



## **Development Of Diameter Protocol Stack For Client**

**Sowmya. A. V**

Dayananda Sagar College of Engineering, Bangalore, India

**Nagarathna**

Dayananda Sagar College of Engineering, Bangalore, India

**Ajay Satyanarayana**

TATA ELXSI LTD, Bangalore, India

***Abstract:***

*The DIAMETER protocol is designed to provide an Authentication, Authorization and Accounting (AAA) framework for applications such as network access or IP mobility and also intended to work in local Authentication, Authorization & Accounting and roaming situations.*

*The Diameter protocol was initially developed by the Internet Engineering Task Force (IETF) as an Authentication, Authorization, and Accounting (AAA) framework Diameter was further embraced by the Third Generation Partnership Project (3GPP) as the key protocol for AAA and mobility management in 3G networks.*

*The DIAMETER protocol consists of a base protocol, defined in [RFC 3588] September 2003 and it is obsolete by [RFC 6733] October 2012, and set of applications (also called extension of the base protocol). The base protocol must be supported by all applications. The base protocol must be supported by all applications.*

***Key words:*** DIAMETER, RADIUS, AAA, IMS, SIP

## 1.Introduction

DIAMETER is the newest AAA protocol developed in 2001 from the older AAA protocol RADIUS (Remote Access Dial-In User Service) by the Internet Engineering Task Force (IETF) as an Authentication, Authorization, and Accounting (AAA) framework intended for applications such as remote network access and IP mobility.

The DIAMETER protocol was developed to resolve the issues that RADIUS left open. In new application areas like Wireless Local Access Network (WLAN) and Voice over IP (VoIP), DIAMETER is better suited and gives better support for roaming users.

DIAMETER is defined as a base protocol used in conjunction with a set of applications. The Diameter Base Protocol [RFC 6733] contains the basic functionality like reliable transport, message delivery and error handling. The Diameter applications are extensions of the basic functionalities that are tailored for a particular usage in a particular environment.

AAA Framework includes:

- Authentication is the verification of the identity of the entity.
- Authorization is the determination whether the requesting entity is allowed access to a particular resource.
- Accounting is the collecting of information about resource usage for the purpose of capacity planning, auditing, billing or cost allocation.

## 2.IP Multimedia Subsystem

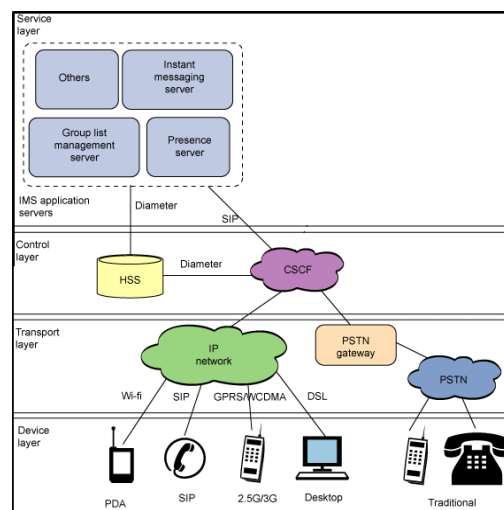


Figure 1: architecture of IMS

IP Multimedia Subsystem or IMS is a standardized Next Generation Networking (NGN) architecture for telecom operators that want to provide mobile and fixed multimedia services. It uses a Voice-over-IP (VoIP) implementation based on a 3GPP standardized implementation of Session Initiation Protocol (SIP), and runs over the standard Internet Protocol (IP). Existing phone systems (both packet-switched and circuit-switched) are supported.

The IMS is the technology that will merge the Internet with the Cellular world. It is network enables and drives efficient converged service offerings. It is the key to delivering multimedia services with telecom-grade quality of service across fixed and mobile accesses. It creates new opportunities for operators who want to deliver attractive, easy-to-use, reliable and profitable multimedia services – including voice, pictures, text and video, or any combination of these – with existing services such as web, email, instant messaging, and videoconferencing available nearly everywhere.

The IMS is based on a horizontally layered architecture, consisting of three layers, namely, Service Layer, Control Layer, and transport Layer.

(i) Service Layer comprises application and content servers to execute value-added services for the user.

