

Diameter Management

DESCRIPTION

Copyright

© Ericsson AB 2015–2017. 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	Understanding Diameter Management	1
1.1	Key Diameter Management Concepts	1
1.2	Diameter Own Node and Connection Directions	1
1.3	Realm Routing Table	2
1.4	Diameter Message Format	2
1.5	Diameter Base	3
1.6	Diameter Application	4
1.7	Diameter Stack	4
1.8	Stack Container	4
2	Basic Diameter Management Procedures	4
3	Diameter Management-Related Alarms	5





1 Understanding Diameter Management

1.1 Key Diameter Management Concepts

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.

The Diameter Management area is represented by a group of Managed Object Classes (MOCs) under *DIA-CFG-Application* within the Managed Object Model (MOM). For general information about the MOM, MOCs, cardinality, and related concepts, refer to *Managed Object Model User Guide*.

1.2 Diameter Own Node and Connection Directions

The Diameter Own node is the Diameter node function provided by an ME. 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.

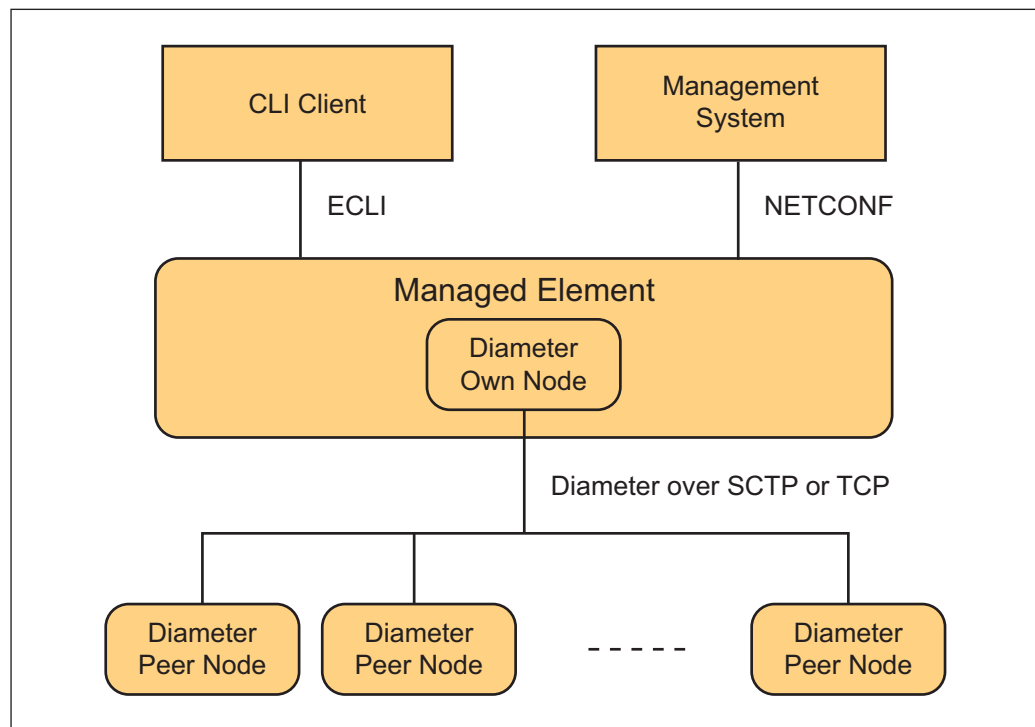


Figure 1 Diameter Management Overview

1.3 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.

1.4 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.

