Oracle® Communications Network Charging and Control

SMINAP Control Agent Technical Guide

Release 15.2

January 2026

Copyright

Contents

About This Document

Scope

The scope of this document includes all the information required to install, configure and administer the SMINAP Control Agent application.

Audience

This guide was written primarily for system administrators and persons installing, configuring and administering the SMINAP Control Agent application. However, sections of the document may be useful to anyone requiring an introduction to the application.

Prerequisites

A solid understanding of UNIX and a familiarity with IN concepts are an essential prerequisite for safely using the information contained in this technical guide. Attempting to install, remove, configure or otherwise alter the described system without the appropriate background skills, could cause damage to the system; including temporary or permanent incorrect operation, loss of service, and may render your system beyond recovery.

Although it is not a prerequisite to using this guide, familiarity with the target platform would be an advantage.

This manual describes system tasks that should only be carried out by suitably trained operators.

Related Documents

The following documents are related to this document:

- *Advanced Control Services Technical Guide*
- *Charging Control Services Technical Guide*
- *Charging Control Services User's Guide*
- *Service Management System Technical Guide*
- *Service Management System User's Guide*
- *Service Logic Execution Environment Technical Guide*

Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Network Charging and Control (NCC) documentation.

Formatting Convention	Type of Information
Special Bold	Items you must select, such as names of tabs. Names of database tables and fields.
Italics	Name of a document, chapter, topic or other publication. Emphasis within text.
Button	The name of a button to click or a key to press. **Example:** To close the window, either click **Close**, or press **Esc**.
Key+Key	Key combinations for which the user must press and hold down one key and then press another. Example: **Ctrl+P** or **Alt+F4**.
`Monospace`	Examples of code or standard output.
`Monospace Bold`	Text that you must enter.
variable	Used to indicate variables or text that should be replaced with an actual value.
menu option > menu option >	Used to indicate the cascading menu option to be selected. Example: **Operator Functions > Report Functions**
hypertext link	Used to indicate a hypertext link.

Chapter 1

System Overview

Overview

Introduction

This chapter provides a high-level overview of the application. It explains the basic functionality of the system and lists the main components.

It is not intended to advise on any specific Oracle Communications Network Charging and Control (NCC) network or service implications of the product.

In this chapter

This chapter contains the following topic:

- What is SMINAP Control Agent?

What is SMINAP Control Agent?

Introduction

The SMInap Control Agent (SMINAPCA) interface has been developed on the SLEE to support CAPv1 call flows. CAPv1 is the signalling protocol used for mobility management in roaming calls. CAPv1 call flow is applicable to both Mobile Originating as well as Mobile Terminating call related activities.

Features

- SMINAPCA is a Customized Application for Mobile networks Enhanced Logic (CAMEL) Application Part (CAP) version 1 (CAPv1) gateway.
- SMINAPCA is positioned in between the Signalling Transport (SIGTRAN) interface and slee_acs. It converts the CAPv1 signals coming from SIGTRAN interface to INAP (Intelligent Network Application Part) signals and sends them to slee_acs.

Chapter 2

Configuration

Overview

Introduction

This chapter explains how to configure the Oracle Communications Network Charging and Control (NCC) application.

In this chapter

This chapter contains the following topics:

Configuration Overview

Introduction

This topic provides a high level overview of how the SMINAPCA interface is configured. There are configuration options which are added to the configuration files that are not explained in this chapter. These configuration options are required by the application and should not be changed.

Define the necessary entries in **SLEE.cfg** to redirect the desired network SKs to SMINAPCA. Here we are associating the service key 350 to smInapCA.

```
SERVICE=CCS_SMINAPCA 1 slee_acs CCS
SERVICEKEY=INTEGER 99 CCS_SMINAPCA
SERVICEKEY=INTEGER 350 smInapCa #350
```

Above configurations will ensure:

- Network traffic received by the SIGTRAN stack that matches the SK 350 is sent to SMINAPCA instance.
  For more information about how to create hexadecimal SK, see the *SIGTRAN Technical Guide*.
- Service Key 99 triggers the CCS service in slee_acs.

