SAPC Troubleshooting Guide

Ericsson Service-Aware Policy Controller

Troubleshooting

1/154 51-AXB 901 33/7 Uen Z

Copyright

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 Introduction

Attention!

The N+1 architecture is a fully restricted feature. All N+1 related content has been included in official documentation in order to maintain a single documentation track. This feature cannot be enabled without approval from Ericsson.

The purpose of this document is to provide detailed instructions to locate and fix different problems in the SAPC typically in live sites.

This document requires strong knowledge of the product and used Component Based Architecture (CBA) components. It is addressed to both Ericsson personnel and System Administrators.

This document does not contain periodic maintenance tasks and instructions to change the configuration of the main functions within the SAPC. The System Administrator Guide contains this type of information.

Note: The described procedures executed by 'sapcadmin' user can also be executed by the 'sapctroubleshooter' user unless it is specified not.

2 Troubleshooting Tools

This section describes the tools that can be used to troubleshoot the SAPC.

2.1 forall

This command script launches the same CLI command or commands to several SAPC nodes, according to the following use:

```
sapcadmin@SC-1:~> forall
```

The `forall` command runs *<'command'>* in all nodes included in `<node_group>`.

```
Usage: forall <node_group> <'command'>
```

The valid values for `<node_group>` are the following:

- Output of the `'immHelper ng'` command
- AllNodes
- SCs
- PLs
- cluster
- control
- payload

<'command'> must be quoted. It can be any command that is executable through ssh.

Examples:

```
forall payload 'ps -fe | grep pcrf-proc'
forall SCs 'hostname ; exportfs'
forall cluster 'hostname ; uptime'
forall PLs 'hostname; netstat -anp | grep -c 3868'
```

The reserved *<'control'>* , *<'payload'>* , and *<'cluster'>* values have the same effect as the *<'SCs'>* , *<'PLs'>* , and *<'AllNodes'>* node groups. However, they can be used even when Information Model Management (IMM) is not available because they extract the information from the cluster file system hierarchy.

2.2 immHelper

This command helps to know the current SAPC components state, according to the following use:

```
sapcadmin@SC-1:~> immHelper

Usage: immHelper <command: su|su2|sg|sg2|si|si2|ng|ng2> [FILTER]

command:
       ng: SAPC node groups  | ng2: detailed ng (show [L]ocked/[U →
]nlocked nodes)
       su: service units     | su2: detailed su (includes node na →
me)
       sg: service groups    | sg2: detailed sg (includes node gr →
oup)
       si: service instances | si2: detailed si (includes availab →
ility)
       comp: components
       sw: installed software

     You could grep results with 'SAPC' or whatever ...

Example:
"immHelper su | grep -i license" prints the license monitor servi →
ce unit in the system.

Example Output
[Service Unit DN]                                                 →
                                    [AdminState]  [OpState]     [ →
PresenceState] [ReadinessState]
safSu=PL-3,safSg=2N,safApp=ERIC-sapc.licenseMonitor.payload       →
                                    UNLOCKED(1)   ENABLED(1)    I →
NSTANTIATED(3) IN-SERVICE(2)
safSu=PL-4,safSg=2N,safApp=ERIC-sapc.licenseMonitor.payload       →
                                    UNLOCKED(1)   ENABLED(1)    U →
NINSTANTIATED(1) IN-SERVICE(2)

Done!
```

2.3 amfHelper

This command executes actions on Service Units, according to the following use:

```
sapcadmin@SC-1:~> amfHelper
```

Wrapper to execute actions on Service units using amf-adm. It encapsulates the complexity of "Preinstantiable" and handles lock/unlock and lock-in/unlock-in in correct way. The repair option tries to unlock or repair matched service group and service unit that are locked or in a wrong status.

```
Use: amfHelper -f <filter> [-e <filter>] [-a <action>] [-v]
```

```
Parameters:
     -f <filter>: Service unit and service group egrep filter. F →
or example: amfHelper -f 'pcrf|CDiameter'
     -e <filter>: Service unit and service group excluded egrep  →
filter. For example: amfHelper -e 'pcrf|CDiameter'
     -a <action>: stop | start | restart | status | repair . Int →
eractive menu if missing.
     -v: verbose mode.
Examples:
   amfHelper -f 'pcrf|CDiameter' -a stop
   amfHelper -f 'sapc' -a stop
   amfHelper -f 'sapc' -e 'pcrf|CDiameter' -a stop
   amfHelper -a repair -f 'sapc'
```

2.4 sapcHealthCheck

This command performs several checkups to verify the status of the system: SAPC deploy, SAPC CPU&MEM real time status, TIPC communication, DRBD devices, CMW status, active FM alarms, existing coredumps, Data Base and error logs in the system. It also provides an overall status in function of checkups results.

According to the activity (installation, upgrade, O&M or scaling workflow), this script applies different checkups and uses different criteria for overall status. Moreover, the command allows getting each checkup independently.

The use of the command is done according to the following usages:

```
sapcadmin@SC-1:~> sudo sapcHealthCheck -h

Usage: sapcHealthCheck [-t <seconds>] [-p CHECKUP ]
       sapcHealthCheck [-t <seconds>] [BATCH]

OPTIONS:
  -h, --help        help
  -t, --timeout     timeout seconds for checking platform command →
s. Set on 300 sg by default.
  -p, --param       specific checkup

CHECKUP:
  SAPCInstallation
  CPUMEM
  Connectivity
  DRBD
  CoreMiddleware
  Alarms
  CoreDumps
  SystemOperative
  DataBase
  ErrorLogs

BATCH
  -d, --deploy     deployment/installation checkups
  -u, --upgrade    upgrade checkups
  -o, --oam        operation and maintenance checkups
  -s, --scaling    scaling checkups
```

Attention!

This command reports Health Check NOK for Not ACTIVE zone in Geographical Redundancy deployments.

2.5 BackupFormatter

The BackupFormatter tool is used to export information contained in the backups of the SAPC internal database.

For detailed information on the BackupFormatter tool, refer to Toolkit Description.

2.6 Session-handler

2.6.1 Session Handling per Subscriber

The session-handler accesses Database Service (DBS) and retrieves the session model for the subscriber identified by a specific traffic identifier, including MSISDN, IMSI, or SIP-URI.

Show Operation

The session-handler tool shows relevant information regarding the bound sessions of the subscriber for all the applicable protocols (Gx, Smp, N7, N15_AMPC, N15_UEPC, Rx, N5N30_POLAUTH, Sy, and Sd).

The session-handler tool also supports to show the session information for specific subscriber and specific peer node with applicable protocol (Gx, Smp, N7, N15_AMPC, N15_UEPC) and related information for bound sessions (Rx, N5N30_POLAUTH, Sy, Sd).

The session-handler tool supports to show only the information of AF sessions (Rx and N5N30_POLAUTH protocols).

Delete Operation

Depending on the requested action, the tool also allows the deletion of retrieved sessions from the DBS. By default, any request for a “delete” action requires confirmation from the user.

Session deletion is not done in DBS directly. The following traffic policy control processes are notified about those Gx, Smp, N7, N15_AMPC, N15_UEPC, Rx, N5N30_POLAUTH, Sd, or Sy sessions which need to be deleted from the session model of the subscriber.

- `pcrf-proc`
- `sapc-mobility-policy`
- `sapc-app-detection`
- `subs-charging-proc`
- `session-policy-control`
- `sapc-rest-mobility-policy`
- `ue-policy-control`
- `policy-authorization-control`

The appropriate notify message would be sent to peer Network Elements to notify session for that specific subscriber removed if parameter `–notify` used.

The session-handler tool supports to remove the session for specific subscriber and specific peer node with applicable protocol (Gx, Smp, N7, N15_AMPC, N15_UEPC).

The session-handler tool also supports to remove only bounded AF (Rx and N5N30_POLAUTH) sessions for specific subscriber and specific peer Node with applicable protocol (Gx, N7).

By default, the verification that the required sessions are deleted from DBS is configured to be done with a maximum of five checking retries and an interval of two seconds between those retries. These default values can be modified in the configuration file that is described later.

If after all the retries, the deletion of some session could not be verified by any reason, a descriptive warning message is shown and the unverified sessions are listed.

Note: In the Geographical Redundancy Active-Active deployment, the behaviors of the delete operation are different based on the state of the SAPC peers.

When both SAPC peers are in the DISTRIBUTED state, the delete operation running on one SAPC peer removes the retrieved sessions based on session mastership. For example, when the system administrator running the delete operation on SAPC1, then SAPC1 removes the retrieved sessions bound to SAPC1. The retrieved sessions bound to SAPC2 are removed by SAPC2.

When the SAPC peers are not in the DISTRIBUTED state, the delete operation running on one SAPC peer removes all retrieved sessions, regardless of the session mastership.

Note: In case traffic separation configuration is implemented, this tool cannot run if the traffic payload's default gateway could not reach the external DB.

2.6.1.1 Configuration for Session Handling per Subscriber

The session-handler tool uses a configuration file to set internal values for session handling of specific subscriber. By default, the configuration file `session-handler.cfg` can be found in the following path: `/storage/system/config/sapc/session-handler.cfg`.

The users can also provide a different configuration file by means of the `--cfg` option when running the session-handler.

The default content in this configuration file is as follows:

```
# thrift configuration to connect with pcrf-proc
pcrf-proc-thrift-host=localhost
pcrf-proc-thrift-port=${@SAPC_PORT_PCRF@}
# thrift configuration to connect with session-policy-control
session-policy-control-thrift-host=localhost
session-policy-control-thrift-port=${@SAPC_PORT_SESSION_POLICY_CONTROL_REAUTH@}
# thrift configuration to connect with rest-mobility-policy-control
rest-mobility-policy-thrift-host=localhost
rest-mobility-policy-thrift-port=${@SAPC_PORT_REST_MOBILITY_POLICY_REAUTH@}
thrift-connection-timeout=30
thrift-send-recv-timeout=200
# maximum number of retries while checking for sessions deletion
deletion-verification-retries-number=5
# number of seconds between retries while checking for sessions deletion
deletion-verification-time-between-retries=2
```

The values of `<SAPC_PORT_PCRF>` for the SAPC PCRF traffic, and `<SAPC_PORT_SESSION_POLICY_CONTROL_REAUTH>` and `<SAPC_PORT_REST_MOBILITY_POLICY_REAUTH>` for the SAPC PCF traffic are established by the deployment of the SAPC. The values correspond to the ports used by the traffic policy control processes, pcrf-proc process for the SAPC PCRF, session-policy-control and rest-mobility-policy processes for the SAPC PCF, to receive session termination requests through the Thrift protocol.

2.6.2 Session Handling per Node

The session-handler also supports to access DBS and handles the sessions on a per-node basis for both PDU sessions for a specific SMF (N7), AMF (N15_AMPC) and AMF (N15_UEPC) including node ID, node IP, or FQDN, and IP session by a specific PCEF (Gx) or SGSN-MME (Smp) `peerId`.

Show Operation

The session-handler tool shows the total number of sessions basis on the applicable protocols (N7, N15_AMPC, N15_UEPC, Gx, Smp, and N5N30_POLAUTH) as specified `sessionType`. The show action can also show the progress of an ongoing session cleanup.

The session-handler tool supports to show only the number of bound AF sessions (Rx and N5N30_POLAUTH protocols).

Delete Operation

When performing the delete action, the session-handler triggers a session cleanup which removes all related sessions of the specified peer node. For an SMF, all N7 sessions and the existing binding Rx , N5/N30, and Sy sessions are removed from the DBS. For an AMF, N15 (AMPC) or N15 (UEPC) can be removed from the DBS based on the session type separately. For a PCEF, all Gx sessions and related bound sessions (Rx, Sy, Sd) are removed from DBS. For a SGSN-MME, all Smp sessions are removed from DBS.

The session-handler tool also supports to have only N5N30_POLAUTH sessions remove with specific NEF node.

The session-handler tool also supports to remove only all bounded AF sessions for specified PCEF node (Rx sessions) or SMF node (Rx and N5N30_POLAUTH) with parameter `–onlyAF`.

Depending on the number of sessions to be removed, it can take hours for the session cleanup to complete. After performing a `delete` action, users can use the `show` action to show if the deletion is completed, or refer to the logged events for results and detailed information.

When a deletion for a specific peer node is ongoing, and the user runs the tool with `delete` action again for the same peer node, then the tool shows the progress of the ongoing deletion. No new deletion is triggered. If the user runs the tool with `delete` action for a different peer node, the tool triggers a new deletion.

By default, any request for a `delete` action that triggers a new deletion requires confirmation from the user.

Note: In the Geographical Redundancy Active-Active deployment, the behaviors of the delete operation are different based on the state of the SAPC peers.

When both SAPC peers are in the DISTRIBUTED state, the delete operation running on one SAPC peer removes the retrieved sessions based on session mastership. For example, when the system administrator running the delete operation on SAPC1, then SAPC1 removes the retrieved sessions bound to SAPC1. The retrieved sessions bound to SAPC2 are removed by SAPC2.

When the SAPC peers are not in the DISTRIBUTED state, the delete operation running on one SAPC peer removes all retrieved sessions, regardless of the session mastership.

Note: In case traffic separation configuration is implemented, this tool cannot run if the traffic payload's default gateway could not reach the external DB.

2.6.3 Run Session-handler Tool

This tool runs on any traffic payload according to the following usage:

Steps

1. This step is optional. Acquire the syntax of the session-handler with the following command:

```
sapcadmin@PL-3:~> sudo session-handler --help
```

The syntax of the session-handler tool is as follows:

```
session-handler --trafficId <trafficId> --peerNode <peernode> --action <action> [--notify] [--sessionType <sessionType>] [--onlyAF] [--noConfirm] [--cfg <configFile>] | --help
```

Parameters	Description
--help	This help message.
--trafficId *<trafficId>*	The traffic ID value (IMSI, MSISDN, or SIP-URI value) identifying the subscriber whose sessions are going be shown or deleted.
-- peerNode *<peernode>*	The peer node identifier (host@realm for IP session, node ID, node IP, or FQDN for PDU session) to specify the peer node whose sessions are to be shown or deleted.

Parameters	Description
--action *<action>*	Action to be executed. The valid values are: `show`, `delete`.
--notify	This option is applied only for the delete action with specific traffic ID. Notify peer side sessions are removed.
--sessionType *<sessionType>*	It indicates the session type. Mandatory if only `--peerNode` is used. The valid values are: `Gx` | `N7` | `N15_AMPC` | `N15_UEPC` | `N5N30_POLAUTH` | `Smp`. On deletion: — For session handling of a specific subscriber, if `Gx` is specified, the Gx sessions and their dependent Rx, Sy and Sd sessions will be deleted. If `Smp` is specified, the Smp session will be deleted. If `N7` is specified, the N7 sessions and their dependent Rx, N5N30_POLAUTH and Sy sessions will be deleted. If `N15_AMPC` is specified, the N15_AMPC sessions will be deleted. If `N15_UEPC` is specified, the N15_UEPC sessions will be deleted. If this option is omitted, all the sessions (Gx, Smp, N7, N15_AMPC, N15_UEPC, Rx, N5N30_POLAUTH, Sy, Sd) will be deleted. — For session handling of a specific peer node, if `Gx` is specified, the tool only filters the Gx sessions for the specified PCEF node, and deletes the retrieved Gx sessions and their dependent Rx, Sd and Sy sessions. If `Smp` is specified, the tool only filters the Smp sessions for the specified SGSN-MME node, and deletes the retrieved Smp sessions. If `N7` is specified, the tool only filters the N7 sessions for the specified SMF node, and deletes the retrieved N7 sessions and

Parameters	Description
	their dependent Rx, N5/N30 and Sy sessions. If `N15_AMPC` is specified, the tool only filters the N15_AMPC sessions for the specified AMF node, and deletes the retrieved N15_AMPC sessions. If `N15_UEPC` is specified, the tool only filters the N15_UEPC sessions for the specified AMF node, and deletes the retrieved N15_UEPC sessions. If `N5N30_POLAUTH` is specified, the tool only filters the N5N30_POLAUTH sessions for the specified AF/NEF node, and deletes the retrieved N5N30_POLAUTH sessions only.
--onlyAF	Apply the requested action only to AF sessions (Rx and N5N30_POLAUTH).
--noConfirm	Do not ask for confirmation in case of a delete action request.
--cfg *<configFile>*	A specific configuration file to replace the default configuration file and to override the default configuration values.