(ii)Control layer comprises network control servers for managing call or session set-up, modification and release. The most important of these is the Call Session Control Function (CSCF).

(iii)Connectivity Layer comprises of routers and switches, for both the backbone and the access network

Device layer consists of user terminal devices like UE (user equipment), PDA, Desktop, and Laptop which can connect to the IMS network.

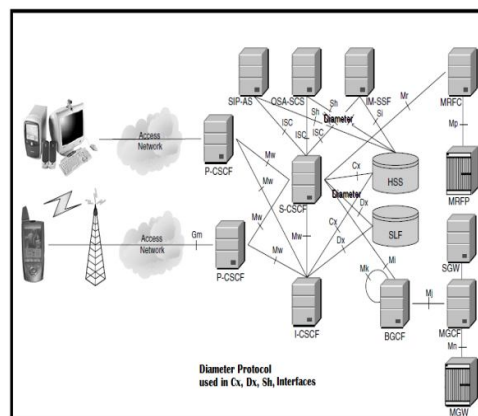


Figure 2: Reference point in IMS

The reference points namely, Cx, Dx, Sh uses DIAMETER protocol.

Cx lies between I-CSCF and HSS, S-CSCF and HSS.

Dx lies between I-CSCF and SLF, S-CSCF and SLF.

Sh lies between HSS and SIP-AS; HSS and OSA-SCS.

### 3. Diameter Protocol Stack Architecture

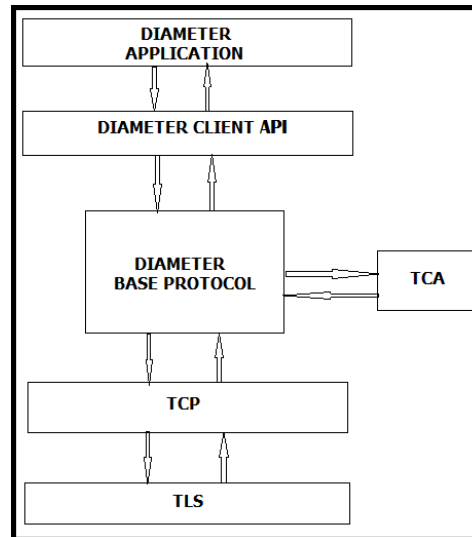


Figure 3: Architecture of Diameter Protocol Stack

The above shows the schematic of DIAMETER protocol stack architecture for application server in IP Multimedia subsystem (IMS).

- Diameter base protocol: The Diameter base protocol provides basic services to one or more applications (also called functions) that each runs in a different Diameter instance.
- Basic functionalities provided by base protocol are as follows: reliable transport, message delivery, Delivery of AVPs, Capability Negotiation, Error Notification, Accounting, and extensibility via new command codes and AVPs.
- Diameter Client API (Application programming interface): This acts as the interface between the diameter base protocol and the diameter application.
- Diameter application: Diameter application is an application specific protocol used for transfer of application specific functions and messages.

- Diameter applications are namely as follows: NASREQ, EAP, Mobile IPV4, Credit Control and other 3GPP interfaces like Cx, Dx, Sh, Ro, Rf etc

#### 4. Implementation Of Diameter Protocol Stack

Implementation begins with the sequence of development of the diameter base protocol and as well as the corresponding required diameter application protocol. API is another important development which acts as interface between the diameter base and its corresponding diameter application protocol.

DIAMETER is application layer protocol comes under TCP/IP protocol suite and it runs over reliable transport protocols, TCP (Transmission Control Protocol) and TLS (Transport Layer Security).

#### 5. Flow Diagram

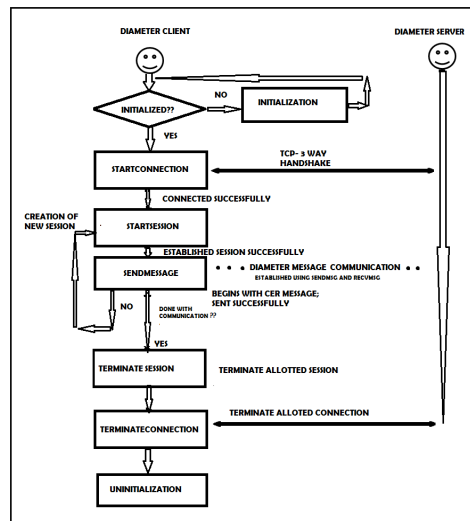


Figure 4: API Flow of Diameter Client

#### 6. Diameter Client API

Diameter Client API is used to perform the vital operations carried out in Diameter Core functionalities and acts as an interface for the Diameter Base Protocol and Diameter Application.

