DIAMETER
Architecture and Base Protocol

EFORT
http://www.efort.com

Diameter is the protocol used within EPS/IMS architectures for AAA (Authentication, Authorization, and Accounting). It is intended to work both in home networks and in roaming situations between visited and home networks.
Diameter is specified primarily as a base protocol by the IETF in RFC 3588 and then by the RFC 6733 which obsoletes RFC 3588. The name is a pun on the name of the predecessor protocol, RADIUS(Remote Authentication Dial In User Service) - a diameter is twice the radius. Diameter is not directly backward compatible but does provide an upgrade path for RADIUS.
DIAMETER base protocol must be used in conjunction with DIAMETER applications (also called DIAMETER interfaces) which complement the base protocol functionality. The base protocol contains the basic functionality and is implemented in all Diameter nodes, independently of any particular application. Applications are extensions to the basic functionality that are tailored for a particular usage of Diameter in a particular environment.
DIAMETER applications are used in mobile environments within different architectures, including EPS (Evolved Packet System), IMS (IP Multimedia Subsystem), PCC (Policy and Charging Control), GAA/GBA (Generic Authentication Architecture / Generic Bootstrapping Architecture) and M2M (Machine to Machine). More than 60 DIAMETER applications have already been defined for telecommunications, particularly by 3GPP.
This tutorial introduces the DIAMETER architecture and DIAMETER based protocol.

1 DIAMETER Architecture

The DIAMETER architecture consists of a number of entities :

- Diameter Node: A host process that implements the Diameter protocol.
- Diameter Peer: A diameter node that has a direct transport connection with another diameter node.
- Client : A Diameter Client is a device at the edge of the network that performs access control. Examples of Diameter clients are MME (Mobility Management Entity), PCEF (Policy and Charging Enforcement Function) in EPS architecture.
- Server : A Diameter Server is one that handles authentication, authorization, and accounting requests for a particular realm. Example of Diameter server is HSS (Home Subscriber Server) and PCRF (Policy and Charging Rules Function) in EPS architecture.
- Agent : A Diameter Agent is a Diameter node that provides relay, proxy, redirect or translation services.
  - Relay Agent : Relay Agents are Diameter agents that accept requests and route messages to other Diameter nodes based on information found in the messages (e.g., Destination-Realm). This routing decision is performed using the Realm Routing Table, which informs about the next hop for a given destination-Realm. Relays dot not perform any application level processing. Relay Agents modify Diameter messages by inserting and removing routing information, but do not modify any other portion of a message. Relays should not maintain session state but must maintain transaction state.
  - Proxy Agent : Similarly to relays, proxy agents route Diameter messages using the Diameter Routing Table. However, they differ since they modify messages to

implement policy enforcement. Proxies may maintain session state and must maintain transaction state. Since enforcing policies requires an understanding of the service being provided, Proxies must only advertise the Diameter applications they support. Example of Proxy agent is the Diameter Routing Agent (DRA).

- Redirect Agent : Redirect Agents do not relay messages, and only return an answer with the information necessary for direct communication with destination. Redirect Agents do not modify messages. Since redirect agents do not receive answer messages, they cannot maintain session state. Further, since redirect agents never relay requests, they are not required to maintain transaction state. Since redirect agents do not perform any application level processing, they provide relaying services for all Diameter applications, and therefore must advertise the Relay Application Identifier. Example of Redirect agent is the SLF (Subscription Locator Function in IMS).
- Translation Agent : A Translation Agent translates between two protocols, such as RADIUS and Diameter or MAP and DIAMETER. In this case, the translation agent supports a RADIUS to Diameter migration, allowing server conversions to Diameter, for example, while permitting the NASes to be converted at a slower pace. Example of Translation agent is one which translates S6 Diameter Interface into Cx MAP interface because an HLR is deployed in Evolved Packet System instead of HSS..

