CHAPTER 8

Diameter

Diameter is a networking protocol which is derived from RADIUS protocol. It is considered to be the next generation Authentication, Authorization, and Accounting (AAA) protocol. This is the other core protocol used in the IP Multimedia Subsystem (IMS) architecture for IMS Entities to exchange AAA related information. Cisco Prime AR supports Diameter Applications based on the Diameter Base Protocol defined in RFC 6733.

Diameter is composed of a base protocol and a set of applications which allows it to extend its services to new access technologies. The base protocol provides basic mechanisms for reliable transport, message delivery, and error handling. Each application is defined by an application identifier and associated with commands. Each command is defined with mandatory Attribute Value Pairs (AVPs) and non-mandatory AVPs including vendor-specific AVPs.

The base protocol must be used in conjunction with a Diameter application. Each application relies on the services of the base protocol to support a specific type of network access.

The following is the list of applications supported by Cisco Prime AR:

- Diameter Network Access Server Application (NASREQ, RFC 4005)
- Diameter Base Accounting (RFC 6733)

This chapter contains the following sections:

- Before You Begin
- Diameter Server Startup Log
- Diameter Stack Level Messages
- Configuring Authentication and Authorization for Diameter
- Configuring Diameter Accounting
- Configuring the Diameter Application in Cisco Prime AR
- Writing Diameter Application in Cisco Prime AR
- Diameter Routing Agent
- Support for SCTP including Multihoming

Before You Begin

Each Diameter application is identified by the unique application id and the set of commands associated with it and application specific AVPs. Cisco Prime AR requires addition of Diameter BaseApplication, NASREQApplication, and BaseAccounting Application to perform Diameter Authentication and Accounting.

To configure the BaseApplication, NASREQApplication, and BaseAccounting Application in Cisco Prime AR, follow the below steps in order from **/opt/CSCOar/bin/** directory:

Step 1 Execute the below command to import Diameter BaseApplication AVPs:

./aregcmd -s -f /cisco-ar/examples/cli/add-BaseProtocolAVPs.rc

Step 2 Execute the below command to import Diameter BASEApplication:

./aregcmd -s -f /cisco-ar/examples/cli/add-BaseApplication.rc

Step 3 Execute the below command to import Diameter NASREQApplication AVPs:

./aregcmd -s -f /cisco-ar/examples/cli/add-NASREQAVPs.rc

Step 4 Execute the below command to import Diameter NASREQApplication:

./aregcmd -s -f /cisco-ar/examples/cli/add-NASREQApplication.rc

Step 5 Execute the below command to import Diameter BaseAccounting application:

./aregcmd -s -f /cisco-ar/examples/cli/add-BaseAccountingApplication.rc

For registering NASREQApplication, configure **/Radius/Advanced/Diameter/General/AuthApplicationIdList** to 1.

For registering BaseAccounting, configure **/Radius/Advanced/Diameter/General/AcctApplicationIdList** to 3.

Diameter Server Startup Log

When Cisco Prime AR starts, Diameter server also starts.

The log file shows the following:

```
09/30/2012  6:38:47.419 name/radius/1 Info Server 0 Diameter Server Started
09/30/2012  6:38:47.437 name/radius/1 Info Protocol 0  Starting diameter core
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0             Product : Cisco
Prime Access Registrar
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0             Version : 6
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0           Vendor Id : 0
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0    Auth Application : 0
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0    Auth Application : 1
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0    Acct Application : 3
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0          Dictionary :
/cisco-ar/conf/diadictionary.xml
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0            Identity :
10.81.79.43
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0               Realm : abc.com
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0          TCP Listen : 3868
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0         SCTP Listen : 3868
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0   Watch-Dog timeout : 500
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0            Use IPv6 : 0
```

```
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0 Re-transmission Int : 8
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0    Max Re-trans Int : 3
09/30/2012  6:38:47.447 name/radius/1 Info Protocol 0    Recv Buffer Size : 20480
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0      Hostnames Used :
10.81.79.43
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0  Dumping Peer Table
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0       Expire Time 1
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0     Peer : Host = 10.77.240.54,
Port = 3868, Server-Identity =  , Server-Realm =  , TLS = 0
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0     Peer : Host = 10.77.240.53,
Port = 3868, Server-Identity =  , Server-Realm =  , TLS = 0
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0   Dumping Route Table
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0             Exp Time : 0
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0               Route  : Realm =
dia.com, Action = 2, Redirect-Usage = 0
09/30/2012  6:38:47.448 name/radius/1 Info Protocol 0
Application Id=1, Vendor=0
09/30/2012  6:38:47.449 name/radius/1 Info Protocol 0                         Server
= 10.77.240.53, metric = 2
09/30/2012  6:38:47.449 name/radius/1 Info Protocol 0  Auth Stateful Auth : stateful
09/30/2012  6:38:47.449 name/radius/1 Info Protocol 0    Auth Session(T) : 30
09/30/2012  6:38:47.449 name/radius/1 Info Protocol 0    Auth Lifetime(T) : 360
09/30/2012  6:38:47.449 name/radius/1 Info Protocol 0       Auth Grace(T) : 30
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0       Auth Abort(T) : 20
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0     Acct Session(T) : 30
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0    Acct Interim Int : 5
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0      Acct Real-Time : 0
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0           Debug Log : enabled
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0           Trace Log : enabled
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0            Info Log : enabled
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0         Console Log : enabled
09/30/2012  6:38:47.450 name/radius/1 Info Protocol 0          Syslog Log : disabled
```

Diameter Stack Level Messages

The following are the stack level messages that are exchanged between the diameter peers:

- Capabilities Exchange Message
- Watchdog Message

Capabilities Exchange Message

When Diameter peers establish a transport connection to Cisco Prime AR, they will exchange the Capabilities Exchange messages. This message allows the discovery of a peer's identity and its capabilities (protocol version number, supported Diameter applications, security mechanisms, etc.)

The log file shows the following:

```
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0 Peer Capabilities
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0             Hostname :
10.77.240.54
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0               Realm : cisco.in
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0             Host IP : type=1,
10.77.240.150
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0            VendorId : 11
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0         Product Name : Cisco
Prime Access Registrar
```

```
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0           Orig State :
1094807040
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0  Auth Application Id : 1
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0  Acct Application Id : 3
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0           Inband Sec : 0
09/30/2012  6:38:57.525 name/radius/1 Info Protocol 0         Firmware Ver : 1
09/30/2012  6:38:57.526 name/radius/1 Info Protocol 0 Statistics for the peer
10.77.240.54 is sent with code value 505
09/30/2012  6:38:57.526 name/radius/1 Info Protocol 0 Statistics for the peer
10.77.240.54 is sent with code value 508
```

Watchdog Message

The Device-Watchdog-Request and Device-Watchdog-Answer messages are used to proactively detect transport failures. Device Watchdog message time interval is configurable in Cisco Prime AR.

The log file shows the following:

```
10/07/2012 10:44:15.143: Log: Watchdog msg from [thomas.cisco.com.cisco1.com],
state=1254936955, time=1254937455.
```

Terminating Diameter User Session

In Cisco Prime AR, Diameter Session management is independent of Diameter accounting. Session termination is conveyed by a specific Session-Termination message rather than an Accounting Stop message.

The log file shows the following:

```
10/07/2012 10:37:39.299: Log:  *** Session termination request received ***
10/07/2012 10:37:39.299: Log:  Session id=thomas.cisco.com.cisco1.com;{;H;
10/07/2012 10:37:39.299: Log:  From Host: thomas.cisco.com
10/07/2012 10:37:39.299: Log:  From Realm: cisco1.com
10/07/2012 10:37:39.299: Log:  From User: invaliduser
10/07/2012 10:37:39.299: Log:  Termination Cause: 5003
10/07/2012 10:37:39.299: Log:  Auth Application Id: 1
10/07/2012 10:37:39.300: Log: Session disconnect for Session-Id:
thomas.cisco.com.cisco1.com;1254936955;124122
```

Note In Cisco Prime AR, session management does not support Diameter messages. Diameter base stack (RFC 6733) will maintain the session.

Configuring Authentication and Authorization for Diameter

This section describes how to configure Cisco Prime AR to perform authentication and authorization and how to configure a local service and userlist.

See Table 4-7 for more information on Diameter client properties.

This section contains the following topics:

- Configuring Local Authentication and Authorization
- Configuring External Authentication Service

Configuring Local Authentication and Authorization

In Diameter, an AA-Request packet is a request for authentication and authorization. Authentication checks username and password credentials, while authorization typically involves returning the correct information to allow the service a user is authorized to have. Cisco Prime AR performs AA and returns the appropriate Diameter attributes in an AA-Answer packet.

For adding a Diameter peer in Cisco Prime AR, configure a new entry in the clients (including Policy and Charging Rules Functions (PCRF), Home Subscriber Servers (HSS), Mobility Management Entities (MME), Online Charging Systems (OCS), and others) and remote server object.

The following shows an example configuration for adding a Diameter peer (NAS/Client) in Cisco Prime AR.

```
Name = diameter-client
Description =
Protocol = diameter
HostName = 10.81.79.42
 Vendor =
IncomingScript~ =
OutgoingScript~ =
Port = 3868
SCTP-Enabled = FALSE

[ //localhost/Radius/Services/dia-local ]
Name = dia-local
Description =
Type = diameter
Realm = abc.com
Role = Local
AuthenticationService = local-users
AccountingService = local-file
Peers/

 DefaultAuthenticationService~ = dia-local
 DefaultAuthorizationService~ = dia-local
```

Note You should restart the Cisco Prime AR server if you change any Diameter specific configuration.

See Table 4-7 and Table 4-21 for more details.

Configuring a Local Service and UserList

See "Configuring a Local Service and UserList" section on page 6-1 for more information on how to configure a local service and user list.