### *6.1.Initialization API*

Initialization API used to initialize the DIAMETER Client stack which performs in the registration of the stack with TCA, allocate the memory for the stack resources and configuration of the stack.

### *6.2.Start Connection API*

StartConnection API is used for the connection establishment between DIAMETER client and DIAMETER server which deals with the creation of transmission sockets, timers.

### *6.3.Startsession API*

StartSession API is used for the session establishment between the connected DIAMETER client and DIAMETER server which deals with the creation sessions and filling details.

### *6.4.Sendmessage API*

SendMessage API is used to send the Diameter message packet in encoded form to another Diameter Node (Diameter Server) in allotted Connection\_ID and Session\_ID; by creating new transaction using Transaction\_ID. Flow Diagram of this API is given in Figure 4.

### *6.5.Terminatesession API*

TerminateSession API is used to terminate the session which has been established between the connected DIAMETER client and DIAMETER server which deals with the deletion of sessions and its details.

### *6.6.Terminateconnection API*

TerminateConnection API is used for the terminating the connection established between DIAMETER client with the DIAMETER server which deals with the deletion of transmission sockets, timers.

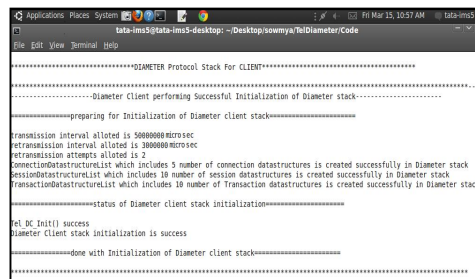
### 6.7.Uninitialization API

UnInitialization API used to uninitialized the DIAMETER Client stack which perform de-registration of the stack with TCA, de-allocate the memory allotted for the stack resources and configuration of the stack.

## 7.Result

Snapshots of all APIs are given as follows:

7.1.Initialization Api Success Initialization API is called with valid parameter value resulting in SUCCESS of Initialization of Stack.



```

Applications Places System
tata-ims5@tata-ims5-desktop: ~/Desktop/soomya/7d1Diameter/Code
File Edit View Terminal Help

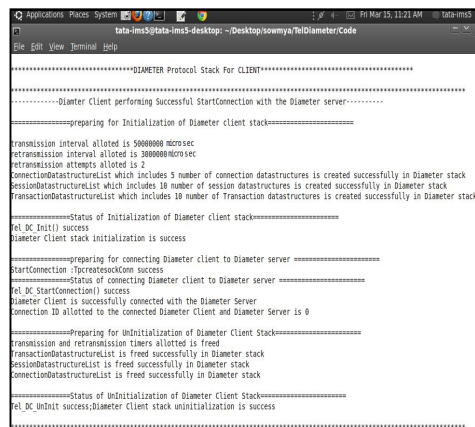
*****DIAMETER Protocol Stack For CLIENT*****
*****Diameter Client performing Successful Initialization of Diameter stack*****
*****preparing for Initialization of Diameter client stack*****
Transmission interval allotted is 5000000microsec
Retransmission interval allotted is 3000000microsec
Retransmission attempts allotted is 2
ConnectionDataStructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDataStructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDataStructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
*****Status of Diameter client stack initialization*****
tel_dc_Init() success
Diameter Client stack initialization is success
*****done with Initialization of Diameter client stack*****

