Test Case Document

Diameter Base Protocol API/Stack

Version 1.0

Objective

Product Overview

Diameter is the AAA protocol selected by 3GPP to provide Authentication, Authorization and Accounting (AAA) services in the IMS. The Diameter Base Protocol [RFC 3588] contains the basic functionality required for AAA model and is mandated in all Diameter nodes. The Diameter applications are extensions of the basic functionality that are tailored for a particular usage of Diameter in a particular environment. Diameter runs over reliable transport protocols, TCP and SCTP. This diameter project will implement IP Multimedia Subsystem's Home Subscriber Server supporting Cx/DX [2] and Sh [3] interfaces as defined by 3GPP standards, Online Charging System (OCS) [4] and Offline Charging System (CDF&CGF). This document specifically describes the Diameter Base Protocol API. The Diameter Base Protocol implementation (API) will address all the requirements mandated by the IETF RFC 3588.

The following components in IMS will use Diameter Base Protocol API in their implementation.

- Home Subscriber Server (HSS)
- Subscriber Locator Function (SLF)
- Charging Data Function (CDF) &
- Online Charging System (OCF).

Table of Contents

Abbreviations

Following are the abbreviations that have been used in the document:

API: Application Program Interface.
AVP: Attribute Value Pair.

1.Introduction

1.1 Purpose and Scope of Test Plan

The purpose of this document is to describe and document the test cases for the Diameter Base Protocol API. This Document only documents the test cases for black box/functional testing. The test cases included in this document cover the various Diameter Base Protocol scenerios.

2. Document(s) used an input

Software requirement specification document is used to review requirements. The Test cases are written with respect to those requirements.

3.Testing Strategy/Approach

The testing strategy used for this project is white box testing black box testing, regression testing, compliance testing and interoperability testing. The white box testing will be performed by the development team. The black box testing strategy and other testing strategies are the responsibility of QA team. The test cases for black box testing strategy are documented in the current document. Other strategies will require some compliance tools and third party applications that will be used by QA team. One such test suite is *Seagull*. The QA team has customized and configured seagull for testing of Diameter Base Protocol API/Stack compliance and functionality in black box fashion. The Seagull test suite can be found in the Diameter Project folder on the CD delivered with this milestone.

3.1 Functional Testing

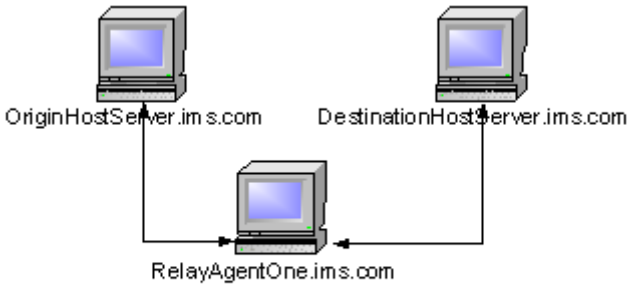
Functional testing is requirement based testing. Tests verify that the system behaves correctly from the user / business perspective and functions are according to the requirements, models or any other design paradigm used to specify the application.

4.Test Cases

Product: Diameter Server Version 1.0

4.1. Deployment Architecture Test Cases

Test Case # 01



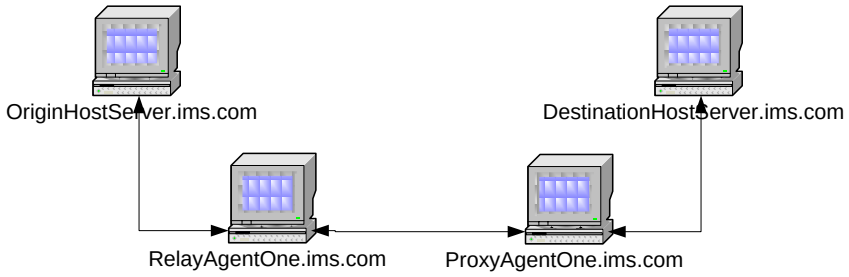
Test Case 01
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RelayAgent**
Test Case ID	0100
Purpose	To test that the message should be received successfully by the Destination Host. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host, when there is only one Relay Agent in the Route traversed by the message.
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Relay Agent Name :** RelyAgentOne.ims.com

Peer Table At OriginServer.ims.com	**Realm Table At OriginServer.ims.com**
Entry No. 01: Host ID : RelyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : RelyAgentOne.ims.com Static/Dynamic : Static Expiry Time :

	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP :** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP :** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host	
Expected Results	In this case the message will be successfully received by the "**DestinationHost.ims.com"** .The Destination Host should return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS** to **"OriginServer.ims.com".**	
Post Condition	Diameter Server must be in message receiving state	

Test Case # 02



Test Case 02
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RelayAgent** and one **ProxyAgent**
Test Case ID	0101
Purpose	To test that the message should be received successfully by the Destination Host with the below given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host , when there is one Relay Agent and one Proxy Agent in the route traversed by the message.
Pre-requisite	**Origin Host Name:** OriginServer.ims.com.ims.com **Destination Host Name:** DestinationHost.ims.com.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Relay Agent Name :** RelyAgentOne.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com

Peer Table at OriginServer.ims.com

Entry No. 01:
Host ID :
RelayAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

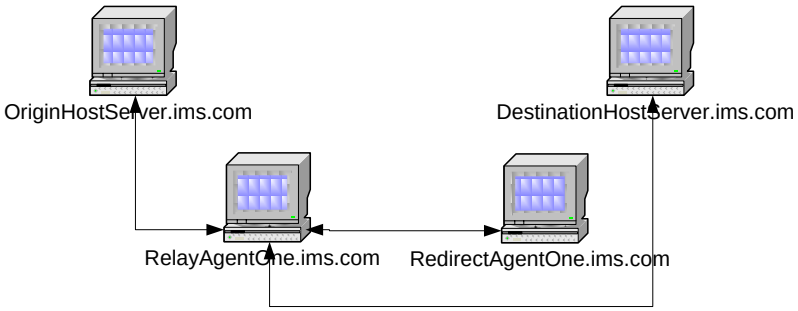
Realm Table at OroginServer.ims.com

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s :
RelyAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :

	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table At ProxyyAgentOne.ims.com **Entry No. 01:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP**: DestinationHost.ims.com **Destination Realm AVP**: DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host	

Expected Results	In this case the message will be successfully received by the "**DestinationHost.ims.com"** .The Destination Host should return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS** to **"OriginServer.ims.com".**
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 03



Test Case 03
Deployment Architecture

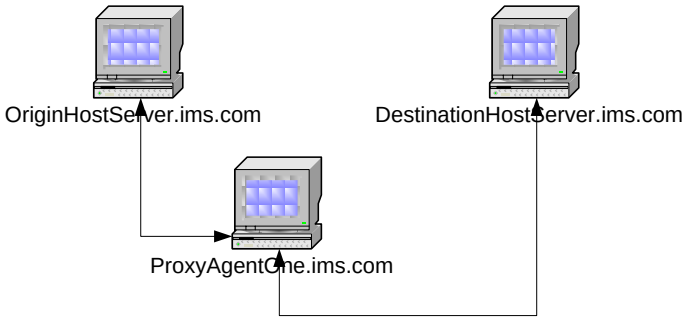
Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RelayAgent** and one **RedirectAgent**
Test Case ID	0102
Purpose	To test that the message should be received successfully by the Destination Host with the below given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	Message delivery from Origin Host to destination Host, when there is one Relay Agent and one Redirect Agent in the Route traversed by the message.
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Relay Agent Name :** RelyAgentOne.ims.com **Redirect Agent Name**: RedirectAgentOne.ims.com

Peer Table at OriginServer.ims.com	**Realm Table at RelayAgentOne.ims.com**
Entry No. 01: Host ID : RelayAgentOne.ims.com	**Entry No. 01** Realm Name :

	StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : RelayAgentOne.ims.com Static/Dynamic : Static Expiry Time :	
	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : RedirectAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : RedirectAgentOne.ims.com Static/Dynamic : Static Expiry Time :	
	Peer Table at RedirectAgentOne.ims.com **Entry No. 01:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RedirectAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : REDIRECT Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :	
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com		

	Application ID AVP: 1234
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host
Expected Results	1. When the message is received by **"RedirectAgentOne.ims.com"** from **"RelayAgentOne.ims.com", t**he **"RedirectAgentOne.ims"** will return the message with E bit set and a Result_Code of **DIAMETER_REDIRECT_INDICATION** .The message must contain an additional Redirect-Host AVP containing DestinationHost.ims.com. 2. Since the **"RelayAgentOne.ims.com"** doesn't have "**DestinationHost.ims.com"** in its peer table, therefore it has to create connection with "DestinationHost.ims.com", and also place it in its peer table. 3. In this case the message will be successfully received by the **"DestinationHost.ims.com"** .The Destination Host must return a message with R bit clear having Result_Code with **DIAMETER_SUCCESS** to **"OriginServer.ims.com".**
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 04



Test Case 04
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **ProxyAgent.**
Test Case ID	0103
Purpose	To test that the message should be received successfully by the Destination Host with the below given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host, when there is only one Proxy Agent in the Route traversed by the message.

<table>
<tr><td>Pre-requisite</td><td>
Origin Host Name: OriginServer.ims.com
Destination Host Name: DestinationHost.ims.com
Destination Realm Name: DestinationRealm.ims.com
<table>
<tr><td><u>Peer Table at OriginServer.ims.com</u>

Entry No. 01:
Host ID : ProxyAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at OriginServer.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s :
ProxyAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td><u>Peer Table at ProxyAgentOne.ims.com</u>

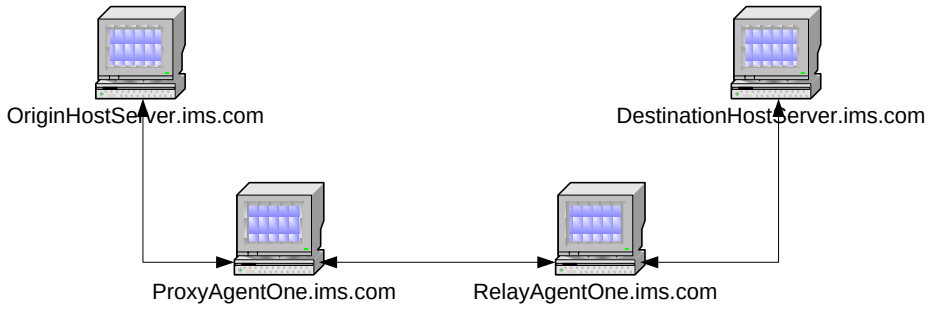
Entry No. 01:
Host ID : OriginServer.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

Entry No. 02:
Host ID : DestinationHost..ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at ProxyyAgentOne.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : PROXY
Server ID/s:
DestinationHost.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
</table>
</td></tr>
<tr><td>Input Data</td><td>Origin Host AVP: OriginServer.ims.com
Destination Host AVP: DestinationHost.ims.com
Destination Realm AVP: DestinationRealm.ims.com
Application ID AVP: 1234</td></tr>
<tr><td>Steps</td><td>1. Send a Massage from Origin Host to Destination Host
2. Receive the Message reply from the Destination Host</td></tr>
<tr><td>Expected Results</td><td>The message will be successfully received by the “DestinationHost.ims.com” .The Destination Host must return a message with R bit Clear having Result_Code of DIAMETER_SUCCESS to “OriginServer.ims.com”.</td></tr>
</table>

Post Condition	The Diameter Server must be in a state to receive message

Test Case # 05



Test Case 05
Deployment Architecture

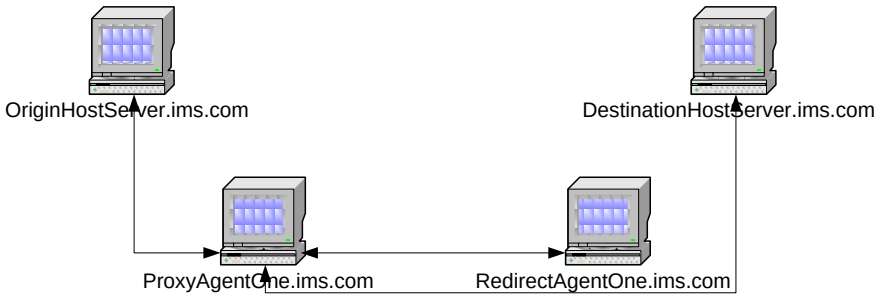
Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **ProxyAgent** and one **RelayAgent.**
Test Case ID	0104
Purpose	To test that the message should be received successfully by the Destination Host with the below given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host, when there is one Proxy Agent and one Relay Agent in the Route traversed by the message.
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com **Relay Agent Name :** RelayAgentOne.ims.com