2. Use appropriate command to handle sessions for specific subscriber or peer node:

   — the following syntax shows all supported options for session handling based on traffic ID:

   ```
   session-handler --trafficId <trafficId> --action <action>
   [--notify] [--sessionType <sessionType>] [--onlyAF] [--
   noConfirm] [--cfg <configFile>]
   ```

   — the following syntax shows all supported options for session handling based on specific traffic ID and specific peer node:

   ```
   session-handler --trafficId <trafficId> --action <action> --
   peerNode <peernode> [--notify] --sessionType <sessionType>
   [--onlyAF] [--noConfirm] [--cfg <configFile>]
   ```

   See Examples of Session Handling per Subscriber on page 12 for examples of the commands.

   — the following syntax shows all supported options for session handling based on peer node:

```
session-handler --peerNode <peernode> --action <action> --
sessionType <sessionType> [--noConfirm] [--onlyAF]
```

See Examples of Session Handling per Node on page 19 for examples of the commands.

2.6.4 Examples of Session Handling per Subscriber

The following examples show the typical uses of session-handler tool to handle sessions for certain subscriber specified by traffic ID.

Example 1 Both Gx+Rx and N7 + Rx + N5N30 sessions created for specific traffic ID ‘460001234567891‘:

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 460001234567891 --action show

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [460001234567891] -> adminSubsId [460001234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [460001234567891]
---------------------------------------------
Gx sessions:
  IP session (Gx):
    ipAddr (IP@APN@PCEF) = [223.136.18.23@Default@ggsnNodeHostname.nodeHostRealm.com]
    diamSessionId (Gx) = [ggsnNodeHostname.nodeHostRealm.com;460001234567891;1637117130752064301]
    creationTime = [Nov 17, 2021; 03:45:33]
    modificationTime = [Nov 17, 2021; 03:45:33]
    peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
    pccRules = { [MMTel_Service_audio], [MMTel_Service_audio], [MMTel_Service_video], [MMTel_Servi →
ce_video], [StaticInternet] }

    Af sessions:
      Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;46000123 →
4567891;1637117126420559032]
        creationTime = [Nov 17, 2021; 03:45:33]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]

Smp sessions:

N7 sessions:
  IP session (N7):
    ipAddr (IP@DNN@SMF@SUPI@PduSessionId@Timestamp) = [223.136.18.24@dnn_mbb.com@http://192.168.14 →
.42:7071/@imsi-460001234567891@123@1637117133]
    smPolicyId (SUPI@GPSI@PduSessionId@Timestamp) = [imsi-460001234567891@@123@1637117133]
    creationTime = [Nov 17, 2021; 03:45:33]
    modificationTime = [Nov 17, 2021; 03:45:33]
    peerId = [http://192.168.14.42:7071/@5G@Default]
    pccRules = { [MMTel_Service_audio], [MMTel_Service_audio], [MMTel_Service_video], [MMTel_Servi →
ce_video] }

    Af sessions:
      Af session (N5N30_POLAUTH):
        afSessionId (IP@diamSessonId) = [223.136.18.24@223.136.18.24@1637117133278554984]
        creationTime = [Nov 17, 2021; 03:45:33]
        peerId = [http://192.168.14.42:7091/@5G]
        state = [0]
      Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;46000123 →
4567891;1637117126420562464]
        creationTime = [Nov 17, 2021; 03:45:33]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]

Sy sessions:

Sd sessions:
```

```
N15_AMPC sessions:

N15_UEPC sessions:
```

Example 2   Remove all the sessions created under this specific trafficID ‘460001234567891’ without command confirm

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 460001234567891 --action delete --noConfirm

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [460001234567891] -> adminSubsId [460001234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [460001234567891]
---------------------------------------------
Gx sessions:
  IP session (Gx):
    ipAddr (IP@APN@PCEF) = [223.136.18.23@Default@ggsnNodeHostname.nodeHostRealm.com]
    diamSessionId (Gx) = [ggsnNodeHostname.nodeHostRealm.com;460001234567891;1637117324814936004]
    creationTime = [Nov 17, 2021; 03:48:50]
    modificationTime = [Nov 17, 2021; 03:48:50]
    peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
    pccRules = { [MMTel_Service_audio], [MMTel_Service_audio], [MMTel_Service_video], [MMTel_Servi →
ce_video], [StaticInternet] }

    Af sessions:
      Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;46000123 →
4567891;1637117324828293822]
        creationTime = [Nov 17, 2021; 03:48:50]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]

Smp sessions:

N7 sessions:
  IP session (N7):
    ipAddr (IP@DNN@SMF@SUPI@PduSessionId@Timestamp) = [223.136.18.24@dnn_mbb.com@http://192.168.14 →
.42:7071/@imsi-460001234567891@123@1637117330]
    smPolicyId (SUPI@GPSI@PduSessionId@Timestamp) = [imsi-460001234567891@@123@1637117330]
    creationTime = [Nov 17, 2021; 03:48:50]
    modificationTime = [Nov 17, 2021; 03:48:50]
    peerId = [http://192.168.14.42:7071/@5G@Default]
    pccRules = { [MMTel_Service_audio], [MMTel_Service_audio], [MMTel_Service_video], [MMTel_Servi →
ce_video] }

    Af sessions:
      Af session (N5N30_POLAUTH):
        afSessionId (IP@diamSessonId) = [223.136.18.24@223.136.18.24@1637117330754889573]
        creationTime = [Nov 17, 2021; 03:48:50]
        peerId = [http://192.168.14.42:7091/@5G]
        state = [0]
      Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;46000123 →
4567891;1637117324828297441]
        creationTime = [Nov 17, 2021; 03:48:50]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]

Sy sessions:

Sd sessions:

N15_AMPC sessions:

N15_UEPC sessions:

# # # # # # # # # # # # # # # # # # # # # # 

Exception: THRIFT_EAGAIN (timed out)
WARNING: there was a problem while sending the sessions termination request. Sessions deletion cou →
```

```
ld be incomplete.

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted N7 session corresponding to:
  ipAddr (IP@DNN@SMF@SUPI@PduSessionId@Timestamp) = [223.136.18.24@dnn_mbb.com@http://192.168.14.4 →
2:7071/@imsi-460001234567891@123@1637117330]
  smPolicyId (SUPI@GPSI@PduSessionId@Timestamp) = [imsi-460001234567891@@123@1637117330]
  creationTime = [Nov 17, 2021; 03:48:50]
  peerId = [http://192.168.14.42:7071/@5G@Default]

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@223.136.18.24@1637117330754889573]
  creationTime = [Nov 17, 2021; 03:48:50]
  peerId = [http://192.168.14.42:7091/@5G]

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;460001234567891;163711 →
7324828297441]
  creationTime = [Nov 17, 2021; 03:48:50]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]

Deleted Gx session corresponding to:
  ipAddr (IP@APN@PCEF) = [223.136.18.23@Default@ggsnNodeHostname.nodeHostRealm.com]
  diamSessionId (Gx) = [ggsnNodeHostname.nodeHostRealm.com;460001234567891;1637117324814936004]
  creationTime = [Nov 17, 2021; 03:48:50]
  peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;460001234567891;163711 →
7324828293822]
  creationTime = [Nov 17, 2021; 03:48:50]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
```

Example 3 Remove only AF sessions for this specific traffic ID '460001234567891'. Both Rx session bound with Gx session, and Rx + N5N30 sessions bound with N7 session are removed.

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 460001234567891 --action delete --onlyAF --noCo →
nfirm

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [460001234567891] -> adminSubsId [460001234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [460001234567891]
---------------------------------------------
Af sessions:
  Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;46000123 →
4567891;1637117867361907936]
        creationTime = [Nov 17, 2021; 03:57:54]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]
  Af session (N5N30_POLAUTH):
        afSessionId (IP@diamSessonId) = [223.136.18.24@223.136.18.24@1637117874353274254]
        creationTime = [Nov 17, 2021; 03:57:54]
        peerId = [http://192.168.14.42:7091/@5G]
        state = [0]
  Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;46000123 →
4567891;1637117867361912119]
        creationTime = [Nov 17, 2021; 03:57:54]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]
```

```
# # # # # # # # # # # # # # # # # # # # # # 


Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@223.136.18.24@1637117874353274254]
  creationTime = [Nov 17, 2021; 03:57:54]
  peerId = [http://192.168.14.42:7091/@5G]

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;460001234567891;163711  →
7867361912119]
  creationTime = [Nov 17, 2021; 03:57:54]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;460001234567891;163711  →
7867361907936]
  creationTime = [Nov 17, 2021; 03:57:54]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
```

Example 4 Remove all N7 and bound AF (Rx+N5N30) sessions for this specific traffic ID ‘460001234567891’. Notify message would be sent to related SMF.

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 460001234567891 --action delete --sessionType N  →
7 --notify --noConfirm

ouput:[SC-1:~ #  Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [460001234567891] -> adminSubsId [460001234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [460001234567891]
---------------------------------------------
Gx sessions:

Smp sessions:

N7 sessions:
  IP session (N7):
    ipAddr (IP@DNN@SMF@SUPI@PduSessionId@Timestamp) = [223.136.18.24@dnn_mbb.com@http://192.168.14  →
.42:7071/@imsi-460001234567891@123@1637118057]
    smPolicyId (SUPI@GPSI@PduSessionId@Timestamp) = [imsi-460001234567891@@123@1637118057]
    creationTime = [Nov 17, 2021; 04:00:57]
    modificationTime = [Nov 17, 2021; 04:00:57]
    peerId = [http://192.168.14.42:7071/@5G@Default]
    pccRules = { [MMTel_Service_audio], [MMTel_Service_audio], [MMTel_Service_video], [MMTel_Servi  →
ce_video] }

    Af sessions:
      Af session (N5N30_POLAUTH):
        afSessionId (IP@diamSessonId) = [223.136.18.24@223.136.18.24@1637118057947103580]
        creationTime = [Nov 17, 2021; 04:00:57]
        peerId = [http://192.168.14.42:7091/@5G]
        state = [0]
      Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;46000123  →
4567891;1637118051101052925]
        creationTime = [Nov 17, 2021; 04:00:57]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]

Sy sessions:

Sd sessions:

N15_AMPC sessions:

N15_UEPC sessions:

# # # # # # # # # # # # # # # # # # # # # # 

Exception: THRIFT_EAGAIN (timed out)
```

```
WARNING: there was a problem while sending the sessions termination request. Sessions deletion could be incomplete.

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted N7 session corresponding to:
  ipAddr (IP@DNN@SMF@SUPI@PduSessionId@Timestamp) = [223.136.18.24@dnn_mbb.com@http://192.168.14.42:7071/@imsi-460001234567891@123@1637118057]
  smPolicyId (SUPI@GPSI@PduSessionId@Timestamp) = [imsi-460001234567891@@123@1637118057]
  creationTime = [Nov 17, 2021; 04:00:57]
  peerId = [http://192.168.14.42:7071/@5G@Default]

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@223.136.18.24@1637118057947103580]
  creationTime = [Nov 17, 2021; 04:00:57]
  peerId = [http://192.168.14.42:7091/@5G]

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;460001234567891;1637118051101052925]
  creationTime = [Nov 17, 2021; 04:00:57]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
```

Example 5 Remove Gx session created by specific PCEF node (identified with host@realm ‘ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com’) and bound Rx session for this specific traffic ID ‘460001234567891’

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 460001234567891 --peerNode ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com --sessionType Gx --action delete --noConfirm

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [460001234567891] -> adminSubsId [460001234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [460001234567891]
---------------------------------------------
Gx sessions:
  IP session (Gx):
    ipAddr (IP@APN@PCEF) = [223.136.18.23@Default@ggsnNodeHostname.nodeHostRealm.com]
    diamSessionId (Gx) = [ggsnNodeHostname.nodeHostRealm.com;460001234567891;1637118745755441745]
    creationTime = [Nov 17, 2021; 04:12:29]
    modificationTime = [Nov 17, 2021; 04:12:29]
    peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
    pccRules = { [MMTel_Service_audio], [MMTel_Service_audio], [MMTel_Service_video], [MMTel_Service_video], [StaticInternet] }

    Af sessions:
      Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;460001234567891;1637118741166275425]
        creationTime = [Nov 17, 2021; 04:12:29]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]

Smp sessions:

N7 sessions:

Sy sessions:

Sd sessions:

N15_AMPC sessions:
```

```
N15_UEPC sessions:

# # # # # # # # # # # # # # # # # # # # # # 


Deleted Gx session corresponding to:
  ipAddr (IP@APN@PCEF) = [223.136.18.23@Default@ggsnNodeHostname.nodeHostRealm.com]
  diamSessionId (Gx) = [ggsnNodeHostname.nodeHostRealm.com;460001234567891;1637118745755441745]
  creationTime = [Nov 17, 2021; 04:12:29]
  peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.23@afNodeHostname.afNodeHostRealm.com;460001234567891;163711  →
8741166275425]
  creationTime = [Nov 17, 2021; 04:12:29]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
```

Example 6 Remove only AF sessions (Rx + N5N30) bound with N7 session created by specific SMF node (identified by IP:Port ‘http://192.168.14.42:7071‘) under this traffic ID ‘460001234567891‘

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 460001234567891 --peerNode http://192.168.14.42  →
:7071 --sessionType N7 --action delete --onlyAF --noConfirm

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [460001234567891] -> adminSubsId [460001234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [460001234567891]
---------------------------------------------
Af sessions:
  Af session (N5N30_POLAUTH):
        afSessionId (IP@diamSessonId) = [223.136.18.24@223.136.18.24@1637119231238788507]
        creationTime = [Nov 17, 2021; 04:20:31]
        peerId = [http://192.168.14.42:7091/@5G]
        state = [0]
  Af session (Rx):
        afSessionId (IP@diamSessonId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;46000123  →
4567891;1637119224531180812]
        creationTime = [Nov 17, 2021; 04:20:31]
        peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
        state = [0]


# # # # # # # # # # # # # # # # # # # # # # 


Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@223.136.18.24@1637119231238788507]
  creationTime = [Nov 17, 2021; 04:20:31]
  peerId = [http://192.168.14.42:7091/@5G]

Deleted Af session corresponding to:
  id (IP@diamSessionId) = [223.136.18.24@afNodeHostname.afNodeHostRealm.com;460001234567891;163711  →
9224531180812]
  creationTime = [Nov 17, 2021; 04:20:31]
  peerId = [afNodeHostname.afNodeHostRealm.com@afNodeHostRealm.com]
```

Note: For the N7 sessions, the `ipAddr` attribute displayed in the output could be different than the `IP@DNN@SMF@SUPI@PduSessionId@Timestamp` format. In case of having a pending session (a session that is established without an IP address), the displayed ipAddr format is: `SUPI:PDUSessionId@DNN@SMF@SUPI@PduSessionId@Timestamp` instead.

