CPS vDRA Configuration Guide, Release 26.1.0

First Published: 2026-04-23

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Preface

About This Guide

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This document overrides the same document available in the 22.1.0. For other functionality refer to the 22.1.0 documentation at Cisco.com.

This document is a part of the Cisco Policy Suite documentation set.

For information about available documentation, see the *CPS Documentation Map* for this release at Cisco.com.

Audience

This guide is best used by these readers:

- Network administrators
- Network engineers
- Network operators
- System administrators

This document assumes a general understanding of network architecture, configuration, and operations.

Additional Support

For further documentation and support:

- Contact your Cisco Systems, Inc. technical representative.
- Call the Cisco Systems, Inc. technical support number.
- Write to Cisco Systems, Inc. at support@cisco.com.
- Refer to support matrix at https://www.cisco.com/c/en/us/support/index.html and to other documents related to *Cisco Policy Suite*.

Conventions (all documentation)

This document uses the following conventions.

Conventions	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means reader take note. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS.

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Note Regulatory: Provided for additional information and to comply with regulatory and customer requirements.

Communications, Services, and Additional Information

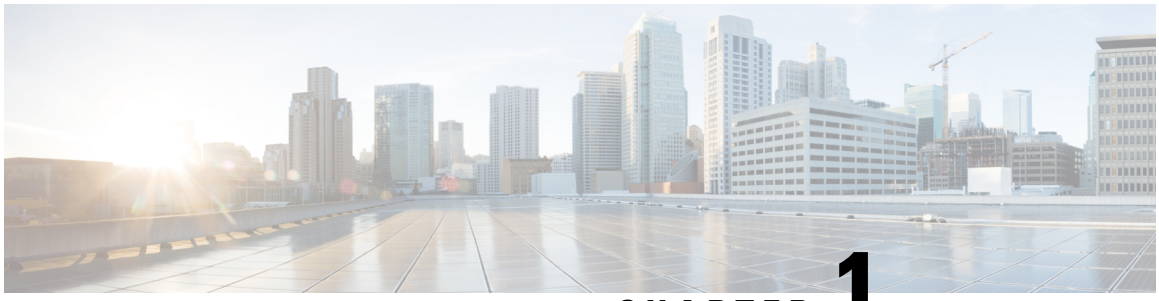
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Important Notes

Important Any feature or GUI functionality that is not documented may not be supported in this release or may be customer specific, and must not be used without consulting your Cisco Account representative.



CHAPTER 1

Introduction

CPS vDRA Overview

CPS Diameter Routing Agent (vDRA) is the functional element in a network that routes messages to the destination node based on routing algorithms.

CPS vDRA is primarily responsible for routing messages and sending responses back to the origin node.

CPS vDRA is compliant with IETF RFC 3588 and 3GPP 29.212 and 29.213 message AVPs.

Functions of DRA

DRA performs the following functions in the network:

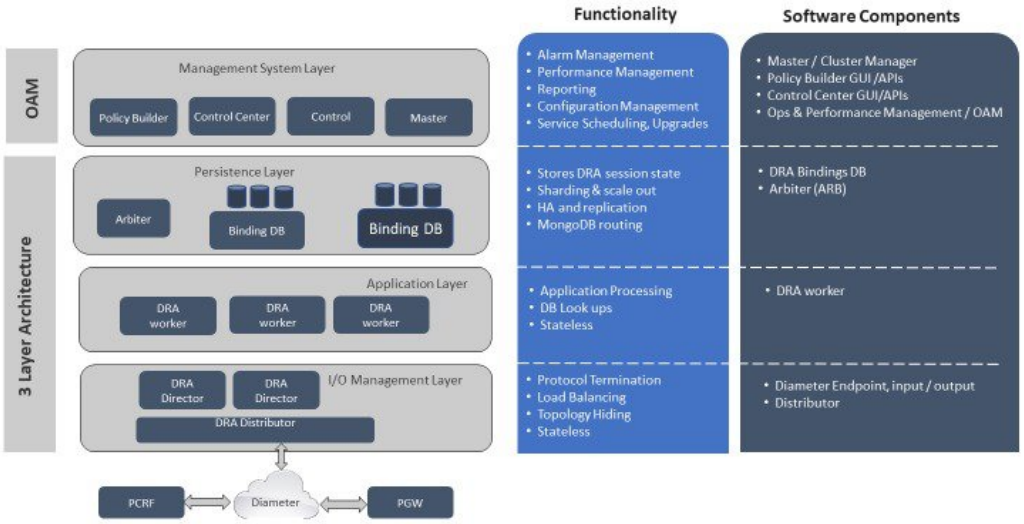
- Peer Aggregation:
  - Provides an aggregation point to eliminate full mesh of endpoint peer connections.
  - Addition of endpoint does not require reconfiguration of endpoints. Requires only the configuration of a new endpoint on DRA.
- Intelligent Routing:
  - Provides intelligent load balancing behavior for endpoints (PGW, AF).
  - Endpoints typically only route to primary/secondary peer connections.
  - Route requests to servers (PCRF, OCS) based on content of Diameter AVPs (Called-Station-ID, IPv4 Address, IPv6 Address and IMSI-APN combined).
  - Weighted routing of requests to diameter servers (PCRF, OCS).
- Binding:

- Route requests for related diameter sessions to the same Diameter server (PCRF, OCS). For example, DRA binds Gx and Rx to the same IP session using the framed IPs.

- Relay:
  - DRA provides mechanism to relay request to another DRA. In certain cases, when route for the request is found on a remote DRA, the request is relayed to the remote DRA.

CPS vDRA Architecture

The following figure illustrates the components of CPS vDRA architecture.



DRA Director is stateless node. DRA Director has diameter stack running on it, which connects to the external network functions (for example, PCEF, PCRF, AF). DRA Director receives request messages from origin peer, applies routing algorithm, forwards messages to the destination peer. DRA Director then gets answer messages for the requests, which are forwarded back to the origin peer.

DRA Processor is also stateless node that interacts with session and binding databases in the persistence tier to store session and bindings.

DRA Database is used to store bindings and sessions, with MongoDB database running on them. DRA Database uses application based client sharding to distribute data among multiple databases. MongoDB replicates data across multiple databases within the replica set to provide high availability.

Each of these tiers can be scaled horizontally by deploying more virtual machines.

Types of CPS vDRA

CPS vDRA can be deployed as IMS or Policy or a combination of both.

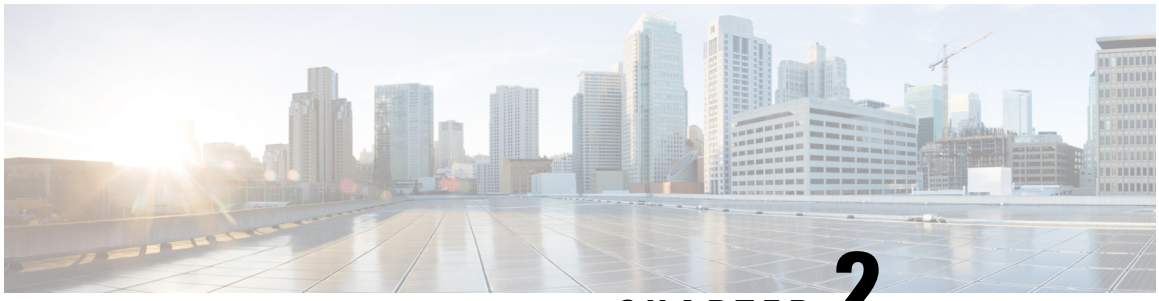
- Policy DRA is a functional element that supports Gx, Rx, Gy, Sy, and Sd diameter interfaces.

Policy DRA has a binding function that ensures diameter messages for Gx and Rx sessions for the same IP-CAN session are routed to the same PCRF when multiple and separately addressable PCRFs have been deployed.

- IMS DRA is a functional element that supports many diameter interfaces including S6a/S6d, S6b, Sh, Cx, SLh, SLg, SWm, SWx, SWa, and STa.

  IMS DRA supports SLF-based routing to ensure Diameter messages are routed to an HSS and AAA server that can provide service for a UE based on a subscriber key (that is, IMSI or MSISDN).

- Combination DRA is a functional element that supports both Policy DRA and IMS DRA functionality.



CHAPTER 2

User Configuration

Basic Configuration

Before you begin using CPS vDRA, perform the following basic configurations in CPS DRA:

- Configure Systems
- Configure Diameter Application
- Configure Routing AVP Definitions

Configure Systems

In CPS DRA, navigate to the **System and Plugin Configuration**.

Configure the stack in **DRA Configuration** plugin.

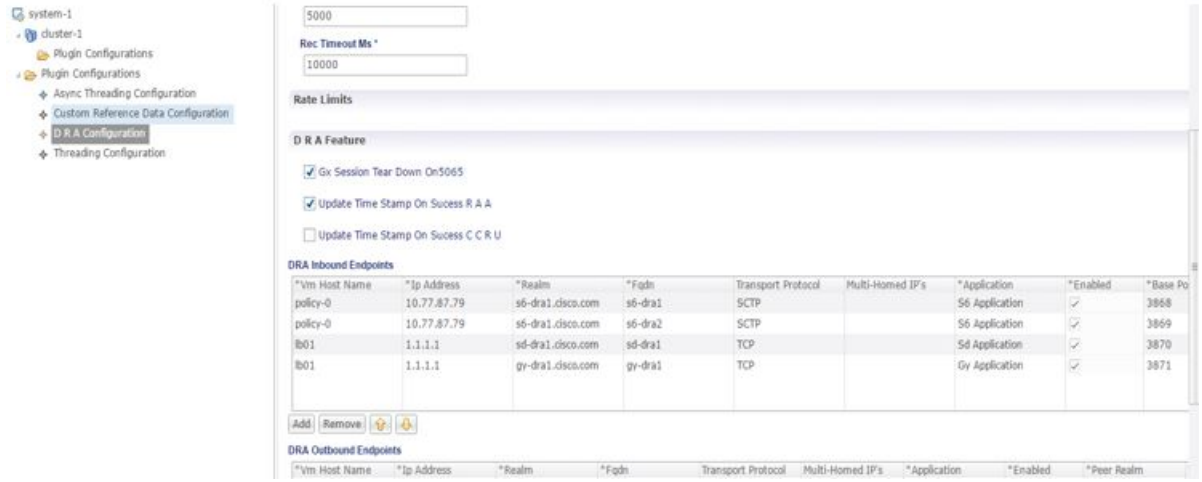
Configure the **DRA Inbound Endpoints** for incoming peer connections and **DRA Outbound Endpoints** for outgoing peer connection.

You can choose the Transport Protocol as TCP and SCTP depending on your requirement.

You can also specify the IPv4 or IPv6 address configuration for the stack connection.

The following image shows a sample configuration.

Figure 1: Sample Systems Configuration



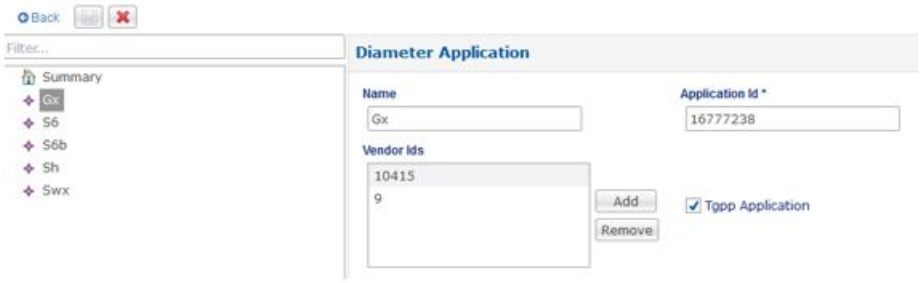
For more information, see DRA Configuration.

Configure Diameter Application

Configure the Diameter applications that are required to be connected over various interfaces with CPS vDRA.

The following image is a sample of a Gx application configuration:

Figure 2: Sample Diameter Application Configuration



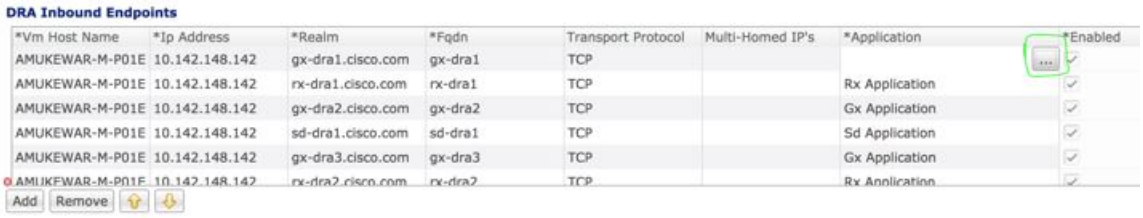
For more information, see Diameter Application, on page 41.

Configure Multiple Diameter Applications for a Peer Connection

Previously, vDRA supported a single application on a peer connection. In this release, vDRA supports multiple applications on a peer connection.

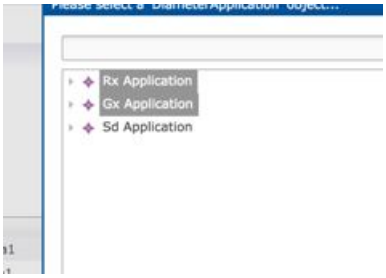
To configure multiple applications for a peer connection, go to vDRA Inbound Endpoints in DRA Plugin configuration. In the Applications field, select the button as shown:

Figure 3: DRA Inbound Endpoints



Select all the applications you require.

Figure 4: Application Selection



The following example shows multiple Diameter applications for a peer connection:

Figure 5: Multiple Applications

Configure Routing AVP Definitions

Configure the Routing AVP definitions to route calls on the basis of the AVPs that are present in diameter message.

In the **Routing AVP Definition** page, you specify the Application name and the table for table-driven routing.

In the **Diameter Application** page, configure the Application Route for table-driven routing.

The following screenshots show a sample configuration:

Figure 6: Routing Avp Definition

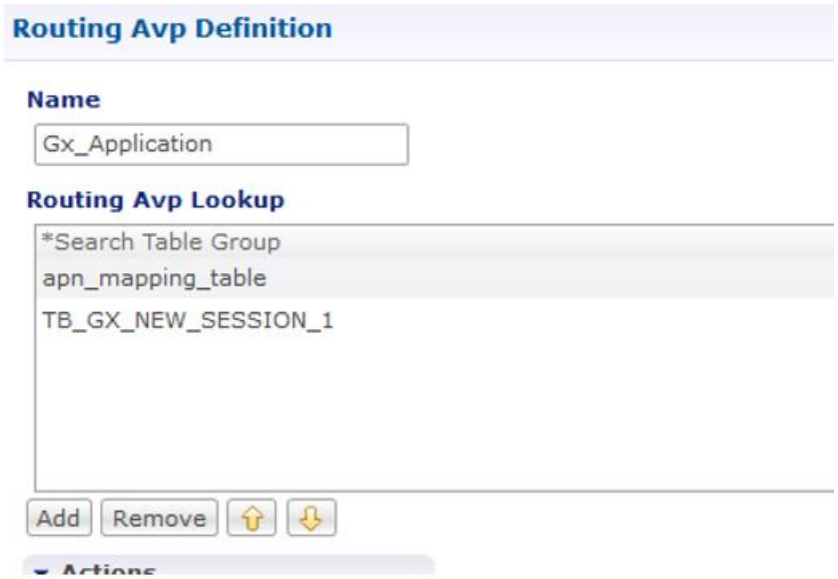
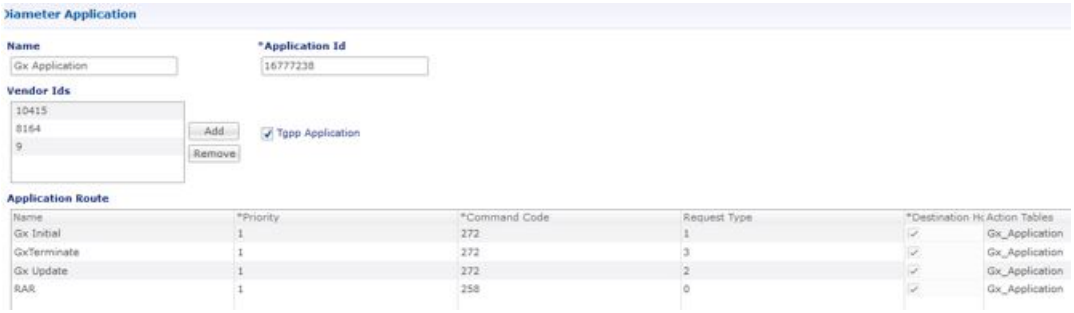


Figure 7: Diameter Application



Routing Techniques

You can define the routing of calls based on destination host, SRK, or a table.

Configure Destination Host Routing

Destination host based routing is the basic and default routing technique used by CPS vDRA.

When the incoming diameter request contains the destination-host AVP that has the direct connection with the CPS vDRA, vDRA routes the message directly to that connected host.

Before you begin

Stack must be up and running.

For more information, see Basic Configuration, on page 5.

After configuring the stacks, Diameter endpoints are ready to initiate/accept Diameter connections for the defined IP address, Port, and Application-ID.

Policy DRA

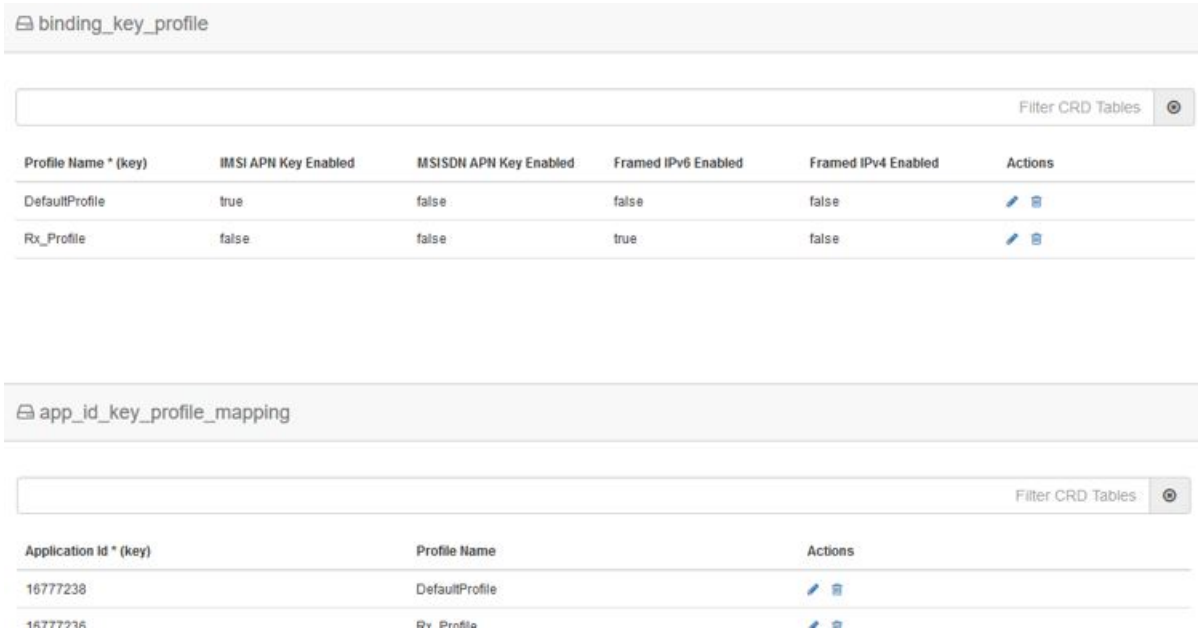
For Policy DRA, you must configure the binding keys for Gx sessions.

Binding helps Policy DRA route the related Gx/Rx sessions to the same PCRF or destined PCRF.

A binding database is needed to map search keys to PCRF binding information. Each binding has a search key and binding data associated with it. The supported search keys are:

- IMSI + APN
- IPv6
- IPv4
- MSISDN + APN

Figure 8: Policy DRA Sample Configuration



If the Binding Key Profile and mapping to Application ID is not configured properly, the following errors may occur:

- Gx Calls – Session binding failure in database resulting in call failures.
- Rx Calls – Binding Retrieval failure resulting in call failures.

Note IMS DRA does not require bindings. Hence, Binding Key Profile is only valid for Policy DRA.

Configure SRK Based Routing

You can configure SRK based routing in one of the following ways:

Configure Secondary Peer Fallback

Note This configuration is valid for both Policy and IMS DRA.

In SRK based routing, you can configure routing to a set of peers. This can be used for alternate routing (secondary and tertiary routes) when the Destination Host routing fails, and for binding data to select a peer for related diameter sessions. The Session Routing Table is configured within a "Peer Group SRK Mapping" Table. If a routing with dest-host fails, CPS vDRA will try to find out secondary routes on the basis of SRK.

Once the SRK of failed peer is determined by CPS vDRA, it will try to find an UP peer that is a member of:

- A peer group matching the entire SRK label. If it finds one, it will route the message to that peer.
- If it cannot find one and it is a two-label SRK, then it will try to find an UP peer in a peer group whose label2 part of its SRK matches the label2 part of the lookup SRK (where the label 1 part may be different). If it finds one, then it will route the message to that peer.

The following screenshot shows an example of SRK configuration:

Figure 9: SRK Configuration

peer_group_srk_mapping

Filter CRD Tables

Peer Group * (key)	Session Routing Key	Destination Host Routing Rule *	Destination Host Replace	Destination Realm Replace	Actions
GX_DC_1		preferred			
SD_DC_2		preferred			
GX_DC_2		preferred			
GX_DC_3	clusterb.dc1	preferred			
SD_DC_3	clusterb.dc2	preferred			
RX_DC_2	clusterb.dc1	preferred			

Configure Binding Retriever for Rx Calls

Note This configuration is valid for Policy only.

Rx (AAR) Message processing: Policy DRA receives the AAR request from Application Function (AF). AAR messages does not have destination host and the destination PCRF has to be found out by vDRA using the keys such as:

1. Framed Ipv6 Address

2. Framed IP address
3. IMSI APN key
4. MSISDN APN key

Binding is created by vDRA when Gx-CCRI is received at DRA. DRA creates the bindings on the basis of CRD configurations and the availability of AVPs in Gx message. If the configured keys are present in Gx message, vDRA creates and stores the binding in Binding Database. On receiving AAR request, vDRA searches for the session stored in bindings on the basis of Rx profile, and will determine the SRK of Gx-PCRF peer. DRA will then forward the Rx request to the Rx peer having the same SRK. [SRK will be configured as mentioned in following section].

Configure SLF Based Routing

Note This configuration is valid for IMS DRA only.

SLF Routing works with two major configurations and tables within CRD:

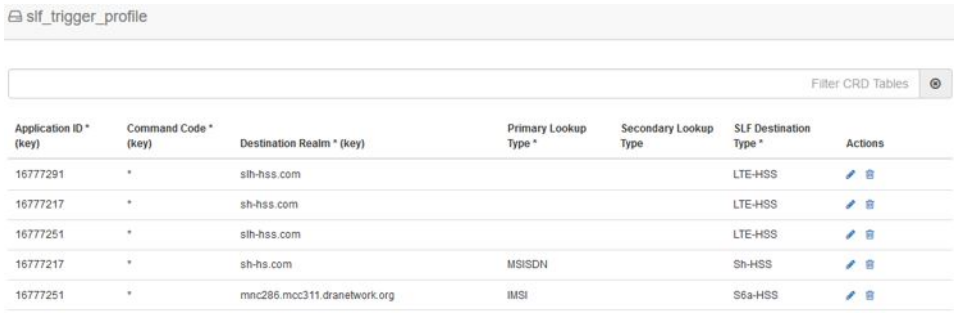
1. SLF Trigger Profile: For the incoming Diameter requests where Destination-Host is not present (or Destination-Host is present with same of DRA-Host Name) this SLF Trigger table is triggered. In this Table, there are three inputs that you need to configure:
   - Application-Id: Diameter Application ID for which the SLF Query is to happen.
   - Command-Code: Diameter Command Code for which the SLF query is to happen.