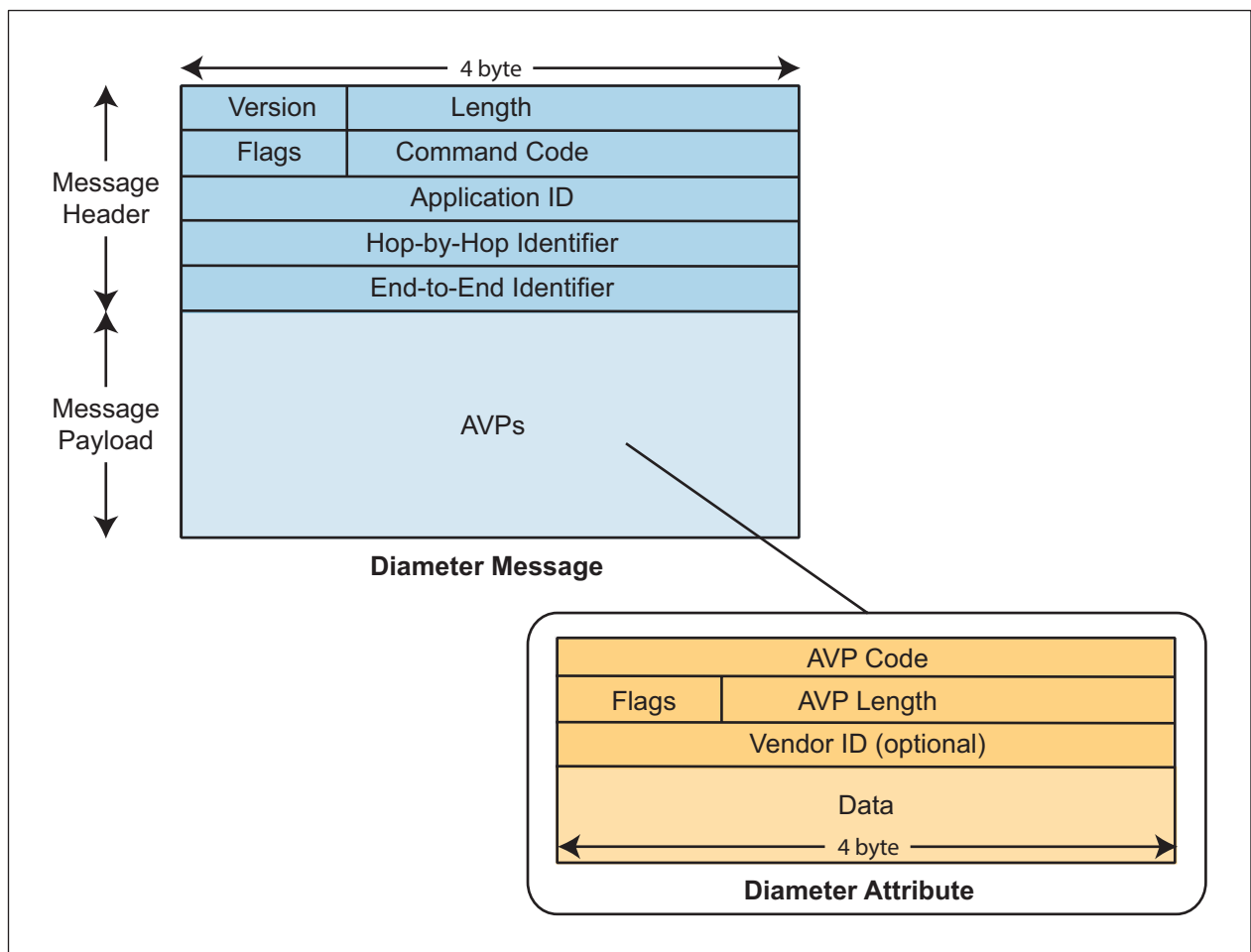


Figure 2 Diameter Message Format and AVP Header

1.5 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

1.6 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.

1.7 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.

1.8 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 Basic Diameter Management Procedures

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

- 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 Diameter Management-Related Alarms

Table 1 Diameter-Related Alarms

Alarm	Description
<i>vDicos, Diameter Link Congestion</i>	The alarm is raised because of congestion at the Diameter transport layer.
<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.