

Diameter Maintenance and Extensions
(DIME)
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

J. Korhonen
Renesas Mobile
H. Tschofenig, Ed.
Nokia Siemens Networks
February 25, 2013

The Diameter Overload Control Application (DOCA)
draft-korhonen-dime-ovl-01.txt

Abstract

This specification documents a Diameter Overload Control Application (DOCA), which uses the normal Diameter application approach for the capability negotiation, propagation and management of Diameter overload control information between Diameter nodes.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Requirements	3
3. DOCA Overview	4
4. DOCA Commands	5
5. Attribute Value Pairs	6
5.1. OC-Information AVP	6
5.2. OC-Scope AVP	7
5.3. OC-Applications AVP	8
5.4. OC-Action AVP	9
5.5. OC-Algorithm AVP	9
5.6. OC-Level AVP	10
5.7. OC-Utilization AVP	11
5.8. OC-Tocl AVP	11
5.9. OC-Sending-Rate AVP	11
5.10. OC-Best-Before AVP	12
5.11. OC-Origin AVP	12
5.12. OC-Priority AVP	12
5.13. Attribute Value Pair flag rules	13
6. Transport Considerations	13
7. Examples	14
8. IANA Considerations	15
8.1. Application Identifiers	15
8.2. SCTP Payload Protocol Identifier	15
8.3. Command Codes	15
8.4. AVP Codes	15
8.5. Result-Code Values	15
8.6. New Registries	16
9. Security Considerations	16
10. Acknowledgements	16
11. References	17
11.1. Normative References	17
11.2. Informative References	17
Appendix A. Design Justification	17
Authors' Addresses	18

1. Introduction

The existing toolbox offered by the Diameter Base Protocol [RFC6733] to prevent and recover from signaling overload situations is rather limited. Apart from out-of-band altering of the transport connection congestion control behavior or other non-standard application level throttling, the protocol error `DIAMETER_TOO_BUSY`, the permanent error `DIAMETER_UNABLE_TO_COMPLY` (for some unspecified reason) and the Disconnect-Cause Attribute Value Pair (AVP) code `BUSY` or `DO_NOT_WANT_TO_TALK_TO_YOU` are more or less all there is. Unfortunately, the mentioned three indications are coarse, concern one peer connection at a time or lack detailed information for problem diagnosis and mitigation. They also treat all applications in a single Diameter node (identified by a single `DiameterIdentity`) as a lump. There is no way communicate any kind of grouping of applications or what is the scope/partitioning of the delivered information. Furthermore, there is no way to signal when the overload situation is over. The request initiator and forwarders are therefore forced to keep re-submitting their messages to determine whether the situation has changed.

The situation is further complicated by the hop-by-hop nature of Diameter deployments. This makes the propagation of possible overload situation information non-trivial, even for existing protocol errors since every intermediate Diameter node is allowed to react to the error situation. Either the information is never propagated to the originator of the request or it takes an unacceptable long time.

A problem statement of overload control for Diameter and requirements are documented in [I-D.ietf-dime-overload-reqs]. This specification describes a solution to the Diameter overload Diameter Overload Control Application (DOCA), which fulfills the requirements of [I-D.ietf-dime-overload-reqs] and defines a Diameter application to convey overload information between Diameter nodes.

Note: The recent publication of [I-D.campbell-dime-overload-data-analysis] illustrates the overload information data model and the design space. As the working group makes progress in deciding about specific features this document will be updated accordingly.

2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. DOCA Overview

Any the DOCA capable Diameter node MAY initiate a DOCA-Report-Request at any given time. The receiver of the DOCA-Report-Request acknowledges with a DOCA-Report-Answer and includes the Result-Code AVP indicating whether it could honor the action/report in the request. The DOCA-Report-Answer SHOULD also piggyback overload control information.

A DOCA client MUST set the Auth-Session-State AVP to the value NO_STATE_MAINTAINED and SHOULD include the OC-Information AVP with overload information into the DOCA-Report-Request, if available. The DOCA-Report-Response message MUST contain the Auth-Session-State AVP set to value NO_STATE_MAINTAINED.

Note that information exchanges regarding various DOCA related timers serve only as a hint since they cannot be enforced. Consequently, care should be taken not to send DOCA-Report-Requests too frequently.

When a Diameter node receives overload control information and is also requested to act on it, the DOCA functionality is applied to all specified applications within a given scope. How the Diameter node accomplishes the node wide DOCA action enforcement is implementation specific.

