

Diameter Management

DESCRIPTION

Copyright

© Ericsson AB 2015, 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
2	Functions and Concepts	3
2.1	Types of Operation	6
3	Managed Object Model	9
4	Configuration Management	11
5	Fault Management	13





1 Introduction

This document provides an overview of the management model and concepts associated with the Diameter managed area.

A managed area is represented by a group of Managed Object Classes (MOCs) within the Managed Object Model (MOM).





2 Functions and Concepts

Diameter Management provides a management interface for Diameter handling of the Managed Element (ME).

Diameter Overview

Diameter is a protocol providing connection and message handling services. A Diameter node represents a generic node that implements the Diameter Base Protocol in compliance with RFC 6733, and acts either as a client, a server, or an agent.

In a network, a Diameter node has connections to one or many other Diameter nodes (its Peer nodes). Diameter messages are passed over such connections. Two Diameter nodes can be connected by a single or multiple connections. Multiple connections enable load balancing across processors and higher throughput.

Connections can either be established over the Transmission Control Protocol (TCP) or the Stream Control Transmission Protocol (SCTP). Those protocols can be run over IPv4 and IPv6.

Diameter Own Node and Connection Directions

The Diameter Own node is the Diameter node function provided by an ME, see Figure 1.

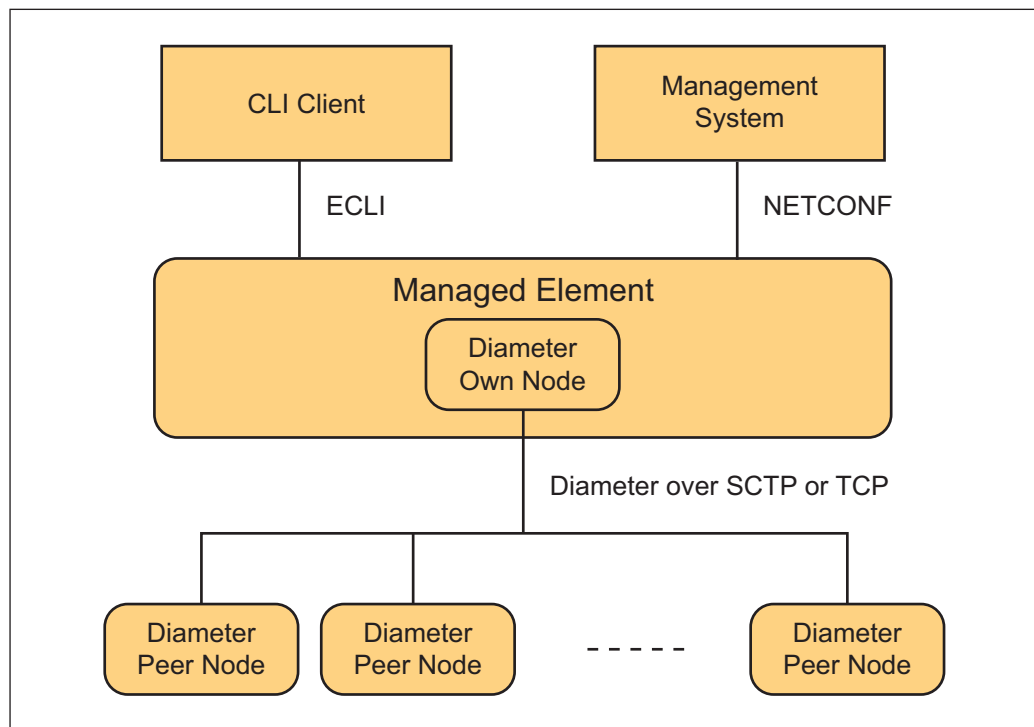


Figure 1 Diameter Management Overview

An outgoing connection is a connection initiated by the Diameter Own node to a Peer node. The Diameter Own node is the connection initiator for such connection. An incoming connection is a connection initiated by a Peer node of the Diameter Own node. The Diameter Own node is the connection responder for such connection.

Realm Routing Table

The Realm Routing Table (RRT) is the routing table used in each Diameter node to route Diameter messages. A Diameter node uses its RRT to determine whether a Diameter message must be handled locally, forwarded to the next hop, or redirected.

Diameter Message Format

The Diameter message consists of a message header and a message payload. The payload consists of one or more Attribute-Value Pairs (AVPs). An AVP includes a header and is used to encapsulate protocol-specific data (for example, routing information) and authentication, authorization, or accounting information. The format of the Diameter header and AVP header is shown in Figure 2.

