

Configuration Guide for Diameter

Ericsson Service-Aware Policy Controller

User Guide

Copyright

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document [Trademark Information](#).



Contents

1	Configuration Prerequisites	1
2	Diameter Configuration Overview	2
3	How To Apply Diameter Configuration Changes	3
4	Diameter Dictionary	4
4.1	How to Modify Diameter Dictionaries	4
4.2	Diameter Dictionary Format	5
5	Traffic Separation for Diameter Interfaces	8
6	Configure DRA Scenarios	9
6.1	Realm Routing Table	9
6.1.1	Troubleshooting	11
6.2	Configure Realm Routing Table as Diameter Server	11
6.3	Configure Realm Routing Table as Diameter Client	14
7	Configure Connections to OCS (Sy)	18
7.1	Direct Connection between the SAPC and the OCS with one realm	18
7.2	No Direct Connection between the SAPC and the OCS	20
7.3	Configure Connections to an OCS with Several Diameter Realms	22
7.4	Script to Automate Configuration Values	26
8	Configure SCTP Multihoming	28
8.1	Path Separation, Path Diversity and Multihoming	31
9	Reference List	32





1 Configuration Prerequisites

Before configuring the SAPC in an operational network, assure that:

- CBA Components are installed.
- The SAPC product software is installed.
- To have a detailed understanding of the function.



2 Diameter Configuration Overview

To make easier initial configuration and avoid manual procedures, the SAPC provides at installation time some Diameter data for the various diameter interfaces that supports (for example Gx, Rx, Sy):

- Product name, vendor identifier (Ericsson), and firmware.
- Capabilities exchange related data: supported application and vendor identifiers.
- Dictionaries containing Diameter messages and AVPs.
- Transports (both TCP and SCTP) to act as diameter server and accept incoming traffic in all the SAPC PLs (scalable), without the need of defining the client diameter peers.

Note: Transports to act as diameter client have to be set.

Apart of transports for connections at diameter level, depending on the diameter application it is needed to configure data at application level (for example, for Gx, PCEFs have to be set as `DiameterNode`, refer to Configuration Guide for Access and Charging Control (Gx))

Some other data are set during the SAPC deployment procedure, using Adapt Cluster Tool (see Customer and Diameter traffic VIP sections):

- Origin-Host
- Origin-Realm
- Virtual IP address and port for the different diameter interfaces



3 How To Apply Diameter Configuration Changes

To execute the Diameter configuration steps:

Steps

1. Log as sapcadmin in to a System Controller (SC):

```
ssh -X sapcadmin@<OAM VIP>
```

2. To add new configuration objects:

- a. Create an .xml file with the content of the corresponding objects and attribute values (taking as templates the examples provided in this document)
- b. Execute:

```
immcfg -f <filename>.xml
```

3. To change an attribute in an object, use **immcfg -a <attribute-name>=<value> .**

For example:

```
immcfg -a originRealm=operatorRealm.com  
otpdiaService=Pcrf,otpdiaProduct=SAPC"
```

For other configuration options, use **immcfg --help**.



4 Diameter Dictionary

Diameter dictionary files contain the commands and AVPs supported by the SAPC. There is one dictionary file for each Diameter application that the SAPC supports (for example Gx). Dictionaries are used for validations of incoming messages.

For further information of the supported messages and AVPs, and the errors that the SAPC may return, see the corresponding Interface Description document.

4.1 How to Modify Diameter Dictionaries

To modify the provided dictionary, adding or modifying AVPs, follow next steps:

Steps

1. Access the SAPC, executing following command:

```
<External_Machine>:$ ssh root@<VIP_OAM>
```
2. Access to any PL executing following command:

```
ssh root@PL-x
```
3. Change directory executing following command:

```
<PL-x>:# cd /storage/system/config/sapc/.
```
4. List available Diameter dictionaries executing following command:

```
<PL-x>:# /opt/diacc/bin/DiaDictManager list.
```
5. Select one of the available dictionaries listed in previous step and dump it to a file <<dictionary-name>>.dict executing the following command:

```
<PL-x>:# /opt/diacc/bin/DiaDictManager get <<dictionary-name>>.
```
6. Use any text editor to manipulate <<dictionary-name>>.dict according to it is wanted:

```
<PL-x>:# vi <<dictionary-name>>.dict.
```
7. Apply changes in the dictionary<<dictionary-name> executing the following command:

```
<PL-x>:# /opt/diacc/bin/DiaDictManager add <<dictionary-name>>.dict.
```




4.2 Diameter Dictionary Format

The Diameter dictionary is a file with different sections where each section starts with a tag and ends at the start of the next section or the end of file.

The tags consist on an ampersand character followed by a keyword.

The contents of the following tags can be modified to add new AVPs or to modify existing AVPs:

Table 1 Diameter Dictionary Section Tags

Section Tag	Description
@avp_vendor_id	<p>List the AVPs supported by the SAPC for Vendor - Ids 5535 (3GPP2) and 193 (Ericsson).</p> <p>Do not include AVPs for Vendor - Id 10415 (3GPP) as they are considered as default.</p> <p>Format:</p> <pre>@avp_vendor_id <Number></pre> <p>Example:</p> <pre>@avp_vendor_id 193 Customer-Id Gx-Capability-List Rule-Space-Decision Rule-Space-Suggestion Charging-Rule-Authorization Authorization-State One-Time-Redirect-Control Content-Filtering-Profile-Id</pre>
@avp_types	<p>List the supported AVPs with the following fields:</p> <ul style="list-style-type: none"> —Name: A string with the AVP name. —Code: An integer number with the AVP code. —Type: A string with the AVP data format according to . —Flags: A string with the characters V, when Vendor - Id is present and M when the AVP is mandatory, or a single - character if no AVP flag. <p>Format:</p> <pre>@avp_types <Name> <Code> <Type> <Flags></pre> <p>Example:</p>