The following messages are logged in the trace file at the time of authenticating a valid user:

```
06/03/2012  7:26:00.138: P195: Diameter Packet received from 10.81.79.42
06/03/2012  7:26:00.139: P195: Trace of Diameter-Access-Request packet
06/03/2012  7:26:00.139: P195:    Session-Id = .;1096298391;16
06/03/2012  7:26:00.139: P195:    Auth-Application-Id = 1
06/03/2012  7:26:00.139: P195:    Origin-Host = 10.81.79.42
06/03/2012  7:26:00.139: P195:    Origin-Realm = abc1.com
06/03/2012  7:26:00.139: P195:    Destination-Realm = abc.com
06/03/2012  7:26:00.139: P195:    Auth-Request-Type = 3
06/03/2012  7:26:00.139: P195:    User-Name = bob
06/03/2012  7:26:00.139: P195: Using Client: murdoch
06/03/2012  7:26:00.139: P195: Authenticating and Authorizing with Service dia-local
```

```
06/03/2012  7:26:00.139: P195: Calling Service local-users for authentication and
authorization
06/03/2012  7:26:00.139: P195: Getting User bob's UserRecord from UserList Default
06/03/2012  7:26:00.140: P195: user list user bob's password matches
06/03/2012  7:26:00.140: P195: Trace of Diameter-Access-Accept
06/03/2012  7:26:00.140: P195:    Auth-Application-Id = 1
06/03/2012  7:26:00.140: P195:    User-Name = bob
06/03/2012  7:26:00.140: P195:    Auth-Request-Type = 3
06/03/2012  7:26:00.140: P195:    Result-Code = 2001
```

The following messages are logged in the trace file at the time of authenticating an invalid user:

```
10/02/2012 22:54:58.512: P74: Diameter Packet received from 10.81.79.42
10/02/2012 22:54:58.512: P74: Trace of Diameter-Access-Request packet
10/02/2012 22:54:58.512: P74:    Session-Id = .;1096298391;1
10/02/2012 22:54:58.512: P74:    Auth-Application-Id = 1
10/02/2012 22:54:58.512: P74:    Auth-Request-Type = 3
10/02/2012 22:54:58.512: P74:    Destination-Realm = abc.com
10/02/2012 22:54:58.512: P74:    Origin-Host = 10.81.79.42
10/02/2012 22:54:58.512: P74:    Origin-Realm = abc1.com
10/02/2012 22:54:58.512: P74:    User-Name = james
10/02/2012 22:54:58.512: P74: Tracing the packet after running the rules and policies
10/02/2012 22:54:58.512: P74: Trace of Diameter-Access-Request packet
10/02/2012 22:54:58.512: P74:    Session-Id = .;1096298391;1
10/02/2012 22:54:58.512: P74:    Auth-Application-Id = 1
10/02/2012 22:54:58.512: P74:    Auth-Request-Type = 3
10/02/2012 22:54:58.512: P74:    Destination-Realm = abc.com
10/02/2012 22:54:58.512: P74:    Origin-Host = 10.81.79.42
10/02/2012 22:54:58.512: P74:    Origin-Realm = abc1.com
10/02/2012 22:54:58.512: P74:    User-Name = james
10/02/2012 22:54:58.512: P74: Using Client: murdoch
10/02/2012 22:54:58.512: P74: Authenticating and Authorizing with Service dia-local
10/02/2012 22:54:58.512: P74: Calling Service local-users for authentication and
authorization
10/02/2012 22:54:58.512: P74: Getting User jame's UserRecord from UserList Default
10/02/2012 22:54:58.513: P74: Failed to get User jame's UserRecord from UserList
Default
10/02/2012 22:54:58.513: P74: Trace of Diameter-Access-Reject
10/02/2012 22:54:58.513: P74:    Auth-Application-Id = 1
10/02/2012 22:54:58.513: P74:    User-Name = james
10/02/2012 22:54:58.513: P74:    Auth-Request-Type = 3
10/02/2012 22:54:58.513: P74:    Result-Code = 4001
```

Configuring External Authentication Service

See Table 4-17 for more information on how to configure external authentication service.

Configuring Diameter Accounting

This section describes Diameter Accounting in Cisco Prime AR as defined in Internet RFC 6733. This section explains the following:

- Understanding Diameter Accounting
- Setting Up Local Accounting
- Diameter Accounting Log Examples

Understanding Diameter Accounting

Diameter Accounting is the process of collecting and storing the information contained in Accounting-Event, Accounting-Start, and Accounting-Interim and Accounting-Stop messages. Internet RFC 6733 describes the protocol for sending accounting information between a Network Access Server (NAS) and a DIAMETER server.

Note Cisco Prime AR uses TCP port number 3868 as its default port for Diameter accounting messages. Accounting/Authentication port number is configurable in Cisco Prime AR.

Setting Up Local Accounting

See Chapter 7, "Setting Up Accounting" for more information.

Setting up Oracle Accounting

See Chapter 7, "Oracle Accounting" for more information.

Diameter Accounting Log Examples

This section provides examples of Diameter accounting information recorded in an accounting log file.

Accounting Event Packet

```
Tue, 20 Oct 2012 15:27:18.340
    Session-Id = thomas.cisco.com.cisco1.com;1256052431;900083
    Origin-Host = thomas.cisco.com
    Origin-Realm = cisco1.com
    Destination-Realm = cisco.com
    Accounting-Record-Type = 1
    Accounting-Record-Number = 1
    Acct-Application-Id = 3
    Accounting-Sub-Session-Id = 1
    Acct-Interim-Interval = 5
    Accounting-Realtime-Required = 0
    Origin-State-Id = 1256052431
```

Accounting Start Packet

```
Tue, 20 Oct 2012 15:49:57.086
    Session-Id = thomas.cisco.com.cisco1.com;1256053789;847161
    Origin-Host = thomas.cisco.com
    Origin-Realm = cisco1.com
    Destination-Realm = cisco.com
    Accounting-Record-Type = 2
    Accounting-Record-Number = 1
    Acct-Application-Id = 3
    Accounting-Sub-Session-Id = 1
    Acct-Interim-Interval = 5
    Accounting-Realtime-Required = 0
```

```
    Origin-State-Id = 1256053789
```

Account Interim Packet

```
Tue, 20 Oct 2012 15:50:12.338
    Session-Id = thomas.cisco.com.cisco1.com;1256053789;847161
    Origin-Host = thomas.cisco.com
    Origin-Realm = cisco1.com
    Destination-Realm = cisco.com
    Accounting-Record-Type = 3
    Accounting-Record-Number = 4
    Acct-Application-Id = 3
    Accounting-Sub-Session-Id = 1
    Acct-Interim-Interval = 5
    Accounting-Realtime-Required = 1
    Origin-State-Id = 1256053789
```

Accounting Stop Packet

```
    Tue, 20 Oct 2012 15:50:18.116
    Session-Id = thomas.cisco.com.cisco1.com;1256053789;847161
    Origin-Host = thomas.cisco.com
    Origin-Realm = cisco1.com
    Destination-Realm = cisco.com
    Accounting-Record-Type = 4
    Accounting-Record-Number = 6
    Acct-Application-Id = 3
    Accounting-Sub-Session-Id = 1
    Acct-Interim-Interval = 5
    Accounting-Realtime-Required = 1
    Origin-State-Id = 1256053789
```

Trace of Successful Accounting

The following is a trace example of a a successful accounting sequence:

```
    10/02/2012 12:05:03.146: P161: Trace of Diameter-Accounting-Request packet
    10/02/2012 12:05:03.146: P161:    Session-Id =
    10.81.79.42.cisco5.com;1317577008;898336
    10/02/2012 12:05:03.146: P161:    Accounting-Record-Number = 1
    10/02/2012 12:05:03.146: P161:    Accounting-Record-Type = 2
    10/02/2012 12:05:03.146: P161:    Destination-Realm = abc.com
    10/02/2012 12:05:03.146: P161:    Origin-Host = 10.81.79.42
    10/02/2012 12:05:03.146: P161:    Origin-Realm = cisco5.com
    10/02/2012 12:05:03.146: P161:    Accounting-Realtime-Required = 0
    10/02/2012 12:05:03.146: P161:    Accounting-Sub-Session-Id = 1
    10/02/2012 12:05:03.146: P161:    Acct-Application-Id = 3
    10/02/2012 12:05:03.146: P161:    Acct-Interim-Interval = 5
    10/02/2012 12:05:03.146: P161:    Origin-State-Id = 1317577008
    10/02/2012 12:05:03.146: P161: Tracing the packet after running the rules and policies
    10/02/2012 12:05:03.146: P161: Trace of Diameter-Accounting-Request packet
    10/02/2012 12:05:03.146: P161:    Session-Id =
    10.81.79.42.cisco5.com;1317577008;898336
    10/02/2012 12:05:03.146: P161:    Accounting-Record-Number = 1
    10/02/2012 12:05:03.146: P161:    Accounting-Record-Type = 2
    10/02/2012 12:05:03.146: P161:    Destination-Realm = abc.com
    10/02/2012 12:05:03.147: P161:    Origin-Host = 10.81.79.42
    10/02/2012 12:05:03.147: P161:    Origin-Realm = cisco5.com
    10/02/2012 12:05:03.147: P161:    Accounting-Realtime-Required = 0
```

```
10/02/2012 12:05:03.147: P161:    Accounting-Sub-Session-Id = 1
10/02/2012 12:05:03.147: P161:    Acct-Application-Id = 3
10/02/2012 12:05:03.147: P161:    Acct-Interim-Interval = 5
10/02/2012 12:05:03.147: P161:    Origin-State-Id = 1317577008
10/02/2012 12:05:03.147: P161: Using Client: murdoch
10/02/2012 12:05:03.147: P161: Accounting with Service dia-local
10/02/2012 12:05:03.147: P161: Calling Service local-file for accounting
10/02/2012 12:05:03.123: P161: Trace of Diameter-Accounting-Response packet
10/02/2012 12:05:03.123: P161:    Session-Id =
10.81.79.42.cisco5.com;1317577008;898336
10/02/2012 12:05:03.123: P161:    Result-Code = 2001
10/02/2012 12:05:03.123: P161:    Origin-Host = 10.77.247.117
10/02/2012 12:05:03.123: P161:    Origin-Realm = abc.com
10/02/2012 12:05:03.123: P161:    Accounting-Record-Type = 2
10/02/2012 12:05:03.123: P161:    Accounting-Record-Number = 1
10/02/2012 12:05:03.123: P161:    Acct-Application-Id = 3
10/02/2012 12:05:03.123: P161:    Accounting-Sub-Session-Id = 1
10/02/2012 12:05:03.123: P161:    Error-Reporting-Host = 10.77.247.117
10/02/2012 12:05:03.123: P161:    Accounting-Realtime-Required = 1
10/02/2012 12:05:03.123: P161:    Acct-Interim-Interval = 5
10/02/2012 12:05:03.123: P161:    Origin-State-Id = 1317576779
```