Add the below line to configure the smInapCA interface.

```
INTERFACE=smInapCA smInapCA.sh /IN/service_packages/SMINAPCA/bin/ EVENT
```

For more information on SLEE configuration, see *SLEE Technical Guide*.

SMINAPCA Configuration

Introduction

SMINAPCA reads its configuration from **eserv.config.smInapCA** file. The **eserv.config.smInapCA** file is located in the **/IN/service_packages/SMINAPCA/etc** directory.

Configuration File Format

To organize the configuration data within the **eserv.config** file, some sections are nested within other sections. Configuration details are opened and closed using either { } or [ ].

- Groups of parameters are enclosed with curly brackets – { }
- An array of parameters is enclosed in square brackets – [ ]
- Comments are prefaced with a # at the beginning of the line

To list things within a group or an array, elements must be separated by at least one comma or at least one line break. For example:

```
maximumDurations = [
   called number prefix (default: "" / none)
   maximum time in deci-seconds (default: 0 / infinite)
   { prefix = "0704", maxTime = 900 }
   { prefix = "0701", maxTime = 9000 }
  ]
```

SMINAPCA eserv.config.smInapCA example section

SMINAPCA adds a cut-down `SMINAPCA` section to **eserv.config.smInapCA** file. It is not a full list of all parameters that are available.

The SMINAPCA section contains initial values that you need to amend to suit a specific installation. After amended, SMINAPCA runs with no further changes to Oracle.

Editing the File

Open the configuration file on your system using a standard text editor. Do not use text editors, such as Microsoft Word, that attach control characters. These can be, for example, Microsoft DOS or Windows line termination characters (for example, ^M), which are not visible to the user, at the end of each row. This causes file errors when the application tries to read the configuration file.

Always keep a backup of your file before making any changes to it. This ensures you have a working copy to which you can return.

Loading eserv.config Changes

If you change the configuration file, you must restart the appropriate parts of the service to enable the new options to take effect.

Chapter 3

Background Processes

Overview

Introduction

This chapter explains the process which runs automatically as part of the Oracle Communications Network Charging and Control (NCC) application. This process is started automatically by the SLEE.

In this chapter

This chapter contains the following topic:

- SmInapCA

SmInapCA

Purpose

The SMInap Control Agent (SmInapCA) interface is developed on the SLEE to support CAPv1 call flows for roaming traffic.

Startup

This task is started by the SLEE, by the following line in the **SLEE.cfg** configuration file:

```
INTERFACE=smInapCA smInapCa.sh /IN/service_packages/SMINAPCA/bin/ EVENT
```

Configuration

To load and operate, the smInapCA reads the smInapCA section of the eserv.config.smInapCA file. The high-level structure of the SMINAPCA section is shown below:

```
smInapCA = {
    maximumDurations = [
        # maxTime specified in deci-seconds
        { prefix = "701", maxTime = 9000 }
        { prefix = "704", maxTime = 900 }
    ]
    noAnswerTimeout = 30
    noAnswerTimeoutNetworkLatency = 1
    globalActivityTestInterval = 100
    activityTestTimeout = 50
    capV1Support = true
    appTimeout = 0
    appExpiryReleaseCause = 31
    achExpiryReleaseCause = 31
    atExpiryReleaseCause = 31
    replaceZeroReleaseCause = 31
    appServiceKey = 1
    hlcExtensionId = 1
    pointCode = 0
    subSystemNumber = 0
    globalTitleType = 1
```

```
        globalTitleNoA = 5
        globalTitleTT = 0
        globalTitleNP = 4
        globalTitle = ""
        # Separated TCAP buffer timeout in deci-seconds, 0 for no buffer, default 0
        erbBufferTimeout = 0
        cirBufferTimeout = 0
    }
```

Parameters

Parameters of the SmInapCA are listed below.

