Diameter Base Protocol

Functional Specification

1091349 1.23a

Diameter Base Protocol

Functional Specification

1091349 1.23a

Continuous Computing Corporation

9380 Carroll Park Drive San Diego, CA 92121-2256

***Phone:** +1 (858) 882-8800 Fax: +1 (858) 777-3389*

***Web:** http://www.ccpu.com*

Diameter Base Protocol
Functional Specification
1091349 1.23a

Contributors: Continuous Computing Development Team, Naveen D’cruz, Kevin MacDowell.

Printed in U.S.A.

Contents

Figures

Tables

Preface

Objective

This document provides a functional description of the Diameter Base Protocol software (p/n 1000349), release 1.2, designed by Continuous Computing Corporation. This product is referred as Diameter in the rest of the document.

Audience

Continuous Computing assumes that the readers of this document are familiar with telecommunication protocols, specifically AAA protocols.

Document Organization

This document contains the following sections:

Section	Description
1 Introduction	Provides an overview of the product, including the product description and features.
2 Application	Describes the network architecture and protocol stacks.
3 Protocol Conformance	Specifies the standards to which the software conforms, and the features supported.
4 Product Highlights	Defines Trillium Advanced Portability Architecture (TAPA) and describes the product-specific features.
5 Licensing Options	Describes the licensing information for this software.
6 Memory Size	Details the performance characteristics and memory size, which includes total code sizes.

Section	Description
Appendix A: Addendum	Describes the addendum changes.

Document Set

The suggested reading order of this document set is:

1. *Diameter Base Protocol Functional Specification*

   Contains the features and highlights of the protocol and system characteristics, including the memory characteristics and conformance details.

2. *Diameter Base Protocol Training Course*

   Describes the features and interfaces of the software, including code samples, data flow diagrams, and a list of its files.

3. *Diameter Base Protocol Service Definition*

   Describes the procedures and layer manager interface used for passing information between the software and other software elements. The Interface Primitives section describes the services of the software. The Procedures section describes and shows the flow of primitives and messages across the interfaces.

4. *AQU Interface Service Definition*

   Describes the internal upper layer primitives for the AQU interface with the Diameter service user. The Interface Primitives section describes the software services. The Interface Service Definition describes the interface procedures defined for the service provider software.

5. *Diameter Base Protocol Porting Guide*

   Describes the files and procedures necessary to port the software to the operating system, into a specific processor family and system architecture. The document lists the product, common and sample files associated with the software.

6. *Diameter Base Protocol Software Test Sample*

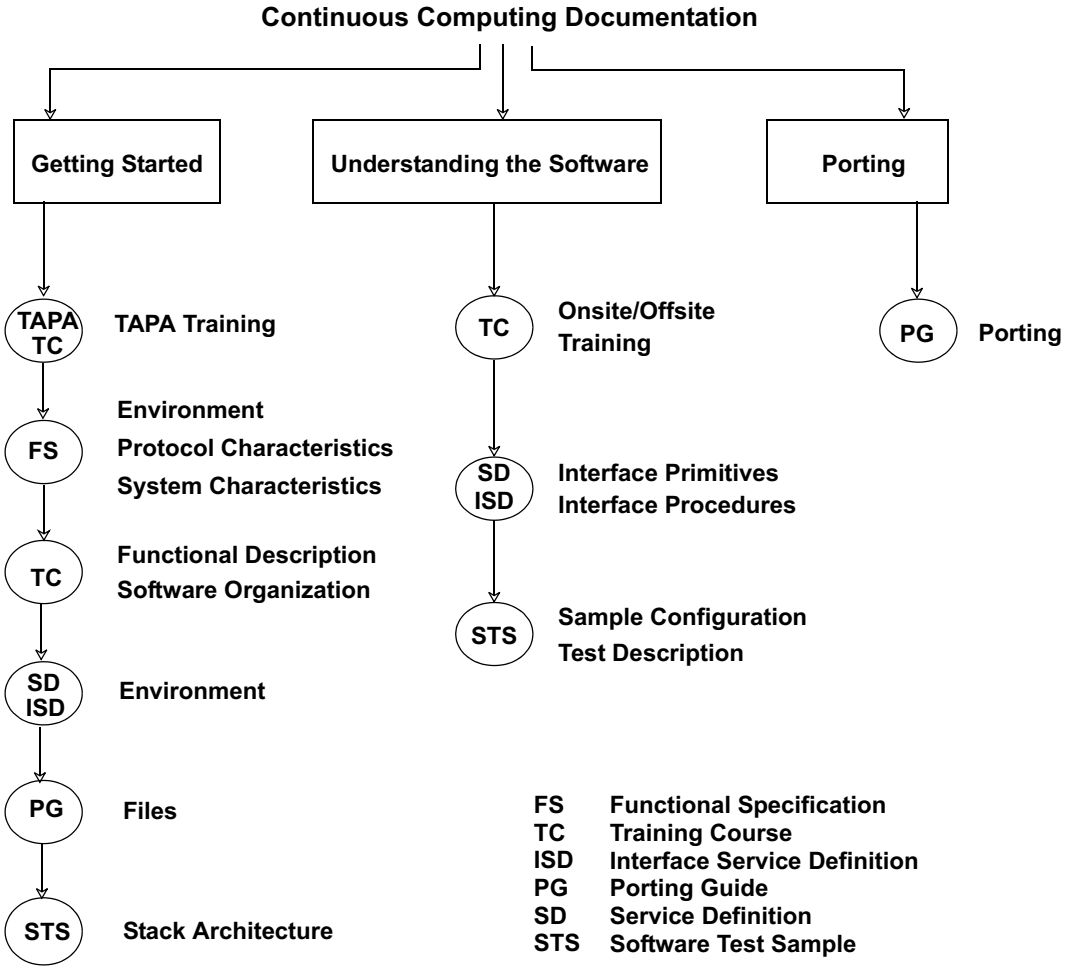
   Describes the sample files delivered with the product and the procedures to build a sample test. This test partially demonstrates the product initialization, configuration and execution. The document may contain data flow diagrams illustrating the correct operation of the software.

7. *User Guide*

   Contains an overview of the design and operation of the product. The document describes the key data structures and operations of the module. With the help of examples, the document describes product use. The functions that access the services of the product are also covered in this document.

