ROS-object of class scf, initiates the Traffic Management related contract.
-- A ROS-object of class ssf, responds to this contract.

  INITIATOR CONSUMER OF {inTrafficManagementPackage {networkSpecificB1, networkSpecificB2,
networkSpecificB3}
    }
  ID      id-inScfToSsfINTrafficManagement
}

inScfToSsfServiceManagement CONTRACT::= {
-- dialogue initiated by SCF with ActivateServiceFiltering
--The inScfToSsfServiceManagement contract expresses the form of the service
-- in which the SCF, a ROS-object of class scf, initiates the Service Management related contract.
-- A ROS-object of class ssf, in the context of a separate contract, responds to this initiation.

  INITIATOR CONSUMER OF {serviceManagementActivatePackage {networkSpecificB1, networkSpecificB2}
    }
  ID      id-inScfToSsfServiceManagement
}

inSsfToScfServiceManagement CONTRACT::= {
-- dialogue initiated/ended by SSF with ServiceFilteringResponse
-- The inSsfToScfServiceManagement contract expresses the form of the service in
-- which the SSF, a ROS-object of class ssf, initiates the Service Management related contract
-- for reporting Service Management results.

  INITIATOR CONSUMER OF {serviceManagementResponsePackage {networkSpecificB1, networkSpecificB2}
    }
  ID      id-inSsfToScfServiceManagement
}

inScfToSsfStatusReporting CONTRACT::= {
-- dialogue initiated by SCF with StatusReporting Operations
--The inScfToSsfStatusReporting contract expresses the form of the service
--in which the SCF, a ROS-object of class scf, initiates the Status Reporting related contract.
--A ROS-object of class ssf, responds to this contract.

  INITIATOR CONSUMER OF {
    statusReportingPackage {networkSpecificB1, networkSpecificB2}
  }
  ID      id-inScfToSsfStatusReporting
}

inScfToSsfTriggerManagement CONTRACT::= {
-- dialogue initiated by SCF with the trigger management operations CreateOrRemoveTriggerData
-- or ManageTriggerData
-- The inScfToSsfTriggerManagement contract expresses the form of the service in which the SCF,
-- a ROS-object of class scf, initiates the Trigger Management related contract
-- A ROS-object of class ssf, in the context of a separate contract, responds to this initiation.
  INITIATOR CONSUMER OF {triggerManagementPackage {networkSpecificB1, networkSpecificB2,
networkSpecificB3}
    }
  ID      id-inScfToSsfTriggerManagement
}

```

-- Packages --

```

scfActivationPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    initialDP { b1, b2}
  }
  ID id-package-scfActivation}

srf-scfActivationOfAssistPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    assistRequestInstructions {b1, b2}
  }
  ID id-package-srf-scfActivationOfAssist}

assistConnectionEstablishmentPackage { B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    establishTemporaryConnection {b1,b2}
  }
  ID id-package-assistConnectionEstablishment}

genericDisconnectResourcePackage { B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    disconnectForwardConnection |
    disconnectForwardConnectionWithArgument {b1, b2}
  }
  ID id-package-genericDisconnectResource}

nonAssistedConnectionEstablishmentPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    connectToResource {b1, b2}
  }
  ID id-package-nonAssistedConnectionEstablishment}

connectPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    connect {b1, b2}
  }
  ID id-package-connect}

callHandlingPackage {B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    releaseCall {b2}
  }
  ID id-package-callHandling}

bcsmEventHandlingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    requestReportBCSMEvent {b1, b2}
  }
  SUPPLIER INVOKES {
    eventReportBCSM {b1, b2}
  }
  ID id-package-bcsmEventHandling}

chargingEventHandlingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    requestNotificationChargingEvent {b2}
  }
  SUPPLIER INVOKES {
    eventNotificationCharging {b1, b2}
  }
  ID id-package-chargingEventHandling}

ssfCallProcessingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    collectInformation {b1} |
    selectFacility {b1, b2}|
    continue
  }
  ID id-package-ssfCallProcessing}

scfCallInitiationPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {

```

```

        initiateCallAttempt {b1, b2}
    }
    ID      id-package-scfCallInitiation}

timerPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        resetTimer {b1, b2}
    }
    ID      id-package-timer}

billingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        furnishChargingInformation {b1, b2}
    }
    ID      id-package-billing}

chargingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        applyCharging {b1, b2}
    }
    SUPPLIER INVOKES    {
        applyChargingReport {b1, b2}
    }
    ID      id-package-charging}

trafficManagementPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        callFiltering {b1, b2}
    }
    ID      id-package-trafficManagement}

inTrafficManagementPackage {B1: b1, B2: b2, B3: b3 } OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        callGap {b1, b2, b3}
    }
    ID      id-package-inTrafficManagement}

serviceManagementActivatePackage {B1: b1, B2: b2, B3: b3} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        activateServiceFiltering {b1, b2, b3}
    }
    ID      id-package-serviceManagementActivate}

serviceManagementResponsePackage {B1: b1, B2: b2 } OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        serviceFilteringResponse {b1, b2}
    }
    ID      id-package-serviceManagementResponse}

callReportPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        callInformationRequest {b1}
    }
    SUPPLIER INVOKES    {
        callInformationReport {b1, b2}
    }
    ID      id-package-callReport}

signallingControlPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        sendChargingInformation {b1, b2}
    }
    ID      id-package-signallingControl}

activityTestPackage OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        activityTest
    }
    ID      id-package-activityTest}

statusReportingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
    CONSUMER INVOKES    {
        cancelStatusReportRequest {b1, b2}|
        requestCurrentStatusReport {b1, b2}|
        requestEveryStatusChangeReport {b1, b2}|
        requestFirstStatusMatchReport {b1, b2}
    }
}

```

```

SUPPLIER INVOKES {
  statusReport {b1, b2}
}
ID id-package-statusReporting}

cancelPackage {B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    cancel {b2}
  }
  ID id-package-cancel}

cphResponsePackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    continueWithArgument {b1, b2}|
    disconnectLeg {b1, b2}|
    mergeCallSegments {b1, b2}|
    moveCallSegments {b1, b2}|
    moveLeg {b1, b2}|
    createCallSegmentAssociation {b1, b2} |
    splitLeg {b1, b2}
  }
  ID id-package-cphResponse}

exceptionInformPackage {B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    entityReleased {b2}
  }
  ID id-package-entityReleased}

triggerCallManagementPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    setServiceProfile {b1, b2}
  }
  ID id-package-triggerCallManagement}

triggerManagementPackage {B1: b1, B2: b2, B3:b3} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    createOrRemoveTriggerData {b1, b2, b3}|
    manageTriggerData {b1, b2}
  }
  ID id-package-triggerManagement}

uSIHandlingPackage {B1: b1, B2: b2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {
    requestReportUTSI {b1, b2}|
    sendSTUI {b1, b2}}
  SUPPLIER INVOKES {
    reportUTSI {b1, b2}
  }
  ID id-package-uSIHandling
}

```

-- Abstract Syntaxes --

```

ssf-scfGenericAbstractSyntax ABSTRACT-SYNTAX::= {
  GenericSSF-SCF-PDUs
  IDENTIFIED BY id-as-ssf-scfGenericAS}

GenericSSF-SCF-PDUs::= TCMMessage {{SsfToScfGenericInvokable},
  {SsfToScfGenericReturnable}}