`maximumDurations`

Syntax:	`maximumDurations = [`
	`{ prefix = "0704", maxTime = 900 }`
	`{ prefix = "0701", maxTime = 9000 }`
	`]`

Description:	Gets the max duration .First gets the prefix for the number from the number tree, and then gets the max duration for the prefix from the number map.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	prefix "" or none maxTime 0 or infinite
Example:	`{ prefix = "0704", maxTime = 900 }`

`noAnswerTimeout`

Syntax:	`noAnswerTimeout = "`*`value`*`"`
Description:	no-answer timeout in seconds. This will be overridden by RRBCSM event.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	30 seconds
Example:	`noAnswerTimeout = 30`

`noAnswerTimeoutNetworkLatency`

Syntax:	`noAnswerTimeoutNetworkLatency = "value"`
Description:	no-answer timeout network latency in seconds will be added to the noAnswerTimeout taken from either configuration or from the RRBCSM event to allow the noAnswerTimeout on the switch to timeout before the noAnswerTimeout on the smInapCA adaptor.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	1 second
Example:	`noAnswerTimeoutNetworkLatency = 1`

globalActivityTestInterval

Syntax:	`globalActivityTestInterval = "value"`
Description:	Time interval between the Activity Test Response received from network and another Activity Test signal sent to network.
Type:	Integer
Optionality:	Mandatory
Allowed:	
Default:	100 deci-second
Example:	`globalActivityTestInterval = 100`

activityTestTimeout

Syntax:	`activityTestTimeout = "value"`
Description:	Sets the timeout interval for an Activity Test signal sent to network.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	50 deci-second
Example:	`activityTestTimeout = 50`

capV1Support

Syntax:	`capV1Support = value`
Description:	capV1Support will provide a check on RRBCSM events by allowing only Answer and Disconnect events.
Type:	Boolean
Optionality:	Optional
Allowed:	
Default:	true
Example:	`capV1Support = true`

appTimeout

Syntax:	`appTimeout = value`
Description:	Application request timeout in seconds.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	infinite (0 seconds)
Example:	`appTimeout = 0`

appExpiryReleaseCause

Syntax:	`appExpiryReleaseCause = value`
Description:	Application request timeout release cause sent to TCAP interface on timer expiry.
Type:	Integer
Optionality:	Optional

Allowed:
Default: 31 (normal)
Example: `appExpiryReleaseCause = 31`

achExpiryReleaseCause

Syntax: `achExpiryReleaseCause = value`
Description: ApplyCharging timer expiry release cause sent to TCAP interface on timer expiry.
Type: Integer
Optionality: Optional
Allowed:
Default: 31 (normal)
Example: `achExpiryReleaseCause = 31`

atExpiryReleaseCause

Syntax: `atExpiryReleaseCause = value`
Description: ActivityTest timer expiry release cause sent to TCAP interface on timer expiry.
Type: Integer
Optionality: Optional
Allowed:
Default: 31 (normal)
Example: `atExpiryReleaseCause = 31`

replaceZeroReleaseCause

Syntax: `replaceZeroReleaseCause = value`
Description: Zero release cause substitution, replace any network-bound release call zero cause with this value.
Type: Integer
Optionality: Optional
Allowed:
Default: 31 (normal)
Example: `replaceZeroReleaseCause = 31`

appServiceKey

Syntax: `appServiceKey = value`
Description: Application Service Key.
Type: Integer
Optionality: Optional
Allowed:
Default: 1 (CCS)
Example: `appServiceKey = 1`

hlcExtensionId

Syntax: `hlcExtensionId = value`
Description: High Layer Compatibility Extension Id.