	<pre>@avp_types Bearer-Usage 1000 Enumerated MV CC-Input-Octets 412 Unsigned64 - CC-Output-Octets 414 Unsigned64 - Flows 510 Grouped MV Monitoring-Key 1066 OctetString V RAT-Type 1032 Enumerated V</pre>
@messages	<p>Defines the messages supported by the SAPC and the composition of AVPs inside each message.</p> <p>When a new AVP is defined in the dictionary, include the AVP in the corresponding message.</p> <p>Example:</p> <pre>CC-Request ::= < Diameter Header: 272, REQ, PXY > < Session-Id > { Auth-Application-Id } { Origin-Host } { Origin-Realm } { Destination-Realm } { CC-Request-Type } { CC-Request-Number } [Destination-Host] [Origin-State-Id] *[Subscription-Id] *[Supported-Features]</pre>
@grouped	<p>Defines the contents of the grouped AVPs.</p> <p>Example:</p> <pre>Usage-Monitoring-Information ::= < AVP Header: 1067 > [Monitoring-Key] [Granted-Service-Unit] [Used-Service-Unit] [Usage-Monitoring-Level] [Usage-Monitoring-Report] [Usage-Monitoring-Support] *[AVP]</pre>
@enum Name	<p>Defines the values of an enumerated AVP. The content consists on the names and the corresponding integer values.</p> <p>Example:</p> <pre>@enum Bearer-Operation TERMINATION 0 ESTABLISHMENT 1 MODIFICATION 2</pre>



@define Name	<p>Defines the possible values of a certain AVP. The content consists on the names and the corresponding integer values.</p> <p>Example:</p> <pre>@define Bearer-Control-Options BCO_DOWNGRADE 1 BCO_UPGRADE 2 BCO_EARP 4 BCO_REJECT 8</pre>
--------------	---



5 Traffic Separation for Diameter Interfaces

It is possible to define different traffic VIPs for different Diameter interfaces (for example, Gx and Rx), according to customer needs.

To apply traffic separation, follow the procedure described in the customization Adapt Cluster Tool, see **Traffic VIP Networks** section.



6 Configure DRA Scenarios

These scenarios are considered when the SAPC is not directly connected to the diameter peer. In case the SAPC is directly connected to the diameter peer see [Diameter Configuration Overview](#) on page 2.

6.1 Realm Routing Table

It is not needed to configure in the SAPC the routing table for incoming traffic, as the DRA peer includes the Destination-Host AVP in the diameter messages (set to the SAPC diameter identity).

The following configuration objects are related to route outgoing messages, when there is no direct connection between the SAPC and the peer:

Table 2 Diameter Routing Table Data

Object	Description	
OtpdiaSelector	<p>Mapping of Destination-Host/Realm in an outgoing request to a next-hop Origin-Host/Realm.</p> <p>The service (for example PCRF or Sy), application id (for example Gx, Rx), and destination of an outgoing request are matched against configured selector objects to determine how to route it.</p> <p>If there are multiple matches, the order in which the mappings are applied to find an appropriate target peer is random, except that more specific matches are preferred to less specific ones, that is:</p> <ul style="list-style-type: none"> —Destination-Host has precedence over —Destination-Realm, has precedence over —Application identifier <p>If any OtpdiaDomain in the list specifies more than one host attribute, the order in which these are selected is undefined. If a given host/realm matches multiple peer connections, one is chosen randomly.</p>	
	Attributes	
	service Mandatory Multivalued string	DN pointing to an OtpdiaService, to which the routing applies.



Object	Description	
	applicationId Optional Multivalued integer 32.	Application id to which the routing applies. Applies to all applications if unspecified.
	destination Multivalued Optional	DN of an object of class OtpdiaDomain, specifying the Destination-Realm and (optionally) Destination-Host values to which the routing applies. When it is not set, applies to all destinations.
	peer	Represents the next hop Diameter peer with direct connection to the SAPC. DN pointing to an OtpdiaCons, whose heads are OtpdiaDomain DNs specifying the Origin-Realm and (optionally) Origin-Host values of peers to send a request matching the service, applicationId, and destination. For a given selector object, the list of OtpdiaDomain instances pointed by the OtpdiaCons defined in the peer attribute are sorted in order of preference.
OtpdiaDomain	Represents some or all peers in a realm. It is used as both the domain and range of the mapping provided by OtpdiaSelector.	
	Attributes	
	realm Mandatory String	Destination-Realm
	host Optional String	Hostname inside the realm. No value represents all hosts in the realm.
OtpdiaCons	List to ordered objects	
	Attributes	
	head	DN of an element of the list.
	tail	DN pointing to an OtpdiaCons, the remaining tail of the list. The last element of a list does not set a tail.



6.1.1 Troubleshooting

Problem: not being able to route an outgoing diameter message.

Trouble symptoms: find inside trace messages, Error = <19>. Example:

```
{ file = "IncomingMsgCallbacks.cc", function = "otpdiaHandleError" →
", line = 128, msg = "Received an error on an outgoing request. E →
rror = <19>" }
...
{ file = "IncreaseGxRaaCounterTask.cc", function = "run", line = →
86, msg = "Error sending RAR. SessionId: c5-192-168-243-66-epg801 →
.epc.mnc001.mcc228.3gppnetwork.org;1482417902;86, Protocol: Gx" } →
...
msg = "Sent outgoing message of type: 'INTERNAL_UPDATE'" }
msg = "Received an error on an outgoing request. Error = <19>" }
msg = "Error sending RAR. SessionId: c5-192-168-243-66-epg801.ep →
c.mnc001.mcc228.3gppnetwork.org;1482417902;14000, Protocol: Gx" }
```

Cause: incorrect or missing configuration of the diameter routing table.

