CPS vDRA Configuration Guide, Release 25.2.0

First Published: 2025-10-30

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

CONTENTS

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 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.

Note The PATS/ATS, ANDSF, and MOG products have reached end of life and are not supported in this release. Any references to these products (specific or implied), their components or functions in this document are coincidental and are not supported. Full details on the end of life for these products are available at: https://www.cisco.com/c/en/us/products/wireless/policy-suite-mobile/eos-eol-notice-listing.html.

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.

Conventions	Indication
[ ]	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

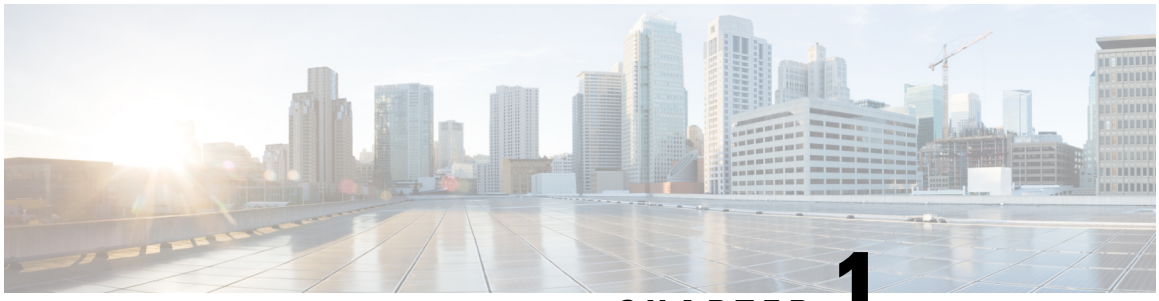
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- 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:

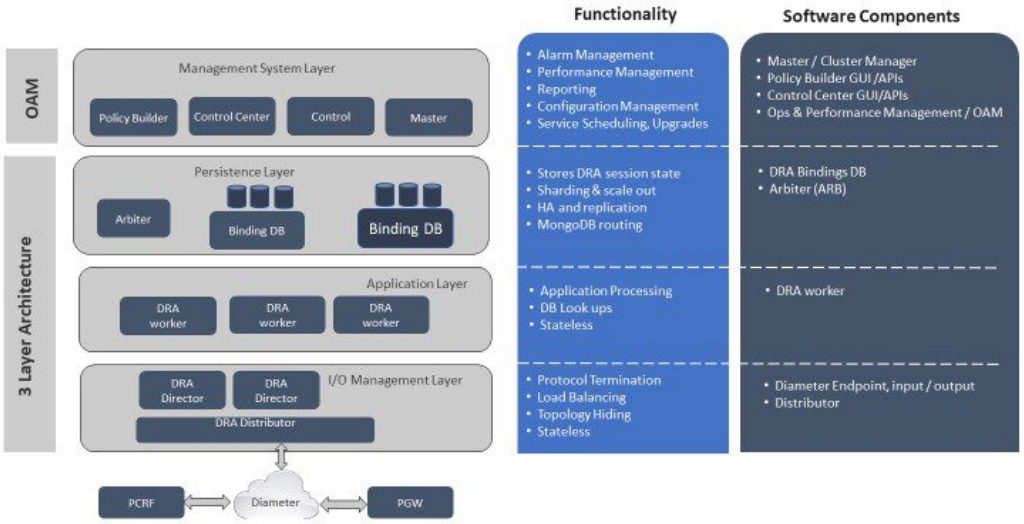
- Peer Aggregation:
  - Provides an aggregation point to eliminate full mesh of endpoint peer connections.
  - Addition of endpoint does not require reconfiguration of endpoints. Requires only the configuration of a new endpoint on DRA.
- Intelligent Routing:
  - Provides intelligent load balancing behavior 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.

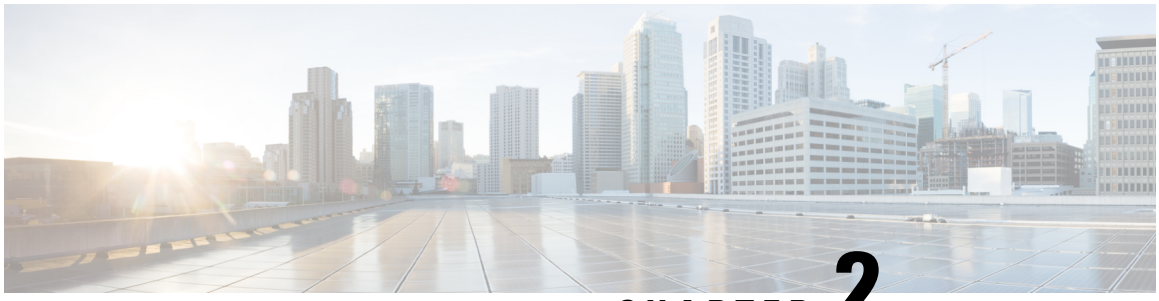
- Policy DRA is a functional element that supports Gx, Rx, Gy, Sy, and Sd diameter interfaces.

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.

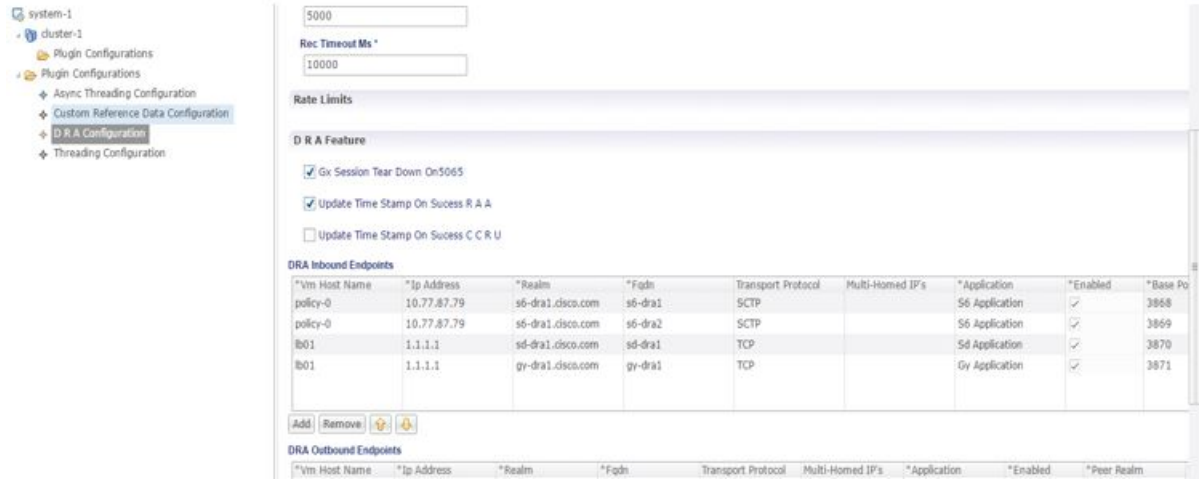
Configure the **DRA Inbound Endpoints** for incoming peer connections and **DRA Outbound Endpoints** for outgoing peer connection.

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



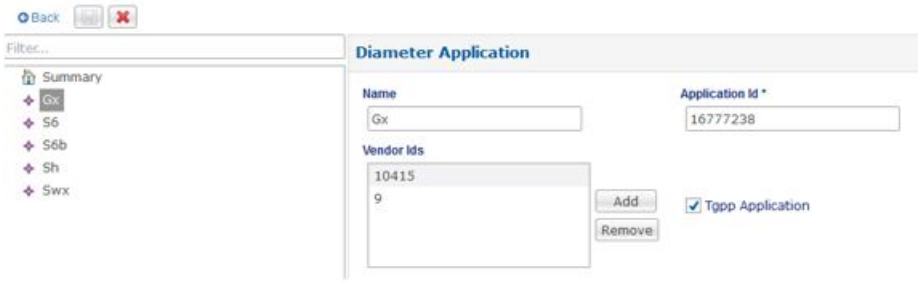
For more information, see DRA Configuration.

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



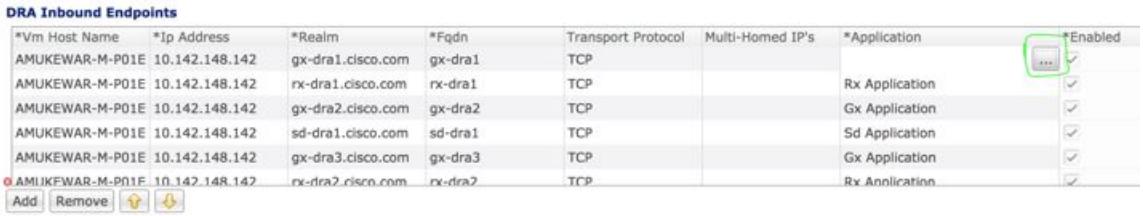
For more information, see Diameter Application, on page 85.

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.

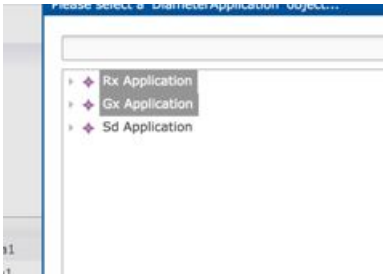
To configure multiple applications for a peer connection, go to vDRA Inbound Endpoints in DRA Plugin configuration. In the Applications field, select the button as shown:

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

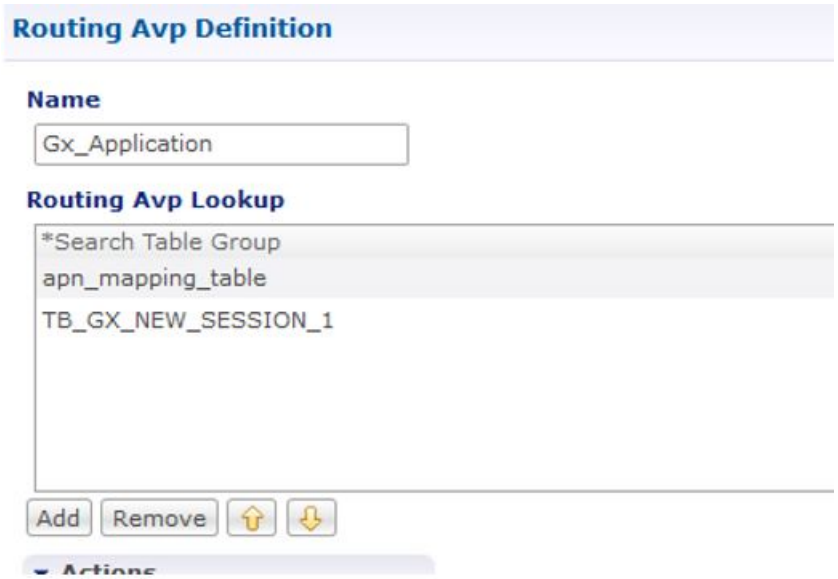
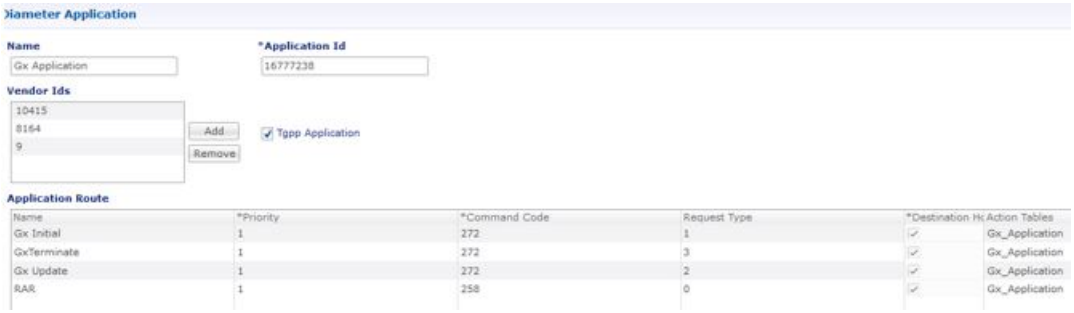


Figure 7: Diameter Application



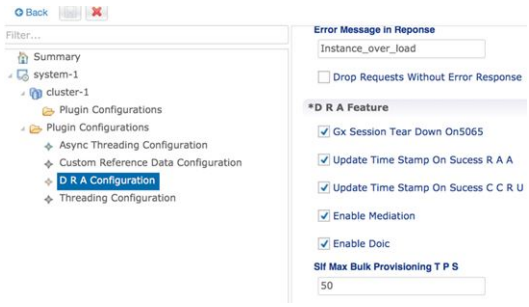
Enable Mediation

By default, mediation feature is disabled.

To enable the mediation, log in to Policy builder, select the “Enable Mediation” checkbox in **Systems** > **DRA Configuration** > **DRA Feature**.

Finally, publish the configuration changes.

Figure 8: Enable Mediation



Enable DOIC

By default, DOIC feature is disabled.

To enable the DOIC, log in to Policy Builder, select the "Enable DOIC" checkbox in **Systems** > **DRA Configuration** > **DRA Feature**.

You must also enable DOIC for the group in Peer Group SRK Mapping table as described in Peer Group SRK Mapping, on page 133.

Configure throttling using DOIC. For more information, see Configure Throttling of Diameter Messages Using DOIC, on page 26.

Publish the configuration changes.

Configure Interfaces for SCTP Multi-homing

As a pre-requisite for SCTP multi-homing, you must first move the physical interfaces of Director VM inside the diameter-endpoint container.

To move the IPv4 interfaces, perform the following commands in vDRA Master CLI mode:

```
config diameter host dra-director-0-dra-director-52wumnq5l2yl
 interface ens3 ipv4 address 10.77.87.79 broadcast 10.77.87.255 prefix-length 24
diameter host dra-director-0-dra-director-52wumnq5l2yl interface ens3
route default gateway 10.77.87.1
diameter host dra-director-0-dra-director-52wumnq5l2yl interface ens5
 ipv4 address 10.225.115.199 broadcast 10.225.115.255 prefix-length 24
diameter host dra-director-0-dra-director-52wumnq5l2yl interface ens5
 route default gateway 10.225.115.1
```

To move the IPv6 interfaces, perform the following commands in vDRA Master CLI mode:

```
config diameter host dra-director-0-dra-director-52wumnq5l2yl
interface ens6 ipv6 address 2003:3051::114 prefix-length 64
diameter host dra-director-0-dra-director-52wumnq5l2yl interface
ens6 route default gateway 2003:3051::1
diameter host dra-director-0-dra-director-52wumnq5l2yl interface
ens7 ipv6 address 2003:3052::114 prefix-length 64
diameter host dra-director-0-dra-director-52wumnq5l2yl interface
ens7 route default gateway 2003:3052::1
```

Commit the configuration.

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

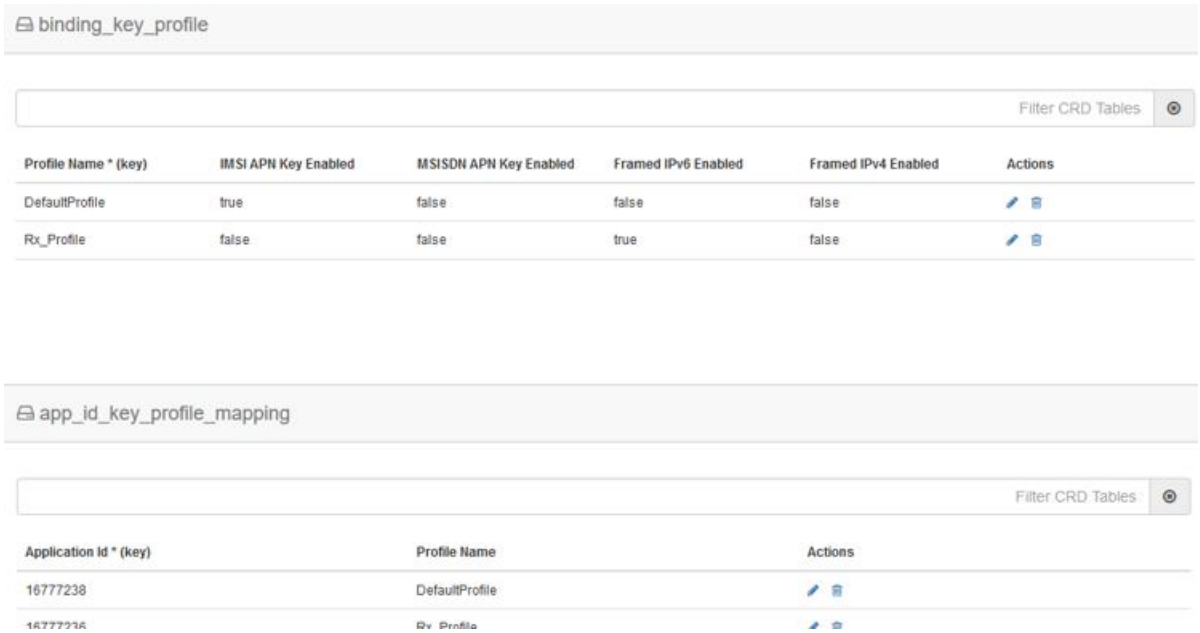
For Policy DRA, you must configure the binding keys for Gx sessions.

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 9: 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 10: 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:

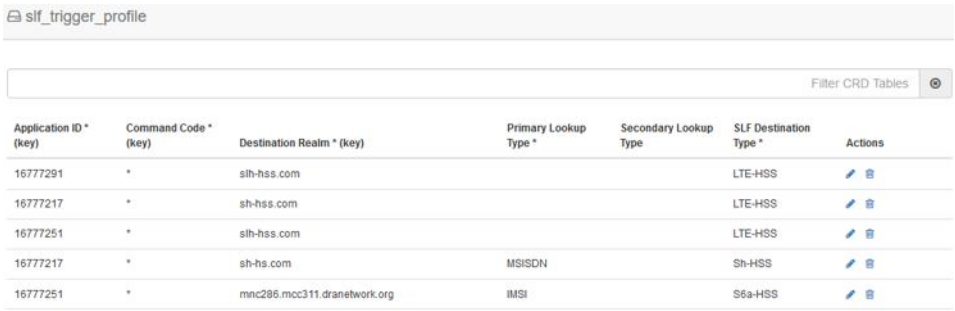
1. SLF Trigger Profile: For the incoming Diameter requests where Destination-Host is not present (or Destination-Host is present with same of DRA-Host Name) this SLF Trigger table is triggered. In this Table, there are three inputs that you need to configure:
   - Application-Id: Diameter Application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

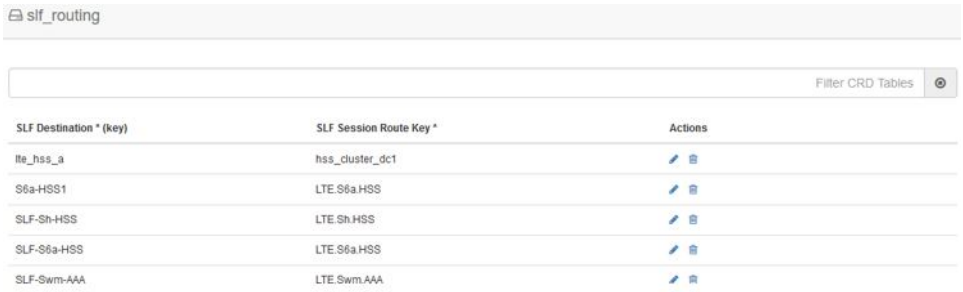
Figure 11: SLF Trigger Profile



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 12: 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 96.
4. CRD configuration: For more information, see Custom Reference Data Tables, on page 96.

Advanced Features

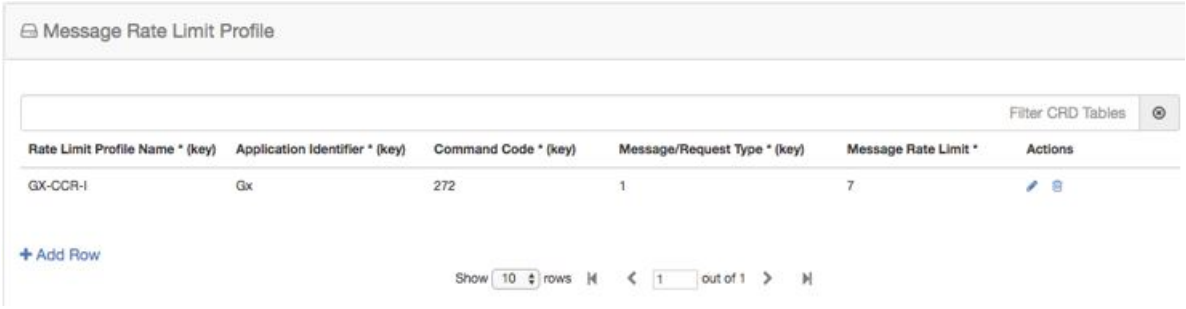
Configure Rate Limiting

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 13: Message Rate Limit Profile

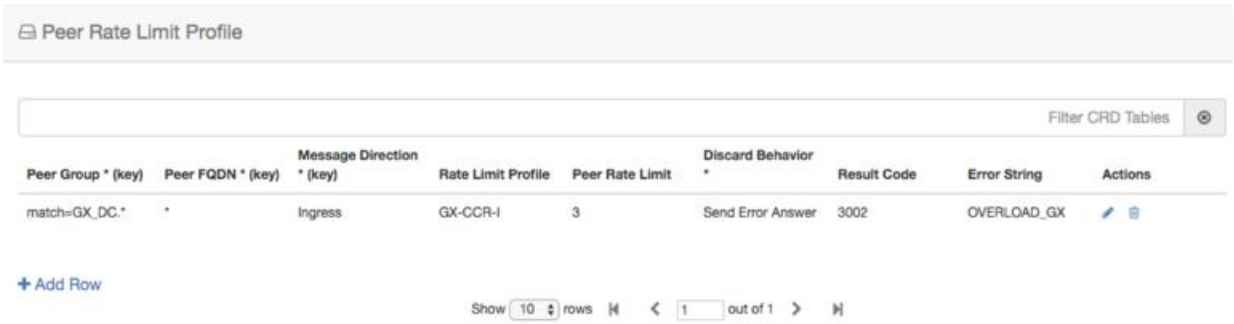


2. Configure the Peer Rate Limiting CRD table.

   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 14: Peer Rate Limit Profile



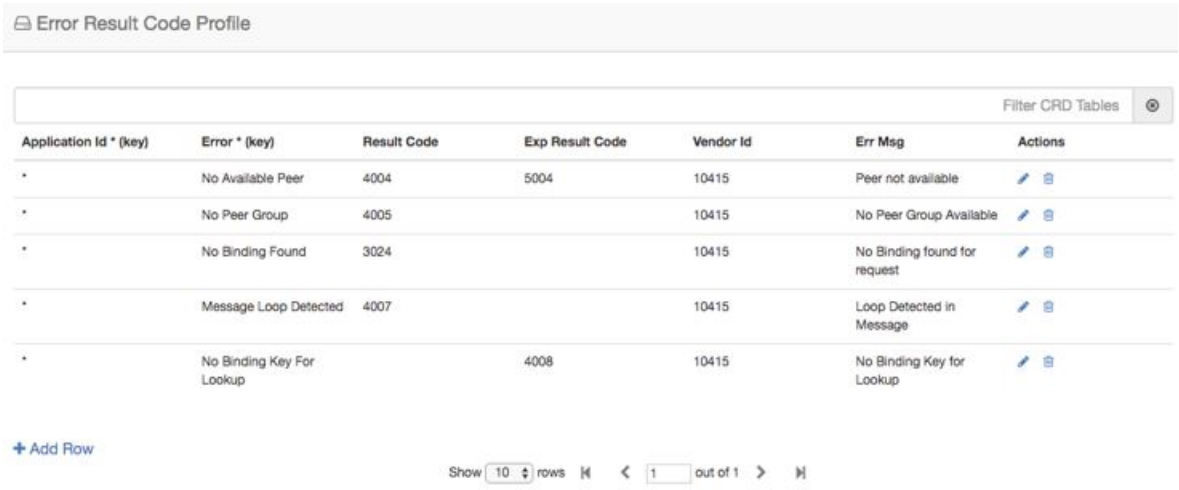
Configure Error Result Code Profile

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 15: 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.

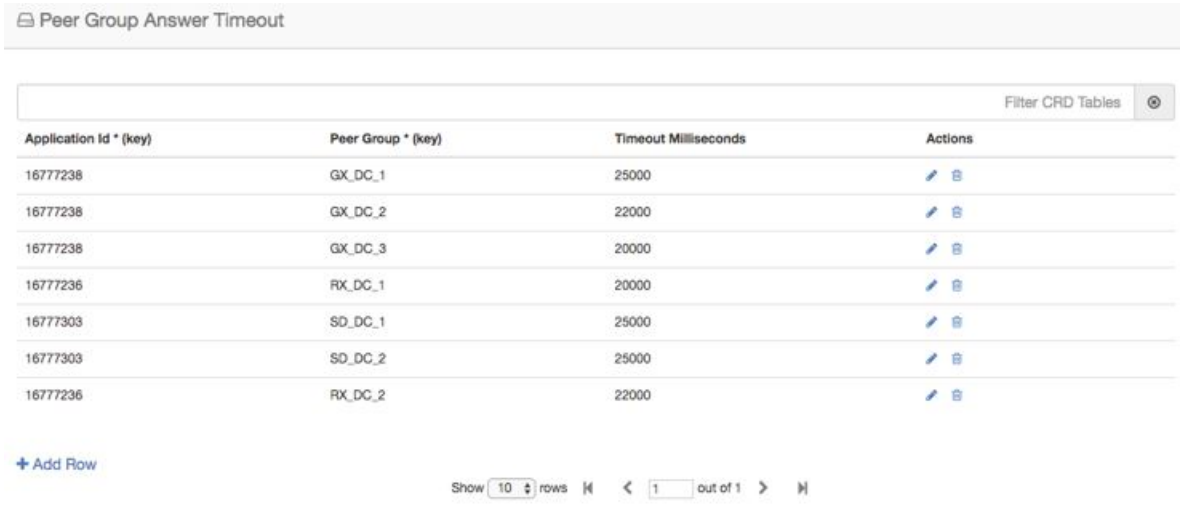
Configure Peer Group Answer Timeout

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 16: Peer Group Answer Timeout

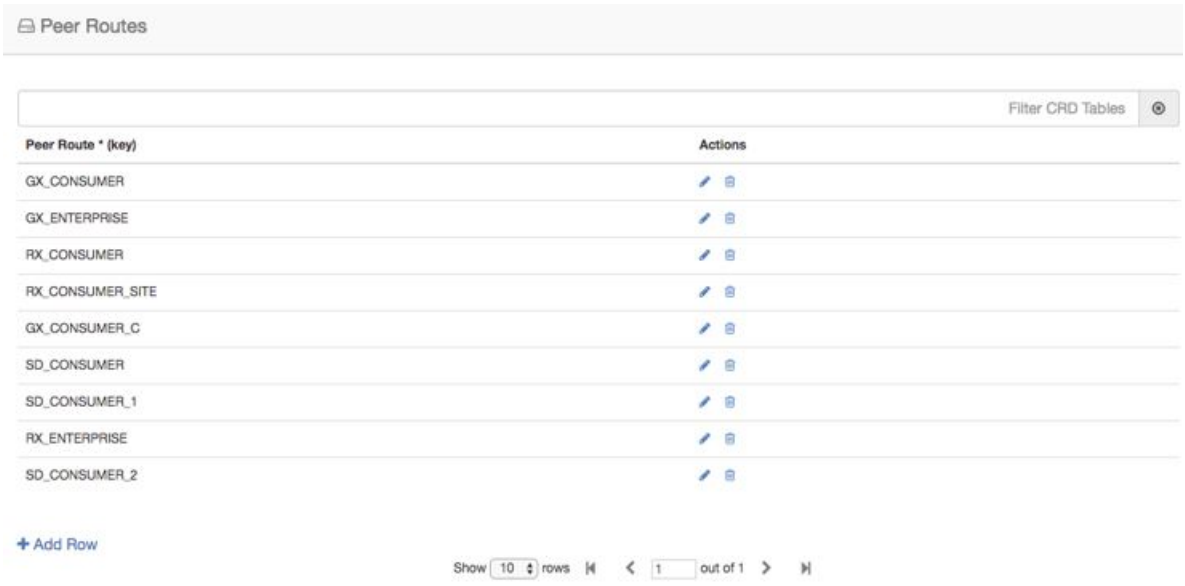


Configure Peer Load Balancing

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 17: Peer Routes



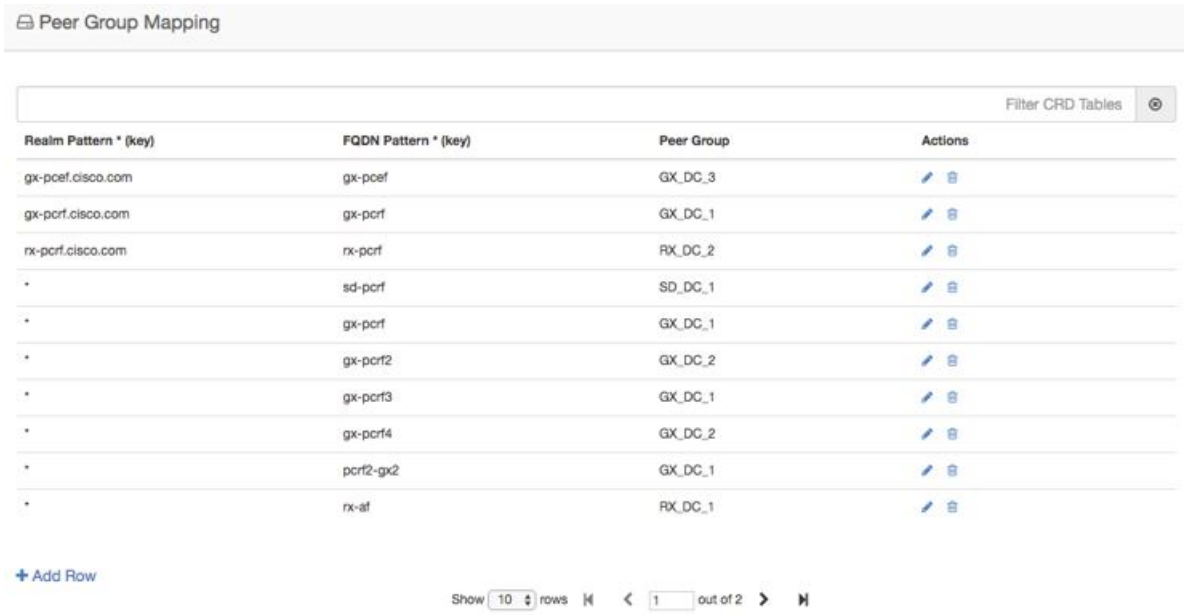
2. Configure the Peer Group Mapping table. This table defines the mapping of peers and realm with peer group.
   - Once the peer group is derived, CPS vDRA looks up this table to find the peers that belong to the derived peer group. Once DRA lists the peers in the peer group, it tries to match these 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 18: Peer Group Mapping



3. Configure the Peer Routing table.

   This table requires the following inputs:

   - Peer Route – derived from the Peer Route table
   - System Id – the current system ID of the CPS vDRA system
   - Peer Group – derived from the Peer Group Mapping table

   The outputs of this table are:

   - Precedence
   - Weight

   Precedence and Weight:

   Weight and Precedence are used to load balance the traffic among peers.

   If two peer groups are configured for same peer route and two peer groups are active then use the Precedence to select the row. If precedence is same then based on weight traffic will be load balanced among the two peer groups.

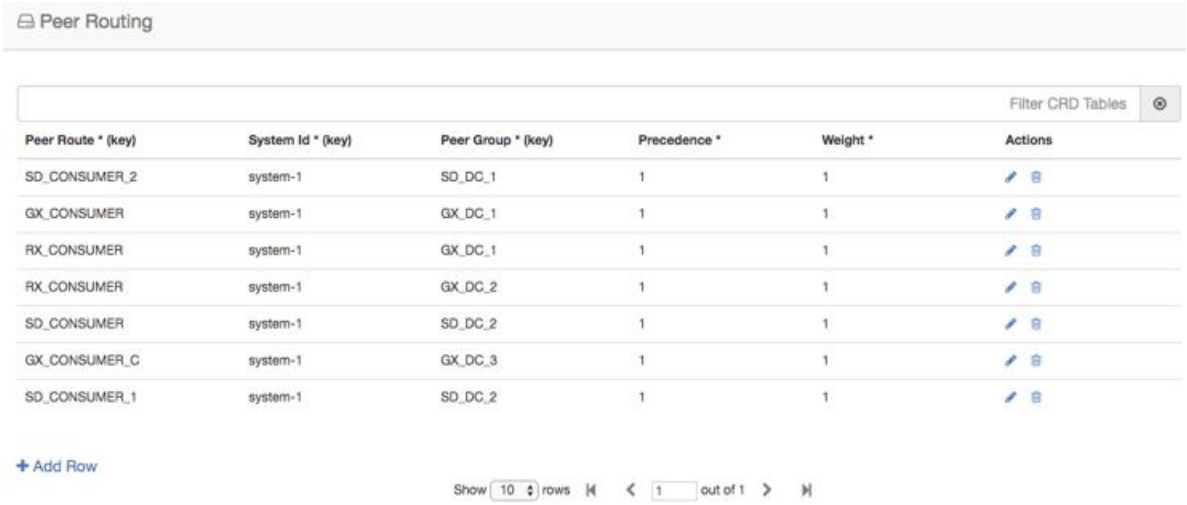
   In the following example, PR_1 and PR_2 have same precedence and PG_1 and PG_2 are active. Hence, traffic will be load balanced between PG_1 and PG_2 in the 1:2 ratio.

```
peer route   system ID  peer group  precedence  weight
PR_1             *       PG_1         10         10
PR_1             *       PG_2         10         20
```

   If two peer routes have different precedence and both peer groups are active then, peer group with lowest precedence value will be selected. In the following example, PG_1 will be selected.

```
peer route   system ID  peer group  precedence  weight
PR_1             *       PG_1          5         10
PR_1             *       PG_2          10        20
```

Figure 19: Peer Routing

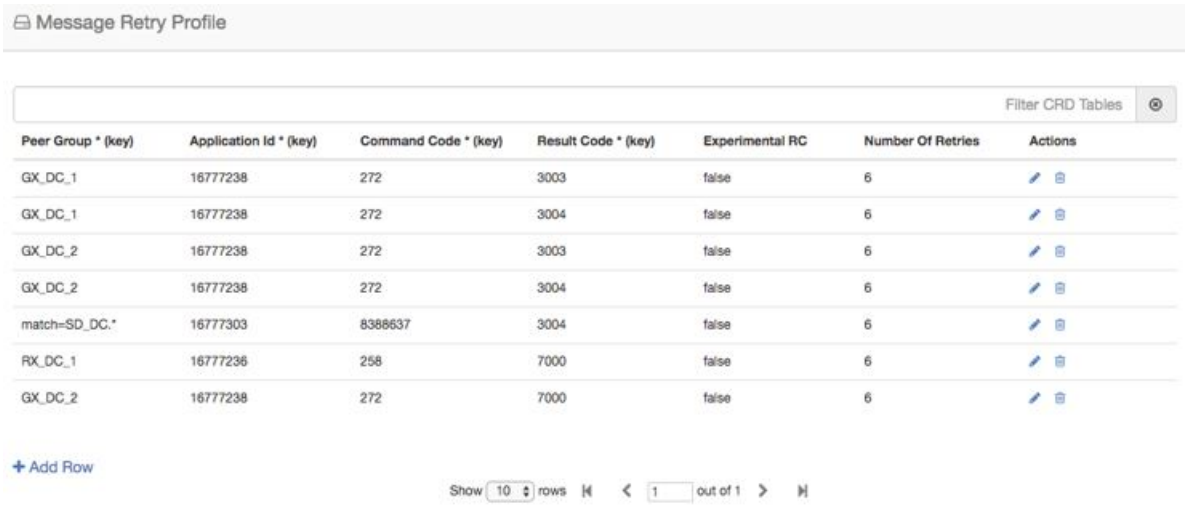


Configure Message Retries

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 20: 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.

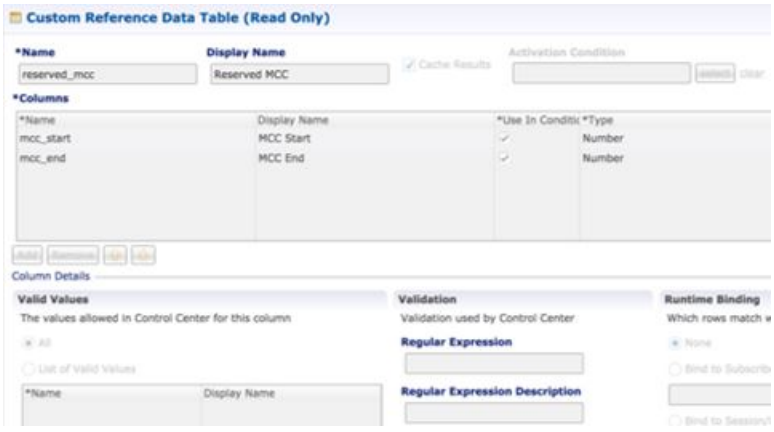
At the end, if no peer is found, retries stop. The retry also stops when the number of retries is exhausted and no response is received.

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 21: Reserved IMSI CRD Table



For more information, see Reserved IMSI, on page 115.

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

Figure 22: MCC Range

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

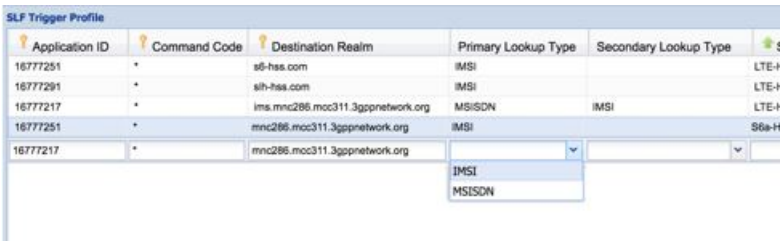
Configure Multiple Lookup in SLF Trigger Profile

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.

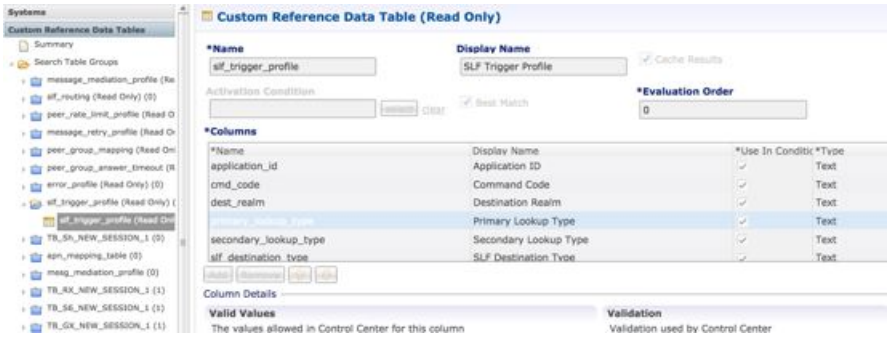
First, configure the columns in the CRD table as shown:

Figure 23: 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 24: SLF Trigger Profile

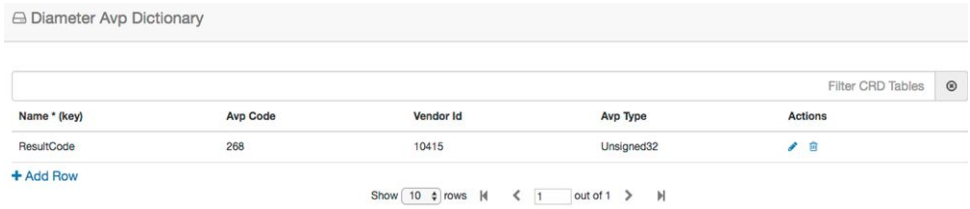


Modify Result Code for AVPs Using Mediation Rules

You can configure CRDs to modify the result code of AVPs and overwrite them with a specified value.

1. Define the AVPs in the Diameter Avp Dictionary as shown in the following example for ResultCode:

Figure 25: ResultCode AVP in Diameter Avp Dictionary



2. Add the condition that is used to match AVPs with a particular result code. In this example, ResultCodeIs3xxx matches the result code AVP with regular expression 3.* and checks for any 3xxx result code.

Figure 26: Add Result Code Condition in Avp Condition Profile

3. Define the action to be performed in the Avp Action Profile. In this case, ChangeResultcode is used to overwrite AVP values with the value 3002.

Figure 27: Define Action in Avp Action Profile

4. Add the mediation rule in the Message Mediation Profile.

   In the following example, the mediation rule is considered on answer in ingress direction, when application is 16777251, command is 318, and condition profile ResultCode3xxx is met. When all criteria are satisfied, the “ChangeResultCode” action profile is applied.