Example 7 Both AMPC + UEPC sessions created for specific traffic ID '240811234567891'

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 240811234567891 --action show

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [240811234567891] -> adminSubsId [240811234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [240811234567891]
---------------------------------------------
Gx sessions:

Smp sessions:

N7 sessions:

Sy sessions:

Sd sessions:

N15_AMPC sessions:
  N15_AMPC session:
    sessionId (SUPI@GPSI@Timestamp) = [imsi-240811234567891@@1655976606969713]
    creationTime = [Jul 07, 1983; 03:29:21]
    peerId = [http://192.168.14.42:7083/]

N15_UEPC sessions:
  N15_UEPC session:
    sessionId (SUPI@Timestamp) = [imsi-240811234567891@1655976602622494]
    creationTime = [May 17, 1983; 19:55:42]
    peerId = [http://192.168.14.42:7083/]

# # # # # # # # # # # # # # # # # # # # # #
```

Example 8 Both AMPC + UEPC sessions delete for specific traffic ID '240811234567891'

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 240811234567891 --action delete

Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [240811234567891] -> adminSubsId [240811234567891]
---------------------------------------------
Session data model corresponding to trafficId -> [240811234567891]
---------------------------------------------
Gx sessions:

Smp sessions:

N7 sessions:

Sy sessions:

Sd sessions:

N15_AMPC sessions:
  N15_AMPC session:
    sessionId (SUPI@GPSI@Timestamp) = [imsi-240811234567891@@1655977078071374]
    creationTime = [Jun 10, 1998; 17:03:42]
    peerId = [http://192.168.14.42:7083/]

N15_UEPC sessions:
  N15_UEPC session:
    sessionId (SUPI@Timestamp) = [imsi-240811234567891@1655977074293757]
    creationTime = [Apr 27, 1998; 23:43:25]
    peerId = [http://192.168.14.42:7083/]


# # # # # # # # # # # # # # # # # # # # # # #

Deletion requires confirmation...
Current deletion with no specific option(s).
Do you want to proceed with the deletion? [y/n]
y
Exception: THRIFT_EAGAIN (timed out)
WARNING: there was a problem while sending the sessions termination request. Sessions deletion cou
ld be incomplete.

Deleted N15_AMPC session corresponding to:
  sessionId (N15_AMPC) = [imsi-240811234567891@@1655977078071374]
  adminSubsId = [imsi-240811234567891]

...
Deletion still not verified for some sessions. Retrying the check after 2 seconds.

Deleted N15_AMPC session corresponding to:
  sessionId (N15_AMPC) = [imsi-240811234567891@@1655977078071374]
  adminSubsId = [imsi-240811234567891]

Deleted N15_UEPC session corresponding to:
  sessionId (N15_UEPC) = [imsi-240811234567891@1655977074293757]
  adminSubsId = [imsi-240811234567891]
```

2.6.5 Examples of Session Handling per Node

The following examples show the typical uses of session-handler tool to handle sessions for certain peer nodes.

Example 9 Show and remove all N7 sessions for an SMF peer node identified with IP:Port

This is an example to show the total number of N7 sessions for an SMF identified by IP address `http://192.168.14.42:7071`.

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://192.168.14.42:7071 --action show --sessio
nType N7

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> http://192.168.14.42:7071
---------------------------------------------
[ 20000 ] active [ N7 ] sessions on node [ http://192.168.14.42:7071 ].

# # # # # # # # # # # # # # # # # # # # # # #
```

This is an example to delete the N7 sessions for an SMF identified by IP address http://192.168.14.42:7071, without asking the user for confirmation.

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://192.168.14.42:7071 --action delete --sess
ionType N7 --noConfirm

Initializing DBN API
```

```
Waiting for DBN (false)...
--------------------------------------------
Session data model corresponding to peerNode -> http://192.168.14.42:7071
--------------------------------------------

Session cleanup for node [ http://192.168.14.42:7071 ] is ongoing, [ 20000 ] active [ N7 ] session →
s should be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # #
```

This output demonstrates how the tool shows the number of N7 sessions to be deleted first. Then, without asking the user for confirmation, it removes all the related N7 sessions and binding Rx, N5N30_POLAUTH, and Sy sessions.

If the deletion is still ongoing, the tool shows the progress of the ongoing deletion when command given again.

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://192.168.14.42:7071 --action delete --sess →
ionType N7 --noConfirm

Initializing DBN API
Waiting for DBN (false)...
--------------------------------------------
Session data model corresponding to peerNode -> http://192.168.14.42:7071
--------------------------------------------
Session cleanup for node [ http://192.168.14.42:7071 ] is ongoing, [ 20000 ] active [ N7 ] session →
s should be cleaned, [ 13000 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # #
```

If the deletion is finished, the tool shows that all related N7 sessions are removed:

```
Initializing DBN API
Waiting for DBN (false)...
--------------------------------------------
Session data model corresponding to peerNode -> http://192.168.14.42:7071
--------------------------------------------
No active session is found for node [ http://192.168.14.42:7071 ].

# # # # # # # # # # # # # # # # # # # # # # #
```

Example 10 Show and remove all N15_AMPC sessions for an AMF peer node identified by fqdn: 'http://traf:7081/'

Show the number of opened sessions with AMF peerNode http://traf:7081/

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://traf:7081/ --action show --sessionType N1 →
5_AMPC

Initializing DBN API
Waiting for DBN (false)...
--------------------------------------------
Session data model corresponding to peerNode -> [http://traf:7081/]
--------------------------------------------
[ 500 ] active [ N15_AMPC ] sessions on node [ http://traf:7081/ ].
# # # # # # # # # # # # # # # # # # # # # # #
```

Delete opened sessions with AMF peerNode http://traf:7081/ with confirm.

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://traf:7081/ --action delete --sessionType  →
N15_AMPC

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [http://traf:7081/]
---------------------------------------------
Session cleanup for node [ http://traf:7081/ ] is ongoing, [ 500 ] active [ N15_AMPC ] sessions sh  →
ould be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # #

Deletion requires confirmation...
Current deletion with specific option(s) '--peerNode http://traf:7081/
Do you want to proceed with the deletion? [y/n]
Y
```

This output demonstrates how the tool shows the number of N15 (AMPC) sessions to be deleted and asks the user for confirmation. When the user confirms the deletion, all related N15 (AMPC) sessions are removed from the DBS.

Example 11 Show and remove all N5N30_POLAUTH sessions for NEF/AF peer node identified with IP:Port ‘http://10.200.71.131:7091/‘

Only the number of N5N30 sessions would be counted for specific NEF/AF peer node.

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://10.200.71.131:7091/ --action show --sessi  →
onType N5N30_POLAUTH

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [http://10.200.71.131:7091/]
---------------------------------------------
[ 500 ] active [ N5N30_POLAUTH ] sessions on node [ http://10.200.71.131:7091/ ].

# # # # # # # # # # # # # # # # # # # # # # #
```

Only the N5N30 sessions removed for specific NEF/AF peer node with confirmation.

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http:// 10.200.71.131:7091/ --action delete --se  →
ssionType N5N30_POLAUTH

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [http://10.200.71.131:7091/]
---------------------------------------------
Session cleanup for node [ http://10.200.71.131:7091/ ] is ongoing, [ 500 ] active [ N5N30_POLAUT  →
H ] sessions should be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # #

Deletion requires confirmation...
Current deletion with specific option(s) '--peerNode http://10.200.71.131:7091/@5G
Do you want to proceed with the deletion? [y/n]
y
```

This output demonstrates how the tool shows the number of N5N30 sessions to be deleted and asks the user for confirmation. When the user confirms the deletion, all related N5N30 sessions are removed from the DBS.

Example 12 Show and remove all Gx sessions for a PCEF peer node identified with host@realm

Show the number of the sessions for the specific PCEF node 'esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusrealmfor5g.com'

```
sapcadmin@PL-3:~> sudo session-handler --peerNode esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusr →
ealmfor5g.com --action show --sessionType Gx

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusre →
almfor5g.com]
---------------------------------------------
[ 500 ] active [ Gx ] sessions on node [ esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusrealmfor5g →
.com ].

# # # # # # # # # # # # # # # # # # # # # #
```

Remove the sessions created for specific PCEF node 'esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusrealmfor5g.com'

```
sapcadmin@PL-3:~> sudo session-handler --peerNode esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusr →
ealmfor5g.com --action delete --sessionType Gx

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusre →
almfor5g.com]
---------------------------------------------
Session cleanup for node [ esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusrealmfor5g.com ] is ongo →
ing, [ 500 ] active [ Gx ] sessions should be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # #
```

This output demonstrates how the tool shows the number of Gx sessions to be deleted and asks the user for confirmation. When the user confirms the deletion, it removes all the related Gx sessions and binding Rx, Sd, Smp, and Sy sessions.

Example 13 Show and remove all SMP sessions for a SGSN-MME node by peerId ‘mme0100.sxrealm.com@sxrealm.com’

Show the number of the SMP sessions with peerId ‘mme0100.sxrealm.com@sxrealm.com’

```
sapcadmin@PL-3:~> sudo session-handler --peerNode mme0100.sxrealm.com@sxrealm.com --sessionType Sm →
p --action show

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [mme0100.sxrealm.com@sxrealm.com]
---------------------------------------------
```

```
[ 100 ] active [ Smp ] sessions on node [ mme0100.sxrealm.com@sxrealm.com ].

# # # # # # # # # # # # # # # # # # # # # # # #
```

Delete related number of SMP sessions without confirm

```
sapcadmin@PL-3:~> sudo session-handler --peerNode mme0100.sxrealm.com@sxrealm.com --sessionType Sm →
p --action delete --noConfirm

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [mme0100.sxrealm.com@sxrealm.com]
---------------------------------------------
Session cleanup for node [ mme0100.sxrealm.com@sxrealm.com ] is ongoing, [ 100 ] active [ Smp ] se →
ssions should be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # # #
```

This output demonstrates how the tool shows the number of Smp sessions to be deleted and without asking the user for confirmation, all related Smp sessions are removed from the DBS.

Example 14 Show and remove all AF sessions (Rx + N5N30) bounded with specific SMF node (N7 session)

Show the number of the AF sessions include Rx and N5N30 session bounded with specific SMF node identified by SMF Ip:port

```
sapcadmin@PL-3:~> sudo session-handler --peerNode https://10.200.71.131:7072/ --sessionType N7 --a →
ction show --noConfirm –onlyAF

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [https://10.200.71.131:7072/]
---------------------------------------------
[ 70 ] active [ AF ] sessions on node [ https://10.200.71.131:7072/ ].

# # # # # # # # # # # # # # # # # # # # # # # #
```

Remove all the AF sessions include Rx and N5N30 session bounded with specific SMF node identified by SMF Ip:port

```
sapcadmin@PL-3:~> sudo session-handler --peerNode https://10.200.71.131:7072/ --sessionType N7 --a →
ction delete --noConfirm –onlyAF

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [https://10.200.71.131:7072/]
---------------------------------------------
Session cleanup for node [ https://10.200.71.131:7072/ ] is ongoing, [ 70 ] active [ AF ] session →
s should be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # # #
```

This output demonstrates how the tool shows the number of AF sessions include Rx and N5N30_POLAUTH sessions bound with specific SMF peer. Then, without asking the user for confirmation, it removes all the related AF sessions (Rx, N5N30_POLAUTH) only,no impact for N7 sessions.

Example 15 Show and remove all AF sessions (Rx) bounded with specific PCEF node (Gx session)

Show the number of the Rx session bounded with specific PCEF node identified by peerId

```
sapcadmin@PL-3:~> sudo session-handler --peerNode esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusr
ealmfor5g.com --sessionType Gx --action show --onlyAF

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusre
almfor5g.com]
---------------------------------------------
[ 158 ] active [ AF ] sessions on node [ esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusrealmfor5g
.com ].

# # # # # # # # # # # # # # # # # # # # # # #
```

Remove all the Rx sessions bounded with specific PCEF node identified by peerId

```
sapcadmin@PL-3:~> sudo session-handler --peerNode esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusr
ealmfor5g.com --sessionType Gx --action delete --onlyAF

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusre
almfor5g.com]
---------------------------------------------
Session cleanup for node [ esmdx0900.gxrel10plusrealmfor5g.com@gxrel10plusrealmfor5g.com ] is ongo
ing, [ 158 ] active [ AF ] sessions should be cleaned, [ 0 ] sessions have been cleaned.

# # # # # # # # # # # # # # # # # # # # # # #

Deletion requires confirmation...
Current deletion with specific option(s) '--peerNode esmdx0900.gxrel10plusrealmfor5g.com@gxrel10pl
usrealmfor5g.com
Do you want to proceed with the deletion? [y/n]
y
```

This output demonstrates how the tool shows the number of AF sessions include Rx sessions bound with specific PCEF peer. Then, without asking the user for confirmation, it removes all the related Rx sessions only, no impact for Gx sessions.

Example 16 Show and remove N15_UEPC sessions for an AMF peer node identified with host@realm

Show N15_UEPC sessions for an AMF peer node identified with host@realm

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://192.168.14.42:7083/ --sessionType N15_UEP
C --action show

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
```

```
Session data model corresponding to peerNode -> [http://192.168.14.42:7083/]
---------------------------------------------
[ 1 ] active [ N15_UEPC ] sessions on node [ http://192.168.14.42:7083/ ].


# # # # # # # # # # # # # # # # # # # # # #
```

Remove N15_UEPC sessions for an AMF peer node identified with host@realm

```
sapcadmin@PL-3:~> sudo session-handler --peerNode http://192.168.14.42:7083/ --sessionType N15_UEP →
C --action delete

Initializing DBN API
Waiting for DBN (false)...
---------------------------------------------
Session data model corresponding to peerNode -> [http://192.168.14.42:7083/]
---------------------------------------------
Session cleanup for node [ http://192.168.14.42:7083/ ] is ongoing, [ 1 ] active [ N15_UEPC ] sess →
ions should be cleaned, [ 0 ] sessions have been cleaned.


# # # # # # # # # # # # # # # # # # # # # #

Deletion requires confirmation...
Current deletion with specific option(s) '--peerNode http://192.168.14.42:7083/
Do you want to proceed with the deletion? [y/n]
y
```

2.7 sapcSessionCollector

This command collects session data from dynamic Persistent Object Types (POTs) and saves the output files in the `/cluster/brf/backup/sessions` folder.

The supported dynamic POTs are EPC_UeIpSessionPot, EPC_IpSessionPot, AfSessionPot, SubsChargingDataPot, SdSessionPot and SmpSessionPot.