SsfToScfGenericInvokable OPERATION::= {
  activateServiceFiltering {networkSpecificB1, networkSpecificB2, networkSpecificB3} |
  activityTest |
  applyCharging {networkSpecificB1, networkSpecificB2} |
  applyChargingReport {networkSpecificB1, networkSpecificB2} |
  callInformationReport {networkSpecificB1, networkSpecificB2} |
  callInformationRequest {networkSpecificB1} |
  cancel {networkSpecificB2} |
  cancelStatusReportRequest {networkSpecificB1, networkSpecificB2} |
  collectInformation {networkSpecificB1} |
  connect {networkSpecificB1, networkSpecificB2} |
  connectToResource {networkSpecificB1, networkSpecificB2}|
  createCallSegmentAssociation {networkSpecificB1, networkSpecificB2} |
  disconnectForwardConnection |
  disconnectForwardConnectionWithArgument {networkSpecificB1, networkSpecificB2}|
}

```

```

disconnectLeg {networkSpecificB1, networkSpecificB2} |
entityReleased {networkSpecificB2} |
establishTemporaryConnection {networkSpecificB1, networkSpecificB2} |
eventNotificationCharging {networkSpecificB1, networkSpecificB2} |
eventReportBCSM {networkSpecificB1, networkSpecificB2} |
furnishChargingInformation {networkSpecificB1, networkSpecificB2} |
initialDP {networkSpecificB1, networkSpecificB2} |
mergeCallSegments {networkSpecificB1, networkSpecificB2} |
moveCallSegments {networkSpecificB1, networkSpecificB2} |
moveLeg {networkSpecificB1, networkSpecificB2} |
releaseCall { networkSpecificB2} |
reportUTSI {networkSpecificB1, networkSpecificB2} |
requestCurrentStatusReport {networkSpecificB1, networkSpecificB2} |
requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2} |
requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2} |
requestNotificationChargingEvent {networkSpecificB2} |
requestReportBCSMEEvent {networkSpecificB1, networkSpecificB2} |
requestReportUTSI {networkSpecificB1, networkSpecificB2} |
resetTimer {networkSpecificB1, networkSpecificB2} |
sendChargingInformation {networkSpecificB1, networkSpecificB2} |
sendSTUI {networkSpecificB1, networkSpecificB2} |
serviceFilteringResponse {networkSpecificB1, networkSpecificB2} |
setServiceProfile {networkSpecificB1, networkSpecificB2}|
splitLeg {networkSpecificB1, networkSpecificB2} |
statusReport {networkSpecificB1, networkSpecificB2} |
playAnnouncement {networkSpecificB1, networkSpecificB2, networkSpecificB3 } |
promptAndCollectUserInfo {networkSpecificB1, networkSpecificB2,
networkSpecificB3 } |
scriptClose {networkSpecificB1, networkSpecificB2} |
scriptEvent {networkSpecificB1, networkSpecificB2} |
scriptInformation {networkSpecificB1, networkSpecificB2} |
scriptRun {networkSpecificB1, networkSpecificB2} |
specializedResourceReport |
promptAndReceiveMessage {networkSpecificB1, networkSpecificB2, networkSpecificB3 }
}

```