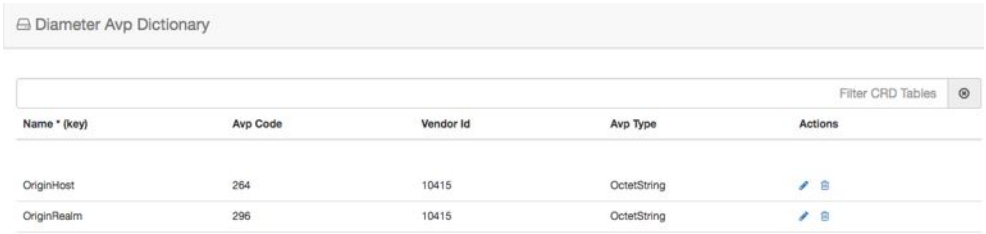
Figure 28: Mediation Rule

Hide Topology Using Mediation Rules

Define mediation rules to hide topology.

1. Define the AVPs in the Diameter Avp Dictionary as shown in the following example for OriginHost and OriginRealm:

Figure 29: Diameter Avp Dictionary



2. Define the action to be performed in the Avp Action Profile. In this case, Hiding is used to overwrite the Origin Host AVP with the value “dra” and to overwrite the Origin Realm AVP with the value “dra.cisco.com”.

Figure 30: Define Action in Avp Action Profile

3. Add the mediation rule in the Message Mediation Profile.

   In the following example, mediation rule is considered on request and answer in egress direction. It is applicable to all application, all commands, all peer groups and all message types (request/answer). When criteria satisfied, the “Hiding” action profile is applied.

Figure 31: Mediation Rule

Configure Mediation Rule for Proxy-Unaware Endpoints

When endpoints are not proxy-aware, PCRF may send an RAR to the vDRA without Destination Host AVP because it is not proxy friendly.

In such cases, you can configure vDRA to modify the Destination-Host of the RAR based on a regex match of the Diameter Session-ID.

1. Define the AVPs in the Diameter Avp Dictionary as shown in the following example for DestinationHost and SessionId:

Figure 32: Diameter Avp Dictionary

2. Define the action to be performed in the Avp Action Profile. In this case, ExtractHostFromSessionId is used to overwrite the DestinationHost AVP with the value dynamically derived from the regex string:

```
SubString("${SessionId}", "([^;]+);.*" , 1)
```

The session ID is taken from the message, then regex ([^;]+);.* is applied on the session ID and host is extracted.

For example, if the session ID is “pcef;1450914337;172.30.96.2;0”.

Regex ([^;]+);.* is applied on session ID and it will get 2 groups:

- group1= “pcef”
- group2 = “1450914337;172.30.96.2;0”

Hence, the Destination Host is set with value pcef.

Figure 33: Define Action in Avp Action Profile

3. Add the mediation rule in the Message Mediation Profile.

In the following example, mediation rule is considered on Gx RAR in ingress direction. When all the criteria are met, the “ExtractHostFromSessionId” action profile is applied. Action will set the DestinationHost AVP just after receiving message.

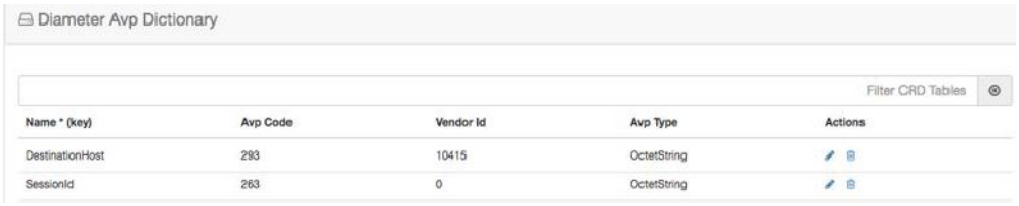
Figure 34: Mediation Rule

Add Prefix to an AVP Using Mediation Rules

You can define a prefix for an AVP.

1. First, define the AVPs in the Diameter Avp Dictionary as shown in the following example for the AVP, DestinationHost:

Figure 35: Defining DestinationHost in Diameter Avp Dictionary



2. Define the PrependLabel profile in the Avp Action Profile that adds the prefix "core-" to the DestinationHost AVP.

Figure 36: Define PrependLabel Profile in Avp Action Profile

3. Add the mediation rule in Message Mediation Profile as shown in the following example.

   In this example, the mediation rule is applied on s6 AIR request message in egress direction. When the peer group also matches hss-g, it will apply the “PrependLabel” action profile. Action will add the prefix “core-“ to destination host.

Figure 37: Mediation Rule in Message Mediation Profile

Configure Throttling of Diameter Messages Using DOIC

About DOIC

vDRA can throttle or divert Diameter requests towards PCRF, HSS, AAA, and OCS servers based on reporting of overloaded conditions using the architecture described in RFC 7683 Diameter Overload Indication Conveyance (DOIC).

The DOIC feature in vDRA involves the following high-level processes:

DOIC Enablement
DOIC feature in enabled/disabled in vDRA and also for the peer group.

DOIC Capability Exchange
If the DOIC feature is enabled, then for every request received, vDRA adds the OC-Supported-Features AVP to the request message to announce the DOIC support for loss of algorithm.

OCS Map
The OCS Map is the local vDRA cache, which contains the information of each reporting node with its latest overload control state information. On receipt of OC-OLR AVP from the peer, vDRA looks for the entry in OCS Map and updates the overload control state object.

Peer Overload Detection
vDRA checks the OCS entry from the OCS Map to determine whether the overload treatment is required or not.

Table Lookup
vDRA queries the Message Class Profile table to retrieve the Message classification. The message classification is used to query the DOIC Profile table to determine the Abatement Action that must be applied.

Overload Treatment
There are three types of overload treatment supported: Forward, Divert, and Drop. For more information, see DOIC Profile, on page 129.

Configuring DOIC in vDRA

To configure vDRA for throttling, perform the following steps:

1. Enable DOIC feature in vDRA in the Policy Builder as described in Enable DOIC, on page 9.
2. Enable DOIC for the peer group using the Peer Group SRK mapping table as described in Peer Group SRK Mapping, on page 133.
3. Define the AVP condition in the Diameter AVP Dictionary table. For more information, see Diameter Avp Dictionary, on page 130.
4. Configure the Message Class Profile table to get the message classification as output. For more information, see Message Class Profile, on page 136.

   If AVP conditions are evaluated to be true, then get Message Class. Message Class can be one of P0, P1, P2, P3, P4.
5. Use the Message class to define the abatement action in the DOIC Profile table. For more information, see DOIC Profile, on page 129.

Configure Throttling of Diameter Messages Using DRMP

vDRA can throttle Diameter requests based on the message priority sent in the Diameter request using the architecture described in RFC 7944 Diameter Routing Message Priority (DRMP).

To configure vDRA to use DRMP, perform the following steps as described in Configure Throttling of Diameter Messages Using DOIC, on page 26. Ensure you use the DRMP AVP in the Diameter Avp Dictionary table.

With this configuration, messages with Message Class P0 are not throttled in either rate limiter or by DOIC. For the rest of the messages, the throttling is applied as configured in DOIC table or in rate limiter.

Rate limiter currently throttles message regardless of the message class/ DRMP value. However, it will not throttle the message with message class P0.

Configure vDRA for eMPS

You can configure vDRA to recognize Diameter messages as Enhanced Multimedia Priority Services (eMPS) based on Diameter Application-Id, Command-Code, and AVP values.

An eMPS request is marked as a Priority 0 (or P0 in message classification) request and is not throttled or dropped.

1. Specify the AVP in the Diameter Avp Dictionary as shown in the following example:

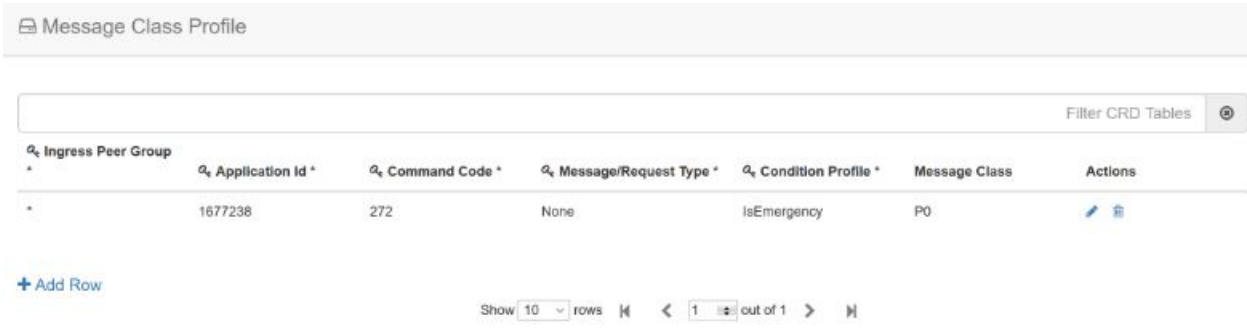
   Figure 38: Diameter Avp Dictionary

2. Specify the AVP condition in the Avp Condition Profile table.

   Figure 39: Avp Condition Profile

3. In Message Class Profile, configure the Message/Request Type as 'None', Message Class as P0, and Ingress Peer Group as '*', so that the message is treated as eMPS irrespective of the type of message (request or response) and the peer group it is sent to.

   Figure 40: Message Class Profile

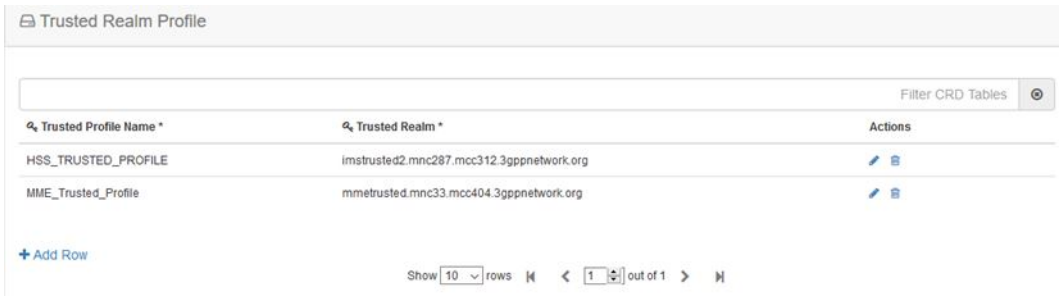


Configure Topology Hiding for S6a/d Diameter Application

Configure topology hiding for the S6a/d Diameter application (Diameter interface between the MME/SGSN and the HSS).

1. Configure a trusted realm profile: Define the name of a Trusted Profile and associate a realm which can be trusted with it.

   Figure 41: Trusted Realm Profile



2. Configure the Protected realm (where vDRA is to provide the topology hiding) and link it to the trusted profile.

   Depending on the requests, if the destination/origin realm is trusted, vDRA skips the topology hiding. If the realm is not trusted, vDRA provides topology hiding.

   Figure 42: Protected Realm Trusted Profile

3. In the MME Alias Map, define the MME FQDN, which needs to be protected, and bind it to the Aliases that are used for topology hiding.

   Figure 43: MME Alias Map

4. Define all the HSS Aliases that are used for HSS topology hiding by vDRAin the HSS Aliases table. Specify whether the defined alias is shared or not by multiple protected HSSs.

Figure 44: HSS Aliases

5. In the HSS Alias Map, map the protected HSS FQDN with the HSS ALIAS defined in previous HSS Aliases table. These aliases are used for the topology hiding of the protected HSSs.

Figure 45: HSS Alias Map

Manual Peer Disconnection using REST API

CPS vDRA provides a REST API that you can use to manually disconnect peer connections from vDRA. To disconnect a single peer connection, you need to provide the peer connection key information in the API method.

Disconnect a single peer connection

To disconnect a single peer, provide the peer-connection-key (that you can find in the peer endpoint details) in the PUT method as shown:

```
PUT localActivePeerEndpoints/disconnect/key/<peer-connection-key>
```

Responses:

```
200 OK
{
    "success": {
        "code": 0,
        "message": "Request completed successfully"
    }
}
```

Note that success means the request is accepted, and vDRA attempts to disconnect the peer connection. The success response does not indicate that the peer connection has been disconnected.

Failure response:

```
404 Not Found

{
    "error": {
        "code": 2014,
        "message": "Data for key <peer-connection-key> is not found"
    }
}
```

Policy DRA Relay Configuration

Policy DRA requires a full-mesh topology to be able to send traffic between different sites. In a multi-site DRA network, relay connections are required between every pair of relay endpoints.

Relay Endpoint Configuration

Relay endpoint is used to indicate the IP address/port combination at which vDRA listens for diameter connections. These connections are used to send/receive diameter traffic between sites.

Relay endpoints for vDRA can be configured using the Policy Builder vDRA configuration plugin.

The following fields are required for the configuration:

Vm Host Name
VM Host Name is the host name of the VM. The value '*' can be used to match any host name.

Instance Id
The Instance ID field should be set to '1'. Currently, no other value is supported.

Ip Address
IPv4 or IPv6 address of the interface used to listen for relay connections. Currently, only physical IP addresses are supported. VIP addresses are not supported.

Realm
Supports configurable relay endpoint realm name.

Port
Port that is used to listen for relay connections. The default value is '4868'. Currently, the range of ports supported is '4868-4878'.

Fqdn
FQDN is the string that is used to send the relay endpoint information to the other sites.

Enabled
Used to enable or disable the relay endpoint. It should be enabled for the relay endpoint to start listening on the configured port and IP address.

Note Ensure that the DNS configuration is completed before publishing the PB changes. If DNS configuration changes are made later, the Diameter endpoints must be restarted for the changes to take effect.

Control Plane Configuration

The control plane configuration performs the following functions:

- If the IP address is a site-local address, a global Control Plane server is created that listens to connections.
- If the IP address not a site-local address, the connections are initiated from diameter-endpoint containers to the global Control Plane server running at specified IP/port. These connections are used to send and receive global control plane information between sites.

Note It is recommended not to use VIP address. Instead use physical IP address for control-plane relay link.

The following example illustrates control plane configuration:

```
admin@orchestrator# show running-config control-plane
control-plane relay gx-dra1-relay-v6-1
 address [2001:421:27c1:913:250:56ff:fea6:12]
 port    6379
```

Note The IPv6 address must be encapsulated with square [] brackets.

DNS Host Configuration

The following entities must be DNS resolvable and must have DNS Host configuration entries:

- Control Plane servers from all sites
- Relay Endpoint FQDNs from all sites

The following command is an example of DNS Host configuration:

```
admin@orchestrator# show running-config network dns host
network dns host gx-dra1-relay-v6 local
 address 2001:421:27c1:913:250:56ff:fea6:48
```

Virtual-IP Configuration

The following entities must have virtual-IP configuration entries:

- Control Plane Servers with VIP addresses

Relay Configuration for a 6-Site Policy DRA System

The following example describes how to configure the relay configuration for a six site Policy DRA system.

Relay Endpoints

There must be two relay endpoint entries configured in Policy Builder per site; one entry each for the two diameter-endpoint containers.

Control Plane Configuration

There must be at least two global Control Plane servers configured per site; one on each diameter-endpoint for redundancy.

The global Control Plane servers must have connections between each other. This means that every site must have control plane configuration entries for the control plane servers of every other site.

For a six site system, every site must have 12 Control Plane configuration entries (6 sites x 2 global Control Plane servers per site) configured in CLI.

DNS Host Configuration

There should be one DNS Host entry for each of the global Control Plane configuration entries. In a six site system, every site should have 12 DNS Host entries related to global Control Plane entries of all sites.

There should be one DNS Host entry for each Relay endpoints of every site. In a six site system, every site should have 12 DNS Host entries related to Relay Endpoint entries of all sites.

Virtual-IP Configuration

If any global Control Plane servers have VIP addresses, they should have Virtual-IP Configuration.

Every site should have Virtual-IP Configuration for any VIP addresses that are associated to that site.

Configuring Application based Sharding

Restriction

- Migration from hash-based sharding to zone based sharding is not supported.
- Since this feature supports only static sharding configurations, all database configurations needs to be committed with single commit operation.
- Currently, dynamic addition of new shards after initial configuration is not supported as it supports only static shard configurations. If one needs to add new shards or edit existing shards, it is mandatory to clean up the complete database and reconfigure again.

Procedure

Step 1 Login to Binding VNF CLI to configure application sharding on Binding VNF.

a) Configure sharded cluster master.

```
database cluster session sharded-cluster-master <true | false>
```

true: enables sharded cluster master. Internally sharding metadata creation gets triggered.

false: Sharding metadata creation is not enabled.

Example:

```
database cluster session sharded-cluster-master true
```

Note
Only one site among the mated pair which initially gets configured has to be enabled (true) so that sharding metadata creation happens from that site. For rest of the sites, it just adds members and uses existing sharding metadata.

b) Configure multi-database collections: This configuration enables the creation of internal logical shards.

```
database cluster session multi-db-collections <1..4>
```

Default: 1

For example, if configured with 2, each sharding database metadata creates an extra logical shard for the shard configured as a part of database creation. So total number of shards will be double the number.

```
database cluster session multi-db-collections 2
```

c) Configuring sharding database.

```
database cluster <cluster name> sharding-db <sharding-db-name> address <ip address> [port <port>]
database cluster <cluster name> sharding-db-seed <shardingdb name>
```

Example:

```
database cluster session sharding-db shdb-4 address 192.168.11.43
database cluster session sharding-db shdb-5 address 192.168.11.44
database cluster session sharding-db-seed shdb-4
```

Note
Port is optional. By default, sharding database uses port 27019.

d) Configure shards.

```
database cluster <cluster-name> shard <shard-name>
```

Non-arbiter shard member:

```
database cluster <cluster-name> shard <shard-name> shard-server <server-name> storage-engine MMAPv1
 address <ipaddress> port <port> priority <priority>
```

Arbiter shard member:

```
database cluster <cluster-name> shard <shard-name> shard-server <server-name> arbiter true
storage-engine MMAPv1 address <ipaddress> port <port>
```

Shard seed:

```
database cluster <cluster-name> shard <shard-name> shard-server-seed <seed-server>
```

Note
Seed server has to be one of the non-arbiter member from the same shard.

Example:

```
database cluster session shard shard-21
database cluster session shard shard-21 shard-server server-x storage-engine MMAPv1 address
192.168.11.43 port 27026 priority 10
database cluster session shard shard-21 shard-server server-y storage-engine MMAPv1 address
192.168.11.44 port 27026 priority 5
database cluster session shard shard-21 shard-server arbiter-21 arbiter true storage-engine MMAPv1
 address 192.168.11.42 port 27026
database cluster session shard shard-21 shard-server-seed server-x
```

e) Make sure all the databases are UP with appropriate status using `show database status` commad.

Sample output:

```
admin@orchestrator[an-dbmaster]# show database status cluster-name session
                                                          CLUSTER
ADDRESS         PORT   NAME        STATUS     TYPE         NAME     SHARD     REPLICA SET
-------------------------------------------------------------------------------------------------
192.168.11.42  27026  arbiter-21  ARBITER    replica_set  session  shard-21  rs-shard-21
192.168.11.43  27026  server-x    PRIMARY    replica_set  session  shard-21  rs-shard-21
192.168.11.44  27026  server-y    SECONDARY  replica_set  session  shard-21  rs-shard-21
192.168.11.42  27027  arbiter-22  ARBITER    replica_set  session  shard-22  rs-shard-22
192.168.11.43  27027  server-x    SECONDARY  replica_set  session  shard-22  rs-shard-22
192.168.11.44  27027  server-y    PRIMARY    replica_set  session  shard-22  rs-shard-22
192.168.11.43  27019  session     PRIMARY    shard_db     session  shdb-4    session-sharddb
192.168.11.44  27019  session     SECONDARY  shard_db     session  shdb-5    session-sharddb

admin@orchestrator[an-dbmaster]#
```

Step 2 Login to DRA VNF CLI to configure application sharding on DRA VNF.

a) Configure shard metadata database connection.

```
binding shard-metadata-db-connection <dbName> <address> <port>
```

Example:

```
binding shard-metadata-db-connection drasession 182.22.31.207 27019
```

- <*dbName*>: Database name for which connection URI needs to be set.

  Possible values are: all, drasession, imsiapn, ipv4, ipv6, msisdnapn, range.

- <*address*>: Address of the binding shard metadata database. This is either an IP address or an FQDN.
- <*port*>: Port of the binding shard metadata database.

Step 3 Login to DRA VNF CLI to configure same connection pool on imsiapn-msisdnapn database.

Note
We do not recommended configuring connection pool for small setups. It is required for setups whose database spans across 48 shards or more.

a) Configure same connection pool on imsiapn-msisdnapn database transactions.

```
binding cluster-binding-dbs imsiapn-msisdnapn
```

Example:

```
binding  cluster-binding-dbs imsiapn-msisdnapn
```

Example

The following are the examples for complete set of database cluster configuration on Binding VNF.

```
config
database cluster session sharded-cluster-master true
database cluster session multi-db-collections 2
database cluster session sharding-db shdb-4 address 192.168.11.43 port 27019
database cluster session sharding-db shdb-5 address 192.168.11.44 port 27019
database cluster session sharding-db-seed shdb-4

database cluster session shard shard-21
database cluster session shard shard-21 shard-server server-x storage-engine MMAPv1 address
 192.168.11.43 port 27026 priority 10
database cluster session shard shard-21 shard-server server-y storage-engine MMAPv1 address
 192.168.11.44 port 27026 priority 5
database cluster session shard shard-21 shard-server arbiter-21 arbiter true storage-engine
 MMAPv1 address 192.168.11.42 port 27026
database cluster session shard shard-21 shard-server-seed server-x

database cluster session shard shard-22
database cluster session shard shard-22 shard-server server-y storage-engine MMAPv1 address
 192.168.11.44 port 27027 priority 10
database cluster session shard shard-22 shard-server server-x storage-engine MMAPv1 address
 192.168.11.43 port 27027 priority 5
database cluster session shard shard-22 shard-server arbiter-22 arbiter true storage-engine
 MMAPv1 address 192.168.11.42 port 27027
database cluster session shard shard-22 shard-server-seed server-y
commit
```

The following is an example to configure shard metadata database connection for session and all bindings on DRA VNF.

```
config
binding shard-metadata-db-connection drasession  182.22.31.207  27019
binding shard-metadata-db-connection ipv6  182.22.31.207  27019
binding shard-metadata-db-connection ipv4  182.22.31.208  27019
```

```
binding shard-metadata-db-connection imsiapn  182.22.31.208  27019
binding shard-metadata-db-connection msisdnapn  182.22.31.208  27019
commit
```

Configuring Binding Database Overload

Procedure

Login to DRA VNF CLI to configure maximum record limit on session and bindings database.

```
binding db-max-record-limit <db-Name> <limit>
```

<*db-Name*>: Database name on which maximum record limit has to be set.

Possible values are drasession, imsiapn, ipv4, ipv6, msisdnapn.

<*limit*>: Numeric value indicating maximum records that could be stored in given database.

Example:

```
binding db-max-record-limit drasession 10000
```

Example

The following is an example for setting maximum record limit on session and all bindings database:

```
config
binding db-max-record-limit drasession 10000
binding db-max-record-limit ipv6 10000
binding db-max-record-limit ipv4 5000
binding db-max-record-limit imsiapn 5000
binding db-max-record-limit msisdnapn 5000
commit
```

Change Admin User Password for MongoDB Authentication

Caution The following procedure is service impacting and needs to be execute on all sites at the same time.

Procedure

Step 1 Execute the following step on all binding VNF of all sites.

```
db-authentication change-password database mongo user adminuser
```

When prompt is displayed, enter the current password and new password.

```
admin@orchestrator[binding-master]# db-authentication change-password database mongo user
adminuser
```

```
Value for 'current-password' (<string>): ******
Value for 'new-password' (<string>): ********
result SUCCESS
```

Step 2 Execute the following step on all DRA VNF of all sites.

```
db-authentication change-password database mongo user adminuser
```

When prompt is displayed, enter the current password and new password.

```
db-authentication change-password database mongo user
adminuser
Value for 'current-password' (<string>): ******
Value for 'new-password' (<string>): ********
result SUCCESS
```

Step 3 Execute `db-authentication sync-password database mongo` command on all binding VNF of all sites.

```
admin@orchestrator[binding-master]# db-authentication sync-password database mongo
result
SUCCESS : Mongo password sync successful
```

Step 4 Execute `db-authentication sync-password database mongo` command on all DRA VNF of all sites.

```
db-authentication sync-password database mongo
result
SUCCESS : Mongo password sync successful
```

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
```

Configuring Zone Aware Sharding

Restriction

- Migration from hash-based sharding to zone based sharding is not supported.
- Database reconfigure steps must be followed to enable zone aware sharding on existing database cluster.
- Configure the databases on Binding VNF site by site. Not recommended to apply database configurations on all sites in parallel.
- Within single commit all the database configurations needs to be committed.
- Currently, dynamic addition of new shards after initial configuration is not supported.
- You should make sure that there are no overlapping IPv6 addresses within a zone or across multiple zones within a database cluster (or) across multiple database clusters while doing the database configuration. IPv6 address mentioned as the start/end values while configuring the ranges should be unique.
- If you want to add new shards or edit existing shards, it is mandatory to clean up the complete database and reconfigure again.

Procedure

Step 1 Login to Binding VNF CLI to configure database for zone sharding.

a) Configuring sharding database.

```
database cluster <cluster name> sharding-db <shardingdb name> address <VM ip address> [port
<portNum>]
```

```
database cluster <cluster name> sharding-db-seed <shardingdb name>
```

Example:

```
admin@orchestrator[an-dbmaster]#
database cluster binding sharding-db shdb-1 address 182.22.31.10 port 27020
database cluster binding sharding-db shdb-2 address 182.22.31.11
database cluster binding sharding-db shdb-3 address 182.22.31.12
database cluster binding sharding-db-seed shdb-1
```

Note
<portNum> is optional. By default, the port number is 27019.

b) Enable IPv6 zone sharding.

```
database cluster <cluster name> ipv6-zone-sharding true
```

Example:

```
admin@orchestrator[an-dbmaster]# database cluster binding ipv6-zone-sharding true
```

c) Create zone and its range(s).

```
database cluster <cluster name> ipv6-zones-range <zone-name> zone-range <range-name> start <pool
 starting Prefix> end <pool ending Prefix>
```

Example:

```
admin@orchestrator[an-dbmaster]# database cluster binding ipv6-zones-range pune zone-range range1
 start 2003:3051:0000:0001 end 2003:3051:0000:0500
```

Note
It is possible to create multiple ranges for each zone. Configure the ranges in 64-bit Framed IPv6 Prefixes only.

d) Mapping of zone to shard.

```
database cluster <cluster name> shard <shard name> zone-name <zone-name>
```

Example:

```
admin@orchestrator[an-dbmaster]# database cluster binding shard shard-1 zone-name pune
```

e) Sample database configuration with two shards (two zones with two ranges per zone).

Step 2 Login to the master(primary/seed) mongo sharding database and check configured ranges, zones, shard/range mapping information.

```
database cluster binding sharded-cluster-master true
database cluster binding ipv6-zone-sharding true
database cluster binding ipv6-zones-range mumbai zone-range r1 start 2008:5000:0000:0100 end 2008:5000:0000:0500
database cluster binding ipv6-zones-range mumbai zone-range r2 start 2009:5000:0000:0100 end 2009:5000:0000:0500
database cluster binding ipv6-zones-range pune zone-range r1 start 2011:6000:0000:0001 end 2011:6000:0000:0500
database cluster binding ipv6-zones-range pune zone-range r2 start 2012:6000:0000:0001 end 2012:6000:0000:0500
database cluster binding sharding-db shdb-1 address 182.22.31.10
database cluster binding sharding-db shdb-2 address 182.22.31.11
database cluster binding sharding-db shdb-3 address 182.22.31.12
database cluster binding sharding-db-seed shdb-1

database cluster binding shard shard-1
database cluster binding shard shard-1 shard-server server-a storage-engine MMAPv1 address 182.22.31.13 port 27017 priority 10
database cluster binding shard shard-1 shard-server server-b storage-engine MMAPv1 address 182.22.31.14 port 27017 priority 5
database cluster binding shard shard-1 shard-server arbiter-1 arbiter true storage-engine MMAPv1 address 182.22.31.11 port 27017
database cluster binding shard shard-1 shard-server-seed server-a

database cluster binding shard shard-2
database cluster binding shard shard-2 shard-server server-b storage-engine MMAPv1 address 182.22.31.14 port 27018 priority 10
database cluster binding shard shard-2 shard-server server-a storage-engine MMAPv1 address 182.22.31.13 port 27018 priority 5
database cluster binding shard shard-2 shard-server arbiter-2 arbiter true storage-engine MMAPv1 address 182.22.31.11 port 27018
database cluster binding shard shard-2 shard-server-seed server-b

database cluster binding shard shard-1 zone-name mumbai
database cluster binding shard shard-2 zone-name pune
```

a) Check configured ranges.

Example:

```
use ipv6ShardDB
switched to db ipv6ShardDB
binding-sharddb:PRIMARY>
binding-sharddb:PRIMARY> db.shards.find()
```

```
binding-sharddb:PRIMARY> db.shards.find()
{ "_id" : 1, "name" : "shard-1", "hosts" : "182.22.31.13:27017,182.22.31.14:27017", "zone" : "mumbai" }
{ "_id" : 2, "name" : "shard-2", "hosts" : "182.22.31.13:27018,182.22.31.14:27018", "zone" : "pune" }
{ "_id" : 3, "name" : "shard-3", "hosts" : "182.22.31.13:27020,182.22.31.14:27020", "zone" : "hyd" }
{ "_id" : 4, "name" : "shard-4", "hosts" : "182.22.31.13:27021,182.22.31.14:27021", "zone" : "bglr" }
{ "_id" : 5, "name" : "shard-5", "hosts" : "182.22.31.13:27022,182.22.31.14:27022", "zone" : "chennai" }
{ "_id" : 6, "name" : "shard-6", "hosts" : "182.22.31.13:27023,182.22.31.14:27023", "zone" : "hyd" }
{ "_id" : 7, "name" : "shard-7", "hosts" : "182.22.31.13:27024,182.22.31.14:27024", "zone" : "bglr" }
{ "_id" : 8, "name" : "shard-8", "hosts" : "182.22.31.13:27025,182.22.31.14:27025", "zone" : "pune" }
binding-sharddb:PRIMARY>
```

b) Checking configured zones.

Example:

```
binding-sharddb:PRIMARY> db.zoneinfo.find()
{ "_id" : 1, "name" : "r1", "start" : "2017:6000:0000:0001", "end" : "2017:6000:0000:0500", "zone" : "bglr" }
{ "_id" : 2, "name" : "r2", "start" : "2018:6000:0000:0001", "end" : "2018:6000:0000:0500", "zone" : "bglr" }
{ "_id" : 3, "name" : "r1", "start" : "2013:6000:0000:0001", "end" : "2013:6000:0000:0500", "zone" : "chennai" }
{ "_id" : 4, "name" : "r2", "start" : "2014:6000:0000:0001", "end" : "2014:6000:0000:0500", "zone" : "chennai" }
{ "_id" : 5, "name" : "r1", "start" : "2015:6000:0000:0001", "end" : "2015:6000:0000:0500", "zone" : "hyd" }
{ "_id" : 6, "name" : "r2", "start" : "2016:6000:0000:0001", "end" : "2016:6000:0000:0500", "zone" : "hyd" }
{ "_id" : 7, "name" : "r1", "start" : "2008:5000:0000:0100", "end" : "2008:5000:0000:0500", "zone" : "mumbai" }
{ "_id" : 8, "name" : "r2", "start" : "2009:5000:0000:0100", "end" : "2009:5000:0000:0500", "zone" : "mumbai" }
{ "_id" : 9, "name" : "r1", "start" : "2011:6000:0000:0001", "end" : "2011:6000:0000:0500", "zone" : "pune" }
{ "_id" : 10, "name" : "r2", "start" : "2012:6000:0000:0001", "end" : "2012:6000:0000:0500", "zone" : "pune" }
```

c) Checking shard/range mapping.

Example:

```
binding-sharddb:PRIMARY> db.buckets.find()
{ "_id" : ObjectId("5cb9845c63ebec62ea803d42"), "bucket-id" : 1, "shard" : 4, "migration" : false, "zone" : "bglr" }
{ "_id" : ObjectId("5cb9845c63ebec62ea803d43"), "bucket-id" : 2, "shard" : 4, "migration" : false, "zone" : "bglr" }
{ "_id" : ObjectId("5cb9845c63ebec62ea803d44"), "bucket-id" : 3, "shard" : 4, "migration" : false, "zone" : "bglr" }
{ "_id" : ObjectId("5cb9845c63ebec62ea803d45"), "bucket-id" : 4, "shard" : 4, "migration" : false, "zone" : "bglr" }
{ "_id" : ObjectId("5cb9845c63ebec62ea803d46"), "bucket-id" : 5, "shard" : 4, "migration" : false, "zone" : "bglr" }
{ "_id" : ObjectId("5cb9845c63ebec62ea803d47"), "bucket-id" : 6, "shard" : 4, "migration" : false, "zone" : "bglr" }
{ "_id" : ObjectId("5cb9845c63ebec62ea803d48"), "bucket-id" : 7, "shard" : 4, "migration" : false, "zone" : "bglr" }
```

Modifying Zone Aware Sharding

Note This section is applicable when you want to add/update/delete zones and ranges after performing initial database configuration.

If you have modified already configured zones and ranges in database, you need to perform the following steps to have sharding database metadata updated with new modified configuration commits.

Procedure

Step 1 Connect to session database cluster sharding primary database for which configuration changes have been committed.

a) Use `show database status` command to get the sharding database primary IP address and port number of a particular database cluster.

b) Use `docker exec -it orchestrator bash` command to connect to sharding database.
c) Connect to sharding database.

If MongoDB authentication is enabled, execute `mongodb --ipv6 mongodb://adminuser:password@[IPAddress]:Port/admin` command.

OR

If MongoDB authentication is disabled, execute `mongodb --ipv6 mongodb://[IPAddress]:Port` command.

d) Verify that after completing 1.c, on page 44, the prompt is redirected to mongo shell of the same.

Step 2 Remove the zone collection information using the following commands from inside mongo shell:

```
use ipv6ShardDB
db.zoneinfo.remove({})
```

Step 3 Wait for 30 seconds and verify that zone collection information is updated with the new committed configurations.

```
db.zoneinfo.find()
exit
```

Configure Docker Overlay for vDRA

This configuration allows to enable the Docker Overlay network driver and detach the Weave network in vDRA.

Before you begin

Before you migrate from Weave network to Docker Overlay,

- Verify that the system and all container services are fully operational and healthy.
- Ensure the cps.pem file is present in both `/home/cps/` and orchestrator container `/data/keystore/`.
- Execute the **network refresh-overlay-config true** CLI command before starting the network migration to back up the existing overlay-scripts folder and re-create the latest files.

Procedure

Step 1 Login to the Global Configuration mode.

Step 2 Enter the **network migrate-to-overlay true** to enable the Docker Overlay.

```
admin@orchestrator[site3-dra-master0]# network migrate-to-overlay true
```

Step 3 *Optional:* Run the **network consul-cleanup true** CLI command for recovery, if the network migration is stuck with the consul module.

Step 4 Verify if the system is healthy and run the **network detach-weave true** command to detach the Weave network, once you migrate to Docker Overlay network.

```
admin@orchestrator[site3-dra-master0]# network detach-weave true
```

Configure Weave Network

This configuration allows to enable the Weave network and detach the Overlay network in vDRA.

Before you begin

The prerequsites remain the same for configuring Overlay and configuring Weave network.

- Verify that the system and all container services are fully operational and healthy.
- Ensure the cps.pem file is present in both `/home/cps/` and orchestrator container `/data/keystore/`.
- Execute the **network refresh-overlay-config true** CLI command before starting the network migration to back up the existing overlay-scripts folder and re-create the latest files.

Procedure

Step 1 Login to the Global Configuration mode.

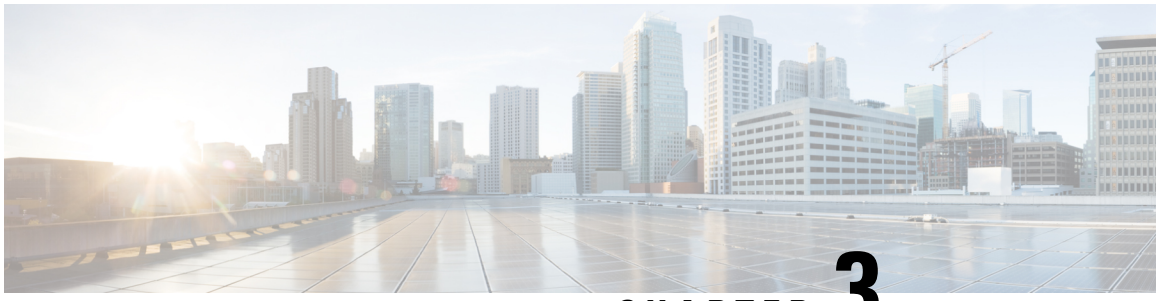
Step 2 Enter the **network migrate-to-weave true** to enable the Weave network.

```
admin@orchestrator[site3-dra-master0]# network migrate-to-weave true
```

Step 3 *Optional:* Run the **network consul-cleanup true** CLI command for recovery, if the network migration is stuck with the consul module.

Step 4 Verify if the system is healthy and run the **network detach-overlay true** command to detach the Overlay network, once you migrate to Weave network.