Configuring the Diameter Application in Cisco Prime AR

For proxying a diameter application message in Cisco Prime AR, ensure that you do the following:

- Importing Application Specific Cisco AVPs to Cisco Prime AR Internal Database
- Configuring the Transport Management Properties
- Registering Applications IDs
- Configuring the Diameter Peers
- Configure the Diameter Service

Importing Application Specific Cisco AVPs to Cisco Prime AR Internal Database

You need to import the diameter application specific command codes and AVPs to the Cisco Prime AR internal database. The following is an example for importing Gy application command codes and AVPs. Ensure that you execute the following commands in the specified order to import the Diameter AVPs for BASE, NASREQ, and Gy applications.

```
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/add-BaseProtocolAVPs.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/add-BaseApplication.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/add-NASREQAVPs.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/add-NASREQApplication.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/add-BaseAccountingApplication.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gx-Gxx-Gy-Rx-S9.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gx-Gy-Gxx-S9-S6-Rx.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Cx-Gx-S9-Gy.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Cx-Wx-Sh-Gy.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gx-Gxx-S9-Gy.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gx-Rx-Gy-S9.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gxx-S9-Gy.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gy-Cx-Sh.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gy-Gx-S9.rc
```

```
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gy-S6.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/Common-Gy-S6-Sh.rc
/cisco-ar/bin/aregcmd -s -f
/cisco-ar/examples/cli/add-CreditControl-Gy-ApplicationAVPs.rc
/cisco-ar/bin/aregcmd -s -f /cisco-ar/examples/cli/add-CreditControl-Gy-Application.rc

Server 'Radius' is Running, its health is 10 out of 10
```

Configuring the Transport Management Properties

You need to log into the aregcmd using the CLI interfaceand configure the Transport Management properties in the **Radius/Advanced/Diameter/**.

```
/opt/CSCOar/bin/./aregcmd -s
Cisco Prime Access Registrar Configuration Utility
Copyright (C) 1995-2012 by Cisco Systems, Inc.  All rights reserved. Logging in to
localhost
[ //localhost ]
LicenseInfo = AR-DRN-2000TPS 5.1(2000TPS:expires on 1-Feb-2012) Radius/
Administrators/
Server 'Radius' is Running, its health is 10 out of 10
--> cd Radius/Advanced/Diameter/
--> cd TransportManagement/
    [ //localhost/Radius/Advanced/Diameter/TransportManagement ]
    Identity =
    Realm =
    TCPListenPort = 3868
    SCTPListenPort = 3868
    EnableIPV6 = FALSE
    WatchdogTimeout = 500
    ReconnectInterval = 500
    MaxReconnections = 3
    RequestRetransmissionInterval = 100
    MaxRequestRetransmissionCount = 3
    ReceiveBufferSize = 2048
    AdvertisedHostName/
```

You need to set the Identity and AdvertisedHostName properties to IP Address or hostname of the machine in which Cisco Prime AR is installed.

```
--> set Identity 10.77.240.69
Set Identity 10.77.240.69

--> cd AdvertisedHostName
set 1 10.77.240.69
Set 1 10.77.240.69
Set the Realm in which Cisco Prime Access Registrar server is present.
--> set Realm cisco.com
Set Realm cisco.com

Save the configuration

--> save

Validating //localhost...
Saving //localhost...

ls
    Identity = 10.77.240.69
    Realm = cisco.com
```

```
TCPListenPort = 3868
SCTPListenPort = 3868
EnableIPV6 = FALSE
WatchdogTimeout = 500
ReconnectInterval = 500
MaxReconnections = 3
RequestRetransmissionInterval = 100
MaxRequestRetransmissionCount = 3
ReceiveBufferSize = 2048
AdvertisedHostName/
1. 10.77.240.69
```

The description for these properties is available at:

http://www.cisco.com/en/US/docs/net_mgmt/access_registrar/5.1/user/guide/objects.html#wp1145662

Note Cisco Prime AR can only listen to one port for diameter connections. In the above configuration, the port number is 3868. All of the diameter clients must use this port number to communicate with the Cisco Prime AR.

Registering Applications IDs

You need to register the applications IDs for which Cisco Prime AR needs to route the Diameter Messages.

Registering the Gy application to a diameter stack

To register the Gy application to a diameter stack,

Step 1 Move to the **//localhost/Radius/Advanced/Diameter/General** directory.

```
[ //localhost/Radius/Advanced/Diameter ]
IsDiameterEnabled = TRUE
General/
TransportManagement/
SessionManagement/
Applications/
Commands/
Diameter Dictionary/

--> cd General/

[ //localhost/Radius/Advanced/Diameter/General ]
Product = Cisco Prime Access Registrar
Version = 6.0
AuthApplicationIdList =
AcctApplicationIdList =
```

The description for these properties is available at

http://www.cisco.com/en/US/docs/net_mgmt/access_registrar/6.0/user/guide/objects.html#wp1145646

Step 2 Set the AuthApplicationIdList to list of colon separated values of Application Ids.

```
--> set AuthApplicationIdList "4"

Set AuthApplicationIdList 4
```

Configuring the Diameter Peers

You need to configure the Diameter Peers such as clients and servers in the **/Radius/Clients** directory. The following is an example for configuring the Diameter Peers such as GGSN and OCS:

```
ggsn/
        Name = ggsn
        Description =
        Protocol = diameter
        HostName = GGSN-Gy
        Vendor =
        IncomingScript~ =
        OutgoingScript~ =
        Port = 3868
        SCTP-Enabled = FALSE
        Server-Identity =
        Server-Realm =

    ocs/
        Name = ocs
        Description =
        Protocol = diameter
        HostName = 192.168.30.88
        Vendor =
        IncomingScript~ =
        OutgoingScript~ =
        Port = 50301
        SCTP-Enabled = FALSE
        Server-Identity =
        Server-Realm =

    ocs1/
        Name = ocs1
        Description =
        Protocol = diameter
        HostName = 192.168.30.86
        Vendor =
        IncomingScript~ =
        OutgoingScript~ =
        Port = 60301
        SCTP-Enabled = FALSE
        Server-Identity =
        Server-Realm =
```

The description for these properties is available at:

http://www.cisco.com/en/US/docs/net_mgmt/access_registrar/6.0/user/guide/objects.html#wp1145646

Note In order to resolve the hostnames and get the IP addresses, the Cisco Prime AR should either be configured with a DNS server IP, or the client's hostnames and IP addresses should be included in the /etc/hosts file.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 Cisco Prime AR localhost.localdomain localhost
172.16.29.7 GGSN-Gy
::1 localhost6.localdomain6 localhost6
```

Configure the Diameter Service

To configure the Diameter Service to route the Diameter Messages based on the Realm,

Step 1 Add a Service of type diameter in /Radius/Services/.

```
--> cd  /Radius/Services/
--> add  dia-proxy

Added dia-proxy


--> cd dia-proxy

[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type =

--> set Type diameter

Set Type diameter
```

Step 2 Set role as Proxy and realm to which Cisco Prime AR needs to route the diameter messages.

```
--> ls

[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type = diameter
    Realm = abc.com
    Role = Local
    IncomingScript~ =
    OutgoingScript~ =
    AuthenticationService =
    AccountingService =
    Peers/

Set the Role as proxy
--> set Role Proxy

Set Role Proxy

--> set Realm mcprealm.com

Set Realm mcprealm.com
```

Step 3 Add a Gy application.

```
--> ls

[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
    Description =
    Type = diameter
    Realm = mcprealm.com
    Role = Proxy
    IncomingScript~ =
    OutgoingScript~ =
    Applications/

--> cd Applications/
```

```
[ //localhost/Radius/Services/dia-proxy/Applications ]
    Entries 0 to 0 from 0 total entries
    Current filter: <all>

--> add Gy

Added Gy

--> cd Gy/

[ //localhost/Radius/Services/dia-proxy/Applications/Gy ]
    Name = Gy
    Description =
    ApplicationID =
    DeMultiplexCCTerminateRequest = FALSE
    EnableSticky = FALSE
    MultiplePeersPolicy = Failover
    Peers/
```

Step 4 Set the application ID as 4 for Gy application and configure the sticky properties.

```
--> set ApplicationID 4

Set ApplicationID 4

--> set EnableSticky TRUE

Set EnableSticky TRUE

--> ls

[ //localhost/Radius/Services/dia-proxy/Applications/Gy ]
    Name = Gy
    Description =
    ApplicationID = 4
    DeMultiplexCCTerminateRequest = FALSE
    EnableSticky = TRUE
    StickySessionKey =
    StickyCreationCmdList =
    StickyDeletionCmdList =
    MultiplePeersPolicy = Failover
    Peers/

--> set StickySessionKey Session-Id#1

Set StickySessionKey Session-Id#1

--> set StickyCreationCmdList 272

Set StickyCreationCmdList 272

--> set StickyDeletionCmdList 272::CC-Request-Type=3

Set StickyDeletionCmdList 272::CC-Request-Type=3

--> set MultiplePeersPolicy RoundRobin

Set MultiplePeersPolicy RoundRobin

--> ls

[ //localhost/Radius/Services/dia-proxy/Applications/Gy ]
    Name = Gy
```

```
    Description =
    ApplicationID = 4
    DeMultiplexCCTerminateRequest = FALSE
    EnableSticky = TRUE
    StickySessionKey = Session-Id#1
    StickyCreationCmdList = 272
    StickyDeletionCmdList = 272::CC-Request-Type=3
    MultiplePeersPolicy = RoundRobin
    Peers/
```

Step 5 Add the OCS peers to which Cisco Prime AR needs to load balance the diameter Gy messages matching the Destination-Realm mcprealm.com.