Type:	Integer
Optionality:	Optional
Allowed:	
Default:	1 (Note: extension type: Asn1Integer)
Example:	`hlcExtensionId = 1`

pointCode

Syntax:	`pointCode = value`
Description:	Originating SCCP Point Code (PC).It may be overridden on command-line with '-pc' parameter.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	0 (unset)
Example:	`pointCode = 0`

subSystemNumber

Syntax:	`subSystemNumber = value`
Description:	Originating SCCP Sub-System Number (SSN).It may be overridden on command-line with '-ssn' parameter.
Type:	Integer
Optionality:	Optional
Allowed:	
Default:	0 (unset)
Example:	`subSystemNumber = 0`

globalTitleType

Syntax:	`globalTitleType = value`
Description:	Originating SCCP Global Title Type .It is used only if global title defined and may be overridden on command-line with '-gt_type' parameter.
Type:	Integer
Optionality:	Optional
Allowed:	Type 1 equals <noa>,<BCD address digits>" Type 2 equals <trans type><BCD address digits>" Type 3 equals <trans type>,<num plan>,<BCD address digits>" only BCD encoding scheme is supported Type 4 equals <trans type>,<num plan>,<noa>,<BCD address digits>"
Default:	4 (network-specific)
Example:	`globalTitleType = 4`

globalTitleNoA

Syntax:	`globalTitleNoA = value`
Description:	Originating SCCP Global Title Nature-of-Address (NoA).It is used only if global title defined and may be overridden on command-line with '-gt_noa' parameter.

Type: Integer
Optionality: Optional
Allowed:
Default: 0 (network-specific)
Example: `globalTitleNoA = 4`

globalTitleTT

Syntax: `globalTitleTT = value`
Description: Originating SCCP Global Title Trans-Type (TT).It is used only if global title is defined.It may be overridden on command-line with '-gt_tt' parameter.
Type: Integer
Optionality: Optional
Allowed:
Default: 0 (network-specific)
Example: `globalTitleTT = 0`

globalTitleNP

Syntax: `globalTitleNP = value`
Description: Originating SCCP Global Title Number Plan (NP).It is used only if global title is defined and may be overridden on command-line with '-gt_np' parameter.
Type: Integer
Optionality: Optional
Allowed:
Default: 4 (network-specific)
Example: `globalTitleNP = 1`

globalTitle

Syntax: `globalTitle = value`
Description: Originating SCCP Global Title (GT).It may be overridden on command-line with '-gt' parameter.
Type: Integer
Optionality: Optional
Allowed:
Default: "" (none)
Example: `globalTitle = ""`

erbBufferTimeout

Syntax: `erbBufferTimeout = value`
Description: Define the time in deci-seconds, the erb messages will be buffered, waiting for the expected associated message. After that time, buffered message will be sent anyway.
Type: Integer
Optionality: Optional
Allowed:
Default: 0(messages will not be buffered at all)

Example: `erbBufferTimeout = 0`

`cirBufferTimeout`

Syntax: `cirBufferTimeout = value`

Description: Define the time in deci-seconds, the cir messages will be buffered, waiting for the expected associated message. After that time, buffered message will be sent anyway.

Type: Integer

Optionality: Optional

Allowed:

Default: 0(messages will not be buffered at all)

Example: `cirBufferTimeout = 0`

Chapter 4

About Installation

Overview

Introduction

This chapter provides information about the installed components for the Oracle Communications Network Charging and Control (NCC) application described in this guide. It also lists the files installed by the application that you can check for, to ensure that the application installed successfully.

In this chapter

This chapter contains the following topics:

- Installation Overview
- Checking the Installation

Installation Overview

Introduction

For information about the following requirements and tasks, see Installation Guide:

- NCC system requirements
- Pre-installation tasks
- Installing and removing NCC packages

SMINAPCA Package

An installation of Oracle Communications Network Charging and Control SMINAP Control Agent includes the following package on the SLC:

- smInapCA

Checking the Installation

Introduction

Refer to the following checklists to ensure that SMINAPCA is installed correctly:

Checklist - SLC

Follow the steps in this checklist to ensure SMINAPCA is installed correctly on an SLC machine.