```
admin@orchestrator[site3-dra-master0]# network detach-overlay true
```



CHAPTER 3

Policy Builder Configuration

Plug-in Configuration

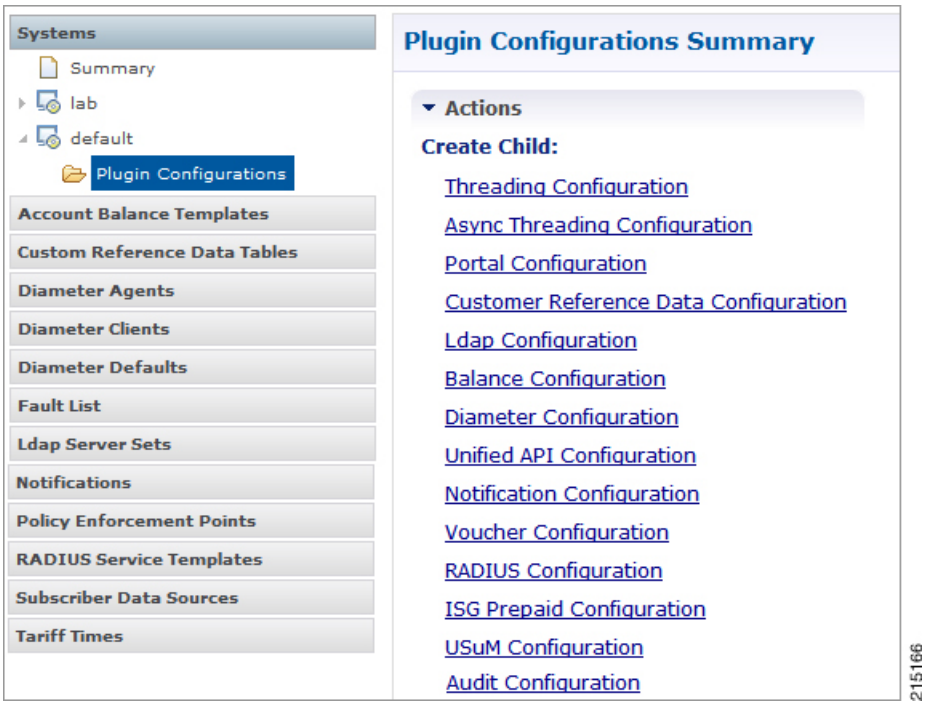
Cisco Policy Builder provides core plug-ins for customizing and optimizing your installation.

- 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 46: Create Child Action



Threading Configuration

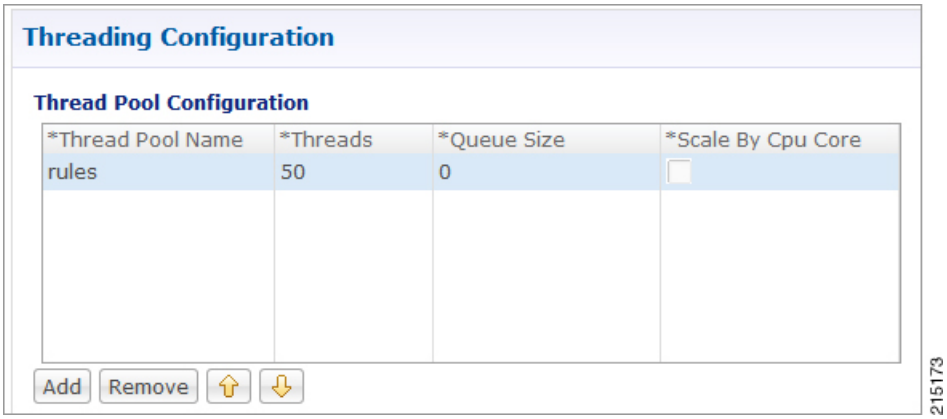
A threading configuration utility is provided for advanced users.

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.

A configuration example is shown below:

Figure 47: Thread Pool Configuration



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.

Parameter	Description
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.
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.

Important When you add new fields in CRD, manually update the new fields with appropriate values for all the existing entries in CRD. Otherwise DRA doesn't show any values for these new fields for existing entries and this can cause routing failures.

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.
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.

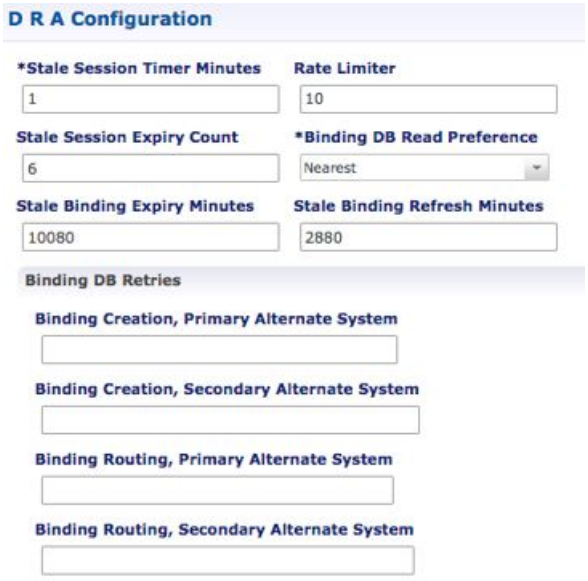
Parameter	Description
Avp Persists	Use this table to configure certain AVPs that you want to store in the session database. AVPs that are not configured as part of this table, are not persisted. • Name: Enter the name for the AVP value. • Avp Name: The name of the CRD/policy derived AVP. To retrieve the stored AVPs from the session, use the Customer Reference Data Debug AVPs. This retriever is used to send the stored AVPs in any diameter message, and available in the **PolicyState/Session data to Custom AVP Mapping** under Custom AVP Profiles. **Restriction** When you configure the AVP Persists table in the Policy Builder, for each AVP, configure both the AVP name and name. If no values are added for these fields, then the particular AVP is not added to the Gx session. This scenario leads to unavailability of the specific AVP and hence, no custom AVP are sent.

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 48: 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 **Note** Once session becomes stale and crosses configured Stale Session Timer Minutes, vDRA generates audit RAR for that session. If there is no audit RAR or the result code in RAA is other than 5002/2001, stale session expiry count gets decremented by one and the same is updated in session database. vDRA performs this operation until stale session expiry count reaches zero. Once stale session expiry count reaches zero, session is deleted.
Rate Limiter	Indicates the number of audit RARs per second that should be sent out by CPS vDRA. Rate Limter value is per worker value. Total number of audit RAR processed is calculated as Rate Limiter value * number of workers. **Note** • If primary database is Mongo Shard DB, then rate limiter value should be set as follows: The value to be set in the Rate Limit would be = 1000 • If primary database is Application Shard DB, then rate limiter value should be set as follows: The value to be set in the Rate Limit would be = 1000/No. of workers 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) For more information about binding DB audits and stale records, see Binding DB Audit, on page 58.
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. **Note** The primary system and the current vDRA system must share a common session database.
Binding Creation, Secondary Alternative System	Name of secondary vDRA system to retry Gx CCR-i **Note** The secondary system and the current vDRA must share a common session database.
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 65.
DRA Outbound Endpoints	Refer to DRA Outbound Endpoints, on page 67.
Relay Endpoints	Refer to Relay Endpoints, on page 73.

Settings

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

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

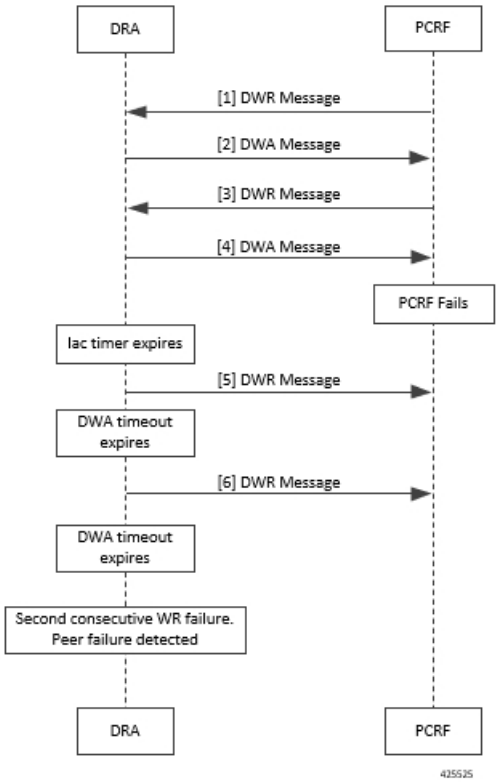
Table 5: DRA Configuration - Settings Parameters

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)
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)

Parameter	Description
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)
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.
Response Timeout Ms	Response timeout in milliseconds. Default: 1700 ms

The following figure illustrates the timers in peer detection:

Figure 49: vDRA Peer Detection Failure



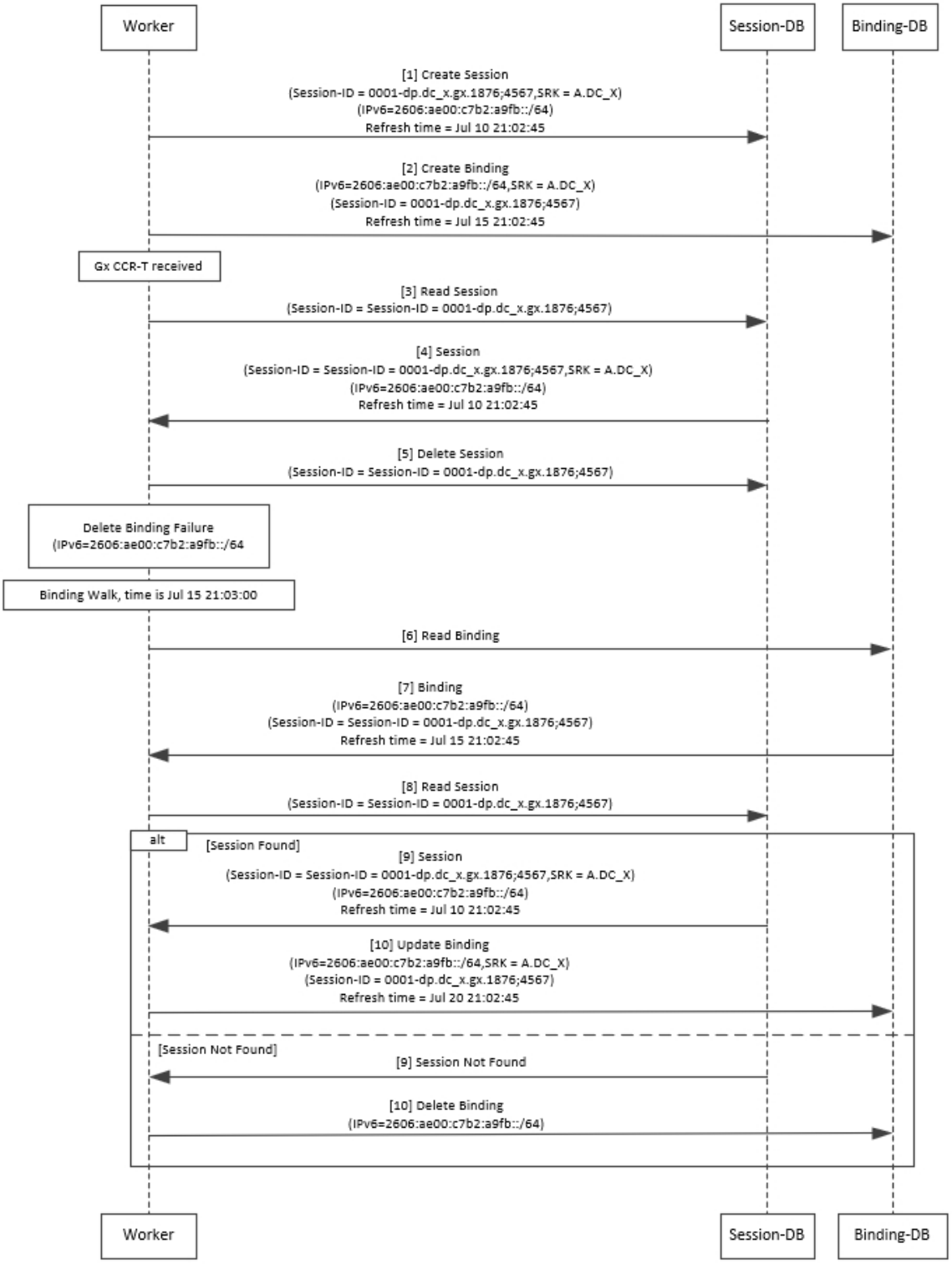
Binding DB Audit

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 50: DRA Binding Audit, Stale Binding Cleanup



Note There is a housekeeping thread to process stale sessions/bindings which does the following tasks in sequential order:

1. Process Stale Session Expiration: Generate Audit RAR OR delete the session if stale session expiry count has reached 0.
2. Process expiration of binding: Remove the bindings for which there is no corresponding session.

The stale session expiry task is scheduled to run every minute. This means that the stale session expiry processing is not guaranteed to happen exactly at the configured stale session expiry minutes interval. The stale session expiry processing can happen at any time within the configured stale session expiry minutes to configured stale session expiry minutes + 1 min interval.

However, if the previous task execution of the above mentioned three points takes longer time to complete due to large number of stale sessions/stale bindings, the stale session expiry would run post the previous task completion which can lead to a longer delay than expected 1 minute.

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.
Update Time Stamp On Success C C R U	When this check box is selected, session timestamp will be updated on receipt of success CCR-U (Result-Code: 2001) from PCEF. [2] Default is unchecked (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.
Enable Proxy Bit Validation	Enables P bit validation. vDRA validates the P bit in the Diameter request and, if set, the message maybe proxied, relayed, or redirected. If this option is disabled, the P bit in the request is not checked and the request is not considered proxiable. Default: Enabled.

Parameter	Description
Enable Mediation	Enable advanced mediation capabilities in both egress and ingress direction. This feature allows you to configure vDRA to change the value of the Result-Code in Diameter Answer, use mediation to hide topology, prepend label to Destination Host AVP, etc.
Enable Doic	Enable or disable abatement action for Diameter requests towards PCRF, HSS, AAA, and OCS servers based on reporting of overloaded conditions using the architecture described in RFC 7683 Diameter Overload Indication Conveyance (DOIC). DOIC can be enabled/disabled at peer group level in Peer Group SRK Mapping table. If the destination peer is congested or overloaded, you can choose to either forward, divert, or drop messages.
Enable PCRF Session Query	Enables or disables the PCRF session query. If you enable this, Policy DRA then supports a fallback routing for Rx AARs for VoLTE using the PCRF session query. This ensures that VoLTE calls can complete in the event that IPv6 binding is not found in the binding database. For an Rx AAR with an IPv6 binding query, vDRA provides the ability to route the Rx AAR based on an API query to the PCRF to determine if it has a session for the IPv6. The queries can be made in parallel to a configured set of query points on PCRFs. The Framed-IPv6 AVP from the Rx must be provided in the request to the PCRF. PCRF returns an SRK to be used for routing, similar to existing binding lookups.
Create IPv6 Bindings based on PCRF Session Query	Enables creation of IPv6 binding record in the database based on PCRF session query. When PCRF session query result (success) is received and if IPv6 record is not present in the database, vDRA creates an IPv6 binding record based on the response from the PCRF. If any CCR-I is received for the same IPv6 record, then it overwrites the IPv6 binding record. For any CCR-T, vDRA deletes the IPv6 binding record from database. **Note** Ensure you also enable PCRF Session Query for this feature to work. The Stale Binding Expiry and Refresh Minutes are used to clear these binding records from the database. For more information, see Binding DB Audit, on page 58.

Parameter	Description
Enable Best Effort Binding	When selected allows the operator to enable the best effort binding creation configuration on a per APN basis. The configuration is enabled on a per APN basis and controls any or all of the following bindings (for best effort): • IPv6 • IPv4 • MSISDN/APN • IMSI/APN • Session Default is unchecked. Best effort bindings are those bindings for which DRA does not wait for DB write operations to be completed. DRA forwards the CCR without waiting for DB write and there is an asynchronous write call for best effort bindings. If there is no matching APN found in the best effort binding table from CCR-I, DRA takes the legacy behavior and treats all bindings as mandatory. The bindings to be created is primarily decided by binding creation profile and then DRA examines the best effort table to find the best effort and mandatory bindings. The session can be marked as best effort and in such cases session is not created if session Db is down but the CCR is forwarded.
Slf Max Bulk Provisioning TPS	Rate at which subscribers are provisioned in the SLF database. SLF bulk provisioning generates high number of database write operations in a short duration of time. To spread out the operations over a period of time and mitigate the performance issue, configure the TPS. The rate limit adds delay between transactions and thereby limits the number of transactions executed per second. For more information about SLF bulk provisioning, see the *CPS vDRA Operations Guide*.
A A R Priority Processing	In vDRA 19.4.0 and later release, this parameter has been deprecated and no longer supported. By default, when application-based client sharding is used, AAR processing is prioritized on workers.

[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.

[2] The time stamp is updated on generation of Stale RAR. Also, if a success CCR(U)/CAA(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**:

Note To handle loads of 15 K TPS or more, create multiple TCP connections with PCRF and apply the same configuration to all DRA Directors.

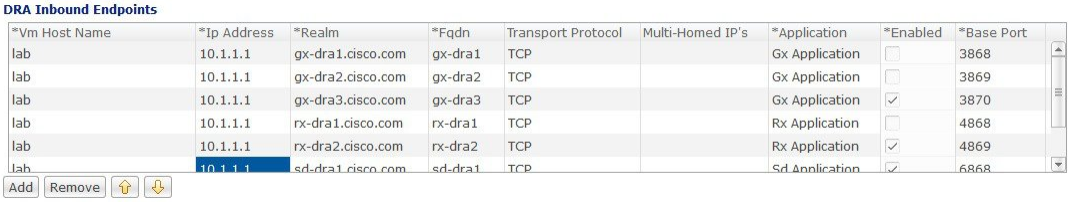
Table 9: DRA Configuration - DRA Inbound Endpoints Parameters

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.
Transport Protocol	Allows you to select either 'TLS', 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. **TLS**: Enables the connection as TLS from inbound . The supported TLS version is 1.2 and only for Rx application it is supported.

Parameter	Description
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. **Note** Both IPv4 and IPv6 are supported in vDRA endpoint configuration. For IPv6, you can enter either short or long format. 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 51: DRA Inbound Endpoints - Example Configuration



DRA Outbound Endpoints

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

Table 10: DRA Configuration - DRA Outbound 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.
Transport Protocol	Allows you to select either 'TCP' or 'SCTP' for the selected CPS vDRA 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.

Parameter	Description
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. **Note** Both IPv4 and IPv6 are supported in vDRA endpoint configuration. For IPv6, you can enter either short or long format. 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.
Enabled	Check to enable the endpoint.
Peer Realm	Diameter server realm.
Peer Host	Diameter server host. By default, the connection is initiated on the standard diameter port (3868). If a different port needs to be used than the peer name must be defined using the host:port format.

An example configuration is shown below:

Figure 52: DRA Outbound Endpoints - Example Configuration

DRA Outbound Endpoints

*Vm Host Name	*Ip Address	*Realm	*Fqdn	Transport Protocol	Multi-Homed IP's	*Application	*Enabled	*Peer Realm	*Peer Host
lab	10.1.1.1	gx-dra1.cisco.com	gx-dra9	TCP		Gx Application	✓	pcrf2-gx2.cisco.com	gx-pcrf
lab	10.1.1.1	rx-dra1.cisco.com	rx-dra9	TCP		Rx Application	✓	rx-pcrf.cisco.com	rx-pcrf:4868

Add Remove

Enable TLS and MTLS for Diameter Encryption

RFC 6733 Protocol Model

According to the RFC 6733 protocol model, you can configure the security details to initialize the TLS or MTLS connection.

Figure 53: Security Handshake for TLS Connection

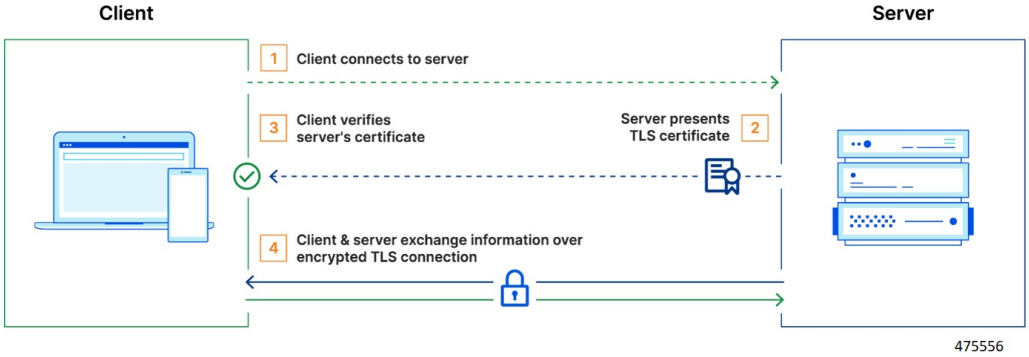
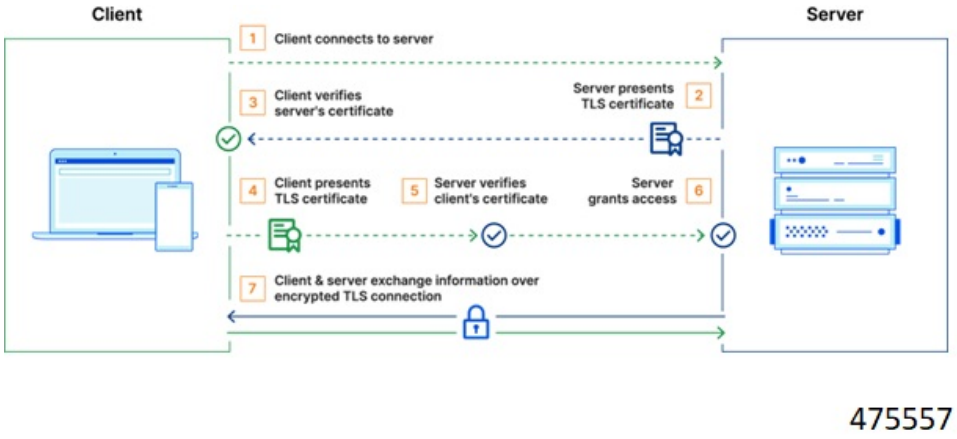


Figure 54: Security Handshake for MTLS Connection



The sequence for the data transmission is as follows:

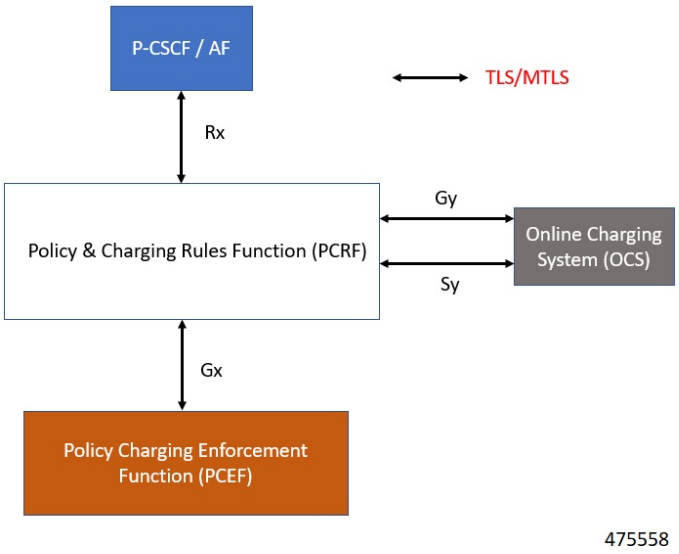
- Establishes TCP Connection
- Establishes TLS or MTLS connection over TCP.
- Exchanges CER/CEA message between the peers over TLS or MTLS.

- Exchanges application data over TLS or MTLS.

Feature Description

The vDRA supports a Transport Layer Security (TLS) and MTLS (Mutual Transport Layer Security) secure channels for diameter peer connection. The following architecture describes TLS and MTLS in DRA.

Figure 55: TLS/MTLS in DRA



Enabling TLS Protocol in the Policy Builder

Use the Policy Builder to enable the TLS protocol.

1. Log in to the Policy builder.
2. In the **DRA Inbound Endpoint**, from the **Transport Protocol** drop-down list, choose **TLS** to enable a connection as TLS from inbound. The supported version of TLS is 1.2 and it supports the Gx, Rx, Gy and Sy application.

   You can publish the configuration after providing necessary stack details.

Enabling MTLS in Policy Builder

The MTLS is configurable in the Policy Builder GUI.

1. Log in to the Policy builder.
2. In the **DRA Inbound Endpoint**, from the **Transport Protocol** drop-down list, choose **MTLS** to enable a connection as MTLS from inbound. The supported version of MTLS is 1.2 and it supports the Gx, Rx, Gy and Sy application.

An example configuration is shown below:

Figure 56: DRA Inbound Endpoints - Example Configuration

Importing Certificate through CLI

Prerequistes: Ensure that a cps.pem file is present in `/data/keystore` in the orchestrator container before executing the CLI.

Follow the steps to import certificates through CLI:

1. Copy the certificates files to the master VM under `/data/orchestrator/pemKey` to import the *tls* certificate.
2. Load the certificates to the Diameter application using the following CLI command

   `dra-tls cert import` *`certificate file private file`*

   a. Input certificate and private files for the CLI command.

   b. Enter the keystore password to encrypt the certificate file. Backend script converts files into JKS with encryption and copies to the diameter-endpoint containers.

Note

- Ensure to enter a Password with minimum of six characters, Alphanumeric, and special characters.
- Renegotiation of TLS Handshake for an established connection with the new certificate from the server side [Diameter] without any call failures are not supported.

Example 1:

dra-tls cert import certificate.pem private.pem

admin@orchestrator[pn-master-0]# ***dra-tls cert import tls-cert.pem private.pem***

*enter the Keystore Password for this private.pem cert:*********

Importing keystore /data/pemKey/certificate-tls.p12 to /data/pemKey/diameter-endpoint-tls.jks...

Example 2

admin@orchestrator[pn-master-0]# ***dra-tls cert import CA-cert.pem CA-key.pem***

*enter the Keystore Password for this private.pem cert:********

Importing keystore /data/pemKey/certificate-tls.p12 to /data/pemKey/diameter-endpoint-tls.jks...

Warning:

The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore /data/pemKey/diameter-endpoint-tls.jks -destkeystore /data/pemKey/diameter-endpoint-tls.jks -deststoretype pkcs12".

Import Successfully Completed for 192.1.XX.XX.

Importing Certificate to 192.1.XX.XX.

Import Successfully Completed for 192.1.XX.XX.

Importing Certificate to 192.1.XX.XX.

Import Successfully Completed for 192.1.XX.XX.

Importing Certificate to mongo-admin-a:27017 Database.

Certificate Imported Successfully.

Note

- The copy of the generated jks file will be maintained in Mongo Admin DB for high availability during VMDK upgrade.
- You can use the complex password that includes alpha numeric and special characters for generating JKS via CLI command [Supported Special Characters è ***! @ # $ % . , ^ & ' * "***].

Creating TLS Certificate Before Expiration and Raising Alerts

vDRA supports the following function:

- Installation of a new certificate on the Directors before expiration of a TLS certificate

Note After installation, the same certificate must be installed on the client.

- After replacing a new certificate, the client initiates reestablishment of connections within the maintenance window to avoid call failures.
- Monitoring the certificate validity will be every one hour, from the time of application restart.
- Alert notification prior to the certificate expiration date based on the following alert notification metrics.

Table 11: Alert Notification

Expiration in Days	Alert Level
60 days	Minor
40 days	Major

Expiration in Days	Alert Level
14 days	Critical

For more information, see the *Application Notifications* table and *Alert Rules* section in the *CPS vDRA SNMP and Alarms Guide*.

Inservice Certificate Management

Ensure to follow the procedure to install a new TLS certificate on the Director before the TLS certificate expiration:

- Place the new certificate in the following path `/data/orchestrator/pemKey/`.
- After placing a new updated certificate on the Master VM, use the same CLI command to replace the existing certificate.

  The existing connection from the older certificate remains connected and there should not be any call failure.
- To get the new certificate in place, terminate the existing connection and the new connection must be negotiated by the client.

Relay Endpoints

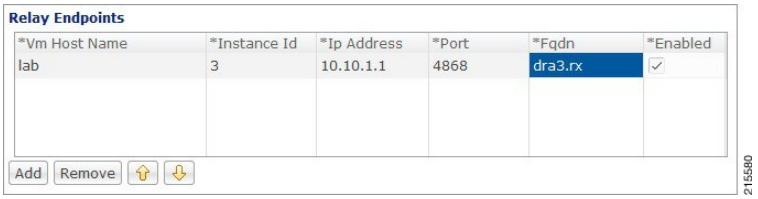
The following parameters can be configured under **Relay Endpoints**:

Table 12: 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. **Note** The relay endpoints must be configured on physical IPs and not on virtual IPs.
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 57: Relay Endpoints - Example Configuration



Policy Routing for Real IPs with Relay Endpoints

vDRA relay links consist of a control plane and a data plane.

The control plane uses virtual IPs and the data plane uses real IPs.

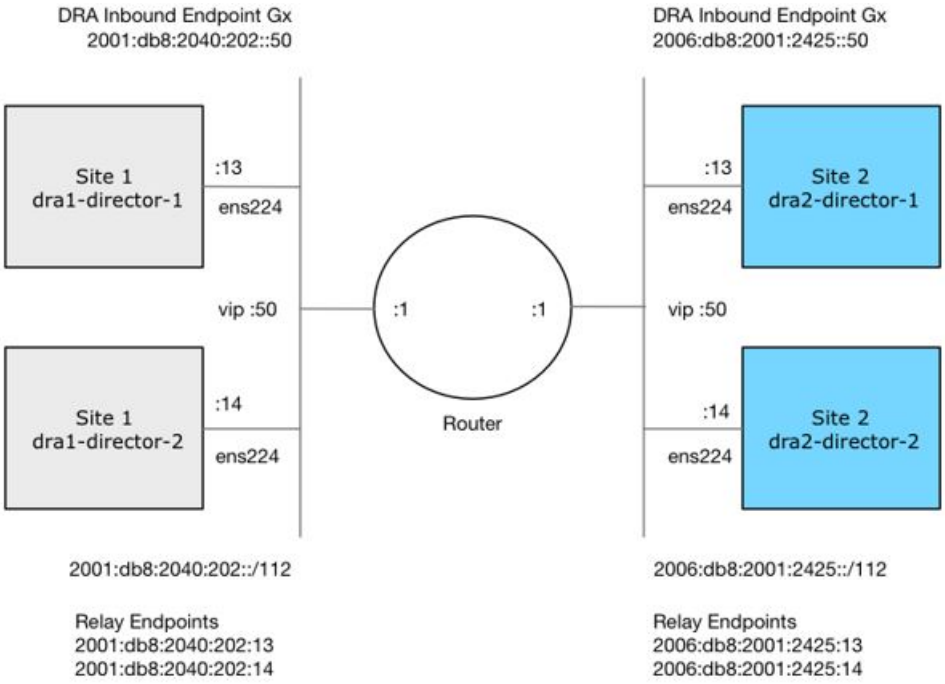
If the control and data plane use the same links, and those links are configured with VIPs, by default, the data plane uses the VIP as its source address for outgoing connections. The data plane uses the VIP as the source address only if the VIP is active on the data plane's outgoing interface.

To avoid this situation, policy routing is used to force the data plane to use the real IP address of the outgoing interface instead of the VIP.

Example of a vDRA Relay Endpoints

In the following example network, only the DRA director VMs and their relay links are displayed. In a real scenario, many more links may exist on the DRA director VMs.

Figure 58: Example of Relay Endpoints



Policy Routing

Linux policy routing includes rules and routing tables. The rules identify traffic and point to a user-defined routing table. The routing table contains customized routes.

To prevent the Relay Link's data plane from using the VIP as a source address, a rule is created to identify the real IP in the destination address and identify the desired routing table.

Configure Policy Routing

The following configuration procedure is performed on Site 1 dra1-director-1. Repeat the procedure for all other dra-directors and modify the IP addresses accordingly.

Perform the following steps on each dra-director VM to configure policy routing:

1. Create a custom routing table
2. Create an IP rule for each remote relay endpoint's real IP address
3. Add a route to the custom routing table that specifies the real IP source address

Set up Custom Routing Table

Set up the custom routing table as shown in the following example:

```
echo "200 dra.relay" | sudo tee --append /etc/iproute2/rt_tables
```

Define IP Rules

The following rules match the packets destined to the real IPs of interface ens224 on dra2-director1 and dra2-director2:

```
ip -6 rule add to 2006:db8:2001:2425::13 table dra.relay
ip -6 rule add to 2006:db8:2001:2425::14 table dra.relay
```

Define the Route

The following example of the route uses the router's interface as the next hop and specifies ens224's real IP address as the source address for outgoing packets.

```
ip route add 2006:db8:2001:2425::/112 via
2001:db8:2040:202::1 src 2001:db8:2040:202::13 table dra.relay
```

Validate the Routing

Use the following example commands to validate the route selection for remote relay real IP and VIP addresses.

```
ip -6 route show table dra.relay
ip -6 route get 2006:db8:2001:2425::13
ip -6 route get 2006:db8:2001:2425::14
ip -6 route get 2006:db8:2001:2425::50
```

Persistent Configuration

In order for the Policy Routing configuration to survive a reboot, add the configuration commands to `/etc/network/interfaces` under interface ens224 as shown below:

```
auto ens224
iface ens224 inet static
address 192.169.22.13
netmask 255.255.255.0
iface ens224 inet6 static
```

```
address 2001:db8:2040:202::13
netmask 112
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
up ip -6 rule add to 2006:db8:2001:2425::13 table dra.relay
up ip -6 rule add to 2006:db8:2001:2425::14 table dra.relay
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src 2001:
db8:2040:202::13 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
down ip -6 rule del to 2006:db8:2001:2425::13 table dra.relay
down ip -6 rule del to 2006:db8:2001:2425::14 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src
2001:db8:2040:202::13 table dra.relay
```

Configure Policy Routing with Deployer/Installer

Configure the VM artifacts and the cloud config to set up policy routing using the deployer.

VM Artifacts

Add Policy Route configuration to the DRA director VM's interfaces.esxi file as shown in the following example:

```
cps@installer:/data/deployer/envs/dra-vnf/vms/dra-director
/dra-director-1$ cat interfaces.esxi
auto lo
iface lo inet loopback

auto ens160
iface ens160 inet static
address 10.81.70.191
netmask 255.255.255.0
gateway 10.81.70.1

auto ens192
iface ens192 inet static
address 192.169.21.13
netmask 255.255.255.0

auto ens224
iface ens224 inet static
address 192.169.22.13
netmask 255.255.255.0
iface ens224 inet6 static
address 2001:db8:2040:202::13
netmask 112
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
up ip -6 rule add to 2006:db8:2001:2425::13 table dra.relay
up ip -6 rule add to 2006:db8:2001:2425::14 table dra.relay
up ip route add 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src
2001:db8:2040:202::13 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1
down ip -6 rule del to 2006:db8:2001:2425::13 table dra.relay
down ip -6 rule del to 2006:db8:2001:2425::14 table dra.relay
down ip route del 2006:db8:2001:2425::/112 via 2001:db8:2040:202::1 src
2001:db8:2040:202::13 table dra.relay

auto ens256
iface ens256 inet static
address 192.169.23.13
netmask 255.255.255.0
cps@installer:/data/deployer/envs/dra-vnf/vms/dra-director/dra-director-1$
```

Cloud Config

Create the dra.relay routing table on the dra-directors by adding the following bootcmd: to user_data.yml and storing the file at `/data/deployer/envs/dra-vnf/vms/dra-director/user_data.yml`. The sed command prevents adding a routing table every time the VM boots.

```
bootcmd:
 - "sed -i -e '/^200 *dra.relay/d' /etc/iproute2/rt_tables"
 - "sh -c \"echo '200     dra.relay' >> /etc/iproute2/rt_tables\""
```

Example of user_data.yml:

```
#cloud-config
debug: True
output: {all: '| tee -a /var/log/cloud-init-output.log'}

users:
  - name: cps
    sudo: ['ALL=(ALL) NOPASSWD:ALL']
    groups: docker
    ssh-authorized-keys:
      - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDzjJjndIvUiBta4VSIbd2gJmlMWcQ8wtejgAbi
XtoFZdtMdo9G0ZDEOtxHNNDPwWujMiYAkZhZWX/zON9raavU8lgD9+YcRopWUtujIC71YjtoxIjWIBBbrtqt
PlUXMUXQsi91RQbUtslENP+tSatS3awoQupyBMMSutyBady/7Wq0UTwFsnYs5Jfs8jIQuMfVQ9uJ4mNn7wJ0
N+Iaf27rE0t3oiY5DRN6j07WhauM6lCnZ1JDlzqmTnTHQkgJ3uKmQa5x73tJ1OW89Whf+R+dfslVn/yUwK/
vf4extHTn32Dtsxkjz7kQeEDgCe/y7owimaEFcCIfEWEaj/50jegN cps@root-public-key

resize_rootfs: true

write_files:
  - path: /root/swarm.json
    content: |
     {
        "role": "{{ ROLE }}",
        "identifier": "{{ IDENTIFIER }}",
        "master": "{{ MASTER_IP }}",
        "network": "{{ INTERNAL_NETWORK }}",
        {% if WEAVE_PASSWORD is defined %}"weavePw": "{{ WEAVE_PASSWORD }}",
        {% endif %}
        "zing": "{{ RUN_ZING | default(1) }}",
        "cluster_id": "{{ CLUSTER_ID }}",
        "system_id": "{{ SYSTEM_ID }}"
     }
    owner: root:root
    permissions: '0644'
  - path: /home/cps/.bash_aliases
    encoding: text/plain
    content: |
      # A convenient shortcut to get to the Orchestrator CLI
      alias cli="ssh -p 2024 admin@localhost"
      alias pem="wget --quiet http://171.70.34.121/microservices/latest/cps.pem ;
      chmod 400
cps.pem ; echo 'Retrieved \"cps.pem\" key file'"
    owner: cps:cps
    permissions: '0644'
  - path: /etc/pam.d/common-password
    content: |
     #
     # /etc/pam.d/common-password - password-related modules common to all services
     #
     # This file is included from other service-specific PAM config files,
     # and should contain a list of modules that define the services to be
     # used to change user passwords.  The default is pam_unix.

     # Explanation of pam_unix options:
```

```
     #
     # The "sha512" option enables salted SHA512 passwords.  Without this option,
     # the default is Unix crypt.  Prior releases used the option "md5".
     #
     # The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
     # login.defs.
     #
     # See the pam_unix manpage for other options.

     # As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
     # To take advantage of this, it is recommended that you configure any
     # local modules either before or after the default block, and use
     # pam-auth-update to manage selection of other modules.  See
     # pam-auth-update(8) for details.

     # here are the per-package modules (the "Primary" block)
     password   requisite                      pam_pwquality.so retry=3 minlen=8
     minclass=2
     password   [success=2 default=ignore]      pam_unix.so obscure use_authtok
     try_first_pass sha512 remember=5
     password   sufficient                     pam_sss.so use_authtok
     # here's the fallback if no module succeeds
     password   requisite                      pam_deny.so
     # prime the stack with a positive return value if there isn't one already;
     # this avoids us returning an error just because nothing sets a success code
     # since the modules above will each just jump around
     password   required                       pam_permit.so
     # and here are more per-package modules (the "Additional" block)
     # end of pam-auth-update config
    owner: root:root
    permissions: '0644'
runcmd:
 - [vmware-toolbox-cmd, timesync, enable ]
bootcmd:
 - "sed -i -e '/^200 *dra.relay/d' /etc/iproute2/rt_tables"
 - "sh -c \"echo '200     dra.relay' >> /etc/iproute2/rt_tables\""
```

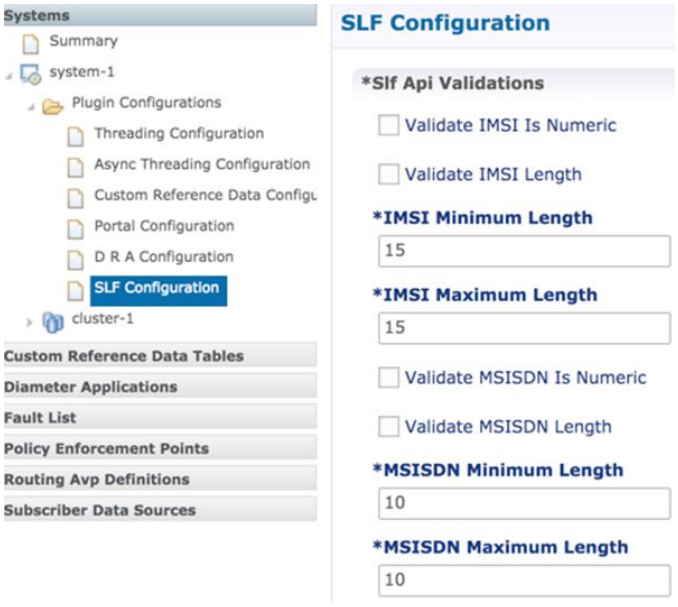
SLF Configuration

You can specify whether the IMSI and MSISDN values are validated in SLF API.

By default, SLF validation is disabled.

To set up SLF validation, create SLF Configuration from the Plugin Configuration in Policy Builder.

Figure 59: SLF Configuration



The following table describes the SLF API validations that you can configure:

Table 13: SLF Configuration

Field	Description
Validate IMSI is Numeric	If checked: IMSI received in the SLF API request must be numeric If unchecked: IMSI numeric validation is not performed on the IMSI received in the SLF API request
Validate IMSI Length	If checked: IMSI length is validated based on the specified IMSI Minimum Length (inclusive) and IMSI Maximum Length (inclusive) If unchecked: IMSI length validation is not performed on the IMSI received in the SLF API request
Validate MSISDN is Numeric	If checked: MSISDN received in the SLF API request must be numeric If unchecked: MSISDN numeric validation is not performed on the MSISDN received in the SLF API request

Field	Description
Validate MSISDN Length	If checked: MSISDN length is validated based on the specified MSISDN Minimum Length (inclusive) and MSISDN Maximum Length (inclusive) If unchecked: MSISDN length validation is not performed on the MSISDN received in the SLF API request

Ingress and Egress API Rate limit Configuration

Feature Description

The vDRA uses PCRF session query to query SRK from PCRF to route the request and then recreates the binding entry. There is no rate limit for a PCRF session query triggered from vDRA. Similarly, Ingress APIs (Binding/Session/SLF/CRD/SVN/Topology/Grafana/Promethus) does not have an overload protection mechanism.

In the CPS 22.1.0 and later releases, vDRA supports a configurable option to rate-limit the incoming traffic and outgoing traffic on the API interface at director level. This rate limiting process protects the system when acting as a client or server. Also, to prevent any back pressure and working on stale messages, vDRA supports configurable queue size and length message SLAs.

Egress API Rate Limiting

vDRA supports PCRF Session Query API rate limits at director level because applying rate limit at worker level can cause uneven distribution of rate limit across Workers.

For example, possibilities of same workers receiving all Rx AAR messages that need PCRF session query, and vDRA can apply rate limit only for that worker. This causes Rx AAR to for that worker even though remaining workers are under rate limit. To avoid this issue, vDRA supports rate limit configurations at the director level.

Note By default, rate limit is not configured for egress API.

The functions of egress rate limiting are:

- The Director triggers PCRF session query based on the configured rate limit. For example, ff configured rate limit is 50, then director allows only first 50 Rx AAR requests per second to trigger PCRF session query and remaining requests are dropped. vDRA sends Rx AAA for dropped PCRF Session query with error message as “PCRF Session Query Throttled”. vDRA maintains internal error code as “027”.
- If PCRF session query gets triggered due to “No Binding Found” error and PCRF session query got rate limited, then vDRA returns an error message:

```
“4006:027 – PCRF Session Query Throttled”
```

- If PCRF session query gets triggered due to “Binding DB Error” error and PCRF session query got rate limited, then vDRA returns error message:

```
"4007:027 – PCRF Session Query Throttled"
```

Ingress API Rate limiting

Following are the categories of Ingress APIs for which you can set rate limits:

- Binding API
- SLF API
- Topology API (Peer/Relay connections)
- OAM API(CRD/PB/CustRefData/Grafana/Promethues/SVN)

The functions of ingress Rate Limiting are:

- Ingress API is rate limited in HAProxy service.
- In vDRA, haproxy-common running in **master/control-0/control-1/directors** is used for load balancing of Policy Builder, Grafana, UI, CC, a so on. The haproxy-common receives request from client and forwards the request to vDRA backend servers.
- Ingress requests reaching haproxy-common is tracked in stick-table with server destination IP as key.
- In frontend, stick-table entries get compared with configured rate limit for respective ingress API. If the stick-table entries are greater than configured rate limit, then HAProxy sends HTTP deny status to the client. Otherwise, vDRA processes the request and send success status to client.
- vDRA returns error code 429 as deny status to the client for all the failed requests due to rate limit.
- Set the rate limit. For example:
  - If you want to set rate limit as 100 and the clients are configured to send requests only to haproxy-common running in master, then set rate limit as 100.
  - If the clients are configured to send requests to haproxy-common running master/control-0, then rate limit should be set as 50. So that two HAProxy running in master/control-0 provides 100 TPS.
  - In DRA, to make sure that DRA reaches the configured rate limit, additional 25 per cent is added to configured rate limit. This is mainly to get approximate rate limit in DRA. For example, If a rate limit is set as 500, then DRA internally adds extra 25 per cent to the configured rate limit 500 and the rate limit is set at 625. Thus, DRA allows requests 500–625.

Sample HAProxy configuration to rate limit ingress API:

```
frontend https_all_servers
       description Unified API,CC,PB,Grafana,CRD-API,PB-API,Promethues
       bind :443

      #ACL for Unified Binding IMSI-APN API
      acl binding_api_imsi_apn path_beg  /dra/api/bindings/imsiApn
/dra/api/deleteBinding/imsiApn
      http-request deny deny_status 429 if binding_api_imsi_apn {
dst,table_http_req_rate(binding_api_imsi_apn_servers) gt 625 }
      use_backend binding_api_imsi_apn_servers if binding_api_imsi_apn

       backend binding_api_imsi_apn_servers
       mode http
       balance source
```

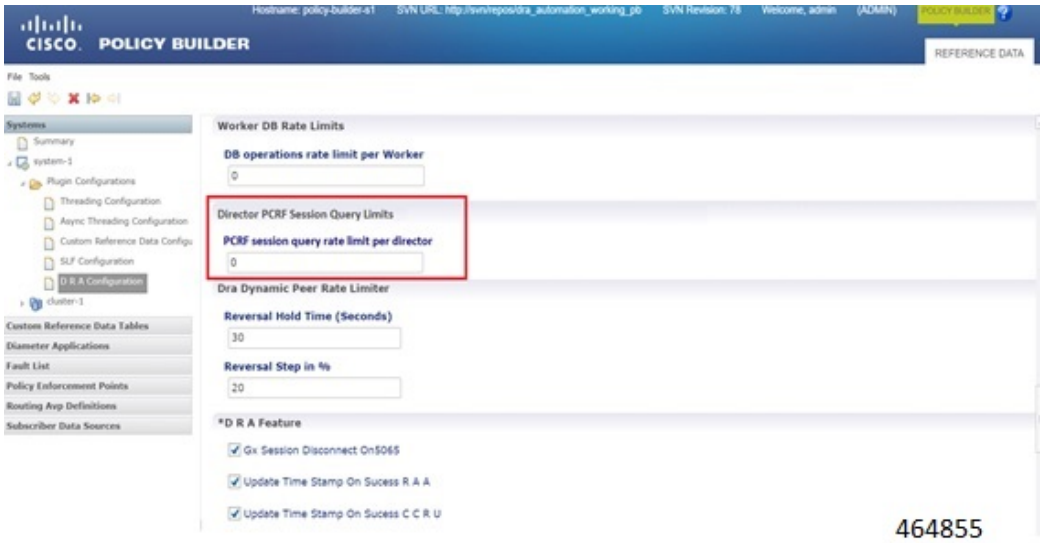
```
        option httpclose
        option abortonclose
        stick-table type ip size 1m expire 1s store http_req_rate(1s)
        http-request track-sc1 dst table binding_api_imsi_apn_servers
        server haproxy-api-s101 haproxy-api-s101:80 check inter 10s resolvers dns
resolve-prefer ipv4

        acl authoriseReadonlyUsers http_auth_group(cps_user_list) qns-ro
        acl authoriseAdminUsers http_auth_group(cps_user_list) qns
       http-request auth realm CiscoApiAuth if !authoriseReadonlyUsers !authoriseAdminUsers