```

Figure 5: Initialization API SUCCESS

### 7.2.Startconnection API SUCCESS

StartConnection API is called resulting in SUCCESS, whose pre-requisite is Initialization API which is SUCCESS; later UnInitialization is called which is SUCCESS.



```

Applications Places System
tata-ims5@tata-ims5-desktop: ~/Desktop/soomya/7d1Diameter/Code
File Edit View Terminal Help

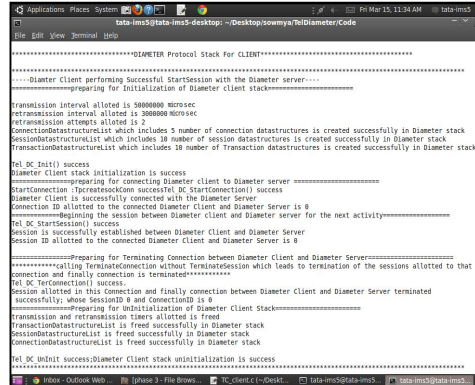
*****DIAMETER Protocol Stack For CLIENT*****
*****Diameter Client performing Successful StartConnection with the Diameter server*****
*****preparing for Initialization of Diameter client stack*****
Transmission interval allotted is 5000000microsec
Retransmission interval allotted is 3000000microsec
Retransmission attempts allotted is 2
ConnectionDataStructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDataStructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDataStructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
*****Status of Initialization of Diameter client stack*****
tel_dc_Init() success
Diameter Client stack initialization is success
*****preparing for connecting Diameter client to Diameter server*****
StartConnection :TpcretesockConn success
*****Status of connecting Diameter client to Diameter server*****
tel_dc_StartConnection() success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the connected Diameter Client and Diameter Server is 0
*****preparing for UnInitialization of Diameter Client Stack*****
Transmission and retransmission timers allotted is freed
TransactionDataStructureList is freed successfully in Diameter stack
SessionDataStructureList is freed successfully in Diameter stack
ConnectionDataStructureList is freed successfully in Diameter stack
*****Status of UnInitialization of Diameter Client Stack*****
tel_dc_UnInit success;Diameter Client stack initialization is success

```

Figure 6: StartConnection API SUCCESS

### 7.3.Startsession API SUCCESS

StartSession API is SUCCESS, and its pre-requisite are Initialization API and StartConnection API is SUCCESS; later UnInitialization is called which is SUCCESS.



```

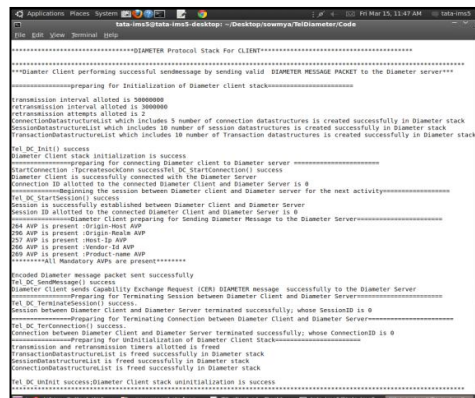
*****DIAMETER Protocol Stack For CLIENT*****
-----Diameter Client performing Successful StartSession with the Diameter server-----
-----preparing for Initialization of Diameter Client stack-----
Transmission interval allotted is 500000000 microsec
Retransmission interval allotted is 300000000 microsec
Retransmission attempts allotted is 2
ConnectionDataStructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDataStructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDataStructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
Tel DC Init() success
Diameter Client stack initialization is success
-----preparing for connecting Diameter client to Diameter server-----
StartConnection ()preconnection successful DC StartConnection() success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the connected Diameter Client and Diameter Server is 0
-----beginning the session between Diameter Client and Diameter server for the next activity-----
Tel DC StartSession() success
Session is successfully established between Diameter Client and Diameter Server
Session ID allotted to the connected Diameter Client and Diameter Server is 0
-----preparing for Terminating Connection between Diameter Client and Diameter Server-----
-----calling TerminateConnection without TerminationDecision which leads to termination of the sessions allotted to that
connection and finally connection is terminated-----
Tel DC TerConnection() success
Session allotted in this connection and finally connection between Diameter Client and Diameter Server terminated
successfully whose SessionID is 0 and ConnectionID is 0
-----preparing for uninitialization of Diameter Client Stack-----
Transmission and retransmission timers allotted is freed
TransactionDataStructureList is freed successfully in Diameter stack
SessionDataStructureList is freed successfully in Diameter stack
ConnectionDataStructureList is freed successfully in Diameter stack
Tel DC UnInit success:Diameter Client stack uninitialization is success