```

SsfToScfGenericReturnable OPERATION ::= {
activateServiceFiltering {networkSpecificB1, networkSpecificB2, networkSpecificB3} |
activityTest |
applyCharging {networkSpecificB1, networkSpecificB2} |
applyChargingReport {networkSpecificB1, networkSpecificB2} |
callGap {networkSpecificB1, networkSpecificB2, networkSpecificB3} |
callInformationRequest {networkSpecificB1} |
cancel {networkSpecificB2} |
cancelStatusReportRequest {networkSpecificB1, networkSpecificB2} |
collectInformation {networkSpecificB1 } |
connect {networkSpecificB1, networkSpecificB2} |
connectToResource {networkSpecificB1, networkSpecificB2} |
continue |
continueWithArgument {networkSpecificB1, networkSpecificB2}|
createCallSegmentAssociation {networkSpecificB1, networkSpecificB2}|
disconnectForwardConnection |
disconnectForwardConnectionWithArgument {networkSpecificB1, networkSpecificB2}|
disconnectLeg {networkSpecificB1, networkSpecificB2}|
establishTemporaryConnection {networkSpecificB1, networkSpecificB2}|
furnishChargingInformation {networkSpecificB1, networkSpecificB2}|
initialDP {networkSpecificB1, networkSpecificB2}|
mergeCallSegments {networkSpecificB1, networkSpecificB2}|
moveCallSegments {networkSpecificB1, networkSpecificB2}|
moveLeg {networkSpecificB1, networkSpecificB2}|
releaseCall { networkSpecificB2}|
requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}|
requestNotificationChargingEvent {networkSpecificB2}|
requestReportBCSMEEvent {networkSpecificB1, networkSpecificB2}|
requestReportUTSI {networkSpecificB1, networkSpecificB2}|
resetTimer {networkSpecificB1, networkSpecificB2}|
sendChargingInformation {networkSpecificB1, networkSpecificB2}|
sendSTUI {networkSpecificB1, networkSpecificB2}|
setServiceProfile {networkSpecificB1, networkSpecificB2}|
splitLeg {networkSpecificB1, networkSpecificB2}|
playAnnouncement {networkSpecificB1, networkSpecificB2, networkSpecificB3 }|
promptAndCollectUserInfo {networkSpecificB1, networkSpecificB2,
networkSpecificB3 }|
scriptClose {networkSpecificB1, networkSpecificB2}|
scriptInformation {networkSpecificB1, networkSpecificB2}|
scriptRun {networkSpecificB1, networkSpecificB2}|
}

```



```

promptAndReceiveMessage {networkSpecificB1, networkSpecificB2, networkSpecificB3 }
}

assistHandoff-ssf-scfAbstractSyntax ABSTRACT-SYNTAX ::= {
  AssistHandoffSSF-SCF-PDUs
  IDENTIFIED BY id-as-assistHandoff-ssf-scfAS}

AssistHandoffSSF-SCF-PDUs ::= TCMMessage {{AssistHandoffSsfToScfInvokable},
  {AssistHandoffSsfToScfReturnable}}

AssistHandoffSsfToScfInvokable OPERATION ::= {
  activityTest |
  applyCharging {networkSpecificB1, networkSpecificB2}|
  applyChargingReport {networkSpecificB1, networkSpecificB2}|
  assistRequestInstructions {networkSpecificB1, networkSpecificB2}|
  cancel {networkSpecificB2}|
  cancelStatusReportRequest {networkSpecificB1, networkSpecificB2}|
  connectToResource {networkSpecificB1, networkSpecificB2}|
  disconnectForwardConnection |
  disconnectForwardConnectionWithArgument {networkSpecificB1, networkSpecificB2}|
  furnishChargingInformation {networkSpecificB1, networkSpecificB2}|
  playAnnouncement {networkSpecificB1, networkSpecificB2, networkSpecificB3 }|
  promptAndCollectUserInformation {networkSpecificB1, networkSpecificB2,
  networkSpecificB3 }|
  requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
  requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
  requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}|
  resetTimer {networkSpecificB1, networkSpecificB2}|
  statusReport {networkSpecificB1, networkSpecificB2}|
  scriptClose {networkSpecificB1, networkSpecificB2}|
  scriptEvent {networkSpecificB1, networkSpecificB2}|
  scriptInformation {networkSpecificB1, networkSpecificB2}|
  scriptRun {networkSpecificB1, networkSpecificB2}|
  sendChargingInformation {networkSpecificB1, networkSpecificB2}|
  specializedResourceReport |
  promptAndReceiveMessage {networkSpecificB1, networkSpecificB2, networkSpecificB3 }
}