When a Diameter node receives (interim) overload information but the overload condition has not exceeded a certain threshold, then the receiver is not required to act based on the received information. However, it is RECOMMENDED that the receiver makes proactive actions to avoid entering the overload condition based on the newly received overload information.

There may be zero or more intermediate Diameter agents on the path between the DOCA client and the DOCA server. Understanding the DOCA functionality is not expected from relays and redirect agents. A Diameter proxy, which obviously understands the DOCA application, MAY inspect the DOCA related AVPs in the DOCA-Report-Request/Answer message pair and depending on the value of the OC-Scope AVP (see Section 5.2) inject its own information. A proxy is always RECOMMENDED to react according to the overload information when it comes to, for example, peer selection and traffic throttling.

When a Diameter agent receives overload control information and is also requested to act on it, the DOCA functionality is applied to all specified applications within a given scope. How the Diameter agent accomplishes the node wide DOCA action enforcement is implementation specific.

4. DOCA Commands

The DOCA-Report-Request (DRR) message is used to report overload condition information. The message can be originated as a result of emerging overload condition or as a periodic unsolicited report.

```
<DOCA-Report-Request> ::= < Diameter Header: TBD2, REQ, PXY >
    < Session-Id >
    { Auth-Application-Id }
    { Origin-Host }
    { Origin-Realm }
    { Destination-Realm }
    { Auth-Request-Type }
    { Destination-Host }
    [ Auth-Session-State ]
    * [ Class ]
    [ Origin-State-Id ]
    * [ Proxy-Info ]
    * [ Route-Record ]

    { OC-Scope }
    [ OC-Algorithm ]
    [ OC-Action ]
    [ OC-Tocl ]
    [ OC-Applications ]
    * [ OC-Information ]

    * [ AVP ]
```

The DOCA-Report-Answer (DRA) message is used as a response to the DOCA-Report-Request. The message MAY piggyback overload condition information in order to avoid unnecessary DOCA-Report-Request messages to the reverse direction.

```
<DOCA-Report-Answer> ::= < Diameter Header: TBD2, PXY >
    < Session-Id >
    { Result-Code }
    { Origin-Host }
    { Origin-Realm }
    [ Auth-Session-State ]
    * [ Class ]
    [ Error-Message ]
    [ Error-Reporting-Host ]
    [ Failed-AVP ]
    [ Origin-State-Id ]
    * [ Redirect-Host ]
    [ Redirect-Host-Usage ]
    [ Redirect-Max-Cache-Time ]
    * [ Proxy-Info ]

    { OC-Scope }
    [ OC-Action ]
    * [ OC-Information ]

    * [ AVP ]
```

5. Attribute Value Pairs

5.1. OC-Information AVP

The OC-Information AVP (AVP Code TBD3) is of type Grouped and contains a set AVPs that identify the source of the overload control information (the OC-Origin AVP), the overload information itself and which applications the information concerns.

```
OC-Information ::= < AVP Header: TBD3 >
    { OC-Origin }
    { OC-Best-Before }
    [ OC-Level ]
    [ OC-Algorithm ]
    [ OC-Sending-Rate ]
    [ Vendor-Id ]
    [ OC-Applications ]
    [ Product-Name ]
    [ OC-Utilization ]
    [ OC-Priority ]
    * [ AVP ]
```

Diameter proxies on path MAY add one or more OC-Information AVPs into the DOCA-Report-Request/answer messages.

5.2. OC-Scope AVP

The OC-Scope (AVP Code TBD4) is of type Unsigned32 and contains the scope where and concerning what the overload control information can be injected. The OC-Scope is formatted as a vector of scope flag bits. The following scopes are supported:

Host scope (0x00000001)

The OC-Information AVP concerns only a single host within a realm (which internally MAY represent of pool).

Realm scope (0x00000002)

The OC-Information AVP concerns a realm. No specific hosts are identified.

Only origin realm (0x00000004)

The OC-Information AVP can only be included by a Diameter node on the path that has the same Origin-Realm as the DOCA client.

Application information (0x00010000)

The OC-Information AVP MAY contain application related information (the OC-Applications AVP).

Node utilization information (0x00020000)

The OC-Information AVP MAY contain node wide load related information (the OC-Utilization AVP).