```
--> cd Peers/

[ //localhost/Radius/Services/dia-proxy/Applications/Gy/Peers ]
    Entries 0 to 0 from 0 total entries
    Current filter: <all>


--> add ocs1

Added ocs1

--> cd ocs1/

[ //localhost/Radius/Services/dia-proxy/Applications/Gy/Peers/ocs1 ]
    Name = ocs1
    HostName =
    Metric = 2
    Weight = 0
    IMSIRanges =

--> set HostName 192.168.30.88

Set HostName 192.168.30.88

--> cd ..

[ //localhost/Radius/Services/dia-proxy/Applications/Gy/Peers ]
    Entries 1 to 1 from 1 total entries
    Current filter: <all>

    ocs1/

--> add ocs2

Added ocs2

--> cd ocs2/

[ //localhost/Radius/Services/dia-proxy/Applications/Gy/Peers/ocs2 ]
    Name = ocs2
    HostName =
    Metric = 2
    Weight = 0
    IMSIRanges =

--> set HostName 192.168.30.86

Set HostName 192.168.30.86
```

Step 6 Save the configuration details.

```
--> save
Validating //localhost...
Saving //localhost...
```

Step 7 Set DefaultAuthenticationService and DefaultAuthorizationService in /Radius directory.

```
--> set DefaultAuthenticationService dia-proxy

     Set DefaultAuthenticationService dia-proxy

--> set DefaultAuthorizationService dia-proxy

     Set DefaultAuthorizationService dia-proxy

--> save
Validating //localhost...
Saving //localhost...

--> exit

--> exit
Logging out of localhost...
```

Step 8 Restart theCisco Prime AR server.

```
/cisco-ar/bin/arserver restart
```

The following illustrates the diameter proxy service configuration for Gy application which load balances the diameter Gy (App ID =4) messages to the remote peers ocs1(192.168.30.88) and ocs2(192.168.30.86).

```
[ //localhost/Radius/Services/dia-proxy ]
    Name = dia-proxy
     Description =
    Type = diameter
       Realm = mcprealm.com
       Role = proxy
      IncomingScript~ =
    OutgoingScript~ =
    Applications/
        Entries 1 to 1 from 1 total entries
        Current filter: <all>

        Gy/
            Name = Gy
            Description =
            ApplicationID = 4
            DeMultiplexCCTerminateRequest = FALSE
            EnableSticky = TRUE
             StickySessionKey = Session-Id#1
             StickyCreationCmdList = 272
             StickyDeletionCmdList = 272::CC-Request-Type=3
             MultiplePeersPolicy = RoundRobin
             Peers/
                Entries 1 to 2 from 2 total entries
                Current filter: <all>

                ocs1/
                    Name = ocs
                    HostName = 192.168.30.88
                    Metric = 2
                    Weight = 0
                    IMSIRanges =
```

```
ocs2/
    Name = ocs2
    HostName = 192.168.30.86
    Metric = 3
    Weight = 0
    IMSIRanges =
```

The description for these properties is available at :

http://www.cisco.com/en/US/docs/net_mgmt/access_registrar/6.0/user/guide/objects.html#wp1145646

Writing Diameter Application in Cisco Prime AR

Cisco Prime AR supports extensibility by allowing users to create new:

- authentication/authorization applications
- accounting applications
- command codes
- AVP's

This section contains the following topics:

- Configuring rex script/service for Diameter
- Scripting in Diameter
- Diameter Environment Variables
- Sample rex script/service
- Traces/Logs

Configuring rex script/service for Diameter

To configure script/service for diameter using aregcmd:

Step 1 Add application specific AVPs in **//localhost/Radius/Advanced/Diameter/Diameter Dictionary** other than Base stack AVPs.

```
[ //localhost/Radius/Advanced/Diameter/Diameter Dictionary/CiscoAVPS ]
    Name = CiscoAVPS
    Description =
    IsVendorSpecific = FALSE
    ApplicationID =
    AVPs/
```

Step 2 Add a new command in **//localhost/Radius/Advanced/Diameter/Commands/** and specify the Request and Answers messages rules.

```
[ //localhost/Radius/Advanced/Diameter/Commands/Ciscocmd ]
    Name = Ciscocmd
    Description =
    CommandCode = 402
    EnableProxyBit = FALSE
    RequestMsgAVPs/
    AnswerMsgAVPs/
```

Step 3 Add a new application in **//localhost/Radius/Advanced/Diameter/Applications/** and specify the commands used by the application.

```
[ //localhost/Radius/Advanced/Diameter/Applications/Ciscoapp ]
    Name = Ciscoapp
    Description =
    IsAuthApplication = TRUE
    IsVendorSpecific = FALSE
    ApplicationID = 12
    ApplicationURI =
    Commands/
```

Step 4 Write a rex script (C/C++) and add it in the scripting point or rex service.

```
[ //localhost/Radius/Services/diaservice ]
    Name = diaservice
    Description =
    Type = rex
    IncomingScript~ =
    OutgoingScript~ =
    OutagePolicy~ = RejectAll
    OutageScript~ =
    Filename = librexscript.so
    EntryPoint = DiaService
    InitEntryPoint =
    InitEntryPointArgs =
```

Refer Sample rex script/service, page 8-19.

Scripting in Diameter

Cisco Prime AR supports 'rex' scripts for Diameter protocol. The script can be configured only as the server incoming script. The commands available for scripting are restricted to 'get' and 'put' on the dictionaries. While setting a value to an attribute, the following convention needs to be followed "<type number>,<value>". For example, if a 'Class' attribute needs to be added to the response dictionary with value as "classvalue", then set it as follows in the script:
pResponse->put(pResponse, "Class", "1,classvalue", REX_REPLACE);

The following is the list of supported scripting types with the respective type numbers:

```
AVP_STRING_TYPE = 1
AVP_ADDRESS_TYPE = 2
AVP_INTEGER32_TYPE = 3
AVP_UINTEGER32_TYPE = 4
AVP_UTF8_STRING_TYPE = 6
AVP_ENUM_TYPE = 7
AVP_TIME_TYPE = 11
```

Setting response attributes via a script is the only mechanism to add authorization attributes for Diameter requests.

Diameter Environment Variables

This section lists the environment variables that you can use in scripts for Diameter messages.

Table 8-1 lists the Diameter Environment variables and descriptions.

Table 8-1 Diameter Environment Variables

Variable	Description
Request-Type Response-Type	String value. Get/Set the request and response type for diameter packet. **Sample Values** Diameter-Access-Request Diameter-Access-Accept Diameter-Access-Reject Diameter-Accounting-Request Diameter-Accounting-Response Diameter-Proxy-Request Diameter-Proxy-Answer
Diameter-Application-Id	String value. Get the application id for the packet. For setting in response, need to use Auth-Application-id or Acct-Application-id AVPs. In Accounting type packet, use Acct-Application-Id AVP to get the application id. **Sample Values** 1 (NASREQ) 3 (Base Accounting)
Diameter-Command-Code	String value. Get command code for the diameter packet. It will work only for the access-request packet, not for the accounting request. **Sample Values** 265 (AA-Request)

Sample rex script/service

```
int REXAPI DiaService( int iScriptingPoint,
                       rex_AttributeDictionary_t* pRequest,
                       rex_AttributeDictionary_t* pResponse,
                       rex_EnvironmentDictionary_t* pEnviron )
{
    if( iScriptingPoint == REX_START_SERVICE || iScriptingPoint == REX_STOP_SERVICE )
        return REX_OK;
    int iRetVal = REX_ERROR;
    const char* pszRequestType = pEnviron->get( pEnviron, "Request-Type");
    const char* pszAppId = pEnviron->get( pEnviron, "Diameter-Application-Id" );
    const char* pszCmdCode=  pEnviron->get( pEnviron, "Diameter-Command-Code" );
    if(!( pszRequestType && pszAppId && pszCmdCode ))
        return iRetVal;
// check the request type, Application id and command code
/*
Request / Response types
```

```
Diameter-Access-Request
Diameter-Access-Accept
Diameter-Access-Reject
Diameter-Accounting-Request
Diameter-Accounting-Response
*/
    if( (strcmp( pszRequestType, "Diameter-Access-Request") == 0) && (strcmp(
pszAppId,"1") ==0 ) && (strcmp( pszCmdCode,"265\
" )== 0 ) )
    {
// our application
// example how to get DiaAttrib from the packet.
        const char* pszSessionId =  pRequest ->get( pRequest,"Session-Id",0,0 );
// print in trace
        if( pszSessionId )
            pEnviron->trace( pEnviron, 5, "Diameter Session Id: %s", pszSessionId );
// example: how to add dia attrib in response packet
        pResponse->put( pResponse, "Calling-Station-Id", "1,00-01-02-03-05", REX_APPEND );
        pEnviron->put( pEnviron, "Response-Type", "Diameter-Access-Accept");
        iRetVal = REX_OK;
    }
    return iRetVal;
}
```

Traces/Logs