Diameter Clients must support the base protocol, which includes accounting. In addition, they must fully support each Diameter application that is needed to implement the client's service.
Diameter Servers must support the base protocol, which includes accounting. In addition, they must fully support each Diameter application that is needed to implement the intended service.
Diameter Relays and redirect agents are, by definition, protocol transparent, and must transparently support the Diameter base protocol, which includes accounting, and all Diameter applications.
Diameter proxies must support the base protocol, which includes accounting. In addition, they must fully support each Diameter application that is needed to implement proxied services.

2 DIAMETER message format

A Diameter message consists of a fixed-length 20-octet header followed by a variable number of AVPs (Attributed Value Pair). The format of a Diameter message is shown on the figure 1.

- The Version field indicates the Diameter protocol version and is set to 1 for now.
- The Command flags field specifies 4 flags for now:
  - R flag (stands for Request) shows whether the message is a request or a response.
  - P flag (stands for Proxiable) shows if the message can be proxied, relayed or redirected or it must be locally processed.
  - E flag (stands for Error) to show if the message contains protocol or semantic errors. When a request message generates a protocol error an answer message is sent back with the ‘‘E’’ bit set in the Diameter header, indicating a protocol error.
  - T flag to show that a message can potentially be a retransmitted message after a link fail-over or is used to aid removal of duplicate messages.
  - r : these flag bits are reserved for future use, and must be set to zero, and ignored by the receiver.
- The command code value indicates the command associated with the message, such as “credit-control-request ” or “accounting-request”, and so on. Every Diameter message

must contain a command code so that the receiver can determine what action it needs to take for each message. The command code is the same of the request and its corresponding answer.

- Application ID identifies the specific application the message is used for, such as S6a/S6d between MME and HSS, Gx between PCEF and PCR, etc.
- Hop-by-hop identifier field carries an identifier that is used to match request and responses over that hop. The sender of the request must ensure that the identifier is unique over the connection on that hop at any given time. The sender of a response must ensure that the identifier value is the same as that in the corresponding request. . The Hop-by-Hop identifier is normally a monotonically increasing number, whose start value was randomly generated. An answer message that is received with an unknown Hop-by-Hop Identifier must be discarded. Hop-by-Hop identifier allows a Diameter response to follow the same route as the corresponding Diameter request.
- End-to-end identifier is an identifier used to detect duplicate messages. The identifier in a response message must match the identifier in the corresponding request message. The identifier must remain locally unique for at least 4 minutes. This identifier and the Origin-Host AVP are used together to detect message duplicates. Note duplicate request could cause duplicate responses but the duplications must not affect any states that were created by the original request.

What follows the DIAMETER command header is a set of AVPs (Attribute Value Pair)

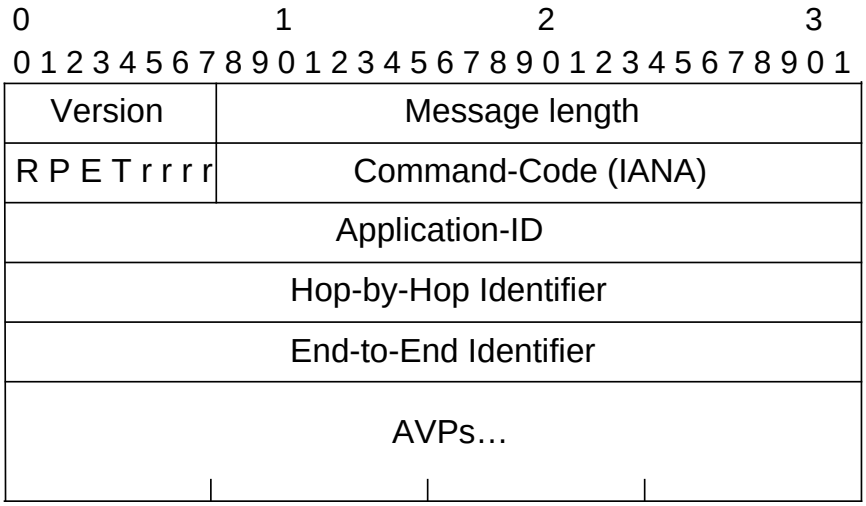


Figure 1 : DIAMETER message format

2.1 Hop-by-hop identifier

Assume the following example shown on figure 2 :