The sapcadmin can use this command:

`sapcadmin@SC-1:~>` **`sudo sapcSessionCollector -h`**

```
Usage: sapcSessionCollector [-h] [-a] [-i POT name [POT
name ...]] [-s size]
```

```

  -h, --help            show this help message and exit
  -a, --all             Collects all supported POTs.
  -i POT name [POT name ...]
                        The supported types of POT include AfSes →
sionPot,
                        EPC_IpSessionPot, EPC_UeIpSessionPot, Sd →
SessionPot,
                        SmpSessionPot and SubsChargingDataPot.
  -s size               Defines the maximum size for each file, →
(unit is
                        megabytes), data will be split into seve →
```

```
ral files if
                        it is too large. (default=10M)
```

Example 17 Collecting data from EPC_UeIpSessionPot

```
sudo sapcSessionCollector -i EPC_UeIpSessionPot
```

Example 18 Collecting data from all supported POTs

```
sudo sapcSessionCollector -a
```

Example 19 Collecting data from EPC_UeIpSessionPot and EPC_IpSessionPot

The file size is 20M. The different POTs are separated by a space.

```
sudo sapcSessionCollector -i EPC_UeIpSessionPot EPC_IpSessionPo  →
t -s 20
```

Output:

```
PL-3 will be used to retrieve the session information:
EPC_UeIpSessionPot: 100%|####################################### →
################################################################ →
################################################| [01:02<00:00,  →
1.77Session/s]
[EPC_UeIpSessionPot] has been written to /cluster/brf/backup/ses →
sions/EPC_UeIpSessionPot_20181115-040608_0.txt
PL-3 will be used to retrieve the session information:
EPC_IpSessionPot: 100%|######################################### →
################################################################ →
################################################| [00:30<00:00,  →
3.23Session/s]
[EPC_IpSessionPot] has been written to /cluster/brf/backup/sessi →
ons/EPC_IpSessionPot_20181115-040712_0.txt
```

2.8 Packet Capture Tool

The Capture tool is a command-line Packets Capture function which is used to capture traffic from the live PL nodes of the SAPC. The generated capture files are saved in a standard PCAP format file and stored in the SC node.

The Capture Tool continues capturing packets until certain condition is reached or terminated by operator. A tar.gz file is created after the capturing stopped and is stored in the same directory.

Limitation:

- Only can run one instance of Capture Tool at a time.
- Only support non-promiscuous mode of capture operation.
- Can not use the Capture Tool during upgrade or configuration change.
- IPsec protected traffic is not decrypted.
- No redundancy function provided by the capture tool.
- Has no impact on the tcpdump process that started by other tools.

Extension tools

The PCAP files can be processed and analyzed offline by using a widespread third party tool, such as tcpdump or ethereal (WireShark).

2.8.1 Starting Packet Capture

Prerequisites

It is recommended to estimate the number of the required packets and execute the command line with the appropriate flags based on the expected packets to avoid bandwidth or storage overload.

Steps

1. Run the command on one of the SC node by sapcadmin:

   ```
   >sudo sapcCaptureTool [-PL List] [-g percentage] [-m time] [-b <true|false>] [-c N] [port port-number [[and port-number] | [or port port-number]*]] [...]
   ```

Parameters	Description
`-PL` *`<PL list>`*	The name of the PL node. Multiple values are separated by comma (,). If not given, the Capture Tool runs on all the PL nodes. Value format: PL-n For example: PL-3, PL-4
`-g percentage`	The percentage of storage that capture files can occupy on the SC storage. The system terminates the

Parameters	Description
	capturing on all PL nodes after the percentage reaches. Value range: `1~100` For example: `50`
`-m time`	The time length in second of capturing the traffic. The system terminates the capturing on all PL nodes after the given time is reached. Value range: 1~n For example: `100`
`-b`	The behavior when error occurs on one or more PL nodes. Value range: true or false Default value: true True - indicates that when the error occurs on one or more PL nodes, the system terminates the capture process on all nodes. False - indicate that when the error occurs on one or more PL nodes, the system terminates the capturing on the failed nodes.
`-c N`	The number of the packets that are planned to be captured for each PL node. The system terminates the capturing on the PL node that captures the defined number of packets.
`-h or --help`	Show help message.
`-f`	Terminate the capturing from a terminal other than the terminal that initiated the sapcCaptureTool process, no zip file will be created.
...	Additional tcpdump parameters, such as filter expressions.

Parameters	Description
	Note: — `- w file` flag is not supported because the file name is defined as *<timestamp>*/PL-*<n>*. — If the port flag is not defined, the Capture Tool uses the default port diameter.

Result: The output file is stored in `/cluster/storage/captureTool/` of the SC node and the file name is the created time of the files plus the name of PL node. For example, `20190513_070825/PL-3.pcap`.

Example 20

```
sapcadmin@SC-1:~>sudo sapcCaptureTool -m 10 -PL PL-3,PL-4 -c 100
```

```
2019-05-20 04:03:49,869 - INFO - 431 - sapcCaptureTool - tcpdum →
p start in PL-3.
2019-05-20 04:03:49,876 - INFO - 431 - sapcCaptureTool - tcpdum →
p start in PL-4.
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), c →
apture size 262144 bytes
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), c →
apture size 262144 bytes
```

Example 21

```
sapcadmin@SC-1:~> sudo sapcCaptureTool port 8086 and port 7072
2021-11-29 08:47:51,474 - INFO - 442 - sapcCaptureTool - Tcpdum →
p starts in PL-3.
2021-11-29 08:47:51,477 - INFO - 442 - sapcCaptureTool - Tcpdum →
p starts in PL-4.
sapcCaptureTool: listening on any, link-type LINUX_SLL (Linux co →
oked), capture size 262144 bytes
sapcCaptureTool: listening on any, link-type LINUX_SLL (Linux co →
oked), capture size 262144 bytes

sapcadmin@SC-1:~> sudo sapcCaptureTool port 8086 or port 9091 or por →
t 7072
2021-11-29 08:44:55,386 - INFO - 442 - sapcCaptureTool - Tcpdum →
p starts in PL-3.
2021-11-29 08:44:55,393 - INFO - 442 - sapcCaptureTool - Tcpdum →
p starts in PL-4.
sapcCaptureTool: listening on any, link-type LINUX_SLL (Linux co →
oked), capture size 262144 bytes
```

```
sapcCaptureTool: listening on any, link-type LINUX_SLL (Linux co →
oked), capture size 262144 bytes
```

2.8.2 Stopping Packet Capture

The Capture Tool can be terminated in the following ways:

- Stop capturing by pressing Ctrl+C. The Capture Tool is stopped on all PL nodes and a tar.gz file is created.
- Stop capturing by running `sapcCaptureTool` command with `-f` flags from other terminal and system does not zip the `.pcap` files.
- Stop automatically when the defined condition is reached. For more information, see Starting Packet Capture on page 27.

The following is an example of terminating the running sapcCaptureTool by `-f` flag.

Steps

1. Run the command from a terminal other than the terminal that initiated the sapcCaptureTool process:

   ```
   sapcadmin@SC-1:~>sudo sapcCaptureTool -f
   ```

   ```
   2019-05-16 10:38:12,698 - INFO - 492 - sapcCaptureTool - ====== Arguments Pars→
   ing ======
   2019-05-16 10:38:12,698 - INFO - 500 - sapcCaptureTool - Pre check capture too→
   l instance...
   2019-05-16 10:38:14,504 - WARNING - 96 - sapcCaptureTool - Tcpdump already exi→
   sts!
   2019-05-16 10:38:14,969 - WARNING - 102 - sapcCaptureTool - Capture tool alrea→
   dy exists!
   2019-05-16 10:38:16,715 - INFO - 107 - sapcCaptureTool - Force to quit...
   ```

2.9 sync-provisioning

This script is used to synchronize a non-up-to-date remote SAPC peer with the provisioning data of the local SAPC in N+1 Geographical Redundancy deployments. This misalignment can be caused by provisioning replication errors or by Dynamic Subscriber OSI update replication errors.

The script reads all provisioning and Dynamic Subscriber OSI update error files of the SAPC peer stored in the node and updates the failed REST resources in the SAPC peer with the content of the local REST resource.

2.9.1 Error Files and Management

The error files are named according to the following schema: *<GEO_ID>_<PL>_<ERROR_TYPE>*.

<GEO_ID> Indicates the SAPC peer where the replication failed.

<PL> Indicates the traffic payload in the local SAPC where the replication operation failed.

<ERROR_TYPE> Indicates the type of replication error:

- `proverr` for Provisioning replication errors (see N+1 Geographical Redundancy on page 45)
- `subserr` for Dynamic Subscriber OSI update replication errors (see Dynamic Subscriber OSI Update in N+1 Geographical Redundancy Failures on page 69)

The error file content is a set of resources that were not correctly replicated in the SAPC peer.

Error files are in the directory `/storage/no-backup/sapc/georedFiles/provisioningErrorFiles`.

To manage the provisioning error files, the SAPC provides a file group called `PolicyGeoRedProvisioningErrorFiles`, which is configurable through COM cliss interface.

To configure the maximum number of error files, and the maximum size of the directory:

Steps

1. Execute the following command:

   ```
   /opt/com/bin/cliss
   ```

2. Check the current configuration for `PolicyGeoRedProvisioningErrorFiles`.

   ```
   > show ManagedElement=1,SystemFunctions=1,FileM=1,FileGroupPo →
   licy=PolicyGeoRedProvisioningErrorFiles

   FileGroupPolicy=PolicyGeoRedProvisioningErrorFiles

      fileGroup

   "ManagedElement=1,SystemFunctions=1,FileM=1,LogicalFs=1,FileG →
   roup=georedFiles,FileGroup=provisioningErrorFiles"

      maxFileGroupSize=200000

      maxNumberFiles=1000
   ```

3. To modify the number of maximum log files before rotation, configure the `PolicyGeoRedProvisioningErrorFiles`.

   ```
   >configure

   (config)> ManagedElement=1,SystemFunctions=1,FileM=1,FileGro →
   upPolicy=PolicyGeoRedProvisioningErrorFiles,maxNumberFiles=20 →
   0
   ```

4. To modify the maximum size of log files, configure the `PolicyGeoRedProvisioningErrorFiles`.

   ```
   >configure

   (config)> ManagedElement=1,SystemFunctions=1,FileM=1,FileGrou →
   pPolicy=PolicyGeoRedProvisioningErrorFiles,maxFileGroupSize=1 →
   000000
   ```

5. Commit the changes.

   ```
   (config)>commit
   ```

6. Check the configuration.

   ```
   > show ManagedElement=1,SystemFunctions=1,FileM=1,FileGroupPo →
   licy=PolicyGeoRedProvisioningErrorFiles

   FileGroupPolicy=PolicyGeoRedProvisioningErrorFiles

      fileGroup

   "ManagedElement=1,SystemFunctions=1,FileM=1,LogicalFs=1,FileG →
   roup=georedFiles,FileGroup=provisioningErrorFiles"

      maxFileGroupSize=100000

      maxNumberFiles=200
   ```

2.9.2 Script Description

The script reads the failed resource URIs one by one from the error files, gets the resource data from the local SAPC, and:

- If the resource exists, executes the `HTTP PUT` operation over that resource in the SAPC peer.
- If the resource is not found in the local SAPC, the script executes an `HTTP DELETE` operation on the resource in the SAPC peer.

The possible results are the following:

- If the resource is successfully updated in the peer, the consistency of this resource is guaranteed between SAPCs.
- If the resource is not successfully updated in the peer, a new entry is logged in a *<GEO_ID>*_SYNC_*<ERROR_TYPE><timestamp>* file, and the related alarm is not cleared. This file is also processed by the `sync-provisioning` script in its next execution.

2.9.3 Script Execution

The script runs in any traffic payload processor, according to the following use:

```
sapcadmin@PL-3:~> sync-provisioning
Usage sync-provisioning -g <geoid> -u <user:password> [ -t <prov →
err|subserr> ] [ -v ]
  -g <geoid>:           Geored id for which provisioning sync i  →
s launched.
  -u <user:password>:   REST SAPC provisioning username and pass →
word.
  -t <error type>:      Type of error to synchronize. Single opt →
ion allowed. Options: proverr|subserr
  -v:                   Verbose output and log written to sync-p →
rovisioning.log file
```

The script must be invoked including the neighbor SAPC peer identity, and the SAPC user and password used for provisioning operations.

Note:
- The default user name for provisioning is `sapcprov`.
- All SAPC peers must have the same SAPC user and password for provisioning operations.

A specific synchronization is allowed by setting the `-t` option and the corresponding error type. If not specified, synchronization for every error type is performed.

The verbose (`-v`) option shows details about every operation carried out on every resource, for example, operation type, URI, result-code, and operation time.

The information is logged in the `/storage/no-backup/sapc/georedFiles/provisioningErrorFiles/sync-provisioning.log` file.

After executing the script, the output on the screen shows the summary of the operations carried out.

If synchronization is successful, the related SAPC alarm is cleared.

Specific execution examples can be found in N+1 Geographical Redundancy on page 45 and Dynamic Subscriber OSI Update in N+1 Geographical Redundancy Failures on page 69 sections.

3 Troubleshooting Functions

3.1 Linux Consoles

The Linux Console is accessed using the SSH protocol towards the System Controller processors using the **sapcadmin** user through the **<OAM VIP>**. For more details, refer to System Administrator Guide. If the **<OAM VIP>** is unavailable, the operation and maintenance scripts cannot be used. SeeOAM Interface Unavailable on page 67.

For more information about available commands, check the LDE Management Guide.

3.2 Internal Database Command Line Tools

The internal database command line tools can provide useful information about database status. These tools offer data on a cluster level, like the status of the processors that form the SAPC and some other information regarding memory consumption and internal connections. To execute every tool, use the command `clurun.sh` from any of the SC or PL processors.

```
sapcadmin@SC-1:~> clurun.sh
```

3.3 COM CLI

This console provides a direct CLI for the COM subsystem and also a textual representation of Management Information Model (MIM). For more details, refer to System Administrator Guide.

3.4 System Health Check

To check if the SAPC is working properly, the sapcHealthCheck is used. It provides information about:

- SAPC Software components installed
- Nodes communication through TIPC
- DRBD status
- CMW and AMF status
- Alarms, coredumps, and error logs in the system

— DBS status

The next example shows the script output using default options for a succeed state:

```
sapcadmin@SC-X:~> sudo sapcHealthCheck

==================== HEALTH CHECK REPORT ====================

Checking the SAPC is installed...
SAPC installation --> OK --> All the 29 ERIC-SAPC SDPs installed →
are used [main version: ERIC-SAPC-CXP9030138_7-R2E73].

Checking TIPC communication...
TIPC --> OK --> All the 4 available nodes at TIPC level are up.

Checking DRBD devices...
DRBD device --> OK

Checking CMW status...
CMW status --> OK --> All the "node comp app su si sg siass csias →
s pm" are OK (Stopped PM jobs are ignored).

Checking AMF status...
AMF status --> OK --> All the AMF entities are OK.

Checking active FM alarms...
Alarms --> OK --> There are no active FM Alarms.

Checking existing coredumps...
Coredumps --> OK --> There are no coredumps.

Checking system operative...
External peers configured and in use --> OK

Checking Data Base...
All Data Base agents working normally --> OK

Checking error logs in the system...
No errors in the system --> OK

*** SAPC HEALTH CHECK SUMMARY ***
 WARNINGS:  0
 ERRORS:    0
**********************************

SAPC Health Check finished: OK
```

3.5 Processor Load (CPU and Memory)

The specific commands to check that CPU and Memory load are described in Preventive Maintenance.

3.6 Alarms and Notifications

For information about how to check system alarms, refer to Preventive Maintenance.

If any alarm is raised, act on the corresponding OPI to make it cease.

3.7 Logging

For further information, about the Logging events generated by the SAPC, refer to Logging Events.

3.8 Measurements

The traffic measurements generated by the SAPC also provide useful information when troubleshooting a problem. For more information, refer to Measurements.

3.9 Core Files

To check the existence of system core files, refer to Preventive Maintenance.

If any, send then to the next level of maintenance support for analysis.