Step	Action
1	Log in to SLC machine as root.
2	Check that the following directory structure exists, with subdirectories: • **/IN/service_packages/SMINAPCA**

Step	Action
3	Check that directories contain subdirectories and that all are owned by: smf_oper user (group esg)

Process list - SLC

If the application is running correctly, the following process should be running on each SLC:

- Started during SLEE startup:
  - smInapCA

Chapter 5

SMINAP Control Agent Call Flows

Overview

Introduction

This chapter lists a sample set of SMINAPCA message flows.

In this chapter

This chapter contains the following topics:

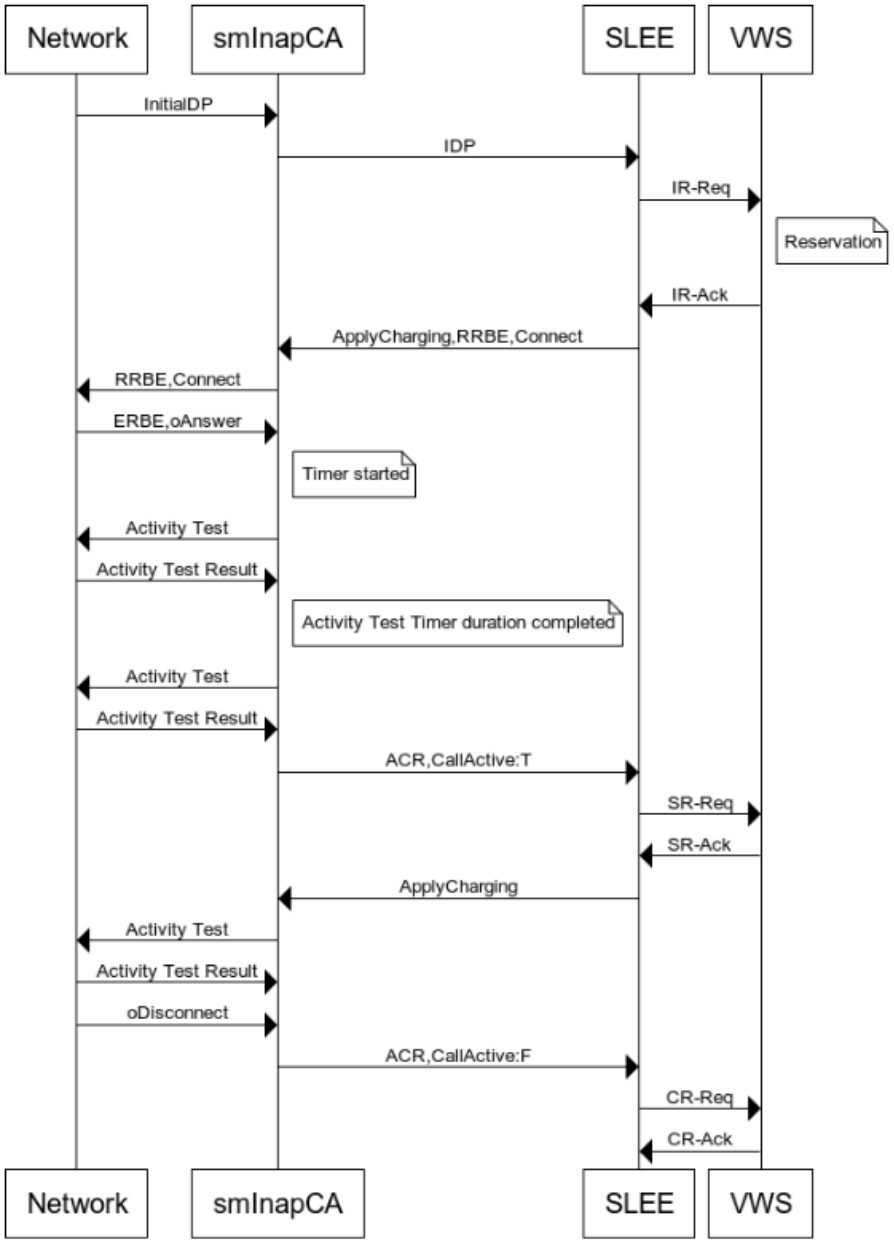
- Detailed Call Flow
- Message Flow between Network and SMINAPCA