Peer Table at OriginServer.ims.com	**Realm Table at OriginServer.ims.com**
Entry No. 01: Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s: ProxyAgentOne.ims.com Static/Dynamic : Static

		Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s: RelayAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s :DestinationHost.ims.com Static/Dynamic : Static Expiry Time
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com	

	Application ID AVP: 1234
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host
Expected Results	The message will be successfully received by the **"DestinationHost.ims.com"** .The the Destination Host must return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS**.
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 06



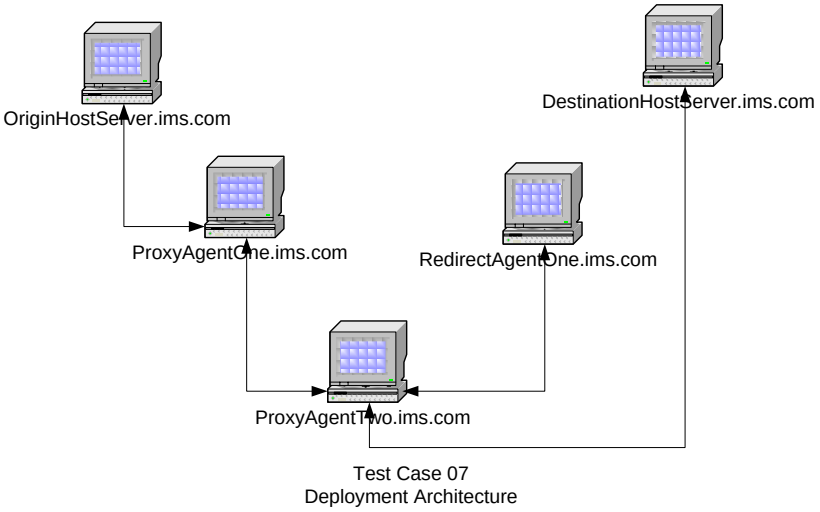
Test Case 06
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **ProxyAgent** and one **RedirectAgent.**
Test Case ID	0105
Purpose	To test that the message should be received successfully by the Destination Host with the below given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host , when there are Proxy Agent and Redirect Agent in the traversed Route
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com **Redirect Agent Name :** RedirectAgentOne.ims.com **Peer Table at** **Realm Table at**

	OriginServer.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**OriginServer.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RedirectAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 03:** Host ID: DestinationHost..ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : RedirectAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RedirectAgentOne.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time :	**Realm Table at RedirectAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : REDIRECT

	TLS Enabled :	Server ID/s: DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP** DestinationHost.ims.com: **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	**1.** When the message is received by the **"RedirectAgentOne.ims.com" AgentOne.ims.com".** Then it must return a message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to "**RelayAgentOne.ims.com"** .The message must contain an additional Redirect-Host AVP containing "**OriginServer.ims.com"**. 2. The message will be successfully received by **"DestinationHost.ims.com"** .The the Destination Host must return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS**.	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 07



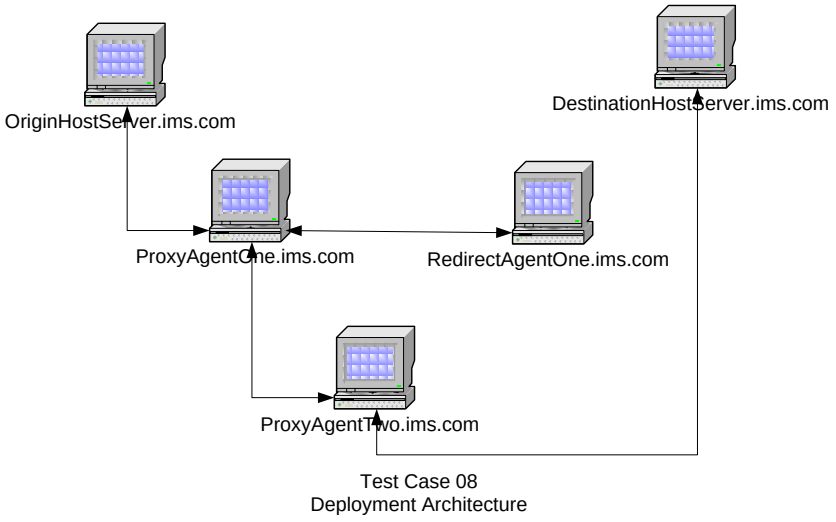
Test Case 07
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having Two **ProxyAgents** and one **RedirectAgent**
Test Case ID	0106
Purpose	To test that the message should be received successfully by the Destination Host with the given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host , when there are two Proxy Agents and Redirect Agent in the traversed Route
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com **Proxy Agent Name:** ProxyAgentTwo.ims.com **Redirect Agent Name :** RedirectAgentOne.ims.com **Peer Table at OriginServer.ims.com** **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Realm Table at OdiginServer.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s :ProxyAgentOne.ims.com

		Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : ProxyAgentTwo.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s: ProxyAgentTwo.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyyAgentTwo.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RedirectAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyyAgentTwo.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : RedirectAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RedirectAgentOne.ims.com **Entry No. 01:** Host ID : RedirectAgentTwo.ims.com	**Realm Table at RedirectAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com

	StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	Application ID : 1234 Local Action : PROXY Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :	
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP**: 1234		
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results		
Expected Results	1. When the message is received by the **"RedirectAgentOne.ims.com"** from "**ProxyAgentOne.ims.com"**. Then it must return a message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to **"ProxyAgentOne.ims.com"** .The message must contain an additional Redirect-Host AVP containing "**DestinationHost.ims.com**." 2. Finally the message will be successfully received by the "DestinationHost.ims.com" .The the Destination Host must return a message of R bit Clear having Result_Code of **DIAMETER_SUCCESS** to "**OriginServer.ims.com".**		
Post Condition	The Diameter Server must be in a state to receive message		

Test Case # 08



Test Case 08
Deployment Architecture

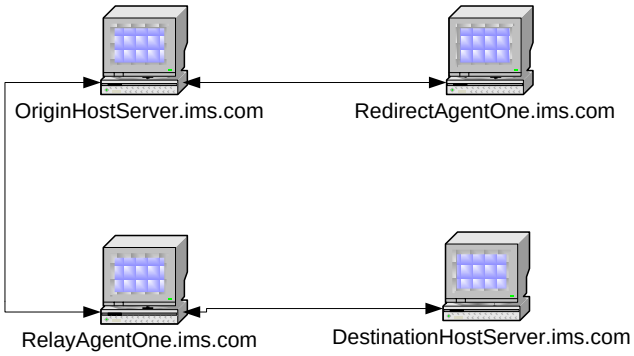
Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having Two **ProxyAgents** and one **RedirectAgent.**
Test Case ID	0107
Purpose	To test that the message should be received successfully by the Destination Host with the given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code **DIAMETER_SUCCESS.**
Scenario	The message delivery from Origin Host to destination Host, when there are Proxy Agent, Redirect Agent and Proxy Agent in the Route traversed by the message.
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Proxy Agent 1 Name:** ProxyAgentOne.ims.com **Redirect Agent Name :** RedirectAgentOne.ims.com **Proxy Agent 2 Name:** ProxyAgentTwo.ims.com

Peer Table at OriginServer.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at OriginServer.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:**	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : RedirectAgentOne.ims.com Static/Dynamic : Static

	Host ID : ProxyAgentTwo.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	Expiry Time :
	<u>Peer Table at RedirectAgentOne.ims.com</u> **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**<u>Realm Table at RedirectAgentOne.ims.com</u>** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : REDIRECT Server ID/s : ProxyAgentTwo.ims.com Static/Dynamic : Static Expiry Time :
	<u>Peer Table at ProxyAgentTwo.ims.com</u> **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**<u>Realm Table at ProxyAgentTwo.ims.com</u>** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com	

	Application ID AVP: 1234
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results
Expected Results	1. When the message is received by **"RedirectAgentOne.ims.com"** from "**ProxyAgentOne.ims.com".** The **"RedirectAgentOne.ims"** will return the message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to **"ProxyAgentOne.ims.com"** .The message must contain an additional Redirect-Host AVP containing **"ProxyAgenttwo.ims.com".** 2. The message will be successfully received by the **"DestinationHost.ims.com"** .The the Destination Host must return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS** to **"OriginServer.ims.com".**
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 09



Test Case 09
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RelayAgents** and one **RedirectAgent**
Test Case ID	0108
Purpose	To test that the message should be received successfully by the

<table>
<tr><td></td><td>Destination Host with the given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of DIAMETER_SUCCESS.</td></tr>
<tr><td>Scenario</td><td>The message delivery from Origin Host to destination Host, when there are two Proxy Agents and one Redirect Agent in the Route traversed by the message.</td></tr>
<tr><td>Pre-requisite</td><td>
Origin Host Name: OriginServer.ims.com
Destination Host Name: DestinationHost.ims.com
Destination Realm Name: DestinationRealm.ims.com
Redirect Agent Name : RedirectAgentOne.ims.com
Relay Agent Name: RelayAgentOne.ims.com
<table>
<tr><td><u>Peer Table at OriginServer.ims.com</u>

Entry No. 01:
Host ID :
RedirectAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

Entry No. 02:
Host ID : RelayAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at OriginServer.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s :
RedirectAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td><u>Peer Table at RedirectAgentOne.ims.com</u>

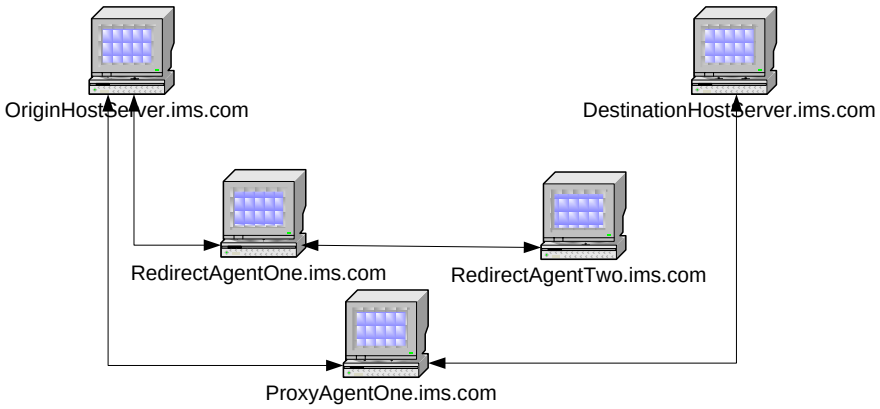
Entry No. 01:
Host ID : OriginServer.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

Entry No. 02:</td><td><u>Realm Table at RedirectAgentOne.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : REDIRECT
Server ID/s :
RelayAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
</table>
</td></tr>
</table>

	Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	
	<u>Peer Table at RlayAgentOne.ims.com</u> **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time :	**<u>Realm Table at RelayAgentOne.ims.com</u>** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP :** DestinationHost.ims.com **Destination Realm AVP :** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	When the message is received by the **"RedirectAgentOne.ims.com"** from **"OriginServer.ims.com".** Then **"RedirectAgentOne.ims.com" must** return the message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to **"OriginServer.ims.com"** .The message must contain an additional Redirect-Host AVP containing DestinationHost.ims.com. The Destination Host must return a message with R bit Clear having Result_Code of DIAMETER_**SUCCESS** to **"OriginServer.ims.com".**	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 10



Test Case 10
Deployment Architecture

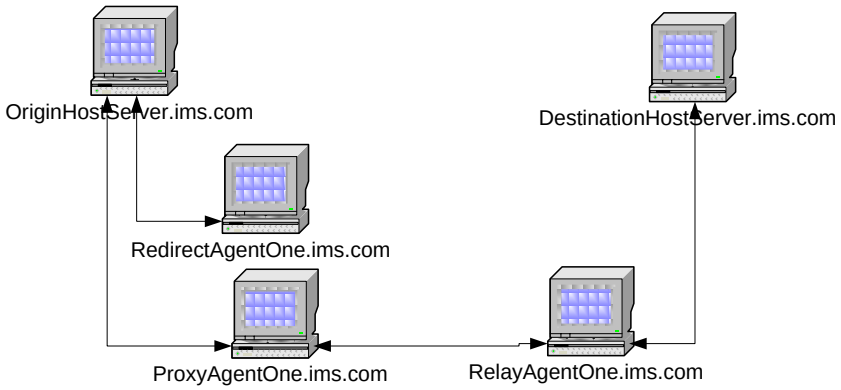
Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having Two **Redirect Agents** and one **ProxyAgent**
Test Case ID	0109
Purpose	To test that the message should be received successfully by the Destination Host with the given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of DIAMETER_**SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host , when there are two Redirect Agents and one Proxy Agent in the Route traversed by the message
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Redirect Agent 1 Name :** RedirectAgentOne.ims.com **Redirect Agent 2 Name :** RedirectAgentTwo.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com