```
09/30/2012 11:13:46.830: P88: Diameter Packet received from 10.81.79.59
09/30/2012 11:13:46.830: P88: Trace of Diameter-Access-Request packet
09/30/2012 11:13:46.830: P88:    Session-Id = .;1096298391;15
09/30/2012 11:13:46.830: P88:    Auth-Application-Id = 1
09/30/2012 11:13:46.830: P88:    Origin-Host = 10.81.79.59
09/30/2012 11:13:46.830: P88:    Origin-Realm = xyz.com
09/30/2012 11:13:46.830: P88:    Destination-Realm = abc.com
09/30/2012 11:13:46.830: P88:    Auth-Request-Type = 1
09/30/2012 11:13:46.830: P88:    User-Name = bob
09/30/2012 11:13:46.830: P88: Tracing the packet after running the rules and policies
09/30/2012 11:13:46.830: P88: Trace of Diameter-Access-Request packet
09/30/2012 11:13:46.830: P88:    Session-Id = .;1096298391;15
09/30/2012 11:13:46.830: P88:    Auth-Application-Id = 1
09/30/2012 11:13:46.830: P88:    Origin-Host = 10.81.79.59
09/30/2012 11:13:46.830: P88:    Origin-Realm = xyz.com
09/30/2012 11:13:46.830: P88:    Destination-Realm = abc.com
09/30/2012 11:13:46.830: P88:    Auth-Request-Type = 1
09/30/2012 11:13:46.830: P88:    User-Name = bob
09/30/2012 11:13:46.830: P88: Using Client: molly
09/30/2012 11:13:46.830: P88: Authenticating and Authorizing with Service
dia-rex-service
09/30/2012 11:13:46.830: P88:     Rex: environ->get( "Request-Type" ) ->
"Diameter-Access-Request"
09/30/2012 11:13:46.830: P88:     Rex: environ->get( "Diameter-Application-Id" ) ->
"1"
09/30/2012 11:13:46.830: P88:     Rex: environ->get( "Diameter-Command-Code" ) ->
"265"
09/30/2012 11:13:46.830: P88:     Rex: request->get( "Session-Id", 0 ) ->
".;1096298391;15"
09/30/2012 11:13:46.830: P88: Diameter Session Id: .;1096298391;15
09/30/2012 11:13:46.830: P88:     Rex: response->put( "Calling-Station-Id",
"1,00-01-02-03-05", 0 ) -> TRUE
09/30/2012 11:13:46.831: P88:     Rex: environ->put( "Response-Type",
"Diameter-Access-Accept" ) -> TRUE
09/30/2012 11:13:46.831: P88: Trace of Diameter-Access-Accept
09/30/2012 11:13:46.831: P88:    Calling-Station-Id = 00-01-02-03-05
```

```
09/30/2012 11:13:46.831: P88:    Auth-Application-Id = 1
09/30/2012 11:13:46.831: P88:    User-Name = bob
09/30/2012 11:13:46.831: P88:    Auth-Request-Type = 3
09/30/2012 11:13:46.831: P88:    Result-Code = 2001
```

Diameter Routing Agent

Service providers transform their 3G and 4G wireless networks with complex services, tiered charging, converged billing, and more by introducing increasing numbers and types of Diameter-based network elements. LTE and IMS networks are the most likely to implement these new network elements—including Policy and Charging Rules Functions (PCRF), Home Subscriber Servers (HSS), Mobility Management Entities (MME), Online Charging Systems (OCS), and others. As a result, as the traffic levels grow, these wireless networks are becoming more difficult to manage and scale without the Cisco Prime AR infrastructure.

The following sections describes the types of diameter agent and how to import the diameter command codes.

- Diameter Relay Agent
- Diameter Proxy Agent
- Importing Diameter Command Codes

Diameter Relay Agent

Relay agent is used to forward a request to the appropriate peer based on the information included in the request. As the relay agent collects the requests from different realms to a specific realm, the configurations of network access servers for every Diameter server change is not required.

The following is an example for Diameter Relay Agent configuration:

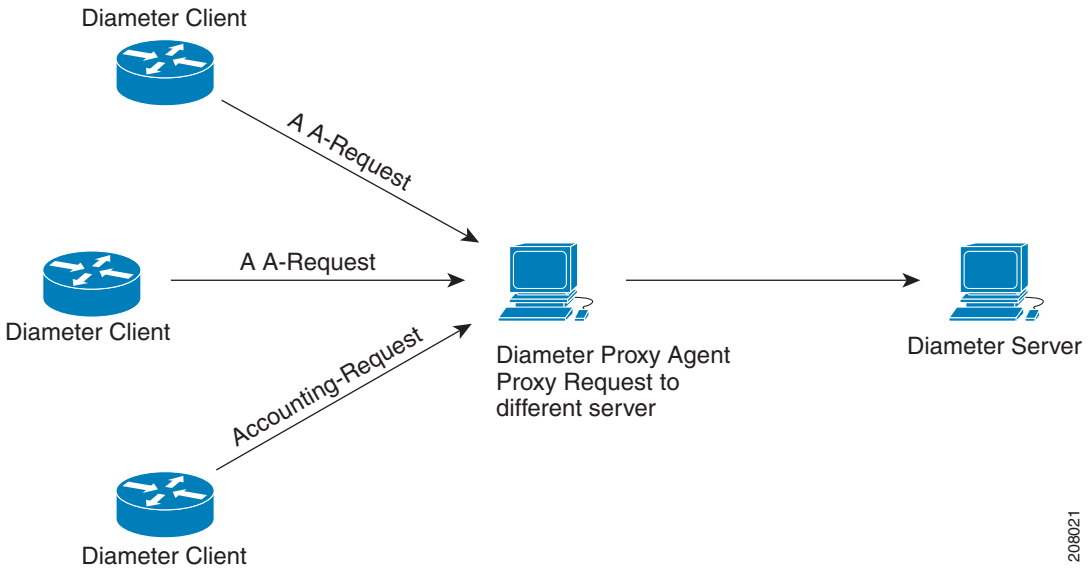
```
[ //localhost/Radius/Services/dia-relay ]
[ dia-relay ]
 Name = dia-relay
 Description =
 Type = diameter
 Realm = cisco.com
 Role = relay
 Peers/
  Entries 1 to 1 from 1 total entries
  Current filter: <all>

 53/
  Name = 53
  HostName = 10.77.240.53
  IsVendorSpecific = tRUE
  VendorSpecificApplicationID = 16777219
  VendorID = 10415
  Metric = 1
```

Diameter Proxy Agent

Proxy agents assist in routing Diameter messages using the Diameter routing table. The messages can be modified to implement policy enforcement. A proxy agent can also be used in forwarding messages, but unlike a relay agent, a proxy agent will modify the message content to provide value added services, enforce rules on different messages or to perform tasks for a specific realm. Figure 8-1 explains the message forwarding process to another domain by a proxy agent.

Figure 8-1 Diameter Proxy Agent



Diameter proxy service works in tandem with the rule policy engine to perform the routing for multiple realms or applications. The following are the multiple peer policies supported by the proxy service:

- RoundRobin
- FailOver
- IMSI Range Based

RoundRobin

In proxy mode, Cisco Prime AR allows distribution of incoming traffic to remote peers using equal weight-based load balancing or differential weight-based load balancing.

In the equal weight-based load balancing, all peers are assigned an equal weight. Cisco Prime AR uniformly shares the incoming load across all of the servers in the Peers list.

In differential weight-based load balancing, a unique weight is assigned to each peer in the service. Cisco Prime AR sends the incoming load to the peers in proportion to the weight configured in the peer list. By default, the weight of remote peer is set to 0. The weights need to be configured as multiples of 10 and the sum of the weights configured in the peer list should be equal to 100. Currently, in weight-based load balancing, Cisco Prime AR supports a maximum of ten peers.

For example, if you have two remote peers with the weights configured in the ratio of 50:50, both the remote peers will receive an equal number of requests. If you have two remote peers with weights configured in the ratio of 70:30, Cisco Prime AR will send 70% of the incoming traffic to one peer and the remaining 30% to another peer for the service. For configuration details, see Configuring Diameter Proxy, page 8-23.

FailOver

When Failover mode is selected, Cisco Prime AR directs requests to the first peer in the list which has the least metric value. The requests are sent to this peer until the peer is online. If the first peer goes down, Cisco Prime AR redirects all requests to the next peer in the list until with lesser metric value coming back online. When the first peer goes down, Cisco Prime AR redirects all requests to the next online peer that has the second least metric value. If the first peer comes back online, the requests are sent again to the first peer. For configuration details, see Configuring Diameter Proxy, page 8-23.

IMSI Range Based

When the International Mobile Subscriber Identity (IMSI) range mode is selected, Cisco Prime AR determines which peers have to take the incoming requests based on the IMSI range configured in the proxy service. The sticky session key must be configured to an AVP that contains the IMSI. In the proxy service, mappings are created between the peers and the IMSI ranges based on which the received packets are directed through the mapped peer.

For example, consider the peers, peer1, peer2, and peer3 with their IMSI range 100000000000000-200000000000000, 200000000000001-300000000000000, and 300000000000001-400000000000000 respectively. When a request with 250000000000000 as IMSI range is received, the request is automatically forwarded to peer2.

Configuring Diameter Proxy

Cisco Prime AR server acts as a proxy agent when you set the role as proxy for a particular realm. In the peer list, you have to configure which application messages need to be proxied and to whom. In the example below, the Base, NASREQ messages and Accounting messages are proxied to gordan-ar1.cisco.com system.

Cisco Prime AR provides two scripting points for modifying the proxy packet. IncomingScript point will run for proxy-request message, OutgoingScript point will run for proxy-response messages specific to the given realm.

See Table 4-21 for more information on Diameter Service properties.

```
[ //localhost/Radius/Services/diameter-proxy ]
Name = diameter-proxy
Description =
Type = diameter
Realm = cisco.com
Role = Proxy
IncomingScript~ =
OutgoingScript~ =
Applications/
Entries 1 to 1 from 1 total entries
Current filter: <all>

NASREQ/
Name = NASREQ
```

```
Description =
IsVendorSpecific = FALSE
ApplicationID = 1
StickyAVP = Session-Id
MultiAVPPosition = 1
StickyCreationCmdList = 265||271::Accounting-Record-Type=2
StickyDeletionCmdList = 275||271::Accounting-Record-Type=4
StickyTimeout = 10000
UseIMSIRangeBasedLoadBalancing = FALSE
MultiplePeersPolicy = RoundRobin
Peers/
Entries 1 to 1 from 1 total entries
Current filter: <all>

hss1/
Name = hss1
HostName = gordon-ar1
Metric = 2
Weight = 0
IMSIRange =
```

The following configuration is an example of the differential weight-based load balancing for the peers with weights configured in the ratio of 70:30. For every 10 requests the Cisco Prime AR has received for S6 Application(16777251), it distributes 7 to hss1 and 3 to hss2.