- A Diameter message is received by a first agent on the path between client and server. This message has Hop-By-Hop Id set to 1.
- The first Relay/Proxy agent stores this id and replaces it by its own Id (i.e., 3). It also stores the transport address of the client the message has been sent from (i.e., IP1, Port1). The same process applies at each hop which is a relay/proxy agent until reaching destination.
- The second Relay/Proxy agent stores the id of the received message (i.e., 3) and replaces it by its own Id (i.e., 4). It also stores the transport address of the first agent the message has been sent from (i.e., IP2, Port3).
- The server reuses the Hop-by-Hop Id received and inserts it in the response (i.e., 4).
- The second Replay/Proxy agent finds in its mapping table an entry for Id = 4. It replaces Id = 4 by Id = 3 and returns the response to the transport address IP2, Port2.

- The first Replay/Proxy agent finds in its mapping table an entry for Id = 3. It replaces Id = 3 by Id = 1 and returns the response to the transport address IP1, Port1 which is received by client.

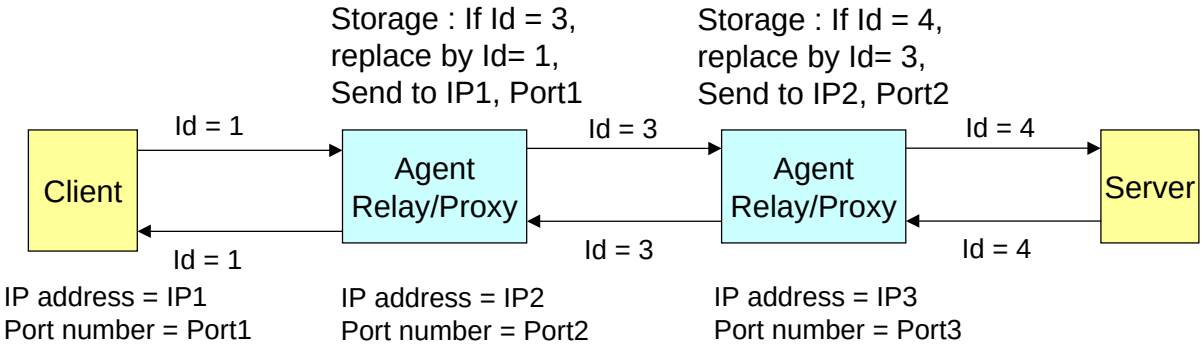


Figure 2 : Example of Hop by Hop Identifier

2.2 Application Id

Each Diameter application must have an IANA (Internet Assigned Numbering Authority) assigned Application Identifier. The base protocol does not require an Application Identifier since its support is mandatory. During the capabilities exchange, Diameter nodes inform their peers of locally supported applications. Furthermore, all Diameter messages contain an Application Identifier, which is used in the message forwarding process.
Relay and redirect agents must advertise the Relay Application Identifier, while all other Diameter nodes must advertise locally supported applications. The receiver of a Capabilities Exchange message advertising Relay service must assume that the sender supports all current and future applications.
Diameter relay and proxy agents are responsible for finding an upstream server that supports the application of a particular message. If none can be found, an error message is returned with the Result-Code AVP set to DIAMETER_UNABLE_TO_DELIVER. Figure 3 lists some Application Ids.

