Introduction to Restcomm Diameter

Table of Contents

Diameter is a computer networking protocol for Authentication, Authorization and Accounting (), as defined in RFC3588. It is a successor to RADIUS, and has been designed to overcome certain RADIUS limitations:

- No transport reliability and flexibility (Diameter uses TCP/SCTP instead of UDP).
- No security within protocol (Diameter supports IPSec (mandatory) and TLS (optional)).
- Limited address space for AVPs (Diameter uses 32-bit address space instead of 8-bit).
- Only stateless mode is possible (Diameter supports both stateful and stateless modes).
- Static peers (Diameter offers dynamic discovery, using DNS, SRV and NAPTR).
- No peer alignment capabilities (Diameter introduces capabilities negotiation).
- No support for transport layer failover. Diameter follows RFC3539, which introduces correct procedures.
- Limited support for roaming (Diameter introduces mechanisms for secure and scalable roaming).
- No extension possible (Diameter allows extension - new commands and AVPs to be defined).

Diameter offers all of the capabilities of the RADIUS protocol, and is compatible with RADIUS. It can also define extensions, or "Applications".

Each application may introduce new types of messages, AVP codes, and state machines. The Message and AVP codes are assigned by the . Each application has its own Application ID and Vendor ID that is used to distinguish between applications. Application code is used to signal to other peers which operations are supported by the connecting peer (Capabilities Exchange / Negotiation).

Message Format

Diameter is a byte based protocol. Each message has a fixed structure, which consists of two parts: header and payload.

The message header structure is common for every message. The content is fixed, as is the length. Message header content includes the code, application and certain bit flags, which helps identify the message in Diameter scope.

The message payload is built up of AVPs. Its content differs for each command and application, though they all define the Session-ID AVP as mandatory.

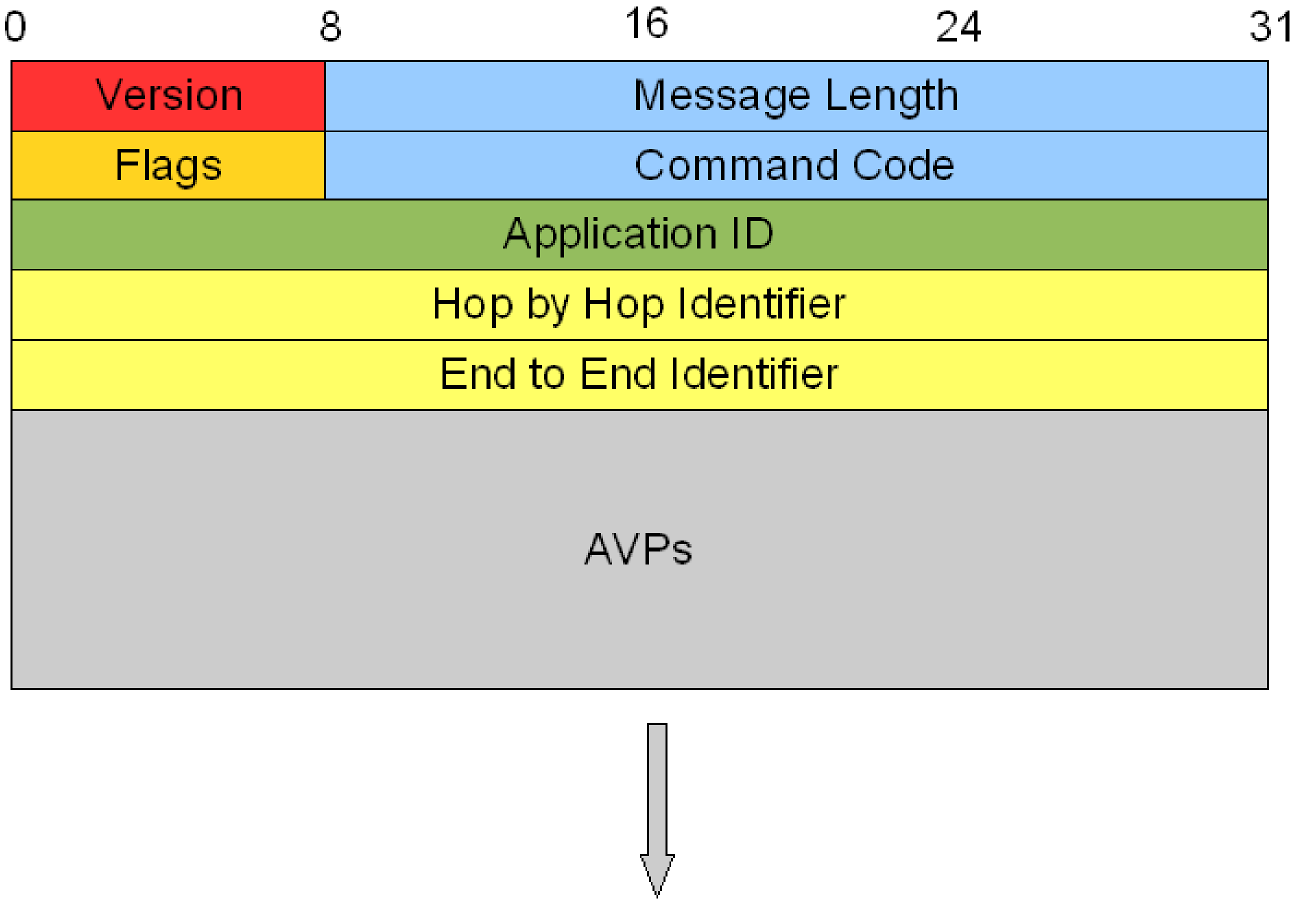

Figure 1. Diameter Message Structure

The header has the following fields:

Message Headers

Version

Indicates the Diameter protocol version. This value is always set to 1.