Peer Table at OriginServer.ims.com	**Realm Table at OriginServer.ims.com**
Entry No. 01: Host ID : RedirectAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time :	**Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : REDIRECT Server ID/s :

	TLS Enabled : **Entry No. 02:** Host ID : RedirectAgentTwo.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	RedirectAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : RedirectAgentTwo.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RedirectAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RedirectAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : REDIRECT Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time :	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY

	TLS Enabled : **Entry No. 02:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	1. When the message is received by **"RedirectAgentOne.ims"** from **"OriginServer.ims.com".** The **"RedirectAgentOne.ims" must** return the message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to **"RedirectAgentTwo.ims.com"** .The message must contain an additional Redirect-Host AVP containing DestinationHost.ims.com. 2. After receiving the message by **"RedirectAgentTwo.ims.com"** from **"OriginServer.ims.com", t**he **"RedirectAgentOne.ims"** will return the message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to **ProxyAgentOne.ims.com** .The message must contain an additional Redirect-Host AVP containing DestinationHost.ims.com. The Destination Host should return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS** to **"OriginServer.ims.com".**	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 11



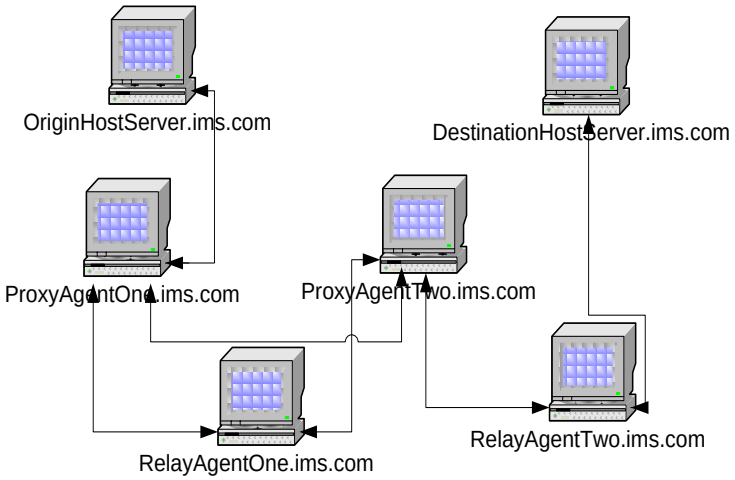
Test Case 11
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RedirectAgent, one ProxyAgent** and one **RelayAgent.**
Test Case ID	0110
Purpose	To test that the message should be received successfully by the Destination Host with the given Peer Table and Realm Table configurations. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.
Scenario	The message delivery from Origin Host to destination Host , when there is a Redirect Agents ,Proxy Agent and Relay Agent in the Route traversed by the message
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Redirect Agent Name :** RedirectAgentOne.ims.com **Proxy Agent Name :** ProxyAgentOne.ims.com **Relay Agent Name:** RelayAgentOne.ims.com **Peer Table at OriginServer.ims.com** **Entry No. 01:** Host ID : RedirectAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** **Realm Table at OriginServer.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : RedirectAgentOne.ims.com Static/Dynamic : Static Expiry Time :

	Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	
	Peer Table at RedirectAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RedirectAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234

	Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	Local Action : PROXY Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP**: DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	1. When the message is received by **"RedirectAgentOne.ims.com"** from "**OriginServer.ims.com", t**he **"RedirectAgentOne.ims" must** return the message with E bit set and Result_Code of **DIAMETER_REDIRECT_INDICATION** to **"ProxyAgentOne.ims.com"** .The message must contain an additional Redirect-Host AVP containing DestinationHost.ims.com. The Destination Host must return a message with R bit Clear having Result_Code of **DIAMETER_SUCCESS**.	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 12



Test Case 12
Deployment Architecture

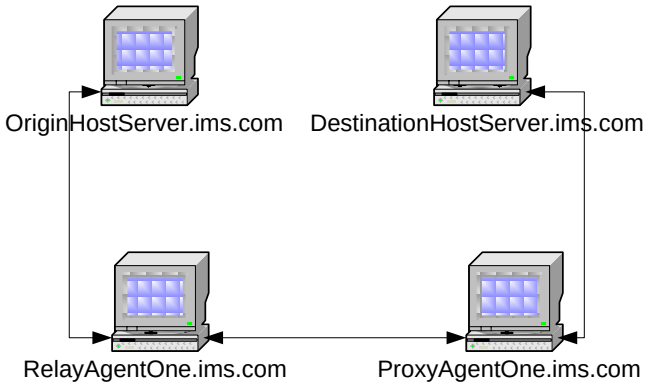
Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having two **ProxyAgents** and two **RelayAgents**
Test Case ID	0111
Purpose	To test the LOOP_DETECTION on Diameter Proxy Agent, when it finds its own name in Route_Record AVP. The Result_Code **"ProxyAgentOne.ims.com"** should return a Result_Code DIAMETR_**LOOP_DETECTED** when received the same Message from **"ProxyAgentTwo.ims.com"** as sent earlier to **"RelayAgentOne.ims.com"**.
Scenario	The message delivery from Origin Host to destination Host, when there is a Proxy Agent, Relay Agents, Proxy Agent and Relay Agent in the Route traversed by the message.
Pre-requisite	**Origin Host Name:** OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Proxy Agent 1 Name :** ProxyAgentOne.ims.com **Relay Agent 1 Name:** RelayAgentOne.ims.com **Proxy Agent 2 Name :** ProxyAgentTwo.ims.com **Relay Agent 2 Name:** RelayAgentTwo.ims.com **Peer Table at OriginServer.ims.com** / **Realm Table at OriginServer.ims.com**

	Entry No. 01: Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : RelayAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : ProxyAgentTwo.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : ProxyAgentTwo.ims.com Static/Dynamic : Static Expiry Time :

	Peer Table at ProxyyAgentTwo.ims.com **Entry No. 01:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : RelayAgentTwo.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyAgentTwo.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : RelayAgentTwo.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RlayAgentTwo.ims.com **Entry No. 01:** Host ID : ProxyAgentTwo.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 03:** Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentTwo.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : PROXY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com	

	Application ID AVP: 1234
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results
Expected Results	After successful receival of message by the "**RelayAgentTwo.ims.com**", it will forward the message to "ProxyAgentOne.ims.com "according to its Realm Routing table entry. Since each Agent adds a Record_Rout AVP with its own name, therefore when the message will be received by "ProxyAgentOne.ims.com" it will find its own name in Record_Route AVP. Hence should return a message with E-bit set having Result_Code of **DIAMETR_LOOP_DETECTED.**
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 13



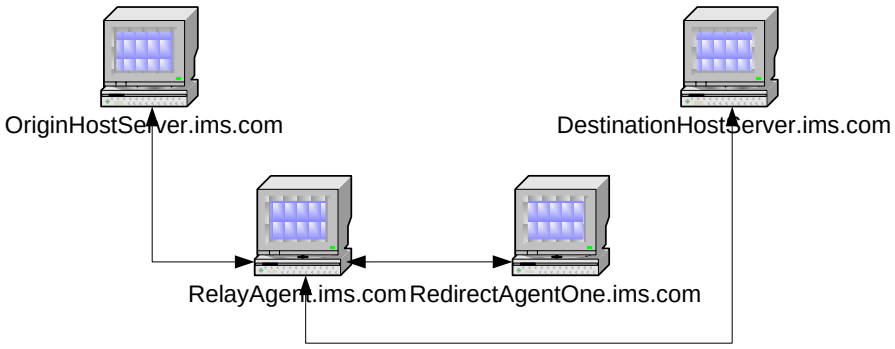
Test Case 13
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RelayAgent** and one **ProxyAgent .**
Test Case ID	0112
Purpose	To test that the message processing on Proxy Agent when it receives a message for a realm which is unreachable according to it the given Peer Table and Realm Table configurations. The **"ProxyAgentOne.ims.com"** should return Result_Code of **DIAMETER_REALM_NOT_SERVED**
Scenario	The message delivery from Origin Host to destination Host , when there is a Relay Agents and Proxy Agent in the Route traversed by the message
Pre-requisite	

	Origin Host Name: OriginServer.ims.com.ims.com **Destination Host Name:** DestinationHost.ims.com.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Relay Agent Name :** RelyAgentOne.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com	
	Peer Table at OriginServer.ims.com **Entry No. 01:** Host ID : RelyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at OriginServer.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : RelyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at RlayAgentOne.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled : **Entry No. 02:** Host ID : OriginServer.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RelayAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealmTwo.ims.com Application ID : 1234 Local Action : RELAY Server ID/s : ProxyAgentOne.ims.com Static/Dynamic : Static Expiry Time :
	Peer Table at ProxyAgentOne.ims.com **Entry No. 01:** Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at ProxyAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : RELAY Server ID/s :

	Entry No. 02: Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	The message with E bit set must return by **"ProxyAgent.one.ims.com"** with Result_Code of **DIAMETER_REALM_NOT_SERVED**. Because the **"ProxyAgentOne.ims.com"** don't have any entry for **"DestinationRealm.ims.com"** in its Realm Routing Table.	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 14



Test Case 14
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **RelayAgent** and one **RedirectAgent**
Test Case ID	0113
Purpose	To test the message processing by a relay agent when it is received for unsupported application. The "**RelayAgentOne.ims.com"** should return a Result_Code of **DIAMETER_APPLICATION_UNSUPPORTED** to

<table>
<tr><td></td><td colspan="2">"OriginServer.ims.com".</td></tr>
<tr><td>Scenario</td><td colspan="2">The message delivery from Origin Host to destination Host , when there is a Relay Agent and redirect Agent in the Route traversed by the message</td></tr>
<tr><td>Pre-requisite</td><td colspan="2">Origin Host Name: OriginServer.ims.com
Destination Host Name: DestinationHost.ims.com
Destination Realm Name: DestinationRealm.ims.com
Relay Agent Name : RelyAgentOne.ims.com
Redirect Agent Name: RedirectAgentOne.ims.com</td></tr>
<tr><td></td><td><u>Peer Table at OriginServer.ims.com</u>

Entry No. 01:
Host ID : RelayAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at OriginServer.ims.com</u>

Entry No. 01
Realm Name : DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s : RelayAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td></td><td><u>Peer Table at RlayAgentOne.ims.com</u>

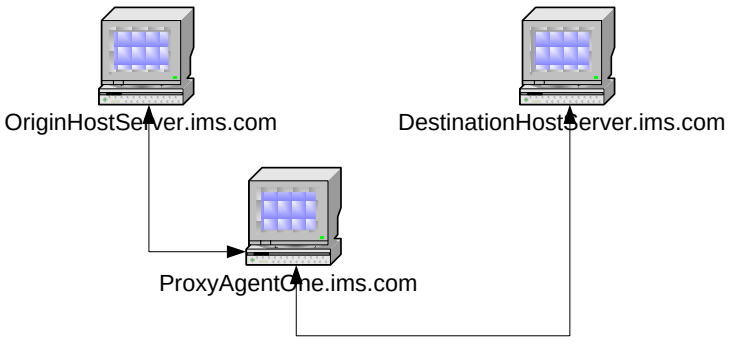
Entry No. 01:
Host ID : RedirectAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

Entry No. 02:
Host ID : OriginServer.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at RelayAgentOne.ims.com</u>

Entry No. 01
Realm Name : DestinationRealm.ims.com
Application ID : 1235
Local Action : RELAY
Server ID/s : RedirectAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td></td><td><u>Peer Table at RedirectAgentOne.ims.com</u></td><td><u>Realm Table at RedirectAgentOne.ims.com</u></td></tr>
</table>