AssistHandoffSsfToScfReturnable OPERATION ::= {
  activityTest |
  applyCharging {networkSpecificB1, networkSpecificB2}|
  applyChargingReport {networkSpecificB1, networkSpecificB2}|
  assistRequestInstructions {networkSpecificB1, networkSpecificB2}|
  cancel {networkSpecificB2}|
  cancelStatusReportRequest {networkSpecificB1, networkSpecificB2}|
  connectToResource {networkSpecificB1, networkSpecificB2}|
  disconnectForwardConnection |
  disconnectForwardConnectionWithArgument {networkSpecificB1, networkSpecificB2}|
  furnishChargingInformation {networkSpecificB1, networkSpecificB2}|
  playAnnouncement {networkSpecificB1, networkSpecificB2, networkSpecificB3 }|
  promptAndCollectUserInformation {networkSpecificB1, networkSpecificB2,
  networkSpecificB3 }|
  requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
  requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
  requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}|
  resetTimer {networkSpecificB1, networkSpecificB2}|
  scriptClose {networkSpecificB1, networkSpecificB2}|
  scriptInformation {networkSpecificB1, networkSpecificB2}|
  scriptRun {networkSpecificB1, networkSpecificB2}|
  promptAndReceiveMessage {networkSpecificB1, networkSpecificB2, networkSpecificB3 }
}

ssfGenericAbstractSyntax ABSTRACT-SYNTAX ::= {
  GenericSCF-SSF-PDUs
  IDENTIFIED BY id-as-ssf-genericAS}

GenericSCF-SSF-PDUs ::= TCMMessage {{ScfToSsfGenericInvokable}, {ScfToSsfGenericReturnable}}

ScfToSsfGenericInvokable OPERATION ::= {
  activateServiceFiltering {networkSpecificB1, networkSpecificB2, networkSpecificB3}|
  activityTest |
  applyCharging {networkSpecificB1, networkSpecificB2}|
  applyChargingReport {networkSpecificB1, networkSpecificB2}|
  callInformationRequest {networkSpecificB1}|
  cancel {networkSpecificB2}|
  cancelStatusReportRequest {networkSpecificB1, networkSpecificB2}|

```

```

collectInformation {networkSpecificB1 }|
connect {networkSpecificB1, networkSpecificB2}|
connectToResource {networkSpecificB1, networkSpecificB2}|
continue |
continueWithArgument{networkSpecificB1, networkSpecificB2}|
createCallSegmentAssociation {networkSpecificB1, networkSpecificB2}|
disconnectForwardConnection |
disconnectForwardConnectionWithArgument {networkSpecificB1, networkSpecificB2}|
disconnectLeg {networkSpecificB1, networkSpecificB2}|
establishTemporaryConnection {networkSpecificB1, networkSpecificB2}|
furnishChargingInformation {networkSpecificB1, networkSpecificB2}|
initiateCallAttempt {networkSpecificB1, networkSpecificB2}|
mergeCallSegments {networkSpecificB1, networkSpecificB2}|
moveCallSegments {networkSpecificB1, networkSpecificB2}|
moveLeg {networkSpecificB1, networkSpecificB2}|
releaseCall { networkSpecificB2}|
requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}|
requestNotificationChargingEvent {networkSpecificB2}|
requestReportBCSMEEvent {networkSpecificB1, networkSpecificB2}|
requestReportUTSI {networkSpecificB1, networkSpecificB2}|
resetTimer {networkSpecificB1, networkSpecificB2}|
sendChargingInformation {networkSpecificB1, networkSpecificB2}|
sendSTUI {networkSpecificB1, networkSpecificB2}|
setServiceProfile {networkSpecificB1, networkSpecificB2}|
splitLeg {networkSpecificB1, networkSpecificB2}|
playAnnouncement {networkSpecificB1, networkSpecificB2, networkSpecificB3 }|
promptAndCollectUserInformation {networkSpecificB1, networkSpecificB2,
networkSpecificB3 }|
scriptClose {networkSpecificB1, networkSpecificB2}|
scriptInformation {networkSpecificB1, networkSpecificB2}|
scriptRun {networkSpecificB1, networkSpecificB2}|
promptAndReceiveMessage {networkSpecificB1, networkSpecificB2, networkSpecificB3 }
}

```