        http-request deny if !METH_GET authoriseReadonlyUsers
```

Configuring Egress API Rate Limit in the Policy Builder

You can configure egress API rate limit for PCRF Session Query per director in the DRA Configuration.

- In the **Policy Builder**, click **DRA Configuration** from the left pane to add the configuration in the system.

Figure 60: Director PCRF Session Query Limits



- Configure the following parameters under DRA Configuration:

Table 14: DRA Configuration Parameters

Parameter	Description
DB operations rate limit per Worker	Specifies that the rate limit is per worker for DB operations. **Default**: By default, the rate limit is in disabled state.
PCRF session query rate limit per director	Specifies that the rate limit is for PCRF session query at Director level. Make sure to select the Director PCRF Session Query Limits" in the Policy Builder to view "PCRF session query limits per director" field. **Default**: By default the rate limit is in disabled state.

Parameter	Description
Reversal Hold Time (Seconds)	Specifies the reversal hold time in seconds.
Reversal Step in %	Specifies the reverstal step in percentage.
Gx Session Disconnect on5065	By default, **Gx Session Disconnect 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 is always a database query to fetch Gx session id. This results in linear increase in database transactions 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. [3] Default is checked (recommended setting). **Important** When using this flag, there is always a database query to fetch Gx session id. This results in linear increase in database transactions with AAR traffic on Rx interface.
Update Time Stamp On Success C C R U	When this check box is selected, session timestamp will be updated on receipt of success CCR-U (Result-Code: 2001) from PCEF. [4] Default is unchecked (recommended setting). **Important** When using this flag, there is always a database query to fetch Gx session id. This results in linear increase in database transactions with AAR traffic on Rx interface.

[3] 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.
[4] The time stamp is updated on generation of Stale RAR. Also, if a success CCR(U)/CAA(2001) comes after generation of Stale RAR, then the Stale RAR counter is reset.

Configuring Ingress API Rate Limit

You can configure Ingress API rate limits to set the environment variables and use them for checking ingress or egress API rate limit in the *haproxy.cfg.tmpl* file. The CLI updates are applied only in haproxy-common containers because haproxy-common is used for load balancing of Policy Builder, Grafana, UI, API, CC, and so on.

After CLI updates the rate limit in haproxy config file in haproxy-common containers, haproxy is restarted automatically to apply new rate limits.

Note Since these CLIs internally applies the rate limit and restart haproxy, you need not manually restart haproxy-common in Master/Control/diameter containers after configuring new rate limits.

You can set common rate limit for all binding API using the CLI **dra set-ratelimit binding-api** *rate limit value*. vDRA provides options to override common rate limits for imsi, imsi-apn, msisdn, msisdn-apn, and ipv6 binding api by specifying binding type in CLI as follows:

dra set-ratelimit binding-api-imsi | binding-api-imsi-apn | binding-api-msisdn
| binding-api-msisdn-apn | binding-api-ipv6]*value*

By default, DRA does not apply any rate limit for ingress APIs.

Use the following CLI commnads to select different ingress API types to set, remove or show rate limits.

- dra set-ratelimit binding-api <rate limit value>
- dra set-ratelimit binding-api-imsi <rate limit value>
- dra set-ratelimit binding-api-imsi-apn <rate limit value>
- dra set-ratelimit binding-api-msisdn <rate limit value>
- dra set-ratelimit binding-api-msisdn-apn <rate limit value>
- dra set-ratelimit binding-api-ipv6 <rate limit value>
- dra set-ratelimit session-api <rate limit value>
- dra set-ratelimit slf-api <rate limit value>
- dra set-ratelimit topology-api <rate limit value>
- dra set-ratelimit oam-api <rate limit value>
- dra remove-ratelimit binding-api
- dra remove-ratelimit binding-api-imsi
- dra remove-ratelimit binding-api-imsi-apn
- dra remove-ratelimit binding-api-msisdn
- dra remove-ratelimit binding-api-msisdn-apn
- dra remove-ratelimit binding-api-ipv6
- dra remove-ratelimit session-api
- dra remove-ratelimit slf-api
- dra remove-ratelimit topology-api
- dra remove-ratelimit oam-api
- dra show-ratelimit
- dra show-ratelimit binding-api

- dra show-ratelimit binding-api-imsi
- dra show-ratelimit binding-api-imsi-apn
- dra show-ratelimit binding-api-msisdn
- dra show-ratelimit binding-api-msisdn-apn
- dra show-ratelimit binding-api-ipv6
- dra show-ratelimit slf-api
- dra show-ratelimit session-api
- dra show-ratelimit topology-api
- dra show-ratelimit oam-api

For more information, see the *CLI Commands* section in the *CPS vDRA Operations Guide*.

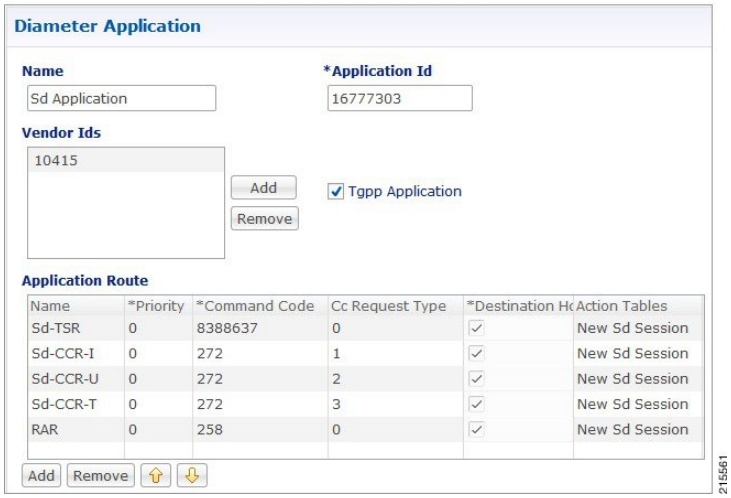
Diameter Application

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 61: Diameter Application - Sd Application Example



The following parameters are configured under Sd Application:

Table 15: Sd Application Parameters

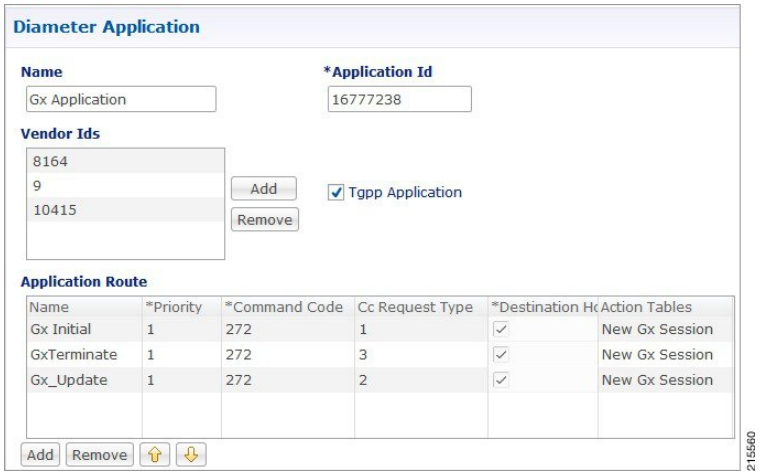
Parameter	Description
Name	Name of the Sd application.
Application Id	16777303, 3GPP specified Application Identifier for Sd interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Sd interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined 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 62: 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 16: Gx Application Parameters

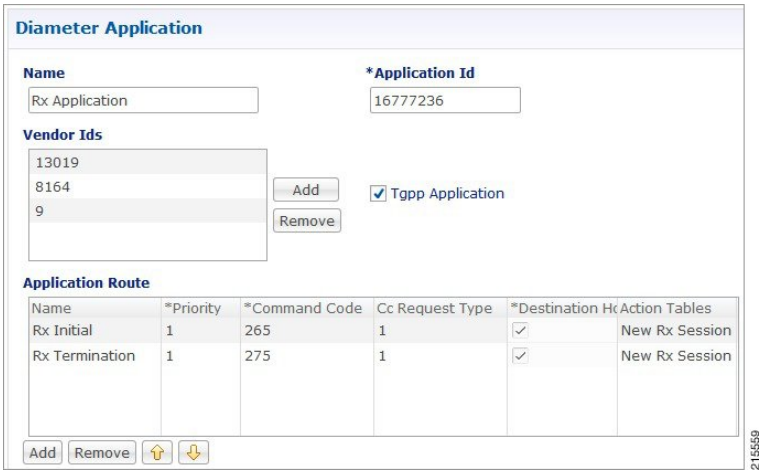
Parameter	Description
Name	Name of the Gx application.
Application Id	16777238, 3GPP specified Application Identifier for Gx interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Gx interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined 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.
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.

Parameter	Description
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 63: Diameter Application - Rx Application Example



The following parameters are configured under Rx Application:

Table 17: 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.
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.

Parameter	Description
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.

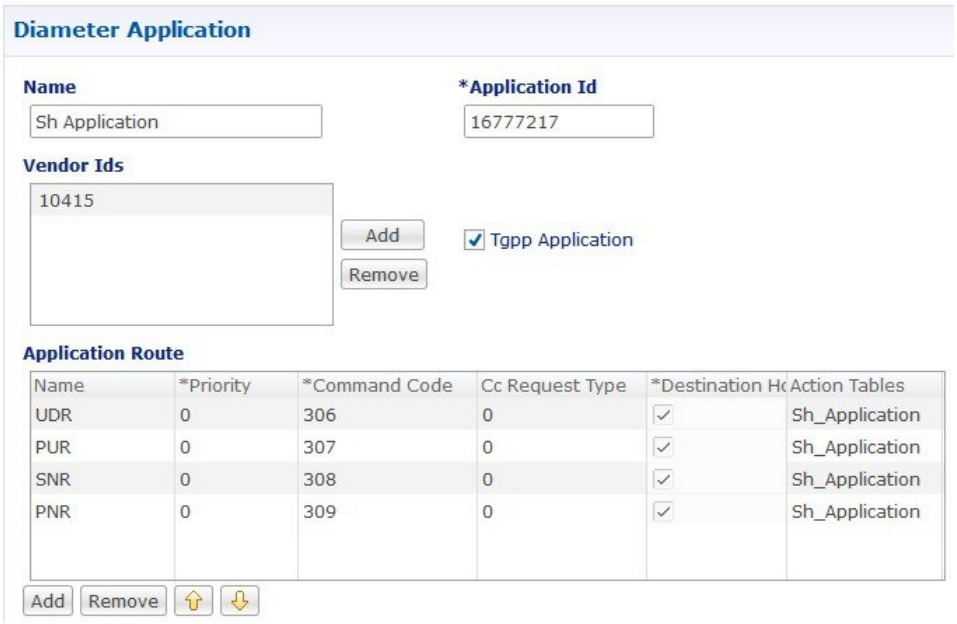
Sh Application

Sh interface is used for communication between AS and HSS for Call data query/Push subscriber profile and subscriber notification procedures.

Note In certain scenarios, the customer might use the Sh interface between PCRF and HSS also.

An example configuration is shown below:

Figure 64: Diameter Application - Sh Application Example



The following parameters are configured under Sh Application:

Table 18: Sh Application Parameters

Parameter	Description
Name	Name of the Sh application.

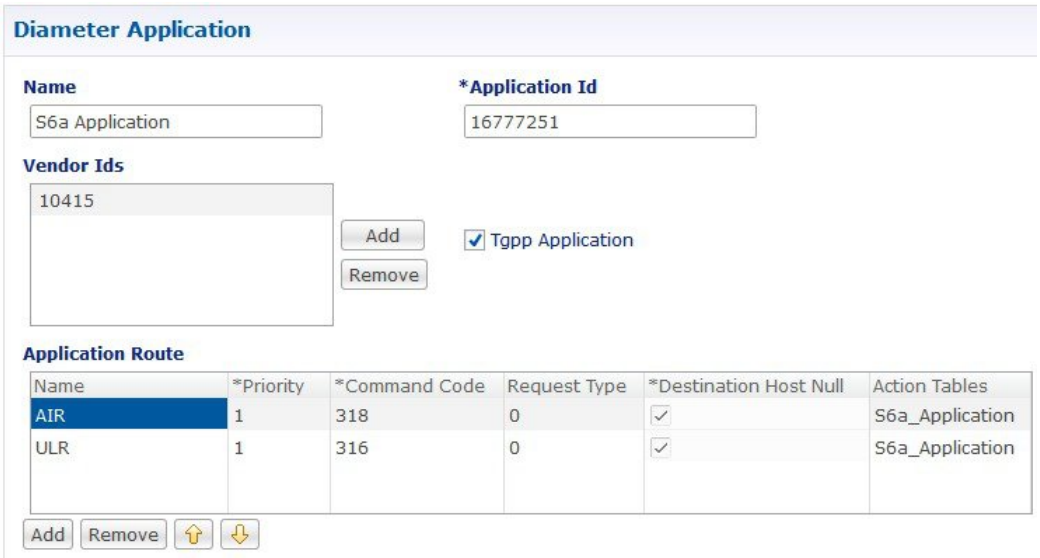
Parameter	Description
Application Id	16777217, 3GPP specified Application Identifier for Sh interface.
Vendor Ids	Vendor Identifiers that are required to be supported on Sh interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP defined application interface.
Application 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	CC-Request-Type is not applicable for Sh interface.
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.

S6a Application

DRA supports S6a interface with the implementation of Subscriber Location Function(SLF) feature. S6a is an interface which supports the mobility management and subscriber data management procedures between MME and HSS in an LTE EPC network.

An example configuration is shown below:

Figure 65: Diameter Application - S6a Application Example



The following parameters are configured under S6a Application:

Table 19: S6a Application Parameters

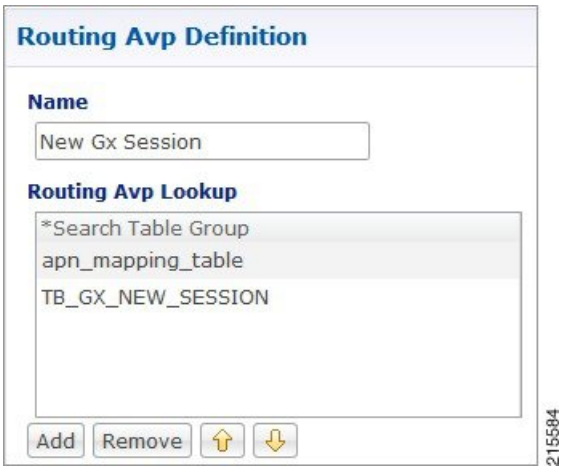
Parameter	Description
Name	Name of the S6a application.
Application Id	16777251, 3GPP specified Application Identifier for S6a interface.
Vendor Ids	Vendor Identifiers that are required to be supported on S6a interface.
Tgpp Application check box	If this check box is selected, indicates this is a 3GPP 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	CC-Request-Type is not applicable for S6a interface.
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.

Routing AVP Definition

Gx Session

An example configuration is shown below:

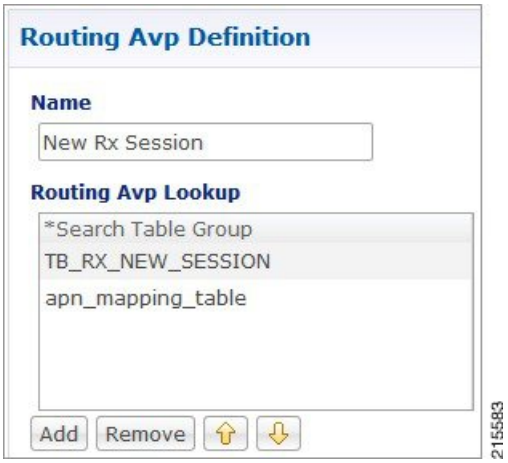
Figure 66: Routing AVP Definition - Gx Session



Rx Session

An example configuration is shown below:

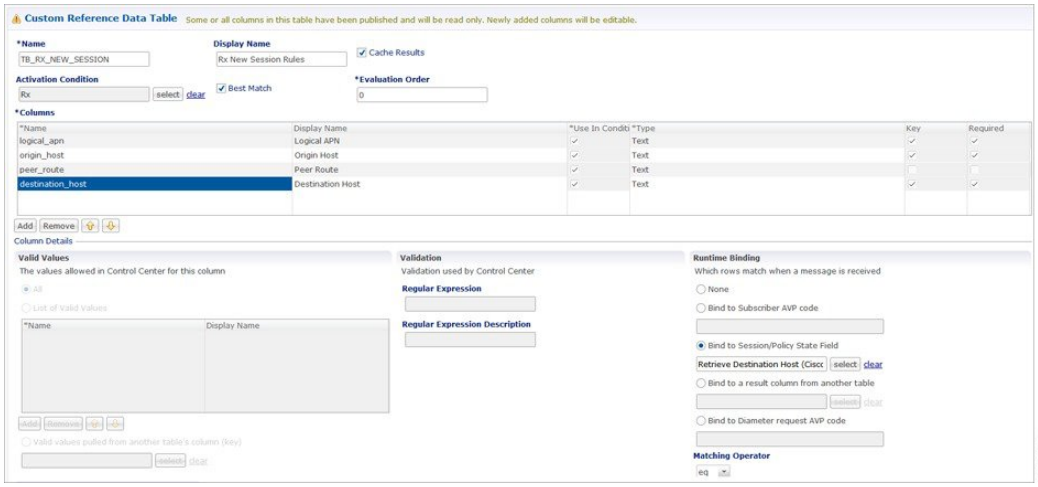
Figure 67: Routing AVP Definition - Rx Session



Rx New Session Rules - CRD Table

An example configuration is shown below:

Figure 68: Rx New Session Rules - CRD Table

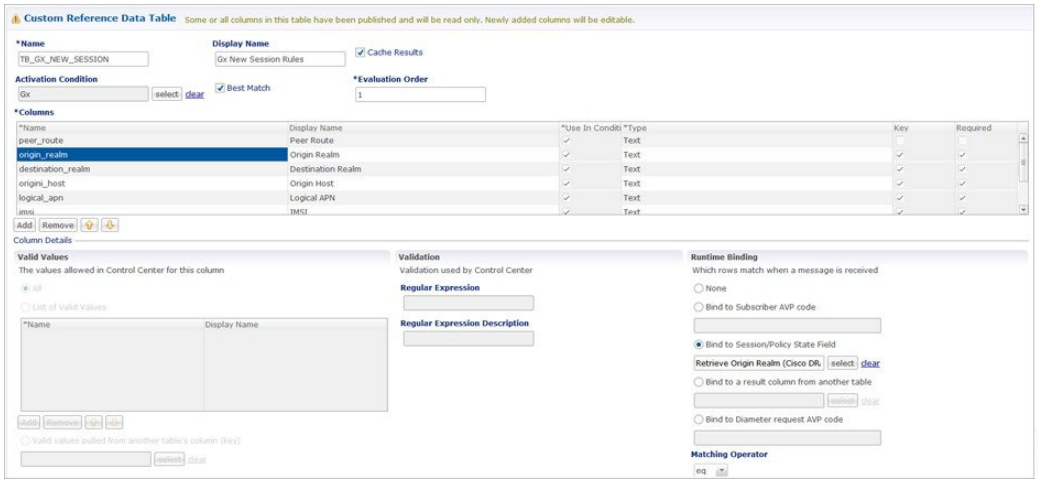


Gx New Session Rules - CRD Table

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:

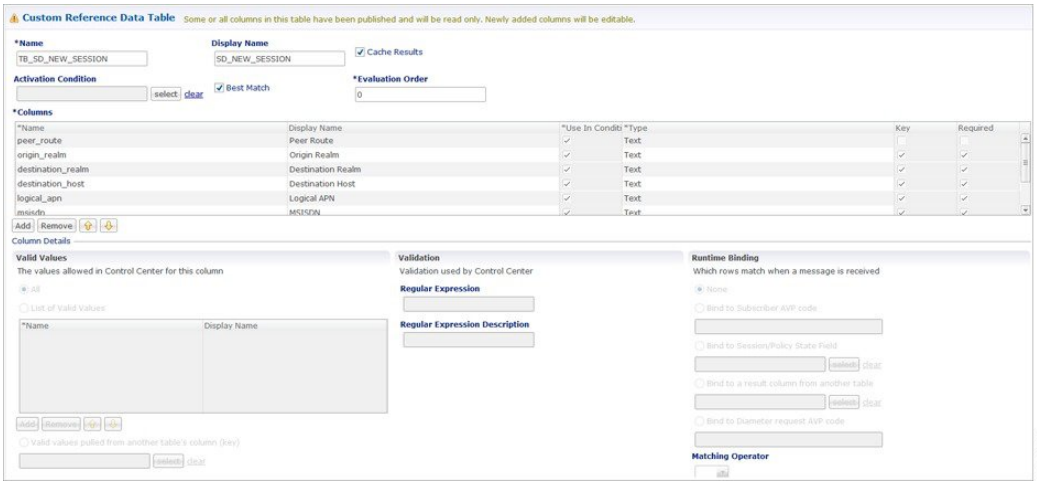
Figure 69: Gx New Session Rules - CRD Table



Sd New Session Rules - CRD Table

An example configuration is shown below:

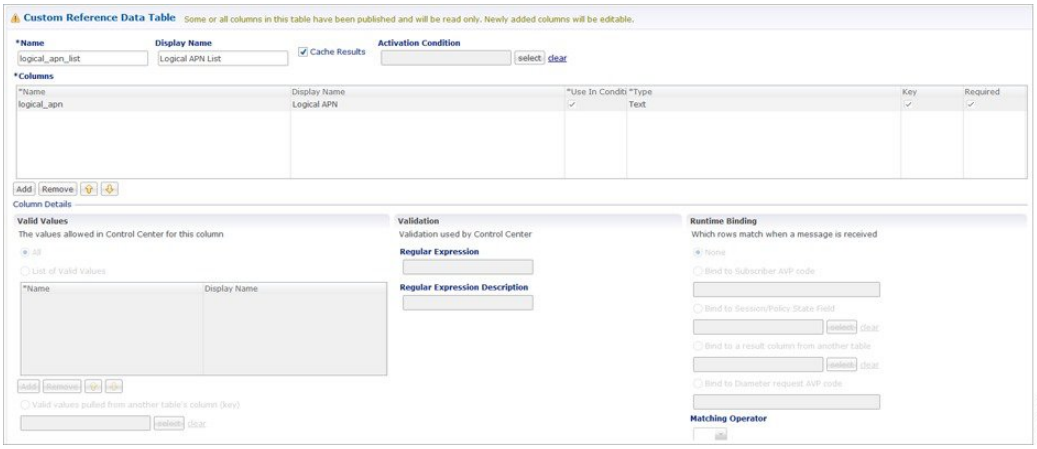
Figure 70: Sd New Session Rules - CRD Table



Logical APN List - CRD Table

An example configuration is shown below:

Figure 71: 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 20: 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
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.

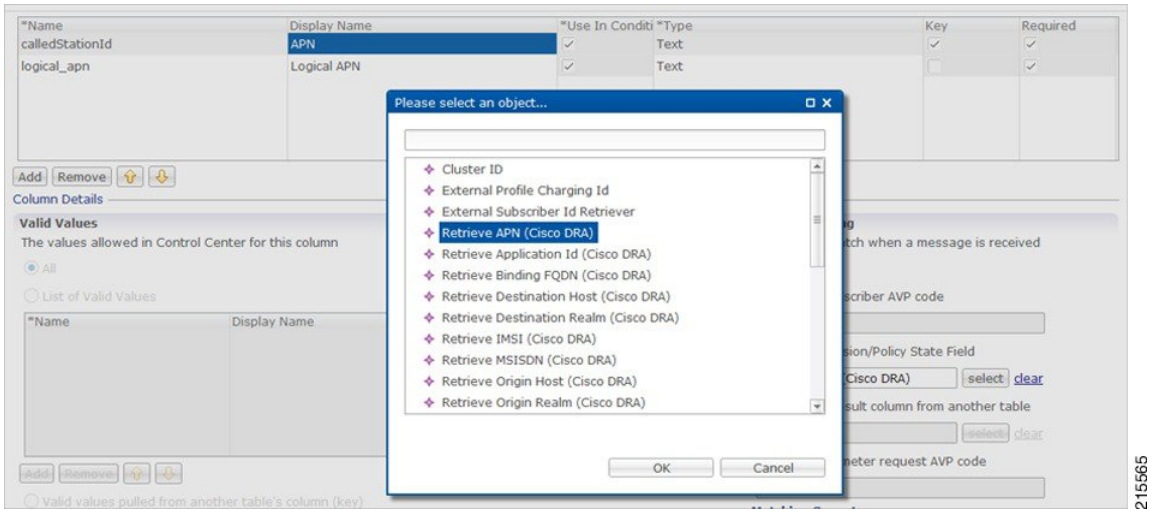
Procedure

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 72: Adding AVPs



Step 3 Repeat these steps to add additional AVPs.

Custom Reference Data Tables

Search Table Groups

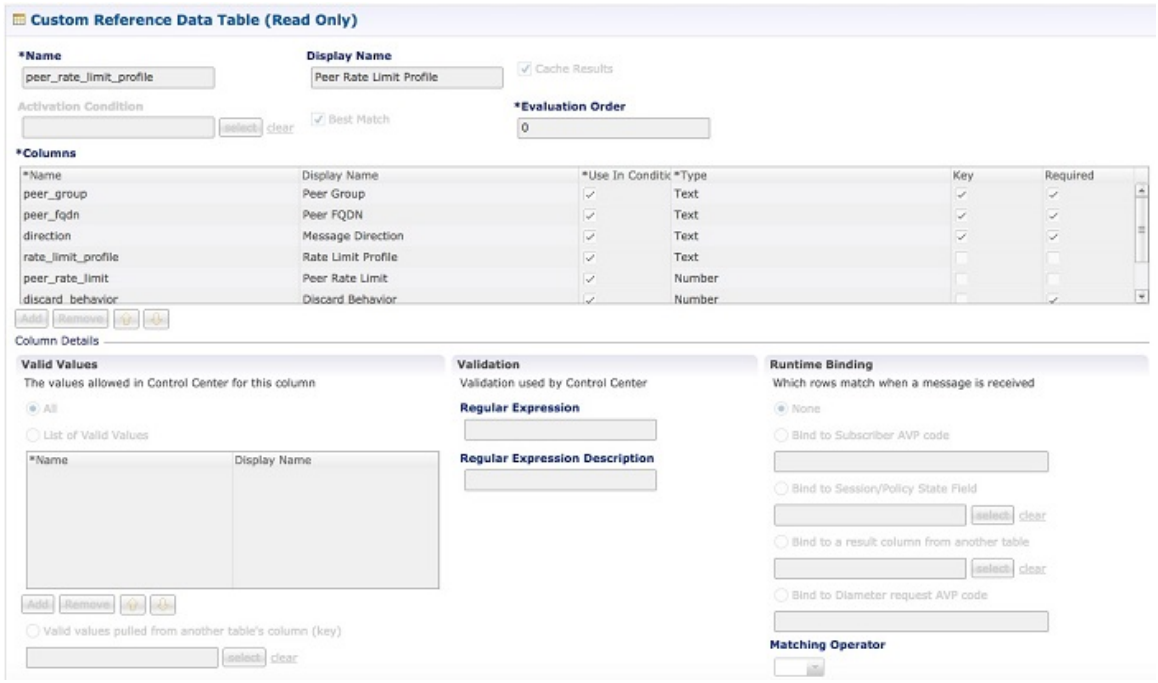
Peer Rate Limit Profile

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 73: 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 139 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.

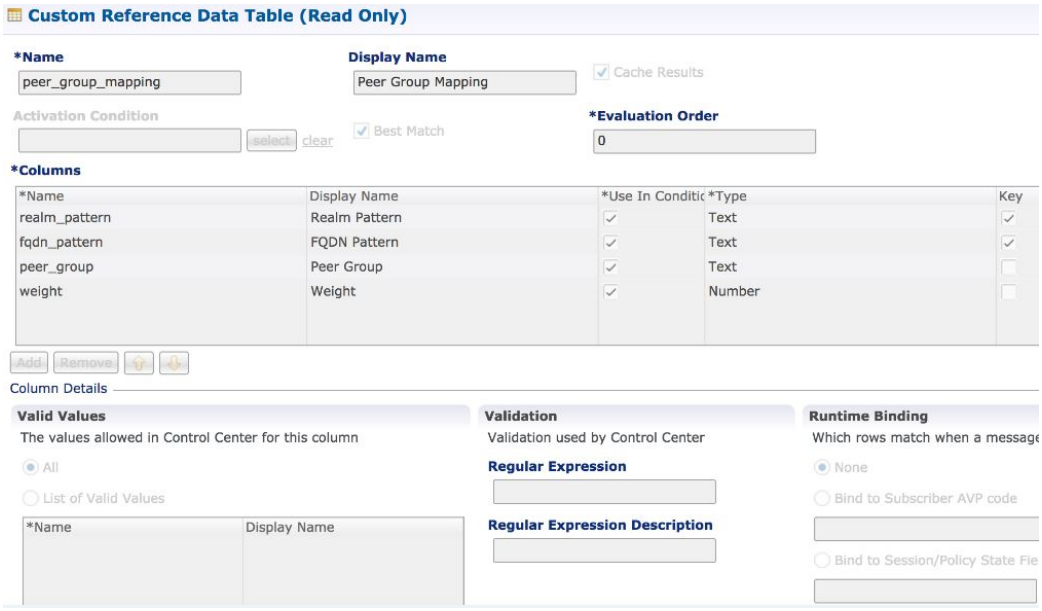
In case Discard Behavior is configured as Drop Message, this field is ignored. This is an optional field when Discard Behavior is configured as Send Error Answer.

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 129.

Peer Group Mapping

Figure 74: Peer Group Mapping - STG

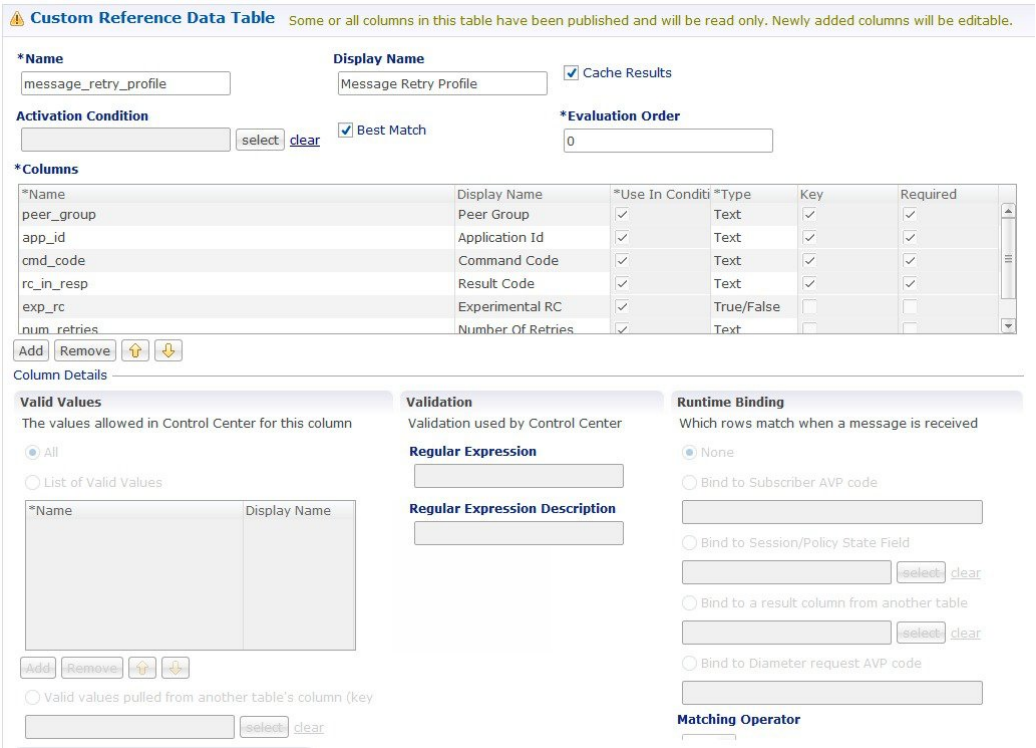


For more information, see Peer Group Mapping, on page 132.

Message Retry Profile

Message retry profile has been added.

Figure 75: 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 136.

Message Mediation Profile

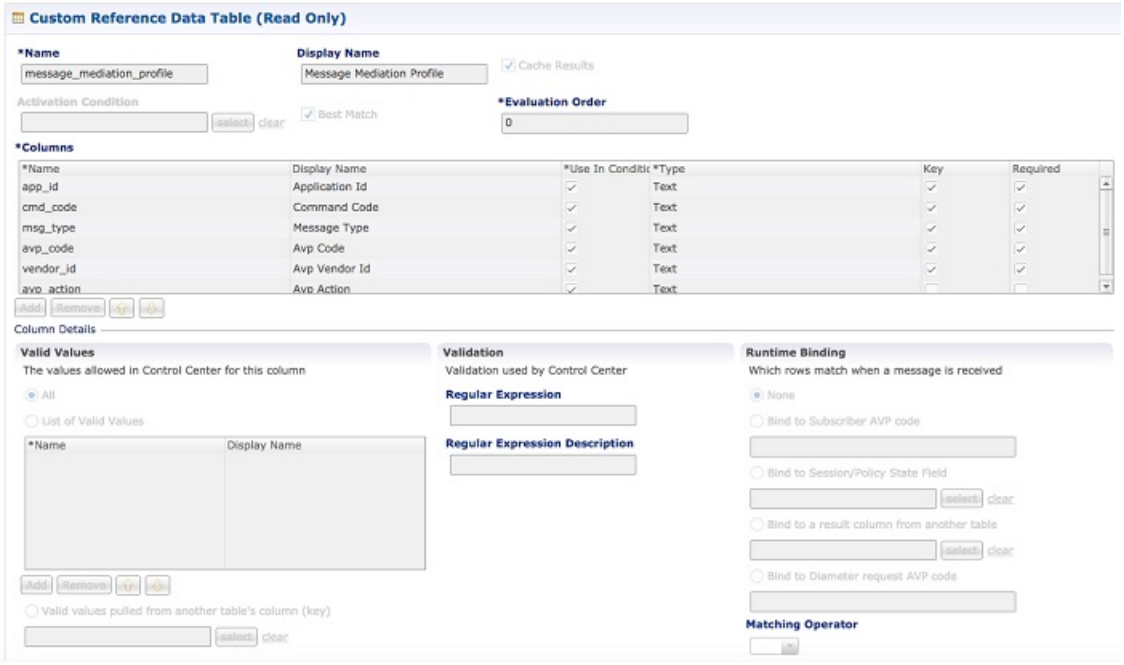
The message mediation profile is used to provide support for mediation of AVPs in Diameter request and answer.

- 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 76: 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.

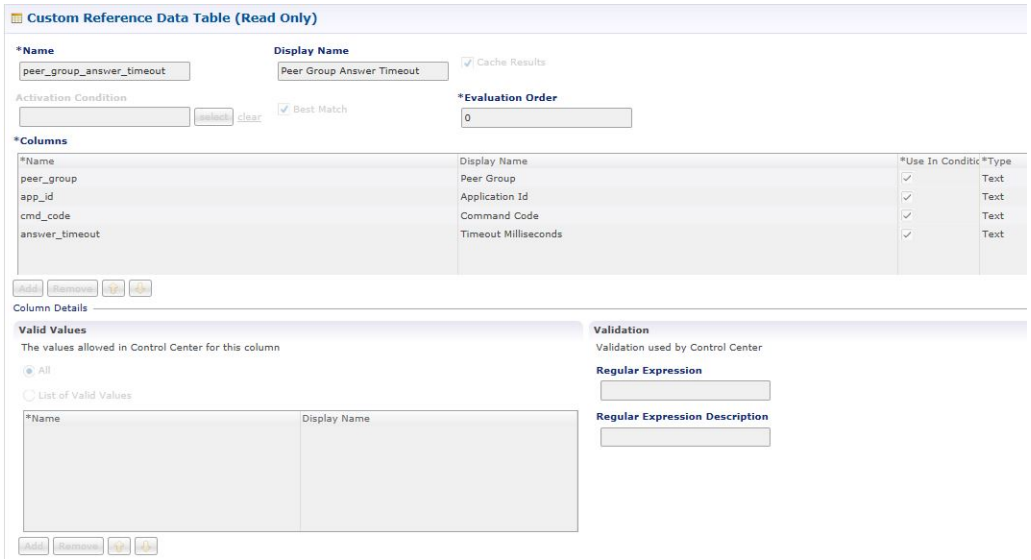
Best Match check box needs to be checked if you want to use the wildcard feature.

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 77: Peer Group Answer Timeout - STG



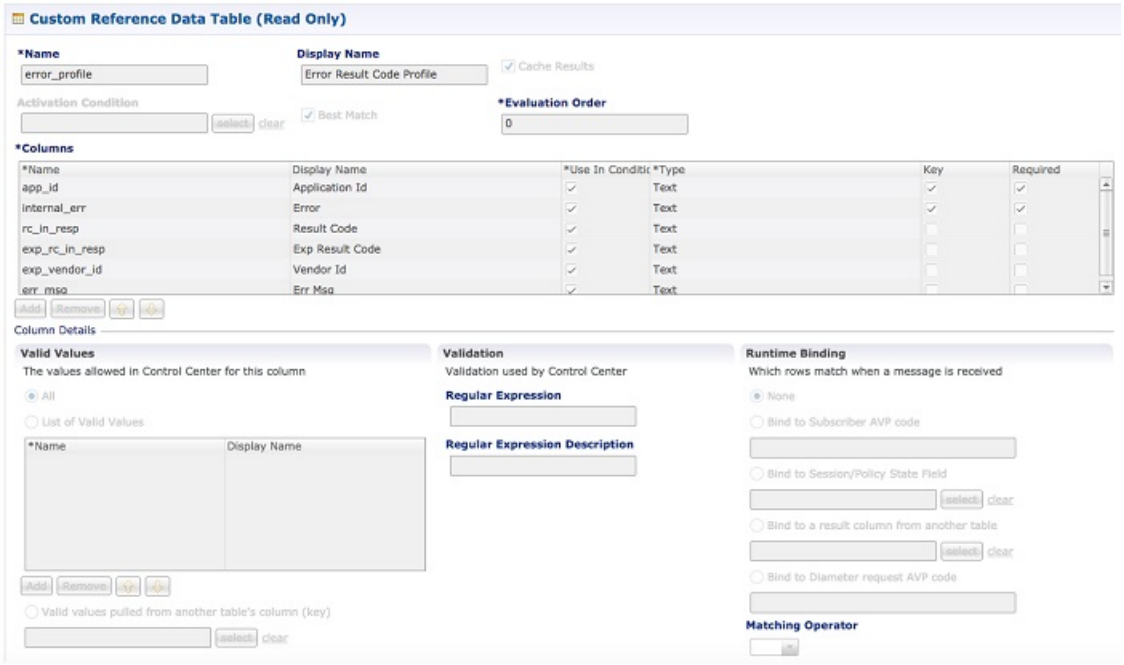
- Application Id: Application Id of the diameter applications.
- Peer Group: Peer group for which the timeout is applied.
- Command code (to enable different timeouts for different Diameter commands)
- Timeout: Timeout in milliseconds.

For more information, see Peer Group Answer Timeout, on page 138.

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 78: 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.

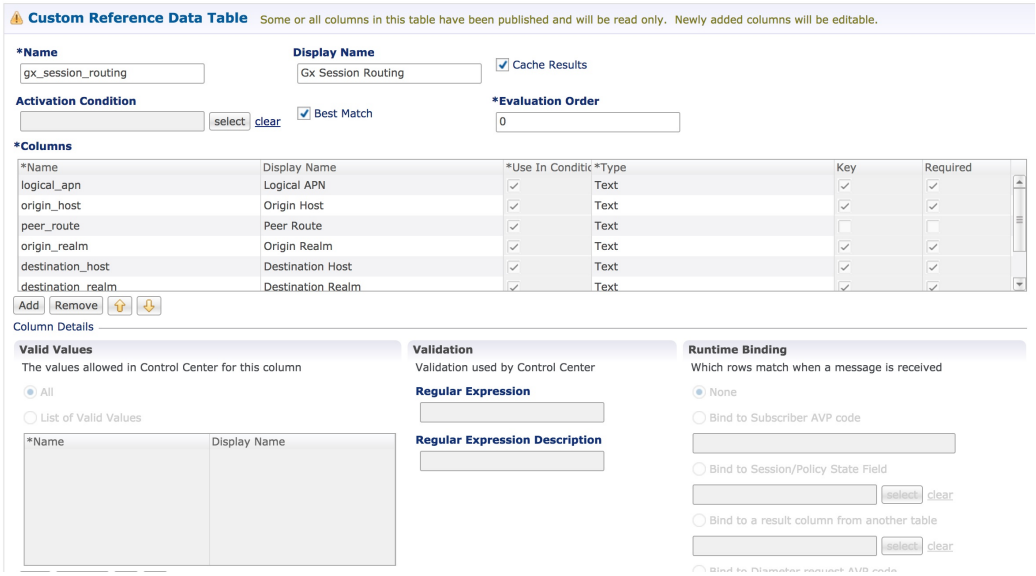
Note Experiment result code will be sent when Result-Code is not configured. If both Result-Code and experimental Result-Code are present, Result-Code would take precedence.

For more information, see Error Result Code Profile, on page 144.

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 79: Gx Session Routing



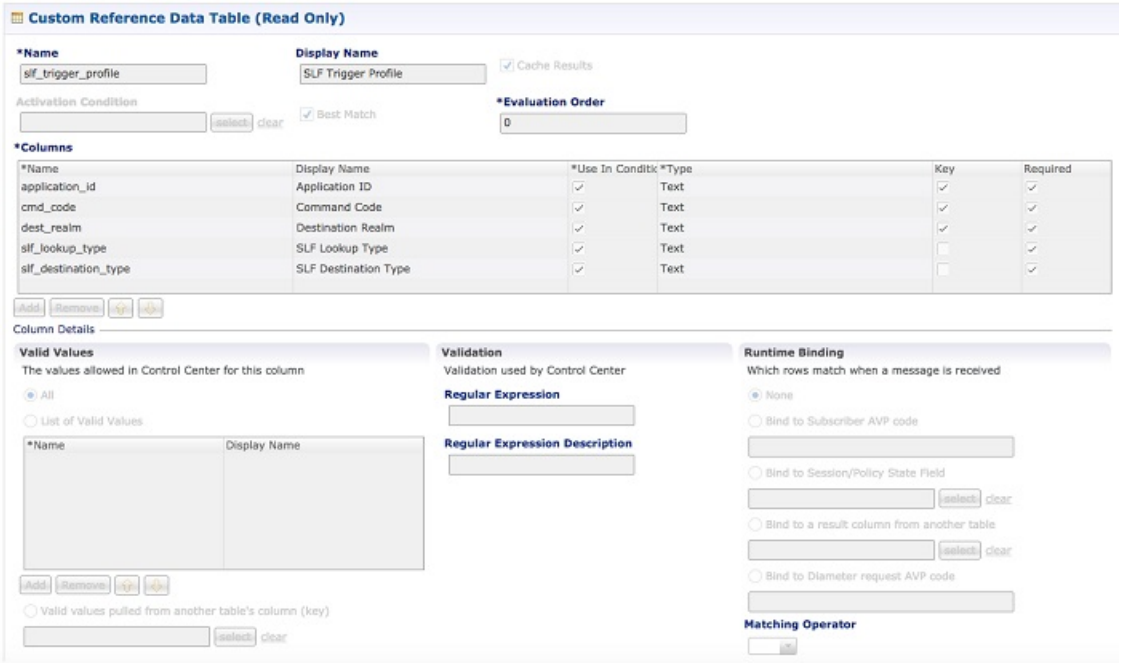
For more information, see Gx New Session Rules, on page 145.

SLF Trigger Profile

This table is used to derive SLF destination type and SLF lookup type. Keys used for this table are: Application Id, cmd_code, and dest_realm. Output of this table are slf_lookup_type and slf_destination_type.

An example configuration is given.

Figure 80: SLF Trigger Profile - STG

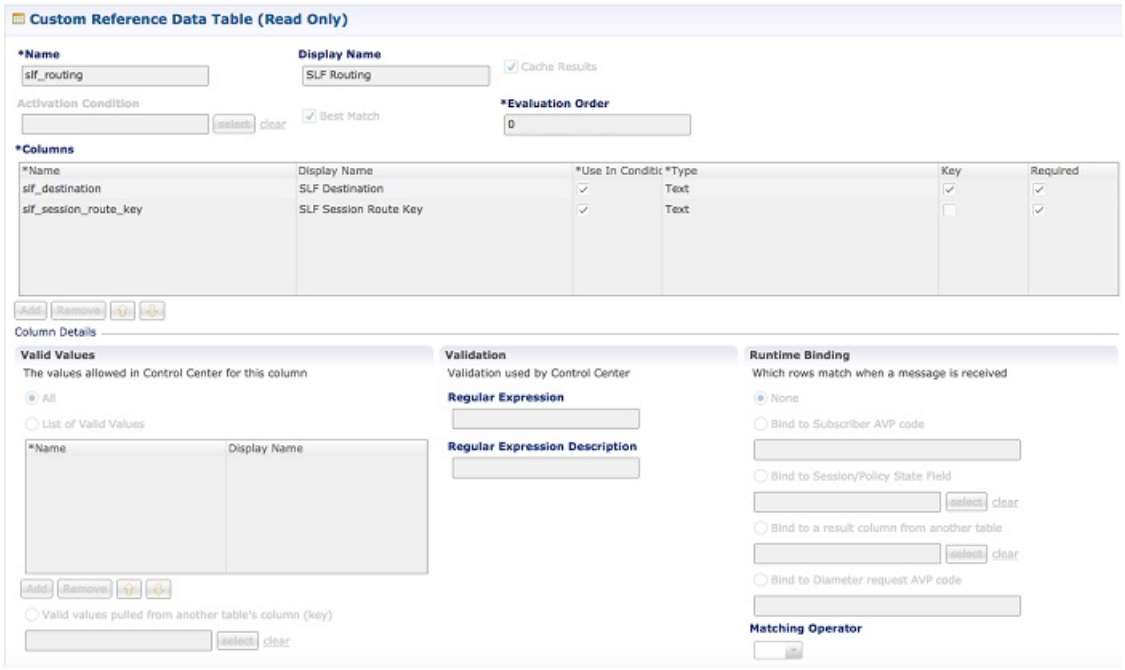


For more information, see SLF Trigger Profile, on page 147.

SLF Routing

This table is used to derive SLF session route key from SLF Destination. An example configuration is given.

Figure 81: SLF Routing - STG



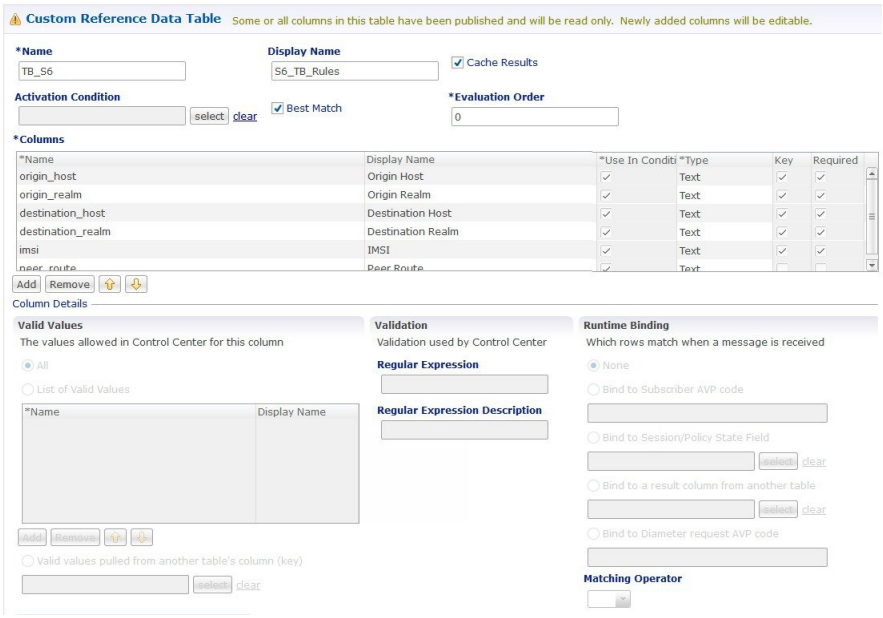
For more information, see SLF Routing, on page 148.

S6/Sh Table Driven Rules

This table is used for the table driven routing of S6/Sh messages. Fields origin_host, origin_realm, dest_realm, dest_host, msisdn, imsi are used as keys to derive the peer_route.

An example configuration is given.

Figure 82: S6 Table Driven Rules - STG



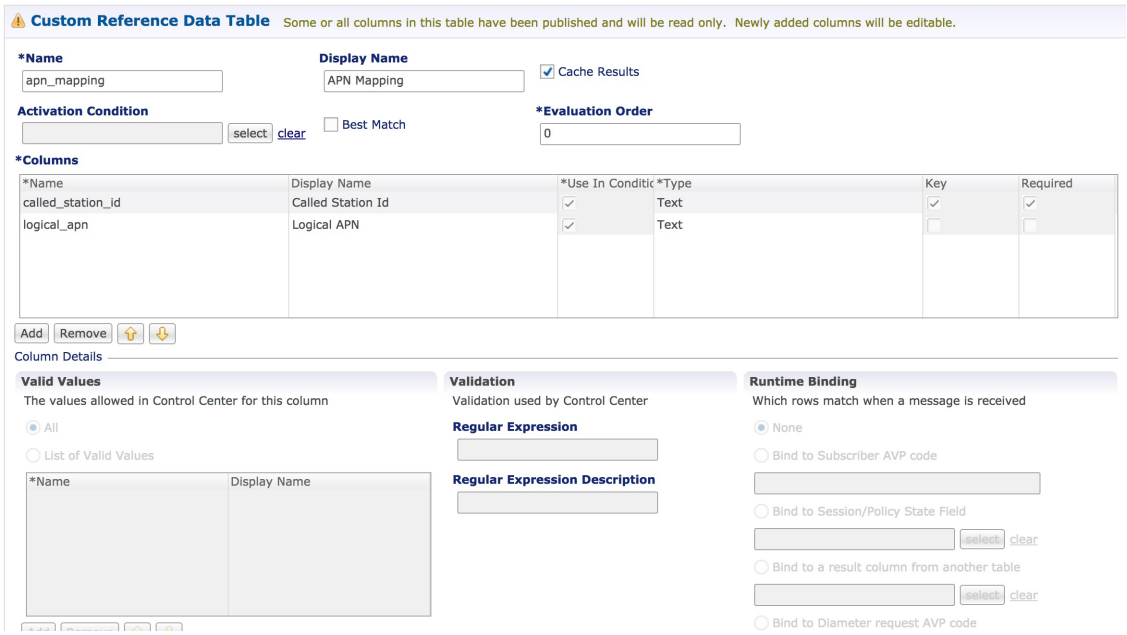
For more information, see S6/Sh Table Driven Rules, on page 148.

Custom Reference Data Tables

APN Mapping

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

Figure 83: APN Mapping - CRD Table



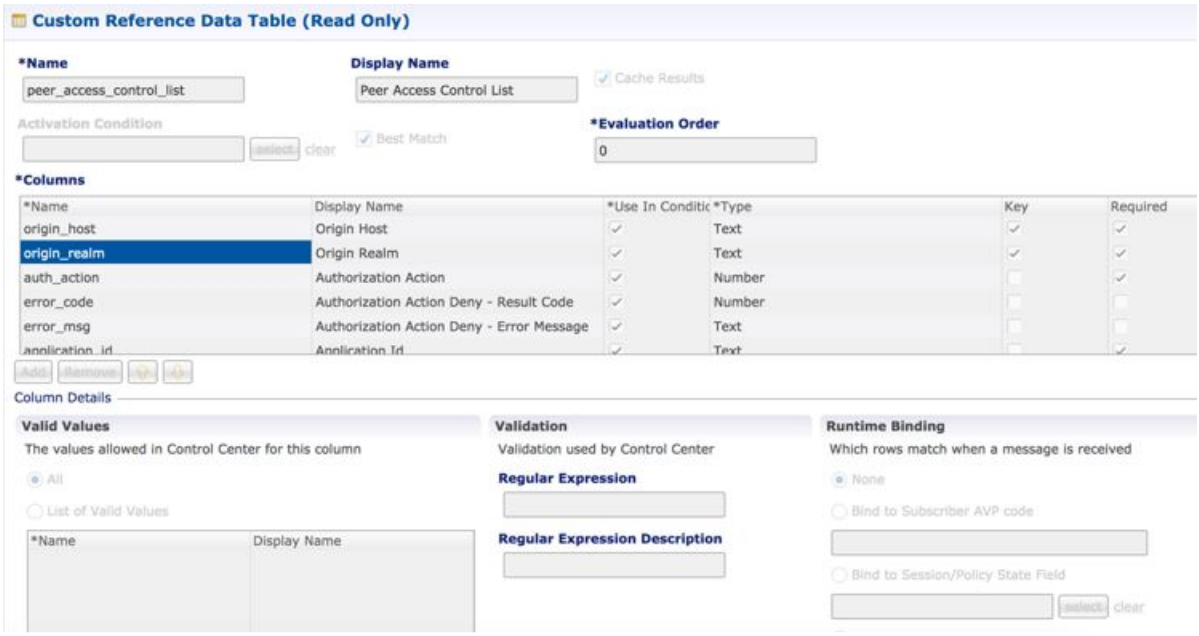
- Called-Station-Id: This is the AVP from which APN is derived. This also is the key column for this table. It is bound to the session or Policy State field as shown in the snapshot.
- Logical_APN: This is the mapped logical name that is used for referencing and processing the message within the system.

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 84: Peer Access Control List



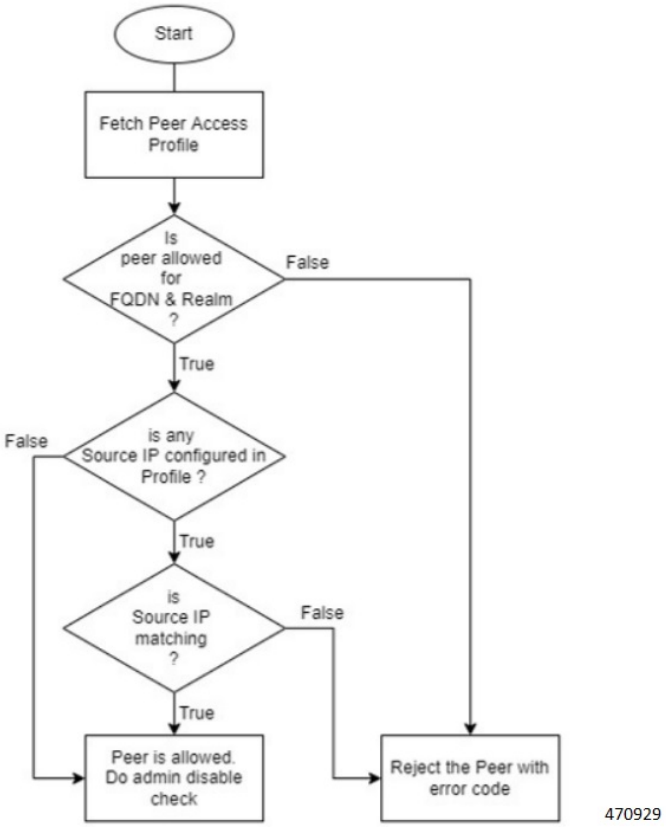
Source-IP Validation

In vDRA, you can allow or deny a peer based on the Source-IP validation. The Source-IP validation is an optional check, which an administrator can decide to configure Source-IP with peer FQDN/Realm or not. Source-IP uses Custom Reference Data (CRD) to persist the configuration. Hence the configuration is limited to a site. To block a peer from connecting to multiple sites, ensure to disable peer on each site.

Call Flow

The following section describes the call flow for Source-IP validation.

Figure 85: Source-IP Validation Call Flow For Connection Handling



The following procedure describes the Source-IP Validation for Connection Handling.

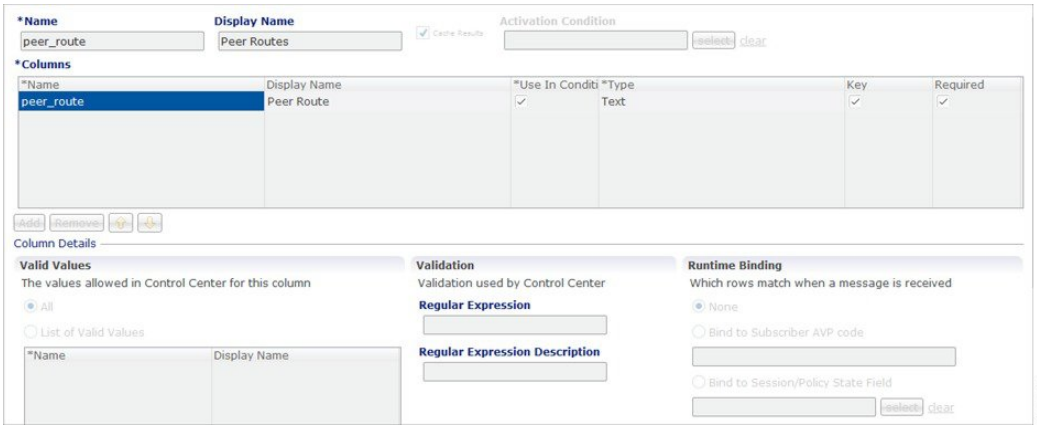
1. Diameter peer initiates a connection by sending a Capability Exchange Request (CER),
2. vDRA applies peer access control policy for the connection.
3. From the CER Request, vDRA fetches the Origin Host, Realm, and Source-IP that is Host-IP-Address AVP of CER.
4. vDRA fetches the Peer Access Profile detail from CRD and validates against the parameters collected from the request.
5. After the access control policy permits peer connection, vDRA responds with a Capability Exchange Answer (CEA), and a successful connection is established.

For more information about configuration, see the *Peer Control List* section in the *Custom Reference Data Configuration* chapter.

Peer Routes

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

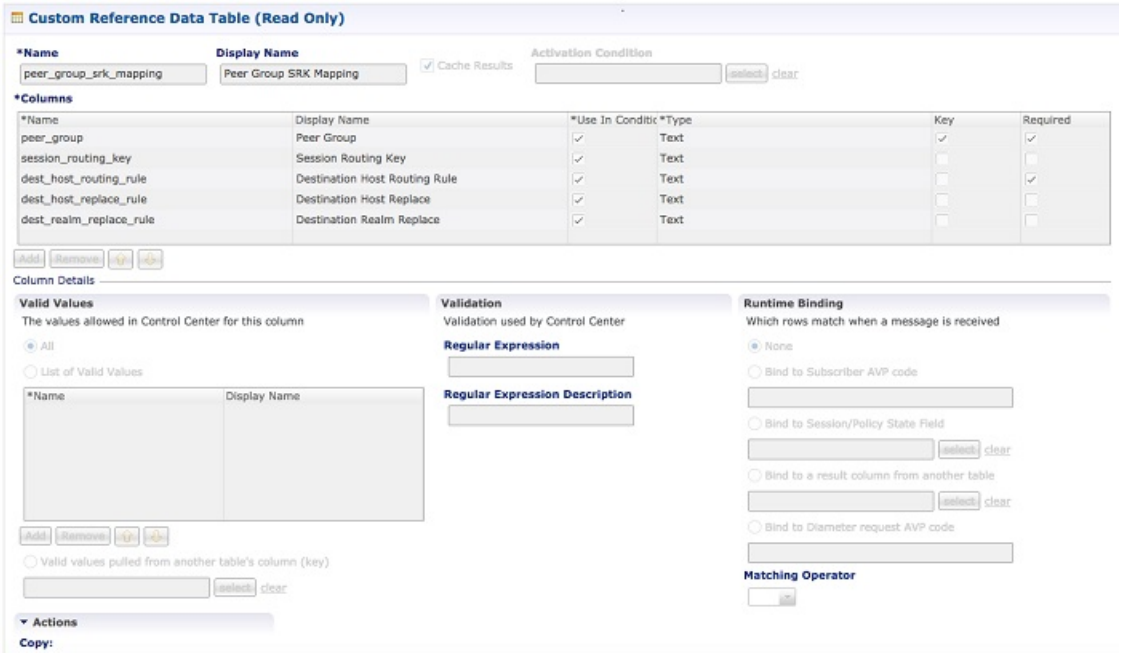
Figure 86: Peer Routes - CRD Table



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 87: Peer Group - CRD Table

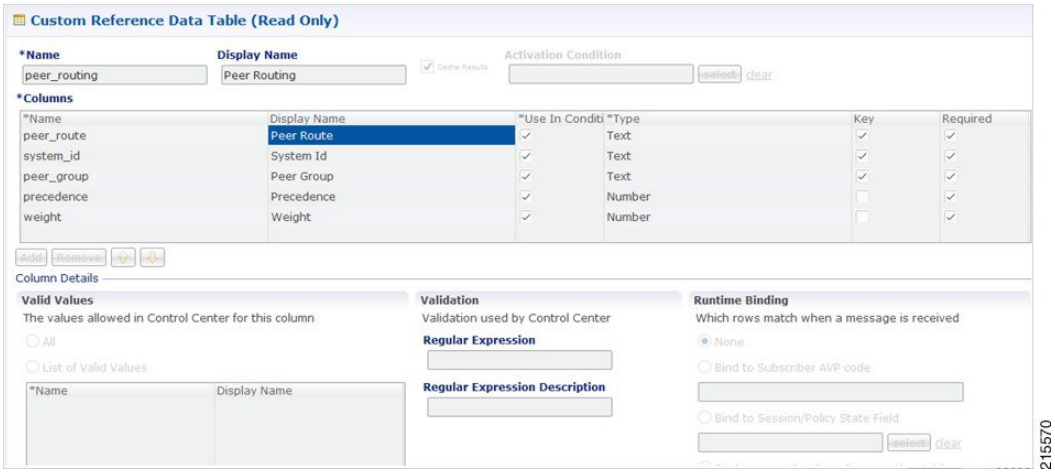


- Peer Group: Name of the peer group.
- Session Routing Key: Routing token for this Peer Group.
- Destination Host Routing Rule: Defines Routing behavior of this group.

Peer Routing

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

Figure 88: 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.

PCRF Session Query Peers

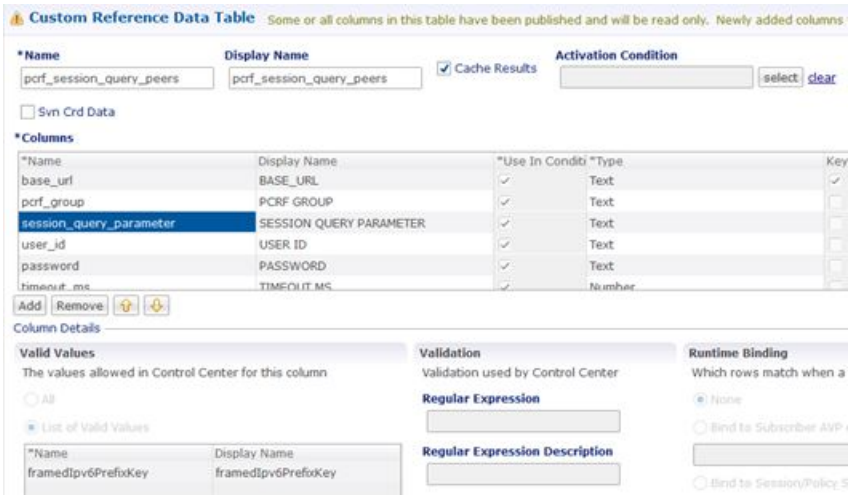
Use this CRD to configure the REST API parameters for Rx AAR fallback routing.

Policy DRA supports a fallback routing for Rx AARs for VoLTE using the PCRF session query.

For an Rx AAR with an IPv6 binding query, vDRA provides the ability to route the Rx AAR based on an API query to the PCRF to determine if it has a session for the IPv6. The queries can be made in parallel to a configured set of query points on PCRFs.

Note Ensure you have enabled PCRF Session Query in the DRA plugin configuration to use this feature.

Figure 89: PCRF Session Query Peers CRD



This CRD contains the following fields:

- base_url: The HTTP URL for the PCRF REST API, supports both HTTP and HTTPS. This does not contain the Rest API endpoint name.
- pcrf_group: The PCRFs can be configured in logical groups by defining the common pcrf_group. vDRA triggers the REST API request one after another for multiple PCRFs configured with same group name. This is to support PCRF with primary and secondary API endpoints. (Optional)
- session_query_parameter: PCRF session query parameter. Currently, only one value is supported: framedIpv6PrefixKey
- user_id: User ID for REST API request if PCRF requires any basic authentication. (Optional)
- password: Password for REST API request if PCRF requires any basic authentication. (Optional)
- timeout_ms: REST API equest timeout value. Default: 250ms. (Optional)

You can also configure a session route key for the PCRF response. When vDRA makes REST API requests to multiple PCRFs for session query using the Framed-IPv6-Prefix received in the Rx AAR message, the PCRF that has the corresponding Gx session sends a session route key in the response. vDRA then uses this key to look up the peer group and route the Rx AAR message to the correct PCRF. To configure a session route key in the response, see the Unified API Plugin Configuration in *CPS Mobile Configuration Guide*.

Additionally, diameter load balancing ensures that when a PCRF is connected to two directors and the PCEF traffic passes on one director, the traffic is then equally distributed to both directors.

vDRA can also load balance session query REST requests across multiple PCRF API endpoints. Previously, all REST queries were sent to the primary endpoint and only if the primary query fails, then the request is sent to secondary. Now, the requests are load balanced across the different PCRF endpoints within a peer group. If the session query results indicate that the PCRF does not have the corresponding Gx session for the IPv6 prefix, then vDRA does not send the query to the other PCRF configured in the same group. Similarly, for all other failures, vDRA sends the session query request to a different PCRF REST API in the same group. It is recommended that a group may contain a maximum of four PCRF REST API endpoints. If there is no group name, the PCRF API endpoint is considered as a standalone PCRF.

IPv6 Ranges System ID Mapping

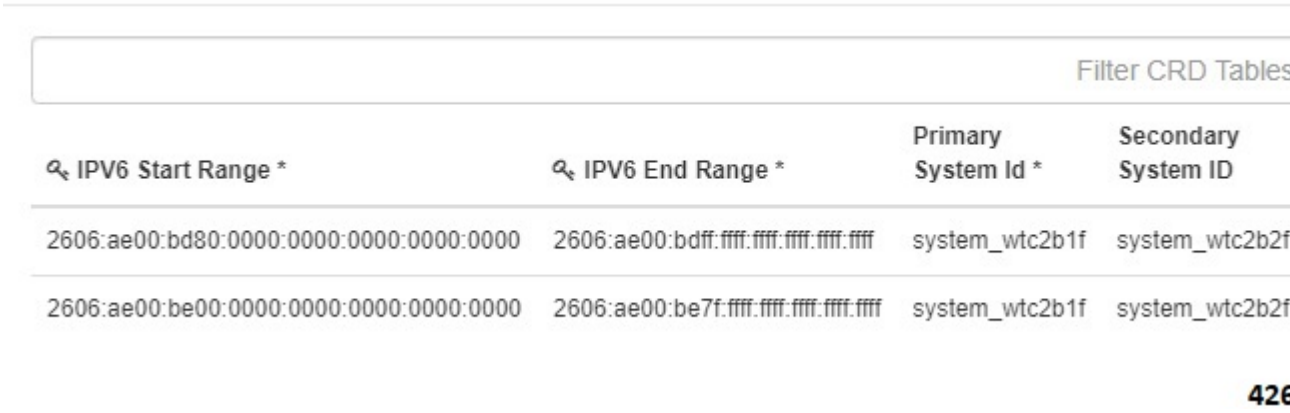
Use this CRD to specify a range of IPv6 addresses and the relay vDRA system ID.

This CRD is used to relay Rx AAR messages to other vDRA clusters based on the IPv6 range defined in the CRD.

When an Rx-AAR reaches vDRA, the AAR is checked for an IPv6 prefix. If there is an IPv6 prefix, then this CRD is checked for IPv6 ranges and to find the related primary and secondary vDRA system ID.

If the primary or secondary system is the current vDRA system-ID, then AAR message is processed locally. If the primary/secondary system ID is not the current vDRA, then current vDRA checks the relay links between current system and primary system. If the relay link is up, the the AAR is relayed to the primary system; else vDRA checks link to the secondary system.

Figure 90: IPv6 Ranges System ID Mapping CRD



Use the following table to specify a range of IPv6 addresses, the primary, and secondary vDRA system IDs.

Table 21: IPv6 Ranges System ID Mapping Fields

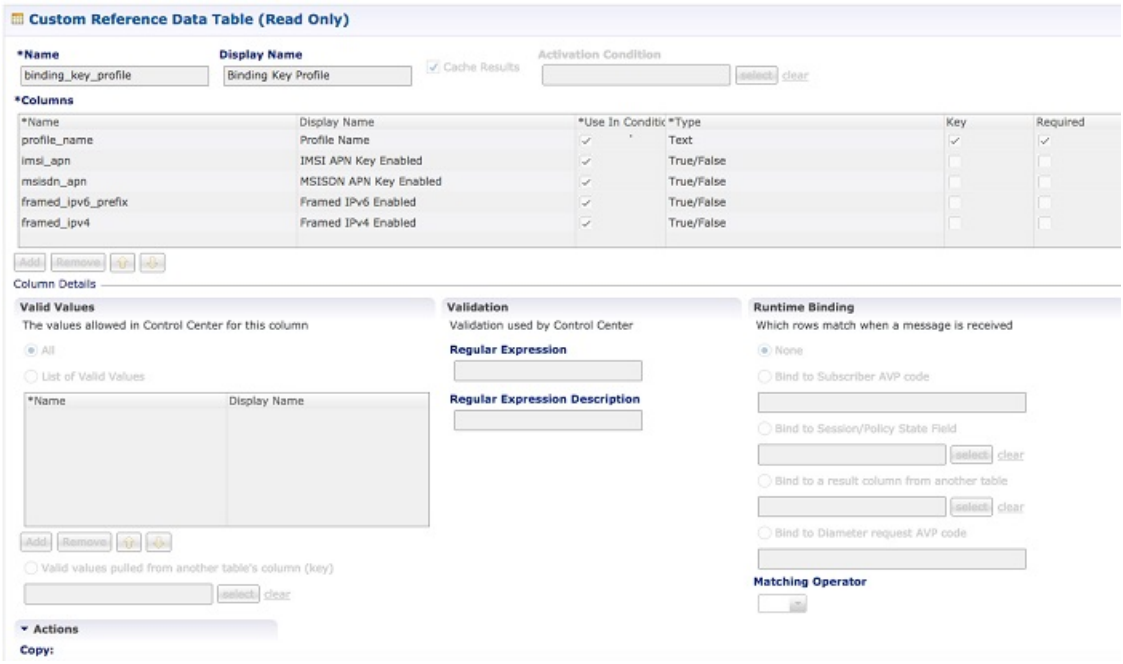
Fields	Description
IPV6 Start Range	Starting IP of IPv6 range in long format.
Note The starting and ending IPv6 range can be of 64/128 bits. The 128-bit not is supported for actual zone range configuration and below. The 128 bit format is supported for 64 bit framedIP range.	
IPV6 End Range	Ending IP of IPv6 range in long format.
Primary system ID	Mandatory field. Indicates the System ID of vDRA in a vDRA cluster to which the request can be relayed.
Secondary system ID	Secondary vDRA to which the request can be relayed if the primary is not present.

Note The ranges are expected to be mutually exclusive and unique. Verify the values when provisioning the same.

Binding Key Profile

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

Figure 91: Binding Key Profile - CRD Table



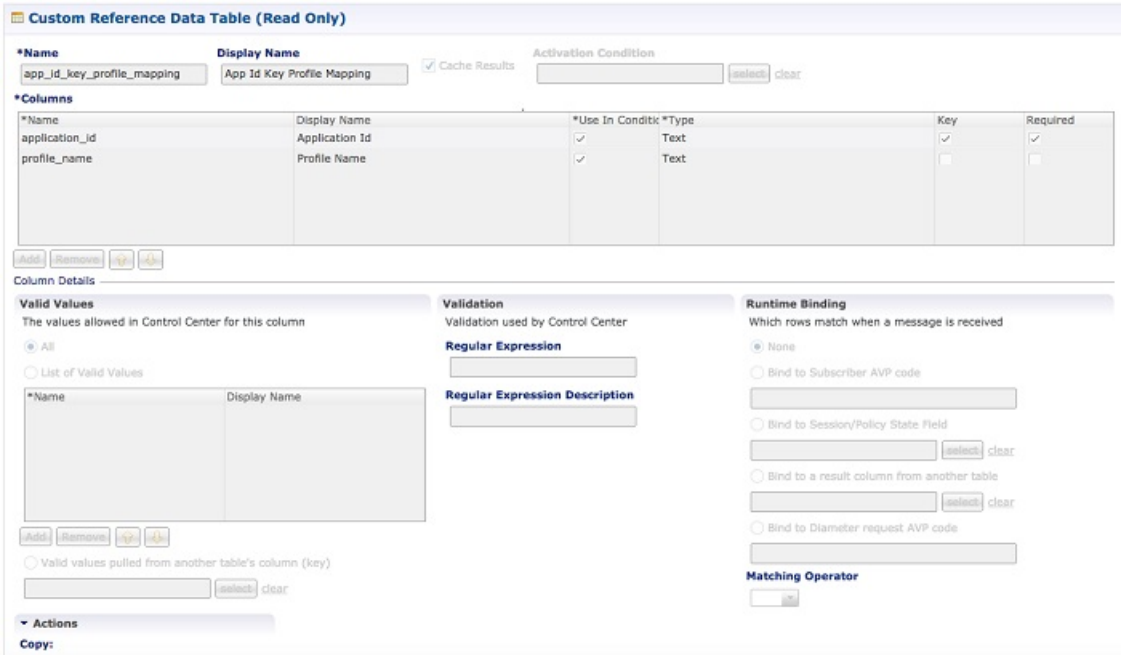
- Profile Name: This is the name given to the Bind profile that is associated with keys that are either enabled and/or disabled.
- MSI APN Key Enabled: Enabling this field would mean that bindings will be stored in IMSI APN collections in bindings database.
- MSISDN APN Key Enabled: Enabling this field 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 135 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:

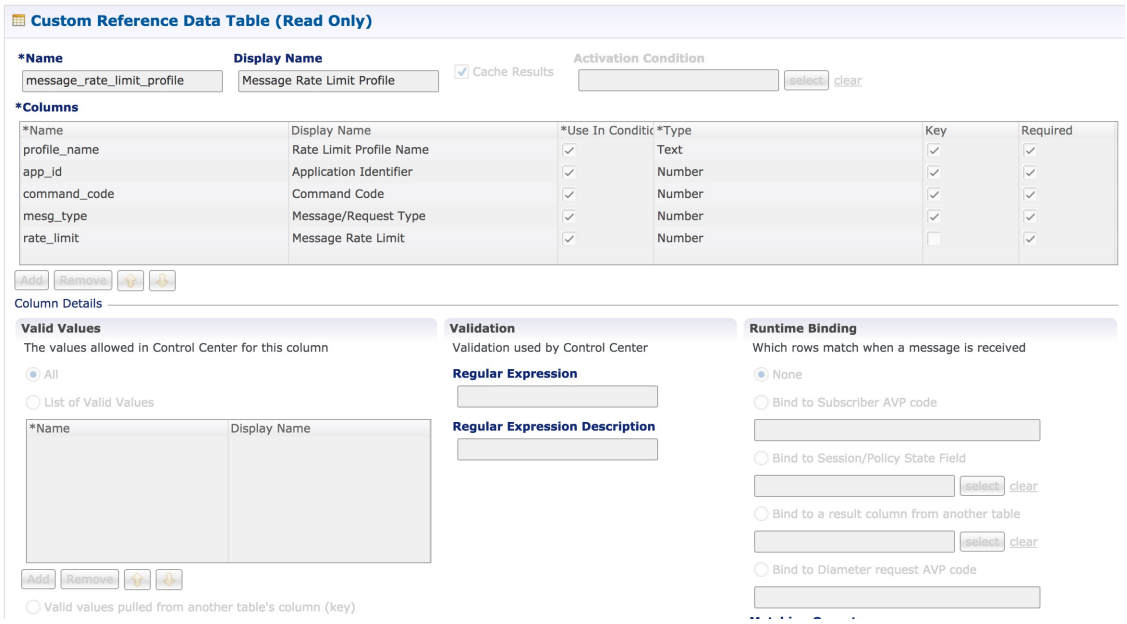
Figure 92: AppId Key Profile Mapping- CRD Table



Message Rate Limit Profile

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

Figure 93: 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.

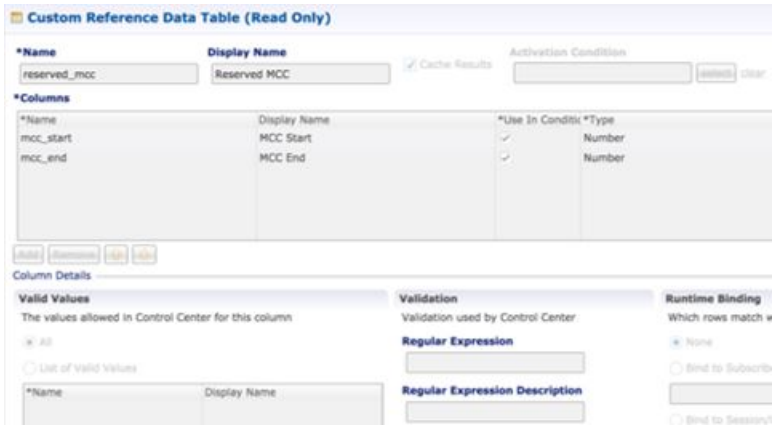
If the IMSI matches a reserved IMSI, the value is ignored for SLF routing.

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 94: Reserved IMSI

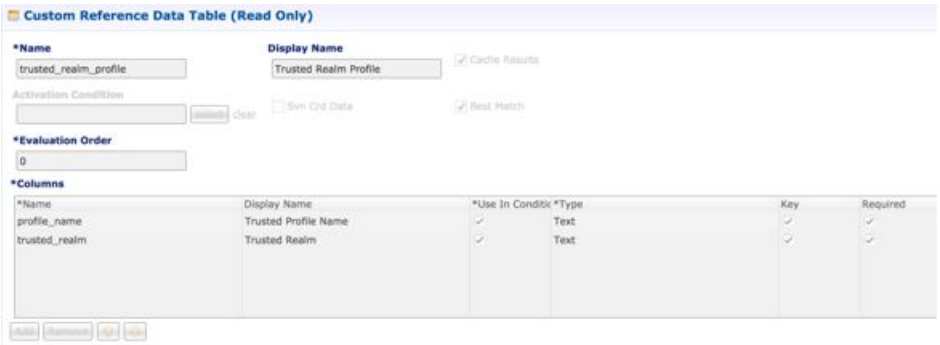


Trusted Realm Profile

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 95: 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 96: 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.

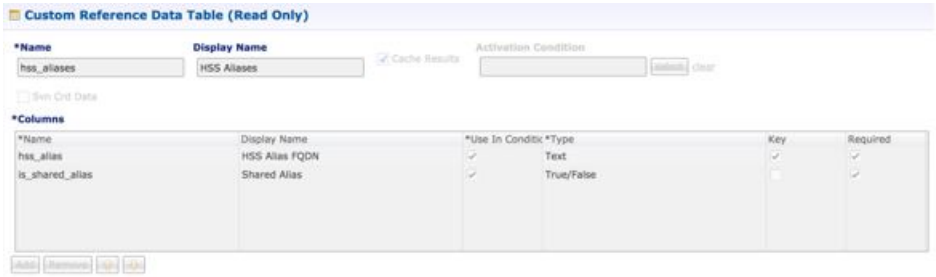
Figure 97: MME Alias Map

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 98: HSS Aliases

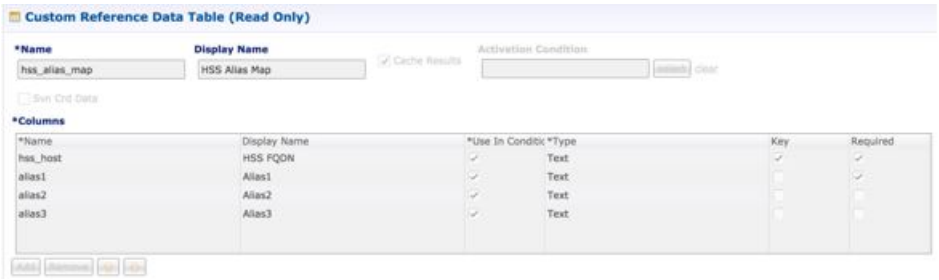


HSS Alias Map

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.

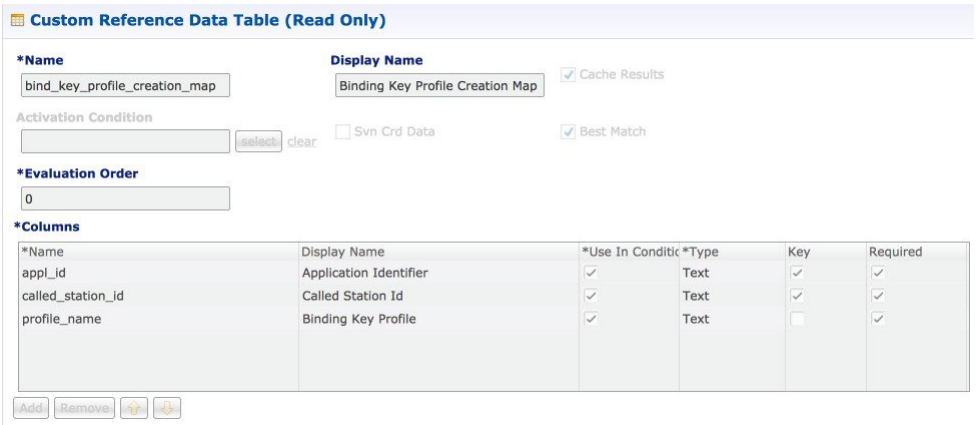
Figure 99: HSS Alias Map



Binding Key Profile Creation Map

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

Figure 100: Binding Key Profile Creation Map - CRD Table



Note If there is no profile configured for any Application ID and Called Station ID pair, then a default profile is automatically selected. This profile has only Framed-IPv4-Enabled as false/disabled, while all other keys are true/enabled.

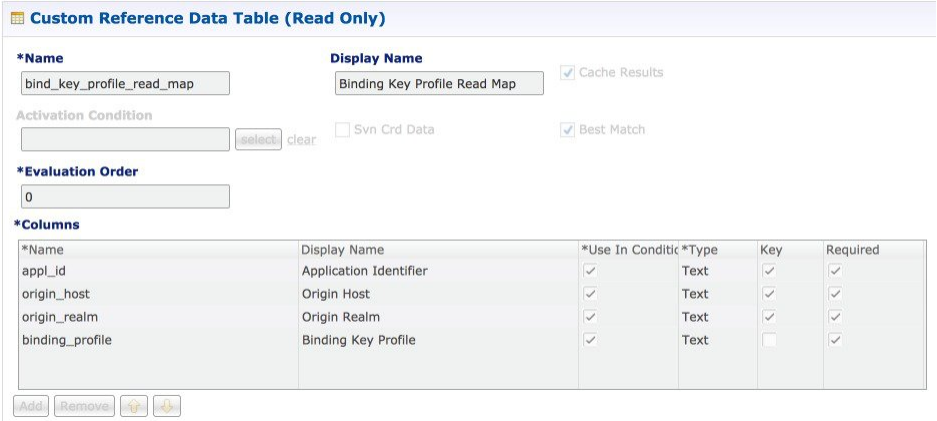
- Application Identifier: Application ID of the message.
- Called Station Id: Called-Station-Id AVP value from the Diameter message.
- Binding Key Profile: Profile name from binding key profile.

Refer to Binding Key Profile Creation Map, on page 151 for configuration in CPS Central.

Binding Key Profile Read Map

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

Figure 101: Binding Key Profile Read Map - CRD Table



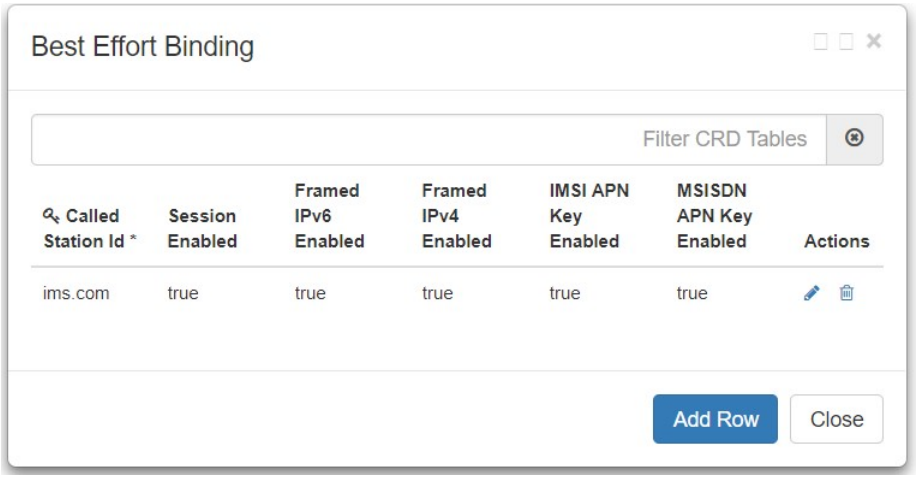
- Application ID: Application ID from the message.
- Origin Host: Origin host from the message.
- Origin Realm: Origin realm from the message.
- Binding Key Profile: Profile name from binding key profile.

Refer to Binding Key Profile Read Map, on page 152 for configuration in CPS Central.

Best Effort Binding

This table enables you to configure best effort binding on APN basis. The Caller Station Id column accepts regular expressions.

Figure 102: Best Effort Binding - CRD Table



Peer Admin Disabled List

Peer Admin Disabled List table is used by PAS to dynamically add/remove peer FQDN to administratively disable/enable peers. To administratively disable a peer, its FQDN should be added to “Peer Admin Disabled

List" table. To enable the peer, FQDN should be removed from the table. This table could also be updated by external systems using CRD API. The configuration changes take effect once CRD table is updated.

CRD table only supports exact matches (equality) of origin FQDN and realms. Pattern based rules are not supported. Since each peer is required to use unique origin-host FQDN, CRD table is designed to just include FQDN to identify a peer.

The CRD is used to persist the configuration. So, the configuration is limited to a site (scope of CRD). To block a peer from connecting to multiple sites, the peer must be disabled on each site.

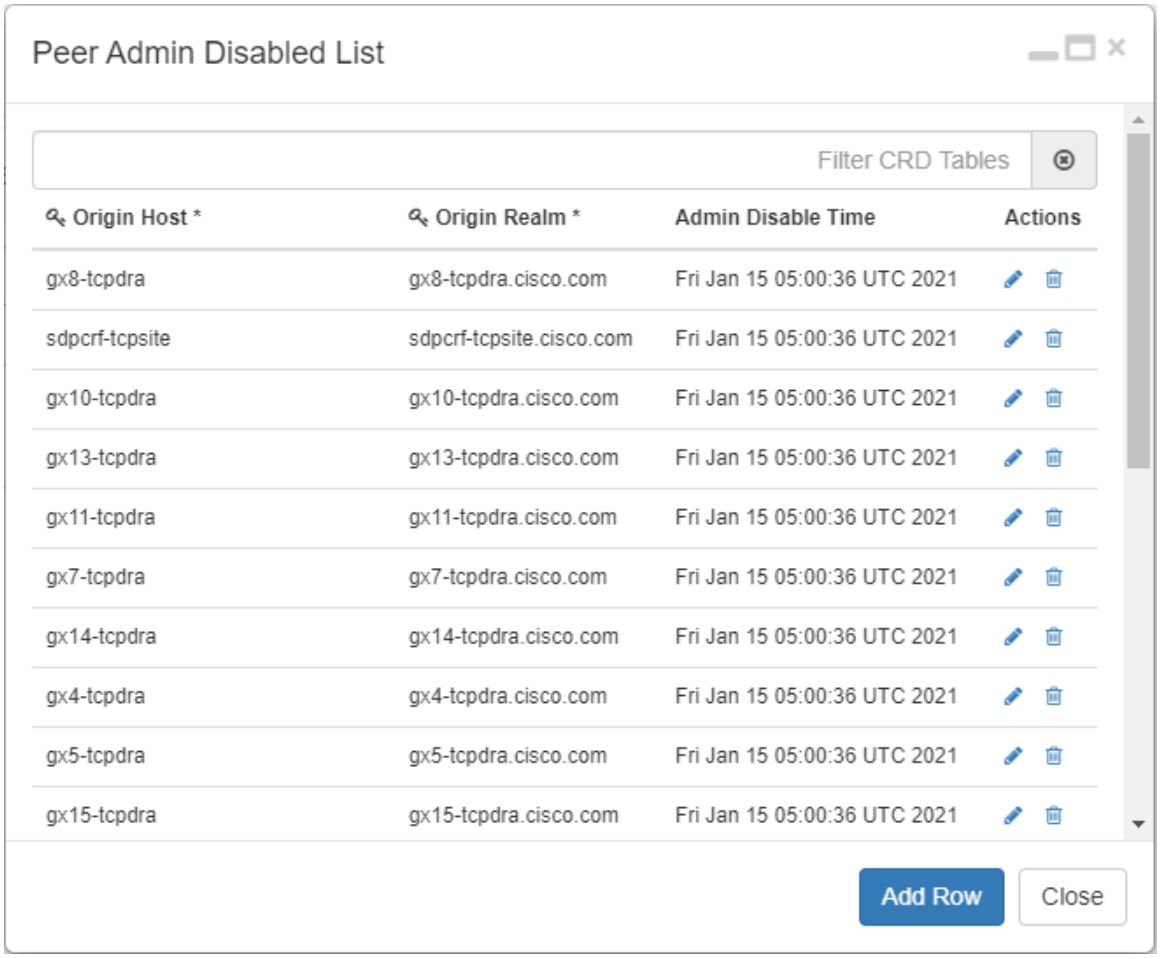
Note Peer Admin Disabled List is applied only for inbound diameter connections. Outbound diameter connections from PAS could be disabled by disabling the corresponding outbound endpoint.

When restoring CRD from backup, Peer Admin Disabled List should be excluded from import so that current configurations are not lost. The table should be included only if the intent is to reset the configuration.

When you add an entry for active peer in **Peer Admin Disabled List** CRD table, it takes effect only after the peer is disconnected and the peer attempts to reconnect. You can use **Active Peer Endpoints** GUI under **DRA Peer Monitoring** to disconnect the peer connection. For more information, refer to *View Filtered Data* section in the *CPS vDRA Administration Guide*.

If you need active peer connections to be administratively disabled, it is recommended to disable the peers using the **DRA Peer Monitoring** GUI only. For more information, refer to *CPS vDRA Administration Guide*.

Figure 103: Peer Admin Disabled List



The CRD table contains the following fields:

- Origin Host: Origin FQDN of peer to be administratively disabled.
- Origin Realm: Origin realm of the peer.
- Admin Disable Time: Time at which disable rule was created. This is read-only field.
- Actions: Edit or delete the current configuration.

The following APIs can be used to administratively disable and enable multiple peers. The APIs support bulk updates when multiple peers are selected in GUI.

- Disable APIs:
  - **API to create multiple rows in CRD:** `/custrefdata/peer_admin_disabled_list/_createRows`
  - **API to disconnect multiple endpoints:** `/dra/api/localActivePeerEndpoints/disconnect`
- Enable API:
  - **API to delete multiple rows in CRD:** `/custrefdata/peer_admin_disabled_list/_deleteRows`

For more information on APIs, refer to *API Endpoints And Examples* section in the *CPS vDRA Operations Guide*.

Attention Peer down alert (DIAMETER_PEER_DOWN) is suppressed for admin disabled peers. There is no change in handling of peer up or peer down state changes and corresponding alerts for admin enabled peers.

SVN Repository Changes

Note This feature has not been validated for all customer deployment scenarios. Please contact your Sales Account team for support.

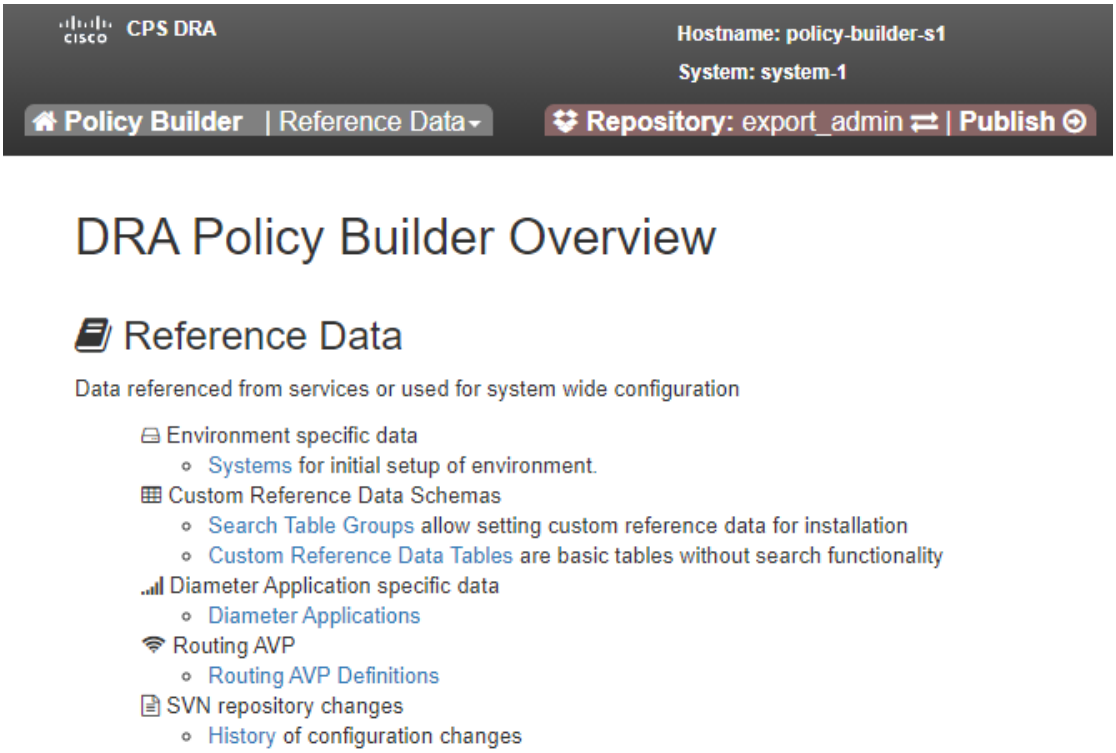
Viewing Summary of SVN Repository Changes in the Policy Builder

The CPS DRA provides GUI support to view history of Policy Builder configuration changes.

Perform the following steps to view the summary of publish changes:

1. In CPS DRA, choose **Policy Builder** > **Policy Builder** > **SVN repository changes** , click the **History of configuration changes** link to open the **History of configuration changes** window.

Figure 104: SVN Repository Changes



2. From the **Choose repository to view history** drop-down list box, choose a repository, and then click **Submit**. The following parameters are displayed for all the published commit changes published.

Figure 105: History of Configuration Changes

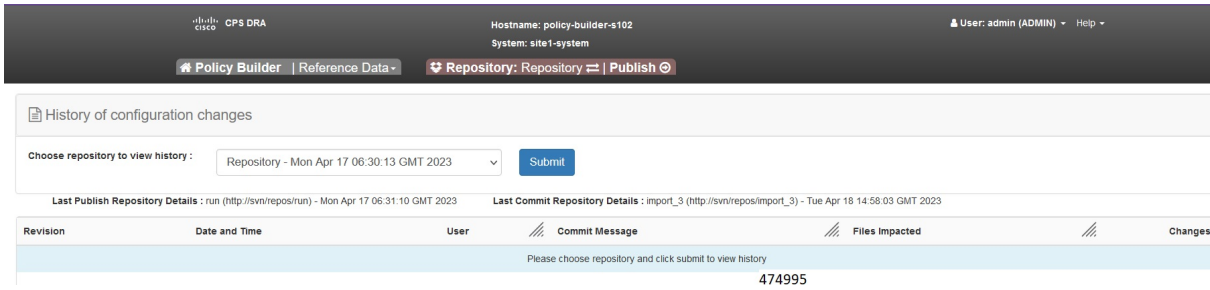


Table 22: History of Configuration Changes Parameters

Field	Description
Revision	Revision number of the SVN commit.
Date and Time	Shows the date and time of the last changes made.
User	Name of the user who made changes.
Commit Message	Commit message entered by user into GUI while publishing summary of changes.
Files Impacted	Shows impacted files during SVN commit changes.
Changes	Click the icon to view differences between two adjacent revisions. To download and save changes, click the Download icon at the top-right corner of the window.

Note DRA Central GUI retrieves the SVN log and SVN differences by using an underlying SVN containers. If SVN container is down then GUI will have issues.

View Last Published and Commit Repository Details

In the Policy Builder, you can view the last published and commit repository details using the API and SVN commands. It displays the following details:

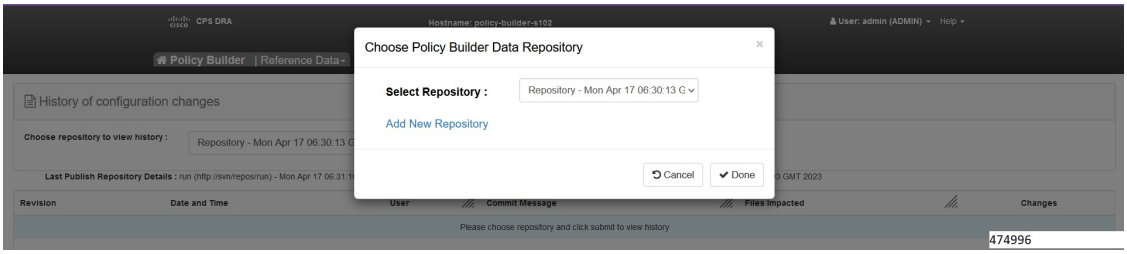
- Last committed repository and published repository in the history page.
- List of repositories sorted based on the last commit order in the DRA central.

API and SVN Commands

1. The following API displays the last published and commit repository details in the GUI page:

```
https://<Master/VIP-IP> / api/repository/actions/svn/repo/
```

Figure 106: Dropdown Repository List



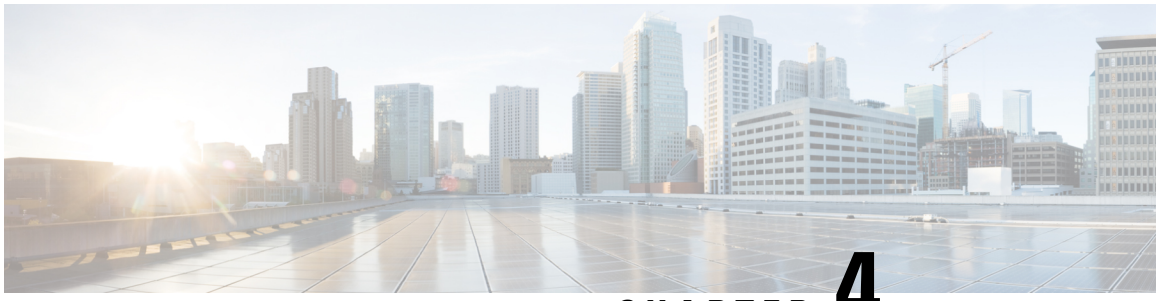
2. The following SVN commands helps to view the list of repositories based on the last commit.