	Entry No. 01: Host ID : RelayAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1235 Local Action : REDIRECT Server ID/s : DestinationHost.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	**"RelayAgentOne.ims.com"** must return a message with E bit set and Result_Code of **DIAMETER_APPLICATION_UNSUPPORTED** to "**OriginServer.ims.com".**	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 15



Test Case 15
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having only one **ProxyAgent**
Test Case ID	0114
Purpose	To test that the message processing on proxy agent for which

<table>
<tr><td></td><td>the proxy agent is unable to deliver according to its Peer Table and Realm Table configurations. The "ProxyAgentOne.ims.com" should return a Result_Code of UNABLE_TO_DELIVER to "OriginServer.ims.com".</td></tr>
<tr><td>Scenario</td><td>The message delivery from Origin Host to destination Host, when there is only one Proxy Agent in the Route traversed by the message.</td></tr>
<tr><td>Pre-requisite</td><td>
Origin Host Name: OriginServer.ims.com
Destination Host Name: DestinationHost.ims.com
Destination Realm Name: DestinationRealm.ims.com
Proxy Agent Name : ProxyAgentOne.ims.com
<table>
<tr><td><u>Peer Table at OriginServer.ims.com</u>

Entry No. 01:
Host ID :
ProxyAgentOne.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at OriginServer.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s :
ProxyAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td><u>Peer Table at ProxyAgentOne.ims.com</u>

Entry No. 01:
Host ID : OriginServer.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

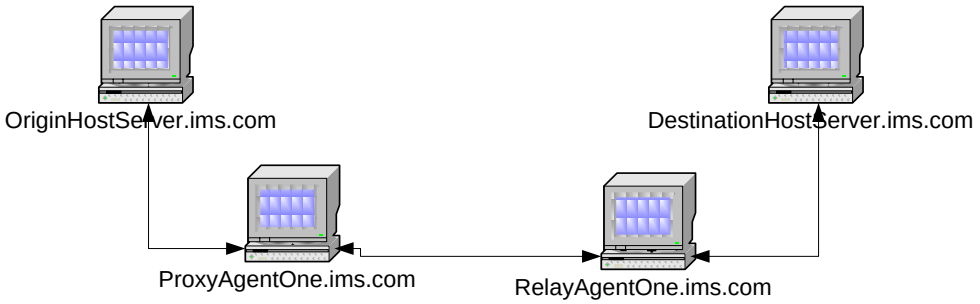
Entry No. 02:
Host ID :
DestinationHost.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at ProxyyAgentOne.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1235
Local Action : PROXY
Server ID/s :
DestinationHostOne.ims.com
Static/Dynamic : Static
Expiry Time :

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1236
Local Action : PROXY
Server ID/s :
DestinationHostTwo.ims.com</td></tr>
</table>
</td></tr>
</table>

	Static/Dynamic : Static Expiry Time	
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	**"ProxyAgentOne.ims.com"** should return a message with E bit set and Result_Code of **UNABLE_TO_DELIVER** to "**OriginServer.ims.com**".	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 16



Test Case 16
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one ProxyAgent **and** one **RelayAgent.**	
Test Case ID	0115	
Purpose	To test that the message processing on proxy agent for which the proxy agent is unable to deliver according to its Peer Table and Realm Table configurations. The "**ProxyAgentOne.ims.com"** should return a Result_Code of **UNABLE_TO_DELIVER** to "**OriginServer.ims.com".**	
Scenario	The message delivery from Origin Host to destination Host , when there are Proxy Agent and Relay Agent in the Route traversed by the message	
	Origin Host Name: OriginServer.ims.com **Destination Host Name:** DestinationHost.ims.com **Destination Realm Name:** DestinationRealm.ims.com **Proxy Agent Name:** ProxyAgentOne.ims.com	

<table>
<tr><td rowspan="4">Pre-requisite</td><td colspan="2">Relay Agent Name : RelayAgentOne.ims.com</td></tr>
<tr><td><u>Peer Table at OriginServer.ims.com</u>

Entry No. 01:
Host ID :
ProxyAgentOne.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at OriginServer.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s :
ProxyAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td><u>Peer Table at ProxyAgentOne.ims.com</u>

Entry No. 01:
Host ID : OriginServer.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

Entry No. 02:
Host ID :
RelayAgentOne.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at ProxyAgentOne.ims.com</u>

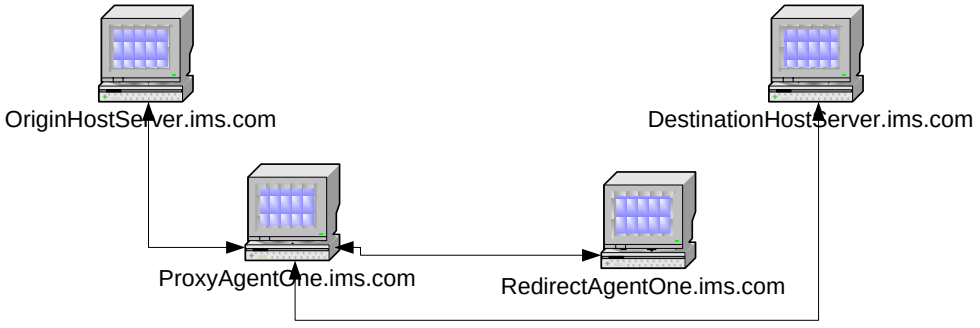
Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : PROXY
Server ID/s :
RelayAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td><u>Peer Table at RlayAgentOne.ims.com</u>

Entry No. 01:
Host ID :
ProxyAgentOne.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at RelayAgentOne.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : PROXY
Server ID/s :
DestinationHost.ims.com</td></tr>
</table>

	Entry No. 02: Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** Missing **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	**"ProxyAgentOne.ims.com"** must return a message with E bit set and Result_Code of **UNABLE_TO_DELIVER** to **"OriginServer.ims.com".**	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 17



Test Case 17
Deployment Architecture

Test Case Title	Message Flow from **OriginHost** to **DestinationHost** in a Deployment Architecture having one **ProxyAgent** and one **RedirectAgent.**
Test Case ID	0116
	To test that the message should be received successfully by the

<table>
<tr><td>Purpose</td><td colspan="2">Destination Host with the given Peer Table and Realm Table configurations. The “RedirectAgentOne.ims” should return Result_Code of DIAMETER_UNABLE_TO_DELIVER to “RelayAgentOne.ims.com”.</td></tr>
<tr><td>Scenario</td><td colspan="2">The message delivery from Origin Host to destination Host , when there are Proxy Agent and redirect Agent in the Route traversed by the message</td></tr>
<tr><td rowspan="3">Pre-requisite</td><td colspan="2">Origin Host Name: OriginServer.ims.com
Destination Host Name: DestinationHost.ims.com
Destination Realm Name: DestinationRealm.ims.com
Proxy Agent Name: ProxyAgentOne.ims.com
Redirect Agent Name : RedirectAgentOne.ims.com</td></tr>
<tr><td><u>Peer Table at OriginServer.ims.com</u>

Entry No. 01:
Host ID :
ProxyAgentOne.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at OriginServer.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : RELAY
Server ID/s :
ProxyAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
<tr><td><u>Peer Table at ProxyAgentOne.ims.com</u>

Entry No. 01:
Host ID :
OriginServer.ims.com StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :

Entry No. 02:
Host ID :
RedirectAgentOne.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at ProxyAgentOne.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 1234
Local Action : PROXY
Server ID/s :
RedirectAgentOne.ims.com
Static/Dynamic : Static
Expiry Time :</td></tr>
</table>

	Entry No. 03: Host ID : DestinationHost.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	
	Peer Table at RedirectAgentOne.ims.com **Entry No. 01:** Host ID : ProxyAgentOne.ims.com StatusT : Static/Dynamic : Static Expiry Time : TLS Enabled :	**Realm Table at RedirectAgentOne.ims.com** **Entry No. 01** Realm Name : DestinationRealm.ims.com Application ID : 1234 Local Action : REDIRECT Server ID/s : DestinationHostTwo.ims.com Static/Dynamic : Static Expiry Time :
Input Data	**Origin Host AVP:** OriginServer.ims.com **Destination Host AVP:** DestinationHost.ims.com **Destination Realm AVP:** DestinationRealm.ims.com **Application ID AVP:** 1234	
Steps	1. Send a Massage from Origin Host to Destination Host 2. Receive the Message reply from the Destination Host and check it for the expected results	
Expected Results	When the message is received by **"RedirectAgentOne.ims.com"** from "**RelayAgentOne.ims.com".** The"**RedirectAgentOne.ims.com"** will return the message with E bit set and Result_Code of **DIAMETER_UNABLE_TO_DELIVER** to **"RelayAgentOne.ims.com"**.	
Post Condition	The Diameter Server must be in a state to receive message	

Test Case # 18

Test Case Title	To test Local Routing Success
Test Case ID	00097
Purpose	To test the message processing on Diameter server when it receives a message destined to one of its local application. The Destination Host must return a Result_Code of **DIAMETER_SUCCESS**.

<table>
<tr><td>Scenario</td><td>The message Processing when it is received by the Diameter Host, destined to one of its supporting local Application.</td></tr>
<tr><td>Pre-requisite</td><td>
<u>Application CallBackMappingTable at DestinationHost.ims.com</u>

Entry 01:
Application ID : 12345
Application Call Back : AppOneCallBack()

Entry 02:
Application ID : 12345
Application Call Back : AppTwoCallBack()

<table>
<tr><td><u>Peer Table at DestinationHost.ims.com</u>

Entry No. 01:
Host ID :
DestinationHost.ims.com
StatusT :
Static/Dynamic : Static
Expiry Time :
TLS Enabled :</td><td><u>Realm Table at destiantionHost.ims.com</u>

Entry No. 01
Realm Name :
DestinationRealm.ims.com
Application ID : 12345
Local Action : LOCAL
Server ID/s :
Static/Dynamic : Static
Expiry Time :</td></tr>
</table>
</td></tr>
<tr><td>Input Data</td><td>
<u>Message received with Following Values</u>

Destination Host Name : DestinationHost.ims.com
Destination Realm : DestinationRealm.ims.com
Destination Application ID : 12345
</td></tr>
<tr><td>Steps</td><td>Receive the Message reply from the Origin Host</td></tr>
<tr><td>Expected Results</td><td>The message will be successfully delivered to the application having ID 12345, and a DIAMETER_SUCCESS code will be returned by the “DestinationHost.ims.com” to the origin Host.</td></tr>
<tr><td>Post Condition</td><td>The Diameter Server must be in a state to receive message</td></tr>
</table>

4.2 Message Validation Test Cases

Test Case # 19

Test Case Title	To test a Diameter Message with AVP not allowed
Test Case ID	00062
Purpose	To test when a peer receives a CER (Capability Exchange request Message) containing a Disconnect Cause AVP which is not allowed in CER Message. In that case the peer should return **DIAMETER_AVP_NOT_ALLOWED** error code; The Failed_AVP AVP must be present in the CEA Message. Failed_AVP AVP must contain the copy of Disconnect Cause AVP received in CER Message
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER (Capability Exchange Request Message)** **CER Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 **Message Data** = Must contain the below given AVP and all other AVPs defined by the RFC for this message **CER Message AVPs** **Disconnect_Cause AVP** (Ungrouped) AVP Header AVP code = 273 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Clear (

	which is reserved Bit) AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DONT_WANT_TO_TALK_TO_YOU
Steps	1. Send DPR (Disconnect-Peer Request Message) to a peer 2. Receive the DPA (Disconnect-Peer Answer Message) from the peer and check it for expected results
Expected Results	**CEA (Capability Exchange Answer Message)** **CEA Message Header** Version = Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message data = Must contain the below given AVPs and all other AVPs defined by the RFC for this message **CEA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_NOT_ALLOWED_AVP **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Since the AVP is of type Grouped, hence must contain the following

	AVPs as Data **Disconnect_Cause AVP** (Ungrouped) AVP Header AVP code = 273 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Clear AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DONT_WANT_TO_TALK_TO_YOU
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 20

Test Case Title	To test a message processing other then CER/CEA having Host_IP_Address AVP
Test Case ID	00075
Purpose	To test that when a diameter peer receives a DPR (Disconnect Peer Request Message) having Host_IP_Address AVP. In that case the peer should return Result_Code AVP with error code DIAMETER_AVP_NOT_ALLOWED. The message must contain the Failed_AVP of type grouped having a copy of Host_IP_Address AVP that was received.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**DPR (Disconnect-Peer Request Message)** **DPR Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 282 **Message Data** = Must contain the below given AVP and all other AVPs defined by the RFC for this message