```

ScfToSsfGenericReturnable OPERATION ::= {
activateServiceFiltering {networkSpecificB1, networkSpecificB2, networkSpecificB3}|
activityTest |
applyCharging {networkSpecificB1, networkSpecificB2}|
applyChargingReport {networkSpecificB1, networkSpecificB2}|
callInformationReport {networkSpecificB1, networkSpecificB2}|
callInformationRequest {networkSpecificB1}|
cancel {networkSpecificB2}|
cancelStatusReportRequest {networkSpecificB1, networkSpecificB2}|
collectInformation {networkSpecificB1 }|
connect {networkSpecificB1, networkSpecificB2}|
connectToResource {networkSpecificB1, networkSpecificB2}|
createCallSegmentAssociation {networkSpecificB1, networkSpecificB2}|
disconnectForwardConnection |
disconnectForwardConnectionWithArgument {networkSpecificB1, networkSpecificB2}|
disconnectLeg {networkSpecificB1, networkSpecificB2}|
entityReleased {networkSpecificB2}|
establishTemporaryConnection {networkSpecificB1, networkSpecificB2}|
eventNotificationCharging {networkSpecificB1, networkSpecificB2} |
resetTimer {networkSpecificB1, networkSpecificB2}|
eventReportBCSM {networkSpecificB1, networkSpecificB2}|
furnishChargingInformation {networkSpecificB1, networkSpecificB2}|
initiateCallAttempt {networkSpecificB1, networkSpecificB2}|
mergeCallSegments {networkSpecificB1, networkSpecificB2}|
moveCallSegments {networkSpecificB1, networkSpecificB2}|
moveLeg {networkSpecificB1, networkSpecificB2}|
reportUTSI {networkSpecificB1, networkSpecificB2}|
requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}|
requestNotificationChargingEvent {networkSpecificB2}|
requestReportBCSMEEvent {networkSpecificB1, networkSpecificB2}|
requestReportUTSI {networkSpecificB1, networkSpecificB2} |
sendChargingInformation {networkSpecificB1, networkSpecificB2}|
sendSTUI {networkSpecificB1, networkSpecificB2}|
serviceFilteringResponse {networkSpecificB1, networkSpecificB2}|
setServiceProfile {networkSpecificB1, networkSpecificB2}|
splitLeg {networkSpecificB1, networkSpecificB2}|
statusReport {networkSpecificB1, networkSpecificB2}|
playAnnouncement {networkSpecificB1, networkSpecificB2, networkSpecificB3 }|
promptAndCollectUserInformation {networkSpecificB1, networkSpecificB2,
networkSpecificB3 }|
}

```

```

scriptClose {networkSpecificB1, networkSpecificB2}|
scriptEvent {networkSpecificB1, networkSpecificB2}|
scriptInformation {networkSpecificB1, networkSpecificB2}|
scriptRun {networkSpecificB1, networkSpecificB2}|
specializedResourceReport |
promptAndReceiveMessage {networkSpecificB1, networkSpecificB2, networkSpecificB3 }
}

```

```

scf-ssfTrafficManagementAbstractSyntax ABSTRACT-SYNTAX ::= {
  TrafficManagementSCF-SSF-PDUs
  IDENTIFIED BY id-as-scf-ssfTrafficManagementAS}

TrafficManagementSCF-SSF-PDUs ::= TCMMessage {{ScfToSsfTrafficManagementInvokable}, {EmptyReturnable}}

ScfToSsfTrafficManagementInvokable OPERATION ::= {
  callFiltering {networkSpecificB1, networkSpecificB2}
}

scf-ssfINTrafficManagementAbstractSyntax ABSTRACT-SYNTAX ::= {
  INTrafficManagementSCF-SSF-PDUs
  IDENTIFIED BY id-as-scf-ssfINTrafficManagementAS}