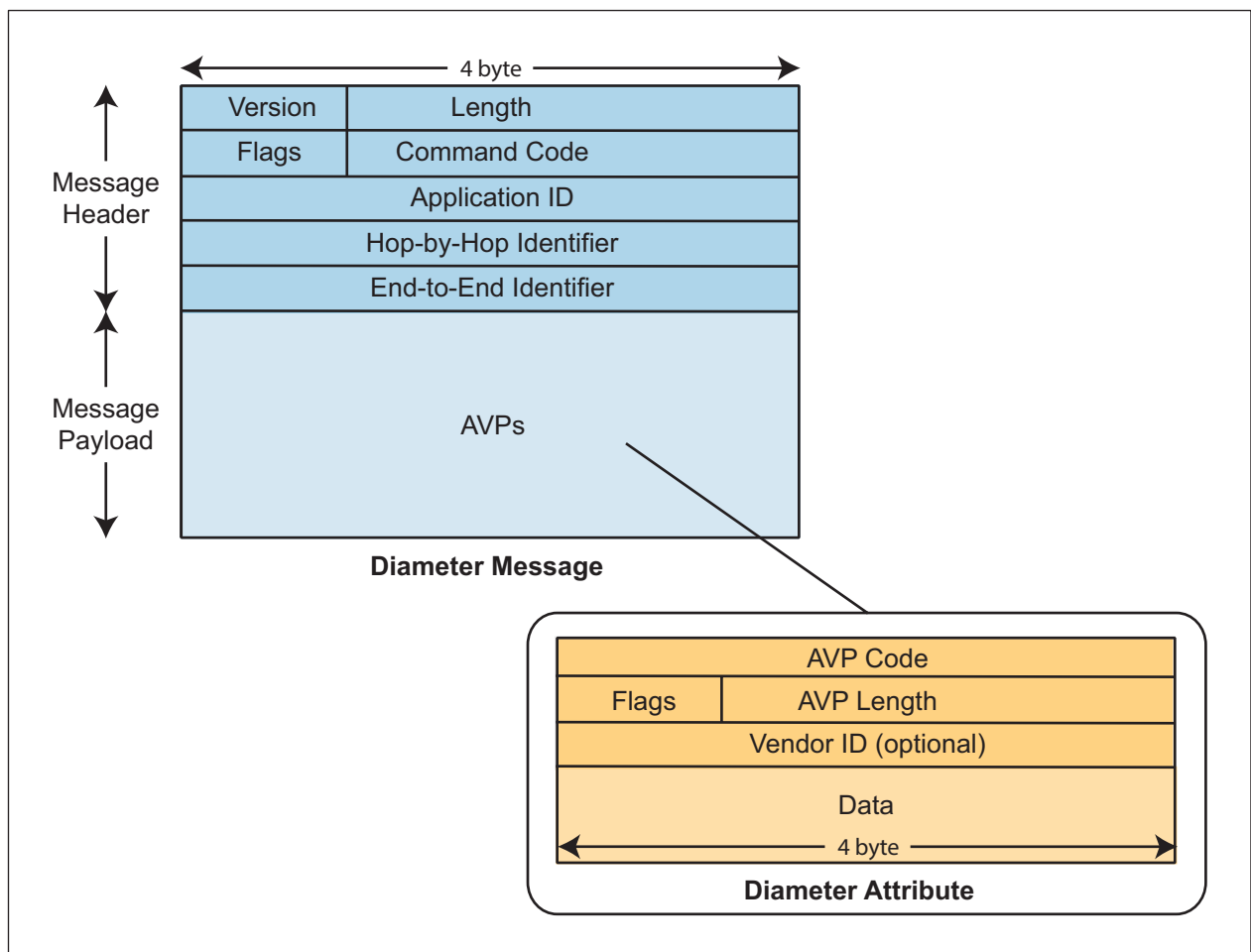


Figure 2 Diameter Message Format

Diameter Base

The Diameter base functions consist of connection handling and message handling.

Connection handling encompasses the following:

- Transport establishment service
- Capabilities exchange service
- Transport failure detection service
- Disconnection service

Message handling encompasses the following:

- Routing of outgoing requests/answers



- Discriminate incoming request/answers based on application ID

Diameter Application

A Diameter application is an extension of the Diameter Base Protocol. It adds new commands or AVPs, or both, to the Diameter base. A Diameter application is identified by its application ID.

Diameter Stack

A Diameter stack is the set of Diameter applications used in Diameter clients and servers to establish Diameter communication according to an interface specification.