	DPR Message AVPs **Host_IP_Address AVP** (Ungrouped) AVP Header AVP code = 257 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Clear (which is reserved Bit) AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = OriginHost.ims.com
Steps	1. Send DPR (Disconnect-Peer Request Message) to a peer **2.** Receive the DPA (Disconnect-Peer Answer Message) from the peer and check it for expected results
Expected Results	**DPA (Capability Exchange Answer Message)** **DPA Message Header** **Version =** Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 282 Message data = Must contain the below given AVPs and all other AVPs defined by the RFC for this message **DPA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_NOT_ALLOWED_AVP **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279

	AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Since the AVP is of type Grouped, hence must contain the following **Host_IP_Address** AVPs as Data **Host_IP_Address AVP** (Ungrouped) AVP Header AVP code = 257 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit Clear AVP flag 5th,6th,7th,8th Bits = All Clear AVP Length = Vendor ID = Empty AVP data = OriginHost.ims.com
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 21

Test Case Title	To test peer behavior on receiving an ASR Message with E and R bits set at the same time
Test Case ID	00068
Purpose	To test that when a peer receives an ASR message having E and R -Bits Set in its message header. The peer should return a message ASA with command code 274 having a Result_Code AVP containing an error code DIAMETER_INVALID_HDR_BITS as AVP value.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**ASR (Abort-Session- Request Message)** ASR Message Header Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 274

	Message Data = Must contain all the AVPs defined by RFC for ASR message
Steps	1. Send the ASR (Abort-Session- Request Message) to Peer. 2. Receive ASA (Abort-Session- Answer Message) from peer and Check it for expected results.
Expected Results	**ASA (Abort-Session- Answer Message)** **ASA Message Header** Version = Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 274 Message Data = Must contain the AVPs Given Below and AVPs defined by RFC for ASA message **ASA Message AVPs** **Result-Code AVP** AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_INVALID_HDR_BITS
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 22

Test Case Title	To test peer behavior on receiving request Message having R and E bits set in message header and one mandatory AVP missing
Test Case ID	00070
Purpose	To test the message processing when the peer receives an ASR message with R and E-Bits set at the same time, and also Origin-Host mandatory AVP missing. In that case the peer should return the Result_Code AVP with error code DIAMETER_INVALID_AVP_BITS. Also there will be no information about Origin-Host missing AVP in the message. The answer message will contain info about the first error

	encountered.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	
Steps	1. Send the ASR (Abort-Session- Request Message) to peer. 2. Receive ASA (Abort-Session- Answer Message) from peer and Check it expected results.
Expected Results	**ASA (Abort-Session- Answer Message)** **ASA Message Header** Version = Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 274 Message Data = Must contain the **Result-Code AVP** Given Below and AVPs defined by RFC for ASA message **ASA Message AVPs** **Result-Code AVP** AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = All Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_INVALID_HDR_BITS
Post Condition	**The Diameter Server must be in a state to receive message**

Test Case # 23

Test Case Title	To test the behavior of a Diameter Peer when it receives a DPR Message having unknown mandatory AVP
Test Case ID	00060
Purpose	To test that when a peer receives a DPR (Diameter Peer Disconnect Request Message) with unknown ACK_THE_MSG AVP, and also the AVP has M-Bit set. In that case the peer should return Message of Command code 282 with E-Bit set.

	The message should contain Result_Code AVP with error code DIAMETER_AVP_UNSUPPORTED. When it received a message having AVP of unknown code with M bit set. The message with this error code must contain Failed_AVP AVP of type grouped. The Failed_AVP AVP must include a copy of ACK_THE_MSG AVP that was received by the peer.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**DPR (Disconnect-Peer Request Message)** **DPR Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 282 Message Data = Must contain the below given AVP and all other AVPs defined by the RFC for this message **DPR Message AVPs** **ACK_AVP AVP** (Ungrouped) AVP Header AVP code = 600 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Clear (which is reserved Bit) AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = ACK_THE_MSG
Steps	Send DPR (Disconnect-Peer Request Message) to a peer . Receive the DPA (Disconnect-Peer Answer Message) from the peer and check it for expected results
Expected Results	**DPA (Capability Exchange Answer Message)** **DPA Message Header** **Version =** Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear

	Message Command Code = 282 Message data = Must contain the below given AVPs and all other AVPs defined by the RFC for this message **DPA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_AVP_UNSUPPORTED **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP Data = since it is grouped AVP , hence must contain the AVP given below. **ACK_AVP AVP** (Ungrouped) AVP Header AVP code = 600 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Clear (which is reserved Bit) AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = ACK_THE_MSG
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 24

Test Case Title	To test the behavior of a diameter agent when it receives a message ACR/ACA with P bit clear, but having Destination_Realm AVP
Test Case ID	00078
Purpose	To test that when a diameter agent receives an ACR or ACA message with P-Bit clear but having Destination-Realm AVP. In that case the Diameter agent should return a message of command code 271 with E bit set. The message should contain the Result_Code AVP with error code DIAMETER_AVP_NOT_ALLOWED. Also the Failed_AVP AVP of type grouped must be included in the message. Failed_AVP should contain a copy of Destination_Realm AVP that was received in a message.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**ACR (Accounting Request Message)** ACR Message Header Version = Message Length = Message R-Bit = Set (Must) Message P-Bit = Clear (Must) Message E-Bit = Clear (Must) Message T-Bit = Clear (Optional) Message 5th,6th, 7th,8th = All Clear Message Command Code = 271 Message Data = Must contain the below given AVPs As message data and other AVPs defined by RFC for the Message **ACR (Accounting Request Message AVPs)** **Destination-Realm AVP** AVP Header AVP code = 283 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DestinationRealm.ims.com •
Steps	1. Send **ACR (Accounting- Request Message)** to a peer 2. Receive **ACA (Accounting- Answer Message)** from the peer and check it for expected results
Expected Results	**ACA (Accounting Answer Message)** ACA Message Header Version =

	Message Length = Message R-Bit = Clear (Must) Message P-Bit = Clear (Must) Message E-Bit = Set (Must) Message T-Bit = Clear (Optional) Message 5th,6th, 7th,8th = All Clear Message Command Code = 271 Message data = Must contain the blew given AVPs as Message data and other AVPs defined by RFC for this message. **ACA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear (Must) AVP flag M-Bit = Set (Must) AVP flag P-Bit = Clear (Optional) AVP flag 4th,5th,6th,7th,8th Bits = Clear (Must) AVP Length = Vendor ID = Default AVP data = DIAMETER_AVP_NOT_ALLOWED **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = since it is grouped AVP so must contain an Origin-Host AVP as given below **Destination-Realm AVP** AVP Header AVP code = 283 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DestinationRealm.ims.com

Post Condition	The Diameter Server must be in a state to receive message

Test Case # 25

Test Case Title	To test the behavior of Diameter peer when it receives ACR message having P bit set in its header but missing Destination_Realm AVP
Test Case ID	00077
Purpose	To test when a diameter peer receives an ACR (Accounting-Request) message with P-Bit set in its Message Header , with missing Destination-realm AVP . In that case the diameter node must return a ACA message with E bit set in message Header. The message should contain Result_Code AVP with error code DIAMETER_MISSING_AVP. Failed_AVP containing a copy of Destination_Realm AVP must be included in the message. Failed_AVP will contain copy of Destination_Realm AVP will be with its code and other fields as expected. The value field will be having minimum length filled with zeros.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in message receiving state
Input Data	**ACR (Accounting Request Message)** ACR Message Header Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 271 Message Data = Must contain all the AVPs defined by RFC for this message except the Destination_Realm AVP which is mandatory if P (Proxiable) bit is set.
Steps	1. Send ACR (Accounting- Request Message) to a peer. 2. Receive ACA (Accounting- Answer Message) from the peer and check it for expected results
Expected Results	**ACA (Accounting Answer Message)** ACA Message Header Version = Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set

	Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 271 Message data = Must contain the below given AVPs as Message data and other AVPs defined by RFC for this message. **ACA Message AVPs** **Result_Code AVP (Ungrouped)** AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear (Must) AVP Length = Vendor ID = Default AVP data = DIAMETER_MISSING_AVP **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = since it is grouped AVP so must contain an Origin-Host AVP as given below **Destination-Realm AVP** AVP Header AVP code = 283 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = filled with Zeros
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 26

Test Case Title	To test the behavior of peer when it receives an AVP with unknown flag bits set
Test Case ID	00063
Purpose	To test that when a peer receives an STR(Session Termination Request Message) with Destination_Realm AVP, and in the header of Destination_Realm AVP V-Bit, M-Bit and 4th-Bit(which is reserve Bit) are set. In that case the peer must return Result_Code AVP with error code DIAMETER_INVALID_AVP_BITS. The message must contain Failed_AVP AVP of type grouped. This grouped AVP must contain a copy of Destination_Realm AVP received by the peer in STR message.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**STR (Session Termination Request Message)** STR Message Header Version = Message Length = Message R-Bit = Set Message P-Bit = Set Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 275 Message Data = Must contain the blew given AVPs as Message data and other AVPs defined by RFC for this message **STA Message AVPs** **Origin-Realm AVP** (Ungrouped) AVP Header AVP code = 296 AVP Flag V-Bit = Set AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Set AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Default AVP data = OriginRealm.ims.com
Steps	1. Send STR (Session-Termination Request Message) to a peer. 2. Receive STA (Session-Termination Answer Message) from the peer and check it for expected results

Expected Results	**STA (Session Termination Answer Message)** STA Message Header Version = Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 275 Message data = Must contain the below given AVPs as Message data and other AVPs defined by RFC for this message. **STA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Filled with ZEEROS AVP data = DIAMETER_INVALID_AVP_BITS **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = since it is grouped AVP so must contain an **Origin-Realm AVP** causes the failure **Origin-Realm AVP (** Ungrouped) AVP Header AVP code = 296 AVP Flag V-Bit = Set AVP flag M-

	Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Set AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = OriginRealm.ims.com
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 27

Test Case Title	To test the message processing of CER/CEA with P bit set
Test Case ID	00079
Purpose	To test that when a Diameter agent receives a CER (Capability Exchange Message Request Message) with P-Bit set in Message Header. In that case the Diameter agent should return a CEA (Capability Exchange Answer Message) with E bit set. The message should contain the Result_Code AVP with error code DIAMETER_INVALID_HDR_BITS.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**CER (Capability Exchange Request Message)** **CER Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Set Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message Data = Must contain all the AVPs defined by the RFC for this message
Steps	1. Send CER (Capability Exchange Request Message) to a peer 2. Receive the CEA (Capability Exchange Answer Message) from the peer and check it for expected results
Expected Results	**CEA (Capability Exchange Answer Message)**

	CEA Message Header **Version =** Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message data = Must contain the below given AVPs and all other AVPs defined by the RFC for this message **CEA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_INVALID_HDR_BITS
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 28

Test Case Title	To test the peer behavior on receiving CER message having two Origin_Host AVPs
Test Case ID	00073
Purpose	To test message processing when a peer received a CER message with two Origin_Host AVPs. On receiving the message the peer should return a message of command code 257 with E bit set. The message must contain the error code with Result_Code AVP having error code DIAMETER_AVP_OCCURS_TOO_MANY_TIMES. The Failed_AVP AVP should be included in the message. The Failed_AVP should contain a copy of the first instance of the Origin_Host AVP.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**CER (Capability Exchange Request Message)** **CER Message Header**

	Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Data = Must contain the below given AVPs **CER Message AVPs** **Origin-Host AVP** AVP Header AVP code = 264 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = OriginServer.ims.com **Origin-Host AVP** AVP Header AVP code = 264 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = OriginServerOne.ims.com
Steps	1. Send CER (capability Exchange Request Message) to a peer. 2. Check the CEA (capability Exchange Answer Message)
Expected Results	**CEA (Capability Exchange Answer Message)** **CEA Message Header** Version = Message Length = Message R-Bit = Clear (Must) Message P-Bit = Clear (Must) Message E-Bit = Set (Must) Message T-Bit = Clear (Optional) Message 5th,6th, 7th,8th = All Clear (Must) Message data = Must contain the blew given AVPs as Message data. **CEA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header