Message Length

Indicates the Diameter message length, including the header fields.

Flags

Composed by eight bits, to provide information regarding the message. The first four bits in the flags octet have the following meaning:

- R = The message is a request (1) or an answer (0).
- P = The message is proxiable (1) and may be proxied, relayed or redirected, or it must be processed locally (0).
- E = The message is an error message (1) or a regular message (0).
- T = The message is potentially being re-transmitted (1) or being sent for the first time (0).

The last four bits are reserved for future use, and should be set to 0.

Command Code

Indicates the command associated with the message.

Application-ID

Identifies the application to which the message is applicable for. The application is an authentication, accounting, or vendor specific application. The `application-id` in the header must be the same as what is contained in any relevant AVPs in the message.

Hop-by-Hop ID

A unique ID that is used to match requests and answers. The header field of the answer message must contain the same value present in the corresponding request. This is how answers are routed back to the peer that sent the message.

End-to-End ID

A time-limited unique ID that is used to to detect duplicate messages. The ID must be unique for at least four minutes. The answer message originator must ensure that this header contains the same value present in the corresponding request.

The message payload is built up from AVPs. Each AVP has a similar structure: a header, and encoded data. Data can be simple (eg, integer, long) or complex (another encoded AVP).

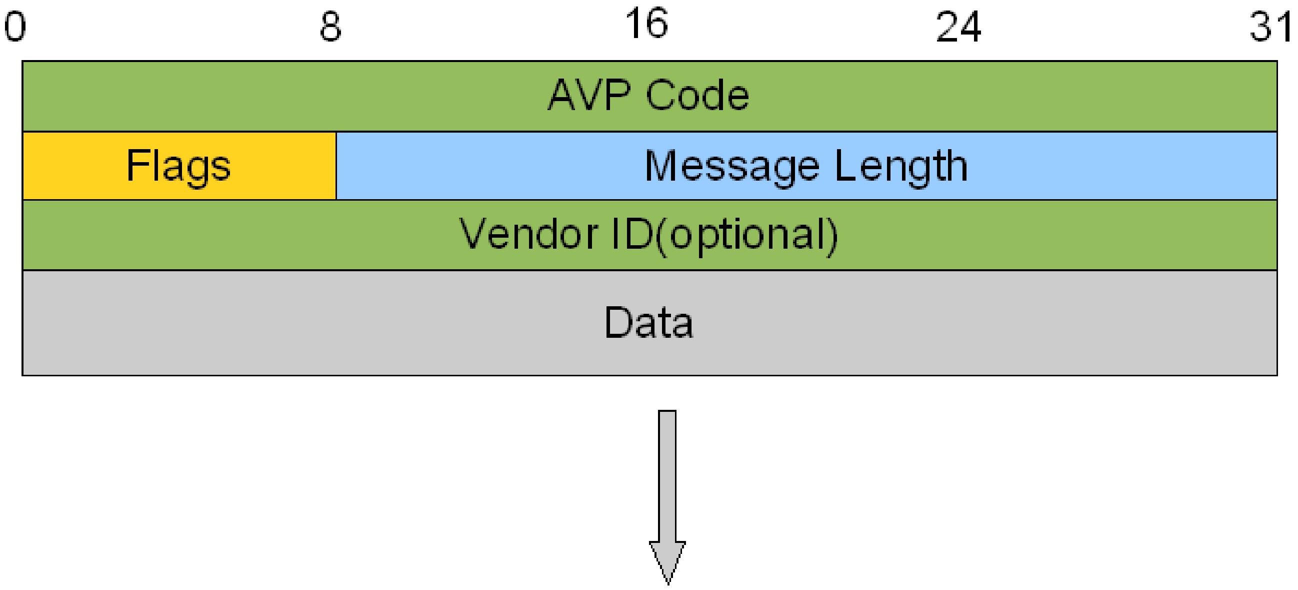

Figure 2. Payload Structure

Payload AVPs

AVP Code

Uniquely identifies the attribute, by combining the specified code with the value contained within the Vendor-ID header field. AVP numbers 1 to 255 are reserved for RADIUS backwards compatibility, and do not require the Vendor-ID header field. AVP numbers 256 and above are used exclusively for the Diameter protocol, and are allocated by IANA.

Flags

Bit flags that specify how each attribute must be handled. Flags octets have the following structure: V M P r r r r r. A full description is available in Section 4.1 of RFC3588. The first three bits have the following meaning:

- V If set, indicates that optional octets (Vendor-ID) is present in AVP header.
- M If set, it indicates that receiveing peer must understand this AVP or send error answer.
- P If set, it indicates the need for encryption for end-to-end security.

The last 5 bits are reserved for future use, and should be set to 0.

AVP Length

Indicates the number of octets in the AVP, including the following information:

- AVP Code
- AVP Length
- AVP Flags
- Vendor-ID field (if present)
- AVP Data

Vendor-ID

An optional octet that identifies the AVP in application space. AVP code and AVP Vendor-ID create a unique identifier for the AVP.

Contents

Restcomm Diameter core is built on top of three basic components:

Stack

Extensible Diameter Stack. It provides basic session support along with application specific sessions.

Multiplexer (MUX)

Diameter Stack multiplexer. Allows different listeners to share the same stack instance.

Dictionary

Diameter Message and AVP Dictionary. Provides an API to access information about AVPs. Dictionary is embeded in the MUX.