3.10 System Messages

Important information about the general status of the different processors can be found as root user in the following files found in any SC processor:

`root@<SC-X>:/var/log/<node-id>/auth*`

`root@<SC-X>:/var/log/<node-id>/kernel*`

`root@<SC-X>:/var/log/<node-id>/messages*`

Note: Where `<SC-X> is SC-1 or SC-2`

Where `<node-id> is SC-1, SC-2, PL-3, PL-4 or PL-n`

3.11 SAPC Reboot

The SAPC can be reloaded with the following commands.

Caution!

The procedure implies almost 30-seconds downtime until the internal database is operating again.

Steps

1. Log on to the system with **sapcadmin** user, through *<OAM VIP>* .
2. Perform the reboot of the SAPC.

   ```
   sapcadmin@SC-X> sudo cmw-cluster-reboot <[--yes]>
   ```

   If `--yes` is specified, the command does not require confirmation.
3. Wait until the node is back.
4. Log on again to the system with **sapcadmin** user, through *<OAM VIP>* .
5. Check the status of the node according to Preventive Maintenance.

3.12 Processor Lock and Unlock

A processor on the SAPC can be locked from the node. A processor locked means that it is not part of the cluster, until the unlocked command is performed (the processor comes back to the node).

Steps

1. Log on to the system with **sapcadmin** user, through *<OAM VIP>* .
2. Perform the lock of the processor in the SAPC.

   ```
   sapcadmin@SC-X> sudo cmw-node-lock <processor>
   ```

   Caution!

   Traffic performance can be affected until the processor is unlocked.
3. To get the processor back on the node, execute the following command:

   ```
   sapcadmin@SC-X> sudo cmw-node-unlock <processor>
   ```

4. Check the status of the node according to Preventive Maintenance.

4 Troubleshooting Procedure

Troubleshooting a problem in the SAPC requires the use of one or more functions described in the previous chapters. The correct use of these tools is needed to prevent overload situations. In a faulty situation, they must be used in the right order to ensure an efficient location of the fault:

Steps

1. Perform a System Health Check described on System Health Check on page 35.
2. Check processors load. See Processor Load (CPU and Memory) on page 37.
3. Check for alarms in the system. To do that, follow Alarms and Notifications on page 37.
4. Check for logs in the system. To do that, follow Logging on page 37.
5. Check the traffic measurements. See Measurements on page 37.
6. Check the capacity measurements and purchased capacity licenses. See Measurements on page 37.
7. Check system core dumps files, follow Core Files on page 37.
8. Check system messages. See System Messages on page 37.

   A troubleshooting workflow is shown in Figure 1.

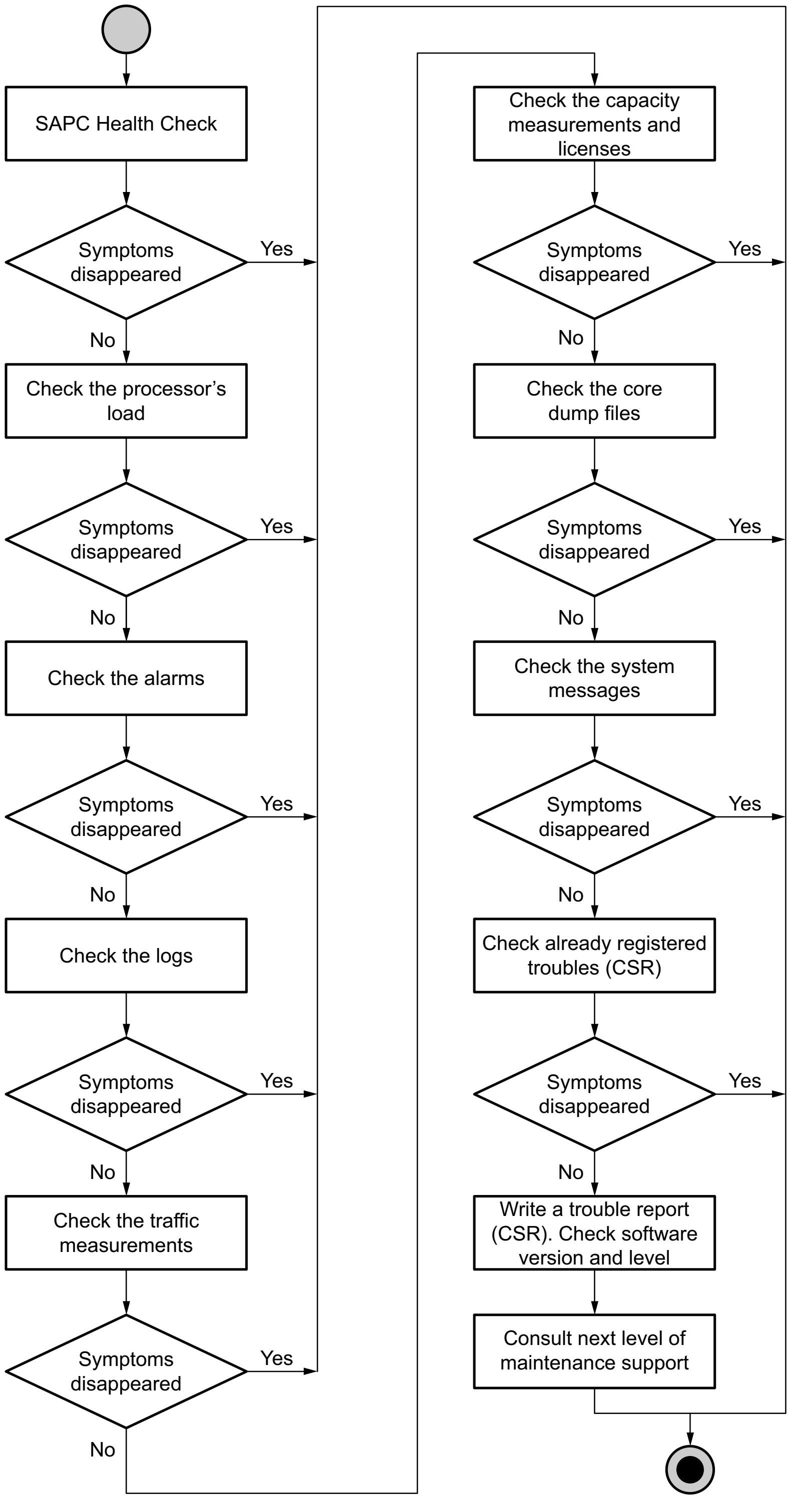
SAPC Health Check
Symptoms disappeared
Yes
No
Check the processor's load
Symptoms disappeared
Yes
No
Check the alarms
Symptoms disappeared
Yes
No
Check the logs
Symptoms disappeared
Yes
No
Check the traffic measurements
Symptoms disappeared
Yes
No
Check the capacity measurements and licenses
Symptoms disappeared
Yes
No
Check the core dump files
Symptoms disappeared
Yes
No
Check the system messages
Symptoms disappeared
Yes
No
Check already registered troubles (CSR)
Symptoms disappeared
Yes
No
Write a trouble report (CSR). Check software version and level
Consult next level of maintenance support

Figure 1 Troubleshooting Workflow

5 Common Faulty Situations

In the following sections, some common problems and possible solutions are described that can appear during normal operation.

5.1 General Failures

5.1.1 License Is Not Active

The Soft-Limit Behavior allows traffic to be normally processed by the SAPC even if a license is expired or if its capacity is exceeded, following the soft-lock principle: alarms and reports are issued but the node service is not restricted. However, if traffic is continuously rejected or misprocessed, it can be caused by a license which is not properly installed, expired, or whose capacity is exceeded. If a license has expired, contact supply organization to request an extension.

Steps

1. Check License Manager status and configuration. Refer to View License Information.

2. Check for specific alarms regarding License Manager or capacity licenses. Refer to SAPC License Management.

3. If there are active alarms about capacity licenses, check the capacity measurements (see Measurements on page 37) and purchased capacity licenses.

4. If a license is not properly activated or installed, reinstall licenses. Refer to Install License Key File.

5. If a license has expired, contact supply organization to request an extension.

6. To restore system functionality temporarily in extraordinary situations, activate the Emergency Unlock mode as described in Activate Emergency Unlock Mode. Consider that the number of activations is limited.

5.1.2 Processor Is Out of Service

The SAPC is composed of several processors. If, during operation, any processor goes out of service, the rest of the traffic processors must handle all the traffic, so it can result in a higher load situation for them. To verify and correct the situation, follow the next step:

Steps

1. Check if the SAPC platform component status is correct following the steps described in System Health Check on page 35.

5.1.3 Load Regulation

The SAPC continuously monitors CPU usage load and memory consumption. If the values of these parameters exceed a configurable threshold, the SAPC rejects requests in Gx , Rx, Smp, Sy, N7, N15, N5N30, Nnrf and Nudr interfaces. It can also reject REST provisioning messages (with HTTP 503 error), SOAP notifications (with HTTP 500 error), and time triggered reauthorizations. This is to guarantee a graceful behavior of the SAPC in overload situations. If DIAMETER_TOO_BUSY, HTTP 503, or HTTP 500 messages are detected continuously during a prolonged period:

Steps

1. Follow the System Health Check described in System Health Check on page 35. If one or more nodes are not working properly, the rest of them can be in an overload situation.

2. Check the value for Load Regulation constraints as described in the Overload Control User Guide User Guide. Adjust the values to the manufacturer recommendation.

3. If none of the previous actions detected a malfunction or erroneous configuration, it is most likely that the SAPC is applying load regulation because of high resource consumption. If this situation persists over time, contact the next level of maintenance support.

5.1.4 User Management Aggregation not Started

If the User Management Aggregation is not started after a deployment with a high number of Payloads, that is 2+60 for example, the following fail appears in `sapcHealthCheck`:

```
Checking CMW status...
CRITICAL ERROR --> The CMW status is NOT OK.
CMW Status result: safComp=sapc.usermgmt.aggregation,safSu=SC-1,s →
afSg=2N,safApp=ERIC-sapc.usermgmt.aggregation
OperState=ENABLED(1)
PresenceState=UNINSTANTIATED(1)
ReadinessState=OUT-OF-SERVICE(1)
safSu=SC-1,safSg=2N,safApp=ERIC-sapc.usermgmt.aggregation
AdminState=LOCKED(2)
OperState=ENABLED(1)
PresenceState=UNINSTANTIATED(1)
ReadinessState=OUT-OF-SERVICE(1)
```

```
safSi=sapc.usermgmt.aggregation-2N-1,safApp=ERIC-sapc.usermgmt.ag →
gregation
AdminState=UNLOCKED(1)
AssignmentState=PARTIALLY_ASSIGNED(3)
```

To start the User Management Aggregation in an SC after the deployment, use the following command:

```
sapcadmin@SC-1:~> sapcUserMgmt -a start -p SC-X
```

SC-X is the SC stated in the `CMW Status result` line of `sapcHealthCheck`.

Note: This command must be executed as root or sapcadmin user. It cannot be executed with the sapctroubleshooter user.

5.2 Provisioning Failures

Representational State Transfer (REST) is used as an interface for the SAPC provisioning purpose. Through REST services commands, it is possible to provision the SAPC with Subscribers, Subscribers Groups, and Policies.

— If REST provisioning is not successful:

  Verify that the provisioned information is correct according to the following documents:

  - Configuration Guides
  - Database Access
  - Integration in Service Chaining
  - Provisioning REST API

— If the internal database does not accept more provisioning entries:

  Check that the database storage capacity limit is not reached.

  To verify this, launch the following command that shows the amount of used and free memory:

```
sapcadmin@SC-1:~> clurun.sh collect_stats -d dbn

Result from [PL-3.dbn]: DbsService=DBN,DbsPU=PL-3
DbsPU.VS.DBS.Mem.NormalHeap.Free 0
DbsPU.VS.DBS.Mem.NormalHeap.Used 61889
DbsPU.VS.DBS.Mem.RecordHeap.Free 3276215
DbsPU.VS.DBS.Mem.RecordHeap.PUsed 0
DbsPU.VS.DBS.Mem.RecordHeap.Used 0
DbsPU.VS.DBS.Mem.TotalHeap.Free 3276215
DbsPU.VS.DBS.Mem.TotalHeap.Used 61889
```

```
...
Result from [PL-4.dbn]: DbsService=DBN,DbsPU=PL-4
DbsPU.VS.DBS.Mem.NormalHeap.Free 0
DbsPU.VS.DBS.Mem.NormalHeap.Used 61978
DbsPU.VS.DBS.Mem.RecordHeap.Free 3276215
DbsPU.VS.DBS.Mem.RecordHeap.PUsed 0
DbsPU.VS.DBS.Mem.RecordHeap.Used 0
DbsPU.VS.DBS.Mem.TotalHeap.Free 3276215
DbsPU.VS.DBS.Mem.TotalHeap.Used 61978
...
```

Caution!

If there is no total heap available, contact Ericsson personnel for more information.

5.2.1 N+1 Geographical Redundancy

In an N+1 Geographical Redundancy deployment, all the provisioning operations executed in the SAPC selected as the primary provisioning node by the customer provisioning system are replicated automatically in the rest of the SAPC peers of the N+1 cluster. If this replication fails, the cluster has an inconsistent status.

When a replication operation fails, the primary provisioning node, which is trying to replicate the provisioning resource, logs the failing resource to a provisioning error file in the following directory:

```
/storage/no-backup/sapc/georedFiles/provisioningErrorFiles
```

The provisioning error file is named according to following schema:

```
<GEO_ID>_<PL>_proverr
```

An example of provisioning error file content is the following:

```
PL-3:/storage/no-backup/sapc/georedFiles/provisioningErrorFiles
# cat SAPC2_PL-3_proverr

/provisioning/v1/subscribers/222222000000

/provisioning/v1/subscribers/222222000001

/provisioning/v1/subscribers/222222000002

/provisioning/v1/subscribers/222222000003

/provisioning/v1/subscribers/222222000004
```

In order to synchronize the provisioning data in the SAPC peer, the REST resources stored in those files must be updated using the sync-provisioning script. Specific alarm Policy Control, Geographical Redundancy Provisioning Failure is cleared in case of a successful synchronization.

For more information about using sync-provisioning script, see sync-provisioning on page 30.

Some examples of handling with provisioning errors are listed below. For demonstrative purposes, only provisioning synchronization is shown (`-t proverr` option used).