Application priorities (0x00040000)

The OC-Information AVP SHOULD priority information (the OC-Priority AVP) so when the overload condition is on, Diameter nodes are able to prioritize between different applications, for example, when dropping or throttling messages.

Any other value is reserved.

A scope is active when a corresponding flag is set in the OC-Scope AVP. During the initialization state a DOCA client includes those scopes it supports and is interested in. A DOCA server then returns the scope that it has in common with the DOCA client (and intends to use). The common scopes are then used during the established state. Note that some scope combinations make little sense while still being valid. The general guide when multiple scopes collide is that the

least restrictive wins.

A sender of the overload information MUST adhere to the scope it announces regarding the information it itself sends.

If a DOCA server does not have a common scope with a DOCA client or the DOCA server cannot agree on one based on a local policy, then the DOCA server MUST send the DOCA-Report-Answer indicating an error and set the Result-Code to the DIAMETER_NO_COMMON_SCOPE value.

5.3. OC-Applications AVP

The OC-Applications (AVP Code TBD5) is of type Grouped and contains a list of Application-IDs of interest when found in the DOCA-Report-Request/Answer command main level and meant to be used during the initialization state to agree on the common set of supported applications of monitoring interest. When used within the OC-Information AVP, the OC-Applications AVP identify those applications the overload information concerns. The OC-Applications AVPs on the command main level and inside the OC-Information AVP MUST NOT have conflicting views of the applications of interest. However, the OC-Applications AVP can be seen as a superset of applications i.e., not all applications of interest need to be included every time into the OC-Information AVP.

```
OC-Applications ::= < AVP Header: TBD3 >
                  * [ Auth-Application-Id ]
                  * [ Acct-Application-Id ]
                  * [ Vendor-Specific-Application-Id ]
                  * [ AVP ]
```

The absence of the OC-Applications AVP indicates the Diameter node has no specific preference or interest in specific applications. The overload information is then signaled as concerning the whole Diameter node. This default behavior is useful when the DOCA does not maintain session state. If there are no common applications, then the DOCA-Report-Answer MUST contain the Result-Code with the DIAMETER_NO_COMMON_APPLICATION value.

When the DOCA maintains state, there is no need to include the OC-Applications AVP into the DOCA-Report-Request/Answer command main level after the initial message exchange. The agreed common set of application is expected to be known by both DOCA client and server throughout the session lifetime.

5.4. OC-Action AVP

The OC-Action (AVP Code TBD6) is of type OctetString and size of one octet. The octet has the following three possible values:

Start (1)

Signals the start of the overload condition. This implies the receiver is requested to act according to the information found in the OC-Information.

Stop (2)

Signals the end of the overload condition.

Interim (3)

Updates the overload information. The interim can be sent during the overload condition or during the normal condition. This is the default value.

Any other value is reserved.

5.5. OC-Algorithm AVP

The OC-Algorithm (AVP Code TBD7) is of type Unsigned32. The contains supported 'algorithms' to mitigate the overload condition. The OC-Algorithm AVP is formatted as a vector of algorithm flag bits. The following 'algorithms' are supported:

Drop (0x00000001)

Messages are plain dropped. It is RECOMMENDED to drop messages selectively based, for example, on application priorities. This is the default algorithm.

Throttle (0x00000002)

The message sending rate is according to the OC-Sending-Rate AVP.

Prioritize (0x00000004)

Apply priorities among applications and the other used means for holding traffic.

Any other value is reserved.

The 'algorithms' are only applied at a Diameter node when the

overload condition has been signaled.

During the initialization state a DOCA client includes those algorithms it supports and is interested in. A DOCA server then returns the algorithm that it has in common with the DOCA client (and intends to use). One or more common algorithms are then used during the established state.

If a DOCA server does not have a common algorithm with a DOCA client or the DOCA server cannot agree on one based on a local policy, then the DOCA server **MUST** send the DOCA-Report-Answer indicating an error and set the Result-Code to the DIAMETER_NO_COMMON_ALGORITHM value.

5.6. OC-Level AVP

The OC-Level (AVP Code TBD8) is of type OctetString and size of one octet. The octet has the following five possible values:

Normal (1)

Everything is in control. Meaningful only when the OC-Action is set to 'Interim' since when the overload condition level is considered normal, the overload condition **SHOULD** be stopped. This is the default value.

Raising (2)

There is a sign of increasing load.