	AVP code = 268 AVP Flag V-Bit = Clear (Must) AVP flag M-Bit = Set (Must) AVP flag P-Bit = Clear (Optional) AVP flag 4th,5th,6th,7th,8th Bits = Clear (Must) AVP Length = Vendor ID = Empty AVP data = DIAMETER_AVP_OCCURS_TOO_MANY_TIMES **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = since it is grouped AVP so must contain an Origin-Host AVP as Data which is given below **Origin-Host AVP** AVP Header AVP code = 264 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = OriginServer.ims.com
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 29

Test Case Title	To test the peer behavior when it receives a DPR message without Disconnect_Cause AVP

Test Case ID	00076
Purpose	To test that when the peer receives a DPR(Disconnect Peer Request Message) having Disconnect_Cause AVP missing .In that case the peer should return a DPA (Disconnect Peer Answer Message)with E bit set , containing Result_Code AVP with error code DIAMETER_AVP_MISSING. Also the Failed_AVP must be including in the message. The Failed_AVP must contain a copy of Disconnect_Cause AVP with it expected fields set, and the value field must be of integer 32 filled with zero.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**DPR (Disconnect-Peer Request Message)** DPR Message Header Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 282 Message Data = Must contain all the AVPs defined by RFC for this message except Disconnect_Cause AVP
Steps	1. Send DPR (Disconnect-Peer Request Message) to a peer. 2. Receive DPA (Disconnect-Peer Answer Message) from the peer and check it for expected result
Expected Results	**DPA (Disconnect-Peer Answer Message)** **DPA Message Header** Version = Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 282 Message data = Must contain the below given AVPs as Message data and other AVPs defined by RFC for this message. DPA Message AVPs **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear

	AVP Length = Vendor ID = Empty AVP data = DIAMETER_AVP_MISSING **Failed_AVP AVP** (Grouped contains copy of missing AVP which is Disconnect_Cause AVP) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear VP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = since it is grouped AVP so must contain an Disconnect_Cause AVP as Data which is given below **Disconnect_Cause AVP** (Ungrouped) AVP Header AVP code = 273 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th Bit = Clear (which is reserved Bit) AVP flag 5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = 32bit field filled with ZEROS
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 30

Test Case Title	To test the peer behavior when it receives a message with two contradicting AVPs
Test Case ID	00072
Purpose	To test that when a peer receives a CER (Capability Exchange Request Message) with two Inband_Security AVPs, with AVP

	values contradicting each other. In that case the peer should return a CEA (Capability Exchange Answer Message)with E-Bit set. The message should contain the Result_Code AVP with error code DIAMETER_CONTRADICTING_AVP. The message should also contain the Failed_AVP. The Failed_AVP should contain the contradicting AVP pair.
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**CER (Capability Exchange Request Message)** **CER Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message Data = Must contain the below given AVPs and all other AVPs defined by the RFC for this message as data **CER Message AVPs** **Inband_Security AVP** (Ungrouped) AVP Header AVP code = 299 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = NO_INBAND_SECURITY **Inband_Security AVP** (Ungrouped) AVP Header AVP code = 299 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = TLS
Steps	1. Send CER (capability Exchange Request Message) to a peer. 2. Receive the CEA (capability Exchange Answer Message) from the peer and check it for expected results.

Expected Results	**CEA (Capability Exchange Answer Message)** **CEA Message Header** **Version =** Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message data = Must contain the below given AVPs and all other AVPs defined by the RFC for this message as data **CER Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = DIAMETER_CONTRADICTING_AVP **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Since the AVP is of type Grouped, hence must contain the following AVPs as Data **Inband_Security AVP** (Ungrouped) AVP Header AVP code = 299 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag

	4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = NO_INBAND_SECURITY **Inband_Security AVP** (Ungrouped) AVP Header AVP code = 299 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = TLS
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 31

Test Case Title	To test the peer behavior when it receives an CER message with two mandatory AVPs missing
Test Case ID	00069
Purpose	To test that when a peer receives an CER (capability Exchange Request Message) with two mandatory AVPs (Origin-Host and Origin-Realm) missing. In this case the peer should return a CEA (capability Exchange Answer Message) with E bit set. This message must contain the Result-Code AVP with result code DIAMETER_MISSING_AVP. Also the message must contain the Failed_AVP. Failed_AVP is a grouped AVP, hence must contain copies of the two missing AVPs (Origin-Host and Origin-Realm) with AVP codes and other fields set as expected in the missing AVPs. The value fields of the missing AVPs should be set to minimum length and filled with zeros.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER (Capability Exchange Request Message)** **CER Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Clear

	Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message Data = Must contain all the AVPs defined by the RFC for this message except Origin-Host AVP and Origin-Realm AVP.
Steps	1. Send CER (capability Exchange Request Message) to a peer. 2. Receive the CEA (capability Exchange Answer Message) from the peer and check it for expected results
Expected Results	**CEA (Capability Exchange Answer Message)** **CEA Message Header** **Version =** Message Length = Message R-Bit = Clear Message P-Bit = Clear Message E-Bit = Set Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 257 Message data = Must contain the below given AVPs as Message data. **CER Message AVPs** **Result_Code AVP (Ungrouped)** AVP Header AVP code = 268 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Since the AVP is of type Grouped, hence must contain the following AVPs as Data **Failed_AVP AVP (Grouped)** AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty

	AVP data = Since the AVP is of type Grouped, hence must contain the following AVPs as Data **Origin-Host AVP** AVP Header AVP code = 264 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Filled With ZEROS **Origin-Realm AVP** AVP Header AVP code = 296 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Filled With ZEROS
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 32

Test Case Title	When a diameter message is received with two AVPs having invalid value in their data fields
Test Case ID	00071
Purpose	To test that the peer should return a message having E-Bit set with the same command code as received .The message should contain a Result_Code AVP with error code of DIAMETER_INVALID_AVP_VALUES. Also Failed_AVP AVP containing the offending AVPs must be included in the

	message
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ACR (Accounting Request Message)** **ACR Message Header** Version = Message Length = Message R-Bit = Set Message P-Bit = Clear Message E-Bit = Clear Message T-Bit = Clear Message 5th,6th, 7th,8th = All Clear Message Command Code = 271 Message Data = Must contain the below given AVPs As message data and other AVPs defined by RFC for the Message **ACR (Accounting Request Message AVPs)** **Origin-Host AVP** AVP Header AVP code = 264 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = 1234567 **Origin-Realm AVP** AVP Header AVP code = 296 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = 1234567
Steps	1. Send ACR (Accounting- Request Message) to a peer. 2. Receive ACA (Accounting- Answer Message) from the peer and check it for expected results
Expected Results	**ACA (Accounting Answer Message)** **ACA Message Header** Version = Message Length = Message R-Bit = Clear (Must) Message P-Bit = Clear (Must)

	Message E-Bit = Set (Must) Message T-Bit = Clear (Optional) Message 5th,6th, 7th,8th = All Clear Message Command Code = Message data = Must contain the below given AVPs as Message data and other AVPs defined by RFC for this message. **ACA Message AVPs** **Result_Code AVP** (Ungrouped) AVP Header AVP code = 268 AVP Flag V-Bit = Clear (Must) AVP flag M-Bit = Set (Must) AVP flag P-Bit = Clear (Optional) AVP flag 4th,5th,6th,7th,8th Bits = Clear (Must) AVP Length = Vendor ID = Default AVP data = DIAMETER_INVALID_AVP_VALUES **Failed_AVP AVP** (Grouped) AVP Header AVP code = 279 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = Since it is an AVP of type grouped , so must contain the below given AVPs **Origin-Host AVP** AVP Header AVP code = 264 AVP Flag V-Bit = Clear AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = 1234567 **Origin-Realm AVP** AVP Header AVP code = 296 AVP Flag V-Bit = Clear

	AVP flag M-Bit = Set AVP flag P-Bit = Clear AVP flag 4th,5th,6th,7th,8th Bits = Clear AVP Length = Vendor ID = Empty AVP data = 1234567
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 33

Test Case Title	Testing Diameter CER Message processing with no common applications supported between the peers.
Test Case ID	00048
Purpose	Testing that Diameter base protocol should return DIAMETER_NO_COMMON_APPLICATION when a CER message is received, and there are no common applications supported between the peers.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message** Diameter Header = {Version=1, Message-Length=,Flags=R is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = NO_INBAND_SECURITY] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	When connection is established with Peer. Both peers do not have any applications in common. One peer receives CER from other.
Expected Results	Result-Code AVP set to DIAMETER_NO_COMMON_APPLICATION , and peer disconnect the transport layer connection.

	CEA Message Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=} { Result-Code = DIAMETER_NO_COMMON_APPLICATION } { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 34

Test Case Title	Testing Diameter DPR Message processing
Test Case ID	
Purpose	Testing that Diameter base protocol should terminate transport connection when receive DPA message
Scenario	Message Validation
Pre-requisite	Diameter Server Should be in receiving state
Input Data	**DPR Message** Diameter Header={Version=1, Message-Length=,Flags=R is set, Command-Code=282, App-ID=0, Hop-by-Hop Id=, End-to-End Id=} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Disconnect-Cause = REBOOTING}
Steps	When Diameter node wants to inform its peer of its intent to disconnect the transport layer. A DPR message is sent to the peer.
Expected Results	DPR sender's peer must reply DPA with following values: Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=282,App-ID=0,Hop-by-Hop Id=, End-to-End Id=} { Result-Code = DIAMETER_SUCCESS } { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"}

	Receiver of DPA should terminate the transport connection.
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 35

Test Case Title	Testing Diameter capabilities negotiation (CER/CEA)
Test Case ID	00057
Purpose	Purpose of this test case is to verify that When two Diameter peers establish a transport connection, they successfully exchange the Capabilities Exchange messages. And also to verify that they successfully discover peer's identity and its capabilities (protocol version number, supported Diameter applications, security mechanisms, etc.)
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message** Diameter Header = {Version=1, Message-Length=,Flags=R is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = NO_INBAND_SECURITY] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	When connection is established with Peer. Both peers do not have any applications in common. One peer receives CER from other.
Expected Results	Result-Code AVP set to = DIAMETER_SUCCESS, and peer's capabilities record is also updated. **CEA Message** Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=}

	{ Result-Code = DIAMETER_SUCCESS} { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Acct-Application-Id = "1200"]
Post Condition	Diameter Server must be in message receiving state

Test Case # 36

Test Case Title	Testing Diameter Message if the route traversed by the request is unacceptable.
Test Case ID	00055
Purpose	Testing that Diameter base protocol should return DIAMETER_AUTHORIZATION_REJECTED if the route traversed by the request is unacceptable.
Scenario	
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"}

	{ Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230] [Route-Record = agent1.advancedims.com] [Route-Record = agent2.advancedims.com] [Route-Record = agent3.advancedims.com]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_AUTHORIZATION_REJECTED, Because route agent2.advancedims.com traversed by request is unacceptable. **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_AUTHORIZATION_REJECTED } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 37

Test Case Title	Testing Diameter Message processing when relay/proxy cannot found upstream server for given application
Test Case ID	00056
Purpose	Testing that Diameter base protocol should return DIAMETER_UNABLE_TO_DELIVER when relay/proxy agents cannot found upstream server that supports the application of a particular message.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-

	to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_UNABLE_TO_DELIVER, and message is returned. **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_UNABLE_TO_DELIVER } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 38

Test Case Title	Testing Diameter Message processing with an AVP having an invalid value in its data portion.
Test Case ID	00045
Purpose	Testing that Diameter base protocol should return DIAMETER_INVALID_AVP_VALUE when an AVP is received having an invalid value in its data portion.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"}

	{ Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Auth-Application-Id = "ABC1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_INVALID_AVP_VALUE, and message is returned. Also include Auth-Application-Id in **`Failed-AVP.`** **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_INVALID_AVP_VALUE } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { User-Name = "USER11"} [Origin-State-Id = 112230] { **`Failed-AVP}`** { Auth-Application-Id = "ABC1200"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 39

Test Case Title	Testing Diameter Message processing with an AVP included more than allowed
Test Case ID	00047
Purpose	Testing that Diameter base protocol should return DIAMETER_AVP_OCCURS_TOO_MANY_TIMES when message was received that included an AVP that appeared more often than permitted in the message definition.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= }

	{ Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES, and message is returned. Also include Origin-Realm in **Failed-AVP.** **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_AVP_OCCURS_TOO_MANY_TIMES } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { User-Name = "USER11"} [Origin-State-Id = 112230] { **Failed-AVP}** { Origin-Realm = "ims.advancedims.org"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 40