Solution: set properly the routing table.

6.2 Configure Realm Routing Table as Diameter Server

Figure 1 shows an example where PCEF and AF are connected to the SAPC by an external DRA function. The Diameter Routing Agent (DRA) function concentrates all Gx and Rx signaling from the PCEF and AF. The SAPC has not direct peer connections (CER/CEA) to the end application nodes, but to the DRA. The diameter peer connection between the SAPC and the DRA is used by both Gx and Rx applications.

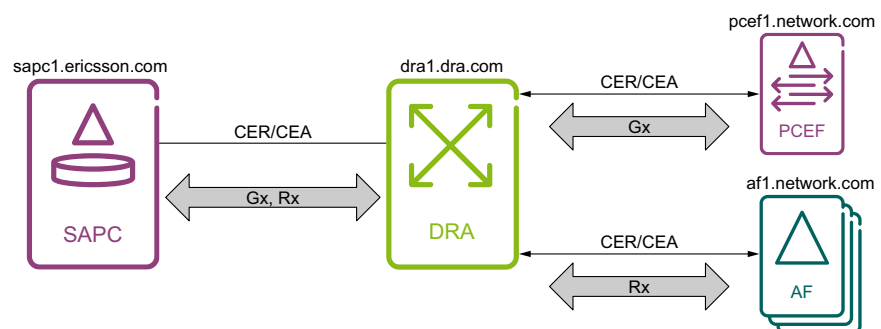


Figure 1 The SAPC Interworking with an External DRA for Gx and Rx

Steps

To configure in the SAPC this scenario:



1. Configure the Realm Routing Table for the applicable traffic:

- a. Create an `OtpdiaSelector` object for service, applicationId and destination.

The `peer` attribute points to a list directly connected peers. If all the peers belong to the same realm, the list may consist of one `OtpdiaCons` object, pointing to a `OtpdiaDomain` object with the realm.

The `destination` attribute points to a `OtpdiaDomain` object with the final Destination-Realm for the outgoing messages. It is not needed when the selector applies to all destinations.

- b. Define an `OtpdiaDomain` object containing the Destination-Realm value.
- c. Define an `OtpdiaCons` object for the `OtpdiaDomain` object that is directly connected to the SAPC (in our example, DRA). The distinguish name for the DRA in the `OtpdiaDomain` and the head attribute in the `OtpdiaCons` must match.

2. Proceed as [How To Apply Diameter Configuration Changes](#) on page 3

3. Configure Gx PCEF's controls, using the PCEF Origin-Host value as `DiameterNode` (refer to [Configuration Guide for Access, Charging and QoS Control \(Gx\)](#)).

Note: It is not needed to configure as transports the PCEF and AF nodes in the SAPC.

[Example 1](#), [Example 2](#) and [Example 3](#) show the routing table for Pcrf (Gx and Rx traffic) service (provided at installation time), assuming the following data:

- Destination-Realm for outgoing messages (Origin-Realm of PCEF's and AFs) = network.com
- dra1.dra.com is the Origin-Host corresponding to the DRA peer (direct connection to the SAPC).

Example 1 Routing Table for DRA for Gx and Rx

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xsi:schemaLocation="http://www.saforum.org/IMMSchema SAI-AIS-IMM-XSD-A.01.01.xsd">

  <object class="OtpdiaSelector">
    <dn>otpdiaSelector=PcrfDrt,otpdiaProduct=SAPC</dn>
    <attr>
```




```

        <name>service</name>
        <value>otpdiaService=Pcrf,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
        <name>peer</name>
        <value>otpdiaCons=PcrfPeer,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
        <name>destination</name>
        <value>otpdiaDomain=GxRxClients,otpdiaProduct=SAPC</value>
    </attr>
</object>
</imm:IMM-contents>

```

Example 2 Routing Table for DRA in Gx and Rx, Realms

```

<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
    <object class="OtpdiaDomain">
        <dn>otpdiaDomain=GxRxDRA,otpdiaProduct=SAPC</dn>
        <attr>
            <name>host</name>
            <value>dra1.dra.com</value>
        </attr>
        <attr>
            <name>realm</name>
            <value>dra.com</value>
        </attr>
    </object>
    <object class="OtpdiaDomain">
        <dn>otpdiaDomain=GxRxClients,otpdiaProduct=SAPC</dn>
        <attr>
            <name>realm</name>
            <value>network.com</value>
        </attr>
    </object>
</imm:IMM-contents>

```

Example 3 DRA Peer for Gx and Rx