Alarming (3)

The overload condition is reaching the level where quick measures **SHOULD** be done to mitigate the overload condition.

Panic (4)

The overload condition is severe. Apply any measure to mitigate the overload condition but still allowed to send messages.

Hold (5)

Do not send any messages, please. When this level is signaled, the OC-Best-Before time **SHOULD NOT** be respected but an explicit overload condition stop has to be received (with an exception the Diameter node realizes its other end has rebooted or otherwise lost its state).

Switch servers (6)

Do not talk to me again. When this level is signaled, the DOCA client MUST switch to an alternative server.

Any other value is reserved.

If the receiver cannot agree on or does not understand the OC-Level AVP value, the an error MUST be returned with the Result-Code AVP set to the value DIAMETER_INVALID_AVP_VALUE and the Failed-AVP AVP containing the OC-Level AVP.

5.7. OC-Utilization AVP

The OC-Utilization (AVP Code TBD9) is of type Float32 and tells the overall utilization level percentage of the Diameter node. Values between 0.0 to 100.0 are valid.

5.8. OC-Tocl AVP

The OC-Tocl (AVP Code TBD10) is of type Unsigned32 and tells the Tolc timer value in milliseconds. This timer defines the interval for sending periodic DOCA-Report-Request messages with the OC-Action AVP set to 'Interim'. The value of zero (0) means no periodic DOCA-Report-Request messages are sent or desired. The default value is 120000.

The OC-Tocl AVP can be considered as a hint for a desired sending rate of subsequent messages.

If a DOCA server find the Tocl value proposed by a DOCA client either too small (i.e. too frequent periodic messages) or too big (i.e. too seldom periodic messages), then the DOCA server MUST send the DOCA-Report-Answer indicating an error and set the Result-Code either to the DIAMETER_TOCL_TOO_SMALL or DIAMETER_TOCL_TOO_BIG value.

5.9. OC-Sending-Rate AVP

The OC-Sending-Rate (AVP Code TBD11) is of type Float32 and tells the the maximum Diameter message sending rate per second the sender of this information wishes to receive Diameter messages. Only positive values are valid. A value of zero (0.0) of the absence of this AVP means the information sender has no specific rate preference.

If a DOCA server finds the sending rate value proposed by a DOCA client too big (i.e., too frequent periodic messages), then the DOCA server MUST send the DOCA-Report-Answer indicating an error and set

the Result-Code to the DIAMETER_RATE_TOO_BIG value.

5.10. OC-Best-Before AVP

The OC-Best-Before (AVP Code TBD12) is of type Time and tells the expiration time/date for the information received in the OC-Information. For example, when the overload condition is on, the expiration of the 'best before' timer causes the same as receiving a DOCA-Report-Request/Answer with the OC-Action set to 'Stop'.

[Editor's note: to be decided whether a duration timer is a better measure. Using Time has the assumptions nodes have actually clocks that a running approximately same time.]

5.11. OC-Origin AVP

The OC-Origin (AVP Code TBD13) is of type DiameterIdentity and tells the identity of the Diameter node that originated included the overload control information. Both host and realm information MUST be included in the OC-Origin AVP. Note, if the OC-Scope AVP indicates only a realm wide scope for the overload information, then the realm part of the OC-Origin AVP is meaningful and the host information only serves as an additional information of the representative for the realm wide information.

5.12. OC-Priority AVP

The OC-Priority (AVP Code TBD14) is of type Unsigned32 and defines the priority level. The value of 0x00000000 is the highest priority and the value of 0xffffffff is the lowest priority. The absence of the OC-Priority AVP means there is not specific priority level defined and the priority SHOULD be considered as the lowest possible.

When used within the OC-Information grouped AVP, the OC-Priority AVP defines the priority for the listed applications within the OC-Applications AVP.