Note If this field is configured with a '*', it indicates that SLF query is expected to happen for all the command codes for the specific application.

   - Destination-Realm: Destination-Realm of the Diameter Endpoint (that is, HSS/AAA) or the Destination Realm of vDRA.

     Based on the Input keys (Application-Id, Command-Code and Destination-Realm) configured, if all the entries (as mentioned above) matches with the incoming message then vDRA(SLF) picks the "SLF Lookup Type" and "SLF-Destination-Type" as configured in the SLF Trigger Table.
   - SLF LookupType- Currently the SLF LookupType can be configured as IMSI or MSISDN. Based on the configured value of IMSI or, MSISDN, vDRA (SLF) further makes a query towards the SLF-DB.
   - SLF Destination-Type-Based on the configured value in the 'SLF Destination Type', vDRA (SLF) makes a further query towards the SLF-DB.

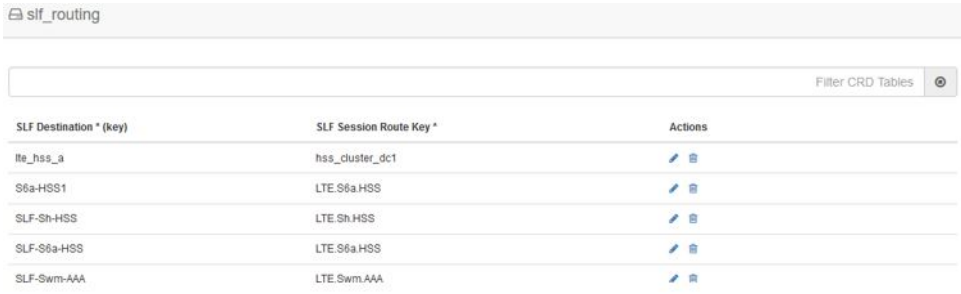
Figure 10: SLF Trigger Profile



2. SLF Routing: After vDRA(SLF) makes the query in SLF-DB based on SLF-LookupType and SLF-Destination-Type, an SLF-Destination is obtained.

   SLF Mapping table consists a mapping of 'SLF-Destination' which is obtained from the SLF database and a SLF-Session-Route-Key(SLF-SRK). In the SLF Mapping Table, based on the SLF-Destination an SLF-Session-Route-Key (SLF-SRK) is derived and further Peer group is derived for routing from the Peer-Group-Peer table as the next step for Routing.

   Figure 11: SLF Routing



Configure Table Driven Routing

CPS vDRA has the ability to use AVPs within the Diameter messages to help determine how to route the traffic.

The AVP being evaluated is customer configurable, through the CPS DRA GUI. The addition, subtraction, or modification of the evaluation AVP is dynamic and affected real time.

Trigger Condition for Table-driven Routing:

- After Destination-Host based routing (first priority) and SRK routing (second priority) conditions are not met, vDRA goes for Table-driven routing as third priority.
- Typically, for Table-driven Routing to trigger, a Diameter message contains a Destination-Realm AVP, but no Destination-Host AVP. So, If the Dest-Host AVP is absent, empty, or equal to the DRA FQDN, then we skip Dest-Host routing altogether and proceed directly to Table-Driven routing.

Note For IMS-DRA only, the router will try to do the SLF routing (if all conditions are met), before moving to table-driven routing.

vDRA can parse and has the ability to route based on the following AVPs:

- Destination-Host
- Destination-Realm
- Origin-Host
- Origin-Realm
- APN (from Called-Station-ID)
- IMSI (from Subscription-ID)
- MSISDN (from Subscription-ID)

Regular-expression matching and combinations of AVPs are also supported. The following configuration is required in Policy Builder:

1. Application Route: For more information, see Basic Configuration, on page 5.
2. Routing AVP definitions: For more information, see Basic Configuration, on page 5.
3. Search table group configurations: For more information, see Search Table Groups, on page 50.
4. CRD configuration: For more information, see Custom Reference Data Tables, on page 50.

Advanced Features

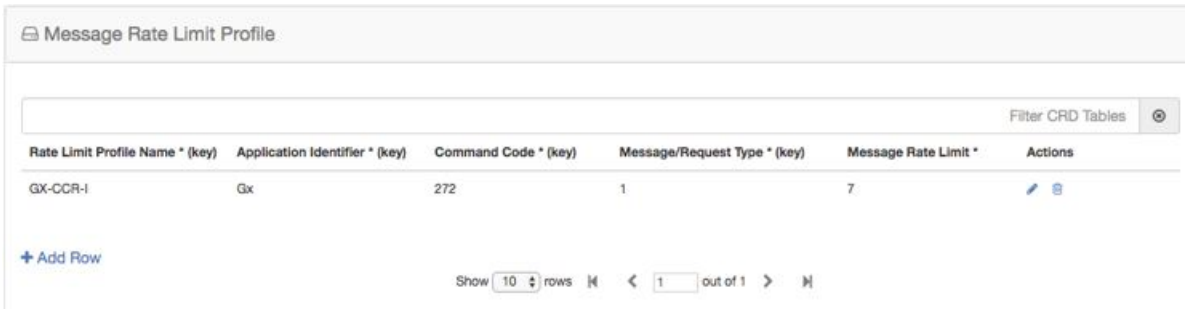
Configure Rate Limiting

You can use CPS vDRA to set rate limiting of traffic coming from and going towards a particular peer. You can configure this for both Ingress and Egress traffic. Rate limit is currently supported at peer level and message level.

1. Configure the Message Rate Limit Profile CRD table.

   Create the rate limit profile with the rate limit profile name, application ID, command code, message type, and message rate limit. For more information, see Search Table Groups – Message Rate Limit Profile table.

Figure 12: Message Rate Limit Profile

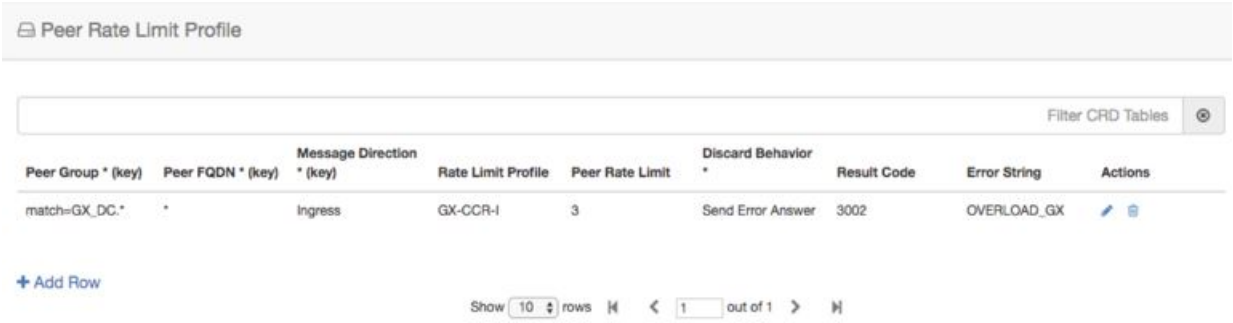


2. Configure the Peer Rate Limiting CRD table.

   Define the peer group, peer fqdn, message direction (ingress or egress), rate limit profile (created in step 1), peer rate limit, discard behavior (whether to silently drop or send error message). For more information, see Search Table Groups – Peer Rate Limit Profile table.

   If you want the discard behavior sent in the error answer, also configure the Result Code, Error String to be sent in the answer.

Figure 13: Peer Rate Limit Profile



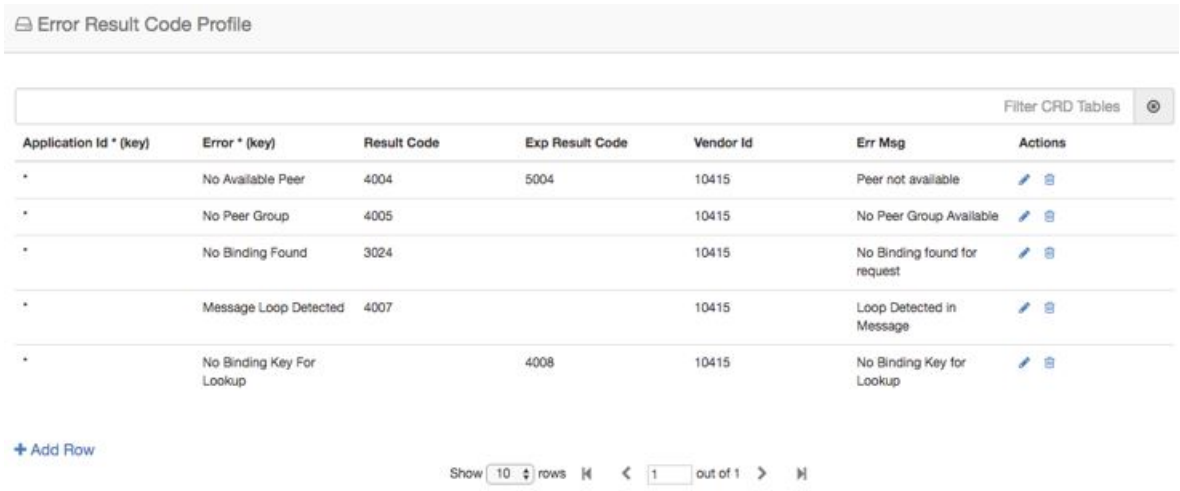
Configure Error Result Code Profile

CPS vDRA generates several internal errors that are not generic, for example, errors such as Timeout triggered, Peer not found, DB error, and so on.

You can configure error codes and error messages for internally generated errors in CPS vDRA.

To configure this error code, add entry in the Error Result Code Profile table as shown in the following image:

Figure 14: Error Result Code Profile



For a particular Application ID, if DRA generates an internal error, the error code defined in this table along with the Error Message is sent back in the answer.

If the result code is not configured and the Experimental Result Code is present, then the Experimental Result-Code is sent back in the answer along with Vendor-Id AVP. Between the Result-Code and the Experimental Result code, the Result-code is of higher priority.

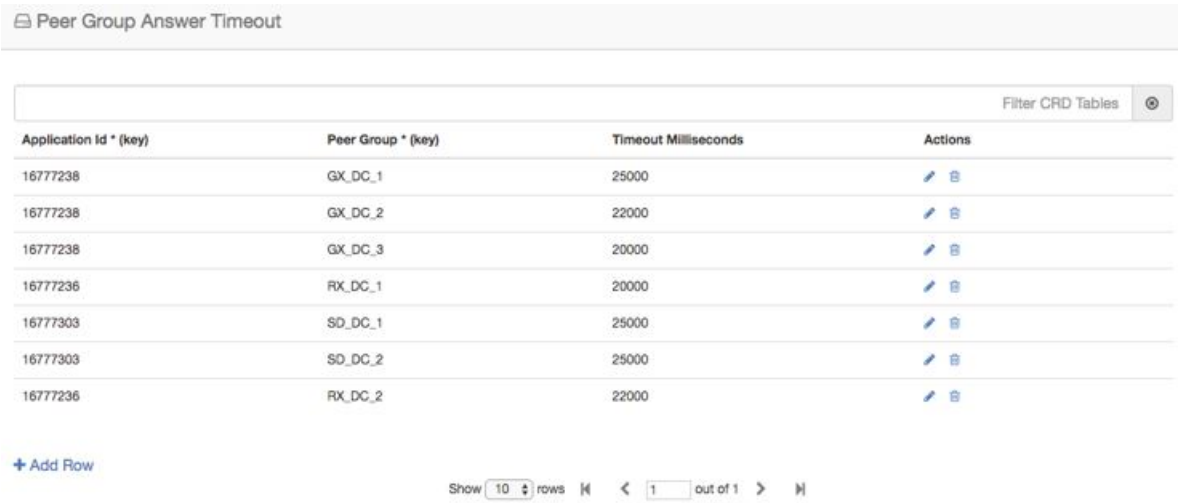
Configure Peer Group Answer Timeout

You can set the different request timeout durations for different application ID and peer group. In CPS vDRA, timeout is the amount of time that vDRA waits for the answer from the destination.

To configure this feature, add an entry in the Peer Group Answer Timeout table. The timeout value is in milliseconds.

The default timeout is 1.7 seconds. You can also override the timeout for specific interface in this table as shown in the following image:

Figure 15: Peer Group Answer Timeout

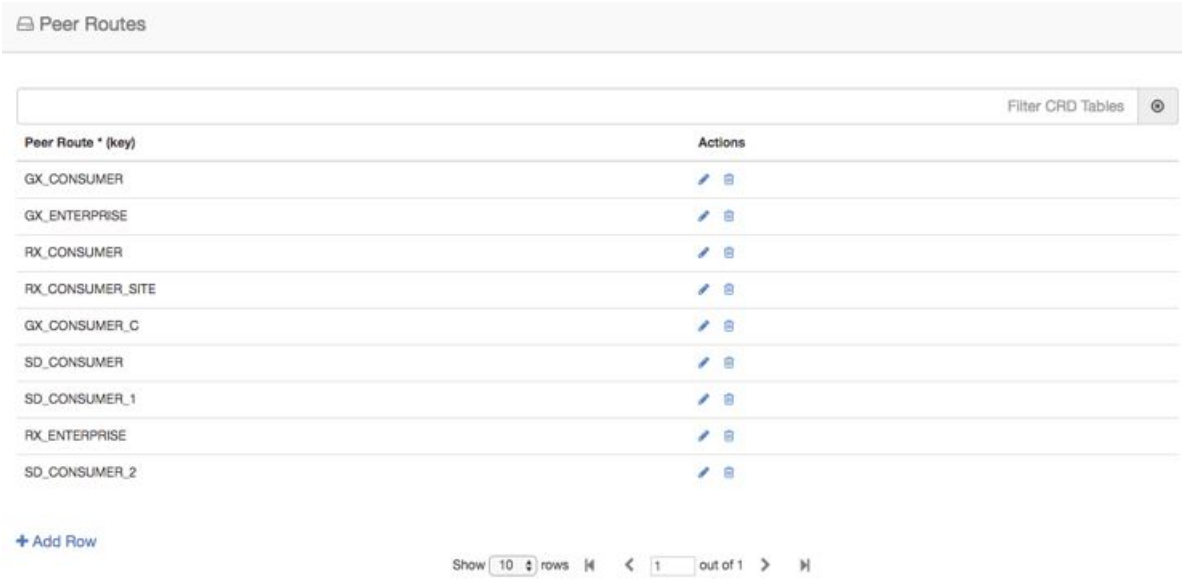


Configure Peer Load Balancing

CPS vDRA has capabilities to route messages based on weight and precedence of the peer. Perform the following steps to define the peer load balancing:

1. Configure the Peer Routes table:
   - This table defines the name of the Peer Routes that are then used in the Peer Routing table.
   - This table is mainly used in Table-driven routing where the peer route is derived from the application table-driven rule tables.

Figure 16: Peer Routes



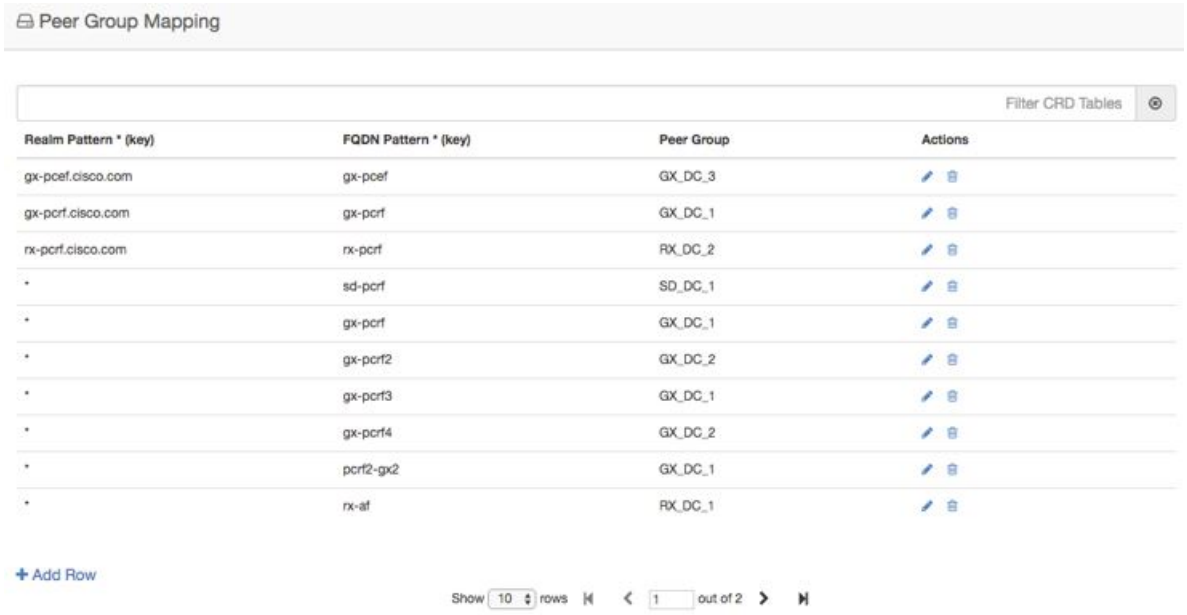
2. Configure the Peer Group Mapping table. This table defines the mapping of peers and realm with peer group.
   - Once the peer group is derived, CPS vDRA looks up this table to find the peers that belong to the derived peer group. Once DRA lists the peers in the peer group, it tries to match these peers with the Active Peer list, that is, peers which are currently connected to CPS vDRA.

     If multiple peers are up in a peer group, CPS vDRA load balances the traffic in a round-robin manner according to peer weight.

     The peer weight range is 0-1000 and the default weight is 100. If a peer weight is 0, then that peer is skipped.

     For example, if there are two peers up in a peer group with weights 100 and 200 respectively, then CPS vDRA load balances traffic between the two peers in the ratio of 33% and 67% respectively.
   - This table is applicable to SRK and Table driven routing only and is not used in Destination Host routing.

Figure 17: Peer Group Mapping

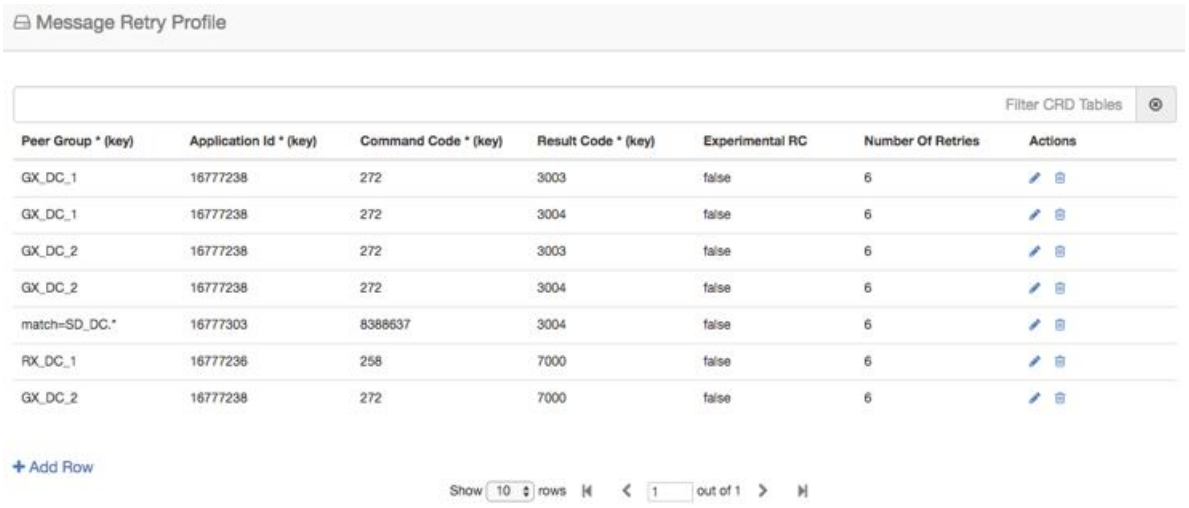


Configure Message Retries

CPS vDRA has the capability of retrying messages to multiple connected peers, in case the destination peer is not up. Retry mechanism works completely on the basis of SRK.

To configure message retries, you need to configure the Message Retry Profile table.

Figure 18: Message Retry Profile



You can configure message retry on the basis of the following inputs:

- Peer Group
- Application Id
- Command Code
- Result-Code

The outputs of this table are:

- Number of retries
- Experimental RC

When the retry criteria matches, then:

1. First, the retry is done on any connected peer in the same peer group.
2. If no peer is found in the same peer group, the next priority is given to the peers in the peer group having the same SRK as the peer group to which the request was originally sent.
3. If the above condition also fails to find a peer, the last priority is given to the peers in the peer group that share the same second label as that of the original peer group.

Note CPS vDRA uses the Peer Group Mapping table to find the peer in the same peer group. Hence, the Peer Group Mapping table configuration is a prerequisite.

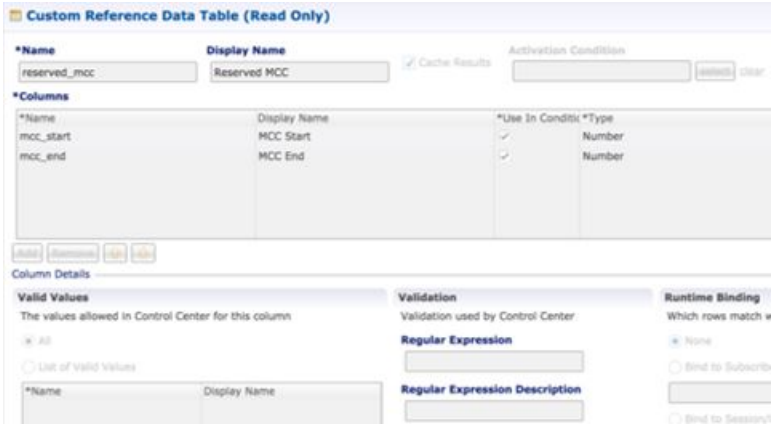
At the end, if no peer is found, retries stop. The retry also stops when the number of retries is exhausted and no response is received.

Configure Reserved IMSIs

You can now specify a reserved MCC range so that vDRA validates a parsed IMSI for SLF routing against a configured list of reserved MCC. If the IMSI matches a reserved IMSI, the value is ignored for SLF routing.

Configure the Reserved IMSI CRD table with columns for MCC Start Range and MCC End Range.

Figure 19: Reserved IMSI CRD Table



For more information, see Reserved IMSI, on page 62.

In DRA, configure the MCC Start Range and MCC End Range.

Figure 20: MCC Range

Any calls within Reserved IMSI range are either routed by alternate means such as table-driven routing or result with an error.

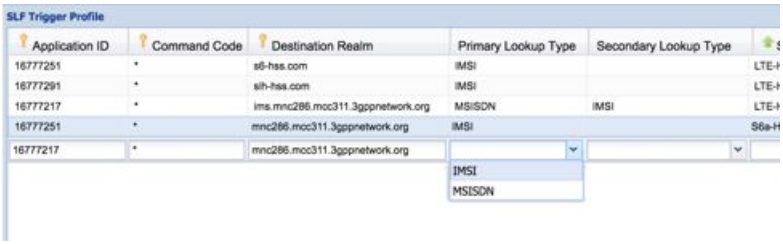
Configure Multiple Lookup in SLF Trigger Profile

Previously, in vDRA, the SLF lookup Type in the SLF trigger table had options only to support two types of lookup, that is, IMSI, MSISDN.

You can now specify Primary and Secondary Lookup Keys in the SLF Trigger Profile Table.