```

<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">

```

```
<object class="OtpdiaCons">
  <dn>otpdiaCons=PcrfPeer,otpdiaProduct=SAPC</dn>
  <attr>
    <name>head</name>
    <value>otpdiaDomain=GxRxDRAs,otpdiaProduct=SAPC</value>
  </attr>
</object>
</imm:IMM-contents>
```

6.3 Configure Realm Routing Table as Diameter Client

Figure 2 shows an example where the SAPC is connected to an OCS with an intermediate DRA for Sy traffic.

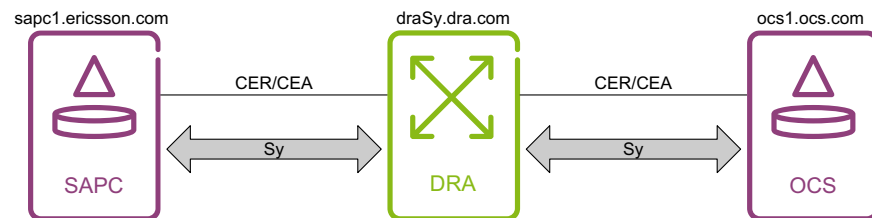


Figure 2 No Direct Connection Between the SAPC and an OCS

To configure the routing table in the SAPC acting as client to not directly connected peers:

Steps

1. Configure the peer that is directly connected to the SAPC using `OtpdiaHost` and `OtpdiaTransportTcp` objects (or `OtpdiaTransportSctpE`)
2. Define the peer directly connected to the SAPC in an `OtpdiaDomain` object; host attribute must match with the value given in `otpdiaHost` configured in [Step 1 in Section 7.2](#)
3. Set the routing table for `PcrfSy OtpdiaService`
 - a. Create an `OtpdiaSelector` object for `PcrfSy` service.
 - b. Define an `OtpdiaDomain` object, which destination `Destination-Realm` value of the OCS.
 - c. Define a `OtpdiaCons` object for the `OtpdiaDomain` object that is directly connected (DRA) to the SAPC. The distinguish name for the DRA in the `OtpdiaDomain` definition and the value for head attribute of `OtpdiaCons` must match.
4. Proceed as [How To Apply Diameter Configuration Changes](#) on page 3



[Example 4](#), [Example 5](#), [Example 6](#), [Example 7](#) and [Example 8](#) show the configuration for Sy traffic when a DRA is between the SAPC and the OCS:

Example 4 Sy not direct connection to OCS, Host

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instances"
                  xsi:schemaLocation="http://www.saforum.org/IMMSchema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaHost">
    <dn>otpdiaHost=SyDRA,otpdiaService=PcrfSy,otpdiaProduct=SAPC</dn>
    <attr>
      <name>address</name>
      <value>192.168.14.42</value>
    </attr>
    <attr>
      <name>port</name>
      <value>13868</value>
    </attr>
  </object>
</imm:IMM-contents>
```

Example 5 Sy not direct connection to OCS, Transport

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instances"
                  xsi:schemaLocation="http://www.saforum.org/IMMSchema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaTransportTcp">
    <dn>otpdiaTransportTcp=SyDRA_transport,otpdiaService=PcrfSy,otpdiaProduct=SAPC</dn>
    <attr>
      <name>address</name>
      <value>192.168.12.40</value>
    </attr>
    <attr>
      <name>port</name>
      <value>0</value>
    </attr>
    <attr>
      <name>host</name>
      <value>:all</value>
    </attr>
    <attr>
      <name>connectTo</name>
    </attr>
  </object>
</imm:IMM-contents>
```



```
        <value>otpdiaHost=SyDRA,otpdiaService=PcrfSy</value>
      </attr>
    </object>
  </imm:IMM-contents>
```

Example 6 Sy not direct connection to OCS, Routing Table

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaSelector">
    <dn>otpdiaCons=SyDrt,otpdiaProduct=SAPC</dn>
    <attr>
      <name>service</name>
      <value>otpdiaService=PcrfSy,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
      <name>destination</name>
      <value>otpdiaDomain=OCS,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
      <name>peer</name>
      <value>otpdiaCons=SyPeer,otpdiaProduct=SAPC</value>
    </attr>
  </object>
</imm:IMM-contents>
```

Example 7 Sy not direct connection to OCS, Realms

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaDomain">
    <dn>otpdiaDomain=SyDRA,otpdiaProduct=SAPC</dn>
    <attr>
      <name>host</name>
      <value>draSy.dra.com</value>
    </attr>
    <attr>
      <name>realm</name>
      <value>dra.com</value>
    </attr>
  </object>
  <object class="OtpdiaDomain">
```



```

<dn>otpdiaDomain=OCS,otpdiaProduct=SAPC</dn>
<attr>
  <name>realm</name>
  <value>ocs.com</value>
</attr>
</object>
</imm:IMM-contents>

```

Example 8 Sy not direct connection to OCS, DRA Peer

```

<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaCons">
    <dn>otpdiaCons=SyPeer,otpdiaProduct=SAPC</dn>
    <attr>
      <name>head</name>
      <value>otpdiaDomain=SyDRA,otpdiaProduct=SAPC</value>
    </attr>
  </object>
</imm:IMM-contents>

```

7 Configure Connections to OCS (Sy)

In addition to configure the SAPC as a Gx server, the SAPC can act as an Sy client. In this case, the SAPC is connected to an OCS directly or by means of a DRA.

Note: For additional information regarding Sy interface, refer to [Configuration Guide for Integration with OCS for Spending Limit Reporting \(Sy\)](#).

7.1 Direct Connection between the SAPC and the OCS with one realm

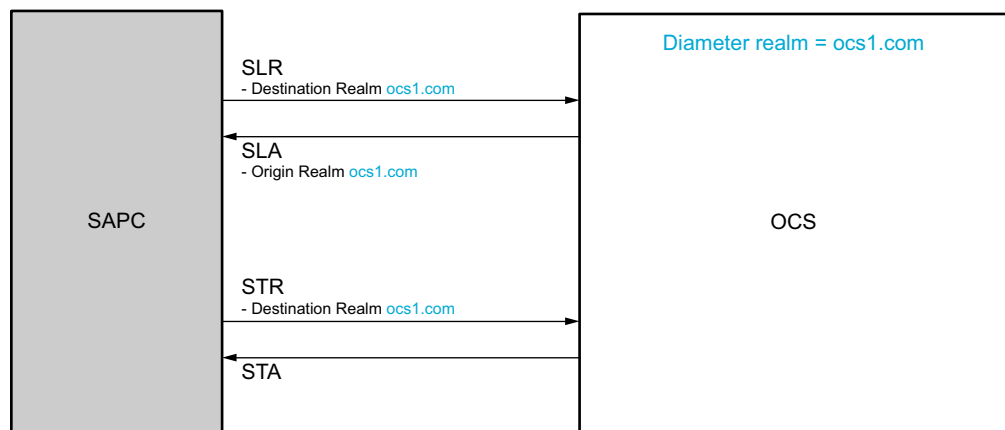


Figure 3 Direct connection to an OCS having only a realm

Steps

For each OCS:

1. Create an `OtpdiaHost` object under RDN
`otpdiaService=PcrfSy,otpdiaProduct=SAPC`
2. Enable the SAPC to act as Sy client creating either the `OtpdiaTransportTcp` or `OtpdiaTransportSctpE` object for all PLs using `:all` value in host attribute.

Note: When the `:all` value in host attribute is used, that means that all the PLs will act as diameter clients for Sy, therefore, SAPC will open as many diameter connections (CER/CEA) as there are PLs in the SAPC cluster.

3. Proceed as [How To Apply Diameter Configuration Changes](#) on page 3



Example 9 Configuration of the SAPC as Sy Diameter Client to direct OCS Peers

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaHost">
    <dn>otpdiaHost=OCS_1,otpdiaService=PcrfSy,otpdiaProduct=SAPC<
/dn>
    <attr>
      <name>address</name>
      <value>192.168.14.42</value>
    </attr>
    <attr>
      <name>port</name>
      <value>13868</value>
    </attr>
  </object>
  <object class="OtpdiaTransportTcp">
    <dn>otpdiaTransportTcp=OCS_1_transport,otpdiaService=PcrfSy,o
tpdiaProduct=SAPC</dn>
    <attr>
      <name>address</name>
      <value>192.168.12.40</value>
    </attr>
    <attr>
      <name>port</name>
      <value>0</value>
    </attr>
    <attr>
      <name>host</name>
      <value>:all</value>
    </attr>
    <attr>
      <name>connectTo</name>
      <value>otpdiaHost=OCS_1,otpdiaService=PcrfSy</value>
    </attr>
  </object>
</imm:IMM-contents>
```