Application	Application Id
Diameter Common Messages	0
NASREQ	1
Diameter Base Accounting, Rf , Gz	3
Diameter Credit Control, Ro, Gy	4
Relay	0xFFFFFFFF
Cx/Dx Interface Application	16777216
Rx Interface Application	16777236
Sh/Dh Interface Application	16777217
Re Interface Application	16777218
S6a/S6d Interface Application	16777251
S13/S13 ' Interface Application	16777252
S9 Interface Application	16777267
Gx Interface Application	16777238
Sy Interface Application	16777302
SWx Interface Application	16777265

Figure 3 : Application Id examples in EPS/IMS

2.3 Diameter message header format example

The format of the Diameter request header shown in figure 4 is as follows :

- The Version field is set to value 1.
- The R flag is set to 1 because it is a requests.
- The length is always multiple of 4 bytes. It includes the header size (20 bytes) plus the size of all the headers in the request. In this example the size of the request is 464 bytes
- P flag is set to 1 so that the request may be handled by agents.
- E flag (stands for Error) is always set to 0 for a request.
- T flag is set to 0 because it is the first transmission of that request.
- The reserved bits are set to 0.
- The command code value indicates the command associated with the message. AIR command code is 318 as specified by the S6a/S6d DIAMETER application.
- Application ID identifies the specific application the message is used for. For S6a/S6d, it is 16777251.
- Hop-by-hop identifier field carries an identifier that is used to match request and responses over that hop.
- End-to-end identifier is an identifier used to detect duplicate messages. The identifier in a response message must match the identifier in the corresponding request message.

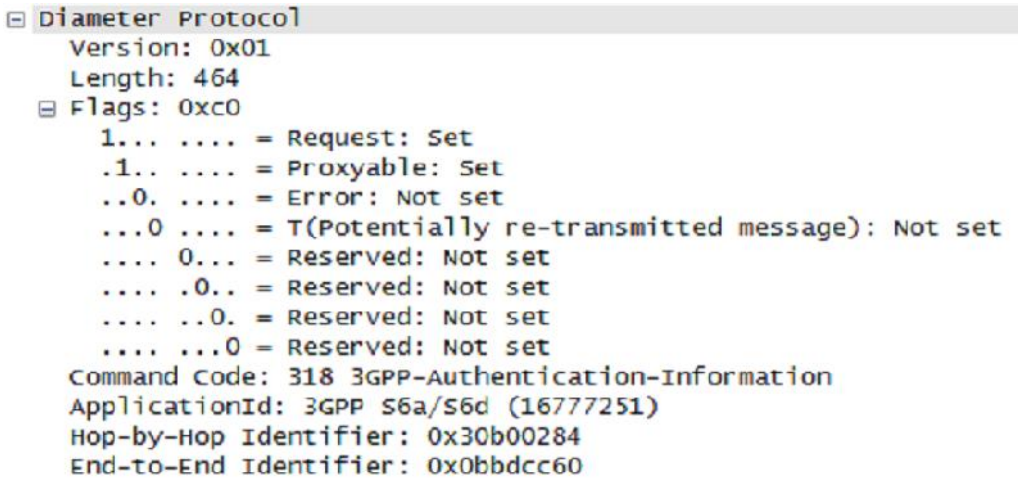


Figure 4 : Example of DIAMETER request header

3 Diameter base protocol messages

For different purposes, Diameter base protocol has defined several types of Diameter messages, which are identified by their command code. For example, an Accounting-Request message recognizes that the message carries accounting-related information, while a Capability-Exchange-Request message recognizes that the message carries capability information of the Diameter node sending the message. Because the message exchange style of Diameter is synchronous, each message has its corresponding counterpart, which shares the same command code. The receiver of an Accounting-Request message prepares an Account-Answer message and sends it to the original sender. Figure 5 lists the messages defined in the Diameter based protocol.