Using Continuous Computing® Documentation

The figure below shows the various user approaches to using the software documentation. First time users must read the documents under the **Getting Started** column, where important sections and subsections are listed to the right of each document. For users familiar with the documentation, but who need to look up certain points concerning software use, **Understanding the Software** column is suggested. The **Porting** column is for users familiar with Trillium software and related telecommunications protocols and wish to install the software immediately onto their development environments.



Notations

The following table displays the notations used in this document:

Notation	Explanation	Examples
Arial	**Titles**	**1.1 Title**
Book Antiqua	Body text	This is body text.
Bold	**Highlights information**	**Loose coupling, tight coupling, upper layer interface**
ALL CAPS	CONDITIONS, MESSAGES	AND, OR CONNECT ACK
Italics	*Document names, emphasis*	*Diameter Functional Specification* This adds *emphasis*.
Courier New Bold	**Code** **Filenames, pathnames**	**PUBLIC S16** **IbMiLibCfgReq (pst, cfg)** **Pst *pst;** **IbMngmt *cfg;;**

Release History

The following table lists the history of changes in successive revisions to this document:

Version	Date	Author (s)	Description
1.23a	August 22, 2007	Naveen Kottapalli	Update for Toolkit Phase-1 release.
1.22a	June 16, 2007	V. Srinivasa Rao	Addendum release for updates.
1.21a	April 16, 2007	V. Srinivasa Rao	Addendum release for updates.
1.2	September 14, 2006	V. Srinivasa Rao	Final release for Diameter software version 1.2.
1.11a	May 19, 2006	Ajith Kumar	Updated the Portability section.
1.1	March 17, 2006	Deepak Wadhwa	Initial release. Conforms to Diameter software release, version 1.1.

1

Introduction

1.1 Product Description

Diameter Base Protocol is the authentication, authorization, and accounting framework for applications such as network access and IP mobility.

The Diameter software is the implementation of the Diameter Base Protocol specified in RFC 3588.

The Diameter software provides the following basic capabilities:

- Support for Diameter Server and Diameter Client.
- Support for Relay agent.
- Failover and Failback. This feature can be configurable.
- Diameter Peer Discovery:
  - Manual Setup: Pre-configured Peer.
  - Dynamic Setup: SLPv2

    Note: Dynamic Peer Discovery using SLPv2 is not tested on Solaris platform.
- Peer State Management.
- Peer Connection Management.
- Diameter Message Encoding/Decoding.
- Ability to extend AVP database to new Diameter applications for application specific or vendor specific AVPs.
- SCTP/TCP transport option.
- Both IPv4 and IPv6 options.

  Note: IPV6 is tested over TCP transport.

- Security Interface provided by TLS/IPSec.
- Signaling message validation, error handling, and reporting.
- Establishing and managing the signaling transport bearers.
- Support for Translation Agent to RADIUS can be built as an application.
- Protocol trace, debug reporting.
- Logging of errors in received messages using Alarms.

The following features are not supported as part of this release:

- Proxy and Re-direct agents are not supported.

  Note: Agents can be built as applications.

- Peer Discovery using NAPTR is not supported.
- Session State machine is not supported. It can be implemented as part of application.
- TLS over Continuous Computing's SCTP is not supported.

1.2 Acronyms

The following table lists the acronyms used in this document:

Acronym	Expansion
3GPP	Third Generation Partnership Project
AVP	Attribute Value Pair
SAP	Service Access Point
SCTP	Stream Control Transmission Protocol
TUCL	TCP/UDP Convergence Layer

For a list of commonly used terms please refer to the Engineering Glossary (part number PREN026) at http://www.ccpu.com/search/glossary/

2

Application

2.1 Network Architecture

The Diameter Protocol provides support for Authentication, Authorization, and Accounting (AAA) in the network.

- Authentication:
  - Authentication of end users prior to allowing access to network.
- Authorization:
  - Authorization of service or authority provided by the network.
- Accounting:
  - Collecting usage information of resources by end users.
  - Monitoring and Billing.

Diameter Protocol is used in both 3GPP and 3GPP2 networks for AAA purposes.

- IETF Applications:
  - Mobile IP, NASREQ, EAP, Credit Control (CC), SIP.
- 3GPP IM (IP Multimedia):
  - Sh Interface, Cx and Dx Interface, Gq Interface, Rf and Ro Interface.
  - WLAN UE/AN and AAA server Accounting (charging).
  - Circuit/packet switched, WLAN domain charging.
  - IM Subsystem online/offline charging.
  - MMS, LCS architecture charging.
- 3GPP2 IMS:
  - Cx, Sh interface online/offline charging.
  - Credit-control.