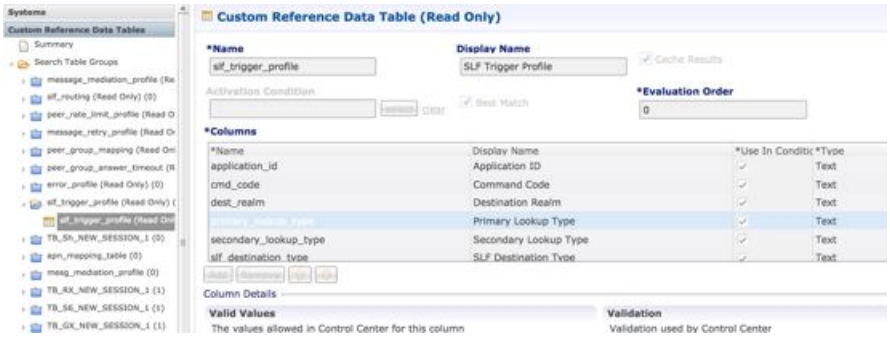
First, configure the columns in the CRD table as shown:

Figure 21: CRD Table Configuration



In the SLF Trigger Profile, you can select the Primary Lookup key from the drop down list. Similarly, you can select the Secondary Lookup key.

Figure 22: SLF Trigger Profile



Configuring MongoDB Authentication

Before you begin

- Currently, MongoDB authentication is supported only for fresh deployments of vDRA where application sharding has been implemented.
- MongoDB authentication is not supported for MongoDB sharding deployments.
- Make sure every shard of database cluster has Primary member present. Execute the following command to verify the same.

  ```
  show database status | tab | include PRIMARY
  ```
- Make sure all the VMs are in CONNECTED state.

  ```
  show docker engine
  ```
- Make sure there are no IP_NOT_REACHABLE alerts present on the system.

  ```
  show alert status | tab | include IP_NOT_REACHABLE
  ```
- Make sure the network cache is up to date with all IPs present on each VM.

  ```
  show network ips
  ```

Procedure

Step 1 Login to Binding VNF CLI and configure MongoDB authentication on single node setup:

a) Set password.

```
db-authentication set-password database mongo password *****
```

Example:

```
admin@orchestrator[an-dbmaster]# db-authentication set-password database mongo password
Value for 'password' (<string>): ********
result SUCCESS
admin@orchestrator[an-dbmaster]#
```

b) Enable transition authentication.

```
db-authentication enable-transition-auth database mongo
```

Example:

```
admin@orchestrator[an-dbmaster]# db-authentication enable-transition-auth database mongo
admin@orchestrator[an-dbmaster]#
```

c) Rolling restart of mongod instances.

```
db-authentication rolling-restart database mongo
```

Example:

```
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart database mongo
admin@orchestrator[an-dbmaster]#
```

d) Monitor rolling restart status.

```
db-authentication rolling-restart-status database mongo
```

Example:

```
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart-status database mongo
result
Rolling Restart: Not Scheduled/Completed
admin@orchestrator[an-dbmaster]#
```

e) Disable transition authentication.

```
db-authentication disable-transition-auth database mongo
```

Example:

```
admin@orchestrator[an-dbmaster]# db-authentication disable-transition-auth database mongo
admin@orchestrator[an-dbmaster]#
```

f) Rolling restart of mongod instances.

```
db-authentication rolling-restart database mongo
```

Example:

```
admin@orchestrator[an-dbmaster]# db-authentication rolling-restart database mongo
admin@orchestrator[an-dbmaster]#
```

Note
During `db-authentication rolling-restart` command execution mongod instances are restarted.

g) Make sure all the databases are UP with correct status.

```
show database status
```

Sample output:

```
admin@orchestrator[an-dbmaster]# show database status
                                                                CLUSTER
ADDRESS         PORT   NAME         STATUS     TYPE          NAME     SHARD     REPLICA SET
-------------------------------------------------------------------------------------------------
192.168.11.42  27036  arbiter-1    ARBITER    replica_set   binding  shard-1   rs-shard-1
192.168.11.43  27036  server-a     PRIMARY    replica_set   binding  shard-1   rs-shard-1
192.168.11.44  27036  server-b     SECONDARY  replica_set   binding  shard-1   rs-shard-1
192.168.11.42  27037  arbiter-2    ARBITER    replica_set   binding  shard-2   rs-shard-2
192.168.11.42  27038  arbiter-3    ARBITER    replica_set   binding  shard-2   rs-shard-2
192.168.11.43  27037  server-a     SECONDARY  replica_set   binding  shard-2   rs-shard-2
192.168.11.44  27037  server-b     PRIMARY    replica_set   binding  shard-2   rs-shard-2
192.168.11.41  27030  binding      SECONDARY  shard_db      binding  shdb-1    binding-sharddb
192.168.11.42  27030  binding      PRIMARY    shard_db      binding  shdb-2    binding-sharddb
```

h) DRA VNF Site-A and Site-B.

```
db-authentication set-password database mongo password *****
```

Example:

```
admin@orchestrator[an-master]# db-authentication set-password database mongo password
Value for 'password' (<string>): ********
result SUCCESS
admin@orchestrator[an-master]#
```

Step 2 Login to Binding VNF CLI and configure MongoDB authentication in mated pair deployments.

Note
During `db-authentication rolling-restart` command execution mongod instances are restarted.

a) On Site A, configure session-AB, imsi-msisdn databases.
b) On Site B, configure session-AB, imsi-msisdn databases.
c) On Site A, configure the password and enable-transition-auth and run rolling-restart.
d) On Site B, configure the password and enable-transition-auth and run rolling-restart.
e) On Site A, disable transition-auth and run rolling-restart
f) Make sure all the databases are UP with appropriate status.

```
show database status
```

Sample output:

```
admin@orchestrator[an-dbmaster]# show database status
                                                                CLUSTER
ADDRESS         PORT   NAME         STATUS     TYPE          NAME     SHARD     REPLICA SET
-------------------------------------------------------------------------------------------------
192.168.11.42  27036  arbiter-1    ARBITER    replica_set   binding  shard-1   rs-shard-1
192.168.11.43  27036  server-a     PRIMARY    replica_set   binding  shard-1   rs-shard-1
192.168.11.44  27036  server-b     SECONDARY  replica_set   binding  shard-1   rs-shard-1
192.168.11.42  27037  arbiter-2    ARBITER    replica_set   binding  shard-2   rs-shard-2
192.168.11.42  27038  arbiter-3    ARBITER    replica_set   binding  shard-2   rs-shard-2
192.168.11.43  27037  server-a     SECONDARY  replica_set   binding  shard-2   rs-shard-2
192.168.11.44  27037  server-b     PRIMARY    replica_set   binding  shard-2   rs-shard-2
192.168.11.41  27030  binding      SECONDARY  shard_db      binding  shdb-1    binding-sharddb
192.168.11.42  27030  binding      PRIMARY    shard_db      binding  shdb-2    binding-sharddb
```

g) DRA VNF Site-A and Site-B.

```
db-authentication set-password database mongo password *****
```

Example:

```
admin@orchestrator[an-master]# db-authentication set-password database mongo password
Value for 'password' (<string>): ********
result SUCCESS
admin@orchestrator[an-master]#
```

Disabling MongoDB Authentication

Note This section is used to disable MongoDB authentication in mated pair deployments.

Procedure

Step 1 Login to Binding VNF CLI and perform the following steps:

Note
The steps need to be performed on all binding VNFs.

a) Enable transition authentication.

```
db-authentication enable-transition-auth database mongo
```

b) Rolling restart of mongod instances.

```
db-authentication rolling-restart database mongo
```

c) Rolling restart status.

```
db-authentication rolling-restart-status database mongo
```

Step 2 Login to DRA VNF CLI and remove the password by using `db-authentication remove-password database mongo` command.

Note
The step needs to be performed on all DRA VNFs.

```
admin@orchestrator[an-master]# db-authentication remove-password
Value for 'password' (<string>): ********
result SUCCESS
admin@orchestrator[an-master]#
```

Step 3 Login to binding VNF CLI and remove the password by using `db-authentication remove-password database mongo` command.

Note
The step needs to be performed on all binding VNFs.

Step 4 Login to binding VNF CLI and perform the following steps:

Note
The steps need to be performed on all binding VNFs.

a) Disable transition authentication.

```
db-authentication disable-transition-auth database mongo
```

b) Rolling restart of mongod instances.

```
db-authentication rolling-restart database mongo
```

c) Rolling restart status.

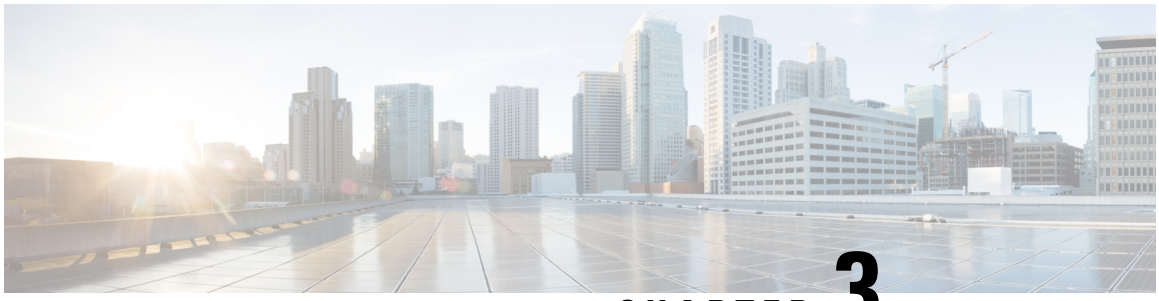
```
db-authentication rolling-restart-status database mongo
```

Step 5 Make sure all the databases are UP with appropriate status.

```
show database status
```

Sample output:

```
admin@orchestrator[an-dbmaster]# show database status
                                                                CLUSTER
ADDRESS         PORT   NAME         STATUS     TYPE         NAME     SHARD     REPLICA SET
-------------------------------------------------------------------------------------------
192.168.11.42  27036  arbiter-1   ARBITER    replica_set  binding  shard-1   rs-shard-1
192.168.11.43  27036  server-a    PRIMARY    replica_set  binding  shard-1   rs-shard-1
192.168.11.44  27036  server-b    SECONDARY  replica_set  binding  shard-1   rs-shard-1
192.168.11.42  27037  arbiter-2   ARBITER    replica_set  binding  shard-2   rs-shard-2
192.168.11.42  27038  arbiter-3   ARBITER    replica_set  binding  shard-2   rs-shard-2
192.168.11.43  27037  server-a    SECONDARY  replica_set  binding  shard-2   rs-shard-2
192.168.11.44  27037  server-b    PRIMARY    replica_set  binding  shard-2   rs-shard-2
192.168.11.41  27030  binding     SECONDARY  shard_db     binding  shdb-1    binding-sharddb
192.168.11.42  27030  binding     PRIMARY    shard_db     binding  shdb-2    binding-sharddb
```



CHAPTER 3

Policy Builder Configuration

Plug-in Configuration

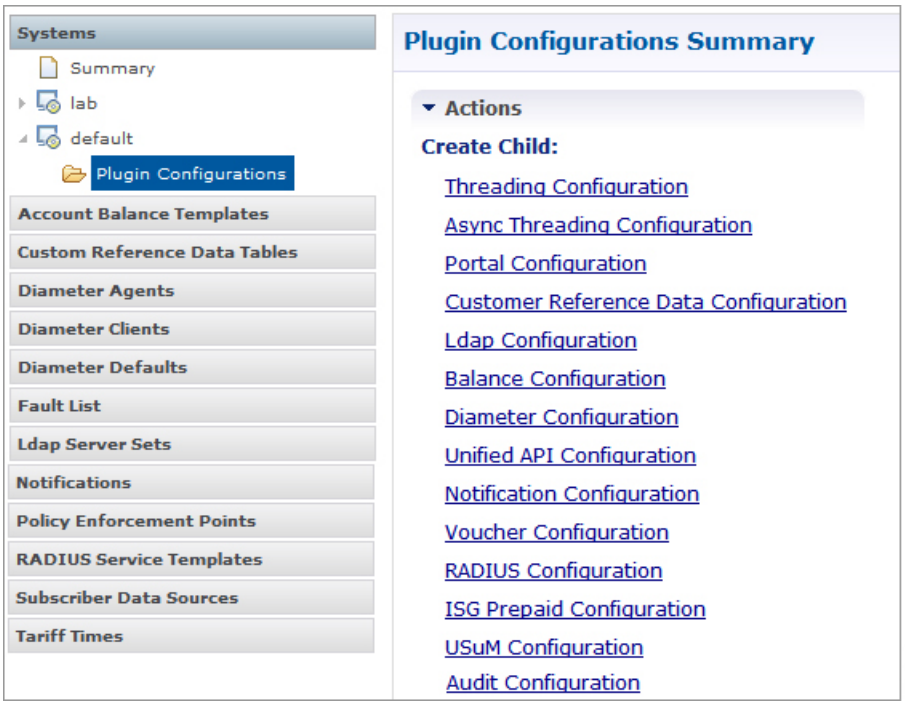
Cisco Policy Builder provides core plug-ins for customizing and optimizing your installation.

- Configurations set at the system level are system-wide except as noted in the bullet items below.
- Configurations set at the cluster level apply to that cluster and the instances in it. A value set here overrides the same value set at the system level.
- Configurations set at the instance level apply to the instance only and override the same value set at the cluster or system level.

Select the **Create Child** action in a **Plug-in Configuration** node in the **Systems** tree to define them. You can change any of the variables from the default, or choose not to use a plug-in, as necessary.

When you create a system from the example, the following configuration stubs appear at the cluster and instance level:

Figure 23: Create Child Action



Threading Configuration

A threading configuration utility is provided for advanced users.

Click **Threading Configuration** in the right pane to add the threading configuration to the system. If you are planning to run the system with higher TPS, then you need to configure Threading Configuration. For further information, contact your Cisco Technical Representative.

The Threading Plug-in having thread pools controls the total number of threads in CPS vDRA that are executing at any given time. Each of these thread pools have a queue associated with it.

The following parameters can be configured under Threading Configuration:

Table 1: Threading Configuration Parameters

Parameter	Description
Thread Pool Name	Name of the thread pool. For more information on the thread pool names and recommended values that can be configured, refer to *Threading Configuration* section in the *CPS vDRA Advanced Tuning Guide*.
Threads	Number of threads to set in the thread pool.
Queue Size	Size of the queue before they are rejected.
Scale By Cpu Core	Select this check box to scale the maximum number of threads by the processor cores.

Async Threading Configuration

Click **Async Threading Configuration** in the right pane to add the configuration in the system.

Use the default values for the Async Threading Plug-in. The Async configuration controls the number of asynchronous threads.

Note Currently, CPS vDRA does not have any asynchronous threads. However, you must add "Async Threading Configuration" and keep this table empty.

The following parameters can be configured under Async Threading Configuration.

Table 2: Async Threading Configuration

Parameter	Description
Default Processing Threads	The number of threads that are allocated to process actions based on priority.
Default Action Priority	The priority assigned to an action if it is not specified in the Action Configurations table.
Default Action Threads	The number of threads assigned to process the action if it is not specified in the Action Configurations table.
Default Action Queue Size	The number of actions that can be queued up for an action if it is not specified in the Action Configurations table.
Default Action Drop	**DropOldestWhenFull**: The oldest queued action is dropped from the queue when a new action is added to a full queue. Otherwise, the new action to add is ignored. **DropWhenFull**: A handler for rejected tasks that silently discards the rejected task. No execution for rejected tasks. **DoNotDrop**: A handler for rejected tasks that runs the rejected task directly in the calling thread of the execute method, unless the executor has been shut down, in which case the task is discarded. Default value is **DropOldestWhenFull**.
Action Configurations Table	
Action Name	The name of the action. This must match the implementation class name.
Action Priority	The priority of the action. Used by the default processing threads to determine which action to execute first.
Action Threads	The number of threads dedicated to processing this specific action.
Action Queue Size	The number of actions that can be queued up.

Parameter	Description
Action Drop Oldest When Full	For the specified action only: When checked, the oldest queued action is dropped from the queue when a new action is added to a full queue. Otherwise, the new action to add is ignored.

Custom Reference Data Configuration

Configure your system, cluster, and instance for the first time to use Custom Reference Data Table plug-in. Then you can create as many tables as needed.

Click **Custom Reference Data Configuration** from right pane to add the configuration in the system.

- HA example:
  - Primary Database Host/IP Address: sessionmgr01
  - Secondary Database Host/IP Address: sessionmgr02
  - Database Port: 27717

The following parameters can be configured under Custom Reference Data Configuration.

Table 3: Custom Reference Data Configuration Parameters

Parameter	Description
Primary Database Host/IP Address	IP address or a host name of the sessionmgr database. For example, sessionmgr01.
Secondary Database Host/IP Address	(Optional) This field is the IP address or a host name of a secondary, backup, or failover sessionmgr database. For example, sessionmgr02.
Database Port	Port number of the sessionmgr. **Note** Make sure that the value for this field is same as filled in for both the Primary Database Host/IP Address and Secondary Database Host/IP Address fields. Default value is 27717.

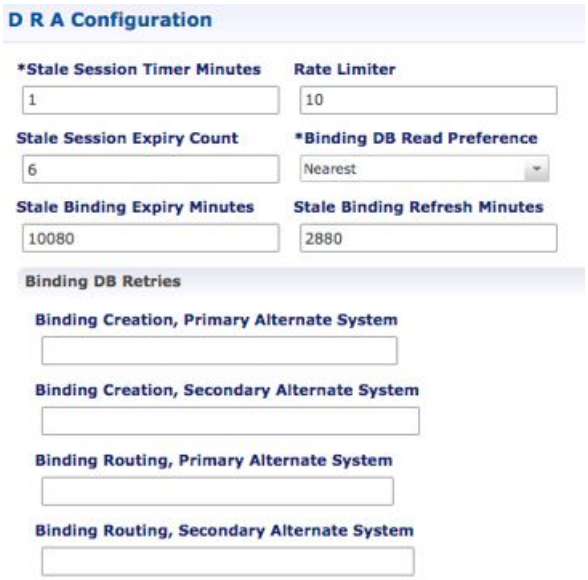
Parameter	Description
Db Read Preference	Describes how sessionmgr clients route read operations to members of a replica set. Select one of the following options from drop-down list: • Primary: All operations read from the current replica set primary member. • PrimaryPreferred: In most situations, operations read from the primary database host. However, if this host is unavailable, operations read from the secondary databse host. • Secondary: All operations read from the secondary members of the replica set. • SecondaryPreferred: In most situations, operations read from secondary members. However, if a secondary database host is unavailable, operations read from the primary database host. Default value is Primary. For more information, see http://docs.mongodb.org/manual/core/read-preference/.
Connection Per Host	Number of connections that are allowed for each database host. Default value is 100. Connection Per Host is a performance tuning parameter and can be changed in case of a performance issue according to the call model and hardware.

For more information on Custom Reference Data API Usage, see the *CPS Operations Guide* for this release.

DRA Configuration

Click **DRA Configuration** from the right pane in Policy Builder to add the configuration in the system.

Figure 24: DRA Configuration



The following parameters can be configured under DRA Configuration:

Table 4: DRA Configuration Parameters

Parameter	Description
Stale Session Timer Minutes	Indicates the time after which the audit RAR should be generated (in the subsequent audit RAR process cycle that runs every minute in CPS vDRA) for sessions that are stale. Default: 180 minutes (recommended value) Minimum: 10 minutes Maximum: 10080 minutes
Rate Limiter	Indicates the number of audit RARs per second that should be sent out by CPS vDRA. Minimum: 1 Maximum: 1000 (maximum number of RAR messages per second from vDRA to PCEF) For information on recommended value, refer to *Audit Rate Limiter* section in the *CPS vDRA Advanced Tuning Guide*.

Parameter	Description
Stale Session Expiry Count	Specifies the number of retries vDRA should do for a stale session if there is no response of audit RAR or if there is Result-Code in RAA (for audit RAR) other than 5002 or 2001. Default: 6 Minimum: 0 (Session deleted without sending RAR) Maximum: 10 For information on recommended value, refer to *Audit Rate Limiter* section in the *CPS vDRA Advanced Tuning Guide*.
Binding DB Read Preference	Used to select the mode when reading from Binding DB. Use "nearest" mode for better performance of traffic that needs only read operation on Binding DB. Default: Nearest For information on recommended value, refer to *Audit Rate Limiter* section in the *CPS vDRA Advanced Tuning Guide*.
Stale Binding Expiry Minutes	Duration after which a binding record is validated against a session record to see if the binding should be deleted because it is stale The timer is initialized when the session is created. The records are deleted when binding expiry time is reached and no active session is found. Otherwise, the timer is updated so the binding record can be audited after another Stale Binding Expiry Minutes. Default: 10080 minutes (168 hours or one week) (recommended value) Minimum: 10 minutes Maximum: 43200 minutes (28 days)
Stale Binding Refresh Minutes	Duration for which the expiry time of the binding database records is refreshed. Default: 2880 minutes (48 hours or 2 days - recommended value). Minimum: 10 minutes Maximum: 10080 minutes (one week) **Note** Stale Binding Refresh Minutes should be greater than Stale Session Timer Minutes. **Important** **Stale Binding Refresh Minutes** parameter has been deprecated from CPS 19.5.0 and later releases. It is recommended to not set this value as zero.

Parameter	Description
Binding Creation, Primary Alternative System	Name of vDRA system to retry Gx CCR-i When vDRA tries to route a Gx CCR-i request, but is unable to reach the database, the configured values of first the primary, then the secondary systems are used to route the Gx CCR-i to a different vDRA to try the database. The retry is stopped if that vDRA also cannot reach the database.
Binding Creation, Secondary Alternative System	Name of secondary vDRA system to retry Gx CCR-i
Binding Routing, Primary Alternative System	Name of vDRA system to retry Rx AAR When vDRA tries to route a Rx AAR request, but is unable to reach the database, the configured values of first the primary, then the secondary systems are used to route the Rx AAR to a different vDRA to try the database. The retry is stopped if that vDRA also cannot reach the database.
Binding Routing, Secondary Alternative System	Name of secondary vDRA system to retry Rx AAR
Settings	Refer to Settings.
Rate Limits	Refer to Rate Limits.
DRA Feature	Refer to DRA Feature.
DRA Inbound Endpoints	Refer to DRA Inbound Endpoints, on page 39.
Relay Endpoints	Refer to Relay Endpoints, on page 41.

Settings

Click **Settings** check box to open the configuration pane.

The following parameters can be configured under **Settings**:

Table 5: DRA Configuration - Settings Parameters

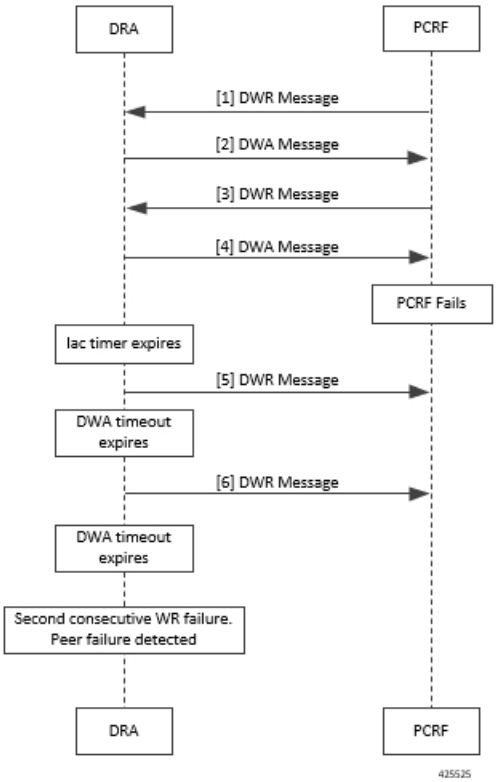
Parameter	Description
Stop Timeout Ms	Determines how long the stack waits for all resources to stop. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)

Parameter	Description
Cea Timeout Ms	Determines how long it takes for CER/CEA exchanges to timeout if there is no response. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)
Iac Timeout Ms	Determines how long the stack waits before initiating a DWR message exchange on a peer connection from which no Diameter messages have been received. The timeout value is in milliseconds. Default: 5000 ms (recommended value) Minimum: 1000 ms Maximum: 30000 ms (30 seconds)
Dwa Timeout Ms	Determines how long the stack waits for a DWA message in response to a DWR message. If no Diameter message (DWA or other message) is received on the peer connection during the first timeout period, the stack counts a failure, sends another DWR message, and restarts the Dwa timer. If no Diameter messages are received during the second timeout period, the stack counts a second failure. After two consecutive failures, the stack considers the peer connection as failed, and closes the connection. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)
Dpa Timeout Ms	Determines how long it takes for a DPR/DPA exchange to timeout if there is no response. The delay is in milliseconds. Default: 5000 ms (recommended value) Minimum: 1000 ms Maximum: 30000 ms (30 seconds)

Parameter	Description
Rec Timeout Ms	Determines how long it takes for the reconnection procedure to timeout. The delay is in milliseconds. Default: 10000 ms (recommended value) Minimum: 1000 ms Maximum: 60000 ms (one minute)
Drain Timeout Ms	Indicates the time that a peer connection remains open for responses to be sent to peers even if DPR is sent or received by vDRA. If a DPR is sent or received by vDRA, vDRA does not route requests to the disconnecting peer connection via any routing (Dest-Host, SRK, Binding, Table-Driven). However, responses and in-flight requests sent to the corresponding peers till the duration of Drain Timeout. This allows vDRA to gracefully shut down when any remote peer sends a DPR so as to minimize the diameter message loss. Default: 2000 ms Maximum: Must be less than Dpa timeout Ms **Note** When vDRA initiates DPR and the remote end PCRF/PGW disconnects TCP connection immediately after sending DPA, response for the in-flight requests are dropped before reaching the configured drain timeout value.