Command-Name	Abbreviation	Command-Code
I. Diameter Connection Management		
Capabilities-Exchange-Request	CER	275
Capabilities-Exchange-Answer	CEA	275
Device-Watchdog-Request	DWR	280
Device-Watchdog-Answer	DWA	280
Disconnect-Peer-Request	DPR	282
Disconnect-Peer-Answer	DPA	282
II. Generic Session Operations		
Abort-Session-Request	ASR	274
Abort-Session-Answer	ASA	274
Re-Auth-Request	RAR	258
Re-Auth-Answer	RAA	258
Session-Termination-Request	STR	275
Session-Termination-Answer	STA	275
III. Offline Charging		
Accounting-Request	ACR	271
Accounting-Answer	ACA	271

Figure 5 : Commands of the Diameter base protocol

3.1 Diameter connection management messages

The communication between two diameter peers starts with the establishment of a transport connection (TCP or SCTP) (Figure 6). The initiator then sends a capabilities-Exchange-Request (CER) to the other peer, which responds with a Capabilities-Exchange-Answer (CEA). Protocol version number, supported Diameter applications, security mechanisms, are examples of capabilities. The connection is then ready for exchanging application messages. If no messages have been exchanged for some time either side may send a Device-Watchdog-Request (DWR) and the other peer must respond with Device-Watchdog-Answer. Either side may terminate the communication by sending a Disconnect-Peer-Request (DPR) which the other peer must respond to with Disconnect-Peer-Answer. After that the transport connection can be disconnected. Diameter connection management message are only exchanged between direct peers. They are never routed through agents.

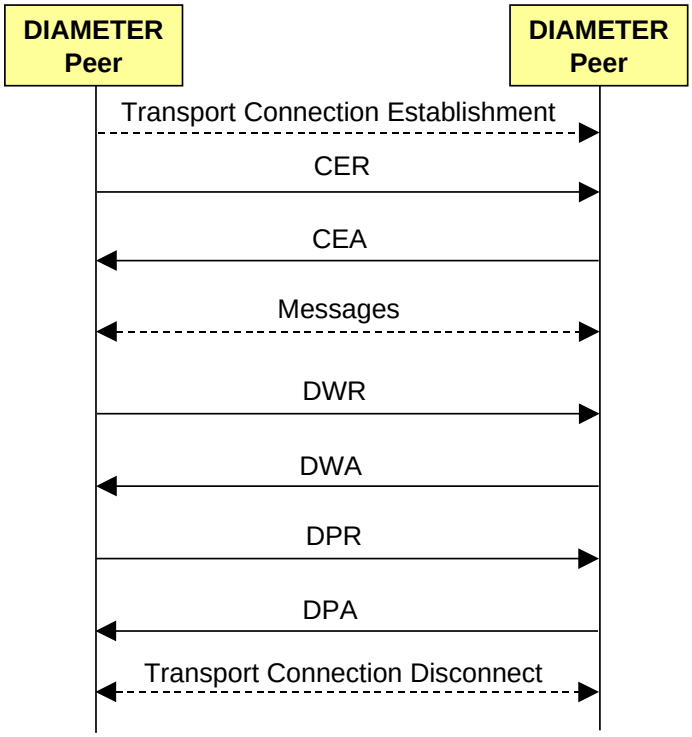


Figure 6 : Diameter connection management messages

3.2 Diameter generic session operations

The Diameter protocol isn't bound to a specific application running on top of it. It focuses on general message exchanging features. Because authentication and authorization mechanisms vary among applications, the Diameter base protocol doesn't define command codes and AVPs specific to authentication and authorization. It is the responsibility of Diameter applications to define their own messages and corresponding attributes based on the application's characteristics.

For example, the AA-Request message is used to request authentication and/or authorization for a given NAS (Network Access Server) user in the NASREQ application (RFC 4005). A authentication and/or authorization session may be established.

A Diameter server may initiate a re-authentication and/or re-authorization service for a particular session by issuing a Re-Auth-Request (RAR) message. For example, for pre-paid services, the Diameter server that originally authorized a session may need some confirmation that the user is still using the services. If a NAS receives an RAR message with Session-Id equal to a currently active session and a Re-Auth-Type that includes authentication, it must initiate a re-authentication toward the user, if the service supports this particular feature.

Session termination messages are only used in the context of authentication and authorization, and only when the session state was maintained.