5.13. Attribute Value Pair flag rules

				+-----+ AVP flag rules +-----+	
Attribute Name	AVP Code	Section Defined	Value Type	MUST	NOT
OC-Information	TBD3	x.x	Grouped	M	V
OC-Scope	TBD4	x.x	Unsigned32	M	V
OC-Application	TBD5	x.x	Grouped	M	V
OC-Action	TBD6	x.x	OctetString	M	V
OC-Algorithm	TBD7	x.x	Unsigned32	M	V
OC-Level	TBD8	x.x	OctetString	M	V
OC-Utilization	TBD9	x.x	Float32	M	V
OC-Tocl	TBD10	x.x	Unsigned32	M	V
OC-Sending-Rate	TBD11	x.x	Float32	M	V
OC-Best-Before	TBD12	x.x	Time	M	V
OC-Origin	TBD13	x.x	DiameterIdentity	M	V
OC-Priority	TBD14	x.x	Unsigned32	M	V

6. Transport Considerations

In case of Stream Control Transmission Protocol (SCTP) transport, the DOCA application is RECOMMENDED to mark its Diameter packets using the DOCA defined SCTP Payload Protocol Identifier (PPID) TBD1. The PPID MAY be used by intermediating network nodes or agents to peek into SCTP message and find out that this is about overload control. Such information can be used for prioritizing SCTP packet handling as an example.

7. Examples

Consider the following simplified scenario shown in Figure 1 where two servers are connected to a proxy. All three nodes understand the DOCA application. These three nodes belong to the same administrative domain and the operator decided that he wants to hide the Diameter topology of his own network. Consequently, aggregate information is provided by the proxy for any Diameter overload message exchange. The Diameter client also supports the DOCA application. Between the client and the Diameter proxy we assume an arbitrary Diameter network that passes Diameter messages back and forth.

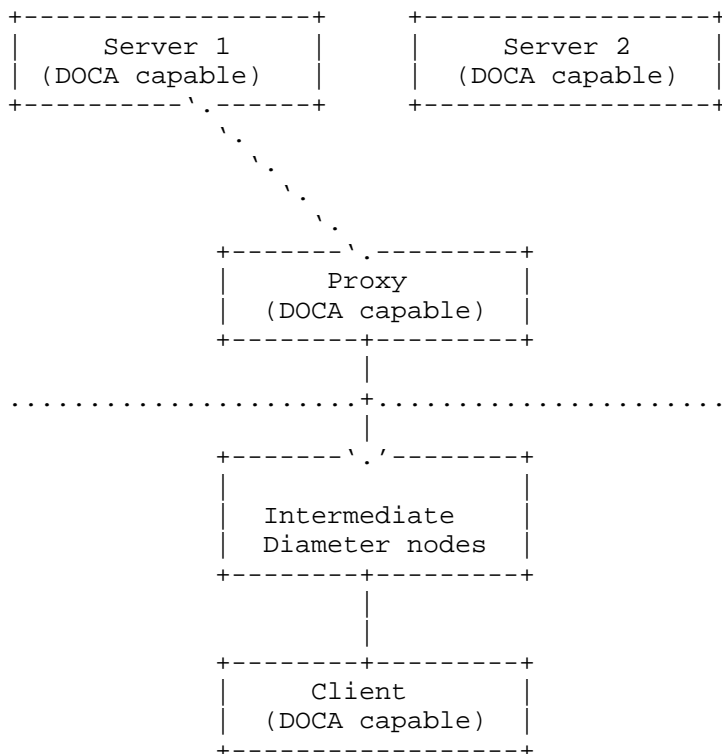


Figure 1

Let us assume that the DOCA exchange is initiated by server 1 who determines that the load situation increases. It sends a DOCA-Report-Request message (with piggybacked overload information) towards the client. The message also instructs the client to reduce

it's sending rate. The proxy, who receives the DOCA-Report-Request decides to change the included OC-Origin information and forwards the request to the client.

When the client receives the DOCA-Report-Request message is processes the content, evaluates the overload information content and reacts accordingly, and returns a DOCA-Report-Answer message back to acknowledge the receipt.

Alternatively, let us assume that the proxy does not forward the message but instead terminates the DOCA-Report-Request received from Server 1. It instead decides to route traffic to the backup server, Server 2. In this case the entire process was transparent for the client.

8. IANA Considerations

8.1. Application Identifiers

This specification reserves a new Diameter Application-ID TBD14 for the Diameter Overload Control Application (DOCA) from the 'Authentication, Authorization, and Accounting (AAA) Parameters' Application IDs registry.

8.2. SCTP Payload Protocol Identifier

Section 6 reserves a new SCTP Payload Protocol Identifier for the DOCA application usage. The value is reserved from the existing SCTP Payload Protocol Identifiers registry.

8.3. Command Codes

Two command codes are defined in Section 4. The DOCA-Report-Request Command Code is TBD and the DOCA-Report-Answer Command Code is TBD. Both are allocated from the 'Authentication, Authorization, and Accounting (AAA) Parameters' Command Codes registry.