Detailed Call Flow

This flow shows the message exchange between network, SMINAPCA, SLEE, and VWS.

The diagram shown below depicts a simple scenario where SMINAPCA receives an IDP from the network. NCC responds by sending a connect and then receives an oAnswer from network. The call is continued and AT/ATRs are exchanged between network and NCC. The call is disconnected by the called party and NCC responded with a Commit Reservation.

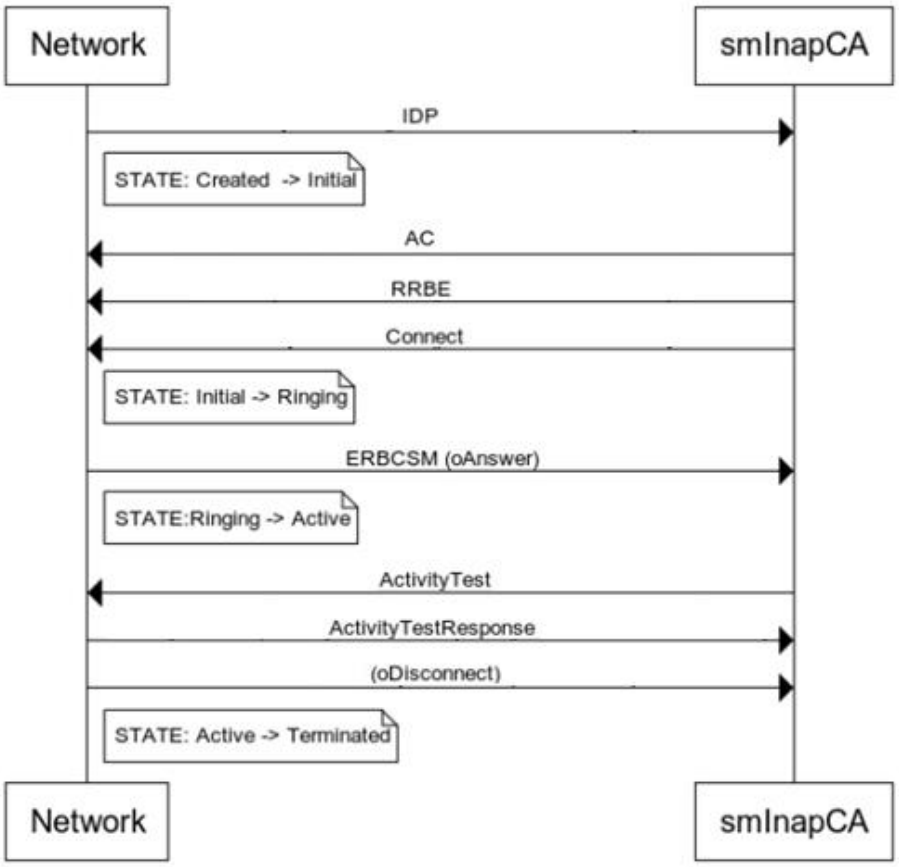
Note: For the simplicity of call flow, SIGTRAN layer is not shown. SIGTRAN is present in between Network and smInapCA. smInapCA is not directly communicating with the network.