A session termination message can be initiated by either the Diameter client or the Diameter server. When a session is deemed to be closed, the Diameter client sends a Session-Termination-Request message to the Diameter server. Alternately, if the Diameter server detects that the session should be closed -- perhaps because the user runs out of credit or just for administrative purposes -- the Diameter server sends an Abort-Session-Request message to the Diameter client.

Figure 7 shows authentication mechanisms where a network access server (NAS) at the edge of a network interacts with a backend AAA server through Diameter protocol to accomplish access control.

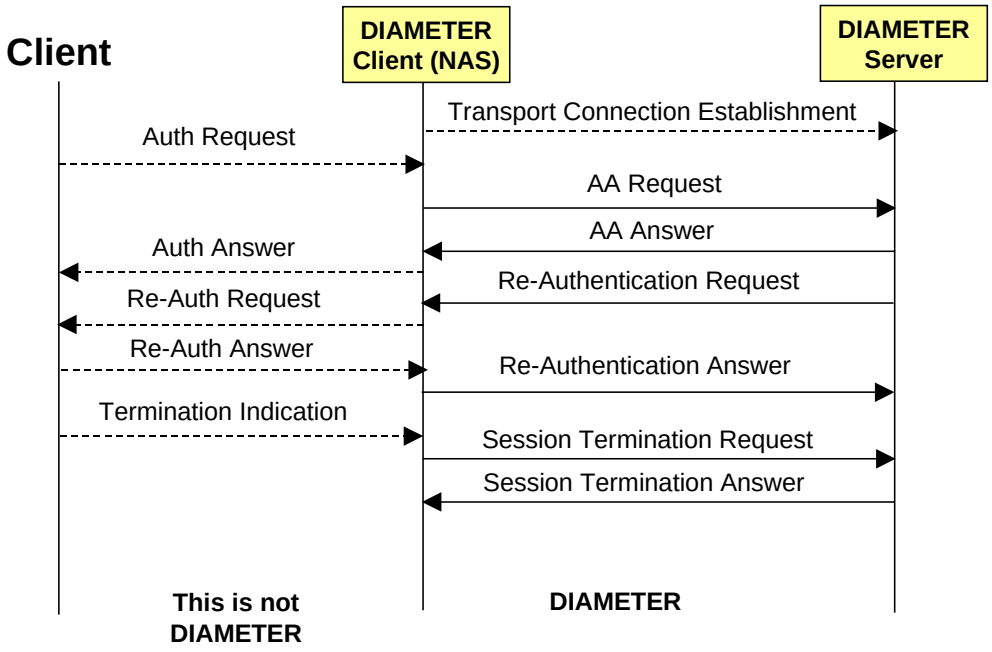


Figure 7 : NAS authentication, re-authentication and session termination

3.3 Diameter accounting messages

The Diameter base protocol provides accounting services to Diameter applications (Figure 8). When an accounting session is not active, there are no resources reserved for it in either the Diameter client or the Diameter server. If successful Accounting Request (ACR) activates an accounting session, in which the accounting records exchanged fall into two categories, based on the accounting service type:

- Measurable length services have clearly defined beginnings and ends (e.g., voice session). An accounting record is created when the service begins and another is sent when the service ends. Optionally, interim accounting records can be produced at certain intervals within the measurable length session.
- One-time events are services without a measurable service length (e.g., SMS transfer). In a one-time event accounting record the beginning of the service and the end of the service actually coincide; therefore, a one-time event only produces a single accounting record.

Accounting records are correlated with the Session-Id AVP, which is a globally unique identifier and present in all AAA messages.