Test Case Title	Testing Diameter Message processing with an invalid message length.
Test Case ID	00052
Purpose	Testing that Diameter base protocol should return DIAMETER_INVALID_MESSAGE_LENGTH when length of received message is invalid.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-

	to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Auth-Application-Id = "1200"} (AVP length is set to 2 bytes in AVP header) { User-Name = "USER11"} [Origin-State-Id = 112230]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_INVALID_MESSAGE_LENGTH, and message is returned. Also include Auth-Application-Id in **`Failed-AVP.`** **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_INVALID_MESSAGE_LENGTH} { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { User-Name = "USER11"} [Origin-State-Id = 112230] { **`Failed-AVP}`** **{ Auth-Application-Id = "1200"}**
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 41

Test Case Title	Testing Diameter Message processing with AVP having an invalid length.
Test Case ID	00051
Purpose	Testing that Diameter base protocol should return DIAMETER_INVALID_AVP_LENGTH when an AVP with an invalid length is received.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message**

	Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Auth-Application-Id = "1200"} (AVP length is set to 2 bytes in AVP header) { User-Name = "USER11"} [Origin-State-Id = 112230]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_INVALID_AVP_LENGTH, and message is returned. Also include Auth-Application-Id in **`Failed-AVP.`** **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_INVALID_AVP_LENGTH } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { User-Name = "USER11"} [Origin-State-Id = 112230] { **`Failed-AVP}`** { **Auth-Application-Id** = "1200"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 42

Test Case Title	Testing Diameter Message processing with CER was received from an unknown peer
Test Case ID	00042
Purpose	Testing that Diameter base protocol should silently discard the request and peer disconnects the transport layer connection.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	CER Message

	Diameter Header = {Version=1, Message-Length=,Flags=R is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.advancedims.com"} { Origin-Realm = "advancedims.com"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = NO_INBAND_SECURITY] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	1. Receive CER from unknown peer. 2. Disconnect connection with peer.
Expected Results	Silently discard the request and peer disconnects the transport layer connection.
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 43

Test Case Title	Testing Diameter Message processing with Invalid AVP's Flag bit is set
Test Case ID	00053
Purpose	Testing that Diameter base protocol should return DIAMETER_INVALID_AVP_BIT_COMBO when the request contained an AVP with which is not allowed to have the given value in the AVP Flags field.
Scenario	
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message** Diameter Header = {Version=1, Message-Length=,Flags=R is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.advancedims.com"} { Origin-Realm = "advancedims.com"}

	{ Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} (Mandatory Bit is set) [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = NO_INBAND_SECURITY] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	1. Receive CER from peer. 2. Disconnect connection with peer.
Expected Results	Result-Code AVP set to = DIAMETER_INVALID_AVP_BIT_COMBO. **CEA Message** Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=} { Result-Code = DIAMETER_INVALID_AVP_BIT_COMBO} { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} {Failed-AVP} { Product-Name = "AdvancedIMS Diameter"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 44

Test Case Title	Testing Diameter Message processing with Invalid header bits
Test Case ID	00041
Purpose	Testing that Diameter base protocol should return DIAMETER_INVALID_HDR_BITS when request was received whose bits in the Diameter header were either set to an invalid combination, or to a value that is inconsistent with the command code's definition.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message**

	Diameter Header = {Version=1, Message-Length=,Flags=RP is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = NO_INBAND_SECURITY] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	When connection is established with Peer, One peer receives CER from other.
Expected Results	Result-Code AVP set to = DIAMETER_INVALID_HDR_BITS. Proxy bit in CER message is set which is not allowed. **CEA Message** Diameter Header={Version=1, Message-Length=,Flags=E,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=} { Result-Code = DIAMETER_INVALID_HDR_BITS } { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 45

Test Case Title	Testing Diameter Message processing with missing AVP
Test Case ID	00046
Purpose	Testing that Diameter base protocol should return DIAMETER_MISSING_AVP when the request did not contain an AVP that is required by the Command Code definition.
Scenario	Message Validation

Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message** Diameter Header = {Version=1, Message-Length=,Flags=R is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = TLS] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	When connection is established with Peer. One peer receives CER from other.
Expected Results	Result-Code AVP set to DIAMETER_MISSING_AVP, and peer disconnect the transport layer connection. **CEA Message** Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=} { Result-Code = DIAMETER_MISSING_AVP } { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 46

Test Case Title	Testing Diameter Message processing with no common security mechanisms b/w peers
Test Case ID	00054
Purpose	Testing that Diameter base protocol should return

	DIAMETER_NO_COMMON_SECURITY when a CER message is received, and there are no common security mechanisms supported between the peers.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	
Steps	
Expected Results	
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 47

Test Case Title	Testing Diameter Message processing with unrecognized bit in header is set
Test Case ID	00050
Purpose	Testing that Diameter base protocol should return DIAMETER_INVALID_BIT_IN_HEADER when an unrecognized bit in the Diameter header is set to one.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message** Diameter Header = {Version=1, Message-Length=,Flags=R is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = TLS] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	When connection is established with Peer. Both peers do not have any security mechanism in common. One peer receives CER from other.
Expected Results	Result-Code AVP set to DIAMETER_NO_COMMON_SECURITY, and peer disconnect the transport layer connection. **CEA Message**

	Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=} { Result-Code = DIAMETER_NO_COMMON_SECURITY } { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 48

Test Case Title	Testing Diameter Message processing with unrecognized realm
Test Case ID	00043
Purpose	Testing that Diameter base protocol should return DIAMETER_REALM_NOT_SERVED when intended realm of the request is not recognized.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**CER Message** Diameter Header = {Version=1, Message-Length=,Flags=R & 7tth bit is set, Command-Code=257, App-ID=0, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800c9"} <---(IP version 4: 192.168.0.201) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"} [Origin-State-Id = 112230] [Supported-Vendor-Id = "22011"] [Auth-Application-Id = "167772151"] [Inband-Security-Id = TLS] [Acct-Application-Id = "1200"] [Firmware-Revision = "1"]
Steps	When connection is established with Peer. And unrecognized bit is set in diameter header. One peer receives CER from other
Expected Results	Result-Code AVP set to DIAMETER_INVALID_BIT_IN_HEADER, and peer disconnect the transport layer connection.

	CEA Message Diameter Header={Version=1, Message-Length=,Flags=,Command-Code=257,App-ID=0,Hop-by-Hop Id=,End-to-End Id=} { Result-Code = DIAMETER_INVALID_BIT_IN_HEADER } { Origin-Host = "dbprotocol.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Host-IP-Address = "0x0001c0a800d2"} <---(IP version 4: 192.168.0.210) { Vendor-Id = "0"} { Product-Name = "AdvancedIMS Diameter"}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 49

Test Case Title	Testing Diameter Message processing with unsupported application-Id
Test Case ID	00040
Purpose	Testing that Diameter base protocol should return DIAMETER_APPLICATION_UNSUPPORTED when unsupported application-Id is sent.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Destination-Realm = "advancedims.com"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_APPLICATION_UNSUPPORTED, and message is returned. **ASA Message**

	Diameter Header = {Version=1, Message-Length=,Flags=EP is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_APPLICATION_UNSUPPORTED } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 50

Test Case Title	Testing Diameter Message processing with unsupported AVP that is marked with the Mandatory
Test Case ID	00044
Purpose	Testing that Diameter base protocol should return DIAMETER_AVP_UNSUPPORTED when received unsupported AVP that is marked with the Mandatory bit.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	**ASR Message** Diameter Header = {Version=1, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Destination-Realm = "advancedims.com"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230] {Unknown-AVP = 111} (Mandatory nit is set) { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230] {Unknown-AVP = 111} (Mandatory nit is set)
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for expected results
Expected Results	Result-Code AVP set to DIAMETER_AVP_UNSUPPORTED,

	and message is returned. ASA Message Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_AVP_UNSUPPORTED } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230] { Failed-AVP} { Unknown-AVP = 111}
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 52

Test Case Title	Testing Diameter Message processing with unsupported version number
Test Case ID	00049
Purpose	Testing that Diameter base protocol should return DIAMETER_UNSUPPORTED_VERSION when a request was received, whose version number is unsupported.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	

ASR Message

Diameter Header = {Version=3, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= }
{ Session-Id = "USER11-2200"}
{ Origin-Host = "nas.ims.advancedims.org"}
{ Origin-Realm = "ims.advancedims.org"}
{ Destination-Host = "hms.ims.advancedims.org"}
{ Destination-Realm = "advancedims.com"}
{ Auth-Application-Id = "1200"}
{ User-Name = "USER11"}
[Origin-State-Id = 112230]

{Unknown-AVP = 111} (Mandatory nit is set)	
Steps	1. Send ASR to a peer. 2. Receive ASA from the peer and check it for results
Expected Results	Result-Code AVP set

	DIAMETER_UNSUPPORTED_VERSION , and returned. **ASA Message** Diameter Header = {Version=1, Message-Length=,Fl Command-Code=274, App-ID=1200, Hop-by-Hop Id End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_UNSUPPORTED } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Post Condition	The Diameter Server must be in a state to receive me

Test Case # 52

Test Case Title	Testing Diameter Message processing with unsupported version number
Test Case ID	00049
Purpose	Testing that Diameter base protocol should return DIAMETER_UNSUPPORTED_VERSION when a request was received, whose version number is unsupported.
Scenario	Message Validation
Pre-requisite	Diameter Server must be in message receiving state
Input Data	

ASR Message

Diameter Header = {Version=3, Message-Length=,Flags=R,P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= }
{ Session-Id = "USER11-2200"}
{ Origin-Host = "nas.ims.advancedims.org"}
{ Origin-Realm = "ims.advancedims.org"}
{ Destination-Host = "hms.ims.advancedims.org"}
{ Destination-Realm = "advancedims.com"}
{ Auth-Application-Id = "1200"}
{ User-Name = "USER11"}
[Origin-State-Id = 112230]

{Unknown-AVP = 111} (Mandatory nit is set)	
Steps	3. Send ASR to a peer. 4. Receive ASA from the peer and check it for expected

	results
Expected Results	Result-Code AVP set to DIAMETER_UNSUPPORTED_VERSION , and message is returned. **ASA Message** Diameter Header = {Version=1, Message-Length=,Flags=P is set, Command-Code=274, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Session-Id = "USER11-2200"} { Result-Code = DIAMETER_UNSUPPORTED_VERSION } { Origin-Host = "proxy.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Auth-Application-Id = "1200"} { User-Name = "USER11"} [Origin-State-Id = 112230]
Post Condition	The Diameter Server must be in a state to receive message

Test Case # 53

Test Case Title	Testing Diameter Message processing with Invalid Command code
Test Case ID	00039
Purpose	Testing that Diameter base protocol should return DIAMETER_COMMAND_UNSUPPORTED when wrong command is sent.
Scenario	Protocol Errors
Pre-requisite	Diameter Server Should be in receiving state.
Input Data	Diameter Server receive a command which is not supported. Diameter Header = {Version=3, Message-Length=,Flags=R,P is set, Command-Code=970, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Destination-Realm = "advancedims.com"}
Steps	1. Send message to a peer. 2. Receive Answer from the peer and check it for expected results
Expected Results	Diameter Server receive a command which is not supported. Diameter base protocol should return

	DIAMETER_COMMAND_UNSUPPORTED. Diameter Header = {Version=3, Message-Length=,Flags=is set, Command-Code=970, App-ID=1200, Hop-by-Hop Id=, End-to-End Id= } {Result-Code = DIAMETER_COMMAND_UNSUPPORTED} { Origin-Host = "nas.ims.advancedims.org"} { Origin-Realm = "ims.advancedims.org"} { Destination-Host = "hms.ims.advancedims.org"} { Destination-Realm = "advancedims.com"}
Post Condition	Diameter Server Should be in message receiving state

4.3 Peer Table Function Validation Test Cases

Test Case # 54

Test Case Title	To test Peer Table Creation SUCCESS case
Test Case ID	00084
Purpose	To check the CreatePeerTable() API behavior by for different Inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTable_Handle containing NULL value
Steps	1. Call DMBase_CreatePeerTable() API 2. Check the output of DMBase_CreatePeerTable()
Expected Results	PeerTable created as the PeerTable_Handle is not NULL now
Post Condition	Peertable created