Example 9 configures the Sy traffic for an Online Charging System called "OCS_1" connected to the SAPC. In the `otpdiaTransportTcp` object, address and port attributes contain the local traffic IP and the port of the SAPC node (in this case, the IP is 192.168.12.40 and the local port 0) whereas in the `otpdiaHost` object address and port attributes contain the IP and the port traffic for the OCS. The `connectTo` attribute in `otpdiaTransportTcp` object references the "OCS_1" OCS configured in `otpdiaHost`.

Note: To get the IP of the SAPC execute `immlist -a address otpdiaTransportTcp=:all,otpdiaService=Pcrf,otpdiaProduct=SAPC` from the SC.

7.2 No Direct Connection between the SAPC and the OCS

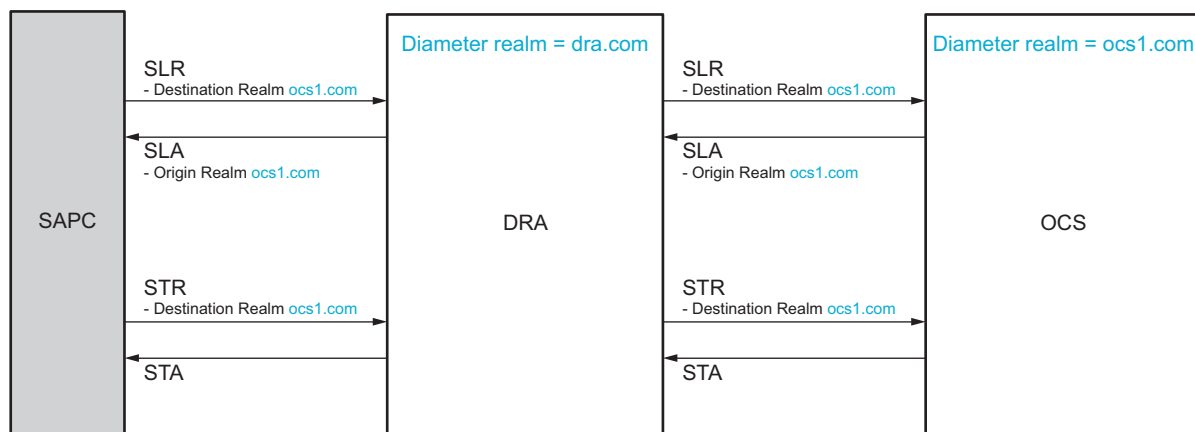


Figure 4 The SAPC is not directly connected to the OCS.

To configure the Routing Table in the SAPC for not directly connected peers (see [Example 10](#)):

Steps

1. Configure the peer that is directly connected to the SAPC using `OtpdiaHost` and `OtpdiaTransportTcp` objects (or `OtpdiaTransportSctpE`) as described in [Direct Connection between the SAPC and the OCS with one realm](#) on page 18
2. Define an `OtpdiaDomain` object with an identifier (in our example, OCS) to the destination OCS and realm value matching the one configured in the `serverRealm` attribute in the `/profiles/online-charging-system/<profileId>` URI in the provisioning REST API.
3. Define the peer directly connected to the SAPC in an `OtpdiaDomain` object with an identifier (in our example, DRA). Host attribute must match with the value given in `otpdiaHost` configured in [Step 1](#)
4. Define a connection in an `OtpdiaCons` object for the `OtpdiaDomain` object that is directly connected to the SAPC (in our example, DRA). The distinguish name for the DRA in the `OtpdiaDomain` definition and the value for `head` attribute in the connection definition must match
5. To set a route (mapping of Destination-Host/Realm for an outgoing Request to a next-hop Origin-Host/Realm), create an `OtpdiaSelector` object.