Figure 2-1 shows the Diameter Network Architecture

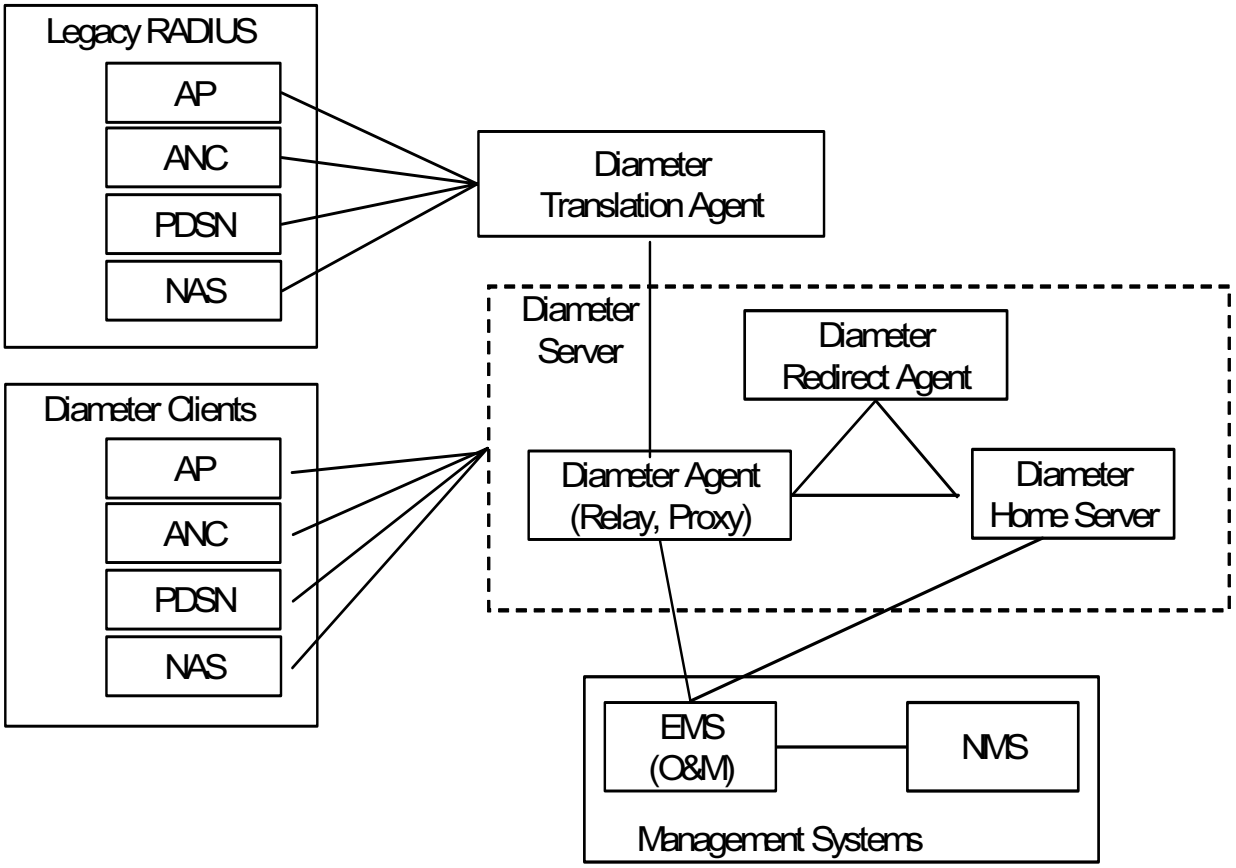


Figure 2-1: Diameter Network Architecture

Different types of Diameter Agent and Server functions have been defined for Diameter Nodes.

Diameter Server

A Diameter Server is one that handles authentication, authorization and accounting requests for a particular realm. By its very nature, a Diameter Server MUST support Diameter applications in addition to the base protocol.

Diameter Client

A Diameter Client is a device at the edge of the network that performs access control. An example of a Diameter client is a Network Access Server (NAS) or a Foreign Agent (FA).

Diameter Node

A Diameter node is a host process that implements the Diameter protocol and acts as a Client, Agent or Server.

Diameter Agent

A Diameter Agent is a Diameter node that provides either relay, proxy, redirect or translation services.

EMS

Element Management System is used as interface between the administrator and the Diameter. This is used for sending configuration, control requests and also to receive any alarms from the Diameter.

NMS

Network Management System is used to monitor and control the various Diameter nodes in the network.

2.2 Diameter Stack Architecture

Figure 2-2 shows the Diameter protocol stack architecture.

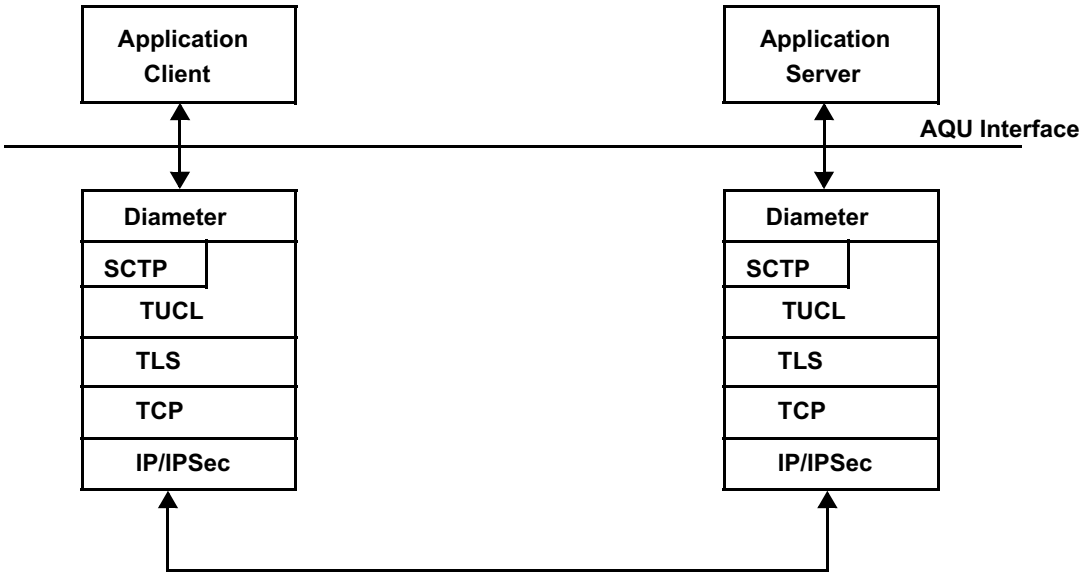


Figure 2-2: Diameter protocol stack architecture

TUCL provides TCP and UDP transport services, Raw IP Support and TLS over TCP to its service users. TUCL uses the socket interface to obtain the services of the native TCP/IP stack in the operating environment. To support TLS, TUCL uses the OpenSSL library.

Diameter uses the TCP and TLS services provided by TUCL layer.

2.2.1 SAP Architecture

Diameter has an Upper SAP per application type. For example: Sh, Cx/Dx, and Rf/Ro applications want to utilize the services of the Diameter layer, then they must have one SAP each for Sh, Cx/Dx, and Rf/Ro respectively.

Diameter has a lower SAP per transport type. If both TCP and SCTP transport options are chosen, then Diameter has one lower SAP towards the TCP and other lower SAP towards the SCTP.

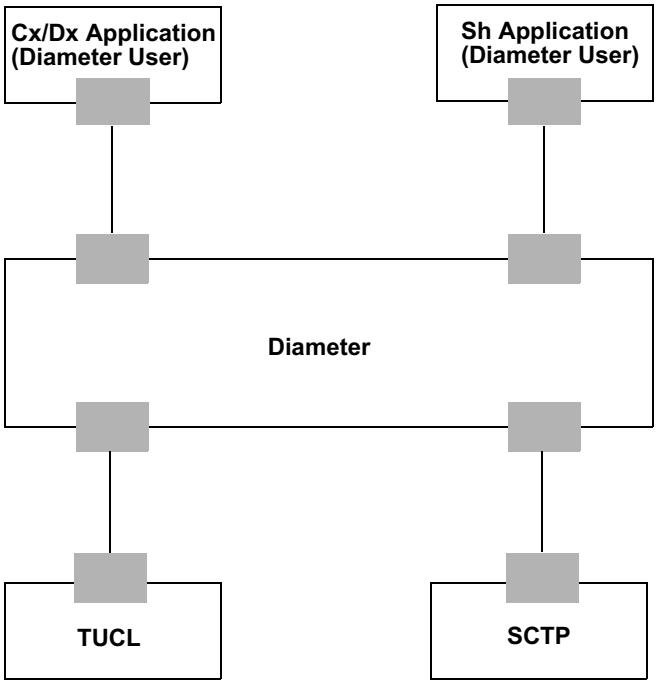


Figure 2-3: SAP Architecture for Diameter

As shown in the Figure 2-3, each application has to open a SAP with the Diameter layer to utilize the services of Diameter layer.