Message Flow between Network and SMINAPCA

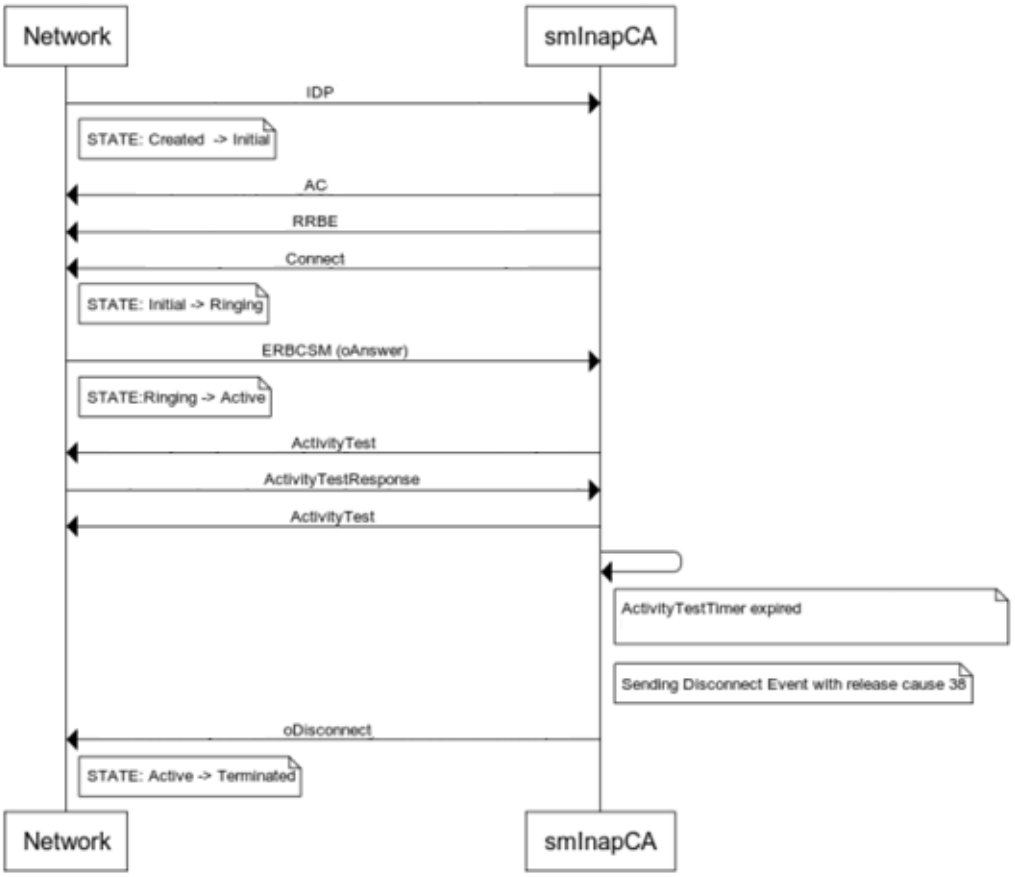
Case 1: When a call is disconnected by B-party

The sequence diagram given below is showing the message flow from Network to SMINAPCA. In a non-CAPv1 call flow, network communicates with NCC using IDP, AC, and ACR messages. Here we are providing CAPv1 support using the control agent which translates CAPv1 signals into CAPv3 and routes it to SLEE. The control agent (SMINAPCA) is converting the IDP, Activity Test, and Activity Test Result messages to INAP signals and is sending them to SLEE.



Case 2: When Activity Test timer expired

The diagram below describes the call flow between network and SMINAPCA when the activity test timer gets expired without receiving any Activity Test Result from network. The disconnect signal in this case is sent by NCC. The diagram is also depicting state change information during a call.


Network
smInapCA
IDP
STATE: Created -> Initial
AC
RRBE
Connect
STATE: Initial -> Ringing
ERBCSM (oAnswer)
STATE:Ringing -> Active
ActivityTest
ActivityTestResponse
ActivityTest
ActivityTestTimer expired
Sending Disconnect Event with release cause 38
oDisconnect
STATE: Active -> Terminated
Network
smInapCA