```

Figure 7: StartSession API SUCCESS

### 7.4.SendMessage API SUCCESS

SendMessage API is SUCCESS by sending valid Diameter message parameter, its pre-requisite are Initialization API, StartConnection API and StartSession API are SUCCESS; sending CER message.



```

*****DIAMETER Protocol Stack For CLIENT*****
-----Diameter Client performing successful SendMessage by sending valid DIAMETER MESSAGE PACKET to the Diameter server-----
-----preparing for Initialization of Diameter Client stack-----
Transmission interval allotted is 500000000 microsec
Retransmission interval allotted is 300000000 microsec
Retransmission attempts allotted is 2
ConnectionDataStructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDataStructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDataStructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
Tel DC Init() success
Diameter Client stack initialization is success
-----preparing for connecting Diameter client to Diameter server-----
StartConnection ()preconnection successful DC StartConnection() success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the connected Diameter Client and Diameter Server is 0
-----beginning the session between Diameter Client and Diameter server-----
Tel DC StartSession() success
Session is successfully established between Diameter Client and Diameter Server
Session ID allotted to the connected Diameter Client and Diameter Server is 0
-----Diameter Client preparing for Sending Diameter Message to the Diameter Server-----
CER AP is present -Origin-Host App
CER AP is present -Origin-Realm App
CER AP is present -Vendor-Id App
CER AP is present -Product-Name App
*****All Mandatory APPs are present*****
Recorded Diameter message packet sent successfully
Tel DC SendMessage() success
Diameter Client sends Capability Exchange Request (CER) DIAMETER message successfully to the Diameter Server
Tel DC TerminateConnection() success
-----preparing for Terminating Session between Diameter Client and Diameter Server-----
Session between Diameter Client and Diameter Server terminated successfully, whose SessionID is 0
Tel DC TerConnection() success
-----preparing for Terminating Connection between Diameter Client and Diameter Server-----
Connection between Diameter Client and Diameter Server terminated successfully, whose ConnectionID is 0
Transmission and retransmission timers allotted is freed
TransactionDataStructureList is freed successfully in Diameter stack
SessionDataStructureList is freed successfully in Diameter stack
ConnectionDataStructureList is freed successfully in Diameter stack
Tel DC UnInit success:Diameter Client stack uninitialization is success

```

Figure 8: SendMessage CER SUCCESS



```

*****Diameter Protocol Stack For Client*****
*****Diameter Client performing successful sendmessage by sending Device Watchdog Request (DWR) DIAMETER MESSAGE PACKET to the
Diameter server*****
*****Preparing for Initialization of Diameter client stack*****
Transmission interval allotted is 30000ms
retransmission interval allotted is 30000ms
*****Initialization attempt*****
ConnectionDataStructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDataStructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDataStructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
Hel_DC_Init() success
Diameter Client stack initialization is success
*****Preparing for connecting Diameter client to Diameter server*****
StartConnection() success[Hel_DC_StartConnection()] success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the Connected Diameter Client and Diameter Server is 0
Hel_DC_StartSession() success
*****beginning the session between Diameter Client and Diameter server for the next activity*****
Session ID is successfully established between Diameter Client and Diameter Server
Session ID allotted to the connected Diameter Client and Diameter Server is 0
*****Diameter Client preparing for Sending Diameter Message to the Diameter Server*****
The App ID present: 0x01000000
*****All Mandatory APPs of DWR are present*****
Recorded Diameter message packet sent successfully
Hel_DC_SendMessage() success
Diameter Client sends Device Watchdog Request (DWR) DIAMETER message successfully to the Diameter Server
*****Preparing for Terminating Session between Diameter Client and Diameter Server*****
Hel_DC_TerminateSession() success
*****Terminating Session between Diameter Client and Diameter Server terminated successfully, whose SessionID is 0*****
*****Preparing for Terminating Connection between Diameter Client and Diameter Server*****
Hel_DC_TerminateConn() success
*****Terminating Connection between Diameter Client and Diameter Server terminated successfully, whose ConnectionID is 0*****
*****Preparing for Uninitialization of Diameter Client Stack*****
Transmission and retransmission timers allotted is freed
TransactionDataStructureList is freed successfully in Diameter stack
SessionDataStructureList is freed successfully in Diameter stack
ConnectionDataStructureList is freed successfully in Diameter stack
Hel_DC_UnInit success:Diameter Client stack uninitialization is success
*****Diameter Client stack uninitialization is success*****