The following 3GPP applications can utilize the services of the Diameter layer:

- Cx/Dx application
- Sh application
- Rf application
- Ro application
- Gq application

TUCL provides the TCP and UDP transport services, Raw IP support and TLS transport over TCP to its service users. TUCL uses the socket interface to obtain the services of the native TCP/IP Stack in the operating environment.

3

Protocol Conformance

3.1 Standards

Continuous Computing's Diameter software supports the below mentioned specifications:

- RFC 3588, "Diameter Base Protocol".
- RFC 3539, "Authentication, Authorization and Accounting Transport Profile".
- RFC 3589, "Diameter Command Codes for 3GPP R5".
- 3GPP TS 29.230 V 6.9.0 (2006-12) 3GPP specific codes and identifiers (Release 6).
- 3GPP TS 29.230 V 7.5.0 (2006-12) 3GPP specific codes and identifiers (Release 6).
- X.S0013-011 IMS Sh Interface.
- X.S0013-006 IMS Cx Interface.
- X.S0013-008 IMS Offline Accounting (Ro/Rf Interface).

Diameter can support the following 3GPP application specs, if configured appropriately:

- 3GPP TS 29.229, "Cx and Dx Interfaces based on Diameter Base protocol".
- 3GPP TS 29.329, "Sh Interface based on Diameter Base protocol".
- 3GPP TS 32.299, "Charging Management; Diameter Charging Applications Ro/Rf".
- 3GPP TS 29.210, "Charging rule provision over Gx Interface".
- 3GPP TS 29.211, "Rx Interface based on Diameter Base protocol".
- 3GPP TS 29.209, "Policy control over Gq Interface".
- ETSI ES 283.034, "E4 Interface based on Diameter Base Protocol".
- ETSI ES 283.035, "E2 Interface based on Diameter Base Protocol".

- ETSI ES 283.026, “Rq Interface based on Diameter Base Protocol”
- RFC 4006, “Diameter Credit Control Application”.

3.2 Feature Support

The following table lists the features supported by the software:

Table 3-1: Features

Feature	Supported?
Diameter supports the protocol functionality for support of different Diameter applications.	Yes
Diameter supports Base Protocol AVPs for encoding and decoding protocol messages. Diameter protocol provides capability to add Application and Vendor specific AVPs.	Yes
Diameter performs message validation, error handling, and reporting for the protocol messages received from the peer node.	Yes
Diameter establishes and manages the signaling transport connections.	Yes
Diameter supports SCTP/TCP transport.	Yes
Diameter provides support for IPSec/TLS for Security purposes. **Note**: TLS support is not available if Continuous Computing’s SCTP is used.	Yes
Diameter provides support for Peer State management.	Yes
Diameter provides support for Relay agent implementation.	Yes
Diameter provides support for agent implementation (Re-direct and Proxy). **Note:** Agents can be built as an application on top of Diameter.	No
Diameter provides support for Client and Server implementation.	Yes
Diameter provides support for Failover and Failback.	Yes

Table 3-1: Features

Feature	Supported?
Diameter provides support for translation Agent. This is considered as part of application and not supplied with Base Protocol implementation.	No
Diameter provides support for peer discovery through SLPv2 protocol. **Note:** In this release SLPv2 support is tested on Linux platform only.	Yes
Diameter provides support for peer discovery through NAPTR.	No
Diameter provides support for Session State machine **Note:** This is considered as part of application and not supplied with Diameter Base protocol implementation.	No

4

Product Highlights

4.1 TAPA

Trillium Advanced Portability Architecture (TAPA) defines the software architecture that ensures portability of Trillium products across various platforms. Each Trillium product, represented by the box in the center of Figure 4-1, is equivalent to an Open Systems Interconnection (OSI) layer. The product has four interfaces that enable it to communicate with the other software modules in the system: the upper layer interface (UI), lower layer interface (LI), layer manager interface (LMI), and system services interface.

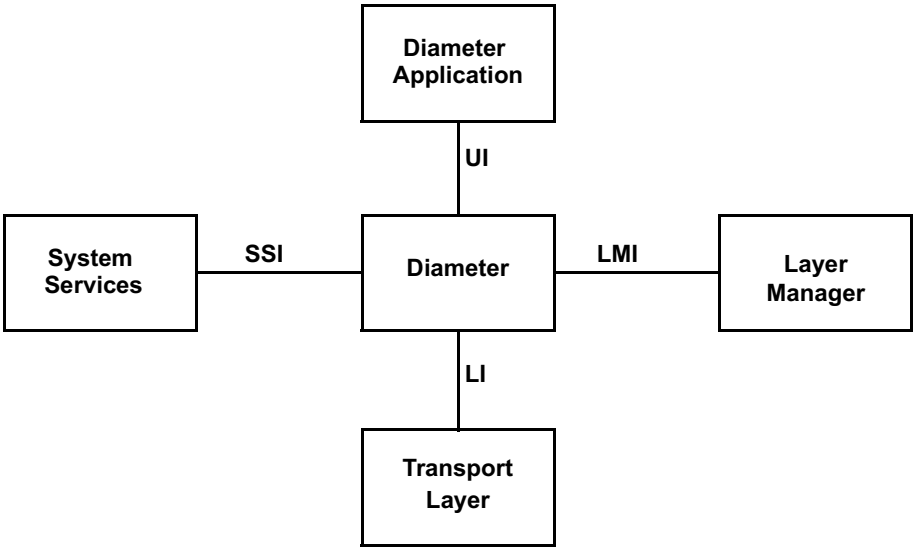


Figure 4-1: Trillium advanced portability architecture (TAPA)

Interfaces

An interface is shared between two layers and used for the exchange of primitives between the layers.