8.4. AVP Codes

New AVPs defined by this specification are listed in Section 5. All AVP codes allocated from the 'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.

8.5. Result-Code Values

This specification adds several Diameter Overload Control Application specific Permanent Failure codes from the 'Authentication,

Authorization, and Accounting (AAA) Parameters' Result-Code AVP Values (code 268) - Permanent Failure registry:

AVP Values	Attribute Name	Reference
5xxx	DIAMETER_NO_COMMON_SCOPE	RFCxxxx
5xxx	DIAMETER_NO_COMMON_ALGORITHM	RFCxxxx
5xxx	DIAMETER_TOCL_TOO_SMALL	RFCxxxx
5xxx	DIAMETER_TOCL_TOO_BIG	RFCxxxx
5xxx	DIAMETER_RATE_TOO_BIG	RFCxxxx

8.6. New Registries

Four new registries are needed under the 'Authentication, Authorization, and Accounting (AAA) Parameters' registry:

- o OC-Scope AVP Values: the policy for this registry is Specification Required.
- o OC-Action AVP Values: the policy for this registry is Standards Action.
- o OC-Level AVP Values: the policy for this registry is Standards Action.
- o OC-Algorithm AVP Values: the policy for this registry is Specification Required.

9. Security Considerations

The security properties of the Diameter Overload Control Application (DOCA) follows the security model of Diameter [RFC6733]. This implies there is no proper means to verify the message and AVP content correctness if multiple intermediate Diameter agents are present on the path between the DOCA client and server. As a result a malicious intermediate could feed incorrect overload control information to DOCA clients and peers, and thus affect negatively to the overload condition recovery. A possible way to overcome the obvious security vulnerability is to mandate the use of end-to-end security at the Diameter AVP level.

As such, like any other Diameter application this document would benefit from a Diameter end-to-end security mechanism. While work is in progress it has not yet been finalized and therefore this specification does not rely on it.

10. Acknowledgements

The author thanks Annett Seefeldt for her constructive comments on

the technical aspects on this document.

11. References

11.1. Normative References

- [I-D.ietf-dime-overload-reqs]
McMurry, E. and B. Campbell, "Diameter Overload Control Requirements", draft-ietf-dime-overload-reqs-04 (work in progress), February 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

11.2. Informative References

- [I-D.campbell-dime-overload-data-analysis]
Campbell, B., Tschofenig, H., Korhonen, J., and A. Roach, "Diameter Overload Data Analysis", draft-campbell-dime-overload-data-analysis-00 (work in progress), February 2013.
- [RFC6408] Jones, M., Korhonen, J., and L. Morand, "Diameter Straightforward-Naming Authority Pointer (S-NAPTR) Usage", RFC 6408, November 2011.

Appendix A. Design Justification

Section 1 discussed the motivation and the background for the Diameter enhancements for an explicit Diameter overload control solution. This specification solves the overload control at the application level instead of extending the Diameter base protocol or piggybacking overload control information on top of existing applications. The reasoning is the following:

1. The support for Diameter overload control capability between Diameter peers is explicit (i.e., a new application-id is advertised) and thus not build on an exchange of optional Attribute Value Pairs (AVPs).
2. The support for Diameter overload control capability between Diameter client and server is explicit.

3. The peer selection follows the existing standards including DNS-based discovery [RFC6408] and does not assume additional peer selection criteria learnt from an exchange of optional AVPs.
4. The application based solution is able to traverse and also propagate overload control information through realms that deploy relay agents without Diameter overload control support.
5. The propagation does not depend on a modified behavior of already specified Diameter command codes.
6. Pretending not to establish a state when there actually is an overload capability and information state still maintained. The state might not be at the application level but is there.
7. Trying to avoid information flooding, especially across administrative domains.
8. The use of the application concept allows established mechanisms for filtering and Diameter traffic engineering, since it behaves like any other Diameter application.
9. The use of the dedicated application allows to isolate (even physically) the overload signaling into a dedicated transport that is not affected by other Diameter messages and network traffic.

Authors' Addresses

Jouni Korhonen
Renesas Mobile
Porkkalankatu 24
Helsinki 00180
Finland

Email: jouni.nospam@gmail.com

Hannes Tschofenig (editor)
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: <http://www.tschofenig.priv.at>