The following figure illustrates the timers in peer detection:

Figure 25: vDRA Peer Detection Failure



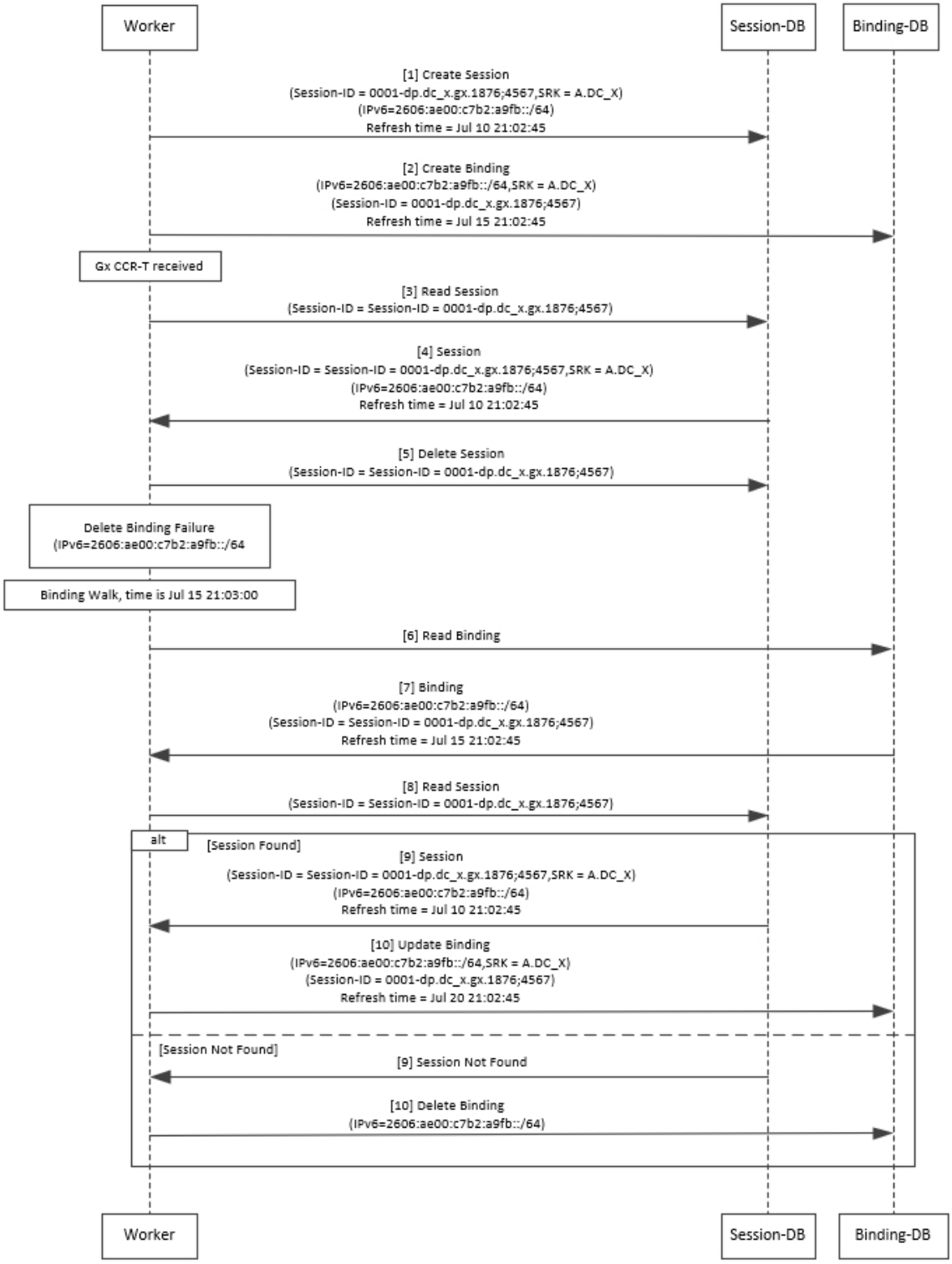
Binding DB Audit

The Binding DB Audit automatically deletes stale records from the binding DBs. When a Gx session record is created, binding records for the session binding keys are also created. When each binding record is created, the binding record expiry time is initialized to the sum of the session creation time and the Stale Binding Expiry Minutes (that you can configure in Policy Builder).

A binding record is deleted when the corresponding session record is deleted. A binding may become stale if it cannot be deleted when its associated session record is deleted (this occurs typically due to database communication failures). The binding records are audited using a binding audit background process. If the audit process finds a binding record with an expiry time in the past, the binding record is checked for staleness by checking the session database for the corresponding session record. If an active session record is found, the binding record expiry time is updated with sum of current time and the Stale Binding Expiry Minutes. If an active session is not found, the binding is considered stale and is deleted. Note that the binding audit process does not perform any Diameter signaling with the GW before deletion.

The following figures illustrate the working of binding DB:

Figure 26: DRA Binding Audit, Stale Binding Cleanup



Rate Limits

Rate limit per process instance on Policy Director (lb) VM can be managed using this configuration.

Default is unchecked, that is, no rate limits for Diameter traffic (recommended setting).

If enabled, the following parameters can be configured under **Rate Limits**:

Table 6: DRA Configuration - Rate Limits

Parameter	Description
Rate Limit per Instance on Policy Director	Allowable TPS on a single instance of policy server (QNS) process running on the Policy Director. Minimum: 1 Maximum: 5000 **Note** Contact your Cisco representative for usecase-specific recommended values.
Result-Code in Response	Indicates the error code that must be used while rejecting requests, due to rate limits being reached. Default: 3004
Error Message in Response	Select the check box to drop the rate-limited messages without sending error response. If the check box is not selected, then the rate limited message are dropped with error response as configured.

Parameter	Description
Drop Requests Without Error Response	Select the check box to drop rate limited messages without sending error response. If the check box is unchecked, then the rate limited messages are dropped with error response as configured. To accommodate configuration to either drop the request or send an error response, a column *Discard Behavior* can be added under Peer Rate Limit Profile. The column may have one of the two possible values: • Send Error Response • Drop Message Default: Unchecked (recommended setting) For more information, refer to Peer Rate Limit. **Important** If both Rate Limit Error Code and Rate Limit Error String are provided along with Rate Limit Action as "Drop Message", the Rate Limit Action will take precedence and the other two fields will be ignored.

Here is the list of the available combinations for rate limiting:

Table 7: Rate Limiting Combinations

Rate Limiting Type	With Error Code	With Error Code and Error Message	Without Error Code (Drop)
Instance Level	Yes	Yes	Yes
Peer Level Egress	Yes	Yes	Yes
Peer Level Egress with Message Level	Yes	Yes	Yes
Egress Message Level (No Peer Level RL)	Yes	Yes	Yes
Peer Level Ingress	Yes	Yes	Yes
Peer Level Ingress with Message Level	Yes	Yes	Yes
Ingress Message Level (No Peer Level RL)	Yes	Yes	Yes

DRA Feature

Click **DRA Feature** check box to open the configuration pane.

The following parameters can be configured under **DRA Feature**:

Table 8: DRA Features

Parameter	Description
Gx Session Tear Down On5065	By default, **Gx Session Tear Down On5065** flag is enabled (recommended setting). When the PCRF responds with a Experimental Result Code of 5065 in AAAnswer on Rx Interface, DRA deletes its internal binding and session created for the transaction.A RAR with appropriate Session-Release-Cause AVP will also be sent to the PCEF. **Important** When using this flag, there will always be a database query to fetch Gx session id. So this means that the database transactions will linearly increase with AAR traffic on Rx Interface.
Update Time Stamp On Success R A A	When this check box is selected, session timestamp will be updated on receipt of success RAA (Result-Code: 2001) from PCEF. [1] Default is checked (recommended setting). **Important** When using this flag, there will always be a database query to fetch Gx session id. So this means that the database transactions will linearly increase with AAR traffic on Rx Interface.

[1] The time stamp is updated on generation of Stale RAR. Also, if a success RAR/RAA(2001) comes after generation of Stale RAR, then the Stale RAR counter is reset.

DRA Inbound Endpoints

The following parameters can be configured under **DRA Inbound Endpoints**:

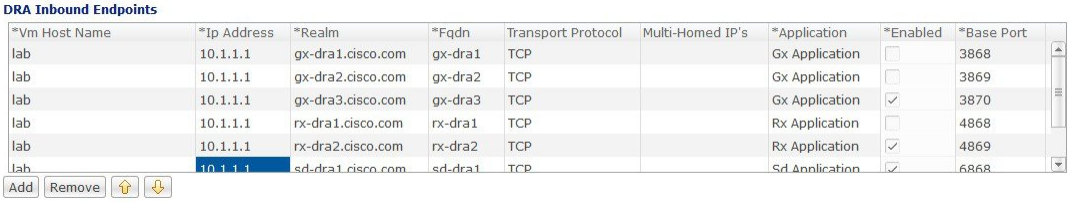
Table 9: DRA Configuration - DRA Inbound Endpoints Parameters

Parameter	Description
Vm Host Name	Host Name of the VM that hosts this CPS vDRA endpoint.
Ip Address	Address on which this CPS vDRA endpoint should bind to.
Realm	Realm of the CPS vDRA endpoint.
Fqdn	Fully Qualified Domain Name of the CPS vDRA end point.

Parameter	Description
Transport Protocol	Allows you to select either TCP' or 'SCTP' for the selected DRA endpoint. Default value is TCP. If the DRA/relay endpoint is to be configured for SCTP, the Transport Protocol should be selected as SCTP for those endpoints.
Multi-Homed IPs	This is a comma separated list of IP addresses that CPS vDRA will use to start the diameter stack with multi-homing enabled for SCTP transport. Diameter stack with TCP transport will still use the existing 'Local Bind Ip' field to specify any specific IP address for TCP stack. CPS vDRA will use the 'Local Bind Ip' to bring up SCTP stack and use it along with the ‘Multi Homing Hosts' to start the SCTP transport with multi-homing support. **Note** While using SCTP multi-homing functionality review the Linux network and gateway configurations for supporting multiple networks on different subnets. CPS supports Centos 6 release and reverse path filtering kernel parameter (rp_filter) values can be set for allowing packets from different subnets on Policy Director VMs. The default behavior in Centos 6 is to discard the packets in such scenarios. The configuration for multi-homing is validated by netstat command on lb01: `netstat -apn | grep 3898`
Application	Refers to 3GPP Application ID of the interface. You can select multiple applications on a peer connection. For example, S6a and SLg on a single IPv4/SCTP Multi-homed peer connection.
Enabled	Check to enable the endpoint.
Base Port	Refers to the port on which the CPS vDRA listens for incoming connections.

An example configuration is shown below:

Figure 27: DRA Inbound Endpoints - Example Configuration



Relay Endpoints

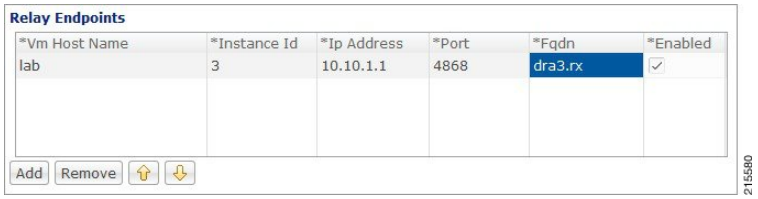
The following parameters can be configured under **Relay Endpoints**:

Table 10: DRA Configuration - Relay Endpoints Parameters

Parameter	Description
Vm Host Name	Host Name of the VM that hosts this Relay endpoint.
Instance Id	Instance Identifier is the ID of the current Instance.
Ip Address	Address on which this DRA endpoint should bind to.
Port	Port is the listening port for this instance.
Fqdn	Fully Qualified Domain Name of the DRA end point.
Enabled	Check to enable endpoint.

An example configuration is shown below:

Figure 28: Relay Endpoints - Example Configuration



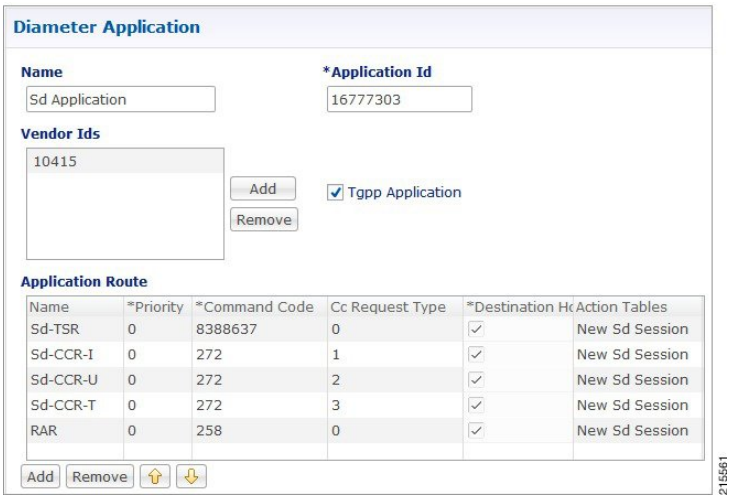
Diameter Application

Sd Application

For Sd, an Application Routing table is used to map specific diameter command codes and CC-Request-Types to a table, typically, an Sd New Session table for routing Sd TSRs to a peer route. The Sd New Session CD table will choose a peer route based on the Destination-Realm. The peer route will then point to a Peer-Group which contains multiple peer connections to a TDF and the DRA will load balance among the TDF peer connections in the Peer Group.

An example configuration is shown below:

Figure 29: Diameter Application - Sd Application Example



The following parameters are configured under Sd Application:

Table 11: Sd Application Parameters

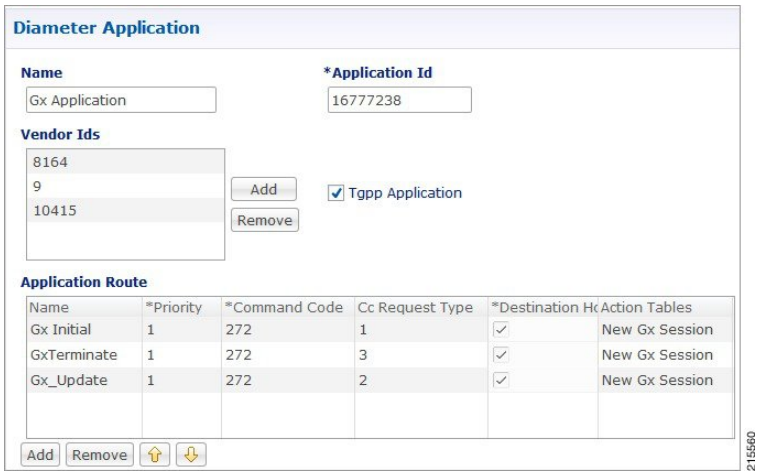
Parameter	Description
Name	Name of the Sd application.
Application Id	16777303, 3GPP specified Application Identifier for Sd interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Sd interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.
Cc Request Type	Indicates if the Credit Control Request type is Initial(1)/Update(2) or Terminate(3).
Destination Host Null	If this check box is selected, indicates if Destination Host will be null in messages received for this application.
Action Tables	Identifies the request routing table for this interface and message.

Gx Application

For Gx, an Application Routing table is used to map specific diameter command codes and CC-Request-Types to a table. When “Destination Host Null” is checked, it means Destination-Host AVP is null. It will then check for table driven routing.

An example configuration is shown below:

Figure 30: Diameter Application - Gx Application Example



C-DRA attempts to do Dest-Host routing before doing table driven routing. If the Dest-Host AVP is absent, empty, or equal to the CDRA FQDN, then we skip Dest-Host routing altogether and proceed to Table-Driven routing.

The following parameters are configured under Gx Application:

Table 12: Gx Application Parameters

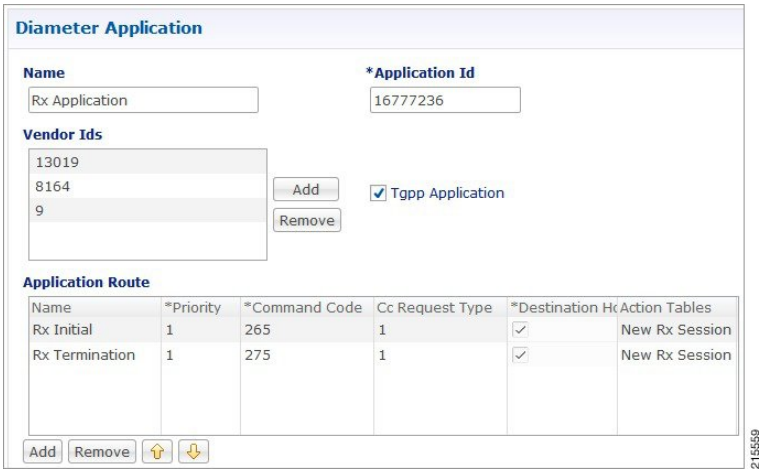
Parameter	Description
Name	Name of the Gx application.
Application Id	16777238, 3GPP specified Application Identifier for Gx interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Gx interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.

Parameter	Description
Cc Request Type	Indicates if the Credit Control Request type is Initial(1)/Update(2) or Terminate(3).
Destination Host Null	If this check box is selected, indicates the message will contain a Destination-Host.
Action Tables	Identifies the request routing table for this interface and message.

Rx Application

Identifies the request routing table for this interface and message.

Figure 31: Diameter Application - Rx Application Example



The following parameters are configured under Rx Application:

Table 13: Rx Application Parameters

Parameter	Description
Name	Name of the Rx application.
Application Id	16777236, 3GPP specified Application Identifier for Rx interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Rx interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application Route table	
Name	Identifier of the route.

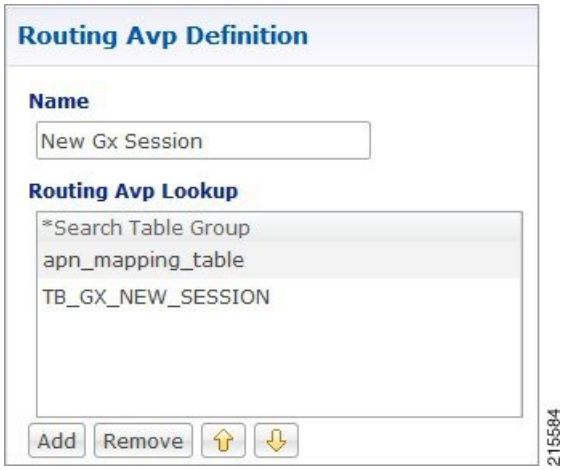
Parameter	Description
Priority	Indicates the priority of the route.
Command Code	Indicates value of command code AVP within the message.
Cc Request Type	Not supported for Rx interface.
Destination Host Null	If this check box is selected, indicates if Destination Host will be null in messages received for this application.
Action Tables	Identifies the request routing table for this interface and message.

Routing AVP Definition

Gx Session

An example configuration is shown below:

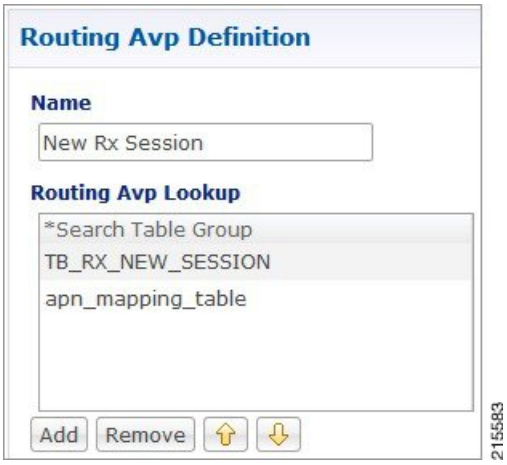
Figure 32: Routing AVP Definition - Gx Session



Rx Session

An example configuration is shown below:

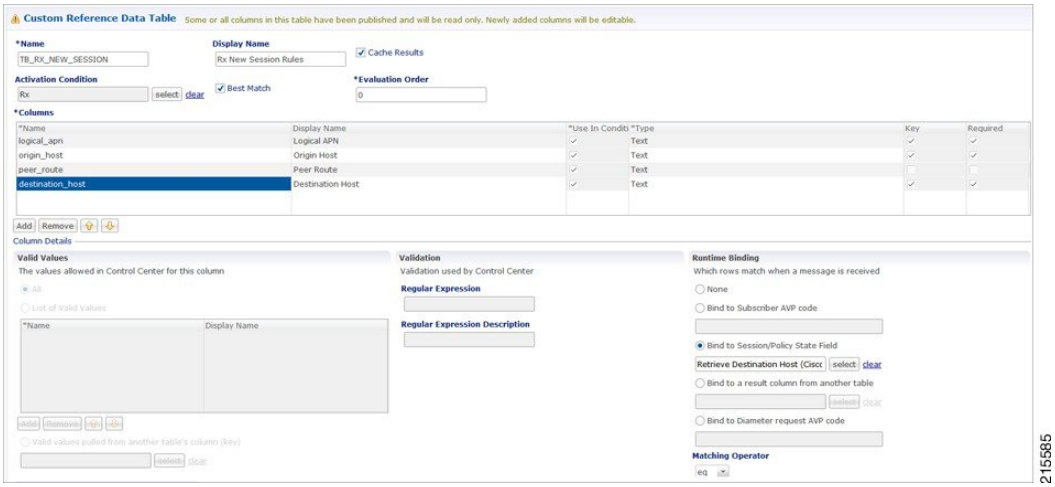
Figure 33: Routing AVP Definition - Rx Session



Rx New Session Rules - CRD Table

An example configuration is shown below:

Figure 34: Rx New Session Rules - CRD Table

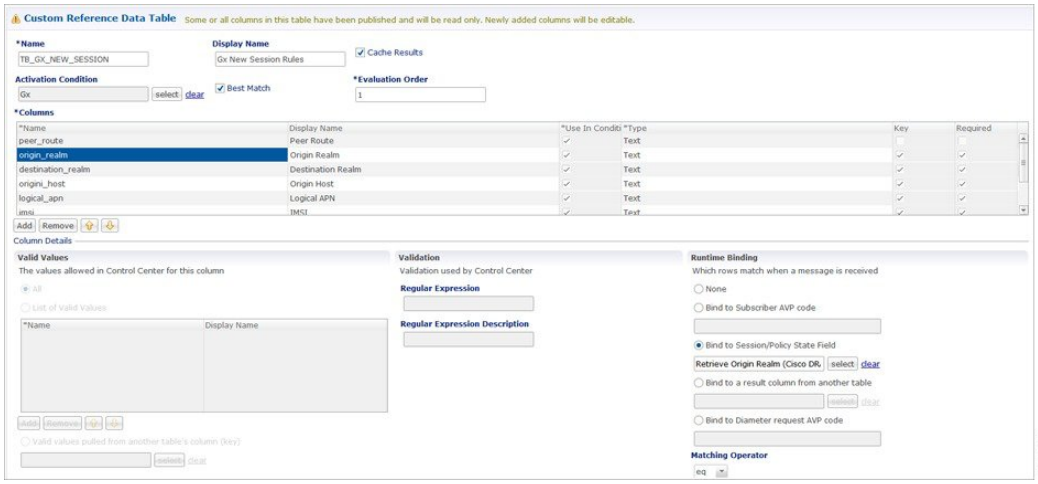


Gx New Session Rules - CRD Table

For Gx, an Application Routing table is used to map specific diameter command codes and CC-Request-Types to a table, typically, for routing Gx CCR-Is. The Gx CCR-I should be routed based on a logical APN and the Origin-Host attribute. Regular expression matching of logical APNs and Origin-Hosts can also be configured. The implementation should be flexible to allow CRDs to be configured for routing of other attributes such as Destination-Realm and Origin-Realm.