INTrafficManagementSCF-SSF-PDUs ::= TCMMessage {{ScfToSsfINTrafficManagementInvokable},
{EmptyReturnable}}

ScfToSsfINTrafficManagementInvokable OPERATION ::= {
  callGap {networkSpecificB1, networkSpecificB2, networkSpecificB3}
}

scf-ssfServiceManagementAbstractSyntax ABSTRACT-SYNTAX ::= {
  ServiceManagementSCF-SSF-PDUs
  IDENTIFIED BY id-as-scf-ssfServiceManagementAS}

ServiceManagementSCF-SSF-PDUs ::= TCMMessage {{ScfToSsfServiceManagementInvokable},
{ScfToSsfServiceManagementReturnable}}

ScfToSsfServiceManagementInvokable OPERATION ::= {
  activateServiceFiltering {networkSpecificB1, networkSpecificB2, networkSpecificB3}
}

ScfToSsfServiceManagementReturnable OPERATION ::= {
  activateServiceFiltering {networkSpecificB1, networkSpecificB2, networkSpecificB3}
}

ssf-scfServiceManagementAbstractSyntax ABSTRACT-SYNTAX ::= {
  ServiceManagementSSF-SCF-PDUs
  IDENTIFIED BY id-as-ssf-scfServiceManagementAS}

ServiceManagementSSF-SCF-PDUs ::= TCMMessage {{SsfToScfServiceManagementInvokable}, {EmptyReturnable}}
}

SsfToScfServiceManagementInvokable OPERATION ::= {
  serviceFilteringResponse {networkSpecificB1, networkSpecificB2}
}

scf-ssfStatusReportingAbstractSyntax ABSTRACT-SYNTAX ::= {
  StatusReportingSCF-SSF-PDUs
  IDENTIFIED BY id-as-scf-ssfStatusReportingAS}

StatusReportingSCF-SSF-PDUs ::= TCMMessage {{ScfToSsfStatusReportingInvokable},
{ScfToSsfStatusReportingReturnable}}

ScfToSsfStatusReportingInvokable OPERATION ::= {

  cancelStatusReportRequest {networkSpecificB1, networkSpecificB2}|
  requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
  requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
  requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}
}

ScfToSsfStatusReportingReturnable OPERATION ::= {

  cancelStatusReportRequest {networkSpecificB1, networkSpecificB2}|
  requestCurrentStatusReport {networkSpecificB1, networkSpecificB2}|
}

```

```
    requestEveryStatusChangeReport {networkSpecificB1, networkSpecificB2}|
    requestFirstStatusMatchReport {networkSpecificB1, networkSpecificB2}|
    statusReport {networkSpecificB1, networkSpecificB2}
  }

scf-ssfTriggerManagementAbstractSyntax ABSTRACT-SYNTAX ::= {
  TriggerManagementSCF-SSF-PDUs
  IDENTIFIED BY id-as-scf-ssfTriggerManagementAS}

TriggerManagementSCF-SSF-PDUs ::= TCMMessage {{ScfToSsfTriggerManagementInvokable},
  {ScfToSsfTriggerManagementReturnable}}

ScfToSsfTriggerManagementInvokable OPERATION ::= {
  createOrRemoveTriggerData {networkSpecificB1, networkSpecificB2, networkSpecificB3}|
  manageTriggerData {networkSpecificB1, networkSpecificB2}
}

ScfToSsfTriggerManagementReturnable OPERATION ::= {
  createOrRemoveTriggerData {networkSpecificB1, networkSpecificB2, networkSpecificB3}|
  manageTriggerData {networkSpecificB1, networkSpecificB2}
}

END
```

---

## History

<b>Document history</b>			
V1.1.1	April 2002	Public Enquiry	PE 20020802: 2002-04-03 to 2002-08-02