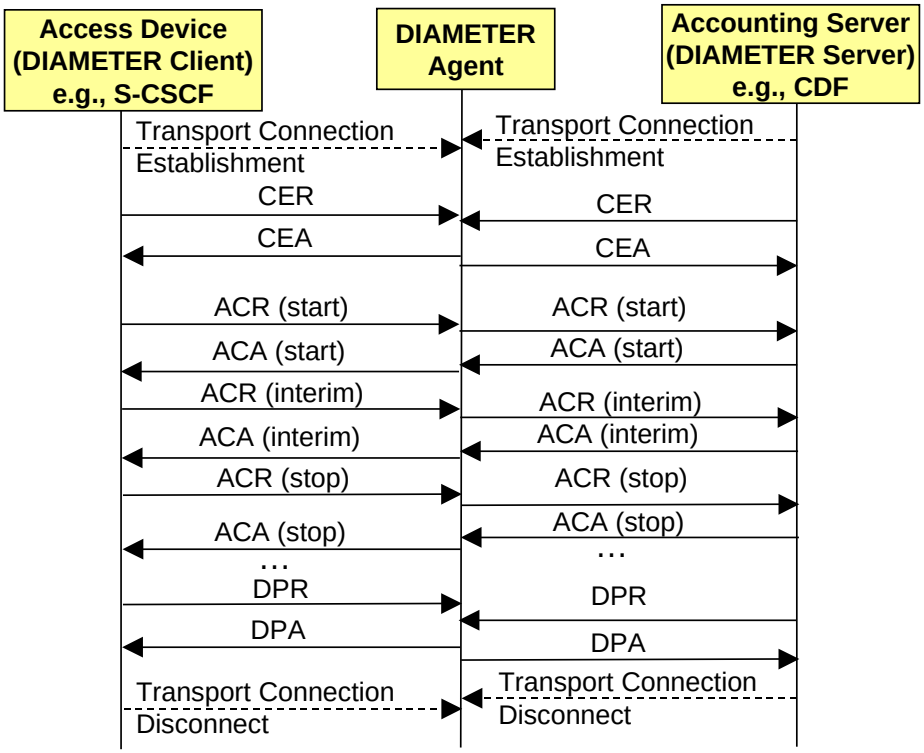


Figure 8 : Diameter accounting messages

4 Attributed Value Pairs (AVP)

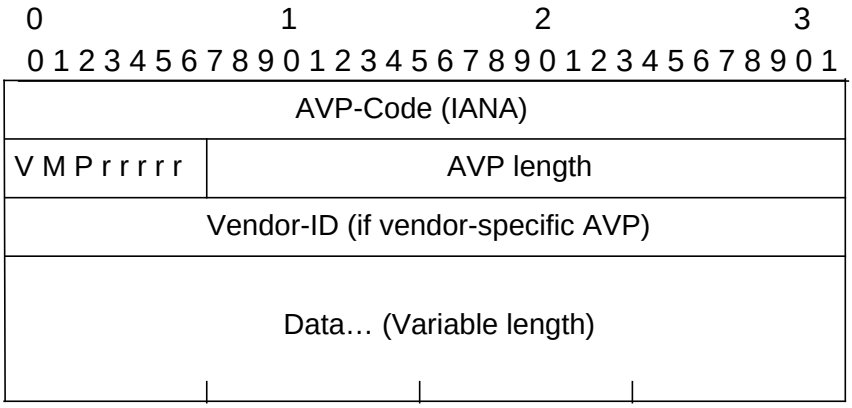
Diameter messages, like RADIUS messages, transport a collection of Attribute-Value Pairs (AVPs). These AVPs carry the detail of AAA as well as routing, security, and capability information between two Diameter nodes. An AVP is a container of data. Figure 9 depicts the structure of an AVP. Each AVP contains an AVP Code, Flags, anAVP Length, an optional Vendor-ID and Data.