An example configuration is shown below:

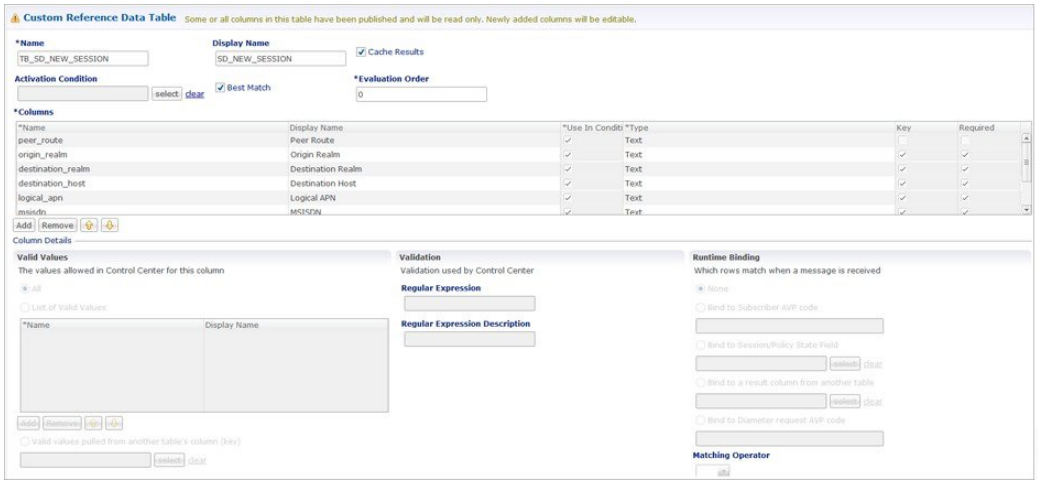
Figure 35: Gx New Session Rules - CRD Table



Sd New Session Rules - CRD Table

An example configuration is shown below:

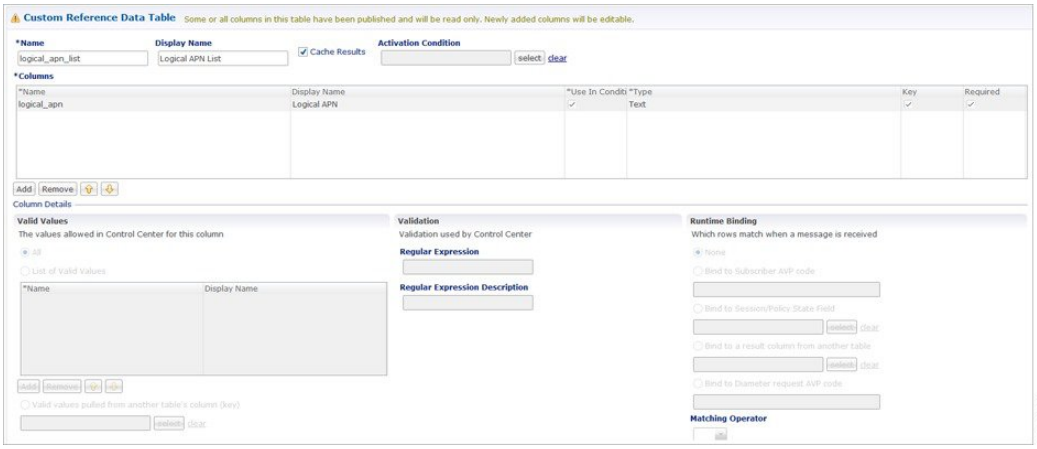
Figure 36: Sd New Session Rules - CRD Table



Logical APN List - CRD Table

An example configuration is shown below:

Figure 37: Logical APN List - CRD Table



Dynamic AVP Retriever for Routing

DRA supports routing messages based on the following AVPs from request message:

- Destination-Host
- Destination-Realm
- Origin-Host
- Origin-Realm
- APN (from Called-Station-ID)
- IMSI (from Subscription-ID)
- MSISDN (from Subscription-ID)

Regular-expression matching and combinations of AVPs is supported. This requirement is not applicable across all messages on different interfaces. The following table shows applicability of the AVP's at a message and interface level.

Table 14: Regular-expression Matching and Combinations of AVPs

Interface	Message	Origin Host	Origin Realm	Destination Host	Destination Realm	APN (Called-Station-ID)	IMSI	MSISDN
Gx	CCR-I	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	CCR-U	No	No	No	No	No	No	No
	RAR	No	No	Yes	No	No	No	No

Interface	Message	Origin Host	Origin Realm	Destination Host	Destination Realm	APN (Called-Station-ID)	IMSI	MSISDN
Sd	TSR	Yes	Yes	Yes	Yes	No	No	No
	CCR-I	Yes	Yes	Yes	Yes	No	No	No
	CCR-U/T	No	No	Yes	No	No	No	No
	RAR	No	No	Yes	No	No	No	No
Rx	RAR	No	No	Yes	No	No	No	No

Dynamic AVP Retrievers are used mostly used in Custom Reference Data where data has to be fetched from messages at runtime.

Configure Dynamic AVP Retriever

The following sample configuration shows how to retrieve the AVP and bind it to a Key Column in the CRD.

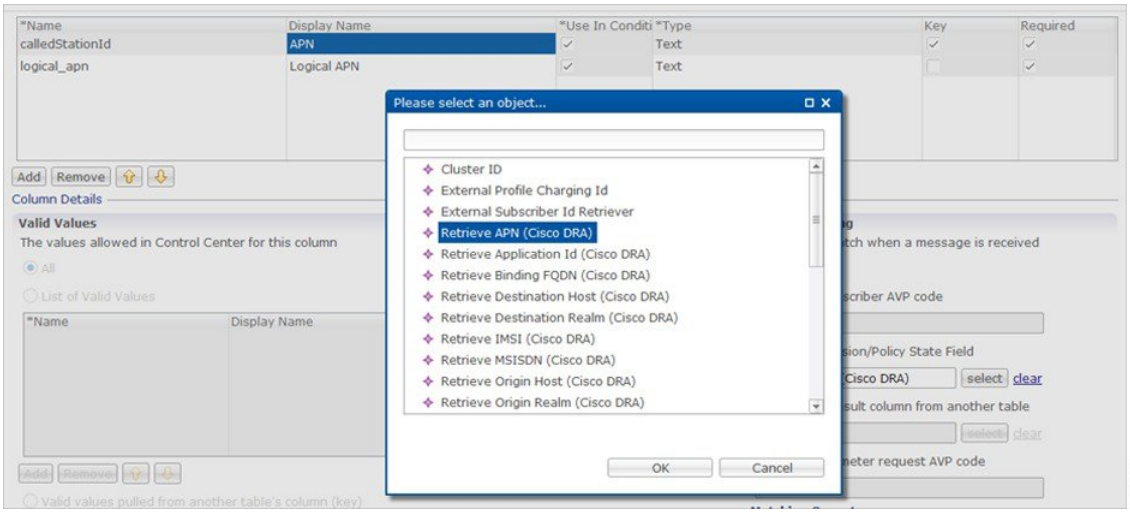
Procedure

Step 1 Select the column name from the **Columns** table and click **select** near **Bind to Session/Policy State Field** to open the **Please select an object...** dialog box.

Note
You can use **Bind to Session/Policy State Field** only for those columns in the **Columns** table where **Key** column has been selected.

Step 2 Select the required object from the dialog box and click **OK**.

Figure 38: Adding AVPs



Step 3 Repeat these steps to add additional AVPs.

Custom Reference Data Tables

Search Table Groups

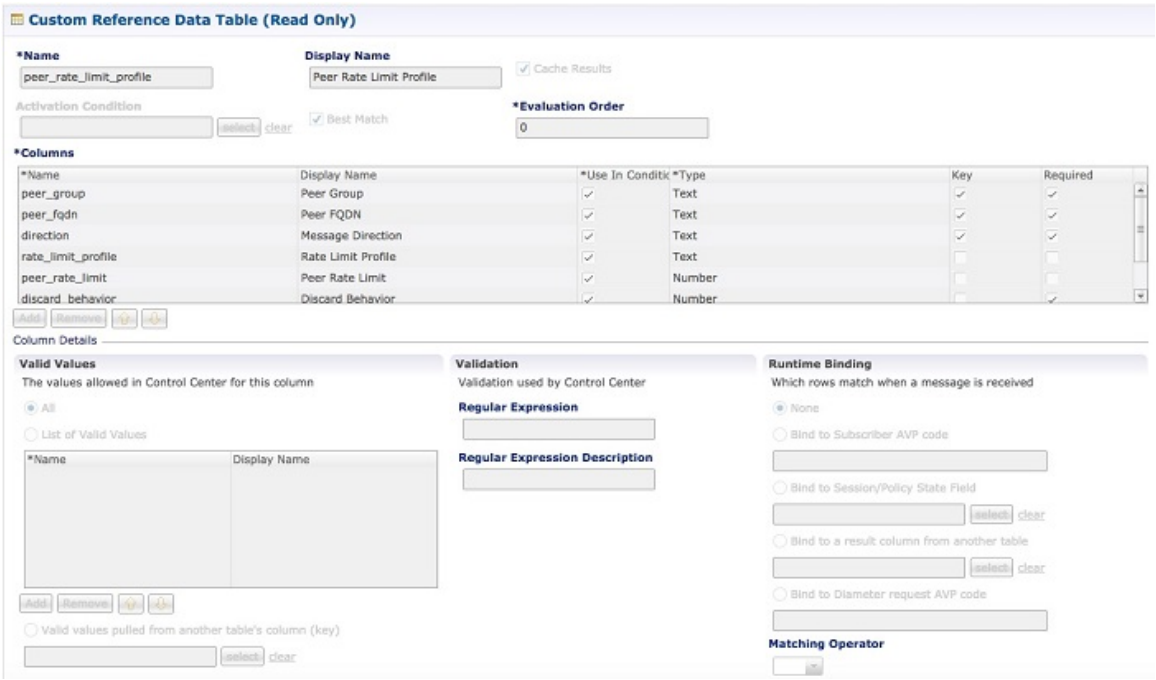
Peer Rate Limit Profile

This is a Search Table Group whose key columns are Peer Group, Peer FQDN or Origin Host in the message and Message Direction.

Using this search table group, the user can configure a maximum rate for each of the configured and defined diameter peers. It also allows the user to configure a maximum rate for each server process.

The peer rate limit is shown below:

Figure 39: Peer Rate Limit - STG



- Peer Group: This is the group of peers classified together using Peer Group and Peer Group Peer values initiating the message.
- Peer FQDN: The origin host of the peer. A specific diameter peer with its Fully Qualified Domain Name can be specified in this field or use wildcards specified by * in this field for any peer or matching peers like hss*.
- Direction: Message direction (Ingress and Egress).

- Ingress: Any diameter messages received by CPS vDRA from diameter peer. The routing decision by CPS vDRA will be taken after the ingress side rate limiting has been applied.
- Egress: Any diameter messages forwarded/routed by CPS vDRA to diameter peer. The egress side rate limiting will be applied after the routing decision has been taken by CPS vDRA.

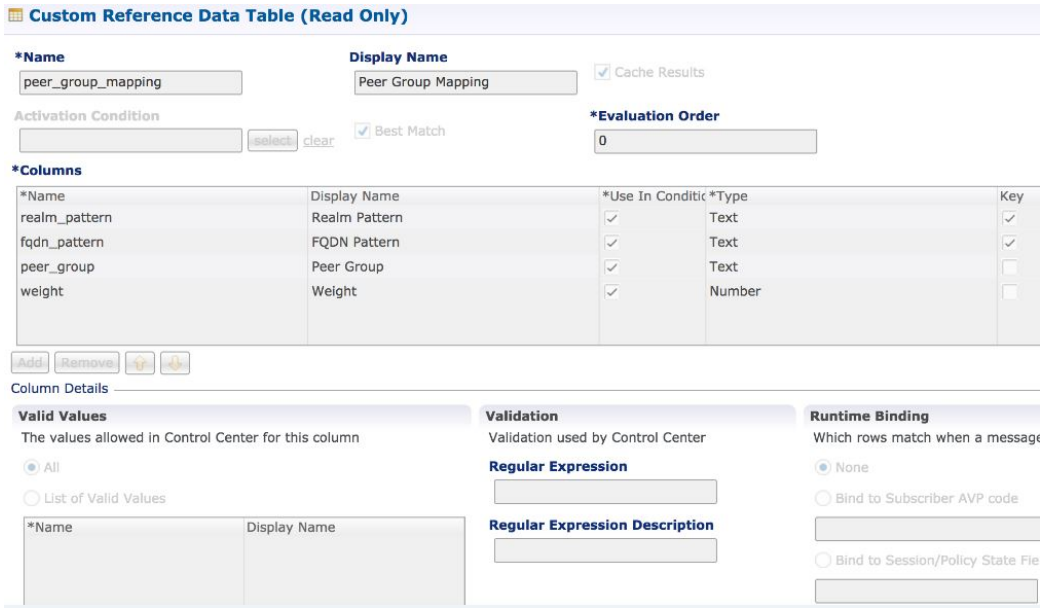
- Peer Rate Limit: This field is to specify the threshold in TPS above which the diameter messages are discarded. This can be left empty if none of the messages are to be dropped or only message level rate limit is to be applied.
- Rate Limit Profile: Profile Name applicable for this Peer Group and Peer, if specified. This profile maps to Rate Limiting at message level. This field enables the rate limit at per message/command code level. See Message Rate Limit Profile, on page 75 for more details.
- Rate Limit Result Code: The result code sent by CPS vDRA for response message towards diameter peer when Discard Behavior is configured as Send Error Answer. In case Discard Behavior is configured as Drop Message, this field is ignored.
- Error String: The string specified in this field is populated by CPS vDRA in AVP Error Message for response message towards diameter peer when Discard Behavior is configured as Send Error Answer. In case Discard Behavior is configured as Drop Message, this field is ignored. This is an optional field when Discard Behavior is configured as Send Error Answer.

Note If both Rate Limit Error Code and Rate Limit Error String are provided along with Rate Limit Action as "Drop Message", the Rate Limit Action takes precedence and the other two fields will be ignored.

For more information, see Peer Rate Limit Profile, on page 68.

Peer Group Mapping

Figure 40: Peer Group Mapping - STG

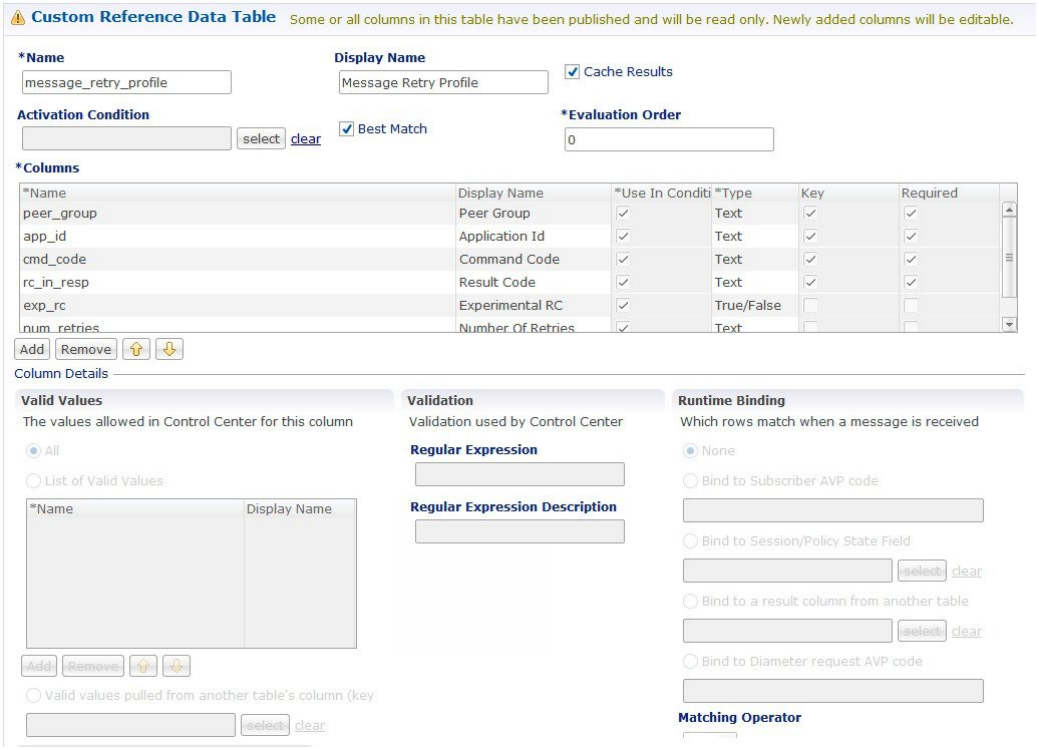


For more information, see Peer Group Mapping, on page 70.

Message Retry Profile

Message retry profile has been added.

Figure 41: Message Retry Profile - STG



- Peer Group: Peer group for which the retry has to be happen.
- Application Id: Application Id of the diameter applications.
- Command Code: Command Code of the message.
- Result Code: Result code received from PCRF for timeout. The value is 7000.
- Experimental RC: Indicates whether result code is experimental or not. This is for future purpose and value in this has no effect on the message retry functionality.
- Number of Retries: Number of retries for the message.

For more information, see Message Retry Profile, on page 73.

Message Mediation Profile

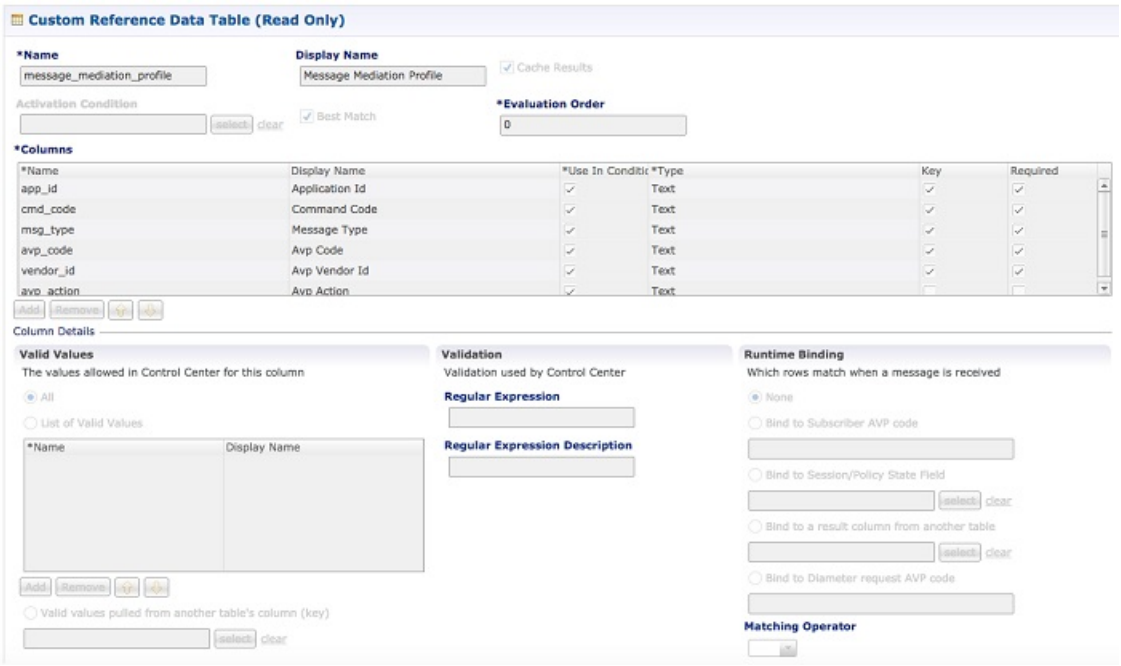
The message mediation profile is used to provide support for mediation of AVPs in Diameter request and answer.

- For Diameter requests, only remove is supported.
- For Diameter answers, the following actions are supported:

- "remove" meaning remove all matching AVPs in the request.
- "copy" meaning copy from the request if no AVPs are present in the answer.
  - If the AVP is present in answer, no action is performed.
- "overwrite" meaning first remove and then copy from the request.
  - Check if the AVP is present in answer, if so remove and add from request.
  - If AVP is not present in answer, copy from request.

A new **Message Mediation Profile** STG has been added:

Figure 42: Message Mediation Profile - STG



- Application Id: Application ID of the Diameter applications.
- Command Code: Command code of the message.
- Message Type : Request/Answer for which the rule has to be applied.
- Avp Code : AVP code of the Diameter message.
- Vendor Id : AVP vendor ID.
- Avp Action : Provides options for copy/remove/overwrite.

Note Application ID, Command Code, AVP Code and Vendor Id are used as key, so no duplicate rows could be defined for this combination and the same AVP action. For example, you cannot define both "remove" and "Copy from request" for the same set of Application ID, Command Code, AVP Code and Vendor Id.

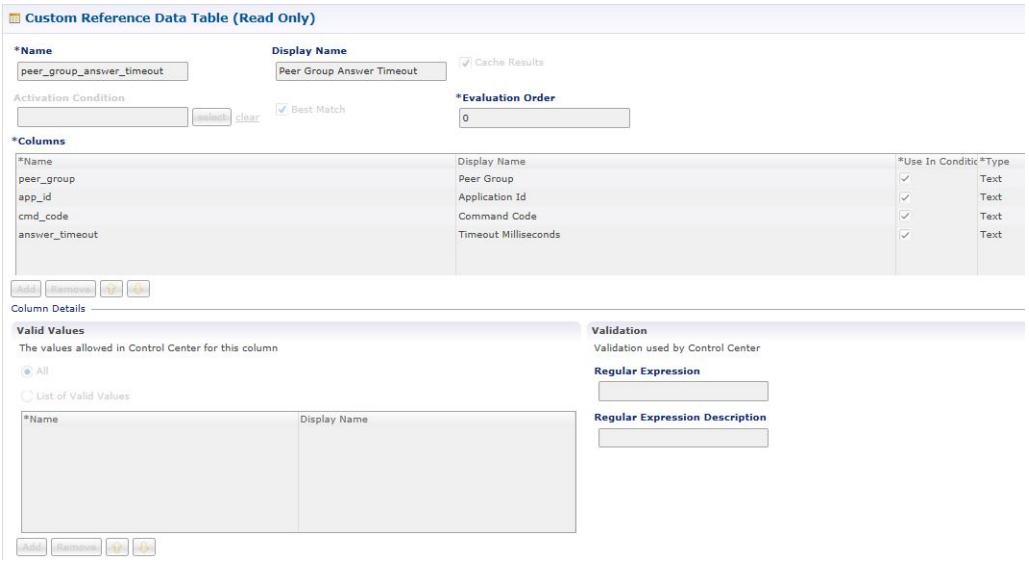
Best Match check box needs to be checked if you want to use the wildcard feature.