Interface	Description
AQU (upper layer) interface	Provides the services of this protocol layer to the upper (service user) layers. The capabilities of the protocol layer are mapped to a set of primitive functions that the upper layer can access for providing higher level services. Functions provided at this interface include binding, unbinding, connection, disconnection, data transfer, and other functions specific to the protocol layer.
Layer manager interface (LAQ)	Manages the layer resources and provides functions such as run-time configuration, control, statistics, status, alarm, and other management functions required and provided by the protocol layer.
System services interface (SSI)	Obtains operating system services using a generic, portable interface that isolates the protocol layer from the underlying platform. Functions provided at this interface include initialization, task management, inter-task communication, timer management, memory management, message and queue management, date and time management, and resource checking.
HIT/SCT (Lower layer) Interface	Diameter lower interface uses sockets for TCP/SCTP transport.

Tight Coupling and Loose Coupling

Trillium software can be tightly or loosely coupled to other layers.

Table 4-1: Coupling options

Coupling	Description
Tight	Uses a synchronous interface at which a primitive is mapped to a direct function call into the destination layer. Control is immediately transferred to the destination layer in a nested fashion.
Loose	Uses an asynchronous interface at which a primitive is packed into a message and posted to the destination layer. Control immediately returns to the source layer. Later, the operating system schedules the destination layer with the posted message. The destination layer retrieves the primitive from the message and process it.

The system services interface is always tightly coupled**.** The other three interfaces can be either tightly or loosely coupled.

Trillium software is event-driven. It handles an event to completion, then handles the next event. Outgoing messages are generated in the same order in which the events arrived.

Diameter Primitives

Interaction between the Diameter software and the upper layer, lower layer, and layer manager takes place using a set of **primitive functions.** The primitives either initiate, or are the result of, the interactions between two layers. These primitives completely define the interaction between layers in the form of:

- Requests: service user to service provider.
- Indications: service provider to service user.
- Responses: service user to service provider.
- Confirmations: service provider to service user.

Figure 4-2 shows the exchange of primitives.

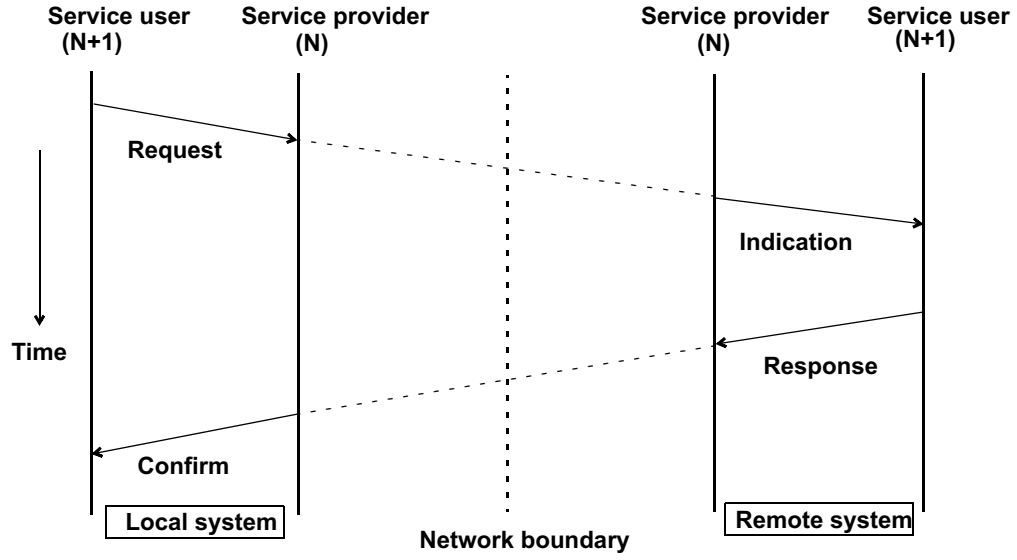


Figure 4-2: Exchange of primitives

Service Access Points

Information flows between layers across Service Access Points (SAPs). The standardized interface of primitives and SAPs allows layers to be defined independently. Modifications can be made to the peer-to-peer protocol of one layer without affecting any upper or lower layer protocol, if the requirements of the layer interface are met.

A detailed description of the primitives used and their associated procedures can be found in the *Diameter Base Protocol Service Definition*.

4.1.1 Portability

The software is written and delivered in the C programming language and can be compiled using any Kernighan and Ritchie or ANSI-compatible compiler.

Continuous Computing has compiled the software under native and cross compilers for many different processors. The software runs on various processors, operating systems and architectures. While the software has not been compiled under all compilers or run on all combinations of processors, operating systems and architectures, adherence to TAPA and its common coding and architectural standards ensures software portability.

Continuous Computing has ported the software to the processors and operating systems listed as follows. The list does not constitute a recommendation for a particular operating system or processor, nor is it a complete list of operating systems and processors onto which Trillium software has been ported. It serves only to show the flexibility and portability of Trillium software.

Processors:

Motorola 68xxx, SPARC xxx, Intel x86

Operating Systems:

Solaris, Windows NT, Linux, and VxWorks

Continuous Computing can provide support for other operating systems if required. Please contact Sales and Support for further information.

4.1.2 Debugging Capabilities

Trillium software provides extensive debugging capabilities. If terminal printing support is available, debug printing can be enabled, using compile-time and run-time options, for printing information about the data flow through the layer.

The software can trace function calls. Every function call contains a trace macro at its entry, which can be mapped to print information about function invocation. This is useful in embedded systems for which other symbolic debugging is not available. Message tracing can be enabled or disabled at runtime.

4.1.3 Error Checking

Trillium software provides extensive error checking and recovery mechanisms for making the software robust and enable it to deal with normal error conditions. The software enables the user to control the level of error checking using compile-time options.

Error checking is done for:

- Protocol
- Interface
- Input
- Output
- Resource

4.1.4 Run-Time Control

The layer manager interface controls many product capabilities at run time. The configuration of the software can be changed dynamically to respond to the needs. Management alarms and logging functions can be dynamically enabled and disabled. Also, where applicable, various protocol elements such as peers can be enabled, disabled, deleted, or reconfigured at run time.

4.1.5 Quality

Continuous Computing Corporation, has been certified and registered as compliant to the ISO 9001:2000 quality system requirements. This signifies that Continuous Computing's internal processes related to the development and distribution of communications software products and services for the next-generation network infrastructure successfully meet internationally recognized quality management systems requirements.