Example 22 Successful Sync-Provisioning Execution

```
sapcadmin@PL-3:~> sync-provisioning -g SAPC2 -u sapcprov:<password> -t proverr
-------------------------------------------------
Processing proverr files ...
-------------------------------------------------


Processing 1 files, total 5 resources

Total 5 unique resources

Start processing

Processed 5 resources

Synchronization successful!

No active provisioning errors: Clearing Provisioning Failure Alarm...
```

Example 23 Unsuccessful Sync-Provisioning Execution due to a Failed Authentication

```
Example of unsuccessful sync-provisioning execution, in this case due to a failed authentication:

sapcadmin@PL-3:~> sync-provisioning -g SAPC2 -u user_unknown:<password> -t proverr
-------------------------------------------------
Processing proverr files ...
-------------------------------------------------

Processing 1 files, total 53 resources

Total 5 unique resources

Start processing
```

```
Error executing internal replicate_resources.py script. Use -v f →
or more information

Synchronization failed! 5 errors found
```

Example 24 Sync-Provisioning Execution in Verbose Mode

```
sapcadmin@PL-3:~> sync-provisioning -g SAPC2 -u sapcprov:<passwo →
rd> -t proverr -v
--------------------------------------------------
Processing proverr files ...
--------------------------------------------------



Processing 1 files, total 5 resources

Total 5 unique resources

Start processing

('GET', '/provisioning/v1/subscribers/222222000000', 200, 0.2389 →
63)

('PUT', '/provisioning/v1/subscribers/222222000000', 201, 0.2451 →
8700000000004)

('GET', '/provisioning/v1/subscribers/222222000001', 200, 0.1048 →
6)

('PUT', '/provisioning/v1/subscribers/222222000001', 201, 0.2456 →
91)

('GET', '/provisioning/v1/subscribers/222222000002', 200, 0.0009 →
22)

('PUT', '/provisioning/v1/subscribers/222222000002', 201, 0.0016 →
36)

('GET', '/provisioning/v1/subscribers/222222000003', 200, 0.0006 →
02)

('PUT', '/provisioning/v1/subscribers/222222000003', 201, 0.0012 →
5)

('GET', '/provisioning/v1/subscribers/222222000004', 200, 0.0005 →
98)

('PUT', '/provisioning/v1/subscribers/222222000004', 201, 0.0011 →
97)
```

```
Processed 5 resources

Synchronization successful!

No active provisioning errors: Clearing Provisioning Failure Alarm...



sapcadmin@PL-3:~> cat /storage/no-backup/sapc/georedFiles/provisioningErrorFiles/sync-provisioning.log

Mon Jul 15 16:50:43 CEST 2019
-------------------------------------------------
Processing proverr files ...
-------------------------------------------------


Processing 1 files, total 5 resources

Total 5 unique resources

Start processing

('GET', '/provisioning/v1/subscribers/222222000000', 200, 0.238963)

('PUT', '/provisioning/v1/subscribers/222222000000', 201, 0.24518700000000004)

('GET', '/provisioning/v1/subscribers/222222000001', 200, 0.10486)

('PUT', '/provisioning/v1/subscribers/222222000001', 201, 0.245691)

('GET', '/provisioning/v1/subscribers/222222000002', 200, 0.000922)

('PUT', '/provisioning/v1/subscribers/222222000002', 201, 0.001636)

('GET', '/provisioning/v1/subscribers/222222000003', 200, 0.000602)

('PUT', '/provisioning/v1/subscribers/222222000003', 201, 0.00125)

('GET', '/provisioning/v1/subscribers/222222000004', 200, 0.000598)

('PUT', '/provisioning/v1/subscribers/222222000004', 201, 0.0011
97)

Processed 5 resources

Synchronization successful!

No active provisioning errors: Clearing Provisioning Failure Ala →
rm...
```

5.3 Failures during the Initial Configuration and Provisioning in Cloud

Optionally, the SAPC can be automatically configured and provisioned in Cloud deployments. During deployment time, the files containing the data are injected to the SAPC and the **auto_provision** script is executed (to find more details, refer to SAPC VNF Descriptor Generator Tool).

In some scenarios, in which the Cloud Infrastructure where the SAPC is deployed presents a slow connection speed, this automatic procedure could not have been executed.

Steps

1. Confirm if the script for automatic configuration and provisioning is successfully executed, looking at the contents of the following log file in SC-1:

   ```
   sapcadmin@SC-1:~> cat /var/log/auto_provision/
   auto_provision.log
   ```

2. Instead, if the last text message is **`"Waiting NDB nodes to be STARTED ..."`**, the execution fails. Then, do it manually from SC-2:

   ```
   sapcadmin@SC-2:~> /usr/local/bin/auto_provision
   ```

5.4 Fair Usage Reporting Failures

If no quota is received from the SAPC, verify that:

Steps

1. The subscriber or subscriber group has `usageLimits` information configured.

2. The content of the Usage Limits is syntactically right. This can be checked by parsing the JSON structure with some external tool (for example: http://jsonformatter.curiousconcept.com).

3. Subscription Date < Current time < Expiry Date.
4. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to `TRUE`.

If quota = `0` is received from the SAPC, check that the applicable Reporting Groups are enabled:

5. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to `TRUE`.

   For further information, refer to Configuration Guide for Fair Usage.

5.5 Multi-Access Failures

TCP connectivity exists between peer and the SAPC.

5.5.1 Diameter Connection Problems

If there is any failure related to Diameter traffic, verify the following checks:

Steps

1. Verify Diameter Flow Policy.

   Check through an ECLI session if all Flow Policies are defined. For that purpose, execute the next command with the associated output:

   ```
   > show
   ManagedElement=1,Transport=1,Evip=1,EvipAlbs=1,EvipAlb=alb_trf
   ,EvipFlowPolicies=1

   EvipFlowPolicies=1
      EvipFlowPolicy=diameter_EBM-BSF-GX-N15-N36-N5-N7-NRF-RX-SD- →
   SMS-SOA-SX-SY-EDB_3868-ipv4
      EvipFlowPolicy=n15_rest-ipv4
      EvipFlowPolicy=nbsf_rest-ipv4
      EvipFlowPolicy=soap-ipv4
      EvipFlowPolicy=n36_rest-ipv4
      EvipFlowPolicy=n5_rest-ipv4
      EvipFlowPolicy=n7_rest-ipv4
      EvipFlowPolicy=nrf_notify_rest-ipv4
      EvipFlowPolicy=diameter_EBM-BSF-GX-N15-N36-N5-N7-NRF-RX-SD- →
   SMS-SOA-SX-SY-EDB_3869-ipv4
   (EvipFlowPolicies=1)>
   ```

2. Verify VIP address for traffic or Diameter port.

   Check through an ECLI session if the VIP-TR and DIAMETER-PORT are set. For that purpose, execute the next command with the associated output considering the VIP address for traffic handling and the 3868-port values:

```
>show all
ManagedElement=1,Transport=1,Evip=1,EvipAlbs=1,EvipAlb=alb_trf
,EvipFlowPolicies=1

EvipFlowPolicies=1
   EvipFlowPolicy=SCTP_diameter

      dest="<VIP-TR>"
      protocol="sctp"
      soGrp="1011250"
      usageState=ACTIVE

   EvipFlowPolicy=diameter_ipv4_<DIAMETER-PORT1>

      dest="<VIP-TR>"
      destPort="<DIAMETER-PORT1>"
      protocol="tcp"
      targetPool="PLs_rr"
      usageState=ACTIVE

   EvipFlowPolicy=diameter_ipv4_<DIAMETER-PORT2>

      dest="<VIP-TR>"
      destPort="<DIAMETER-PORT2>"
      protocol="tcp"
      targetPool="PLs_rr"
      usageState=ACTIVE

   EvipFlowPolicy=soap

      dest="<VIP-TR>"
      destPort="8080"
      protocol="tcp"
      targetPool="PLs_rr"
      usageState=ACTIVE
```

3. Diameter status processes.

   Check through an SSH connection as sapcadmin user to any SC processor if the C-diameter status is OK. For that purpose, execute the next command with the associated output, considering N as the number of PL processors:

```
sapcadmin@SC-X:/> amfHelper -f CDiameter -a status

Searching SUs filtering (egrep) by 'CDiameter' ...

#***>> CDiameter
[Node]     [Service Unit DN]                                   →
                                                    [AdminState]→
  [OpState]     [PresenceState] [ReadinessState] [Preinst] [Ac→
tive/Standby]

Done!
```

```
PL-N       safSu=PL-N,safSg=NWA,safApp=ERIC-sv.SVCDiameter    →
                                                   UNLOCKED(1)→
   ENABLED(1)    INSTANTIATED(3) IN-SERVICE(2)        0      A→
ctive

Done!
```

5.5.2 Diameter Failures

5.5.2.1 DIAMETER RFC 6733 Messages Failures

If there is any problem with the establishment of Diameter connections, it can be because of one of the following reasons:

Steps

1. The `Capabilities-Exchange-Request (CER)` message is rejected with the `DIAMETER_UNKNOWN_PEER` Result-Code.

   — If an `acceptFromDynamicHost` attribute is configured in the corresponding transport configuration, check the values of the following attributes in the `otpdiaDynamicHostAcceptor` object it points to:

   - The `peerOriginRealm` validation pattern matches the incoming origin realm.
   - The `peerOriginHost` validation pattern matches the incoming origin host.
   - The `peerHostIpAddress` validation pattern matches the incoming IP address.
   - The number of existing established connections is below the value configured in the `maxConnectionNr` attribute.
   - The number of existing connections per peer is below the value configured in the `maxPeerConnectionNr` attribute.

   — If an `acceptFromStaticHost` attribute is configured in the corresponding transport configuration, check the values of the following attributes in the `otpdiaHost` object it points to:

   - The `peerOriginHost` validation pattern matches the incoming origin host.
   - The number of existing established connections is below the value configured in the `maxConnectionNr` attribute.
   - The address validation pattern matches the incoming IP address.

Note: The use of the `acceptFrom` attribute is deprecated from C-DIA 3.1 onwards.

By default, the SAPC is deployed with all transport for the `MobilityService` and `Pcrf` services configured with an `acceptFromDynamicHost` attribute pointing to an `otpdiaDynamicHostAcceptor` object called `any` which accepts all incoming connections.

```
sapcadmin@SC-1:~> immlist -a acceptFromDynamicHost `immfind -c OtpdiaTransportTcp`
acceptFromDynamicHost=otpdiaDynamicHostAcceptor=any,otpdiaProduct=SAPC
sapcadmin@SC-1:~ immlist -a peerOriginRealm -a peerOriginHost -a peerHostIpAddress -a maxPeerConnectionNr -a maxConnectionNr otpdiaDynamicHostAcceptor=any,otpdiaProduct=SAPC
peerOriginRealm=.*
peerOriginHost=.*
peerHostIpAddress=.*
maxPeerConnectionNr=4294967295
maxConnectionNr=4294967295
```

2. A request is received for an unsupported application.

   — Check the Application Id and Supported Vendor Id.

   ```
   immlist -a supportedVendorId -a authApplicationId `immfind -c OtpdiaApplications`
   ```

   — Check the Vendor Specific Application Id grouped AVP:

   ```
   immlist -a vendorSpecificApplicationId `immfind -c OtpdiaApplications`
   vendorSpecificApplicationId=otpdiaVendorSpecificApplicationId=Gx
   ```

   From this, we use `vendorSpecificApplicationId=otpdiaVendorSpecificApplicationId=Gx`.

   ```
   immlist otpdiaVendorSpecificApplicationId=Gx,otpdiaProduct=SAPC -a vendorId -a otpdiaVendorSpecificApplicationId -a authApplicationId
   ```

   ```
   vendorId=10415
   otpdiaVendorSpecificApplicationId=otpdiaVendorSpecificApp
   ```

```
licationId=Gx
authApplicationId=16777238
```

3. Incorrect `Origin-Host`, `Origin-Realm`, and `Host-IP-Address` AVPs from the SAPC.

   — Check the `originHost`, `originRealm`, and `hostIpAddress` attribute values.

   ```
   immlist `immfind -c OtpdiaService`
   ```

4. It is not possible to establish two or more connections to the same peer.

   — Check that the `restrictConnections` attribute is set to `false` in the corresponding `otpdiaService` instances.

5.5.3 Gx Failures

5.5.3.1 General Failures

If any AVP related to a concrete control (Bearer Access Control, Service Access Control, QoS Control for the Default Bearer and the APN, Usage Reporting, and so on) is not obtained in `CCA` or `RAR` messages, it can be owing to one of the following reasons:

Steps

1. Check the bit for the corresponding control received in the `Gx-Capability-List` AVP within `CCR` requests.

2. Check also the corresponding control received in the `Supported-Features` AVP.

3. Check the configuration for the controls for the PCEF sending the Gx requests.

   If all mobile session establishments are suddenly rejected, it can be caused by the number of mobile sessions exceeding the capacity license. Check for License Manager alarms and verify the mobileActiveSessions measurement against the purchased capacity license.

5.5.3.2 Gx Access Control Failures

To identify the value of the `Diameter Result-Code` AVP in the answer message. To get this value, a protocol analyzer is recommended to be used (for example, Wireshark). For further information about the Result-Code meaning, refer to Gx Interface Description.

If service authorization is not successful, it can be owing to one of the following reasons:

Steps

1. Subscriber received in the `Subscription-Id` AVP is not found in the SAPC and "Unknown" special EPC-Subscriber entry is not provisioned.

   — Check if the UnknownSubscribers measure is abnormally high.

   — Check if the subscriber is correctly provisioned.

2. Service authorization result is not the expected one:

   — Check if the specific service is correctly defined.

   — Check in the provisioned Subscriber profile the list of allowed and denied services.

3. For static services, check if the service is included in the applicable Rule Space (either the one indicated by the PCEF or the one decided by the SAPC).

4. Check also whether the service is within the list of services to redirect provisioned both for the subscriber and the active groups the subscriber belongs to.

5. Check policies for the specific service. To detect if there is any error in the policy evaluation, activate the warning logging level temporarily.

6. To check if there is a protocol error, activate the warning logging level temporarily.

7. The session to be updated does not exist:

   — Check as root user if there are new logs with the following message:

   ```
   Non-Persistent data storage empty.
   ```

   In the following path:

   ```
   sapcadmin@SC-X:~> /cluster/storage/no-backup/
   coremw/var/log/syslog/sapc/
   ```

   — If the processors load allows it, activate the warning logging level temporarily to check if the session was previously removed.

5.5.3.3 Gx QoS Control for the Default Bearer and the APN Failures

If the QoS Control for the Default Bearer and the APN result are not the expected ones:

Steps

1. Verify that the *Bearer QoS Control* applies for the PCEF (check the configuration of the PCEF in the SAPC).
2. Check the values in the `QoS-Negotiation`, `Qos-Upgrade` , and `QoS-Information` AVPs.
3. Check if the `gxQosDowngraded`, `gxQosUpgraded`, and `gxQosDeactivated` measures are abnormally high. If so, check the values configured in the QoS Profiles associated with the QoS Control for the Default Bearer and the APN (either per service or per bearer) policies. Compare them to the values received in the requested QoS Profile.

   For details about provisioning and configuration, refer to Configuration Guide for Bearer QoS Control and Bandwidth Management.

5.5.3.4 Gx Usage Reporting Failures

If no quota is received from the SAPC, verify that:

Steps

1. The subscriber or subscriber group has the `usageLimits` information configured.
2. The contents of the Usage Limits are syntactically right. This can be checked by parsing the JSON structure with some external tool (for example http://jsonformatter.curiousconcept.com).
3. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to "TRUE". To detect if there is any error in the policy evaluation, activate the **info** logging level temporarily.
4. Subscription Date < Current time < Expiry Date

If no more volume quota available is received from the SAPC, check that the applicable Reporting Groups are enabled:

5. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to "TRUE". To detect if there is any error in the policy evaluation, activate the **info** logging level temporarily.

   For further information, refer to Configuration Guide for Fair Usage.

5.5.4 Rx Failures

If there is any failure related to the Rx interface, verify the following:

Steps

1. Check service classification related configuration, provisioning, and policies.

   Refer to Configuration Guide for Dynamic Policy Control (Rx).

2. Check service authorization related configuration, provisioning, and policies.

   Refer to Configuration Guide for Dynamic Policy Control (Rx).

3. Check service qualification related configuration, provisioning, and policies.

   Refer to Configuration Guide for Dynamic Policy Control (Rx).

4. Check if the counters `rxAaasFailed`, `rxRaasFailed`, `rxAsasFailed`, `rxAaasUnableToComply`, `rxAaasInvalidInfo`, `rxAaasIpSessionNotAvailable`, `RxTerminateUnknownSessions` ,`rxRarsTimeout`, and `rxAsrsTimeout` measures are abnormally high.