For more information, see Message Mediation Profile in Custom Reference Data Tables chapter.

Peer Group Answer Timeout

New search table Peer Group Answer Timeout has been added.

Figure 43: Peer Group Answer Timeout - STG



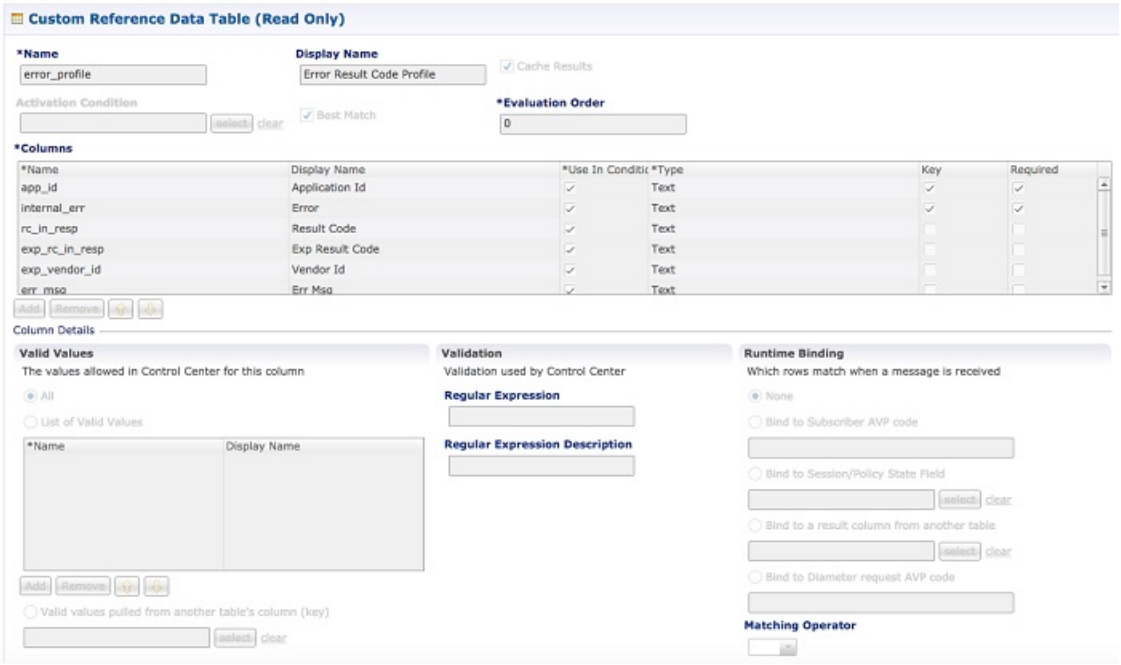
- Application Id: Application Id of the diameter applications.
- Peer Group: Peer group for which the timeout is applied.
- Command code (to enable different timeouts for different Diameter commands)
- Timeout: Timeout in milliseconds.

For more information, see Peer Group Answer Timeout, on page 74.

Error Result Code Profile

Error result code profile can be used to map errors to Result-Code value and an error message string for the Error-Message AVP. It also provides support for configurable error result codes.

Figure 44: Error Result Code Profile - STG



Valid values is the place where all the valid error values can be configured in STG so that they are visible in CRD drop-down.

- ApplicationId: Application ID for which the mapping of Result-Code has to be done.
- Error: Internal error list.
- ResultCode: Result Code to be sent in answer.
- ExpResultCode: Experimental result code to be sent in answer. Vendor-Id will be sent in Answer only for Experimental result-Code.
- ErrMsg: Error message AVP sent in answer.

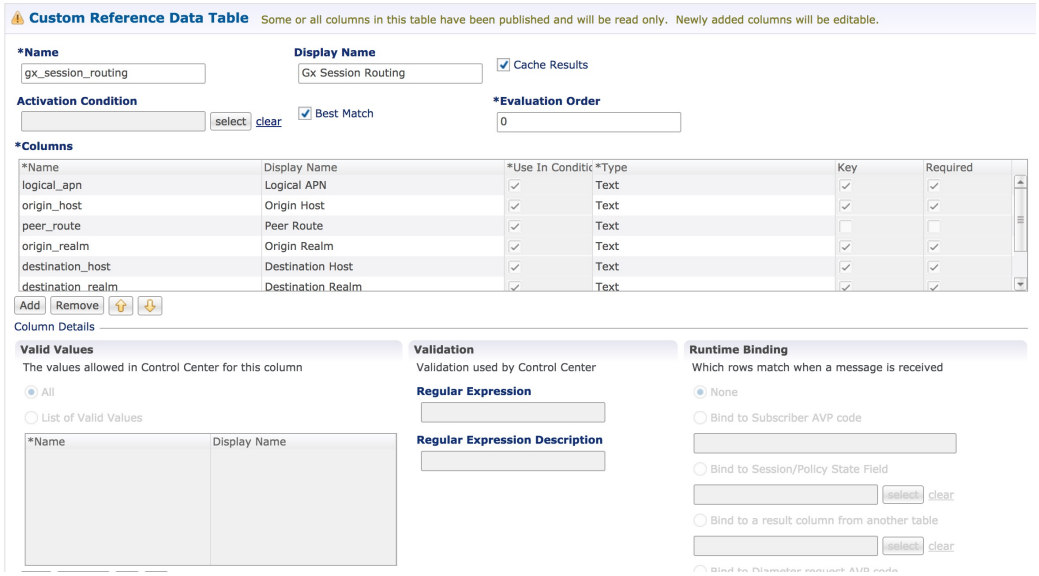
Note Experiment result code will be sent when Result-Code is not configured. If both Result-Code and experimental Result-Code are present, Result-Code would take precedence.

For more information, see Error Result Code Profile, on page 76.

Gx Session Routing

Gx Session Routing table is required for "table driven routing". Here an example for Gx New Session Rules is provided. If table driven routing is required for Rx or Sd, user needs to create similar tables for Sd and Rx as well.

Figure 45: Gx Session Routing



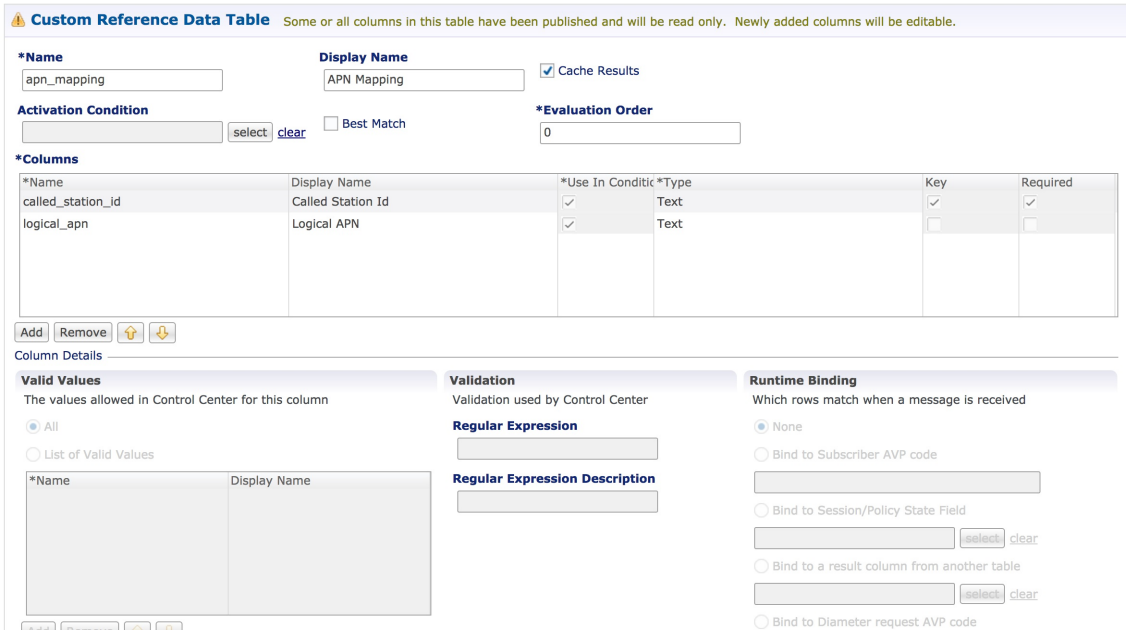
For more information, see Gx New Session Rules, on page 76.

Custom Reference Data Tables

APN Mapping

This table provides information related to APN Mapping. The read-only APN Mapping are shown below:

Figure 46: APN Mapping - CRD Table



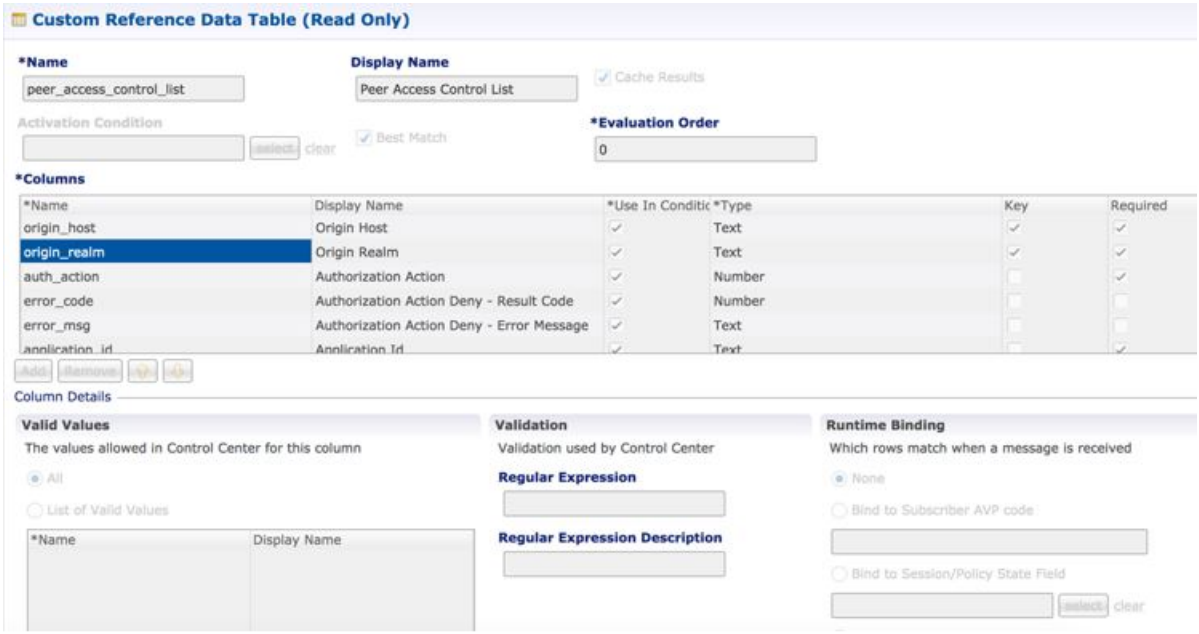
- Called-Station-Id: This is the AVP from which APN is derived. This also is the key column for this table. It is bound to the session or Policy State field as shown in the snapshot.
- Logical_APN: This is the mapped logical name that is used for referencing and processing the message within the system.

Note For sample data configuration, refer the *CPS Control Center Interface Guide for Full Privilege Administrators* for this release.

Peer Access Control List

You can use the Peer Access Control List to specify the list of peers (by realm, FQDN, and applications) that can establish peer connections to vDRA so that unknown peers are not permitted to create Diameter peer connections.

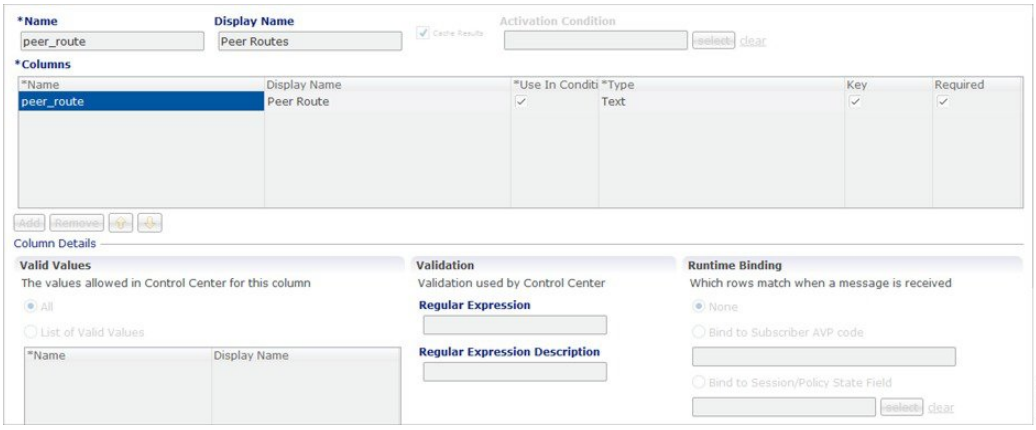
Figure 47: Peer Access Control List



Peer Routes

This tables provides the information related to Peer Routes available in the system. The read-only peer routes are shown below:

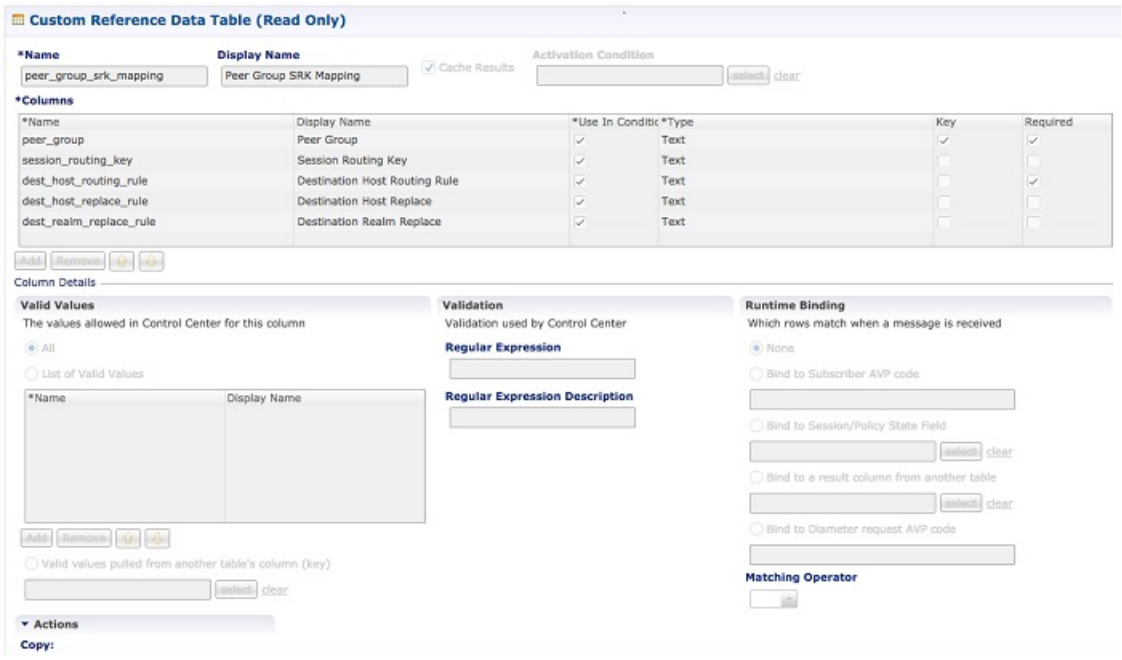
Figure 48: Peer Routes - CRD Table



Peer Group SRK Mapping

This table provides the information related to Peer Groups in the system. The read-only peer groups are shown below:

Figure 49: Peer Group - CRD Table

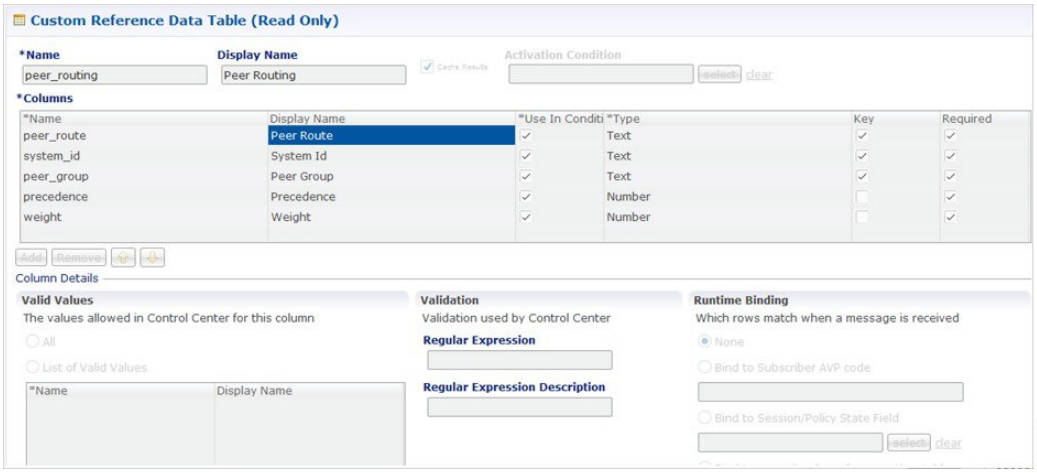


- Peer Group: Name of the peer group.
- Session Routing Key: Routing token for this Peer Group.
- Destination Host Routing Rule: Defines Routing behavior of this group.

Peer Routing

This table provides the information related to peer routing in the system. The read-only peer routings are shown below:

Figure 50: Peer Routing - CRD Table

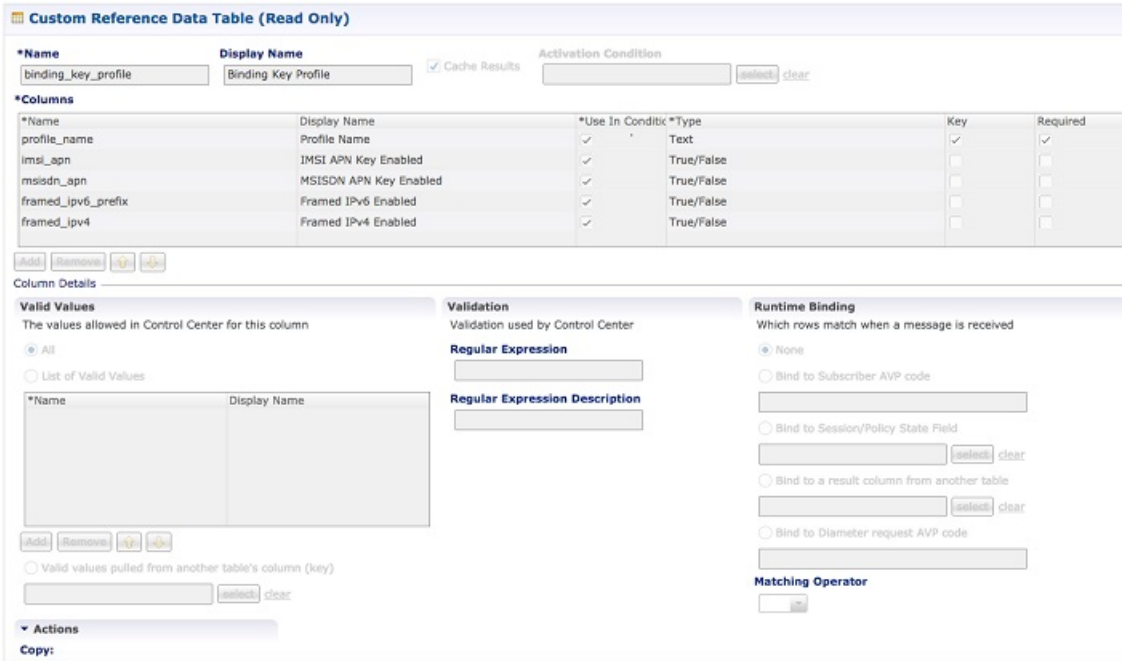


- Peer Route: Identifier of this Peer Route.
- System ID: System Identifier for this VM.
- Peer Group: Identifier of the Peer group on this peer Route.
- Precedence: of the peer group on this Peer Route.
- Weight: Weight of the peer group on this Peer Route.

Binding Key Profile

This table provides the information related to binding key profile in the system. The read-only keys are shown below:

Figure 51: Binding Key Profile - CRD Table



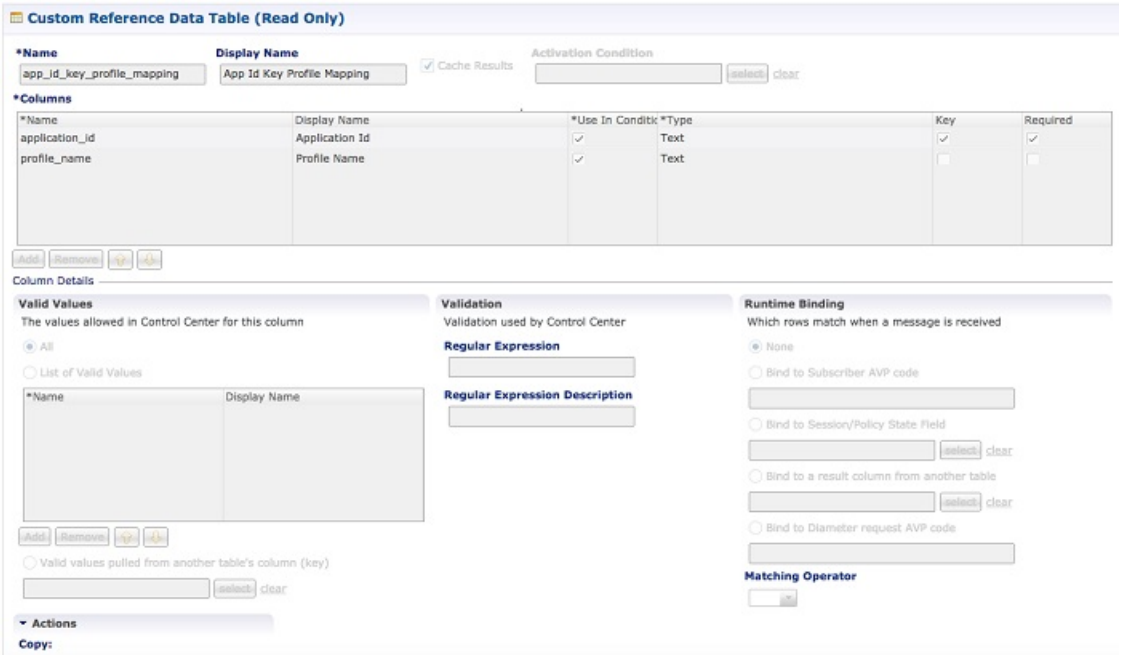
- Profile Name: This is the name given to the Bind profile that is associated with keys that are either enabled and/or disabled.
- MSI APN Key Enabled: Enabling this field would mean that bindings will be stored in IMSI APN collections in bindings database.
- MSISDN APN Key Enabled: Enabling this field would mean that bindings will be stored in MSISDN APN collections in bindings database.
- Framed IPv6 Enabled: Enabling this would mean binding data would be stored in “ipv6bindings” collection.
- Framed IPv4 Enabled: Enabling this would mean binding data getting stored in “ipv4bindings” collection.

Refer to Binding Key Profile, on page 72 for configuration in Control Center.