Stack Container

A stack container is the ME Diameter configuration used by a single Diameter stack or shared by multiple Diameter stacks. A stack container encompasses the following configuration data:

- Own node configuration
- Peer and connection-related configuration
- RRT configuration

2.1 Types of Operation

Diameter Management supports the following operations:

Configure Own Node

This operation configures the Diameter own node, acting either as a client, as a server, or as an agent. The Diameter Base Protocol allows the establishment of direct connections between a generic Diameter node (called the own node) and one or more Diameter peer nodes. Diameter messages can be transferred over the TCP or the SCTP; the latter transferred over IPv4 or IPv6. For further details on how to perform this operation, refer to *Configure Own Node*.

Configure Peer Node and Connections

This operation adds and configures a Peer node and the associated connection, or connections, to the Own node. For further details on how to perform this operation, refer to *Configure Peer Node and Connections*.



Configure Realm Routing Table

This operation adds and configures the RRT. For further details on how to perform this operation, refer to *Configure Realm Routing Table*.

Disable Peer Node

This operation disables a Peer node. It can be done when all connections for the Peer node are disconnected. For further details on how to perform this operation, refer to *Disable Peer Node*.



3 Managed Object Model

The Diameter Management managed area is represented in the *Managed Object Model (MOM)* as follows:

```
ManagedElement
+-DIA-CFG-Application
  +-DIA-CFG-Configuration
  +-DIA-CFG-DictionaryContainer
    +-DIA-CFG-Vendor
    +-DIA-CFG-AvpDef
  +-DIA-CFG-StackContainer
    +-DIA-CFG-OwnNodeConfig
    +-DIA-CFG-PeerNodeContainer
      +-DIA-CFG-NeighbourNode
      +-DIA-CFG-Conn
  +-DIA-CFG-RoutingContainer
    +-DIA-CFG-Drt
      +-DIA-CFG-AccReqContainer
      +-DIA-CFG-AppRouting
      +-DIA-CFG-AuthReqContainer
      +-DIA-CFG-AppRouting
```

For general information about the MOM, MOCs, Managed Objects (MOs), cardinality, and related concepts, refer to *Managed Object Model User Guide*.

The Diameter MOCs are described in Table 1.

Table 1 Diameter Management Managed Object Class Descriptions

Managed Object Class	Description
<i>DIA-CFG-Application</i>	The root of the Diameter Management model.
<i>DIA-CFG-Configuration</i>	Handles a common Diameter application configuration.
<i>DIA-CFG-DictionaryContainer</i>	Container for underlying MOCs.
<i>DIA-CFG-Vendor</i>	Defines a vendor Diameter application.
<i>DIA-CFG-AvpDef</i>	Handles an AVP.
<i>DIA-CFG-StackContainer</i>	Container for underlying MOCs.
<i>DIA-CFG-OwnNodeConfig</i>	Handles the Own node.
<i>DIA-CFG-PeerNodeContainer</i>	Container for underlying MOCs.
<i>DIA-CFG-NeighbourNode</i>	Handles a Peer node.
<i>DIA-CFG-Conn</i>	Handles a Peer node connection.

*Table 1 Diameter Management Managed Object Class Descriptions*

Managed Object Class	Description
<i>DIA-CFG-RoutingContainer</i>	Container for underlying MOCs.
<i>DIA-CFG-Drt</i>	Represents an entry in the RRT.
<i>DIA-CFG-AccReqContainer</i>	Container for underlying MOC.
<i>DIA-CFG-AppRouting</i>	Handles routing information.
<i>DIA-CFG-AuthReqContainer</i>	Container for underlying MOC.



4 Configuration Management

Diameter Management is accessed using NETCONF or the Ericsson Command-Line Interface (ECLI) to manipulate the Management Information Base (MIB).

The following operations can be performed by the user and are described in Operating Instructions using the ECLI:

- *Configure Own Node*
- *Configure Peer Node and Connections*
- *Configure Realm Routing Table*
- *Disable Peer Node*





5 Fault Management

The Diameter Management alarms are described in Table 2.

Table 2 Diameter Management Alarms

Alarm	Description
<i>vDicos, Diameter Link Congested</i>	Raised when the number of messages starts to get discarded because of congestion at the Diameter transport layer as it exceeds a configured threshold value in a given time interval.
<i>vDicos, Diameter Link Disabled</i>	Raised when the relevant Diameter connection is administratively disabled or disabled by the Diameter Peer node.
<i>vDicos, Diameter Link Failure</i>	Raised when a Diameter connection has failed.
<i>vDicos, Diameter Own Node Disabled</i>	Raised when the Diameter Own node has been disabled.
<i>vDicos, Diameter Peer Node Disabled</i>	Raised when a Diameter Peer node has been disabled.