```
[ //localhost/Radius/Services/diameter-proxy ]
Name = diameter-proxy
Description =
Type = diameter
Realm = cisco.com
Role = Proxy
IncomingScript~ =
OutgoingScript~ =
Applications/
Entries 1 to 1 from 1 total entries
Current filter: <all>

S6Application/
Name = S6Application
Description =
IsVendorSpecific = FALSE
ApplicationID = 16777251
EnableSticky = FALSE
MultiplePeersPolicy = RoundRobin
Peers/
Entries 1 to 1 from 1 total entries
Current filter: <all>

hss1/
Name = hss1
HostName = gordon-ar1
Metric = 2
Weight = 70
hss2/
Name = hss2
HostName =  henry-ar1
Metric = 1
Weight = 30
```

The following is an example for FailOver configuration:

```
[ //localhost/Radius/Services/dia-failover ]
Name = dia-failover
Description =
Type = diameter
Realm = cisco.com
Role = Proxy
IncomingScript~ =
OutgoingScript~ =
Applications/
Entries 1 to 1 from 1 total entries
Current filter: <all>

NASREQ/
 Name = NASREQ
 Description =
 ApplicationID = 1
 EnableSticky = FALSE
 MultiplePeersPolicy = Failover
 Peers/
Entries 1 to 3 from 3 total entries
Current filter: <all>

ocs1/
 Name = ocs1
 HostName = 10.77.240.69
 Metric = 1
 Weight = 0
 IMSIRanges =
ocs2/
 Name = ocs2
 HostName = 10.77.240.70
 Metric = 2
 Weight = 0
 IMSIRanges =
ocs3/
 Name = ocs3
 HostName = 10.77.240.80
 Metric = 3
 Weight = 0
 IMSIRanges =
```

The following is an example for IMSI Range Based configuration:

```
[ //localhost/Radius/Services/dia-imsi ]
Name = dia-imsi
Description =
Type = diameter
Realm = epc.com
Role = Proxy
IncomingScript~ =
OutgoingScript~ =
Applications/
Entries 1 to 1 from 1 total entries
Current filter: <all>

NASREQ/
 Name = NASREQ
 Description =
 ApplicationID = 1
 EnableSticky = TRUE
 StickySessionKey = Subscription-Id<Subscription-Id-Data>#1
 StickyCreationCmdList = 275::Accounting-Record-Type=2
```

```
 StickyDeletionCmdList = 275::Accounting-Record-Type=4
 MultiplePeersPolicy = IMSIRangeBased
Peers/
Entries 1 to 2 from 2 total entries
Current filter: <all>

hss1/
 Name = hss1
 HostName = 10.77.240.69
 Metric = 2
 Weight = 0
 IMSIRanges = 1000-2000
hss2/
 Name = hss2
 HostName = 10.77.240.70
 Metric = 1
 Weight = 0
 IMSIRanges = 4000-6000
```

Note The AVPs names entered in the StickySessionKey are case-sensitive. These AVPs should be available in the Cisco Prime AR Diameter Dictionary.

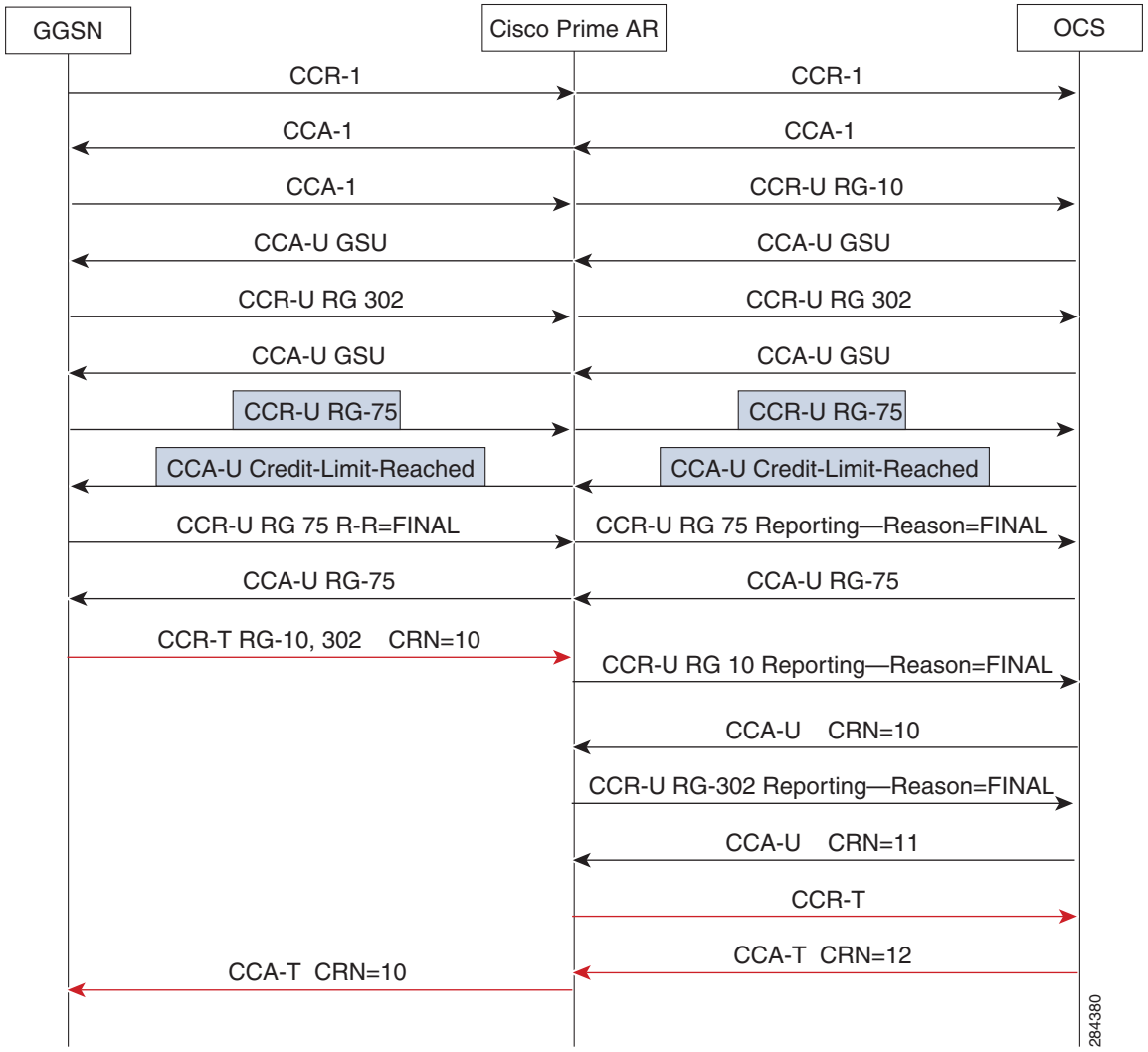
Configuring Cisco Prime AR to Demultiplex the Diameter CCR-T

Cisco Prime AR server generates and sends multiple Credit Control Update (CCR-U) requests corresponding to an incoming diameter Credit Control Termination (CCR-T) request, while proxying Gy messages between the Gateway GPRS Support Node (GGSN) and Online charging system (OCS).

Cisco Prime AR server generates a new hop-to-hop and end-to-end diameter identifier for every CCR-Us generated. The CC-Request-Number is incremented sequentially, from what the GGSN sends in the Credit Control Terminate Request (CCR-T), for each of the CCR-Us generated.

Cisco Prime AR internally maintains a list of Rating Group (RG) values for which OCS sends Credit Control Update Answer (CCA-U) with Result-Code AVP value as Credit-Limit-Reached. While de-multiplexing the CCR-T request into several CCR-Us, the RGs that expired are skipped. Also, Cisco Prime AR maintains the state of each CCR-U/CCA-U transaction with OCS and adds an appropriate result-code in the Multiple Service Credit Control (MSCC) AVP while sending the CCA-T response to the GGSN. Cisco Prime AR server waits until all the CCR-U transactions are completed (response received or time-out) before forwarding the CCR-T to GGSN. However, it will initiate the CCR-Us (for each RG) in parallel to the OCS. During the time-out interval, if there is no response from OCS, Cisco Prime AR sends a response message to GGSN indicating failure in the delivery. Figure 8-2 explains the message flow process from GGSN to OCS through Cisco Prime AR.

Figure 8-2 Message Flow between GGSN and OCS



The following is an example configuration for demultiplexing the Diameter Credit Control Terminate Request:

```
[ //localhost/Radius/Services/Gy-dia-service ]
    Name = Gy-dia-service
    Description =
    Type = diameter
    Realm = ggsn.com
    Role = Proxy
    IncomingScript~ =
    OutgoingScript~ =
    Applications/
        Entries 1 to 1 from 1 total entries
        Current filter: <all>
        Gy/
            Name = Gy
            Description =
            IsVendorSpecific = FALSE
            ApplicationID = 4
```

```
EnableSticky = FALSE
DeMultiplexCCTerminateRequest = TRUE
MultiplePeersPolicy = Failover
Peers/
    Entries 1 to 1 from 1 total entries
    Current filter: <all>
    OCS/
        Name = OCS
        HostName = ocs.it.com
        Metric = 2
        Weight = 0
```

Traces/Logs

Round Robin Load Balancing Traces:

```
06/03/2012  8:54:54.193: P199: Diameter Packet received from 10.81.79.42
06/03/2012  8:54:54.193: P199: Trace of Diameter-Proxy-Request packet
06/03/2012  8:54:54.193: P199:    Command code = 265
06/03/2012  8:54:54.193: P199:    Session-Id = .;1096298391;1
06/03/2012  8:54:54.193: P199:    Auth-Application-Id = 1
06/03/2012  8:54:54.193: P199:    Origin-Host = 10.81.79.42
06/03/2012  8:54:54.193: P199:    Origin-Realm = abc1.com
06/03/2012  8:54:54.193: P199:    Destination-Realm = cisco.com
06/03/2012  8:54:54.193: P199:    Auth-Request-Type = 3
06/03/2012  8:54:54.193: P199:    User-Name = bob
06/03/2012  8:54:54.193: P199:    User-Name = jane
06/03/2012  8:54:54.193: P199:    <Vendor-Specific-Application-Id>
06/03/2012  8:54:54.193: P199:    Command code = 265
06/03/2012  8:54:54.193: P199:    Vendor-Id = 3000
06/03/2012  8:54:54.193: P199:    Acct-Application-Id = 3
06/03/2012  8:54:54.193: P199:    </Vendor-Specific-Application-Id>
06/03/2012  8:54:54.193: P199:    <Vendor-Specific-Application-Id>
06/03/2012  8:54:54.193: P199:    Command code = 265
06/03/2012  8:54:54.193: P199:    Vendor-Id = 195
06/03/2012  8:54:54.193: P199:    Acct-Application-Id = 3
06/03/2012  8:54:54.193: P199:    </Vendor-Specific-Application-Id>
06/03/2012  8:54:54.193: P199:    <Subscription-Id>
06/03/2012  8:54:54.193: P199:    Command code = 265
06/03/2012  8:54:54.193: P199:    Subscription-Id-Data = 1959999
06/03/2012  8:54:54.193: P199:    Subscription-Id-Type = 0
06/03/2012  8:54:54.193: P199:    </Subscription-Id>
06/03/2012  8:54:54.194: P199:    <Subscription-Id>
06/03/2012  8:54:54.194: P199:    Command code = 265
06/03/2012  8:54:54.194: P199:    Subscription-Id-Data = 112456
06/03/2012  8:54:54.194: P199:    Subscription-Id-Type = 1
06/03/2012  8:54:54.194: P199:    </Subscription-Id>
06/03/2012  8:54:54.194: P199:    Route-Record = 10.77.240.72
06/03/2012  8:54:54.194: P199: Processing the diameter proxy packet
06/03/2012  8:54:54.194: P199: Using Client: murdoch
06/03/2012  8:54:54.194: P199: Authenticating and Authorizing with Service dia-proxy
06/03/2012  8:54:54.194: P199: Service dia-proxy: Enabled Sticky
06/03/2012  8:54:54.194: P199: Service dia-proxy: Using Round Robin Load Balancing
06/03/2012  8:54:54.194: P199: Service dia-proxy: Setting the sticky entry to bob
06/03/2012  8:54:54.194: P199: Service dia-proxy: Sending request to remote peer
9,10.77.240.69
06/03/2012  8:54:54.195: Log: Destination peer changed based on Destination-Host AVP
06/03/2012  8:54:54.198: P200: Diameter Packet received from 10.77.240.69
06/03/2012  8:54:54.198: P200: Trace of Diameter-Proxy-Answer packet
06/03/2012  8:54:54.198: P200:    Command code = 265
06/03/2012  8:54:54.198: P200:    Session-Id = .;1096298391;1
06/03/2012  8:54:54.198: P200:    Auth-Application-Id = 1
```

```
06/03/2012  8:54:54.198: P200:    Auth-Request-Type = 3
06/03/2012  8:54:54.198: P200:    Result-Code = 2001
06/03/2012  8:54:54.198: P200:    Origin-Host = 10.77.240.69
06/03/2012  8:54:54.198: P200:    Origin-Realm = cisco.com
06/03/2012  8:54:54.198: P200:    User-Name = bob
06/03/2012  8:54:54.198: P200:    Auth-Grace-Period = 30
06/03/2012  8:54:54.198: P200:    Auth-Session-State = 0
06/03/2012  8:54:54.198: P200:    Session-Timeout = 1000
06/03/2012  8:54:54.198: P200:    Authorization-Lifetime = 360
```

FailOver Traces:

```
06/03/2012 15:12:19.500: P200: Diameter Packet received from 10.81.79.42
06/03/2012 15:12:19.500: P200: Trace of Diameter-Proxy-Request packet
06/03/2012 15:12:19.500: P200:    Command code = 265
06/03/2012 15:12:19.500: P200:    Session-Id = .;1096298391;1
06/03/2012 15:12:19.501: P200:    Auth-Application-Id = 1
06/03/2012 15:12:19.501: P200:    Origin-Host = 10.81.79.42
06/03/2012 15:12:19.501: P200:    Origin-Realm = abc1.com
06/03/2012 15:12:19.501: P200:    Destination-Realm = cisco.com
06/03/2012 15:12:19.501: P200:    Auth-Request-Type = 3
06/03/2012 15:12:19.501: P200:    User-Name = bob
06/03/2012 15:12:19.501: P200:    User-Name = jane
06/03/2012 15:12:19.501: P200:    <Vendor-Specific-Application-Id>
06/03/2012 15:12:19.501: P200:    Command code = 265
06/03/2012 15:12:19.501: P200:    Vendor-Id = 3000
06/03/2012 15:12:19.501: P200:    Acct-Application-Id = 3
06/03/2012 15:12:19.501: P200:    </Vendor-Specific-Application-Id>
06/03/2012 15:12:19.501: P200:    <Vendor-Specific-Application-Id>
06/03/2012 15:12:19.501: P200:    Command code = 265
06/03/2012 15:12:19.501: P200:    Vendor-Id = 195
06/03/2012 15:12:19.502: P200:    Acct-Application-Id = 3
06/03/2012 15:12:19.502: P200:    </Vendor-Specific-Application-Id>
06/03/2012 15:12:19.502: P200:    <Subscription-Id>
06/03/2012 15:12:19.502: P200:    Command code = 265
06/03/2012 15:12:19.502: P200:    Subscription-Id-Data = 1959999
06/03/2012 15:12:19.502: P200:    Subscription-Id-Type = 0
06/03/2012 15:12:19.502: P200:    </Subscription-Id>
06/03/2012 15:12:19.502: P200:    <Subscription-Id>
06/03/2012 15:12:19.502: P200:    Command code = 265
06/03/2012 15:12:19.502: P200:    Subscription-Id-Data = 112456
06/03/2012 15:12:19.502: P200:    Subscription-Id-Type = 1
06/03/2012 15:12:19.502: P200:    </Subscription-Id>
06/03/2012 15:12:19.502: P200:    Route-Record = 10.77.240.72
06/03/2012 15:12:19.502: P200: Processing the diameter proxy packet
06/03/2012 15:12:19.503: P200: Using Client: murdoch
06/03/2012 15:12:19.503: P200: Authenticating and Authorizing with Service dia-proxy
06/03/2012 15:12:19.657: P201: Diameter Packet received from 10.81.79.51
06/03/2012 15:12:19.657: P201: Trace of Diameter-Proxy-Answer packet
06/03/2012 15:12:19.657: P201:    Command code = 265
06/03/2012 15:12:19.657: P201:    Session-Id = .;1096298391;1
06/03/2012 15:12:19.657: P201:    Auth-Application-Id = 1
06/03/2012 15:12:19.657: P201:    Auth-Request-Type = 3
06/03/2012 15:12:19.657: P201:    Result-Code = 2001
06/03/2012 15:12:19.657: P201:    Origin-Host = 10.81.79.51
06/03/2012 15:12:19.657: P201:    Origin-Realm = cisco.com
06/03/2012 15:12:19.658: P201:    User-Name = bob
06/03/2012 15:12:19.658: P201:    Auth-Grace-Period = 30
06/03/2012 15:12:19.658: P201:    Auth-Session-State = 1
06/03/2012 15:12:19.658: P201:    Session-Timeout = 30
06/03/2012 15:12:19.658: P201:    Authorization-Lifetime = 29
```

IMSI Range Based Load Balancing Traces:

```
09/30/2012 18:43:21.357: P159: Trace of Diameter-Proxy-Request packet
09/30/2012 18:43:21.357: P158:    Auth-Application-Id = 1
09/30/2012 18:43:21.357: P159:    Command code = 265
09/30/2012 18:43:21.357: P158:    Auth-Request-Type = 3
09/30/2012 18:43:21.357: P159:    Session-Id = .;1096298391;9
09/30/2012 18:43:21.357: P158:    Origin-Host = 10.77.240.69
09/30/2012 18:43:21.357: P159:    Auth-Application-Id = 1
09/30/2012 18:43:21.357: P158:    Origin-Realm = cisco.com
09/30/2012 18:43:21.357: P159:    Auth-Request-Type = 3
09/30/2012 18:43:21.357: P158:    Result-Code = 2001
09/30/2012 18:43:21.357: P159:    Destination-Realm = dia.com
09/30/2012 18:43:21.357: P158:    User-Name = bob
09/30/2012 18:43:21.357: P159:    Origin-Host = 10.77.240.54
09/30/2012 18:43:21.357: P159:    Origin-Realm = cisco.in
09/30/2012 18:43:21.357: P159:    User-Name = 112156000000001
09/30/2012 18:43:21.357: P159:    Route-Record = 10.77.247.117
09/30/2012 18:43:21.358: P159: Authenticating and Authorizing with Service dia-proxy
09/30/2012 18:43:21.358: P160:    Origin-Host = 10.77.240.69
09/30/2012 18:43:21.358: P160:    Origin-Realm = cisco.com
09/30/2012 18:43:21.358: P159: Service dia-proxy: Enabled Sticky
09/30/2012 18:43:21.358: P160:    Result-Code = 2001
09/30/2012 18:43:21.358: P159: Service dia-proxy: IMSI 112156000000001 found in configured
IMSI Ranges for peer 10.77.240.69
09/30/2012 18:43:21.358: P159: Service dia-proxy: Using IMSI Range Based Load Balancing
09/30/2012 18:43:21.358: P159: Service dia-proxy: Setting the sticky entry to
112156000000001
09/30/2012 18:43:21.358: P159: Service dia-proxy: Sending request to remote peer
10.77.240.69
09/30/2012 18:43:21.358: Log: Destination peer changed based on Destination-Host AVP
09/30/2012 18:43:21.359: P159: Trace of Diameter-Proxy-Answer packet
09/30/2012 18:43:21.359: P159:    Command code = 265
09/30/2012 18:43:21.359: P159:    Session-Id = .;1096298391;9
09/30/2012 18:43:21.359: P159:    Auth-Application-Id = 1
09/30/2012 18:43:21.359: P159:    Auth-Request-Type = 3
09/30/2012 18:43:21.359: P159:    Origin-Host = 10.77.240.69
09/30/2012 18:43:21.359: P159:    Origin-Realm = cisco.com
09/30/2012 18:43:21.359: P159:    Result-Code = 2001
09/30/2012 18:43:21.359: P159:    User-Name = bob
```

Writing Diameter Proxy Extension Scripts

During the Diameter proxy process, Cisco Prime AR uses the extension point scripting to modify the packets. Scripting is supported using C and C++ (rex).

See Configuring rex script/service for Diameter for more details.

Note Use the request dictionary for modifying (get, put, remove) the AVPs. The AVPs names are case sensitive.

Sample Diameter Proxy Extension Script

The following is an example of the sample diameter proxy extension script.