   If all AF session establishments are suddenly rejected, it can be caused by the number of AF sessions exceeding the capacity license. Check for License Manager alarms and verify the `afActiveSessions` measurement against the purchased capacity license.

5.5.5 N5/N30 Failures

When there is any failure related to the N5/N30 interface, check the QoS Exposure related configuration, provisioning, and policies.

Refer to Configuration Guide for Dynamic Policy Control (N5/N30).

5.5.6 Sy Failures

If there is any failure related to the Sy interface, verify the following:

Steps

1. Check the Subscriber Charging related configuration, provisioning, and policies.

   Refer to Configuration Guide for Integration with OCS for Spending Limit Reporting (Sy).

2. Check if the counters `sySlrsTimeout`,`sySlasFailed`, `sySnasFailed`, `syStrsTimeout` , and `syStasFailed` measures are abnormally high.

3. If SLR message is sent by the SAPC to the Online Charging System, but STR message is not sent later on: verify that the destination realm sent within the SLR and the origin realm received within the SLA are both properly

defined in the SAPC DIAMETER routing table. Refer to Configuration Guide for Integration with OCS for Spending Limit Reporting (Sy).

5.5.7 Smp Failures

5.5.7.1 General Failures

For any failure related to Smp interface, verify the following:

Steps

1. If all Smp session establishments are suddenly rejected by returning DIAMETER_UNABLE_TO_COMPLY (5012) in a CCA message, it can be caused by the number of Smp sessions exceeding the capacity license. Check for License Manager alarms and verify the `smpActiveGlobalSessions` measure against the purchased capacity license.
2. Check the PDN-GW and SPID related configuration, provisioning and policies.

   Refer to Configuration Guide for Mobility Based Policy Control for Overlay Deployments (Smp).
3. Check the SMP-presence Reporting Area and Event Triggers related configuration, provisioning and policies.

   Refer to Configuration Guide for Mobility Based Policy Control for Overlay Deployments (Smp).
4. Check if the counters `smpCcasInitFailed`, `smpCcasInitInvalidAvp`, `smpCcasInitMissingAvp`, `smpCcasInvalidInfo`, and `smpCcasRejected` measures are abnormally high.

5.5.7.2 IP-CAN Access Control Failures

For IP-CAN session access control failures, refer to Gx Access Control Failures on page 54.

5.5.8 HTTP/2 Connection Failure

When the failure is related to the PCF traffic, verify the following settings:

Steps

1. Go to the following path on the SC node:

   ```
   # /opt/com/bin/cliss
   ```

2. Check the number of the HTTP/2 connection, the maxstreamid, and the scheduler and verify the settings according to the definition in `PcfForwardProxy`.

   The following is an example of checking the current settings:

   ```
   > ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,PcfForwardProxy=1
   ```

   ```
   (PcfForwardProxy=1)>show
   ```

   ```
   PcfForwardProxy=1
       activeIp="PL-X"
   ```

   ```
   (PcfForwardProxy=1)>show --verbose
   ```

   ```
   PcfForwardProxy=1
      activeIp="PL-X" <read-only>
      connNum=5 <default>
      maxStreamId=-1 <default>
      pcfForwardProxyId="1"
      redundancyMode=NWA <default>
      scheduler="0" <default>
   ```

3. Check the status of client authentication of mutual TLS (mTLS) and verify the setting according to the definition in `PcfSecurity`:

   The following is an example of checking the current settings:

   ```
   > ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,PcfNetwork=1,PcfSecurity=1
   ```

   ```
   (PcfSecurity=1)>show --verbose
   ```

   ```
   PcfSecurity=1
      enableTLS=false <default>
      enableTLSClientAuthenticationAF=false <default>
      enableTLSClientAuthenticationNRF=false <default>
      enableTLSClientAuthenticationRelay=false <default>
      enableTLSClientAuthenticationSMF=false <default>
      enableTLSClientAuthenticationAMF=false <default>
      enableTLSClientAuthenticationUDR=false <default>
      pcfSecurityId="1"
      requireSubjectAltNameinServerCert=false <default>
   ```

5.5.9 N7 Failure

When the failure is related to the N7 reference point between the SMF and the SAPC PCF, verify the following settings:

Steps

1. Check the access and charging related configuration, provisioning, and policies. Refer to Configuration Guide for Access and Charging Control (N7).
2. Check the QoS control related configuration, provisioning, and policies. Refer to Configuration Guide for QoS Control and Bandwidth Management (N7).
3. Check the usage monitoring related configuration, provisioning, and policies. Refer to Configuration Guide for Usage Monitoring Control.
4. Check the N7 related measures. Refer to Measurements.

5.5.10 N15 Failure

For any failure related to N15 interface, verify the following:

Steps

1. Check the SAR related configuration, provisioning and policies.

   Refer to Configuration Guide for Access and Mobility Policy Control (N15).

2. Check the RFSP related configuration, provisioning and policies.

   Refer to Configuration Guide for Access and Mobility Policy Control (N15).

3. Check the PRA and Policy Control Request Trigger related configuration, provisioning and policies.

   Refer to Configuration Guide for Access and Mobility Policy Control (N15).

4. Check the N15 related measures. Refer to Measurements.

5.5.11 Nnrf Failure

When the failure is related to Nnrf interface, check the following settings:

Steps

1. Go to the following path on the SC node:

   ```
   # /opt/com/bin/cliss
   ```

2. Check the PCF peer nodes configuration and verify the current setting according to the `PcfPeerNodes`.

   The following is an example of checking the current settings:

   ```
   >ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,PcfNetwo
   rk=1,PcfPeerNodes=1
   ```

   ```
   (PcfPeerNodes=1)>show all
   ```

   ```
   PcfPeerNodes=1
      PcfPeerNode=nrf1
         ipv4="10.200.71.101"
         port=8080
         type=NRF
   ```

   If find any error, refer to Configuration Guide for Interaction with NRF.

5.5.12 N36 Failure

5.5.12.1 Failure with Locally Configured UDR Information

1. Confirm that `UDR` is not included in the list of `enableNrfDiscoveryNfs` under `PcfAppConfig`.

2. Check the locally configured UDR information.

   a. Go to the following path on the SC node:

      ```
      # /opt/com/bin/cliss
      ```

   b. Check the PCF peer nodes configuration and verify the current setting according to the `PcfPeerNodes`.

      The following is an example of checking the current settings:

      ```
      >ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,P
      cfNetwork=1,PcfPeerNodes=1
      ```

      ```
      (PcfPeerNodes=1)>show all
      ```

      ```
      PcfPeerNodes=1
         PcfPeerNode=udr1
            ipv4="10.200.71.101"
            ipv6="FE80:1234::0000"
            nodeStatus=REGISTERED
            port=8086
            type=UDR
      ```

   For more information, refer to Configuration Guide for Interaction with UDR.

5.5.12.2 Failure with UDR Discovered from the NRF

1. Confirm that `UDR` is included in the list of `enableNrfDiscoveryNfs` under `PcfAppConfig`.
2. Check discovered UDR result in **DiscoveredPeerNodes**.
3. Check NRF configuration in **PcfPeerNodes**.

   For more information, refer to Configuration Guide for Interaction with NRF.

5.5.13 Nbsf Failure

5.5.13.1 No Registration Request Sent out

> If the SAPC PCF does not send out Nbsf_Management_Register request to the BSF, it may be caused by not setting the DNN information. Check the following:

1. Go to the following path on the SC node:

   `#` **`/opt/com/bin/cliss`**

2. Check the DNN configuration in `PcfBsfRegAllowedList`:

   `>`**`ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,PcfAppConfig=1`**

   `(PcfAppConfig=1)>`**`show all`**

```
PcfAppConfig=1
   PcfBsfRegAllowedLists=1
      PcfBsfRegAllowedList=1
         dnn="dnn_ims.com"
   PcfMsgTemplates=1
(PcfAppConfig=1)>
```

5.5.13.2 Failure with Locally Configured BSF Information

> If BSFs are locally configured, check the following:

1. Confirm that `BSF` is not included in the list of `enableNrfDiscoveryNfs` under `PcfAppConfig`.
2. Check the locally configured BSF information.

a. Go to the following path on the SC node:

```
# /opt/com/bin/cliss
```

b. Check the PCF peer nodes configuration and verify the current setting according to the `PcfPeerNodes`.

The following is an example of checking the current settings:

```
>ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,P
cfNetwork=1,PcfPeerNodes=1
```

```
(PcfPeerNodes=1)>show all
```

```
PcfPeerNodes=1
   PcfPeerNode=bsf1
      ipv4="10.200.67.106"
      nodeStatus=REGISTERED
      port=8010
      type=BSF
```

For more information, refer to Configuration Guide for Interaction with BSF.

5.5.13.3 Failure with BSF Discovered from the NRF

> If BSFs are discovered from the NRF, check the following:

1. Confirm that `BSF` is included in the list of `enableNrfDiscoveryNfs` under `PcfAppConfig`.
2. Check discovered BSF result in **DiscoveredPeerNodes**.
3. Check NRF configuration in **PcfPeerNodes**.

   For more information, refer to Configuration Guide for Interaction with NRF.

5.5.14 SCP Failure

5.5.14.1 Failure with Locally Configured SCP Information

> If SCPs are locally configured, check the following:

1. Confirm that `SCP` is not included in the list of `enableNrfDiscoveryNfs` under `PcfAppConfig`.
2. Check the locally configured SCP information.

   a. Go to the following path on the SC node:

```
#/opt/com/bin/cliss
```

b. Check the PCF peer nodes configuration and verify the current setting according to the `PcfPeerNodes`.

   The following is an example of checking the current settings:

```
>ManagedElement=1,PolicyControlFunction=1,PcfConfig=1,PcfNetwork=1,PcfPeerNodes=1
```

```
(PcfPeerNodes=1)>show all
```

```
PcfPeerNodes=1
   PcfPeerNode=scp1
      ipv4="192.168.14.42"
      nodeStatus=REGISTERED
      port=8040
      type=SCP
```

For more information, refer to Configuration Guide for Indirect Communication through SCP.

5.6 End User Notifications Failures

If the SMS/SOAP Notifications fail to be sent, verify the following:

Steps

1. Verify that the `enableDelivery` attribute of the **NotificationConfig** COM object is set to "TRUE".
2. Verify that the End User Notifications are configured properly:
   - For SMS notifications:

     Check if the **SMSCenter** and **SMSDestination** COM object values are correctly configured under the **Network** COM object.
   - For SOAP notifications:

     Check if the **WebServiceEndPoints**, **WebServiceEndPoint**, and **WsDestination** COM object values are correctly configured under the **Network** COM object.
3. Check if the notification policies are correctly configured.
4. Check if the "ConnectionNotificationServerFailed" alarm is raised.
5. Check logs for end-user notifications.

Further information on configuring end-user notifications can be found in Configuration Guide for End User Notifications.

5.7 External Database Failures

If there is any failure related to access to external database, verify the following:

Steps

1. Check through an ECLI session if the VIP address for access to external database <VIP-ExtDB> is defined on the Abstract Load Balancer (alb_edb).

```
> show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
EvipAlb=alb_edb, EvipVips=1
```

```
EvipVip= <VIP-ExtDB>

EvipVip= <VIP-TR>
```

2. Check through an ECLI session if the Local IP address for access to external database is defined on the Entity Data object.

```
> show ManagedElement=1, PolicyControlFunction=1, EntityData=1
```

```
localIp=<VIP-ExtDB>
```

3. Check through an ECLI session the IPs defined for external database.

```
> show ManagedElement=1, PolicyControlFunction=1,EntityData=1,
EDSources=1, EDSource=ExternalRepository
```

```
EDSource=ExternalRepository

definition="def ExternalRepository () { dataSource =
{ url = \"\"; query = \"\"; } fieldDef = { ips
= \"136.225.72.9;136.225.72.17;136.225.72.25\"; port =
\"389\"; } }"
```

4. Check if there is connection to any of the External databases from your PL.

```
sapcadmin@PL-X:~> ping -I <VIP-ExtDB> <External Database IP>
```

5. Check if the external database outgoing connections are correctly established within the active IP. 64 connections per PL are expected.

```
sapcadmin@SC-X:~> forall sapc.payload "hostname; lsof
-i@<External Database IP>:389 | grep -i ESTABLISHED | wc -l"
```

```
PL-3
64
PL-4
64
```

Further information on configuring access to External Database can be found in Database Access.

5.8 EBM Failures

If there is any failure related to the Event-Based Monitoring (EBM) interface, verify the following:

Steps

1. Check if the `enable` attribute of the EventBasedMonitoring COM object is set to `true`.

   ```
   >show ManagedElement=1,PolicyControlFunction=1,EventBasedMonitoring=1,EbmBusinessEvents=1,EbmBusinessEvent=QUOTA_GRANTED
   EbmBusinessEvent=QUOTA_GRANTED
      ebmServerIds
         ""
      enable=true
   >
   ```

2. Check if at least one EBM Server is configured correctly.

3. Check if individual EBM events are enabled.

4. Check if the EbmCommunicationFailure or EbmBufferOverflow alarms are raised.

5. Check if the ebmBusinessEventsNotSent measure is abnormally high.

5.9 SOAP Notification Interface Failures

If there is any failure related to SOAP notification interface, verify the following:

Steps

1. Check through an ECLI session the Flow Policy for SOAP incoming notification service. Misconfigured SOAP incoming notification service flow policy in eVIP prevents the correct binding of the SOAP server process to the listening port.

   ```
   > show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
   EvipAlb=alb_trf, EvipFlowPolicies=1
   ```

   ```
   EvipFlowPolicy=soap
   ```

   This is the print definition of flow policy:

```
> show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
EvipAlb=alb_trf, EvipFlowPolicies=1, EvipFlowPolicy=soap

EvipFlowPolicy=soap

   dest="<VIP-ExtDB>"
   destPort="8080"
   protocol="tcp"
   targetPool="PLs_rr"
   usageState=ACTIVE
```

2. Check if the SOAP incoming notification service port 8080 is listening on Abstract Load Balancer where the VIP for access to external database is configured (alb_tr).

```
sapcadmin@SC-X:~> forall sapc.payload "hostname ; netstat -nap
| grep :8080 | grep LISTEN"

PL-10
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      11954/soap-notifica
PL-11
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      12538/soap-notifica
PL-12
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      12514/soap-notifica
PL-5
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      10432/soap-notifica
PL-6
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      12556/soap-notifica
PL-7
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      11887/soap-notifica
PL-8
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      12822/soap-notifica
PL-9
tcp        0      0 10.41.30.53:8080        0.0.0.0:           →
*               LISTEN      11194/soap-notifica
```

Further information on configuring SOAP incoming notification web service can be found in SOAP Notification Interface Description.

5.10 OAM Interface Unavailable

The **<OAM VIP>** interface can be unavailable due to failures in both **SCs**. In this scenario, **OAM** features are restricted, but the traffic can be still processed for 15 more minutes before the whole cluster reboots due to the SCs absence feature.

Steps

1. Recover at least one of the two **SCs** to prevent the cluster from rebooting. Once it is recovered, the **<OAM VIP>** is available.

2. If no **SC** is recovered in 15 minutes, the cluster goes down. In this scenario, recover at least one of the **SCs** and the system restarts normally. If the **SC** does not recover or the system does not restart normally, contact the next level of maintenance support.

5.11 Backup Failures

5.11.1 Failure Due to Lack of Disk Space

If backup fails due to lack of disk space, perform the following actions:

Steps

1. Free disk space by deleting unused files or older backups.

2. Unlock the Backup and Restore component in order to create a new backup by using the following command:

   ```
   > cmw-maintenance-unlock BRF
   ```

3. Try creating the backup again.

   For further information, refer to Backup and Restore.