```

Figure 9: SendMessage DWR SUCCESS

### 7.5. TerminateSession API SUCCESS

TerminateSession API is called resulting in SUCCESS, and its pre-requisite is Initialization API and StartConnection, StartSession API are SUCCESS; later remaining APIs are called which all are SUCCESS.

```

*****Diameter Client performing Successful TerminateSession which terminates the existing session established with the
Diameter server*****
*****Preparing for Initialization of Diameter client stack*****
Transmission interval allotted is 30000ms
retransmission interval allotted is 30000ms
*****Initialization attempt*****
ConnectionDataStructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDataStructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDataStructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
Hel_DC_Init() success
Diameter Client stack initialization is success
*****Preparing for connecting Diameter client to Diameter server*****
StartConnection() success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the Connected Diameter Client and Diameter Server is 0
Hel_DC_StartSession() success
*****beginning the session between Diameter Client and Diameter server for the next activity*****
Session ID is successfully established between Diameter Client and Diameter Server
Session ID allotted to the connected Diameter Client and Diameter Server is 0
Hel_DC_TerminateSession() success
*****Preparing for Terminating Session between Diameter Client and Diameter Server*****
Session between Diameter Client and Diameter Server terminated successfully, whose SessionID is 0 and ConnectionID is 0
*****Preparing for Terminating Connection between Diameter Client and Diameter Server*****
Hel_DC_TerminateConn() success
*****Terminating Connection between Diameter Client and Diameter Server terminated successfully, whose ConnectionID is 0*****
*****Preparing for Uninitialization of Diameter Client Stack*****
Transmission and retransmission timers allotted is freed
TransactionDataStructureList is freed successfully in Diameter stack
SessionDataStructureList is freed successfully in Diameter stack
ConnectionDataStructureList is freed successfully in Diameter stack
Hel_DC_UnInit success:Diameter Client stack uninitialization is success
*****Diameter Client stack uninitialization is success*****

```

Figure 10: TerminateSession API SUCCESS

### 7.6. Terminateconnection API SUCCESS

TerminateConnection API is called resulting in SUCCESS, and its pre-requisite is Initialization API, StartConnection and/or StartSession API are SUCCESS; later terminate the connection and sessions (if present) by TerminateConnection API and later UnInitialization is called both are SUCCESS.

```

tata-ims3@tata-ims3-desktop: ~/Desktop/soemya/TC/DiameterCode
*****DIAMETER Protocol Stack For CLIENT*****
-----Diameter Client performing Successful TerminateConnection with the Diameter server-----
-----Preparing for Initialization of Diameter client stack-----
Transmission interval allotted is 5000000
Retransmission interval allotted is 300000
Retransmission attempts allotted is 2
ConnectionDatastructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDatastructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDatastructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
Tel_DC_Init() success
Diameter Client stack initialization is success
-----Preparing for connecting Diameter client to Diameter server -----
startConnection (PreestablishConn) success(Tel_DC_startConnection) success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the connected Diameter Client and Diameter Server is 0
-----Preparing for Terminating Connection between Diameter Client and Diameter Server-----
Tel_DC_TerminateConn() success.
Connection between Diameter Client and Diameter Server terminated successfully whose ConnID is 0
-----Preparing for uninitialization of Diameter Client Stack-----
Transmission and retransmission timers allotted is freed
TransactionDatastructureList is freed successfully in Diameter stack
SessionDatastructureList is freed successfully in Diameter stack
ConnectionDatastructureList is freed successfully in Diameter stack
Tel_DC_UnInit success;Diameter Client stack uninitialization is success

