Diameter Maintenance and Extensions (DIME)
Internet-Draft
Intended status: Standards Track
Expires: ~~May 26,~~ **June 5,** 2014

J. Korhonen, Ed.
Broadcom
S. Donovan
B. Campbell
Oracle

~~November 22,~~
December 2, 2013

Diameter Overload Indication Conveyance

~~draft-ietf-dime-ovli-00.txt~~
draft-ietf-dime-ovli-01.txt

Abstract

This specification documents a Diameter Overload Control (DOC) base solution and the dissemination of the overload report information.

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].

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 ~~May 26,~~ **June 5,** 2014.

Copyright Notice

Table of Contents

1. Introduction

This specification defines a base solution for the Diameter Overload Control (DOC). The requirements for the solution are described and discussed in the corresponding design requirements document [I-D.ietf-dime-overload-reqs]. [RFC7068]. Note that the overload control solution defined in this specification does not address all the requirements listed in [I-D.ietf-dime-overload-reqs]. [RFC7068]. A number of overload control related features are left for the future specifications. See Appendix A for more detailed discussion on those.

The solution defined in this specification addresses the Diameter overload control between two endpoints (see Section 5.1). Furthermore, the solution is designed to apply to existing and future Diameter applications, requires no changes to the Diameter base protocol [RFC6733] and is deployable in environments where some Diameter nodes do not implement the Diameter overload control solution defined in this specification.

2. Terminology and Abbreviations

Server Farm

A set of Diameter servers that can handle any request for a given set of Diameter applications. While these servers support the same set of applications, they do not necessarily all have the same capacity. An individual server farm might also support a subset of the users for a Diameter Realm.

[OpenIssue: Is a A server farm assumed to support a single realm? That is, does it support a set of applications in may host a single realm?]

Server Front End

A Server Front End (SFE) is a role that can be performed by a Diameter agent either a relay or a proxy that sits between multiple realms.

Diameter clients and a Server Farm. An SFE can perform various functions for the server farm it sits in front of. This includes some or all of the following functions:

* Routing:

Diameter Routing

* Diameter layer load balancing

* Load Management
* Overload Management

* Topology Hiding

* Server Farm Identity Management

[OpenIssue: We used between non-adjacent nodes relies on the concept of a server farm and SFE for internal discussions. Do we still need those concepts Destination-Realm AVP to explain determine the mechanism? It doesn't seem like we use them much.] Diameter Routing:

Diameter Routing determines realm in which the destination of Diameter messages addressed request needs to either a Diameter Realm and Application be processed. A Destination-Host AVP may also be present in general, or the request to address a specific server using Destination Host. inside the Diameter realm. This function is defined in [RFC6733]. Application level routing specifications that expand on [RFC6733] also exist.

~~Diameter-layer~~

Diameter layer Load Balancing:

Diameter layer load balancing allows Diameter requests to be distributed across the set of servers. Definition of this function is outside the scope of this document.

~~Load-Management:~~

~~This functionality ensures that the consolidated load state for the server farm is collected, and processed. The exact algorithm for computing the load at the SFE is implementation specific but enough semantic of the conveyed load information needs to be specified so that deterministic behavior can be ensured.~~

~~Overload Management:~~

~~The SFE is the entity that understands the consolidated overload state for the server farm. Just as it is outside the scope of this document to specify how a Diameter server calculates its overload state, it is also outside the scope of this document to specify how an SFE calculates the overload state for the set of servers. This document describes how the SFE communicates Overload information to Diameter Clients.~~

Topology Hiding:

Topology Hiding is loosely defined as ensuring that no Diameter topology information about the server farm can be discovered from Diameter messages sent outside a predefined boundary (typically an administrative domain). This includes obfuscating identifiers and address information of Diameter entities in the server farm. It can also include hiding the number of various Diameter entities in the server farm. Identifying information can occur in many Diameter Attribute-Value Pairs (AVPs), including Origin-Host, Destination-Host, Route-Record, Proxy-Info, Session-ID and other AVPs.

~~Server Farm Identity Management:~~

~~Server Farm Identity Management (SFIM) is a mechanism that can be used by the SFE to present a single Diameter identity that can be used by clients to send Diameter requests to the server farm. This requires that the SFE modifies Origin-Host information in answers coming from servers in the server farm. An agent that performs SFIM appears as a server from the client's perspective.~~

~~Throttling:~~

~~Throttling~~

Throttling:

Throttling is the reduction of the number of requests sent to an entity. Throttling can include a client dropping requests, or an agent rejecting requests with appropriate error responses. Clients and agents can also choose to redirect throttled requests to some other entity or entities capable of handling them.

Reporting Node

A Diameter node that generates an overload report. (This may or may not be the actually overloaded node.)

Reacting Node

A Diameter node that consumes and acts upon a report. Note that "act upon" does not necessarily mean the reacting node applies an abatement algorithm; it might decide to delegate that downstream, in which case it also becomes a "reporting node".