6. Proceed as [How To Apply Diameter Configuration Changes](#) on page 3

Example 10 Sy configuration for not direct connection to OCS

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instances"
                  xsi:schemaLocation="http://www.saforum.org/IMMSchema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaHost">
    <dn>otpdiaHost=DRA,otpdiaService=PcrfSy,otpdiaProduct=SAPC</dn>
    <attr>
      <name>address</name>
      <value>192.168.14.42</value>
    </attr>
    <attr>
      <name>port</name>
      <value>13868</value>
    </attr>
  </object>
  <object class="OtpdiaTransportTcp">
    <dn>otpdiaTransportTcp=DRA_transport,otpdiaService=PcrfSy,otpdiaProduct=SAPC</dn>
    <attr>
      <name>address</name>
      <value>192.168.12.40</value>
    </attr>
    <attr>
      <name>port</name>
      <value>0</value>
    </attr>
    <attr>
      <name>host</name>
      <value>:all</value>
    </attr>
    <attr>
      <name>connectTo</name>
      <value>otpdiaHost=DRA,otpdiaService=PcrfSy</value>
    </attr>
  </object>
  <object class="OtpdiaDomain">
    <dn>otpdiaDomain=OCS,otpdiaProduct=SAPC</dn>
    <attr>
      <name>realm</name>
      <value>ocs1.com</value>
    </attr>
  </object>
  <object class="OtpdiaDomain">
    <dn>otpdiaDomain=DRA,otpdiaProduct=SAPC</dn>
```

```
<attr>
  <name>host</name>
  <value>dra123.dra.com</value>
</attr>
<attr>
  <name>realm</name>
  <value>dra.com</value>
</attr>
</object>
<object class="OtpdiaCons">
  <dn>otpdiaCons=Cons1,otpdiaProduct=SAPC</dn>
  <attr>
    <name>head</name>
    <value>otpdiaDomain=DRA,otpdiaProduct=SAPC</value>
  </attr>
</object>
<object class="OtpdiaSelector">
  <dn>otpdiaCons=Selector1,otpdiaProduct=SAPC</dn>
  <attr>
    <name>service</name>
    <value>otpdiaService=PcrfSy,otpdiaProduct=SAPC</value>
  </attr>
  <attr>
    <name>destination</name>
    <value>otpdiaDomain=OCS,otpdiaProduct=SAPC</value>
  </attr>
  <attr>
    <name>peer</name>
    <value>otpdiaCons=Cons1,otpdiaProduct=SAPC</value>
  </attr>
</object>
</imm:IMM-content>
```

Example 10 Configures a route to an OCS in realm ocs1.com through realm dra.com.

7.3 Configure Connections to an OCS with Several Diameter Realms

This is a network deployment in which a logical Charging System exposes different Diameter realms to the SAPC.

Figure 5 shows an example where the SAPC sends an SLR to ocs1.com realm, and the OCS internally decides that such Sy charging session has to be routed to sdp2.ocs1.com. Afterwards, the STR sent from the SAPC to the Charging System, has to reach the same sdp2.ocs1.com realm.

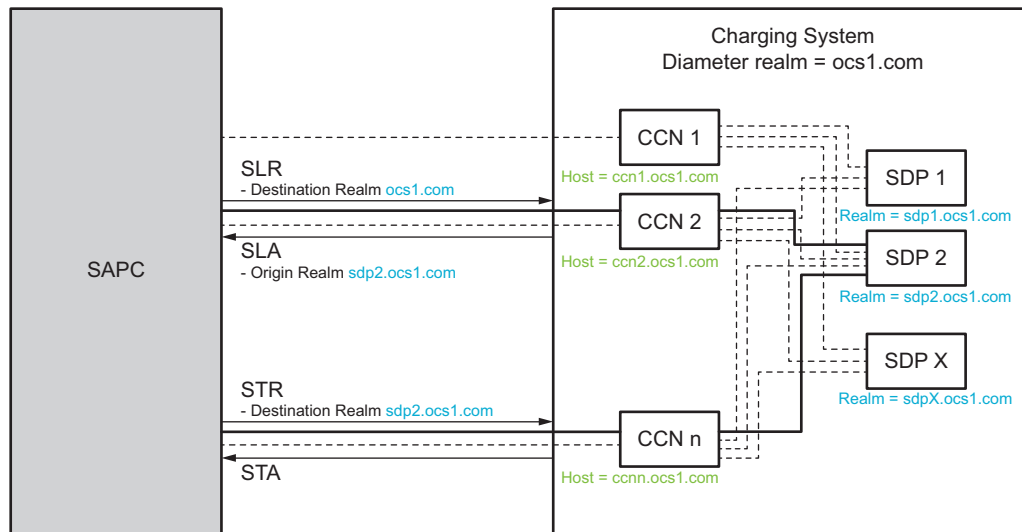


Figure 5 Several Sy Diameter Realms

Steps

1. Configure the peers that are connected directly to the SAPC using `OtpdiaHost` and `OtpdiaTransportTcp` objects (or `OtpdiaTransportSctpE`) as described in [Direct Connection between the SAPC and the OCS with one realm](#) on page 18

2. Define one `OtpdiaDomain` for each node (peers and hosts) present in the topology that can receive/send a Sy message from/to the SAPC.

Example 11. Defines an internal identifier (in our example, OCS) whose realm is the one declared in the SAPC internal database in `serverRealm` attribute in the `/profiles/online-charging-system/<profileId>` URI in the provisioning REST API and is not connected directly to the SAPC.

Example 12. Defines an internal identifier (in our example, CCN) for each peer that is directly connected to the SAPC. Host attribute must match with the value given in `otpDiaHost` configured in [Step 1](#)

Example 13. Defines an internal identifier (in our example, SDP) for each host that can send messages to the SAPC and are not connected directly.

3. Define a connection in an `OtpdiaCons` object just for each `OtpdiaDomain` object that is directly connected to the SAPC. In our example, the ones called CCNs.

This connection is identified by its distinguish name. The distinguish name for the CCN definition and the value for head attribute in the connection definition must match. See [Example 14](#)

4. To set a route, create an `OtpdiaSelector` object with a distinguish name for each host not directly connected. This object includes a destination attribute with the `otpdiaDomain` of the destination host defined in [Step 2](#), and a peer attribute with the `OtpdiaCons` connection to the peer defined in [Step 3](#). See [Example 15](#). The route states that to get OCS and SDP1, Sy message is sent to Cons1 (CCN1) as first step

Example 11 Definition of OCS realm

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaDomain">
    <dn>otpdiaDomain=OCS,otpdiaProduct=SAPC</dn>
    <attr>
      <name>realm</name>
      <value>ocs1.com</value>
    </attr>
  </object>
</imm:IMM-contents>
```