```

Figure 11: *TerminateConnection API SUCCESS with TerminateSession Called before TerminateConnection*

```

tata-ims3@tata-ims3-desktop: ~/Desktop/soemya/TC/DiameterCode
*****DIAMETER PROTOCOL STACK IMPLEMENTATION*****
*****RUNNING*****
*****I'm In Diameter client *****
-----Preparing for Initialization of Diameter client stack-----
Tel_DC_Init() success
Diameter Client stack initialization is success
-----Preparing for connecting Diameter client to Diameter server -----
startConnection (PreestablishConn) success(Tel_DC_startConnection) success
Diameter Client is successfully connected with the Diameter Server
Connection ID allotted to the connected Diameter Client and Diameter Server is 0
*****Beginning the session between Diameter Client and Diameter server for the next activity*****
Tel_DC_StartSession() success
Session is successfully established between Diameter Client and Diameter Server
Session ID allotted to the connected Diameter Client and Diameter Server is 0
-----Without calling Terminate Session directly calling Terminate connection which performs terminating the session
allotted to that respective connection and finally connection allotted between Diameter Client and Diameter Server-----
-----Preparing for Terminating Connection between Diameter Client and Diameter Server-----
Tel_DC_TerminateConn() success.
Session and Connection allotted between Diameter Client and Diameter Server terminated successfully
SessionID and ConnID allotted are as follows:
SessionID 0 terminated successfully
ConnID 0 terminated successfully
-----Preparing for uninitialization of Diameter Client Stack-----
Tel_DC_UnInit success;Diameter Client stack uninitialization is success

```

Figure 12: *TerminateConnection API SUCCESS without TerminateSession Called before TerminateConnection*

### 7.7. Uninitialization API SUCCESS

UnInitialization API is called (having no parameter) resulting in SUCCESS of UnInitialization of Stack by de-allocating all its resources.

```

tata-ims3@tata-ims3-desktop: ~/Desktop/soemya/TC/DiameterCode
*****DIAMETER Protocol Stack For CLIENT*****
-----Diameter Client performing Successful UnInitialization of Diameter stack-----
-----Preparing for Initialization of Diameter client stack-----
Transmission interval allotted is 5000000
Retransmission interval allotted is 300000
Retransmission attempts allotted is 2
ConnectionDatastructureList which includes 5 number of connection datastructures is created successfully in Diameter stack
SessionDatastructureList which includes 10 number of session datastructures is created successfully in Diameter stack
TransactionDatastructureList which includes 10 number of Transaction datastructures is created successfully in Diameter stack
-----status of Initialization of Diameter client stack-----
Tel_DC_Init() success
Diameter Client stack initialization is success
-----Preparing for UnInitialization of Diameter Client Stack-----
Transmission and retransmission timers allotted is freed
TransactionDatastructureList is freed successfully in Diameter stack
SessionDatastructureList is freed successfully in Diameter stack
ConnectionDatastructureList is freed successfully in Diameter stack
-----status of UnInitialization of Diameter client stack-----
Tel_DC_UnInit success;Diameter Client stack uninitialization is success

```

Figure 13: *UnInitialization API SUCCESS*

**8.Reference**

1. V. Fajardo, J. Arkk, J. Loughney, G. Zorn, “Diameter Base Protocol”, Internet draft-ietf (RFC 6733), October 2012.
2. P. Calhoun, H. Akhtar, J. Arkko, E. Guttman, A. Rubens, “Diameter Base Protocol”, Internet draft-ietf (RFC 3588), September 2003.
3. C. Rigney, S. Willens, A. Rubens, and W. Simpson., “Remote Authentication Dial In User Service (RADIUS)”. RFC 2865, IETF Network Working Group, June 2000.
4. Rakesh Khandelwal, TATA Consultancy Services, Ltd. “The Importance of Standard IP Multimedia Subsystem (IMS) Architecture”.
5. Mladen Stanke, Mile Sikic., “Comparison of the RADIUS and Diameter Protocols”, IEEE paper, June 2008
6. Man Yup Lee, Tai-Myoung Chung., “Diameter-Based AAA Architecture to Support Small AAA Client”, IEEE paper 2010.
7. Vinay Kumar. S. B, Manjula N Harihar., “Diameter-Based Protocol in the IP Multimedia Subsystem”, IJSCE, January 2012.