- AVP Code : The AVP code identifies the type of information (attribute) included in the attribute data field of the AVP. AVP code values are standardized by IETF for the AVPs of the base protocol. New applications should try using the existing AVPs to the extent possible. Diameter saves AVP codes 0-255 for backward compatibility with attributes defined by RADIUS. New applications can define additional AVPs if necessary.
- AVP Flags :
  - The “V” bit, known as the Vendor-Specific bit, indicates whether the optional Vendor-ID field is present in the AVP header. When set the AVP Code belongs to the specific vendor code address space. If set to 0, it means absence of a Vendor-ID field, which indicates a standard AVP specified by IETF.
  - The “M” bit indicates if the AVP is mandatory (M bit set to 1) or optional (M bit set to 0). If the sender indicates that support for the AVP is mandatory and the receiver does not understand the AVP, the Diameter request is rejected; If the AVP is optional and not understood by the destination, it may be ignored.
  - “P” bit: The P bit indicates the need for encryption for end-to-end security. Diameter base protocol specifies which AVPs must be protected by end-to-end security measures (encryption) if the message is to pass through a Diameter agent. If a message includes any of those AVPs, it must not be sent unless there is end-to-end security between the originator and recipient of the message.

- rrrrr bits: Indicate existence of 5 reserved bits in the flags field.

- The AVP Length indicates the length of the AVP, including the AVP Code, AVP Length, Flags, Vendor-ID (if present), and the Data field.

- The Data field contains some data specific to the attribute. The field has a length of zero or more octets. The length of the data is derived from the AVP Length field.



V=Vendor bit, M=Mandatory bit, P=Protected bit.
3GPP vendor Id = 10415

Figure 9 : AVP format

Diameter messages may also embed grouped AVPs. Grouped AVP have the same format as single AVPs except that the data field of grouped AVPs contains one or more AVPs rather than raw data.

4.1 AVP format examples

In figure 10, Destination-Realm AVP is an AVP defined by the DIAMETER based protocol. Therefore its V bit is set to 0 and there is no presence of AVP Vendor Id. Visited-PLMN-ID is an AVP defined by 3GPP for its DIAMETER applications. Therefore the V bit is set to 1 and AVP Vendor Id field is present. The corresponding value is 10415 which is 3GPP Vendor Id.
The exact length of an AVP (including the AVP code length (4 bytes) , AVP Flags length (1 byte), possible AVP Vendor Id Length (4 bytes), AVP length (3 bytes) and Value length (N bytes)) is indicated in AVP Length.
If this length is not multiple of 4 bytes, then some padding is added.
If AVP Length indicates 41 (as in the example), the recipient receives 44 bytes and ignores the 3 last bytes.
This is the reason, the length of a DIAMETER command is always multiple of 4 bytes. Length of DIAMETER command = 20 bytes (command header) + sum of the lengths of its AVPs.

AVP Code: 283 Destination-Realm
AVP Flags : 0x40
0… …. = (V) Vendor-specific : Not Set
.1.. …. = (M) Mandatory: Set
..0. …. = (P) Protected: Not set
…0 …. = Reserved: Not set
…. 0… = Reserved: Not set
…. .0.. = Reserved: Not set
…. ..0. = Reserved: Not set
…. …0 = Reserved: Not set
AVP Length: 41 -- (44 padded bytes)
Value: epc.mnc001.mcc208.3gppnetwork.org

AVP Code: 1407 Visited-PLMN-Id
AVP Flags: 0xc0
1… …. = (V) Vendor-specific : Set
.1.. …. = (M) Mandatory: Set
..0. …. = (P) Protected: Not set
…0 …. = Reserved: Not set
…. 0… = Reserved: Not set
…. .0.. = Reserved: Not set
…. ..0. = Reserved: Not set
…. …0 = Reserved: Not set
AVP Vendor Id: 3GPP (10415)
AVP Length: 19 -- (20 padded bytes)
Value: 208.001

Figure 10 : Examples of AVP format

RFC 3588, P. Calhoun et al., Diameter Base Protocol, Sept 2003.
RFC 6733, obsoletes RFC 3588, V. Fajardo et al. Diameter Base Protocol, Oct 2012.
Nadjid Nakhjiri et al., AAA and network security for mobile access, Wiley 2005.