Example 12 Definition of CCNs

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
tance"
                  xsi:schemaLocation="http://www.saforum.org/IMMS
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaDomain">
    <dn>otpdiaDomain=CCN1,otpdiaProduct=SAPC</dn>
    <attr>
      <name>host</name>
      <value>ccn1.ocs1.com</value>
    </attr>
    <attr>
      <name>realm</name>
      <value>ocs1.com</value>
    </attr>
  </object>
</imm:IMM-contents>
```

Example 13 Definition of SDPs

```
<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
ins">
```



```

tance"
        xsi:schemaLocation="http://www.saforum.org/IMMS →
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaDomain">
    <dn>otpdiaDomain=SDP1,otpdiaProduct=SAPC</dn>
    <attr>
      <name>host</name>
      <value>sdp1.ocs1.com</value>
    </attr>
    <attr>
      <name>realm</name>
      <value>ocs1.com</value>
    </attr>
  </object>
</imm:IMM-contents>

```

Example 14 Definition of Connections

```

<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins →
tance"
        xsi:schemaLocation="http://www.saforum.org/IMMS →
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaCons">
    <dn>otpdiaCons=Cons1,otpdiaProduct=SAPC</dn>
    <attr>
      <name>head</name>
      <value>otpdiaDomain=CCN1,otpdiaProduct=SAPC</value>
    </attr>
  </object>
</imm:IMM-contents>

```

Example 15 Definition of Topology

```

<imm:IMM-contents xmlns:imm="http://www.saforum.org/IMMSchema"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins →
tance"
        xsi:schemaLocation="http://www.saforum.org/IMMS →
chema SAI-AIS-IMM-XSD-A.01.01.xsd">
  <object class="OtpdiaSelector">
    <dn>otpdiaCons=Selector1,otpdiaProduct=SAPC</dn>
    <attr>
      <name>service</name>
      <value>otpdiaService=PcrfSy,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
      <name>destination</name>
      <value>otpdiaDomain=OCS,otpdiaProduct=SAPC</value>
    </attr>
  </object>
</imm:IMM-contents>

```



```

    </attr>
    <attr>
      <name>peer</name>
      <value>otpdiaCons=Cons1,otpdiaProduct=SAPC</value>
    </attr>
  </object>

  <object class="OtpdiaSelector">
    <dn>otpdiaCons=Selector2,otpdiaProduct=SAPC</dn>
    <attr>
      <name>service</name>
      <value>otpdiaService=PcrfSy,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
      <name>destination</name>
      <value>otpdiaDomain=SDP1,otpdiaProduct=SAPC</value>
    </attr>
    <attr>
      <name>peer</name>
      <value>otpdiaCons=Cons1,otpdiaProduct=SAPC</value>
    </attr>
  </object>

</imm:IMM-contents>

```

7.4 Script to Automate Configuration Values

Preparing the configuration for the Diameter Routing Table in a deployment as [Figure 5](#) with *n* CCNs and *x* SDPs involves the declaration of many objects and can lead to errors. To ease complex configurations, the SAPC provides a script to generate the XML files. To run this script:

Steps

1. Log as sapcadmin into a System Controller (SC):
2. Generate an XML file named input_file.txt as [Example 16](#) with the definition of attributes related to each OCS, CCN, and SDP
3. Run the following command:

```
generateRoutingTableOCS --config=input_file.txt --populate
```

Note: The option `--populate` applies the output XML files to the SAPC directly

Example 16 XML example for script input

```
<config>
```



```
<ocs>
  <otpdiaDomain>OCS</otpdiaDomain>
  <realm>ocsrealm.com</realm>
</ocs>
<ccns>
  <ccn>
    <otpdiaDomain>CCN1</otpdiaDomain>
    <host>ccn1.realm0</host>
    <realm>realm0</realm>
    <port>13222</port>
    <address>172.16.2.21</address>
  </ccn>
  <ccn>
    <otpdiaDomain>CCN2</otpdiaDomain>
    <host>ccn2.realm0</host>
    <realm>realm0</realm>
    <port>13223</port>
    <address>172.16.2.21</address>
  </ccn>
</ccns>
<sdps>
  <sdp>
    <otpdiaDomain>SDP1</otpdiaDomain>
    <host>sdp1.realm1</host>
    <realm>realm1</realm>
  </sdp>
  <sdp>
    <otpdiaDomain>SDP2</otpdiaDomain>
    <host>sdp2.realm2</host>
    <realm>realm2</realm>
  </sdp>
  <sdp>
    <otpdiaDomain>SDP3</otpdiaDomain>
    <host>sdp3.realm3</host>
    <realm>realm3</realm>
  </sdp>
  <sdp>
    <otpdiaDomain>SDP4</otpdiaDomain>
    <host>sdp4.realm4</host>
    <realm>realm4</realm>
  </sdp>
  <sdp>
    <otpdiaDomain>SDP5</otpdiaDomain>
    <host>sdp5.realm5</host>
    <realm>realm5</realm>
  </sdp>
</sdps>
</config>
```



8 Configure SCTP Multihoming

The SAPC provides configuration for SCTP transport at installation time, by default listening to a single VIP. To use multihoming, follow next steps:

Steps

1. Log as sapcadmin in to a System Controller (SC):

```
ssh -X sapcadmin@<OAM VIP>
```

2. Configure new VIPs and flow policies in eVIP

- a. Find the preinstalled ALB used to handle SCTP traffic:

```
SC-1:~ # immfind -c EvipFlowPolicy | grep -i sctp
```

Example for a configured SAPC:

```
evipFlowPolicyId=SCTP_diameter, evipFlowPoliciesId=1, evi
pAlbId=alb_trf, evipAlbsId=1, evipId=1, thus alb_trf will be
used in the next steps
```

- b. Open an Ericsson Command-Line Interface (CLI) session and enter in configure mode.