```
svn list --xml http://svn/repos/ | grep name
<name>caliperpb</name>
<name>configuration</name>
<name>golden-crd</name>
<name>run</name>
<name>siteB_config</name>
```

Note The SVN commands are executed in the SVN containers.

Limitation

DRA Central GUI retrieves the SVN last publish and SVN commit repositories by using an underlying SVN containers. If SVN container is down then GUI will have issues.



CHAPTER 4

Custom Reference Data Configuration

Logical APN List

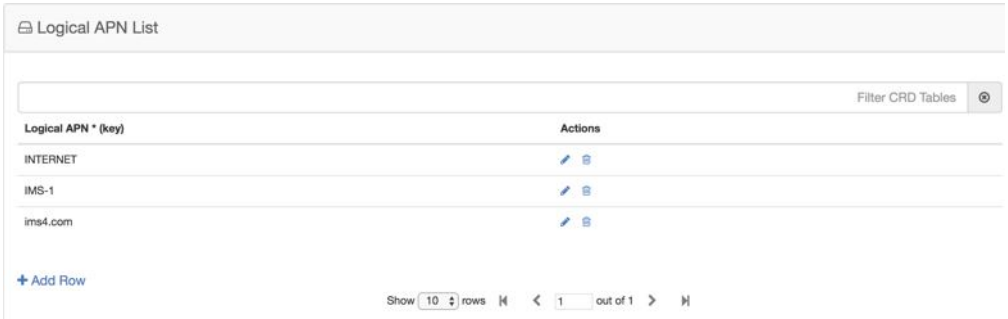
The logical APN feature allows multiple users to access different physical target networks through a shared APN access point. The logical APN feature reduces the amount of APN provisioning required by consolidating access all real APNs through a single virtual APN. Therefore, only the virtual APN needs 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 133.

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

An example configuration is shown below:

Figure 107: Logical APN List



Avp Condition Profile

The Avp Condition profile is used to specify the value and condition to apply to AVP.

The following table describes the fields of Avp Condition Profile:

Table 23: Avp Condition Profile

Field	Description
Profile Name	Profile name of the condition. Each row in the table is a condition. You can define multiple conditions for one Profile Name.
Avp	Avp that condition is applied to.
Avp Value	Value of the condition. If there is no AVP, configure the value as AVP_IS_MISSING

Figure 108: Avp Condition Profile

Avp Action Profile

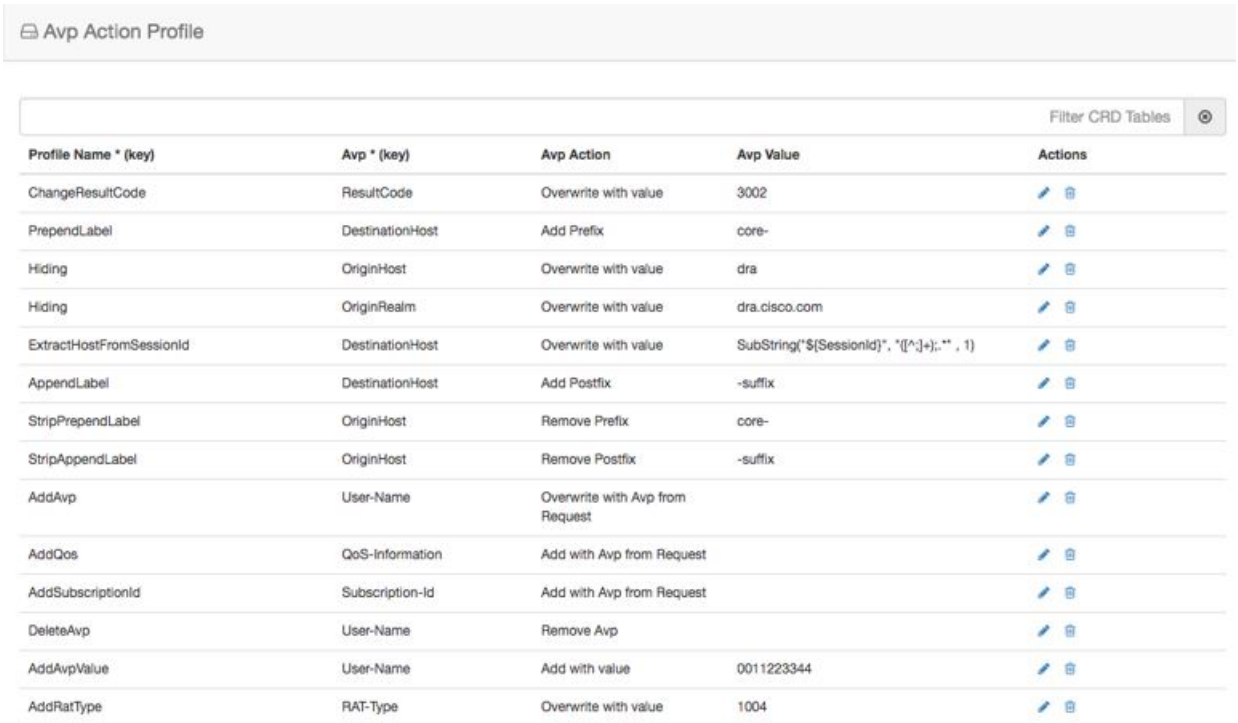
You can use the Avp Action Profile to perform mediation at different directions (ingress, egress). You can modify both request and response messages. You can replace, append, prepend value AVP. The value may be static or dynamically retrieved from another AVP or can be extracted as substring from another AVP.

The following table describes the AVP actions that you can perform in Avp Action Profile:

Table 24: AVP Actions

Avp Action	Description
Remove Avp	Removes the AVP from the message.
Add with value	If the AVP is not present in the message, add the AVP with the value defined in CRD.
Add with Avp from Request	If the AVP is not present in the message, the AVP received from the request message is added to it.
Overwrite with value	Overwrite the AVP with the value defined in CRD.
Overwrite with Avp from Request	Overwrite the AVP with the AVP received from the request message
Add Prefix	Add prefix to the value of AVP.
Add Postfix	Add postfix to the value of AVP.
Remove Prefix	Remove prefix from the value of AVP.
Remove Postfix	Remove postfix from the value of AVP.

Figure 109: Avp Action Profile



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 110: APN Mapping Table

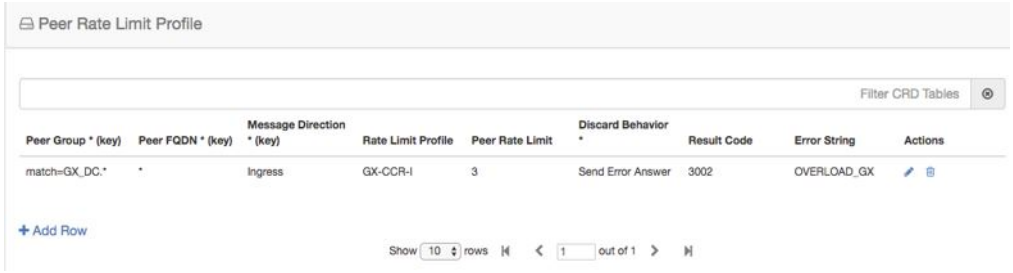
The Called-Station-Id input is case insensitive where it stores all the values in lower case. It converts the upper case entry to a lower case value and checks for a duplicate entry. If the input APN contains any duplicate value, it rejects the value with an error message.

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 111: Peer Rate Limit Profile



DOIC Profile

Use the DOIC Profile table to define the abatement action for Diameter messages in case of Diameter peer overload or congestion.

For more information about DOIC, see Configure Throttling of Diameter Messages Using DOIC, on page 26.

The following table describes the DOIC Profile table parameters:

Table 25: DOIC Profile

Fields	Description	Value
Egress Peer Group	Name of egress peer group.	Referenced from the Peer Group name in Peer Group SRK Mapping.
Message Class	Message classification. Priority P0 is considered for emergency message class. Hence, it cannot be configured in DOIC for throttling. In case abatement treatment is applicable for P0 message as per Loss Algorithm (with random number), the message is forwarded.	P1, P2, P3, P4

Fields	Description	Value
Abatement Action (output)	The abatement action to be taken by vDRA.	Divert, Forward, Drop

The following abatement actions are supported:

- Forward: Forward allows the message to be sent to the destination peer, even though the peer is overloaded.
- Divert: Messages are diverted to the non-congested secondary peer as found using Peer Group SRK Mapping. If the secondary peer is congested, the next non-congested peer is used. If all peers are congested, the messages are dropped.
- Drop: Message is throttled with error response 3002

  When a message is throttled, a default result-code of 3002 and default message "Throttled due to DOIC congestion" is sent.

  The error message can be configured in Error Result Code Profile table with the Error key as Doic Throttled/Dropped.

  Default Error Message is “3002: 012 - Throttled due to DOIC congestion”

Figure 112: DOIC Profile

Diameter Avp Dictionary

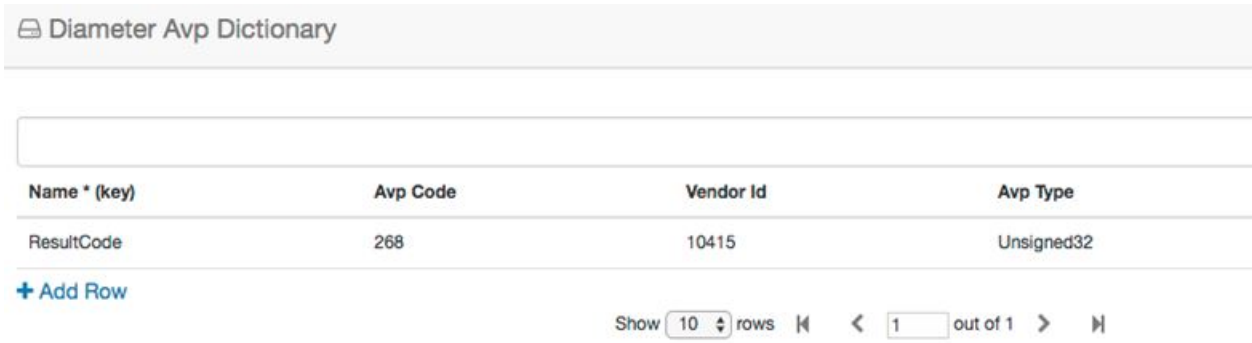
Use the Diameter Avp Dictionary CRD table to define the AVP name, AVP code, vendor ID, and the type.

The following table describes the parameters of the Diameter Avp Dictionary:

Table 26: Diameter Avp Dictionary

Field	Description
Name	Name of the AVP.
Avp Code	Code of the AVP.
Vendor Id	Vendor ID not supported for regx pattern ,wild card entries (*) and it is a mandatory field.
Avp Type	Type of AVP (Integer32, Unsigned32, Integer64, OctetString, UTF8String, Grouped)

Figure 113: Diameter Avp Dictionary



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
- Source IP - Specifies IP or IP range. For example, only subnet or only wildcard (*)

Note

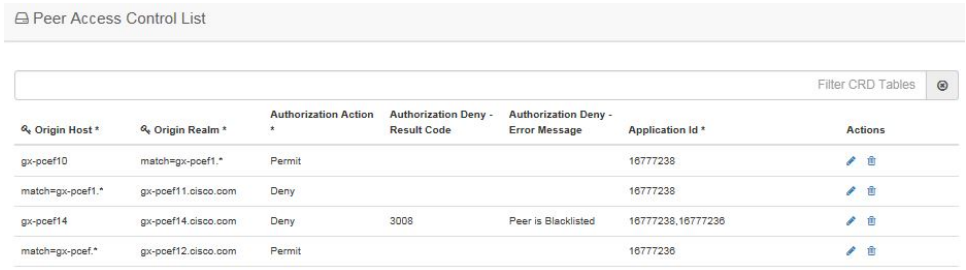
- When source IP is configured as IP / IP Range (that is, subnet) only and wildcard (*) then, values gets stored directly in the CRD without any modification.
- When the entered Source-IP matches wildcard(*) then, the IP address validation and the value to store in CRD is skipped.

- 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 114: Peer Access Control List

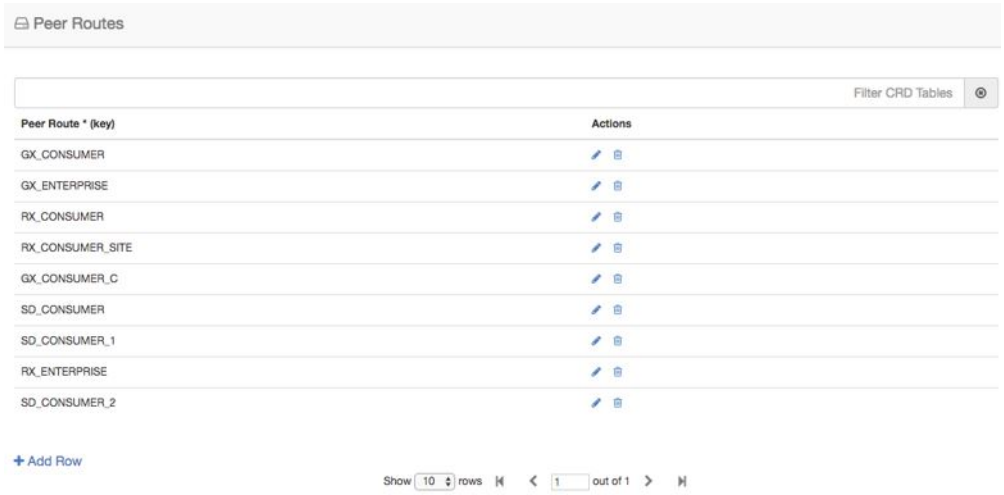


Peer Routes

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.

Note If multiple remote peers (having same FQDN) are connected with DRA and one remote peer goes down after sending a request then response message is also dropped. DRA does not send the request to any other remote peers (having same FQDN).

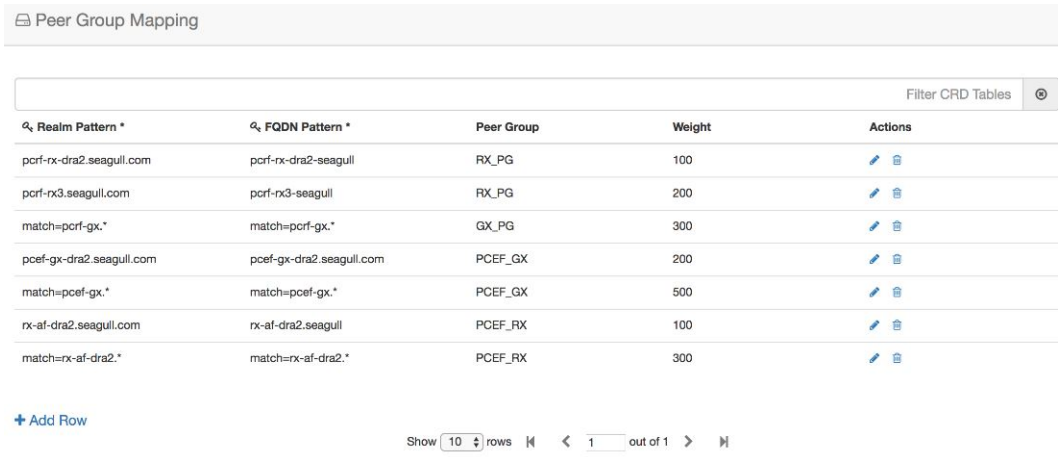
Figure 115: Peer Routes



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.

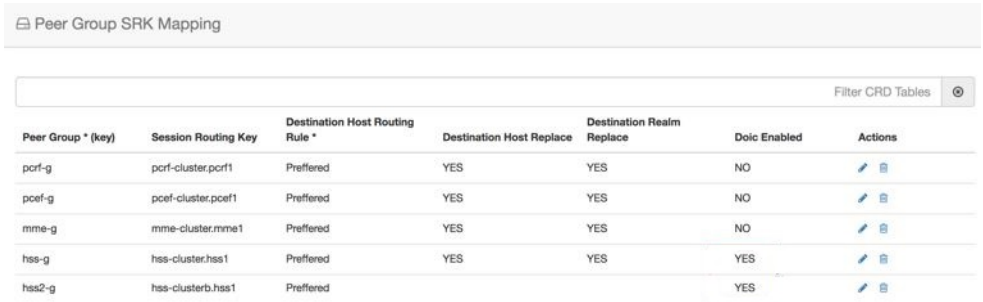
Figure 116: Peer Group Mapping



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.

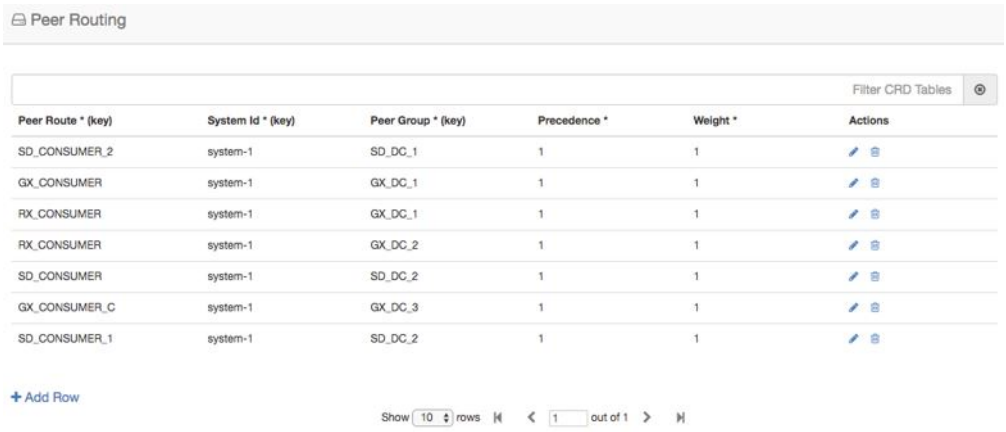
Figure 117: Peer Group SRK Mapping



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 118: Peer Routing



IPv6 Ranges System ID Mapping

Use this table to specify a range of IPv6 addresses and the primary and secondary vDRA system ID.

Figure 119: IPv6 Ranges System ID Mapping

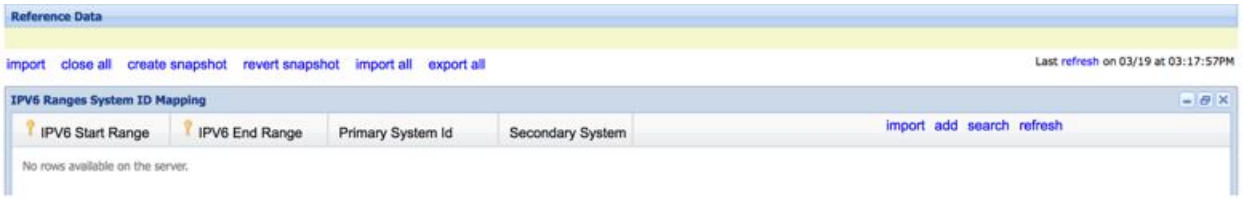


Table 27: IPv6 Ranges System ID Mapping Fields

Fields	Description
IPV6 Start Range	Indicates the start of range in ASCII.
IPV6 End Range	Indicates the end of range in ASCII.
Primary System ID	Mandatory field. Indicates the System ID of vDRA in a vDRA cluster to which the request can be relayed.
Secondary System ID	Secondary vDRA System ID where lookup can happen.

Note The ranges are expected to be mutually exclusive and unique. Verify the values when provisioning the same.

Binding Key Profile

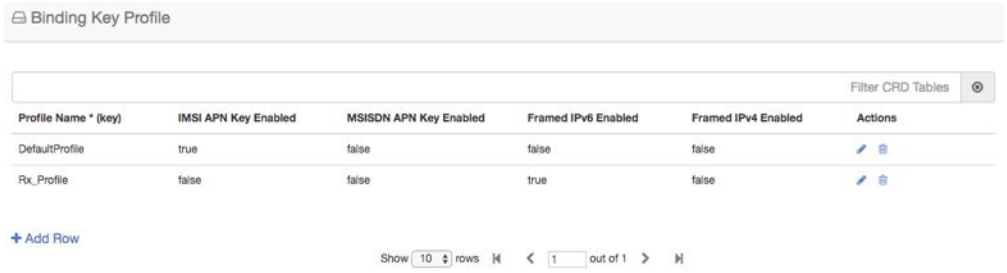
Important For routing to work in DRA, user must configure **AppId Key Profile Mapping** and **Binding Key Profile** tables.

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.

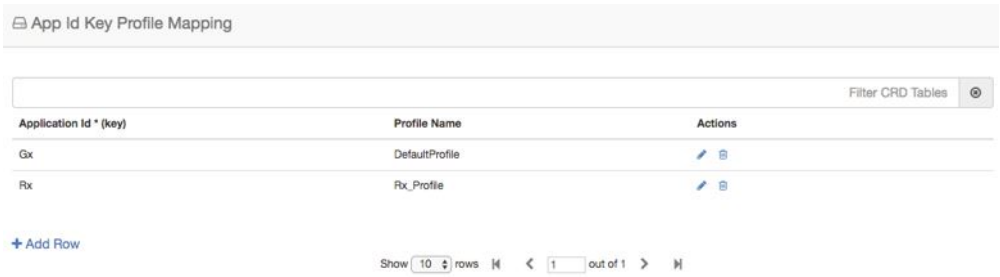
Figure 120: Binding Key Profile



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 121: 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 122: 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.

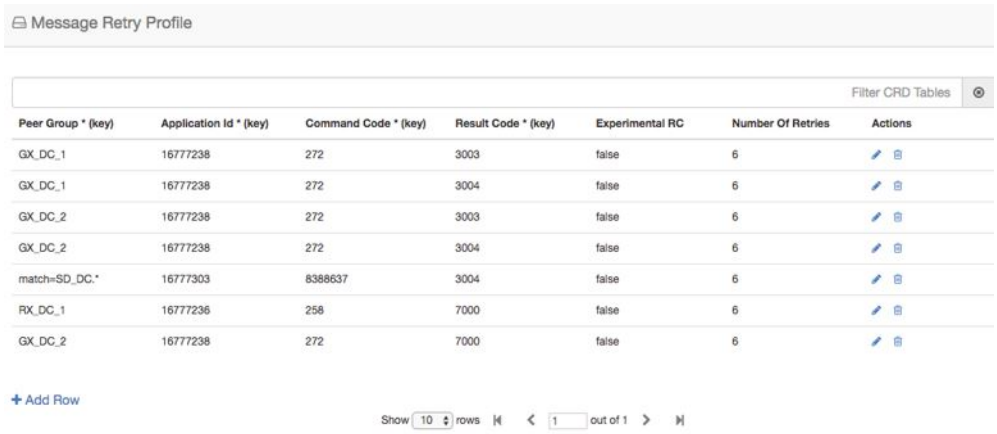
Configurable retry mechanism (i.e., number of retries) per:

- 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 123: Message Retry Profile - Control Center



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.

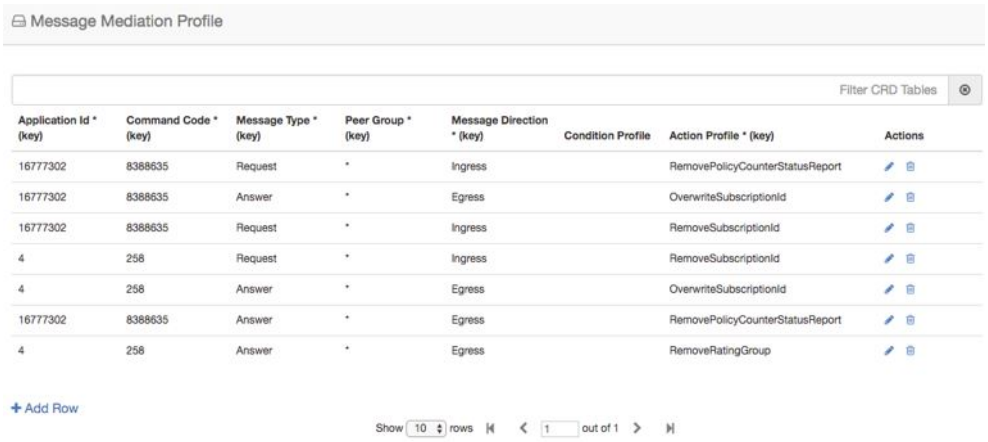
Refer to Message Retry Profile, on page 98 for configuration in **Search Table Group**.

Message Mediation Profile

CPS vDRA supports message mediation profile for following use cases:

- Store an AVP from the request and insert it into the answer. The answer is forwarded from an endpoint or an error generated by CPS vDRA. The known use case for this is storing the MSISDN from a request from the OCS (Sy SNR, Gy RAR) and inserting it in the answer to the OCS. The endpoint cannot handle all cases since the DRA can generate the error response in case of request timeout or inability to route the request to a peer. The MSISDN is in the Subscription-ID AVP.
- Remove an AVP from a request or answer.

Figure 124: Message Mediation Profile - Control Center



Peer Group Answer Timeout

CPS vDRA support for the following use cases:

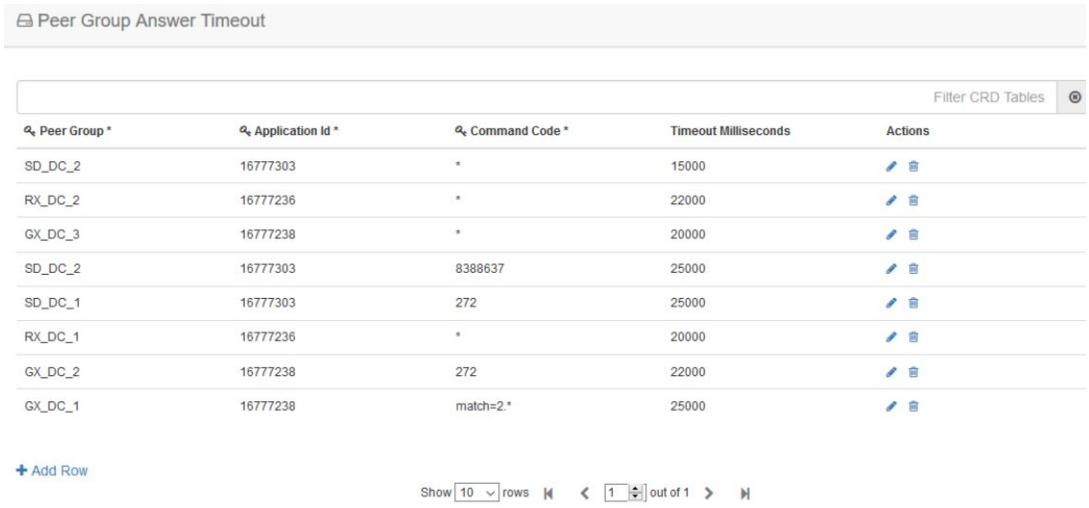
1. Configurable answer timeout for initial try and subsequent retries for the following parameters:
   - Application ID
   - Peer Group (to which request is sent)
   - Command code (to enable different timeouts for different Diameter commands)
   - Timeout value (in milliseconds)
2. Default value if 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 125: 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.

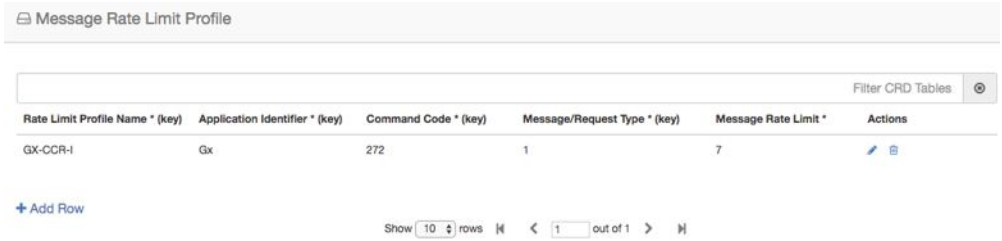
For Policy Builder related configuration, refer to Peer Group Answer Timeout, on page 101.

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 126: Message Rate Limit Profile



Dynamic Peer Rate Limit based on DB VM CPU Usage

Dynamic Peer Rate Throttling

Overload condition on binding databases occur when CCR-I or CCR-T bursts over one or more peer connections, thereby destabilizing the system. To overcome the DB overload condition, DRA supports the following mechanisms to protect the system from such an overload condition:

- Dynamically vary peer message rate limits (CCR-I/T) based on DB CPU load to enable better utilization of available DB capacity.
- Selectively throttle peer connections with traffic burst and continue processing of messages for peers with BAU traffic.

Configure Message Rate Limit Profile to throttle messages on the Director.and Rate limits for each message type in the profile. Dynamic rate limiting allows you to:

- Determine the available DB capacity and dynamically derive the rate limits.
- Configure preferred rate limits and apply dynamic throttling on configured values.

Limitations

Following are the limitations:

- Dynamic peer throttling feature applies only to peer connections with message rate limits configured on ingress direction.
- Throttling impacts all CCR-I/T messages on a peer connection irrespective of whether the binding corresponding to the overloaded DB is enabled.
- If the maximum throttling percentage configured in the profile does not restore the DB CPU load to normal values, the system remains in the same state until the overload condition clears.

Monitoring DB CPU Threshold

Important Configure the **binding shard-metadata-db-connection loadmetrics ip-address port** command to monitor CPU usage of all database VMs:

For more information on binding shard-metadata-db-connection, see the *binding shard-metadata-db-connection* section in the *CPS vDRA Operations Guide*.

For dynamic DB throttling, DB VM CPU statistics is collected periodically (every 10 seconds) and stored in metrics database. These statistics are cached by Workers and used to enforce dynamic DB throttling under high load conditions. But dynamically modifying rate limits on directors require that DB VM CPU statistics are available to directors.

Since the Workers periodically retrieve and cache the required statistics, Workers can monitor for threshold violations and notify Directors on the status. This eliminates the overhead on directors for retrieving the

required statistics. Directors then checks whether CPU thresholds are beached and start to reduce the rate limits of peer groups upon threshold breach.

Threshold monitoring and notification is processed on workers to:

- Perform the threshold monitoring on one of elected Workers. If the elected worker fails, another worker takes over the function.
- Ensure that the threshold status messages are received by directors at least once.
- Ensure that the Worker publishes duplicate messages using different Control Plane Redis connections.
- Not to notify the status of individual DB VM. This reduces the overhead on directors. Only following summary DB CPU status will be notified:
  - Threshold breach
  - Normal

  This results in one threshold status message from wWrker to Directors for every statistics polling period (10 seconds).

Since different peer groups are configured with different CPU thresholds, Worker uses the lowest CPU threshold to generate threshold breach event. The threshold is compared against the highest CPU utilization value among all the DB VMs. Upon threshold breach, corresponding event is published to Directors with highest observed CPU utilization. This ensures that Directors apply throttling based on CPU utilization of most loaded DB VM.

Dynamic Throttling Configuration

Dynamic throttling of message rate limits requires configuring the DB VM thresholds and corresponding throttling percentage to be applied. Specifying multiple thresholds with different throttling percentages enables varying the throttling based on severity of overload condition. For example, consider three different thresholds such as minor, major, critical, and apply throttling percentages increasing with severity.

During configuration of multiple thresholds, throttling applies corresponding to the highest threshold breached.

The following table describes the fields of Dynamic Throttling DB CPU Profile.

Table 28: Dynamic Throttling DB CPU Profile

Field	Description
Name	Name of the profile.
DB CPU Utilization Threshold	CPU utilization value in percentage beyond which throttling is applied.
Throttle Percentage	Throttling percentage that is applied on configured message rate limits. Rate limits are reduced by configured percentage.

To apply throttling profile to peers, use the following Dynamic Peer Rate Limit Profile parameters.

Table 29: Dynamic Peer Rate Limit Profile

Field	Description
Peer Group	Name of the origin peer group. Supports Wildcard values.
Peer FQDN	The origin peer FQDN.
Dynamic Throttling DB CPU Profile	Name of dynamic throttling DB CPU profile.

Dynamic throttling of message rate limits requires enabling of DRA Dynamic Peer Rate Limiter configuration in the Policy builder. If this configuration is not enabled through the Policy Builder, then you cannot view any DB CPU control messages from Worker to Director. For more information, see the Enable DRA Dynamic Peer Rate Limiter, on page 144 section.

Rules for Applying Dynamic Throttling for Peer Connections

Directors monitor DB CPU threshold events from Workers. Upon threshold breach, Directors evaluates the throttling profiles applied to peers against the CPU utilization value and reduces the rate limits based on the threshold breached.

When there is a threshold breach following rules apply for throttling for a peer:

- Dynamic rate limit throttling applies only to peers with message rate limit profile configured for CCR-I/CCR-T in ingress direction.
  - If throttling is applied to peer but the new threshold breached is higher than previous threshold, vDRA evaluates the threshold against the profile and applies throttling corresponding to the higher threshold. When revising the throttling, rate limits are always calculated based on the base or preferred rate limits.

    For example, consider the following configuration

    Message Rate limit:100

    Table 30: Example for Dynamic Throttling DB CPU Profile

Threshold	Throttling Percentage
50	20
55	30
60	40
65	50

    If the DB CPU utilization is in the range of 50-55%, rate limits gets throttled by 20% and effective rate limit is 80. If CPU utilization increases to 62%, then throttling of 50% is applied on configured rate limit of 100 and revised rate limit will be 50.

  - If throttling is applied to peer and the new threshold breached is same or lower than previous threshold, then throttling is upgraded to throttle percentage configured for the next higher threshold.

For example, consider the following configuration:

Message Rate Limit:100

Table 31: Example for Dynamic Throttling DB CPU Profile

Threshold	Throttling Percentage
50	20
55	30
60	40
65	50

When the DB VM CPU usage breaches 50%, throttling of 20% is applied and effective rate limit for all peer connections will be 80

Throttling reduces the load and reduce the CPU usage to < 50%

If the load continues, then CPU usage will increase and breach 50% threshold again. Because throttling for 50% threshold has already been applied, throttling increases to 30% corresponding to threshold 55%. Effective rate limits will be 70. If the threshold breach continues, then throttling increases to 40% (threshold 60) and 50% (threshold 65). Once maximum throttling of 50% has been applied, no further action takes place even if the threshold breach continues.

- Dynamic throttling applies to all messages irrespective of whether the corresponding binding profile has binding corresponding to overloaded DB enabled.
- If throttling is not applied for the peer, apply throttling corresponding to the threshold breached.

Throttling Reversal

Once the CPU utilization of overloaded DB VM(s) falls and remains below the threshold for a hold time, reversal of rate limit reduction is triggered. Configure the hold timer to trigger throttling reversal during normal load condition.

To prevent reversal action from creating a ping-pong effect, reversal will be performed in smaller steps. Throttling will be reversed based on the **Reversal Step in %** values configured in the **Policy Builder**. At each step, vDRA monitors DB load for hold time before applying the next step. If reversal causes threshold breach, roll back is performed.

Note It is recommended to use smaller steps for reversal.

Resilliency

Resilliency supports the following functions:

- **Worker Node Restart or Failure** During threshold monitoring and notification function of Worker node, if the Worker node fails or restarted, another worker takes over the function. The Semaphore

coordinates the role of threshold monitoring. Each Worker periodically (every 5 seconds) tries to acquire a lock on the Semaphore for a duration of 10 seconds. The Worker that acquires the lock performs the threshold monitoring function until it fails or is restarted:

- **Director Restart**: To ensure that new peer connections to a restarted Director are throttled at startup under existing DB overload condition, Director sends a threshold event query message to Worker. The Worker performing the threshold monitoring function responds to the Director with the latest threshold event. The Director processes the Query response,which initiates the query. If DB is overloaded throttling is applied and Query response gets ignored by rest of the Directors.

Enable DRA Dynamic Peer Rate Limiter

Configure the following parameters under DRA Configuration:

1. In the **Policy Builder**, on the right pane, click **DRA Configuration**.
2. In the **DRA Configuration** area, check the **Dra Dynamic Peer Rate Limiter** check box to enable the Dynamic throttling feature. By default this check box is disabled.
3. The following fields are available with default values, when you enable the **Dra Dynamic Peer Rate Limiter** parameter:
   - **Reversal Hold Time (Seconds)**: Specifies reversal hold time. Default is 30 seconds.
   - **Reversal Step in %** : Specifies the reversal step. Default is 20%.

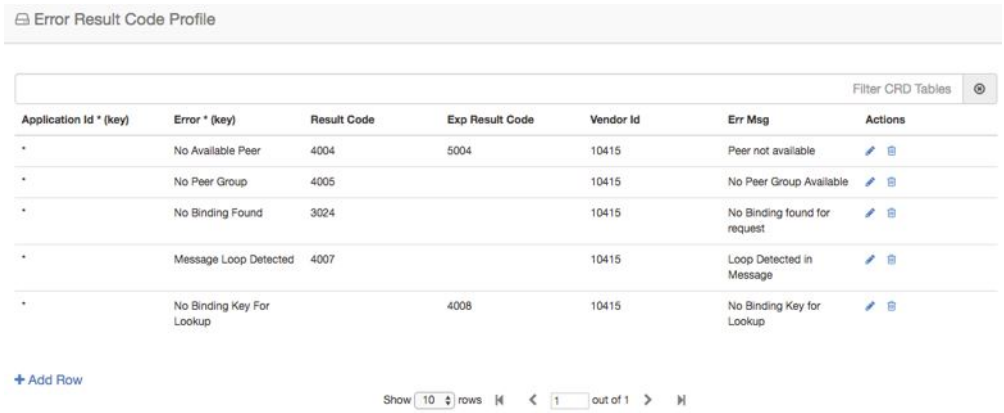
Note Specify the Reversal step within 100% and do not enter decimal values.

   - **Auto Apply Next Level Throttle**: Check the **Auto Apply Next Level Throttle** check box to dynamically apply next level throttling only if the DB CPU is in the configured range.

Error Result Code Profile

Sample CRD data looks like this:

Figure 127: Error Result Code Profile

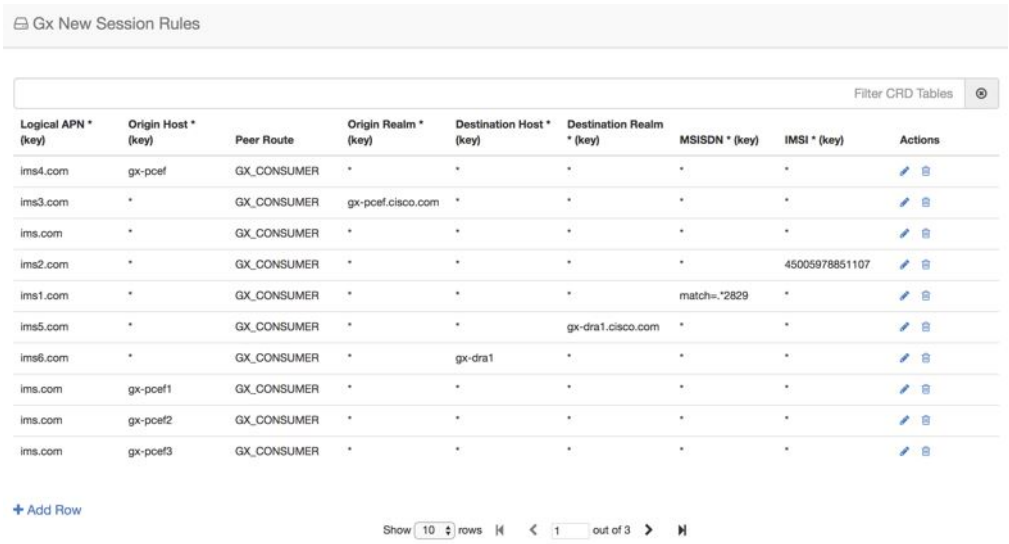


- For any CPS vDRA error or message timeout, CPS vDRAhas the ability to map the error to a Result-Code value and an error message string for the Error-Message AVP.
- Errors include things like "binding not found", "message timeout", "no peer connections".
- The Result Code value is sent in the Result-Code 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 128: Gx New Session Rules



Rest API Error Code Profile

You can configure the HTTP response error code (such as 4xx, 5xx) corresponding to each vDRA Rest API JSON error response code for the GET binding (for example imsi, imsiApn, msisdn, msisdnApn, ipv4, ipv6) Rest API.

This HTTP response code is used in the response for any GET binding Rest API request. If this CRD is not configured with HTTP response codes, then vDRA returns the default HTTP response status code.

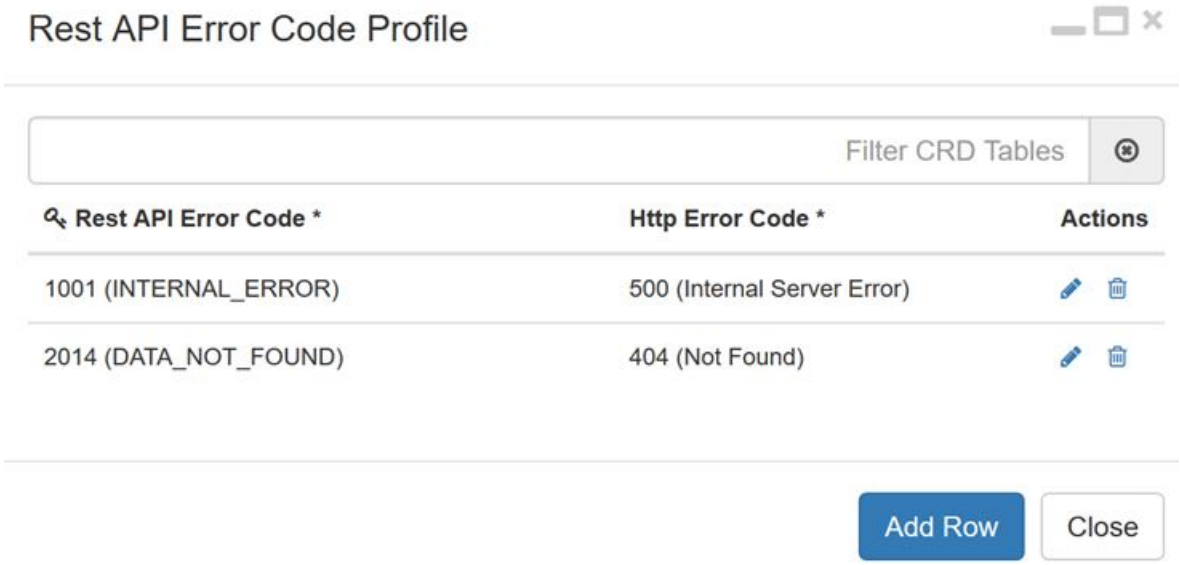
If you do not configure the Rest API HTTP Error Code in the CRD, vDRA uses the default HTTP error codes for GET binding Rest API. For a list of the default HTTP error codes, see the *CPS vDRA Troubleshooting Guide*.

The following table describes the mandatory parameters in the Rest API Error Code profile CRD:

Table 32: Rest API Error Code Profile

Parameter	Description
Rest API Error Code	vDRA Rest API JSON error response code for the GET binding (for example imsi, imsiApn, msisdn, msisdnApn, ipv4, ipv6) Rest API
Http Error Code	HTTP response error code (such as 4xx, 5xx) corresponding to each vDRA Rest API JSON error response code.

Figure 129: Rest API Error Code Profile



SLF Trigger Profile

In this table, there are three input keys: Application Id, Command Code and Destination Realm. If all these input keys are matched from the Diameter incoming requests and trigger condition for the SLF trigger table is matched, then CPS vDRA derives the Primary Lookup Type (IMSI/MSISDN) and SLF Destination Type as output of SLF trigger table. Then a query is made in the SLF Database using the Primary Lookup Type (IMSI/MSISDN) and SLF-Destination-Type.

This table is used in the case when the Diameter Request does not contain any "Destination-Host" AVP or, in case the "Destination-Host" AVP comes with the Diameter Host Name of CPS vDRA.

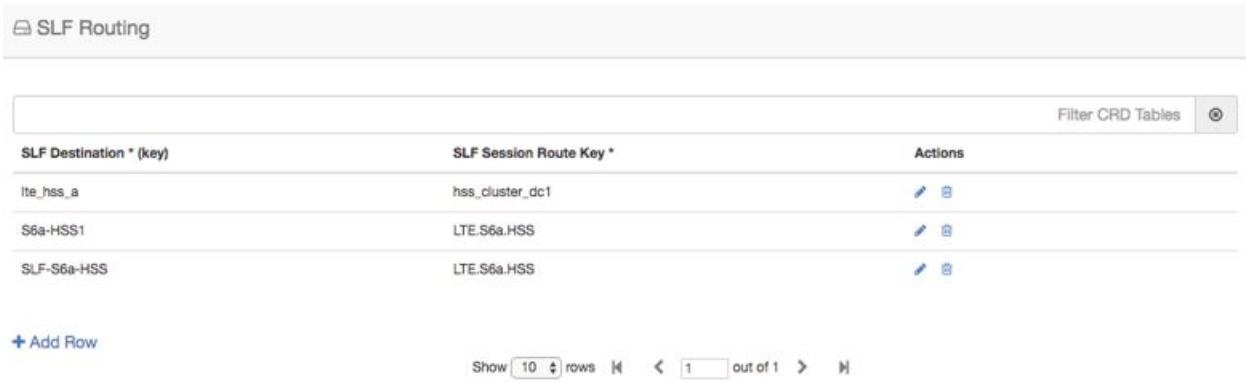
Figure 130: SLF Trigger Profile

Based on Application ID 16777251, Command Code 316 and Destination Realm of `ims.mnc286.mcc311.3gppnetwork.org`, Primary Lookup Type selected is IMSI and SLF Destination Type is selected as S6a-HSS. This Primary Lookup Type and SLF Destination Type is used to query SLF database for the configured lookup type.

SLF Routing

This table contains the mapping of SLF destination and the SRK of peer groups where the message could be routed. The SLF destination is derived from SLF subscriber database.

Figure 131: SLF Routing

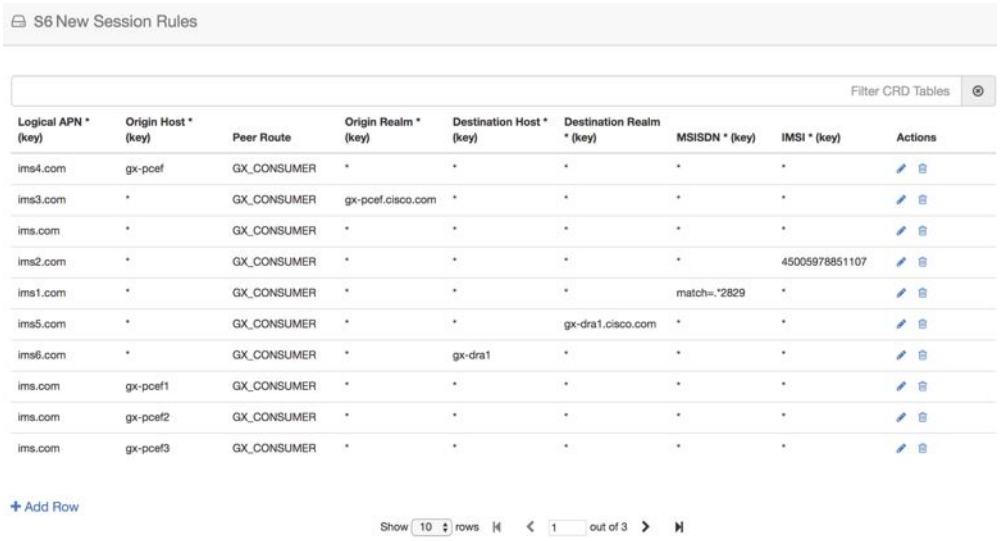


S6/Sh Table Driven Rules

This table is used for table driven routing of S6/Sh messages when the destination host is not available in the incoming request and there is no match SRK found in SLF Trigger table/SLF Mapping table. Keys used for deriving the peer route are Origin Host, Origin Realm, Destination Host, Destination Realm, MSISDN, IMSI and the output is Peer Route.

An S6 Table Driven Rules example configuration is given.

Figure 132: S6/Sh Table Driven 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.

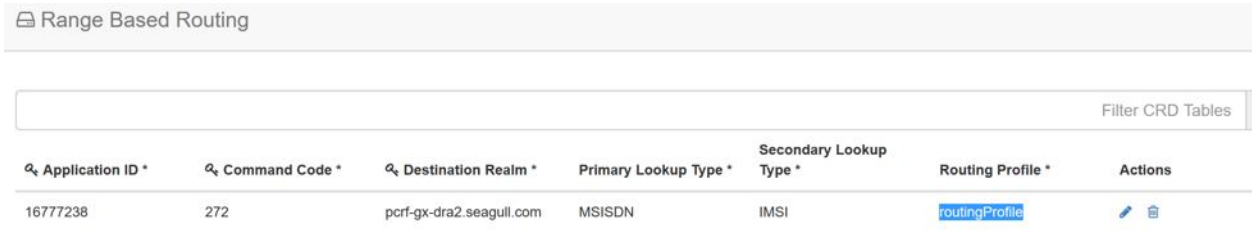
- vDRA checks whether the primary lookup type is IMSI or MSISDN and also checks whether the IMSI/MSISDN value present in the request matches against the range configured in CRD.
- The primary lookup type is evaluated first and if it fails, the secondary lookup type is evaluated.
- If primary lookup type 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 33: 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
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 133: Range Based Routing



IMSI Range

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

Table 34: 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 134: IMSI Range

MSISDN Range

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

Table 35: 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 135: MSISDN Range

Binding Key Profile Creation Map

The available fields are Boolean fields and you can edit them by selecting the check boxes.

Figure 136: Binding Key Profile Creation Map

- APN field supports both wildcard "*" and regex matches like "match=ims.*".

- For Binding Key profile Read Map, both Origin-Host and Origin-Realm support wildcard and regex match.
- Binding Profile and Binding Key Profile fields use values from the Profile name field in Binding Key Profile table. Define the profile using Binding Key Profile to create or read the tables.

The APN value is case insensitive which allows the input as a lower or upper case entry but converts the value to lower case and stores it in the CRD table.

Binding Key Profile Read Map

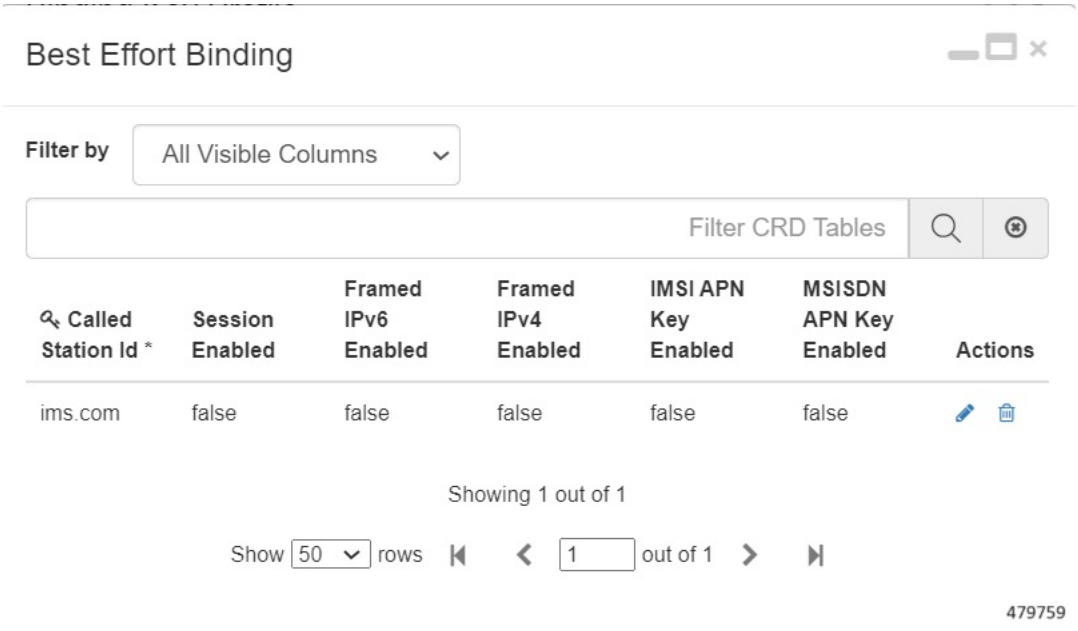
The available fields are Boolean fields and can be edited by selecting the check boxes.

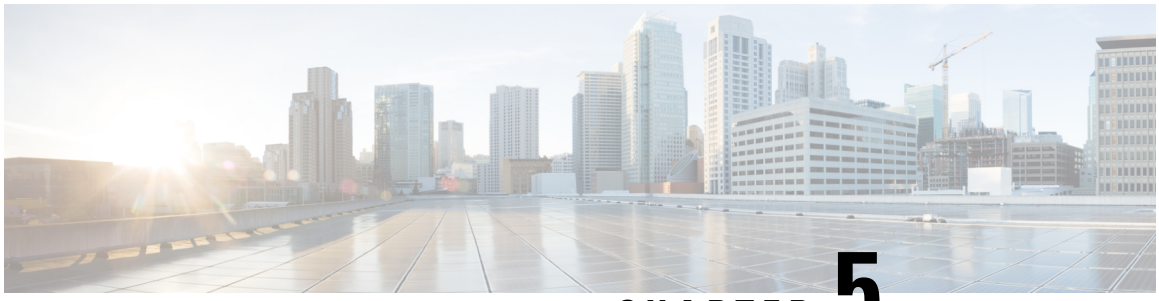
Figure 137: Binding Key Profile Read Map

Best Effort Binding

This table enables you to configure best effort binding on APN basis. The Called-Station-Id is an unique key value on the table that allows the values in lower case and accepts regular expressions.

Figure 138: Best Effort Binding





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 36: 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.
connection-timeout tcp	Idle timer for TCP connections in seconds. A connection is dropped if no traffic is seen for the duration of the timer. Default value is 30 seconds.
connection-timeout tcpfin	Timeout value in seconds for a connection after receiving a TCP FIN packet. A connection is dropped if no traffic is seen for the duration of the timer. Default value is 5 seconds.

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.

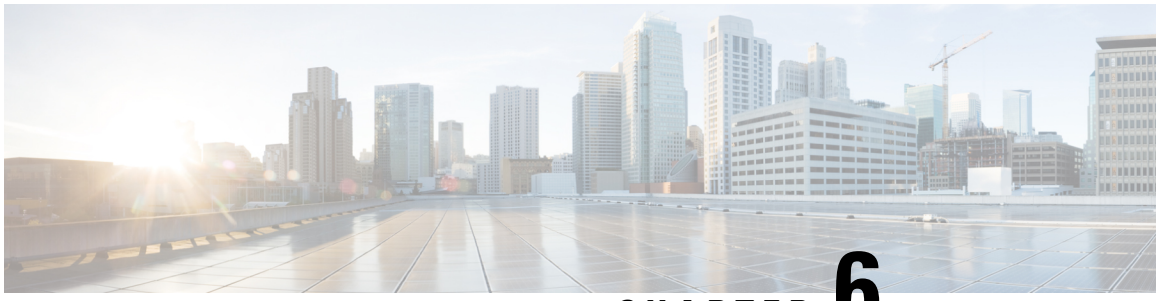
Example:

```
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)
```



CHAPTER 6

Dynamic Transport Selection based on Transaction or Origin Host

Important This feature has not been validated for all customer deployment scenarios. Please contact your Sales Account team for support.

Overview

Reliable and secure telecommunications systems are necessary for effectively managing national security incidents and emergencies. The National Security and Emergency Preparedness (NS/EP) is a set of voice, video, and data services that belong to services available from public packet-switched Service Providers and that provide priority services in support of NS/EP communications. The NS/EP communication systems

include landline, wireless, broadcast, cable television, radio, public safety systems, satellite communications, and the Internet.

Wireless Priority Services (WPS) is one of the NS/EP communications programs that provide personnel priority access and prioritized processing in all nationwide and several regional cellular networks, increasing the probability of call completion.

WPS users, also known as first responders, are responsible for the command and control functions that are critical to the management of response to national security and emergencies. The Evolved Packet Core supports WPS calls that are received from WPS users, In the Cisco Policy Suite, Diameter based interfaces such as Gx and RX that support policy and charging control function for subscribers, captures call from WPS users.

Whenever calls received from WPS users require a separate handling of control plane IP packets, DSCP marking is used. The DSCP marking helps in differentiating WPS and non-WPS users and always call from WPS user to a normal non-WPS user is treated as highest priority.

When the network carries the traffic for WPS users, all the network elements individually and collectively must adhere to the following conditions:

- **Prioritization of Control Plane Traffic**: WPS user's control plane traffic is prioritized over other subscribers between different Network Functions in the LTE Core.
- **Priority Levels**: P1, P2, and P3 are the three priority levels available for WPS users:
  - P1 and P2 users are identified in Home Subscriber System (HSS) and Gateway (GW)
  - Priority levels are used during session attach, bearer creation or during bearer modification
  - P1 and P2 WPS users are always treated as High Priority
  - When WPS -P1 user calls non-WPS user, non-WPS users and P3 WPS users are given high priority dynamically based on a call being placed
  - DSCP markings for prioritized user's control plane IP packets is marked with DSCP=47 while all other users control packets IP packets is marked with DSCP=32

Note In CPS 21.1.0 and later releases, only P1 Priority is supported.

- **Diameter Interfaces**:
  - P-GW, Policy Change Rule Function (PCRF) and Diameter Routing Agent (DRA) uses the configuration of Diameter interfaces such as Gx and Rx interfaces to support policy and charging control for subscribers.
  - P-GW and S-GW uses Non-diameter interfaces such as S5 and S1U interfaces.

Characteristics of Low and High Priority Channels for Diameter Based Interfaces

Low Priority channels indicate normal priority users and High Priority channels indicate Wireless Priority services users during Differentiated Services Code Point (DSCP) markings. The peer connections towards DRA for P-GW (Gx) is shown in the Figures.

Figure 139: High-Level Overview of Low and High Priority Channels over Gx Interface

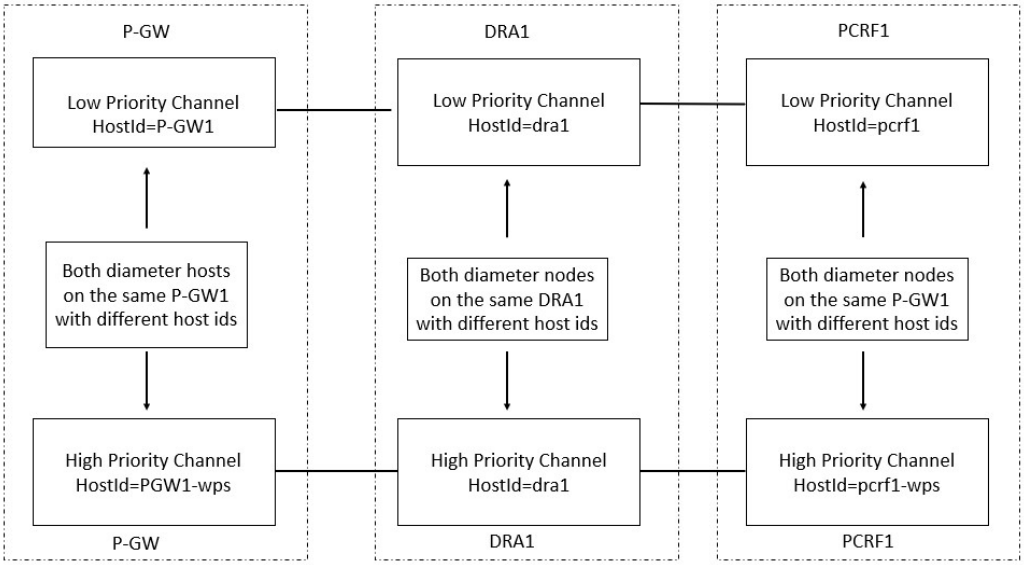


Figure 140: High-Level Overview of Low and High Priority Channels over Rx Interface

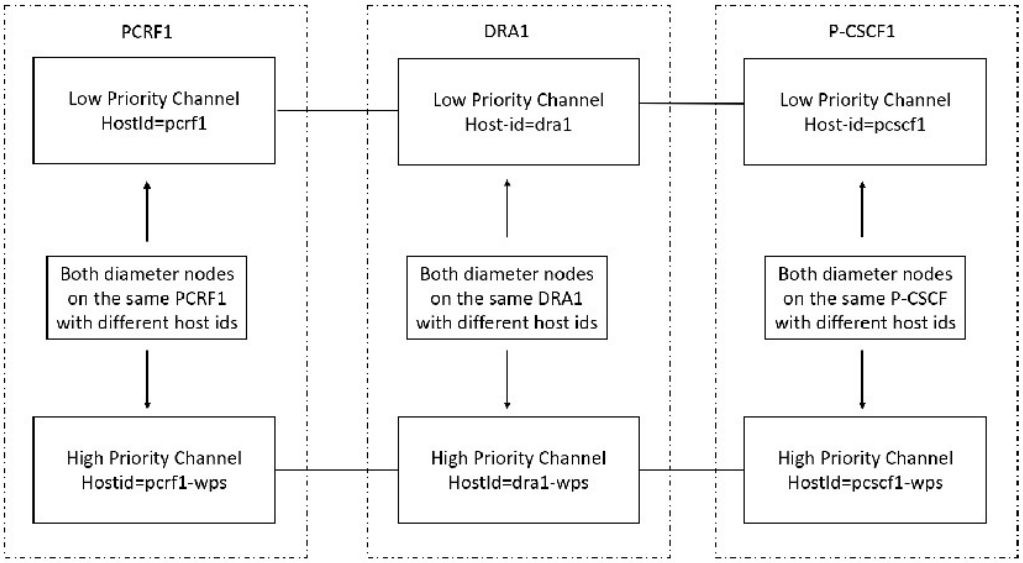


Table 37: Low and High Priority Channels based on Gx or Rx Interfaces

Priority Channel	Diameter Interfaces	IP Layer DSCP	TCP Connection over IP layer	Diameter Host FQDN
Low Priority	Gx/Rx	Equal to 32	32[5]	Not Modified For example, 0001-diamproxy. PGW-Gx', 'dra1', 'pcrf1
High Priority	Gx/Rx	Equal to 47	47	Specific to High Priority. For example, 0001-diamproxy. PGW-Gx-wps', 'dra1-wps', 'pcrf1-wps', 'pcscf1-wps'

[5] This channel is for non-WPS diameter messages but may carry WPS diameter messages in error scenarios, for example when all the WPS Peers are down.

Characteristics of Low Priority and High Priority Channels for S5 and S11 Interfaces

The S5 and S11 interfaces are GTPv2-based (which uses UDP as the transport protocol), Low and High Priority channels. Following table lists the characteristics.

Table 38: Low and High Priority Channels based on Rx Interfaces

Priority Channel	Diameter Interfaces	IP Layer DSCP	TCP Connection over IP layer	Diameter Host FQDN
Low Priority	S11 or S5	32	-	-
High Priority	S11 or S5	47	-	-

Dynamic Transport Selection based on Transaction or Origin Host on Policy Application Server

Transactions for certain Wireless Priority Service (WPS) user sessions are sent or received with different DSCP marking. You can create two sets of connections for Rx and Gx each with different DSCP marking.

Based on the Rx AAR, the Policy Application Server (PAS) chooses the right connection set for all subsequent transactions related to that session until the P-CSCF indicates a different priority. The DRA allows you to create the following policies for WPS users:

- DSCP marking for peer TCP connections: Use this function for WPS to forward WPS messages received from PAS. The WPS messages are treated as high priority in the network.
- Peer Group Message Class Mapping to configure message class for peer groups.
- Peer Group SRK Mapping.
- Peer Route for Priority Messages: Selects peers based on message priority.
- WPS Message Routing: PAS routes WPS messages over available WPS peer connections. If WPS connections are not available, then PAS routes WPS messages over available normal priority peer connection. PAS does not route non-WPS messages over WPS priority peer connection.
- Priority based destination host Re-routing: Supports rerouting of messages with destination-host based routing when WPS message is addressed to normal priority peer. This is allowed when a peer does not know the FQDN of high priority peer. This message rerouting can be enabled through option in Policy builder.
- DRA Relay Endpoint Message class mapping to configure message class for relay endpoints: Supports dedicated relay links for WPS messages with appropriate DSCP marking for the TCP connection. When forwarding WPS messages, PAS selects relay links matching the message priority.

DSCP Marking for Peer Connections

Wireless Priority Service (WPS) solution allows each of the peers connecting to PAS, establish a separate peer connection for normal and WPS messages with distinct origin FQDN. Peers uses distinct PAS endpoint for normal and WPS connection..

At the time of DSCP marking, PAS uses separate endpoints (inbound and outbound) for priority connections. Each diameter endpoint is configured with a DSCP value and all peer TCP connections to the endpoint is marked with the configured DSCP value.

Note Additional endpoints configured for WPS peer connections must use different VIPs as configuring same VIP for multiple endpoints can cause issues with load balancing by DRA distributor

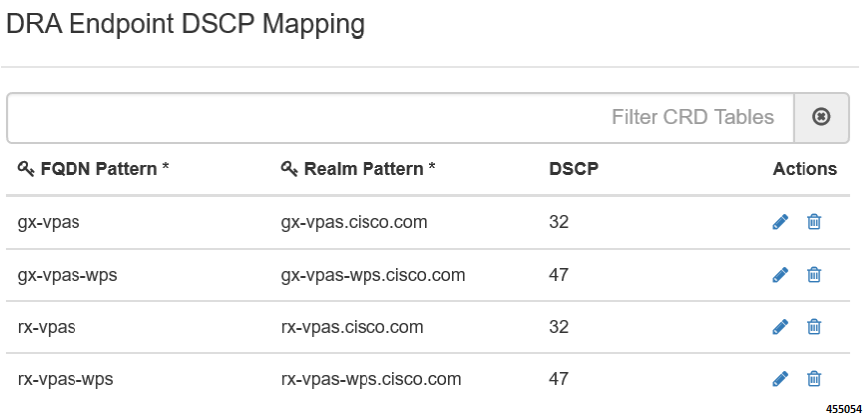
DSCP Mapping for DRA Endpoints

The DRA Endpoints DSCP Mapping allows you to configure different DSCP values for normal and WPS DRA endpoints.

If DSCP mapping is not configured for an endpoint, no action is performed. A default DSCP value is assigned to all endpoints by configuring a mapping using the wild card match as shown in the example:

```
{ FQDN Pattern= *, Realm Pattern= *, DSCP = <default value> }
```

Figure 141: DRA Endpoint DSCP Mapping



Enter or view the values the following field details:

Field	Description
FQDN Pattern	Displays an FQDN pattern of PAS endpoint.
Realm Pattern	Displays a Realm pattern of PAS endpoint.
DSCP	Displays the DSCP value for peer TCP connections to the DRA endpoint.
Actions	Allows you to perform either edit or delete actions.

The configured DSCP values are monitored using the below KPIs:

- peer_message_total
- peer_connection_status
- relay_message_total
- relay_peer_status

If it is not configued, the default DSCP value -1 is shown.

DSCP Marking in Diameter Stack

When creating diameter stack for each endpoint (inbound/outbound), stack manager reads DSCP mapping for endpoints and assigns appropriate value to stack instances. When a peer connection is established with diameter endpoint, stack sets the corresponding DSCP value for the TCP connection. All IP packets corresponding to messages that are forwarded by PAS (outbound) are marked with the specified DSCP value. P-GW, PCRF, and P-CSCF peers handle DSCP marking for inbound messages.

When you update a DSCP mapping in Custom Reference Data, then stack manager detects the configuration change and defines the new DSCP for all new connections. DRA does not change the already set DSCP value without resetting peer connections.

Priority-based Peer Group

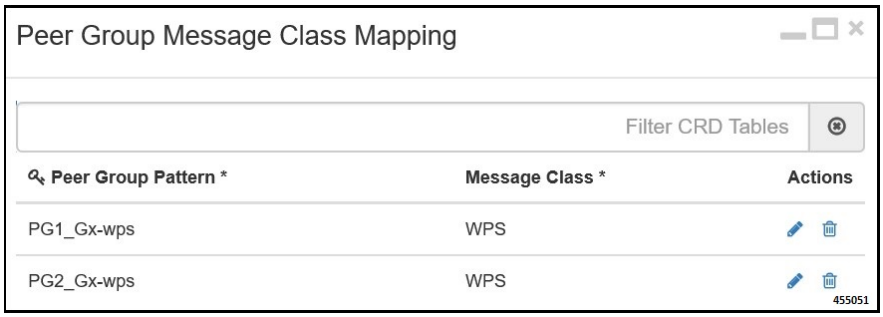
You can group all normal peers (non-WPS peers) under normal peer group and all priority WPS peers under WPS peer group. This way of grouping is useful in routing normal messages to normal peers and WPS messages to priority WPS peers.

Enter the following details in the **Peer Group Message Class Mapping** to configure priority for all peer groups.

Table 39: Priority-based Peer Group

Field	Description
Peer Group Pattern	Enter a peer group pattern name.
Message Class	Enter a Peer traffic and message class value for a peer group.

Figure 142: Assign Peer Groups to WPS Class



You can edit the values to Map peer groups to specific message class. DRA supports mapping peer groups only to message class of WPS. Configure the mapping only when a peer group is restricted to a specific message class. If the mapping is not configured for a peer group, then that peer group is designated by default to handle all message classes.

Peer Group for SRK Mapping

Map Peer groups for WPS and default peer connection to the same Session Routing Key (SRK). This enables DRA to select peer groups of default message class if WPS peer groups are down. In priority based SRK routing, DRA checks for active peers from matching WPS peer groups and fallback to peer groups of default message class if there are no active WPS peersMap Peer groups for WPS and default peer connection to the same Session Routing Key (SRK). This enables DRA to select peer groups of default message class if WPS peer groups are down. In priority based SRK routing, DRA checks for active peers from matching WPS peer groups and fallback to peer groups of default message class if there are no active WPS peers

Mapping of logically related peer groups under same SRK is useful in route selection for below two scenarios.

- For non-WPS users, when Gx session gets created in normal peer and gets updated to WPS session during Rx AAR calls, then DRA checks for message priority Attribute Value Pair (AVP) in AAR request and route the WPS message to WPS peer.

- For WPS users, if there are no active WPS peers in local/remote, then DRA routes WPS messages to normal peers as fallback option.

  The following figure illustrates a sample CRD "Peer Group SRK Mapping" configurations to support normal and WPS peer groups.

Table 40: Peer Group SRK Routing

Field	Description
Peer Group	Enter a Peer group name
Session Routing Key	Enter the Session Routing key information of Peer group.
Destination Host Routing Rule	Specify one of the following Destination Host Routing Rule: • Only • Never • Preferred • Preferred for Update Requests[6]
Destination Host Replace	Choose YES or NO to enable or disable destination host replace.

[6] When Destination-Host routing policy for PCRF Gx peer group is set to **Preferred for Update Requests**, then PAS routes the request as follows:

- PAS resets Destination-Host rule as **Preferred** and route Gx CCR-I request using Table-driven routing when destination host is set as DRA endpoints or destination host is null. If Gx CCR-I request contains destination host AVP as PCRF endpoint, then PAS routes Gx CCR-I request using destination host routing.
- PAS resets Destination-Host rule as **Preferred** and route Gx CCR-U requests using destination host routing and fallback to SRK routing only when PAS failed to find the same PCRF host mentioned in Destination-Host AVP
- PAS resets Destination-Host rule as **Never** and routes Gx CCR-T request using SRK routing

Note
DRA routes second Gx CCR-T request to different PCRF host. PAS supports the new Destination-Host routing rule "Preferred for Update Requests" only for PCRF Gx peer groups. If the new Destination-Host routing rule "Preferred for Update Requests" is configured for any non-Gx peer groups, then PAS sets default Destination-Host rule as "Preferred" for route selection.

Peer Routing for Priority Message

Map WPS peer route with WPS peer groups and default peer route with default peer groups. If you want to map fallback from WPS peer to default peers, then define WPS peer route to both WPS peer groups and default peer groups. Peer route mapped to WPS peer group takes higher precedence and peer route mapped

to default peer group takes lower precedence. If WPS peer groups are inactive, this precedence is used in Table Driven Routing to select WPS peer groups and fallback to default peer groups.

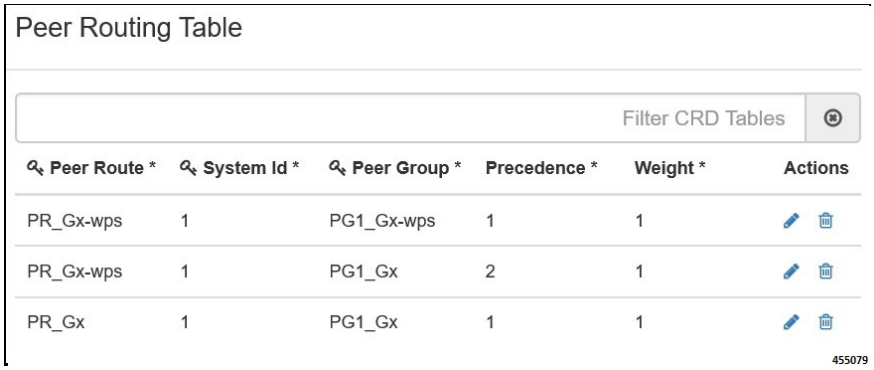
In case of default peer route, only default peer group is mapped and no fallback to WPS peer group is allowed, if default peer groups are inactive.

Enter the Peer Route List details to configure default and WPS peer groups and Peer Routing Table configurations to support default and WPS peer groups.

Table 41: Peer Routing List

Field	Description
Peer Route	Enter a Peer route list. For example, PR_Gx for normal user and PR_Gx-wps for WPS user.
Actions	Allows you to perform either edit or delete actions.

Figure 143: View Peer Route Table Details



Enter Peer Route Table details to configure default WPS peer groups.

Table 42: Peer Routing List

Field	Description
Peer Route	Enter a Peer route name for WPS peer and Non-WPS peer.
System Id	Enter System Id of the same vDRA.
Peer group	Enter a Peer group name for WPS peer and Non-WPS peer.
Precedence	Enter the priority value for selecting WPS peer groups and fallback to default peer groups, if WPS peer groups are inactive.
Weight	Enter the weightage of peer route.
Actions	Allows you to perform either edit or delete actions.

WPS Message Routing

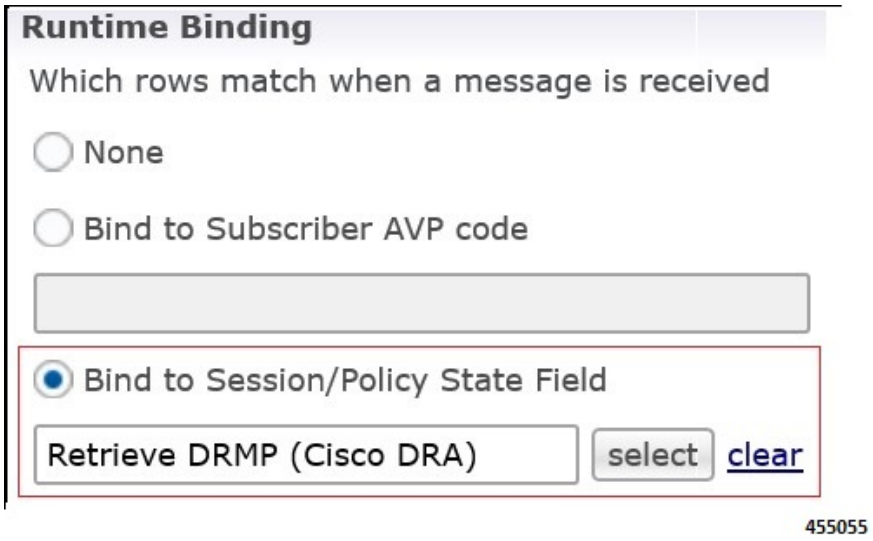
PAS routes WPS messages over available WPS peer connections. If WPS connections are not available, then PAS routes WPS messages over available normal priority peer connection. PAS does not route non-WPS messages over WPS priority peer connection.

Table Driven Routing

Table Driven Routing for WPS messages uses Origin Host or Realm. This is because WPS peers use distinct FQDN and Realm, which ensures that all messages received on WPS peer connections are routed to WPS peers.

To route WPS messages received on default peer connection, include message priorities, and configure appropriate rules matching priority in the routing table. DRA provides the option to retrieve DRMP AVP and Message Class for incoming messages and map them to Gx Routing table.

Figure 144: Runtime Binding



Use the following procedure to specify the Runtime Binding details:

1. In CPS DRA, navigate to **Policy Builder** .
2. Click,**Reference Data** and then choose **Systems**.
3. Click **Custom Reference Data Tables**.
4. In the Runtime Binding area, specify the following details.

Field	Description
None	If no rows require matching when a message is received, click the **None** radio button.

Field	Description
Bind to Subscriber AVP Code	Click the **Bind to Subscriber AVP Code** radio button to retrieve values from an AVP for the subscriber. Also, values from a session AVP or a Policy Derived AVP is displayed.
Bind Session/Policy State Field	Click the **Bind Session/Policy State Field** radio button to select the value from a Policy State Data Retriever, which retrieves a single value for a session. Choose any one of the following DRMP options to indicate after adding new Message priority AVP: • Retrieve DRMP (Cisco DRA): Displays value from DRMP message priority AVP of incoming messages. • Retrieve Message Class (Cisco DRA): Maps Message Class profile with message class type as WPS_P0. This ensures that DRA is not throttling any WPS CCR-I messages

5. In the Gx New Session Rules area, view the following details. Maps peer routes with Origin host, Origin realm and DRMP AVP for normal and WPS peer groups.

Field	Description
Logical APN	The name of the logical Access Point (APN).
Origin Host	The origin host FQDN
Peer Route	Peer route to select active peer groups.
Origin Realm	The Origin Realm.
Destination Host	Displays the destination host FQDN.
Destination Realm	Displays the destination Realm.
MSISDN	Displays the MSISDN subscriber identification attribute.
IMSI	Displays the IMSI subscriber identification attribute.
DRMP/Message Class	Displays either DRMP or custom message class AVP value.

Table Driven Workflow

Note Since **DRMP/Message Class** field is the primary key-in Gx routing table, any old exported CRD dumps should be imported to DRA before adding the new **DRMP/Message Class** field. After adding new **DRMP/Message Class** field, you must manually update these fields with default values (*) for all existing entries in Gx Routing CRD. Otherwise, DRA does not show any values for these new fields and might cause routing failure for Gx CCR-I messages.

The following lists explain the table driven workflow:

- DRA selects peer route based on the match in table row of Table-Driven routing. The matched peer route can have N number of peer groups with different or same precedence.
- DRA creates a sorted list to maintain all peer groups in higher to lower precedence order. It traverses through sorted peer group precedence list and checks whether the WPS peer group is active in local or remote site.
- If local WPS peer group is in active state, then DRA selects local WPS peer group.
- If local WPS peer group is in inactive state, then fallback to remote WPS peer group.
- Only when both local and remote WPS peer groups are in inactive state, then DRA will check for normal peer group in same peer route.
- If DRA fails to find any active normal peer in matched peer route, then DRA sends 3002 ERR response.
- In case of fallback for normal messages, since, DRA does not select peers from WPS peer group at any time, normal peer route should be mapped with only normal peer groups and WPS peer route should be mapped with WPS peer group as first precedence and if needed, then map normal peer group as second precedence.

Destination Host Routing

During Destination Host routing, DRA performs Destination host routing only when destination host AVP is present and it is not pointing to DRA endpoint FQDN. The following conditions apply:

- If destination host peer is active, then it will route the request to that destination peer.
- If destination peer is inactive, then it will fall back to SRK routing. In SRK routing, DRA can select active peer from different groups where all these groups are mapped to same SRK.

DRA gives preference to destination host priority and forwards the message to set destination host peer. P-GW and PCRF sets correct WPS destination host in request based on the message priority.

Supporting Fallback of WPS Gx RAR, Rx RAR, and Rx ASR Messages to non-WPS Peer

vDRA supports fallback of WPS Gx RAR, Rx RAR and Rx ASR messages to non-WPS peer when there is no active WPS peer available locally or globally. Through WPS Suffix keyword configuration, you can identify two connections such as WPS or non-WPS that belong to P-GW /P-CSCF. Also, through configuration, vDRA

controls suffix based destination host routing on peer group. For example, you can enable this fallback for Cisco ASR but not for affirmed P-GW.

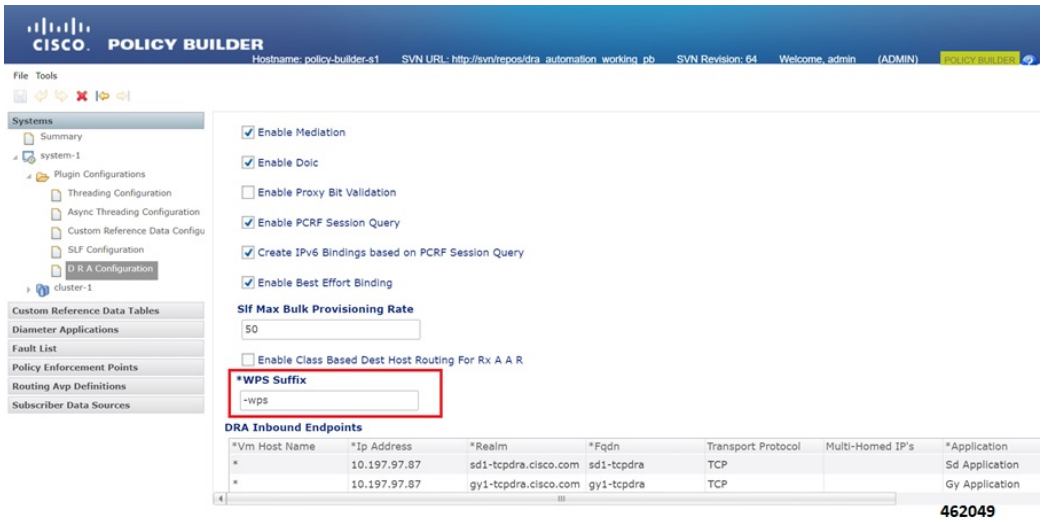
Note Suffix based destination host routing is applicable only for Gx RAR, Rx RAR and Rx ASR messages. This feature is disabled if there is no WPS suffix configured in policy builder and if there are no rows configured in “Suffix Based Dest Host Routing" CRD.

Configuring WPS Suffix in Policy Builder

Use the following procedure to configure WPS suffix keyword in the Policy builder.

1. Log in to the **Policy Builder**
2. Choose **Systems** > **Plugin Configuration** > **DRA Configuration** .
3. In the **WPS Suffix** field, enter a WPS suffix to use across all nodes. For example, you can configure a keyword “-wps” as suffix.

Figure 145: Configure WPS Suffix



Ensure to configure the same suffix keyword across P-GW and PCRF nodes. Otherwise, route failure might occur in WPS Gx/Rx RAR and Rx ASR fallback routing.

For more information about DRA features, see the *DRA Feature* section in the *CPS vDRA Configuration Guide*.

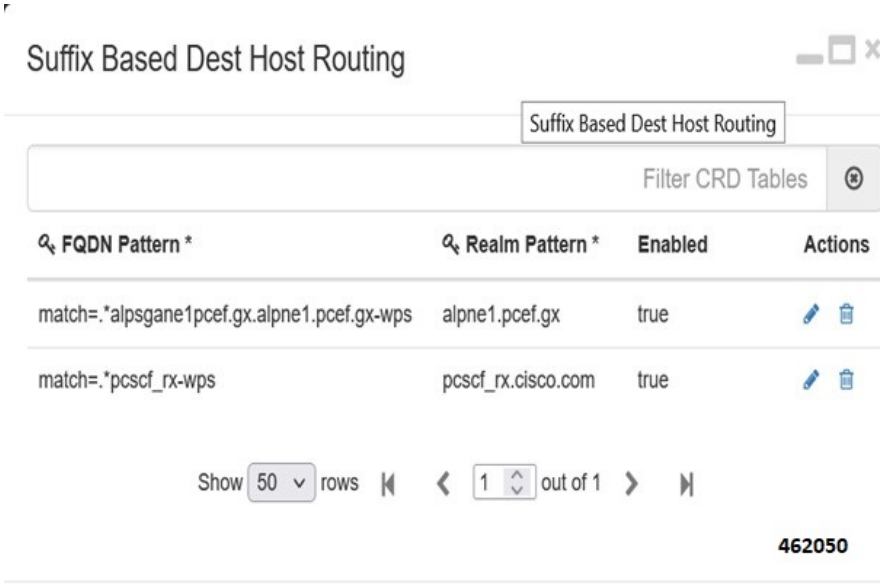
Enabling Suffix Based Dest Host Routing

vDRA supports fallback of WPS Gx RAR, Rx RAR and Rx ASR messages to non-WPS peers only for the peers configured in the **Suffix Based Dest Host Routing** CRD table. vDRA uses this CRD only when normal dest-host routing and SRK routing failed for WPS Gx RAR, Rx RAR and Rx ASR messages.

Note vDRA does not use this CRD for routing of any non-WPS Gx RAR, Rx RAR and Rx ASR messages.

To configure WPS PGW/P-CSCF peers, create a new CRD **Suffix Based Dest Host Routing** as shown in the figure.

Figure 146: Create new Suffix Based dDestination Host Routing CRD



Handling Fallback

Based on WPS3B configurations, all nodes (P-GW/PCRF/PCSCF) have separate FQDN for WPS peers. This WPS FQDN is different from non-WPS FQDN and is suffixed with configured new keyword. For example, if configured suffix keyword is “-wps”, then FQDN have pgw/pgw-wps, pcscf/pcscf-wps, pcrf-gx/pcrf-gx-wps, pcrf-rx/pcrf-rx-wps.

DRA performs WPS Gx RAR, Rx RAR and Rx ASR fallback to non-WPS peer in following ways.

1. After vDRA receives Gx RAR, Rx RAR, and Rx ASR messages with destination host:
   - vDRA tries to route the message using destination host routing and then SRK routing.
   - If the WPS peer mentioned in Dest-Host AVP is inactive and fails to find active route using SRK routing, then vDRA uses “Suffix Based Dest Host“ routing.
2. vDRA gets configured suffix keyword from the Policy Builder and checks whether destination host mentioned in the Dest-Host AVP have the same suffix. The following actions happen:
   - If both are same, vDRA truncates the suffix keyword from destination host and sends the WPS message to non-WPS destination host.

Note After truncating configured suffix from destination host, vDRA first checks for local non-WPS peer and then checks for remote non-WPS peer only if there are no active local peers.

- If both are different, vDRA skips “Suffix Based Dest Host” routing and tries Table Driven routing.
- If vDRA fails to find any active route, then it sends timeout message to PCRF.

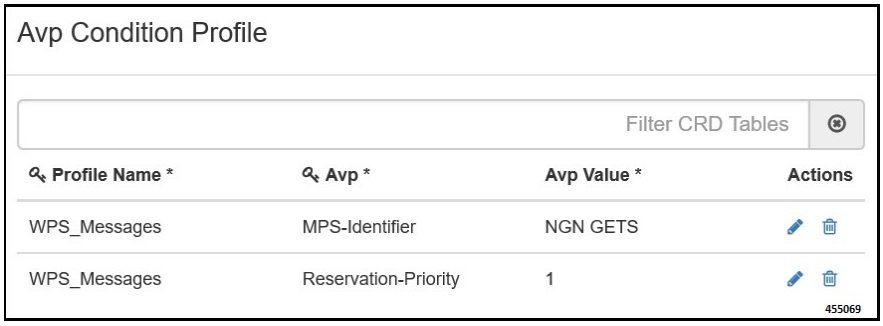
Binding-based Routing

During Binding-based routing, DRA routes Rx messages to the correct peer based on the message priority. For non-WPS users, create a Gx session in normal peers and Rx session in WPS peers. DRA routes all priority Rx messages to WPS peers. For WPS users, DRA routes all priority to both Gx and Rx on WPS peers.

To identify WPS messages, in the AVP Condition Profile area:

- Configure either MPS-Identifier AVP or Reservation Priority in Diameter AVP Dictionary and then map the AVP Condition Profile Custom Reference Data with correct values.
- Configure both MPS-Identifier AVP or Reservation Priority AVPs. Make sure that both the AVPs are mapped under the same AVP Condition Profile.

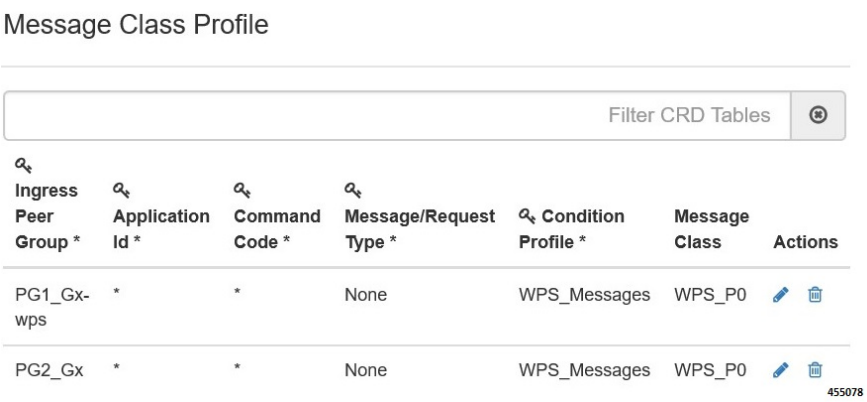
Figure 147: AVP Condition Profile mapping for both MPS-Identifier and Reservation-Priority AVPs



In the Message Class Profile area, DRA classifies new message classes for WPS.

- If Message classes profile is defined then, message classes include message class and message priority. For example, If new message class is WPS_PO, then this indicates WPS message of priority P0.
- If message class profile is not defined, then DRA classifies the message as default message class.

Figure 148: Message Class Profile



Enter the following Message Class Profile Parameters.

Table 43: Message Class Profile Parameters

Field	Description
Ingress Peer group	The name of the Ingress Peer group.
Application ID	The application identifier.
Command Code	Displays diameter message command code
Message/Request Type	Displays either a message or type of the request.
Condition Profile	Displays a condition profile for either WPS or non-WPS messages.
Message Class	Displays the message class and message priority for WPS user.
Actions	Allows you to perform either edit or delete actions.

SRK Routing

The following workflow explains Session Routing Key (SRK) function for WPS:

- For non-WPS users, when Gx session gets created in normal peer and gets updated to WPS session during Rx AAR calls, DRA checks for message priority AVP in AAR request and route the WPS message to peers under WPS message class.
- For WPS users, if there are no active WPS peers, DRA routes WPS messages to default message class peers as fallback option.
- DRA does not route normal messages to WPS peers at any point of time.
- When DRA receives WPS messages, it checks for WPS peers in local site and then fallback to remote site.
- If DRA is not able to find any active peers of WPS message class in local/remote, then it considers default message class peers to route WPS messages.

- DRA checks for default message class peers in local and then fallback to remote site.
- When DRA accepts any peer connection, Peer Manager accepts the peer connection and it will read peer priority from the Peer Group Message Class Mapping and updates the matching message class priority in DRA peer up/down control message.
- Peer Manager publishes the control message to local/global control plane.
- Local/global control plane thread publishes the same to local/global topology manager.
- The local topology manager updates peer message class in peer endpoint state.

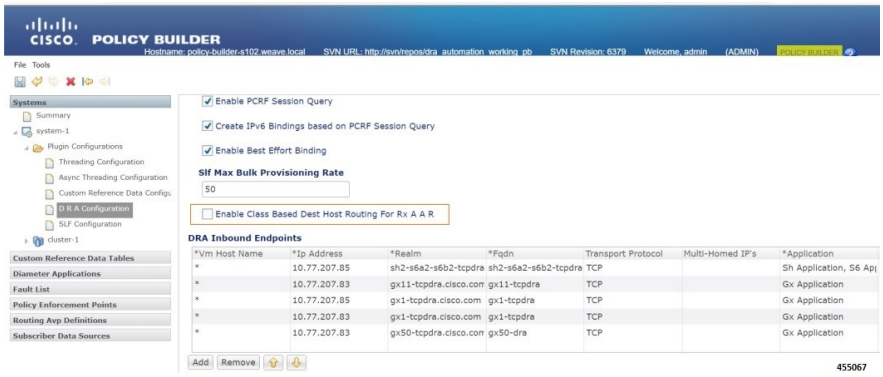
Priority-based Destination Host Rerouting

Priority-based Destination Host Rerouting feature enables DRA to identify mismatch between message class of destination host and AAR and reroutes to PCRF matching the message class. This feature is disabled by default.

In DRA, you can configure message priority AVPs in **AVP Condition Profile**, **Message Class Profile**, and map the profile to message class **WPS_P0**. The **WPS** is used as message class and **P0** indicates to message priority. For more information, refer *Binding based Routing* section.

Once you check the **Enable Class Based Dest Host Routing for Rx AAR** check box, DRA checks the message class of Rx AAR and compares with message class configured for destination peer group. If there is a mismatch in the message class and destination peer group has SRK configured, DRA performs SRK routing instead of destination host routing. SRK routing routes the message to peer matching the message class.

Figure 149: Priority-based Destination Host Rerouting Parameter



PCRF Session Query for WPS Messages

In Diameter Routing Agent (DRA), use WPS PCRF or non-WPS PCRF REST API endpoints, to send WPS PCRF session query to PCRFs, and receive Session Route Key (SRK) information for WPS Rx AAR messages.

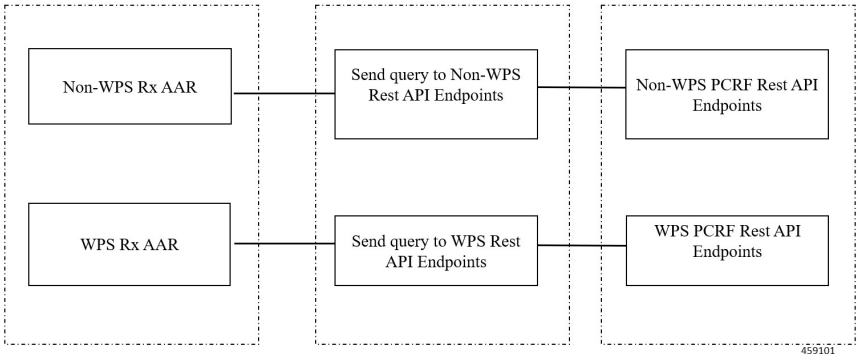
DRA allows the following functionalities:

- Separate Rest API endpoints configuration to support WPS IPv6 binding queries.
- WPS Rest API endpoints to query IPv6 binding for all WPS messages.

- PCRF session query for WPS Rx AAR messages is set with configured DSCP value as 47.
- PCRF session query for non-WPS RX AAR messages is set with configured DSCP value as 32.
- Query parameter class=wps will be added for all WPS PCRF session queries.
- Fallback to non-WPS PCRF Rest API Endpoints. This is to get session route key information for WPS Rx AAR messages when there is any issue in sending query with WPS PCRF Rest API endpoints or WPS PCRF Rest API endpoints not configured.

Architecture

The following illustration depicts WPS and non-WPS IPv6 binding queries.



Processing IPv6 Binding Query for WPS Messages

In DRA, PCRF contains new set of Rest API endpoints to serve IPv6 binding query for WPS messages. Based on message priority of Rx AAR messages, DRA selects configured Rest API endpoints as follows:

- If the incoming Rx AAR message is WPS, then DRA selects Rest API endpoints from **PCRF Session Query Peers** that are configured as message class **WPS**. All WPS Rest API endpoints are marked as **WPS** in **PCRF Peer Group Message Class Mapping** CRD.
- DRA adds **class=wps** as query parameter to the payload to indicate message class as WPS to PCRF for internal prioritization. This is applicable only for WPS PCRF session queries.
- If DRA fails to send PCRF session query using WPS PCRF Rest API endpoints or WPS PCRF Rest API endpoints are not configured, then DRA will fallback to non-WPS PCRF Rest API endpoints to send high priority WPS PCRF session query
- DRA adds **class=wps** as query parameter to the payload even at the time of fallback to non-WPS PCRF Rest API endpoints.
- If the incoming Rx AAR message is non-WPS, then DRA selects non-WPS Rest API endpoints from **PCRF Session Query Peers** to send session query.

Note DRA does not use high priority WPS PCRF Rest API endpoints to send any PCRF session query for non-WPS messages.

Configuring IPv6 Binding Query Messages

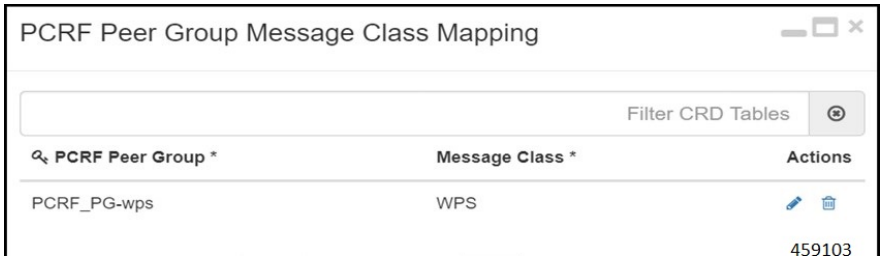
Use the following steps to configure IPV6 binding query messages:

1. Configure separate Rest API Endpoints for WPS IPv6 binding queries:

   In the **PCRF Session Query Peers** CRD, create a separate PCRF group for all WPS-related Rest API endpoints. For more information about configuring the REST API parameters for Rx AAR fallback routing, see the section *PCRF Session Query Peers* in the *Policy Builder Configuration* chapter.

2. Configure message class as WPS for WPS PCRF peer groups. Create a new CRD **PCRF Peer Group Message Class Mapping** as shown in the Figure. Only PCRF peer groups created in CRD are**PCRF Session Query Peers** configured in this new CRD.

   Figure 150: PCRF Peer Group Message Class Mapping Configuration



3. Configure DSCP value for PCRF REST API Endpoints. Use linux command iptables/ip6tables to configure DSCP value as 47 for WPS PCRF Rest API endpoints and DSCP value as 32 for non-WPS PCRF Rest API endpoints.

```
For example:
If non-WPS PCRF Rest API endpoint is http://10.197.99.271:9000/dra/api/bindings and WPS
 PCRF Rest API endpoint is
http://10.197.99.271:9001/dra/api/bindings, then DSCP value can be set as:
iptables -t mangle -A PREROUTING -d 10.197.99.271 -p tcp --dport 9000 -j TOS --set-tos
128
iptables -t mangle -A PREROUTING -d 10.197.99.271 -p tcp --dport 9001 -j TOS --set-tos
188
```

Note Left shift DSCP value by 2 to get TOS value.

(47 << 2) = 188

(32 << 2) = 128

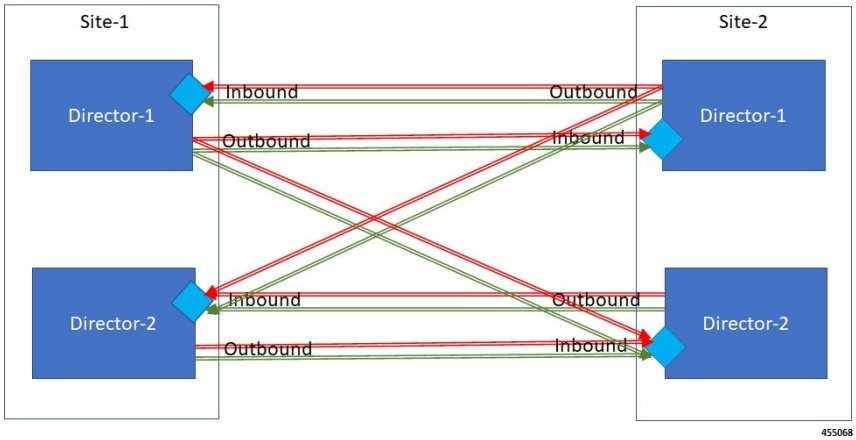
Priority based Relay Routing

Policy Application Server supports Priority-based relay routing for WPS messages through the following mechanism:

- Relay Endpoints for Priority Messages
- Advertising Relay Link Priority in Control Plane

- Relay Link Selection based on Priority

Figure 151: : Priority-based Relay Routing Flow



Relay Endpoints for Priority Messages

As part of the WPS feature, Dimameter Routing Agent (DRA) has two relay connections to relay peers. One relay link for WPS relay messages and another one for normal relay messages.

To route WPS relay messages to WPS relay link, DRA updates relay logic as follows

- Configures two relay endpoints in the Policy Builder. One for WPS relay messages and another one for normal relay messages. Normal and WPS relay endpoints are configured with unique FQDN.
- Configures relay endpoints message class in **DRA Relay Endpoint Message Class Mapping** for WPS.

Advertising Relay Link Priority in Control Plane

The remote relay system identifies the relay priority and routes WPS messages through WPS relay link.

Note Changes to control plane message is backward compatible. Hence, systems that do not support message class priority can ignore this Advertising Relay link priority function.

Selecting Relay Link based on Priority

The following procedural steps describes how DRA selects the relay link based on priority:

1. Creates queue for remote relay endpoints using remote SystemId and remote relay message class. This is mainly to differentiate between WPS relay and default relay queue. DRA maintains two outbound and inbound connections with relay system:

- WPS relay connection to forward WPS messages
- Default relay connection to forward default messages

During routing whenever remote peers are selected to route messages, remote relay endpoint is selected based on the message class of the selected destination host peer or incoming request.

2. DRA compares the message class of message and destination host peer with message class of relay system to select the WPS relay endpoint, to forward only WPS messages, and selects default relay endpoint to forward default messages.
3. When DRA receives control plane messages for remote relay system, it stores the relay systemId in topology Manager along with relay message class.
4. DRA routes WPS relay messages over normal relay endpoint link only when WPS endpoint is down.