AppId Key Profile Mapping

This table stores the mapping between Application Identifiers and Bind Key Profile Names. The Application Identifiers are pre-provisioned for two Application Identifiers as Gx and Rx. Similarly, the BindingKeyProfile is also tied to the Profile Name column of the “BindingKeyType_Profile” table:

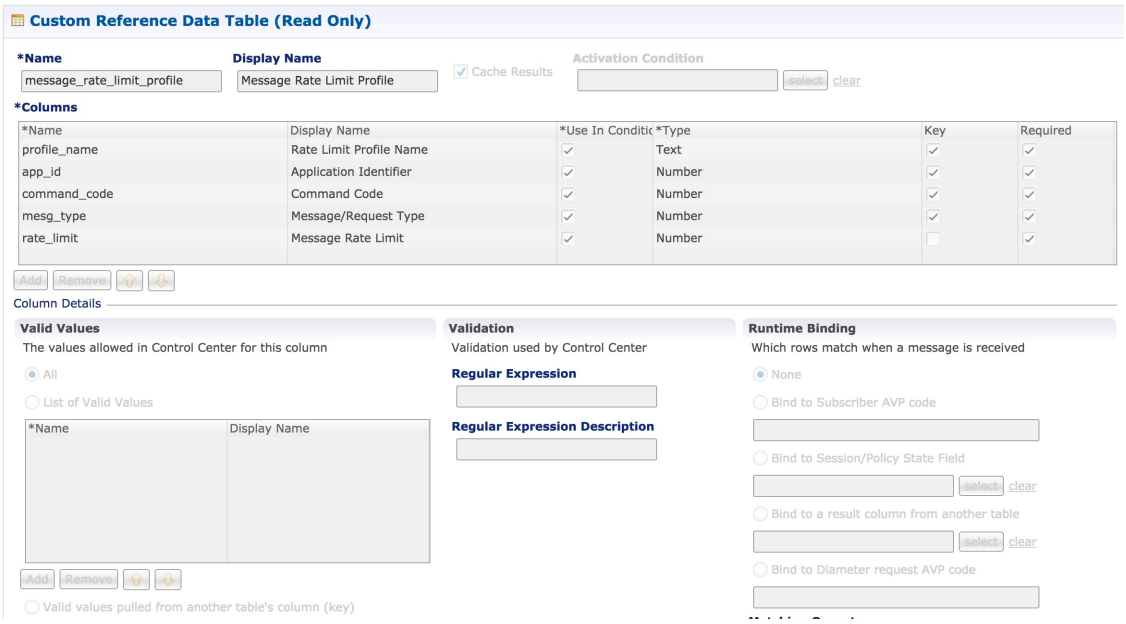
Figure 52: AppId Key Profile Mapping- CRD Table



Message Rate Limit Profile

This table gives a provision to configure Message Rate Limits at a profile level.

Figure 53: Message Rate Limit Profile - CRD Table



- Profile Name: Unique Identifier for a profile.
- Application ID: Application Identifier for this row. 3GPP App Ids only are allowed here.

- Command Code: Command Code of the message that is applicable on the said interface specified by Application Id above.
- Message Type: Initial/Update/Terminate or None for messages that do not have them. The message request type should be same as specified for the command code in Policy Builder under Diameter Application.
- Rate Limit: This field is to specify the threshold in TPS above which the diameter messages are discarded. This value should be more than the Peer Rate Limit in order for message level rate limit to be applied.
- Profile Name: Unique Identifier for a profile.

Refer to Message Rate Limit Profile for configuration in Control Center.

Reserved IMSI

You can configure the Reserved IMSI CRD table to validate a parsed IMSI for SLF routing against a configured list of reserved MCC ranges.

The CRD has two main columns : MCC Start range and MCC End Range. The MCC consists of the first three digits of an IMSI.

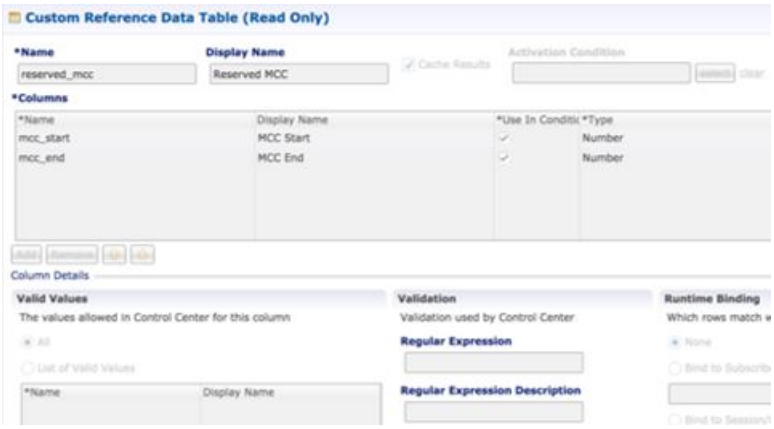
If the IMSI matches a reserved IMSI, the value is ignored for SLF routing.

You can provide support up to ten distinct (non-overlapping) MCC ranges as Reserved IMSIs.

The DRA/SLF ignores AVPs that contain such IMSIs, and continues searching other AVPs in the Diameter request, for a valid address to be used for address resolution.

The following image shows a sample Reserved IMSI configuration:

Figure 54: Reserved IMSI

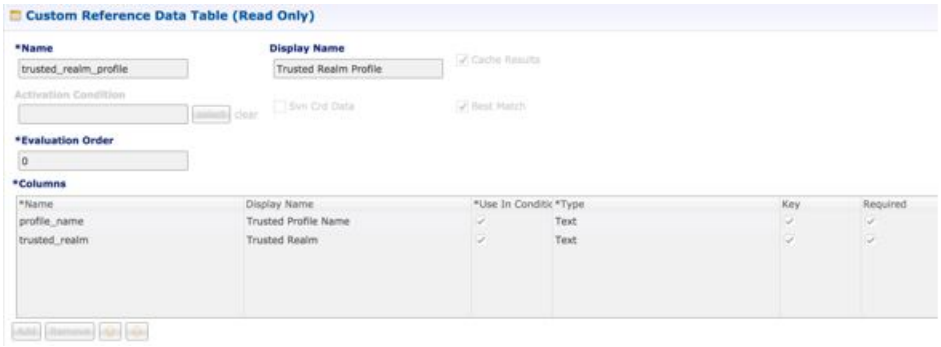


Trusted Realm Profile

Trusted Realm Profile is used for topology hiding. The CRD includes the following columns:

- Trusted Profile Name: Profile Name having a trusted realm mapped to it.
- Trusted Realm: Realm for which Topology Hiding is not required.

Figure 55: Trusted Realm Profile



Protected Realm Trusted Profile Mapping

Protected Realm Trusted Profile Mapping is used for topology hiding. The CRD includes the following columns:

- Protected Realm: Realm that is protected (topology hiding is required).
- Profile Name: Profile having realms that are trusted for this protected realm and that do not require topology hiding.

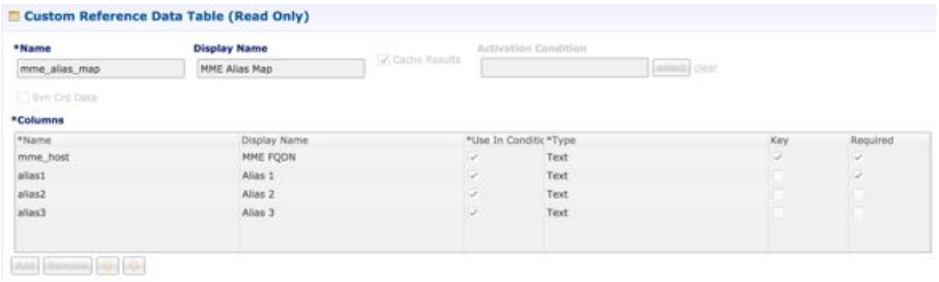
Figure 56: Protected Realm Trusted Profile Mapping

MME Alias Map

MME Alias Map is used for topology hiding. The CRD includes the following columns:

- MME FQDN: FQDN of MME that requires topology hiding.
- Alias1: Mandatory. An alias identity used for the protected host that belongs to an MME in the network.
- Alias 2: Optional. Alternate Alias that can be used for Topology Hiding for the given MME FQDN.
- Alias 3: Optional. Alternate Alias that can be used for Topology Hiding for the given MME FQDN.

Figure 57: MME Alias Map



HSS Aliases

HSS Aliases is used for topology hiding. The CRD includes the following columns:

- HSS Alias FQDN: Alias FQDN used to replace a protected HSS FQDN.
- Shared Alias: Boolean variable used to indicate whether the Alias FQDN is shared across multiple HSS servers or not.

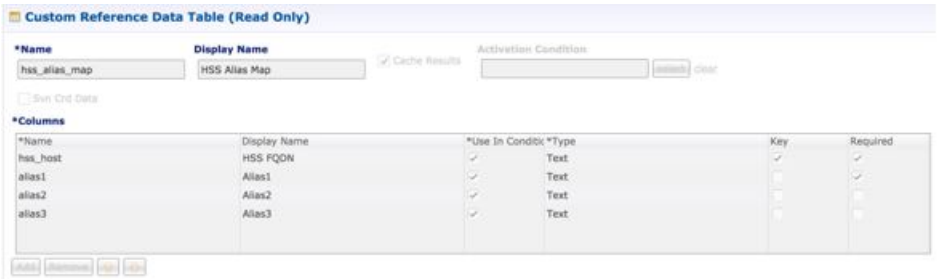
Figure 58: HSS Aliases

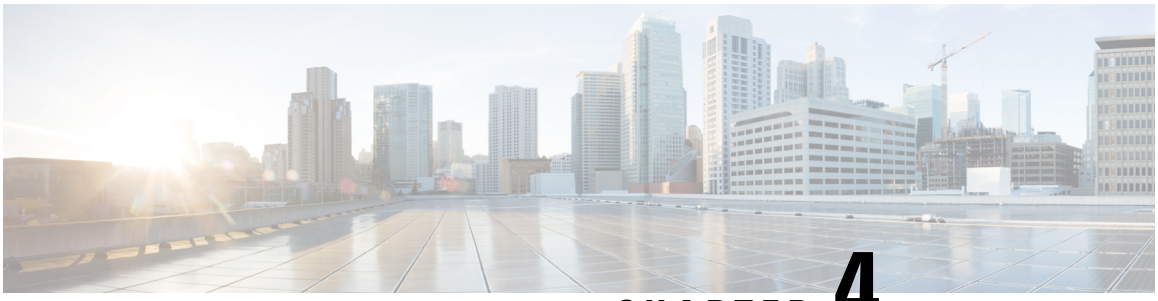
HSS Alias Map

HSS Alias Map is used for topology hiding. The CRD includes the following columns:

- HSS FQDN: FQDN of HSS peer.
- Alias1: Required field which is derived from HSS Alias CRD.
- Alias2: Optional. Alias for the HSS FQDN.
- Alias3: Optional. Alias for the HSS FQDN.

Figure 59: HSS Alias Map





CHAPTER 4

Custom Reference Data Configuration

Logical APN List

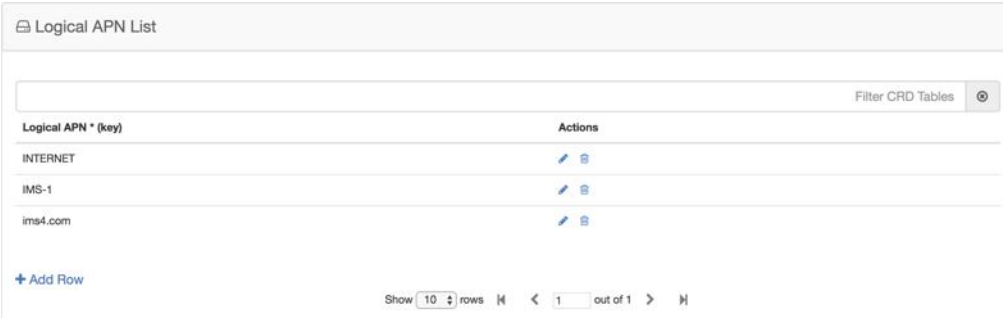
The logical APN feature allows multiple users to access different physical target networks through a shared APN access point. The logical APN feature reduces the amount of APN provisioning required by consolidating access all real APNs through a single virtual APN. Therefore, only the virtual APN needs to be provisioned at Control Centre, instead of each of the real APNs to be reached.

For details on System ID, refer to Peer Routing, on page 71.

For details on Peer Group, refer to Peer Group Mapping and Peer Group SRK Mapping.

An example configuration is shown below:

Figure 60: Logical APN List



APN Mapping Table

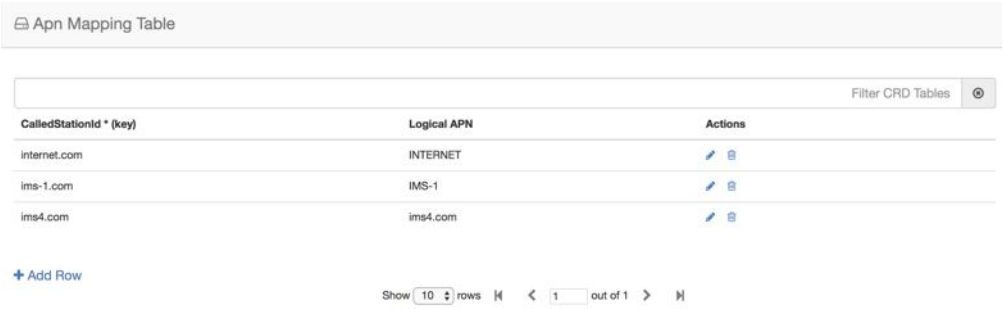
The APN consists of two parts which are as follows:

- The APN Network Identifier. This part of the APN is mandatory.
- The APN Operator Identifier. This part of the APN is optional.

The actual APN of any interface is filled-in with Called-Station-Id AVP. This table keeps a mapping of actual APNs and logical APNs configured in the logical APN list.

The following is an example configuration:

Figure 61: APN Mapping Table



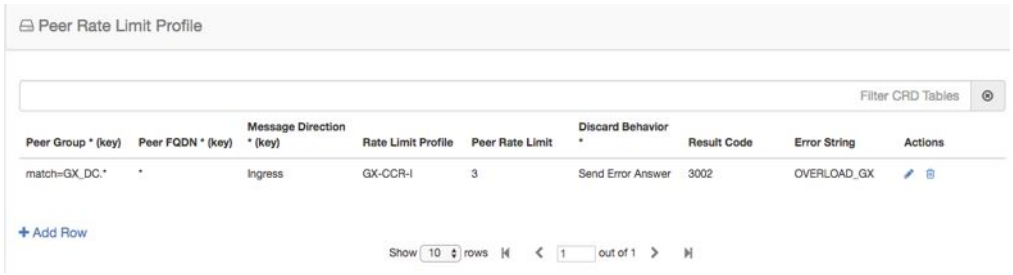
The Called-Station-Id input is case insensitive where it stores all the values in lower case. It converts the upper case entry to a lower case value and checks for a duplicate entry. If the input APN contains any duplicate value, it rejects the value with an error message.

For example, if the input value is `IMS.COM`, it stores the value as `ims.com`.

Peer Rate Limit Profile

CPS vDRA can rate limit traffic coming from and going towards a particular peer. This can work for both Ingress and Egress traffic. User needs to define the peer group, FQDN, traffic direction and the CPS vDRA behavior, whether to silently drop or send error message. User can also define the error code and the error message when error responses need to be sent back.

Figure 62: Peer Rate Limit Profile



Peer Access Control List

You can use the Peer Access Control List to specify the list of peers (by realm, FQDN, application ID, or Source-IP) that can establish peer connections to vDRA.

Peers that are not listed with realm or host in the CRD are allowed to establish peer connections by default.

Specify the following parameters:

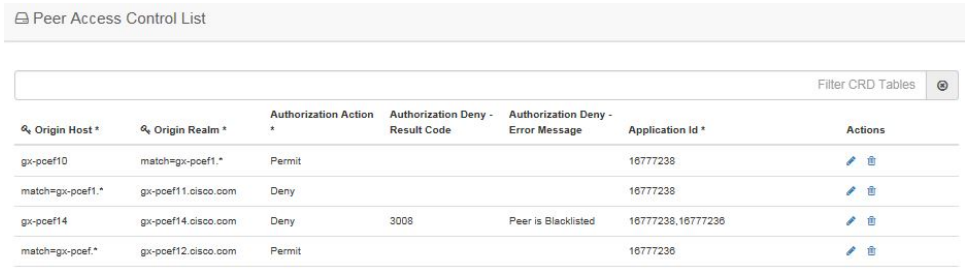
The key fields are Origin Host and Origin realm, hence it is possible to have only one row for each unique pair.

- Origin Host - Diameter identity or FQDN(host) of the client either in full or as a regular expression
- Origin Realm - Diameter Identity or realm of the client either in full or as a regular expression
- Authorization Action: Specifies whether the incoming client connection is allowed or denied.
- Authorization Deny - Result Code: Configurable result code. If not configured, the default value of 3010 (Unknown Application) or 3007 (Unsupported Application) is sent. Applicable only when the Authorization action is set to “Deny”
- Authorization Deny - Error Message: Configurable Message. If not configured default values are Unknown Peer or Unsupported Application.

  Applicable only when the Authorization action is set to “Deny”
- Application ID: single, comma-separated, or regular expression.

If the peer connection is rejected due to mismatch of Applications, customized result-code / error messages are not applicable in this case.

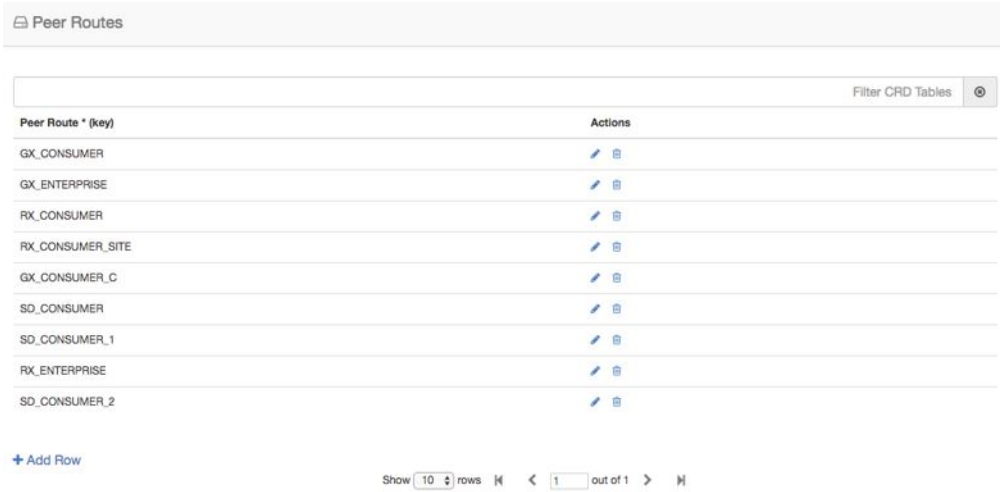
Figure 63: Peer Access Control List



Peer Routes

Request forwarding is done using Peer Routes to discover peers. These routes are different for different interfaces. There can be multiple peer routes for a particular interface.

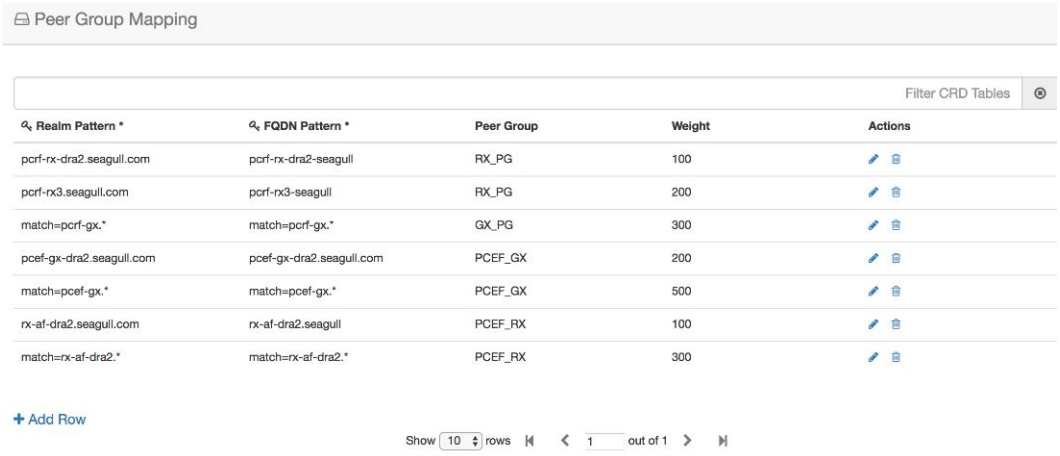
Figure 64: Peer Routes



Peer Group Mapping

One or more peers are combined into single peer group based on their realms patterns and FQDN patterns. Peer groups have respective peer routes.

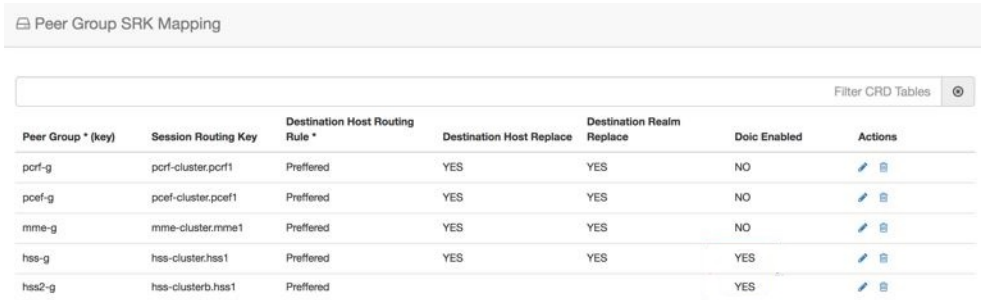
Figure 65: Peer Group Mapping



Peer Group SRK Mapping

All the peer groups consisting of one or more peers are listed in this table. Also various features like Session Key Routing or Destination Host Routing can be configured as Only, Never, Preferred depending upon the need. Use the DOIC Enabled column (YES/NO) to enable or disable Diameter Overload Indication Conveyance (DOIC). This option is used to throttle or divert Diameter requests towards PCRF, HSS, AAA, and OCS servers based on reporting of overloaded conditions.

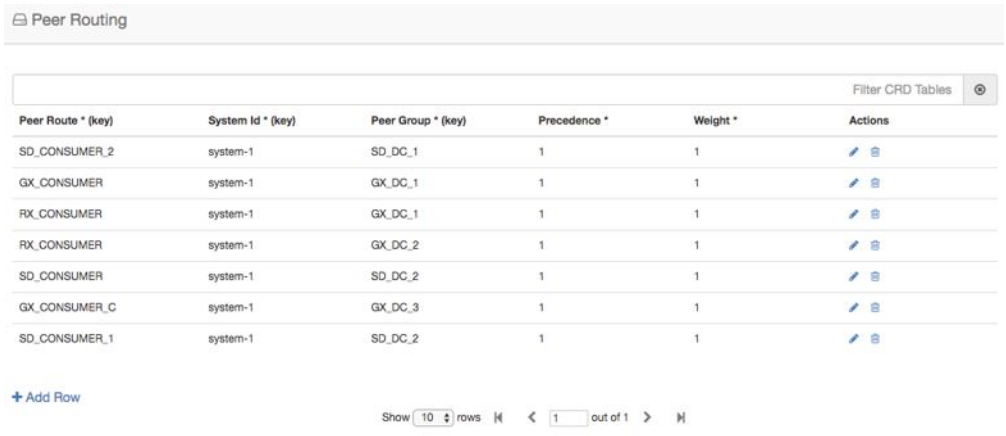
Figure 66: Peer Group SRK Mapping



Peer Routing

This table consists of a mapping of Peer Groups to Peer Routes on a particular CPS vDRA. It also has precedence and weight columns which play a vital role in load balancing behavior of CPS vDRA.