4.2 Product Features

This section describes the primitives that the Diameter software uses for interacting with the upper layer, lower layer, and layer manager. It also describes the procedures that the primitives enable.

4.2.1 Layer Interface Primitives

Several sets of primitives interface the Diameter software with the upper layer, lower layer, and layer manager.

4.2.1.1 Layer Manager Interface Primitives

The layer manager interface is used for managing layer resources. It provides run-time configuration, control, statistics, status, alarm, MIB access, and other management functions required and provided by the protocol layer.

Table 4-2 describes the primitives used for interfacing Diameter with the layer manager.

Table 4-2: Layer manager interface primitives

Primitive	Description
Configuration request Configuration confirm	Configures the software resources.
Peer Add/Delete request	Manages peer configuration in Diameter layer.
AVP/DM dictionary configuration request	Configures the application specific AVP/DM dictionary in Diameter layer.
Status request Status confirm	Gathers information for determining the current state of the software.
Realm Routing Table Add/ Delete request	Manages Realm routing table in Diameter layer.
Statistics request Statistics confirm	Gathers statistics related to number of Diameter messages sent/received, number of error messages received and number of bytes sent/ received.
Control request Control confirm	Controls the software resources.
Trace indication	Provides trace information by logging incoming and outgoing PDUs.
Unsolicited status indication	Provides information indicating a change in the software status.

Table 4-2: Layer manager interface primitives

Primitive	Description
Probe request Probe confirm	Provides the in-memory tables information in the Diameter layer, to Layer Manager using this primitive. This includes Realm, Peer, AVP dictionary, and DM dictionary tables.

Refer to the *Diameter Base Protocol Service Definition* for details.

4.2.1.2 AQU Interface Primitives

Table 4-3 describes the primitives for interfacing Diameter with the Diameter Application (upper layer).

Table 4-3: AQU interface primitives

Primitive	Description
Bind request	Registers the service user with service provider.
Bind confirm	Confirms the registration of the service user with the service provider.
Unbind request	Deregisters the service user with the service provider.
Diameter message request	The service user request the service provider to send a Diameter request message to peer.
Diameter message response	The service user request the service provider to send a Diameter answer message to peer.
Diameter message indication	The service provider informs the service user of an incoming Diameter request message.
Diameter message confirm	The service provider informs the service user of an incoming Diameter answer message.
Diameter remove transaction details request	The service user requests the service provider to remove the Diameter msgs identified by the transaction Ids, which are stored in the Diameter layer.
Diameter remove transaction details confirm	The Service provider confirms the removal of transactions requested by service user.

Table 4-3: AQU interface primitives

Primitive	Description
Diameter error indication	The service provider informs the service user of the error in Diameter protocol message handling.
Diameter peer down indication	The service provider informs the service user, the status of the messages sent to the peer, which is currently not available.
Diameter peer up indication	The service provider informs the service user, the status of the peer which is currently available.
Diameter flow control indication	The service provider informs the service user of the traffic flow conditions.

Refer to the *AQU Interface Service Definition* for details.

4.2.1.3 HIT Interface primitives

Table 4-4 describes the primitives used to interface Diameter with lower layer (TUCL) for TCP transport.

Table 4-4: HIT Interface primitives

Primitive	Description
Bind request	Registers the service user with the service provider.
Bind confirm	Confirms the registration of the service user with the service provider.
Unbind request	Deregisters the service user with the service provider.
Server open req	The service user requests the service provider to open a TCP/TLS or TCP/IPSec server on the given address and the port.
Client connection request	The service user requests the service provider to open a connection to the peer on the given address and port.

Table 4-4: HIT Interface primitives

Primitive	Description
Connection indication	The service provider indicates to the service user of the incoming TCP/TLS or TCP/IPSec connection.
Connection response	The service user accepts the incoming connection to the service provider.
Connection confirm	The service provider informs the service user of the connection acceptance.
Flow control indication	The service provider informs the service user of the traffic flow conditions.
Data request	The service user transfers the data on the established connection to the service provider.
Data indication	The service provider transfers the data to the service user.
Disconnect request	The service user requests the service provider to release the connection to the peer.
Disconnect indication	The service provider indicates the release of connection with the peer to the service user.
Disconnect confirm	The service provider confirms the service user of the connection release.

4.2.1.4 SCT Interface primitives

Describes the primitives used to interface Diameter with lower layer (SCTP) for SCTP transport.

Table 4-5: SCT Interface primitives

Primitive	Description
Bind request	Registers the service user with the service provider.
Bind confirm	Confirms the registration of the service user with the service provider.
Unbind request	Deregisters the service user with the service provider.
End point open request	The service user request the service provider to open an end point.

Table 4-5: SCT Interface primitives

Primitive	Description
End point open confirm	The service provider confirms the opening of an end point to the service user.
End point close request	The service user requests the service provider to close an end point.
End point close confirm	The service provider confirms the closing of an end point to the service user.
Association open request	The service user requests the service provider to open an association.
Association open confirm	The service provider confirms the opening of an association to the service user.
Association open indication	The service provider indicates the service user of an incoming association open request.
Association open confirm	The service user informs the service provider, its acceptance for the incoming association.
Association termination request	The service user requests the service provider to close an association.
Association termination confirm	The service provider confirms the closing of association to the service user.
Association termination indication	The service provider informs the service user of closing of an association.
Data request	The service user transfers the data on the established association to the service provider.
Data indication	The service provider transfers the data to the service user.
Flow control indication	The service provider informs the service user of the traffic flow conditions.

4.2.2 Configuration

Configuration procedures set parameters and environment options for the layer.

4.2.2.1 General Configuration

General configuration sets parameters for the entire Diameter layer. The following parameters can be configured for Diameter.

- Number of Upper SAPs.
- Number of Lower SAPs.
- Pending Message queue size.
  - Please refer section 5.5.4 in RFC 3588 [8], for the description of pending message queue.
- Firmware Revision.
  - Please refer section 5.3.4 in RFC 3588 [8] for the description and usage of this.
- Origin State ID.
  - Please refer sections 8.6 and 8.16 in RFC 3588 [8], for the description and usage of this.
- Maximum number of Peers.
- Maximum number of Realms.
- HostId.
  - It is of type Diameter Identity. For its definition and usage please refer Sections 6.1 and 6.3 in RFC 3588 [8].
- Local Real.
  - Its definition and usage is described in Sections 6.1.1 and 6.4 in RFC 3588 [8].
- Product Name.
  - Please refer Section 5.3.7 in RFC 3588[8] for its description and usage.
- Timer Resolution.
- Number of simultaneous worker threads for Encoder/Decoder.
  - Diameter supports multiple threads for encoding and decoding. This parameter specifies the number if simultaneous threads utilized for encoding/decoding operations. If it is set to “0”, then threads are not created for encoding/decoding operations.
- Memory Thresholds.