```
int REXAPI DiaProxyIN( int iScriptingPoint,
    rex_AttributeDictionary_t* pRequest,
    rex_AttributeDictionary_t* pResponse,
    rex_EnvironmentDictionary_t* pEnviron )
{
```

```
    do
         {
           const char* pszAppId = pEnviron->get( pEnviron, "Diameter-Application-Id"  \
);
           const char* pszCmdCode=  pEnviron->get( pEnviron, "Diameter-Command-Code"  \
);
           if(!( pszAppId && pszCmdCode ))
               break;
           if( (strcmp( pszAppId, "1") ==0 ) && (strcmp( pszCmdCode, "265" )== 0 ) )
           {
            // NASREQ Proxy Request
                 if( pRequest->containsKey( pRequest, "User-Name" ) )
                 {
                   const char* pUsername = pRequest->get( pRequest, "User-Name", 0,   \
0 );
                   pRequest->put( pRequest, "User-Name", "1,Milton", REX_REPLACE );
                   pRequest->put( pRequest, "Class", "1,00-01-02-03-05", REX_APPEND   \
);
                   pRequest->remove( pRequest, "Authorization-Lifetime", 0 );
                 }
                 else
                   pEnviron->trace(pEnviron, 5, "User-Name not found in Request       \
packet ");
           }
         }while(0);
         return REX_OK;
}
```

Traces/Logs

```
05/07/2012  0:26:26.750: P74: Diameter Packet received from spencer-ar1.cisco.com
05/07/2012  0:26:26.750: P74: Trace of Diameter-Proxy-Request packet
05/07/2012  0:26:26.750: P74:    Command code = 265
05/07/2012  0:26:26.751: P74:    Session-Id =
spencer-ar1.cisco.com.cisco1.com;1273217178;706980
05/07/2012  0:26:26.751: P74:    Auth-Application-Id = 1
05/07/2012  0:26:26.751: P74:    Origin-Host = spencer-ar1.cisco.com
05/07/2012  0:26:26.751: P74:    Origin-Realm = cisco1.com
05/07/2012  0:26:26.751: P74:    Destination-Realm = abc.com
05/07/2012  0:26:26.751: P74:    Auth-Request-Type = 3
05/07/2012  0:26:26.751: P74:    User-Name = bob
05/07/2012  0:26:26.751: P74:    Authorization-Lifetime = 49
05/07/2012  0:26:26.751: P74:    Route-Record = toby-ar1.cisco.com
05/07/2012  0:26:26.751: P74: Processing the diameter proxy packet
05/07/2012  0:26:26.751: P74: Running Diameter Proxy Script: diaproxyin
05/07/2012  0:26:26.751: P74:     Rex: environ->get( "Diameter-Application-Id" ) -> "1"
05/07/2012  0:26:26.751: P74:     Rex: environ->get( "Diameter-Command-Code" ) -> "265"
05/07/2012  0:26:26.751: P74:     Rex: request->containsKey( "User-Name" ) -> TRUE
05/07/2012  0:26:26.751: P74:     Rex: request->get( "User-Name", 0 ) -> "bob"
05/07/2012  0:26:26.751: P74:     Rex: request->put( "User-Name", "1,Milton", 0 ) -> TRUE
05/07/2012  0:26:26.751: P74:     Rex: request->put( "Class", "1,00-01-02-03-05", 0 ) ->
TRUE
05/07/2012  0:26:26.751: P74:     Rex: request->remove( "Authorization-Lifetime" ) -> TRUE
05/07/2012  0:26:26.751: P74: After the alteration...
05/07/2012  0:26:26.751: P74: Trace of Diameter-Proxy-Request packet
05/07/2012  0:26:26.751: P74:    Command code = 265
05/07/2012  0:26:26.751: P74:    Session-Id =
spencer-ar1.cisco.com.cisco1.com;1273217178;706980
05/07/2012  0:26:26.751: P74:    Auth-Application-Id = 1
05/07/2012  0:26:26.751: P74:    Origin-Host = spencer-ar1.cisco.com
05/07/2012  0:26:26.751: P74:    Origin-Realm = cisco1.com
```

```
05/07/2012  0:26:26.751: P74:    Destination-Realm = abc.com
05/07/2012  0:26:26.751: P74:    Auth-Request-Type = 3
05/07/2012  0:26:26.751: P74:    User-Name = Milton
05/07/2012  0:26:26.751: P74:    Route-Record = toby-ar1.cisco.com
05/07/2012  0:26:26.751: P74:    Class = 00-01-02-03-05
05/07/2012  0:26:26.760: P75: Diameter Packet received from donald-ar1.cisco.com
05/07/2012  0:26:26.760: P75: Trace of Diameter-Proxy-Answer packet
05/07/2012  0:26:26.760: P75:    Command code = 265
05/07/2012  0:26:26.760: P75:    Session-Id =
spencer-ar1.cisco.com.cisco1.com;1273217178;706980
05/07/2012  0:26:26.760: P75:    Auth-Application-Id = 1
05/07/2012  0:26:26.760: P75:    Auth-Request-Type = 3
05/07/2012  0:26:26.760: P75:    Result-Code = 2001
05/07/2012  0:26:26.760: P75:    Origin-Host = donald-ar1.cisco.com
05/07/2012  0:26:26.760: P75:    Origin-Realm = abc.com
05/07/2012  0:26:26.760: P75:    User-Name = aantonim
05/07/2012  0:26:26.760: P75:    Auth-Grace-Period = 30
05/07/2012  0:26:26.760: P75:    Auth-Session-State = 0
05/07/2012  0:26:26.760: P75:    Session-Timeout = 300
05/07/2012  0:26:26.761: P75: Processing the diameter proxy packet
05/07/2012  0:26:26.761: P75: Running Diameter Proxy Script: diaproxyout
05/07/2012  0:26:26.761: P75:     Rex: request->get( "User-Name", 0 ) -> "aantonim"
05/07/2012  0:26:26.761: P75: After the alteration...
05/07/2012  0:26:26.761: P75: Trace of Diameter-Proxy-Answer packet
05/07/2012  0:26:26.761: P75:    Command code = 265
05/07/2012  0:26:26.761: P75:    Session-Id =
spencer-ar1.cisco.com.cisco1.com;1273217178;706980
05/07/2012  0:26:26.761: P75:    Auth-Application-Id = 1
05/07/2012  0:26:26.761: P75:    Auth-Request-Type = 3
05/07/2012  0:26:26.761: P75:    Result-Code = 2001
05/07/2012  0:26:26.761: P75:    Origin-Host = donald-ar1.cisco.com
05/07/2012  0:26:26.761: P75:    Origin-Realm = abc.com
05/07/2012  0:26:26.761: P75:    User-Name = aantonim
05/07/2012  0:26:26.761: P75:    Auth-Grace-Period = 30
05/07/2012  0:26:26.761: P75:    Auth-Session-State = 0
```

Importing Diameter Command Codes

To import the command codes:

Step 1 Import the Application command code for AVP's using **/cisco-ar/bin/aregcmd –sf** command. The S6a, Gx, and Gy command codes are available in /cisco-ar/examples/cli directory.
For example, /cisco-ar/examples/cli/add-3Gpp-Gx-ApplicationAVPs.rc.

Step 2 Import the Application using **/cisco-ar/bin/aregcmd –sf** command.
For example, /cisco-ar/examples/cli/add-3Gpp-Gx-Application.rc.

Step 3 Restart the Cisco Prime AR server.

Support for SCTP including Multihoming

Cisco Prime AR release enhances the diameter support to the more reliable transport mechanism such as SCTP with multi-homing.

In a SCTP connection, each of the two endpoints during an SCTP association setup can specify multiple points of attachment. Having multiple interfaces allows the data to be automatically sent to alternate addresses when failures occur. Using this support, the Cisco Prime AR runs successfully even when a failure occurs in any of the multiple interfaces.

```
[ //localhost/Radius/advanced/diameter/transportManagement ]
   Identity = localhost
   Realm = abc.com
   TCPListenPort = 3868
   SCTPListenPort = 3868
   EnableIPV6 = FALSE
   WatchdogTimeout = 500
   ReconnectInterval = 500
   MaxReconnections = 3
   RequestRetransmissionInterval = 100
   MaxRequestRetransmissionCount = 3
   ReceiveBufferSize = 2048
   AdvertisedHostName/

--> cd AdvertisedHostName
--> add 1 10.77.240.135
--> add 2 10.77.240.136
--> add 3 10.77.240.137
--> ls
[ //localhost/Radius/advanced/diameter/transportManagement/AdvertisedHostName ]
1. 10.77.240.135
2. 10.77.240.136
3. 10.77.240.137
```

Note The number of AVPs should be set greater than or equal to the number of AdvertisedHostName in order to exchange the capabilites between peers.