Figure 67: Peer Routing



Binding Key Profile

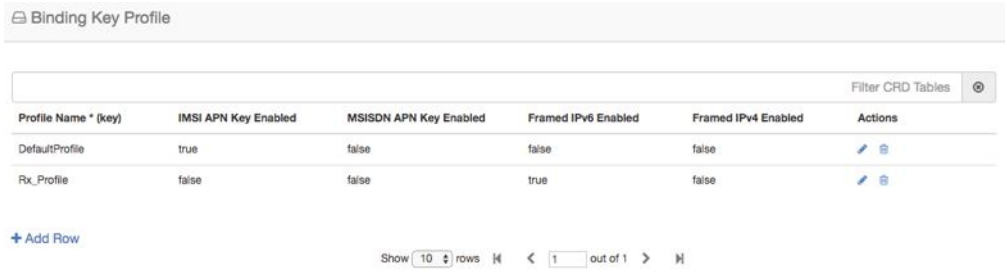
Important For routing to work in DRA, user must configure **AppId Key Profile Mapping** and **Binding Key Profile** tables.

The available fields are Boolean fields and can be edited by selecting the check boxes.

Note It is expected a minimum of one row to be configured with the value “DefaultProfile”. This will be used in case there is nothing configured for an application Id. For this “DefaultProfile”, “imsiAPN” and “FramedIPv6Prefix” should be enabled.

Note The field **MSISDN APN Key Enabled** is a place holder only. Modifying this field will not have an effect on the application behavior.

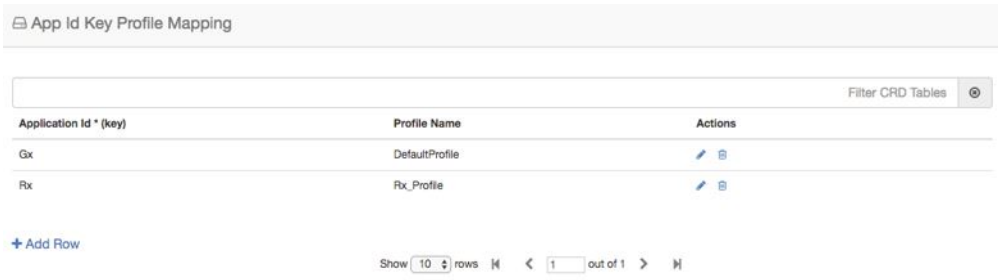
Figure 68: Binding Key Profile



AppId Key Profile Mapping

Important For routing to work in CPS vDRA, you must configure **AppId Key Profile Mapping** and **Binding Key Profile** tables.

Figure 69: AppId Key Profile Mapping



The Binding Key Profile column is tied to the Profile Name column from the previous CRD and takes the available Profile Name in the system.

There are two application Identifiers that have been provisioned in the system which are Gx and Rx and can be tied to the same or different Bind Key Profile as the case may be.

Message Class Profile

To determine the abatement action from the DOIC Profile table (for throttling or diverting Diameter requests), you require a Message class. You can query the Message class from the Message Class Profile table.

The Message Class Profile table takes inputs such as Ingress Peer Group, Application Id, Command Code, Message/Request Type and provides the Condition Profile and Message Class. Message Class can be one of P0, P1, P2, P3, P4.

Figure 70: Message Class Profile

Message Retry Profile

CPS vDRA supports configurable retries, so that the specific behavior of CPS vDRA in congestion scenarios can be configured.

Configurable retry mechanism (i.e., number of retries) per:

- Application ID
- Peer Group
- Answer Timeout error occurred
- Error Result Code of Response

This should be in the form of a CRD and applied to a peer group. The user can use the SRK peers to select an alternate peer.

If all SRK peers fail, the user should use one alternate CPS vDRA if it connects to the SRK. If the SRK matches exactly, CPS vDRA would look for the second label match of SRK like clusterb.dc1 and clusterc.dc1 and retry the message to other peer group.

Figure 71: Message Retry Profile - Control Center

Message Retry Profile

Filter CRD Tables

Peer Group * (key)	Application Id * (key)	Command Code * (key)	Result Code * (key)	Experimental RC	Number Of Retries	Actions
GX_DC_1	16777238	272	3003	false	6	
GX_DC_1	16777238	272	3004	false	6	
GX_DC_2	16777238	272	3003	false	6	
GX_DC_2	16777238	272	3004	false	6	
match=SD_DC.*	16777303	8388637	3004	false	6	
RX_DC_1	16777236	258	7000	false	6	
GX_DC_2	16777238	272	7000	false	6	

+ Add Row

Show 10 rows 1 out of 1

Wild card match is supported for Peer Group, Application Id, Command Code, Result Code columns. For example, 300.* supports all RC starting with 300.

- * is supported to allow all RC.
- * is supported for all peer groups.
- Match = GX_DC_.* is supported for groups starting with GX_DC

RC = 7000 is interpreted as retry for timeout.

Experimental result code is for future purposes and value in that column has no effect on retry processing.

Note **Best Match** check box needs to be checked in Policy Builder if you want to use the wildcard feature.

Refer to Message Retry Profile, on page 52 for configuration in **Search Table Group**.

Peer Group Answer Timeout

CPS vDRA support for the following use cases:

1. Configurable answer timeout for initial try and subsequent retries for the following parameters:
   - Application ID
   - Peer Group (to which request is sent)
   - Command code (to enable different timeouts for different Diameter commands)
   - Timeout value (in milliseconds)

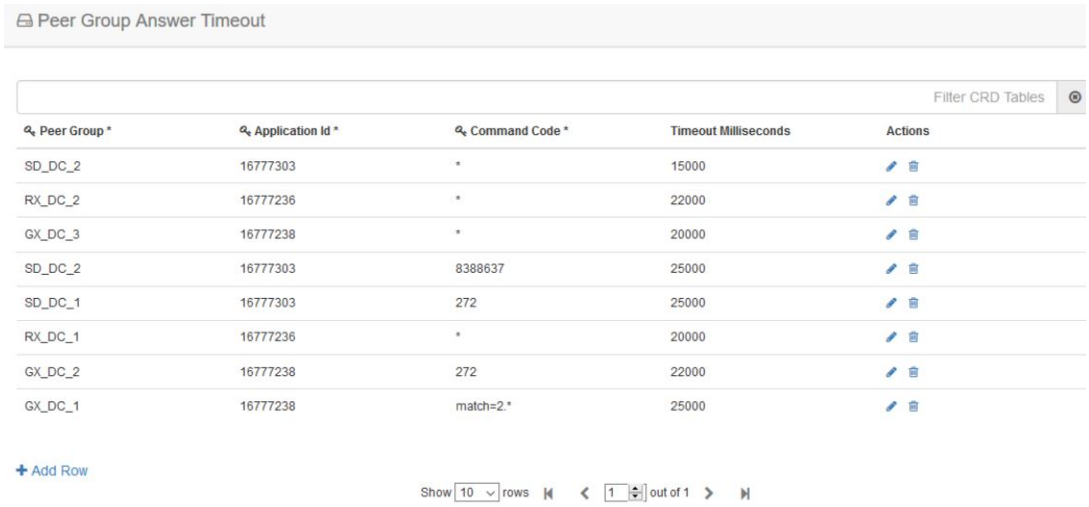
2. Default value if unspecified is 1700 milliseconds.

Peer group answer timeout is applicable for every message routed using:

- Destination host routing
- SRK routing
- Table driven routing

Sample peer group answer timeout is shown below:

Figure 72: Peer Group Answer Timeout



Wild card match is supported for application_id, peer_group, command code. * indicates all application_id, peer_group.

The following rules have been applied for answer timeout:

- Default timeout for any message routed from CPS vDRA is 1700 ms.
- In case of retry, if an alternate group is chosen for routing, corresponding timeout for the peer group is applied.

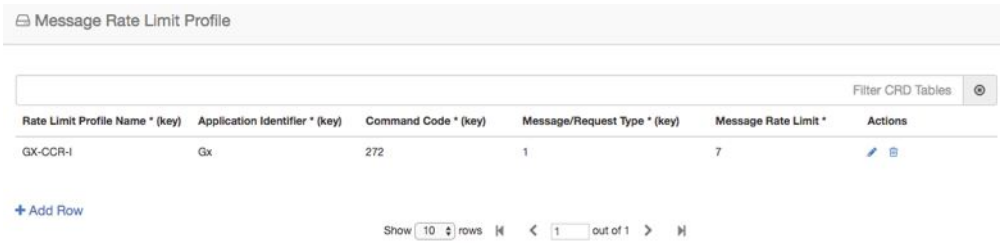
For Policy Builder related configuration, refer to Peer Group Answer Timeout, on page 54.

Message Rate Limit Profile

Further to peer level rate limit, CPS vDRA provides the granularity of limiting diameter traffic at message level for each peer. Message level rate limit always works in conjunction with peer level rate limit and is an additional control in peer level rate limit configuration. Since message level rate limit works in conjunction with peer level rate limit, all the fields specified for peer level rate limit are applicable to message level rate limit.

Message Rate Limit Profile table is used to get the condition for such rate limiting. User can define the type of message, command code and the application for which the limiting has to be implemented.

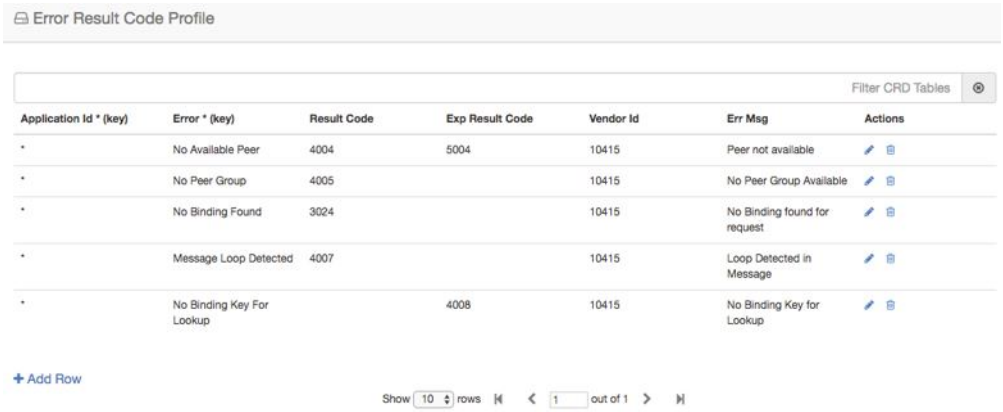
Figure 73: Message Rate Limit Profile



Error Result Code Profile

Sample CRD data looks like this:

Figure 74: Error Result Code Profile



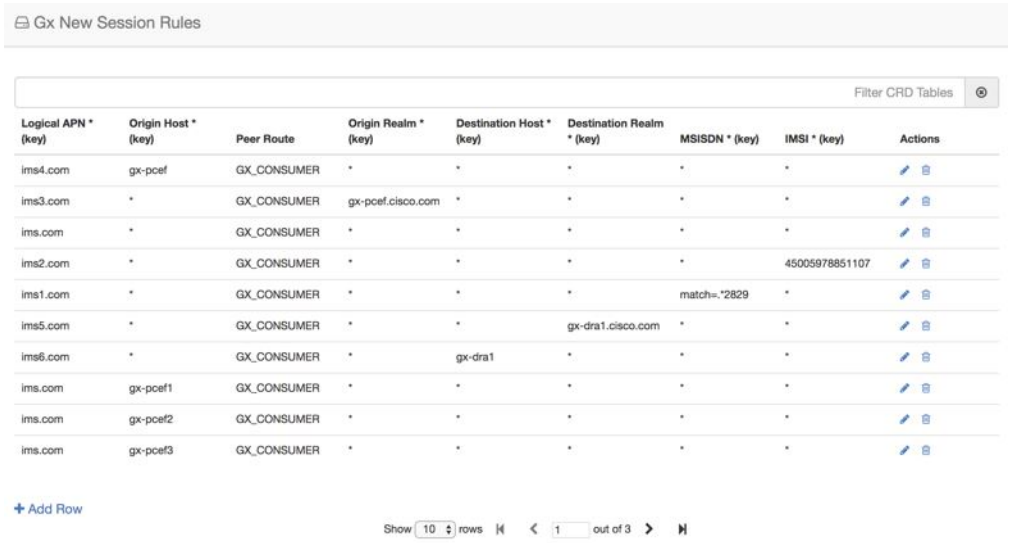
- For any CPS vDRA error or message timeout, CPS vDRAhas the ability to map the error to a Result-Code value and an error message string for the Error-Message AVP.
- Errors include things like "binding not found", "message timeout", "no peer connections".
- The Result Code value is sent in the Result-Code AVP in the response.
- The error message string is sent in the Error-Message AVP in the response.
- When both Result Code and Exp Result Code are configured in this table, Result Code will take precedence. In case Result Code is not configured in this table, Exp Result Code will be sent with Vendor-ID.

Gx New Session Rules

Gx New Session Rules table is used by CPS vDRA when performing Table Driven routing. CPS vDRA could derive the "Peer Route" from this table, when the incoming message has no destination host to be routed to. From peer route, CPS vDRA derives further route where the request could be sent. This table supports both wildcard and exact match for the various parameters. The "Peer Route" used in this table should be defined

in "Peer Routes" table. Here an example for Gx New Session Rules is provided. Similar tables can be created for Rx or Sd.

Figure 75: Gx New Session Rules



Range Based Routing

CPS vDRA provides range-based routing based on MSISDN and IMSI values so that Diameter requests are routed to the correct HSS or AAA server. Range-based routing occurs if the destination-host routing, binding-based routing and SLF-based routing fails.

- vDRA checks whether the primary lookup type is IMSI or MSISDN and also checks whether the IMSI/MSISDN value present in the request matches against the range configured in CRD.
- The primary lookup type is evaluated first and if it fails, the secondary lookup type is evaluated.
- If primary lookup type evaluation fails and if the secondary lookup type is not configured, the request is routed with table-driven routing (if configured).
- If both the primary lookup type credential and the secondary lookup type evaluation fail, the request is rejected or routed with table driven routing (if configured).

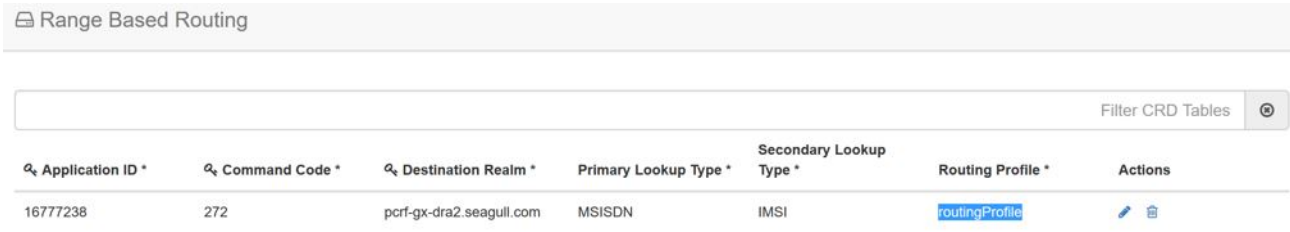
vDRA matches the request against the Range Based Routing table and based on the result of the credential match, SRK routing is initiated.

Table 15: Range Based Routing

Field	Description	Value
Application Id (input)	The diameter application of the message received	Integer value of the application id
Command Code (input)	The message command code	Integer value of the command code

Field	Description	Value
Destination Realm (input)	The destination realm in the message	String value of destination realm
Primary Lookup Type (input)	Primary lookup type for range based routing	IMSI or MSISDN
Secondary Lookup Type (input)	Secondary lookup type for range based routing	NONE or IMSI or MSISDN
Routing Profile (output)	Routing profile	Any string value. (Should match the routing profile in either or both the IMSI and MSISDN range CRD for a successful match).

Figure 76: Range Based Routing



IMSI Range

The IMSI Range is used in range-based routing to configure the range of IMSI values.

Table 16: IMSI Range

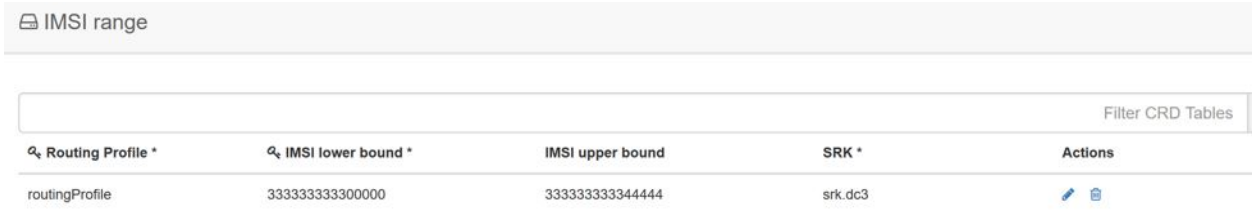
Field	Description	Value
Routing Profile (input)	The routing profile name	Any string value
IMSI lower bound (input)	The lower bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, use the syntax: `match=<regex>`
IMSI upper bound (input)	The upper bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, leave it blank.
SRK (output)	The SRK key	Any string value

Examples:

- For configuring numeric range between 9840510345 to 984059999: Lower bound: 9840510345, Upper bound: 9840598823

- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: match=98405[0-9]*, Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840501333 to 9840502999: Lower bound: match=984050(1|2)[3-9]* , Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: match=98405(([2-7][0-9]*)|(8[0-8][0-4][0-5][0-6])|(1[0-9][2-9][3-9][4-9])), Upper bound : <leave it empty>

Figure 77: IMSI Range



MSISDN Range

The MSISDN Range is used in range-based routing to configure the range of MSISDN values.

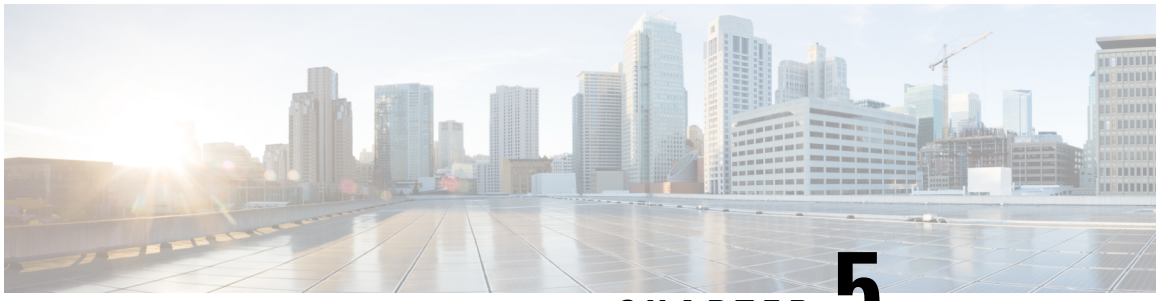
Table 17: MSISDN Range

Field	Description	Value
Routing Profile (input)	The routing profile name	Any string value
IMSI lower bound (input)	The lower bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, use the syntax: `match=<regex>`
IMSI upper bound (input)	The upper bound for the IMSI value	For a numeric range, enter the IMSI value. For a regex, leave it blank.
SRK (output)	The SRK key	Any string value

Examples:

- For configuring numeric range between 9840510345 to 984059999: Lower bound: 9840510345, Upper bound: 9840598823
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: match=98405[0-9]*, Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840501333 to 9840502999: Lower bound: match=984050(1|2)[3-9]* , Upper bound : <leave it empty>
- For configuring regex for numbers in range 9840500000 to 9840599999: Lower bound: match=98405(([2-7][0-9]*)|(8[0-8][0-4][0-5][0-6])|(1[0-9][2-9][3-9][4-9])), Upper bound : <leave it empty>

Figure 78: MSISDN Range



CHAPTER 5

DRA Distributor Configuration

DRA Distributor Configuration Overview

DRA distributor configuration includes the following:

- Configuring the dra-distributor VMs.
- Adding VIPs to the dra-directors.
- Suppressing IPv4 ARP/IPv6 neighbor discovery for the VIPs on the dra-director.
- Adding static routes to clients (PGW, PCRF, and so on) on the dra-director.

Configuring DRA Distributor

Configuring DRA Distributor VM is performed using the ConfD CLI interface.

CLI Configuration

network dra-distributor

Add a dra-distributor cluster

Syntax

```
network dra-distributor <client> <range>
```

The following table describers the DRA Distributor configuration parameters:

Table 18: DRA Distributor Configuration Parameters

Parameter	Description
client	Name of cluster to be configured. Value range is from 1 - 8 characters

Parameter	Description
sync-id	Unique ID per cluster. VMs with the same sync-id synchronize connection data. All VMs in the named dra-distributor synchronize their connection data in case of VM failure. Value range is from 0 - 255
sync-interface	Interface used to send multicast connection sync data. Typically an interface on the Internal network. Example: ens192
global-tracking-service	Container to track for health check of dra-director VMs for all services. Default value is diameter-endpoint
host-ip	IP address of member VM. Value: Any IP address that exists on the VM. Typically, internal IP address.
global-priority	Global priority for all services of the host on which the service must run. Can be overridden in an individual service configuration. Priority range is from 1 to 255. Larger values have higher priority than lower values. Example: 10 has a higher priority than 5.
service-name	Unique name for peer service.
virtual-router-id	Virtual router ID is the identity for a virtual router for hosts that are managed for the virtual IP of the service. Value range is from 0 - 255. For more details, refer to VRRP (Virtual Router Redundancy Protocol) RFC 3768 and keepalive documentation.
tracking-service	Container to track for health check of dra-director VM. Overrides global-tracking-service. Default value is global-tracking-service.
preempt-delay	Preempt delay is delay in seconds before a VIP switches from backup to master. Default value is 30 seconds. Value range is from 1 - 1000.
interface	Interface of the host where the virtual IP is installed as secondary address when active.
service-ip	Virtual IP address of service.
service-port	TCP port of service.

Parameter	Description
service-host-ip	IP address of VM. Used to override global priority.
service-priority	Overrides global-priority. This allows a VIP to run on VM1 and another VIP to run on VM2. Example: Gx VIP on VM1 and Rx VIP on VM2.
preempt	Enable or disable VIP preemption for a single VIP. Default value is true. Value: true, false
real-service-ip	IP address of a dra-director supporting the service.
weight	Relative weight of real-server used by weighted least connection scheduling algorithm. Value range is from 0 - 255. Default value is 1. A value of 0 disables new connections to this real-server.

Sample Configuration

```
network dra-distributor client
 sync-id          1
 sync-interface   ens192
 tracking-service diameter-endpoint
 preempt-delay 5
 host 192.169.21.20
  priority 10
 !
 host 192.169.21.21
  priority 5
 !
 service Gx
  virtual-router-id 60
  interface         ens224
  service-ip        192.169.22.50
  service-port      3868
  real-server 192.169.22.13
   weight 100
  !
  real-server 192.169.22.14
  !
 !
 service Rx
  virtual-router-id 61
  interface         ens224
  service-ip        192.169.25.80
  service-port      3868
  host 192.169.21.20
   priority 4
  !
  host 192.169.21.21
   priority 9
  !
  real-server 192.169.25.13
```

```
  !
  real-server 192.169.25.14
  !
 !
!
network dra-distributor server
 sync-id          2
 sync-interface   ens192
 tracking-service diameter-endpoint
 preempt-delay 5
 host 192.169.21.30
  priority 10
 !
 host 192.169.21.31
  priority 5
 !
 service Gx
  virtual-router-id 70
  interface         ens224
  service-ip        192.169.23.70
  service-port      3868
  real-server 192.169.23.13
   weight 100
  !
  real-server 192.169.23.14
  !
 !
 service Rx
  virtual-router-id 71
  interface         ens256
  service-ip        192.169.28.70
  service-port      3668
  real-server 192.169.28.13
  !
  real-server 192.169.28.14
  !
 !
!
```

Configuration Status Check

To check the distributor status use `show dra-distributor` command.

Example:

```
admin@orchestrator[master-0]# show dra-distributor ?
Possible completions:
  daemon  list  rate  stats
admin@orchestrator[master-0]# show dra-distributor
```

To verify distributor VIPs use `show network ips` command.

To verify director VIP/netfilter rules use the following commands:

```
ip -4 addr show (Confirm VIP address exists)
ip -6 addr show (Confirm VIP address exists)
sudo arptables --list (Confirm rule exist for each vip)
sudo ip6tables --list (Confirm ipv6 neighbor-solicitation/advertisement filters for each
vip)
```