4.2.2.2 Protocol Configuration

Protocol Configuration sets the protocol related parameters for the entire Diameter layer. The following parameters can be configured for the Diameter:

- Watch dog timer.
- Peer State timer.
  - It is used to wait for the peer state events in Peer State machine implementation. Peer State machine is described in section 5.6 in RFC 3588 [8].
- Reconnection timer.
  - It is used to reconnect to the Peers which are in CLOSED state. It defines the time intervals at which Diameter tries for reconnection.
- Peer discovery timer.
  - It is used for dynamically discovering the peers. It defines the time intervals at which Diameter tries to discover the peers.
- Burst break timer, for controlling the number of msgs sent to Lower layer simultaneously.
- Burst Break count, number of msgs can be sent to lower layer before taking a break.
- Failover Option (ON/OFF).
- Peer discovery type, currently SLPv2 only supported.
- Security type (TLS/IPSec or both).
- Max connection retry.
- Local policy.
  - It is used to filter the incoming connections coming from unknown peers (neither configured manually nor discovered dynamically). Diameter either allow or deny those connections based on this parameter.
- Transport parameters.
- Peer discovery Information.
  - This information is used by Diameter for discovering the peers dynamically.
- Peer configuration parameters.
- Realm configuration parameters.
- Vendor ID.

4.2.2.3 Upper SAP Configuration

The upper SAP configuration procedure allocates and configures an upper SAP. Typically there is one upper SAP per application type; for example: Sh, Rf, Ro and Cx/Dx have one upper SAP each, with the Diameter layer. If the SAP is already allocated, this procedure can be used dynamically to reconfigure the SAP parameters. The following parameters can be configured for each upper SAP:

- SAP ID.
- Selector.
- Priority.
- Route.
- Memory region and pool.
- Vendor ID.
  - Please refer to Section 4.1.1 in RFC 3588 [8] for description and usage on this.
- Application ID.
- Supported Vendor ID.
  - Please refer to Section 5.3.6 in RFC 3588 [8] for description and usage on this.
- Application Type (Accounting/Authorization).

4.2.2.4 Lower SAP configuration

The lower SAP configuration procedure allocates and configures a lower SAP. Typically there is one lower SAP per transport type. One for TCP transport and another for SCTP transport option. If the SAP is already allocated, this procedure can be used dynamically to reconfigure the SAP parameters. The following parameters can be configured for each lower SAP:

- Service user SAP ID.
- Service provider SAP ID.
- Selector.
- Priority.
- Route.
- Memory region and pool.
- Destination processor ID.
- Destination Entity ID.
- Destination Instance ID.
- Bind timer value.
- Bind retry count.
- Transport type.
- Max server retry count.
    - Diameter opens a server port to listen the incoming connections. This parameter specifies the maximum number of times tried to open the server port using the services of the provider (SCTP/TCP).
- Server Configuration parameters (TCP + IPSec and TCP + TLS or SCTP + IPSec).

4.2.2.5 Application Specific Diameter Msg Configuration

Application specific Diameter configuration is used to configure the Diameter Msg dictionary per application. The following parameters can be configured for each application specific Diameter Msg configuration:

- Application ID.
- Number of Diameter commands.
- Diameter Msg properties.

Note: This configuration allows the Diameter layer to be used for any 3GPP Diameter application interface.

4.2.2.6 Application Specific AVP configuration

Application specific AVP configuration is used to configure the AVP dictionary per application. The following parameters can be configured for each application specific AVP configuration:

- Application ID.
- Number of AVPs.
- AVP properties.

Note: This configuration allows the Diameter layer to be used for any 3GPP Diameter application interface.

4.2.3 Status

Status information indicates the current state of the software and can be used in software debugging. The layer manager can gather the status information at anytime, without changing any of the examined information, allowing status retrieval for these product resources:

- SAP state for each lower SAP.
- SAP state for each upper SAP.
- System ID status information, consisting of a text string that identifies the part number and version number of the software is use.
- State of each peer (for both manually configured and dynamically discovered).

4.2.4 Statistics

Statistics are gathered to measure the Diameter software messages. The layer manager can gather statistics information at any time, The statistics counters can be reset, if desired, after collecting the statistics.

These statistics are maintained:

- Erroneous messages received.
- Diameter messages transmitted and received (both Diameter base protocol and application messages) per message and application ID type.
- Number of bytes transmitted and received.

4.2.5 Alarms

Alarms are generated to indicate abnormal status changes of the software or SAP. Alarms can also be advisory in nature. Alarms are sent to the Layer manager as unsolicited status indications when a condition requiring attention is detected. The generation of alarms can be enabled or disabled dynamically.

Some Alarms:

- Invalid SAP ID.
- Invalid SAP state.
- Invalid parameter value.
- Memory allocation failure.
- Bind failure.
- Opening of Server failure.
- Peer is connected successfully.
- Peer down.
- Unexpected AVP detected in the message received from the peer.
- Unexpected Diameter message received from the peer.

4.2.6 Trace Generation

Traces are generated whenever Diameter receives or transmits a Protocol Data Unit (PDU). Trace generation can be enabled or disabled dynamically.

4.2.7 Control

Control functions activate and deactivate the layer resources. The layer manager can invoke the control functions anytime. The control feature of the layer manager allows the flexibility to dynamically modify the characteristics of a stack or a resource.

For example, by using control functions:

- Turn on or off the tracing at runtime. Once tracing is turned on, it provides the Service Data Units (SDUs), which are transmitted or received by the layer, to the layer manager. This information can be used for debugging.
- Turn on or off the alarms at runtime. When enabled, alarms provide useful information regarding the state of the protocol stack.
- Turn on or off the different debug levels to aid product-specific system integration.
- Start the bind process on the lower SAP.
- Unbind the Lower SAP.
- Start the initiation of connection to the manually configured peers.
- Delete the SAPs (both Upper and Lower).
- Deletion of the peer(s) from peer table.
- Deletion of the realm(s) from realm table.
- Shutdown the layer.