5.11.2 Failure Due to Provisioning a Large Amount of Subscriber Data

The backup is cancelled with the following log information:

```
>ManagedElement=1,SystemFunctions=1,BrM=1
(BrM=1)>BrmBackupManager=SYSTEM_DATA
(BrmBackupManager=SYSTEM_DATA)>show progressReport, state
state=RUNNING
(BrmBackupManager=SYSTEM_DATA)>show BrmBackup=20210422_031256_Au →
tomTCUpgraded, status
status=BRM_BACKUP_INCOMPLETE
(BrmBackupManager=SYSTEM_DATA)>show progressReport, state
state=CANCELLING
(BrmBackupManager=SYSTEM_DATA)>show progressReport, state
state=CANCELLED
(BrmBackupManager=SYSTEM_DATA)>show BrmBackup=20210422_031256_Au →
tomTCUpgraded, status
ERROR: Failed to get attribute value(s) for status
```

```
(BrmBackupManager=SYSTEM_DATA)>[E] CLI timeout while checking Backup result.
```

This is caused by provisioning a large amount of subscriber data in the provisioning REST API and immediately performing the backup.

Perform the following action:

Try creating the backup again.

5.12 Dynamic Subscriber OSI Update in N+1 Geographical Redundancy Failures

In an N+1 Geographical Redundancy deployment, the Subscriber Operator Specific Information (OSI) changes generated because of the Dynamic Subscriber Profile Update policies evaluation in the local SAPC are replicated automatically in the rest of the SAPC peers of the N+1 cluster. Representational State Transfer (REST) is used as an interface for the Dynamic Subscriber OSI update replication purpose. If replication fails, the cluster has an inconsistent status.

When a replication operation fails, the local node, which is trying to replicate the Subscriber OSI resource, logs the failing resource to a subscriber error file in the following directory:

```
/storage/no-backup/sapc/georedFiles/provisioningErrorFiles
```

The subscriber error file is named according to following schema:

```
<GEO_ID>_<PL>_subserr
```

An example of subscriber error file content is the following:

```
PL-3:/storage/no-backup/sapc/georedFiles/provisioningErrorFiles # cat SAPC2_PL-3_subserr

/provisioning/v1/subscribers/222222000000/operator-specific-infos

/provisioning/v1/subscribers/222222000001/operator-specific-infos
```

In order to synchronize the Subscriber OSI data in the SAPC peers, the REST resources stored in those files must be updated using the `sync-provisioning` script. Specific alarm `Policy Control, Geographical Redundancy Subscriber Operator-Specific-Info Update Replication Failure` is cleared if synchronization success.

For more information about using the `sync-provisioning` script, see sync-provisioning on page 30.

Some examples of handling with Dynamic Subscriber OSI update replication errors are listed below. For demonstrative purposes, only Subscriber OSI synchronization is shown (`-t subserr` option used).

Example 25 Successful Sync-Provisioning Execution

```
sapcadmin@PL-3:~> sync-provisioning -g SAPC2 -u sapcprov:<password> -t subserr
-------------------------------------------------
Processing subserr files ...
-------------------------------------------------

Processing 1 files, total 2 resources

Total 2 unique resources

Start processing

Processed 2 resources

Synchronization successful!

No active provisioning errors: Clearing Subscriber Operator-Specific-Info update replication Failure Alarm...
```

In this case, the `Policy Control, Geographical Redundancy Subscriber Operator-Specific-Info Update Replication Failure` is cleared.

Example 26 Unsuccessful Sync-Provisioning Execution due to a Failed Authentication

Example of an unsuccessful sync-provisioning execution, in this case, due to a failed authentication:

```
sapcadmin@PL-3:~> sync-provisioning -g SAPC2 -u user_unknown:<password> -t subserr
-------------------------------------------------
Processing subserr files ...
-------------------------------------------------

Processing 1 files, total 2 resources

Total 2 unique resources

Start processing

Error executing internal replicate_resources.py script. Use -v for more information
```

```
Synchronization failed! 2 errors found
```

Example 27 Sync-Provisioning Execution in Verbose Mode

```
sapcadmin@PL-3:~> sync-provisioning -g SAPC2 -u sapcprov:<password> -t subserr
-------------------------------------------------
Processing subserr files ...
-------------------------------------------------

Processing 1 files, total 2 resources

Total 2 unique resources

Start processing

('GET', '/provisioning/v1/subscribers/222222000000/operator-specific-infos', 200, 0.163705)

('PUT', '/provisioning/v1/subscribers/222222000000/operator-specific-infos', 204, 0.065372)

('GET', '/provisioning/v1/subscribers/222222000001/operator-specific-infos', 200, 0.162342)

('PUT', '/provisioning/v1/subscribers/222222000001/operator-specific-infos', 201, 0.058036)

Processed 2 resources

Synchronization successful!

No active errors: Clearing Subscriber Operator-Specific-Info update replication Failure Alarm...


sapcadmin@PL-3:~> cat /storage/no-backup/sapc/georedFiles/provisioningErrorFiles/sync-provisioning.log

Tue Feb 23 17:33:06 CET 2021
-------------------------------------------------
Processing subserr files ...
-------------------------------------------------

Processing 1 files, total 2 resources

Total 2 unique resources

Start processing
```

```
('GET', '/provisioning/v1/subscribers/222222000000/operator-spec →
ific-infos', 200, 0.163705)

('PUT', '/provisioning/v1/subscribers/222222000000/operator-spec →
ific-infos', 204, 0.065372)

('GET', '/provisioning/v1/subscribers/222222000001/operator-spec →
ific-infos', 200, 0.162342)

('PUT', '/provisioning/v1/subscribers/222222000001/operator-spec →
ific-infos', 201, 0.058036)

Processed 2 resources

Synchronization successful!

No active errors: Clearing Subscriber Operator-Specific-Info upd →
ate replication Failure Alarm...
```

5.13 BSP portStateTracking

Procedure to enable `portStateTracking` in SAPC. Perform the following steps before an administrative task on the DMX with more than one subrack with BSP configuration:

Steps

1. Check the health of the SAPC.

   Execute the `sapcHealthCheck` command as explained in the SAPC Troubleshooting Guide for getting the SAPC state.

2. Log in to DMX.

   ```
   <>:# ssh -p 2024 advanced@<DMX>
   <>: password:
   ```

3. Show the status of the portSateTraking:

   ```
   <DMX>:>show ManagedElement=1,DmxcFunction=1,Tenant=SAPC, port →
   StateTracking
   <DMX>:>portStateTracking=DISABLED
   ```

4. Enable the portStateTracking:

   ```
   <DMX>:>configure(config)

   <DMX>:>ManagedElement=1,DmxcFunction=1,Tenant=SAPC, portState →
   ```

```
Tracking=ENABLED
<DMX>:>(config)>commit
<DMX>:>show ManagedElement=1,DmxcFunction=1,Tenant=SAPC, port →
StateTracking
<DMX>:>portStateTracking=ENABLED
```

6 Scaling Failures

If the scale in or scale out fails, it can be due to one of the following reasons:

— The Backup and Restore Framework (BRF) component is locked. Unlock it, using the following command, to make it possible to start scaling.

```
> cmw-maintenance-unlock BRF
```

— At least one of the PLs is locked and not operative. Unlock all the PLs. For detailed information on how to do it, see Processor Lock and Unlock on page 38.

— If scaling is not possible even after the above procedures, escalate this issue to the next support level.

7 GeoRed Environments

7.1 Failure When Start Geographical Redundancy After Installation

After a deployment for Geographical Redundancy in a system with high number of PLs, it is possible that the SAPC node does not start working in `Active` state after executing the `start` command in Start Geographical Redundancy procedure. This only could happen to initial deployment, not upgrade and reboots.

The problem is identified if the following error message arises during the execution of the `start` command in COM CLI:

```
#(GeoRedManager=1)>start
ERROR: Call command failed, error code: ComNotExist
```

The workaround or preventive action to fix the problem is the restarting of the Geored Control process by the execution of the following command:

```
SC-1:~ # sapcGeoredControl -a restart -f
```

Execute the `start` command in COM CLI again.

7.2 Failure Getting Distributed State Geographical Redundancy

After a deployment for Geographical Redundancy in a system with high number of PLs, it is possible that the SAPC node does not start working in `Distributed` state after executing `start` command in Start Geographical Redundancy procedure. This only could happen to initial deployment, not upgrade and reboots.

The problem can be identified if `Active` state is shown when executing the `show` command in COM CLI in the Preferred Node:

```
#(GeoRedManager=1)>show
GeoRedManager=1
currentState=ACTIVE
```

The `Synchronizing` state is shown when executing the `show` command in COM CLI in the NonPreferred Node:

```
#(GeoRedManager=1)>show
```

```
GeoRedManager=1
currentState=SYNCHRONIZING
```

Execute the following steps to get `Distributed` state in case the `Distributed` state cannot be reached in both nodes after a while (5 minutes):

Steps

1. In the SAPC configured as Non Preferred, follow the procedure in Temporarily Disable Active-Active Geographical Redundancy. This procedure stops replication and changes the state to Halted.

   ```
   #(GeoRedManager=1)>stop
   ```

2. In the SAPC configured as Preferred, execute the following command to disable replication:

   ```
   immcfg -a isEnabled=0 dbsNetsharedConfigId=1
   ```

3. In the Preferred node, execute the following commands to enable replication:

   ```
   immcfg -a isEnabled=1 dbsNetsharedConfigId=1
   ```

4. In the Halted node, where replication was stopped, follow the procedure in Enable Active-Active Geographical Redundancy. Execute the `show` command to check if the status is `Distributed`:

   ```
   #(GeoRedManager=1)>start
   #(GeoRedManager=1)>show
   GeoRedManager=1
   currentState=DISTRIBUTED
   ```

7.3 Force DBS Full Synchronization

To force a manual DBS full synchronization, perform the following step:

Steps

1. Execute the following command in an SC of the non-preferred node:

   ```
   SC-1:~ # cdsv-cluster-reload CONFIRM
   ```

   Result:

   ```
   Result from [.cdsv.director]:
   Reloading cluster.
   ```

7.4 Failure Completing DBS Synchronization

If specific alarm `DBS, NR, Initial Synchronization Needed` and `DBS, NR, Synchronization Needed` are not cleared for a long time, check if the connection is broken by executing the following commands:

```
SC-2:~ # forall PLs 'echo;hostname; netstat -antp | grep 5666 |   →
grep Dbn'

PL-3
tcp        0      0 10.35.178.222:5666      0.0.0.0:              →
*               LISTEN      26628/Dbn
tcp        0      0 10.35.178.222:5666      10.35.178.223:3277    →
1     ESTABLISHED 26628/Dbn
tcp        0      0 10.35.178.222:32799     10.35.178.223:566     →
6      ESTABLISHED 26628/Dbn

PL-4
tcp        0      0 10.35.178.222:5666      0.0.0.0:              →
*               LISTEN      23125/Dbn
tcp        0      0 10.35.178.222:32862     10.35.178.223:566     →
6      ESTABLISHED 23125/Dbn
tcp        0      0 10.35.178.222:5666      10.35.178.223:3287    →
2     ESTABLISHED 23125/Dbn
SC-2:~ #
```

If alarm `DBS, NR, Connection Lost` is raised along with `DBS, NR, Synchronization Needed`, check the following steps:

1. check if all DbsvAgent is Active on both clusters.

```
SC-x:~ # amfHelper -f SVDbsvAgent -a status
Searching SUs filtering (egrep) by 'SVDbsvAgent' ...
#***>> SVDbsvAgent
[Node]     [Service Unit DN]                                     →
                                                    [AdminStat →
e]  [OpState]     [PresenceState] [ReadinessState] [Preinst]   →
[Active/Standby]
Done!
PL-4       safSu=PL-4,safSg=NWA,safApp=ERIC-sv.SVDbsvAgent       →
                                                     UNLOCKED( →
1)   ENABLED(1)    INSTANTIATED(3) IN-SERVICE(2)       1       →
  Active
PL-3       safSu=PL-3,safSg=NWA,safApp=ERIC-sv.SVDbsvAgent       →
                                                     UNLOCKED( →
1)   ENABLED(1)    INSTANTIATED(3) IN-SERVICE(2)       1       →
  Active
SC-x:~ #
```

2. check if all 5666 port is in LISTEN for both peers.

```
SC-x:~ # forall PLs 'echo;hostname; netstat -antp | grep 566 →
6 | grep Dbn | grep LISTEN'
PL-3
tcp        0      0 10.35.178.222:5666      0.0.0.0:          →
*               LISTEN      26628/Dbn
PL-4
tcp        0      0 10.35.178.222:5666      0.0.0.0:          →
*               LISTEN      23125/Dbn
SC-x:~ #
```

3. check if any PL with unexpected tcp:5666 status other than ESTABLISHED/ LISTEN for both peers.

```
SC-x:~ # forall PLs 'echo;hostname; netstat -antp | grep :566 →
6 | grep Dbn | grep -v LISTEN | grep -v ESTABLISHED '

PL-3

PL-4
SC-x:~ #
```

Troubleshooting

— If and lead to unexpected printout and DBSv cluster state is in Idle, try manually rebooting the PL to exclude HW issue or external connectivity issues:

```
DBSv cluster state checking:
SC-2:~ # cdsv-get-user-state | grep "cluster state"
DBSv - cluster state: Idle
SC-2:~ #

PL reboot
SC-2:~ # cmw-node-reboot PL-X
```

If issues still persist after reboot, lock the PL to check if it is HW issue or infrastructure connectivity issue

```
lock PL
SC-2:~ # cmw-node-lock PL-X
```

— If leads to unexpected printout, execute the following commands to check connectivity between two peers and external network:

```
PL-3:~ # ping <Peer Replication IP> -I <Local Replication IP>
PL-3:~ # netcat -nnvz -w 1 <Peer Replication IP> 5666 -s <Loc →
al Replication IP>
```

7.5 Failure Caused by Incoming Diameter Message Indicating Another SAPC As Destination Host

In the Active-Active or N+1 geographic redundancy deployment, possibility exists that the subsequent diameter request messages for an ongoing session are sent to a SAPC which is not the one previously establishing the session. Normally, this can be handled by setting the destination host and realm for the peer node and DRA to point to the target SAPC explicitly. However, in the unlikely case, the destination host and realm fields are still set as the SAPC that established the session, therefore when another SAPC receives the diameter request message, this SAPC will reject it and reply with Result-Code `DIAMETER_UNABLE_TO_DELIVER (3002)`, if `DIACC_DIASERVER_MESSAGE_DESTINATION_VALIDATION_ENABLED` is set as `true`.

To handle the messages, disable the check services for destination host and realm by performing the following procedures:

Steps

1. In an SC, execute the following:

```
SC-x:~ # cmw-utility immcfg -c CdiaConfigAttribute -a cdiaCfgAttrV →
al=0 cdiaCfgAttr=DIACC_DIASERVER_MESSAGE_DESTINATION_VALIDATION_ENA →
BLED,cdiaSite=cdiaSite
```

2. Restart C-Diameter stack for all PLs one by one from an SC:

Note: The restart can cause a limited traffic impact.

```
SC-x:~ # for su in $(immfind -c SaAmfSU safSg=NWA,safApp=ERIC-sv.S →
VCDiameter); do cmw-utility amfadm lock $su; cmw-utility amfadm unl →
ock $su; done
```