Test Case # 55

Test Case Title	To test Peer Table Creation Failure case
Test Case ID	0117
Purpose	To check the CreatePeerTable() API behavior by for different Inputs
Scenario	Function Validation
Pre-requisite	Server must be in a state to call the validating function

Input Data	PeerTable_Handle not equal to NULL value
Steps	1. Call DMBase_CreatePeerTable() API 2. Check the output of DMBase_CreatePeerTable()
Expected Results	PeerTable cannot be created , as PeerTable allready exist
Post Condition	PeerTable not created

Test Case # 56

Test Case Title	To test Peer Table Entry Creation SUCCESS case
Test Case ID	00086
Purpose	To check the behavior of DMBase_CreatePeerTableEntry() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTableEntry_Handle containing NULL value
Steps	1. Call DMBase_CreatePeerTableEntry() API 2. Check it for the expected results
Expected Results	Success: PeerTableEntry Created as PeerTableEntry_Handle in not NULL now
Post Condition	PeerTable Entry Created

Test Case # 57

Test Case Title	To test Peer Table Entry Creation FAILURE case
Test Case ID	0118
Purpose	To check the behavior of DMBase_CreatePeerTableEntry() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTableEntry_Handle not containing NULL value
Steps	3. Call DMBase_CreatePeerTableEntry() API 4. Check it for the expected results
Expected Results	FAILURE: PeerTableEntry all ready exists cannot be Created
Post Condition	New PeerTable Entry not Created

Test Case # 58

Test Case Title	To test Insertion in Peer Table SUCCESS case
Test Case ID	00088
Purpose	To check the DMBase_InsertIntoPeerTable() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	Empty PeerTable with created PeerTableEntry **Following Values for insertion** HostIdentity = OriginHost.ims.com Status = connected Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes
Steps	1. Call DMBase_InsertIntoPeerTable() API 2. Check it for the expected results
Expected Results	**Success :** Values successfully inserted in PeerTable
Post Condition	PeerTable must contain newly inserted values

Test Case # 59

Test Case Title	To test Insertion in Peer Table Failure
Test Case ID	0119
Purpose	To check the DMBase_InsertIntoPeerTable() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	**PeerTable** having following values HostIdentity = OriginHost.ims.com Status = connected

	Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes **Following Values for insertion** Status = connected Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes
Steps	1. Call DMBase_InsertIntoPeerTable() API 2. Check it for the expected results
Expected Results	**Failure :** HostIdentity missing, Entry cannot be inserted in PeerTable
Post Condition	PeerTable Without Newly inserted values

Test Case # 60

Test Case Title	To test Deletion of Peer Table Entry Success
Test Case ID	00092
Purpose	Function Validation
Scenario	To check the behavior of DMBase_DeletePeerTableEntry() API for different inputs
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTable having entry with following values HostIdentity = OriginHost.ims.com Status = connected Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes **Host Identity** = OriginHost.ims.com
Steps	1. Call DMBase_DeletePeerTableEntry() API 2. Check it for expected results
Expected Results	**Success : PeerTableEntry** successfully deleted
Post Condition	PeerTable entry Deleted

Test Case # 61

Test Case Title	Deletion of Peer Table Entry Failure case

Test Case ID	0120
Purpose	To check the behavior of DMBase_DeletePeerTableEntry() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTable having entry with following values HostIdentity = OriginHost.ims.com Status = connected Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes Host Identity = DestinationHost.ims.com
Steps	1. Call DMBase_DeletePeerTableEntry() API 2. Check it for expected results
Expected Results	**Failure :** PeerTableEntry cannot found for deletion
Post Condition	PeerTable with one entry Deleted

Test Case # 62

Test Case Title	Deletion of Peer Table Entry when PeerTable is Empty
Test Case ID	0121
Purpose	To check the behavior of DMBase_DeletePeerTableEntry() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	Empty PeerTable Host Identity = DestinationHost.ims.com
Steps	1. Call DMBase_DeletePeerTableEntry() API 2. Check it for expected results
Expected Results	**Failure :** PeerTable is emtpy
Post Condition	No entry deleted from peerTable

Test Case # 63

Test Case Title	To test Lookup Peer Table Entry SUCCESS case
Test Case ID	00090
Purpose	To check the behavior of DMBase_LooKUpPeerTable() API for

	different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTable having an entry with following values HostIdentity = OriginHost.ims.com Status = connected Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes HostIdentity = OriginHost.ims.com
Steps	1. Call DMBase_LookUpPeerTable() API 2. Check it for expected results
Expected Results	**Success :** Entry found in PeerTable
Post Condition	PeerTable Entry found

Test Case # 64

Test Case Title	Lookup Peer Table Entry FAILURE case
Test Case ID	0122
Purpose	To check the behavior of DMBase_LooKUpPeerTable() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	PeerTable having an entry with following values HostIdentity = OriginHost.ims.com Status = connected Static/Dynamic = Static Expiration Time = 1 hr TLS Enabled = Yes HostIdentity = DestinationHost.ims.com
Steps	1. Call DMBase_LookUpPeerTable() API 2. Check it for expected results
Expected Results	**Failure :** No entry found in PeerTable
Post Condition	PeerTable Entry does not found

4.4 Realm Table Function Validation Test Cases

Test Case # 65

Test Case Title	To test Realm Table Creation SUCCESS case
Test Case ID	00085
Purpose	To check the CreateRealmTable() API behavior by for different Inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTable_Handle containing NULL value
Steps	1. Call DMBase_CreateRealmTable() API 2. Check the output of DMBase_CreateRealmTable()
Expected Results	**Success:** RealmTable created as the RealmTable_Handle is not NULL now
Post Condition	Control Should return to the Caller function

Test Case # 66

Test Case Title	Realm Table Creation FAILURE case
Test Case ID	0123
Purpose	To check the CreateRealmTable() API behavior by for different Inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTable_Handle not equal to NULL value
Steps	1. Call DMBase_CreateRealmTable() API 2. Check the output of DMBase_CreateRealmTable()
Expected Results	RealmTable cannot be created , as RealmTable allready exist
Post Condition	Control Should return to the Caller function

Test Case # 67

Test Case Title	Realm Table Entry Creation when entry all ready exist
Test Case ID	00087
Purpose	To check DMBase_CreatePeerTableEntry() API for different inputs

Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTableEntry_Handle not equal to NULL
Steps	1. Call DMBase_CreateRealmTableEntry() API 2. Check it for the expected results
Expected Results	**Failure:** PeerTableEntry cannot be Created " RealmTableEntry All ready exists "
Post Condition	Control Should return to the Caller function

Test Case # 68

Test Case Title	To test Realm Table Entry Creation SUCCESS case
Test Case ID	0124
Purpose	To check DMBase_CreatePeerTableEntry() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTableEntry_Handle containing NULL value
Steps	1. Call DMBase_CreateRealmTableEntry() API 2. Check it for the expected results
Expected Results	**Success:** PeerTableEntry Created as RealmTableEntry_Handle in not NULL now
Post Condition	Control Should return to the Caller function

Test Case # 69

Test Case Title	Insertion in Realm Table SUCCESS case
Test Case ID	00089
Purpose	To check the behavior of DMBase_InsertInToTearlmTable() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validator function
Input Data	Empty RealmTable with created Entry Following values for insertion in RealmTable Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo

	Static/Dynamic= Static ExpirationTime = 1 hr
Steps	1. Call DMBase_InsertIntoRealmTable() API 2. Check it for the expected results
Expected Results	**Success :** Values successfully inserted in RealmTable
Post Condition	Control Should return to the Caller function

Test Case # 70

Test Case Title	To test Insertion in Realm Table FAILURE case
Test Case ID	0125
Purpose	To check the behavior of DMBase_InsertInToTearlmTable() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTable with created Entry and having following Entry as well Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr Following values for insertion in RealmTable Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr
Steps	1. Call DMBase_InsertIntoRealmTable() API 2. Check it for the expected results
Expected Results	**Failure :** Entry Allready exists n RealmTable, Entry cannot be inserted
Post Condition	Control Should return to the Caller function

Test Case # 71

Test Case Title	To test Insertion in Realm Table when the value of one field is missing
Test Case ID	0126
Purpose	To check the behavior of DMBase_InsertInToTearlmTable() API for different inputs
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	**RealmTable** with created **Entry** and having following **Entry** as well Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr Following values for insertion in **RealmTable** Realm Name = DestinationRealm.ims.com Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr
Steps	1. Call DMBase_InsertIntoRealmTable() API 2. Check it for the expected results
Expected Results	**Failure :** Application ID missing, Entry cannot be inserted
Post Condition	Control Should return to the Caller function

Test Case # 72

Test Case Title	Lookup Realm Table Entry SUCCESS case
Test Case ID	00091
Purpose	To check the behavior of DMBase_LookUpRealmTableEntry() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function

Input Data	RealmTable with Entry having following values Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr Hash_Table_Key = DestinationRealm.ims.com123456
Steps	1. Call DMBase_LooKUpRealmTable() API 2. Check it for expected results
Expected Results	**Success :** RealmTable Entry Found
Post Condition	Control Should return to the Caller function

Test Case # 73

Test Case Title	Lookup Realm Table Entry FAILURE case
Test Case ID	0127
Purpose	To check the behavior of DMBase_LookUpRealmTableEntry() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTable with Entry having following values Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr Hash_Table_Key = DestinationRealmOne.ims.com123456
Steps	1. Call DMBase_LooKUpRealmTable() API 2. Check it for expected results
Expected Results	**Failure :** RealmTable Entry not Found
Post Condition	Control Should return to the Caller function

Test Case # 74

Test Case Title	Lookup Realm Table Entry when Realm Table is empty
Test Case ID	0128
Purpose	To check the behavior of DMBase_LookUpRealmTableEntry() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	Empty RealmTable **Hash_Table_Key =** DestinationRealmOne.ims.com123456
Steps	1. Call DMBase_LooKUpRealmTable() API 2. Check it for expected results
Expected Results	**Failure :** RealmTable is emtpy
Post Condition	Control Should return to the Caller function

Test Case # 75

Test Case Title	Deletion of Realm Table Entry SUCCESS case
Test Case ID	00093
Purpose	To check the behavior of DMBase_RealmTableEntry() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	**RealmTable** with Entry having following values Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr **Hash_Table_Key =** DestinationRealm.ims.com123456
Steps	1. Call DMBase_DeleteRealmTableEntry() API 2. Check it for expected results
Expected Results	**Success :** RealmTable Entry Deleted
Post Condition	Control Should return to the Caller function

Test Case # 76

Test Case Title	Deletion of Realm Table Entry FAILURE case
Test Case ID	0129
Purpose	To check the behavior of DMBase_RealmTableEntry() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	RealmTable with Entry having following values Realm Name = DestinationRealm.ims.com Application ID = 123456 Local Action = Proxy Sewever Identifier = ServerOne, ServerTwo Static/Dynamic= Static ExpirationTime = 1 hr Hash_Table_Key = DestinationRealmOne.ims.com123456
Steps	1. Call DMBase_DeleteRealmTableEntry() API 2. Check it for the expected results
Expected Results	**Failure :** RealmTable Entry not Found for deletion
Post Condition	Control Should return to the Caller function

Test Case # 77

Test Case Title	Deletion of Realm Table Entry when Realm Table is Empty
Test Case ID	0130
Purpose	To check the behavior of DMBase_RealmTableEntry() API for different input values
Scenario	Function Validation
Pre-requisite	Server Should be in a state to call the validating function
Input Data	Empty RealmTable **Hash_Table_Key =** DestinationRealm.ims.com123456
Steps	1. Call DMBase_DeleteRealmTableEntry() API 2. Check it for expected results
Expected Results	**Failure :** RealmTable is empty
Post Condition	Control Should return to the Caller function

5. Hardware and Software requirements for testing

Hardware Requirements	
CPU	2.16 GHz
RAM	64 MB
Disk Storage	20 GB

5.1 Software Requirements for testing

Operating System
Windows 2000
Windows 2003 Server
Windows XP
Linux

Development Dependencies

Development Dependencies Software
AMPS

6. Test Report Form (Sample)

The following form will be used as the test reporting form.

Test Report Form					
Version #:					
Prepared By:					
Sr#	**Test Case Name**	**Input Data**	**Expected Results**	**Actual Result**	**Status**
01					

This form contains information about test result when product is tested.

7. References

[1] Diameter Base Protocol [IETF RFC 3588]