5

Licensing Options

For licensing purposes, Diameter is identified by a marketing part number 1000349. Continuous Computing supplies a base option with each license to provide core protocol functionality.

Table 5-1: Licensing Options

Part Number	Option	Description
1000349	--	Base Option includes Diameter Base Protocol and TUCL.

Note: Base option does not include TUCL and TLS support. For complete TLS support the following product must be licensed in addition to the Diameter Base option.

Part Number	Option	Description
1000158	01	TCP/UDP Convergence Layer-TLS Support.

6

Memory Size

Continuous Computing Corporation, the leading provider of communications software solutions, offers the industry's leading performer Diameter software solution for IMS Technology.

The Diameter protocol is a cornerstone technology that provides the key component of the IMS infrastructure. Continuous Computing's Diameter software is scalable and supports distributed system architectures. Continuous Computing's Diameter protocol software offers maximum portability and flexibility.

Continuous Computing's Diameter protocol software solution enables communications equipment suppliers to deliver products that meet the requirements of IP Multimedia Subsystem networks.

6.1 Memory Size for Diameter Protocol Implementation

6.1.1 Code and Static Data Size

The code size is the number of bytes of memory needed for the executable code. The code size includes all function calls to system services, layer manager, and the upper and lower layers, but does not include the size of the actual code provided within these functions.

The code size depends on the options delivered, compiler, linker, locator, memory model, and whether delivered features—such as error checking, management capabilities, and protocol capabilities—are used. The code size can be determined from the software link map.

Table 6-1 shows the code size and the compile conditions for Diameter Base Protocol.

Table 6-1: Code size — Diameter

Type	Condition
Product	Diameter version 1.2
Product options	-DAQ -DAQ_TLS -DAQ_SCTP -DAQ_PEER_DISCOVERY -DLCAQU -DLCAQUIAQU -DLCLAQ -DLCAQMILAQ -DLCSCT
Compiler	Microtek 68K Cross compiler (m68-coff-gcc)
Compiler options	-m68040 -Os -Wall -Wno-comment -pipe -Wshadow -Wcast-qual -Wstrict-prototypes -Wmissing-prototypes
Processor	Pentium
Total code size (in bytes)	**230930**

6.2 Static Data Size

The static data size is the number of bytes of memory needed for:

- Initialized variables and structures (for example, state matrices or strings)
- Uninitialized variables and structures (for example, anchors for control points and SAPs)

Static data is allocated at compile time and represents the global variables and structures used by the software. Static data does not include any structures allocated at run time.

The static data size depends on the options delivered, compiler, linker, and memory model. The static data size is determined from the software link map.

A sample compile under the following conditions, tightly coupled with error checking and trace printing disabled, yields the code size in Table 6-2.

Table 6-2: Static Data Size

Type	Condition
Product	Diameter, version 1.2
Product option	-DAQ -DAQ_TLS -DAQ_SCTP -DAQ_PEER_DISCOVERY -DLCAQU -DLCAQUIAQU -DLCLAQ -DLCAQMILAQ -DLCSCT
Compiler	Microtek 68K Cross compiler (m68-coff-gcc)
Compiler options	-m68040 -Os -Wall -Wno-comment -pipe -Wshadow -Wcast-qual -Wstrict-prototypes -Wmissing-prototypes
Processor	Pentium
Total static data size (in bytes)	**1582**

6.3 Dynamic Data Size

The dynamic data size depends on the maximum allowable configuration (A), size of structures allocated for the configurable parameters (B), maximum number of messages to be stored simultaneously (C), and the size of a message buffer (D).

The dynamic data size can be estimated by using the formula:

(A × B) + (C × D).

The message buffer size depends on the system services. The maximum allowable configuration and dynamically allocated structure size are given in Section 6.3.1 and Section 6.3.2 respectively.

6.3.1 Maximum Allowable Configuration

The maximum allowable configurations described in this section represent the Diameter software capacity. Actual maximums may be smaller, depending on the limitations of the system to which this software is ported, such as memory availability.

Table 6-3: Maximum Allowable Configuration

Entity	Maximum Number
Upper SAPs	32767
Lower SAPs	2
Number of peers	255
Number of Realms	255

Note: For testing purpose, the maximum number of lower SAPs allowed is 4. Otherwise in real life scenarios, the maximum allowable configuration for Lower SAPs is 2.

Lower SAP 0 - For Real TUCL

Lower SAP 1 - For Real SCTP

Lower SAP 2 - For Dummy TUCL

Lower SAP 3 - For Dummy SCTP

Dummy TUCL and Dummy SCTP are used only for internal testing to simulate the scenarios. For dummy TUCL and dummy SCTP, there is no lower layer, instead the request/response is loop backed to the same node.

6.3.2 Dynamically Allocated Structure Size

This section describes the sizes of dynamically allocated structures such as control points or SAPs.

Table 6-4: Dynamically allocated structure size

Type	Bytes
Global Control Block (in Bytes)	2766
Each upper SAP (in Bytes)	92
Each lower SAP (in Bytes)	1124
Peer Control Block (in Bytes)	780
Realm control block (in Bytes)	328
Message size in the pending message queue (in Bytes)	48

Appendix A

Addendum

This section is an addendum to version 1.22a of the Diameter Base Protocol Functional Specification.

Note: *The material in this addendum is integrated into the body of the document and noted below.*

Table A-1: List of Changes for this Addendum

Description of Change	Section Reference
Added new specifications in the protocol conformance section.	Section 3.1, "Standards".

References

Refer to the following documents for additional information.

1. *Diameter Base Protocol Service Definition,* Continuous Computing Corporation, (p/n 1092349).
2. *AQU Interface Service Definition*, Continuous Computing Corporation, (p/n 1100073).
3. *HIT Interface Service Definition*, Continuous Computing Corporation, (p/n 1100031).
4. *SCT Interface Service Definition*, Continuous Computing Corporation, (p/n 1100036).
5. *Diameter Base Protocol Training Course,* Continuous Computing Corporation, (p/n 1095349).
6. *Diameter Base Protocol Software Test Sample*, Continuous Computing Corporation, (p/n 1094349).
7. *Diameter Base Protocol Porting Guide*, Continuous Computing Corporation, (p/n 1093349).
8. *Diameter Base Protocol, (RFC 3588)*.