```
>configure
```

- c. Navigate to the ALB which handles SCTP traffic:

```
(config)> dn
ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1, EvipAlb=
<alb_trf>
```

- d. Add VIP needed in SCTP multihoming

```
(config-EvipAlb=<alb_trf>)> EvipVips=1, EvipVip=<SCTP_MULTIHOMI
NG_1>
(config-EvipVip=<SCTP_MULTIHOMING_1>)> address=<VIP1>
```

Where:

- <SCTP_MULTIHOMING_1> must be a string
- <VIP1> must be an IP address

- e. Return to parent DN

```
(config-EvipVip=<VIP1>)> dn
ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1, EvipAlb=
<alb_trf>
```




- f. Add flow policy to route the incoming traffic for the new VIP

```
(config-EvipAlb=<alb_trf>)>
EvipFlowPolicies=1,EvipFlowPolicy=<SCTP_diameter_2>

(config-EvipFlowPolicy=<SCTP_diameter_2>)>
addressFamily="ipv4"

(config-EvipFlowPolicy=<SCTP_diameter_2>)>
dest="<SCTP_MULTIHOMING_1>"

(config-EvipFlowPolicy=<SCTP_diameter_2>)>
protocol="sctp"

(config-EvipFlowPolicy=<SCTP_diameter_2>)>
soGrp="1011250"
```

- g. Repeat sub-steps c, d, e and f for all the required VIPs in SCTP multihoming
- h. Commit the changes and check the active status of newly created eVIP elements

```
(config-EvipAlb=<alb_trf>)> commit

(EvipAlb=<alb_trf>)> show all
EvipFlowPolicies=1,EvipFlowPolicy=<SCTP_diameter_2>

EvipFlowPolicy=<SCTP_diameter_2>

addressFamily="ipv4"

dest="<SCTP_MULTIHOMING_1>"

protocol="sctp"

soGrp="1011250"

usageState=ACTIVE

(EvipAlb=<alb_trf>)> show all
EvipVips=1,EvipVip=<SCTP_MULTIHOMING_1>

EvipVip=<SCTP_MULTIHOMING_1>

address="<VIP1>"

state="ACTIVE"
```

3. Configure diameter SCTP transports

- a. Find the preinstalled OtpdiaTransportSctpE transport:

```
SC-1:~ # immfind | grep SctpE
```



Example for PCRF (Gx and Rx) service

```
otpdiaTransportSctpE=<traffic-  
vip>:all,otpdiaService=Pcrf,otpdiaProduct=SAPC
```

- b. Add all IP addresses to the `address` attribute in `otpdiaTransportSctpE` transport:

```
SC-1:~ # immcfg -a address+="<VIP1>" -a address  
+="<VIP2>" -a address+="<VIP3>"  
otpdiaTransportSctpE=SCTP:all,otpdiaService=Pcrf,otpdia  
Product=SAPC
```

In typical deployments, 1–5 VIP addresses are configured to an ALB, but up to 40 VIP addresses can be configured to an ALB if needed.

Note: It is not possible to configure more than one SCTP transport. There should be only one with all the VIPs addresses.

4. Configure multihoming in SS7CAF:

- a. Change to root user:

```
sapcadmin@SC-1:~ # su
```

- b. Export JRE path to be able to open SS7CAF CLI

```
SC-1:~ # export PATH="$PATH:/opt/sign/EABss7069/jre/  
bin"
```

- c. Enter in the SS7CAF CLI:

```
SC-1:~ # /opt/sign/EABss7050/bin/signmcli -own.conf=/  
cluster/storage/system/config/ss7caf-ana90137/etc/  
signmgr.cnf
```

- d. List all IID (front end processes defined in traffic PLs):

```
cli> STSTP;
```

Check if the result prints `UseAllLocalIPAddresses=Yes` (use all local VIPs)

- e. If `UseAllLocalIPAddresses` parameter is set to No, set it to use all local VIPs:

```
cli> STSTC: IID=0, UseAllLocalIPAddresses=1;
```

```
EXECUTED
```

```
cli> STSTC: IID=1, UseAllLocalIPAddresses=1;
```

```
EXECUTED
```



```
cli> STSTC: IID=2, UseAllLocalIPAddresses=1;
```

```
EXECUTED
```

```
cli> STSTC: IID=3, UseAllLocalIPAddresses=1;
```

```
EXECUTED
```

```
...
```

- f. To apply new configuration, run next command:

```
cli> configure: INITIAL;
```

You should now (re)start the stack.

```
EXECUTED
```

```
cli> procr: ALL;
```

```
EXECUTED
```

The stack restart takes some time. After that, all local VIPs are listening SCTP.

- g. Verify the result:

```
cli> STSTP
```

Check that UseAllLocalIPAddresses value is set to Yes for each front end.

8.1 Path Separation, Path Diversity and Multihoming

In cloud deployments, configure path separation in the Data Center Gateways (DC-GW) and assure that the network infrastructure also supports it.

If path diversity is required, assure that it is supported in the cloud infrastructure. Path diversity might be possible to be provided from the DC-GWs and onwards to the site infrastructure and the IP backbone.



9 Reference List

Online References

1. Diameter Base Protocol - <https://tools.ietf.org/html/rfc6733>