OLR ~~Oveload~~ **Overload** Report.

3. Solution Overview

3.1. Architectural Assumptions

This section describes the high-level architectural and semantic assumptions that ~~underly~~ **underlie** the Diameter Overload Control Mechanism.

3.1.1. Application Classification

The following is a classification of Diameter applications and requests. This discussion is meant to document factors that play into decisions made by the Diameter identity responsible for handling overload reports.

Section 8.1 of [RFC6733] defines two state machines that imply two types of applications, session-less and ~~session-based.~~ **session-based applications.** The primary ~~differentiator~~ **difference** between these types of applications is the lifetime of ~~Session-IDs.~~ **Session-Ids.**

For session-based applications, the ~~session-id~~ **Session-Id** is used to tie multiple requests into a single session.

In session-less applications, the lifetime of the ~~session-id~~ **Session-Id** is a single Diameter ~~transaction.~~

~~The 3GPP-defined S6a application is an example of a session-less application. The following, copied from section 7.1.4 of 29.272, explicitly states that sessions are implicitly terminated and that the server does not maintain session state:~~

~~"Between the MME and the HSS and between the SGSN and the HSS and between the MME and~~ **transaction, i.e.** the ~~EIR, Diameter sessions shall be implicitly terminated. An implicitly terminated~~ session is ~~one for which the server does not maintain state information. The client shall not send any re-authorization or session termination requests to the server.~~

~~The Diameter base protocol includes the Auth-Session-State AVP as the mechanism for the implementation of~~ implicitly terminated ~~sessions.~~

~~The client (server) shall include in its requests (responses) the Auth-Session-State AVP set to the value NO_STATE_MAINTAINED (1), as described in [RFC6733]. As~~ **after** a ~~consequence, the server shall not maintain any state information about this session~~ **single Diameter transaction** and ~~the client shall not send any session termination~~ **a new Session-Id is generated for each Diameter** request. ~~Neither the Authorization-Lifetime AVP nor the Session-Timeout AVP shall be present in requests or responses."~~

For the purposes of this discussion, session-less applications are further divided into two types of applications:

Stateless applications:

Requests within a stateless application have no relationship to each other. The 3GPP defined S13 application is an example of a stateless ~~application.~~ **application [3GPP.29.272], where only a Diameter command is defined between a client and a server and no state is maintained between two consecutive transactions.**

Pseudo-session applications: ~~While this class of application does~~

applications that do not ~~use~~ **rely on** the ~~Diameter Session ID~~ **Session-Id** AVP ~~to correlate requests, there is an implied ordering~~ **for correlation** of ~~transactions defined by~~ **application messages related to** the ~~application.~~ **same session but use other session-related information for this purpose.** The 3GPP defined Cx application ~~[reference]~~ **[3GPP.29.229]** is an example of a pseudo-session application.

~~[OpenIssue: Do we assume that all requests in a pseudo-session typically need to go to the same server?]~~

The ~~accounting application defined in [RFC6733] and the Credit-Control~~ **Credit-Control** application defined in [RFC4006] ~~are examples~~ **is an example of a** Diameter session-based ~~applications.~~ **application.**

The handling of overload reports must take the type of application into consideration, as discussed in Section 3.1.2.

3.1.2. Application Type Overload Implications

This section discusses considerations for mitigating overload reported by a Diameter entity. This discussion focuses on the type of application. Section 3.1.3 discusses considerations for handling various request types when the target server is known to be in an overloaded state. ~~Section 3.1.5 discusses considerations for handling overload conditions based on the network deployment scenario.~~

These discussions assume that the strategy for mitigating the reported overload is to reduce the overall workload sent to the overloaded entity. The concept of applying overload treatment to requests targeted for an overloaded Diameter entity is inherent to this discussion. The method used to reduce offered load is not specified here but could include routing requests to another Diameter entity known to be able to handle them, or it could mean rejecting certain requests. For a Diameter agent, rejecting requests will usually mean generating appropriate Diameter error responses. For a Diameter client, rejecting requests will depend upon the application. For example, it could mean giving an indication to the entity requesting the Diameter service that the network is busy and to try again later.

Stateless applications:

By definition there is no relationship between individual requests in a stateless application. As a result, when a request is sent or relayed to an overloaded Diameter entity - either a Diameter Server or a Diameter Agent - the sending or relaying entity can choose to apply the overload treatment to any request targeted for the overloaded entity.

~~Pseudo-stateful~~

Pseudo-session applications: ~~Pseudo-stateful applications are also stateless applications in that~~

For pseudo-session applications, there is ~~no session Diameter state maintained between transactions. There is, however,~~ an implied ordering of requests. As a result, decisions about which ~~transactions to reject as a result of~~ **requests towards** an overloaded entity **to reject** could take the ~~command code~~ **command code** of the request into consideration. This generally means that transactions later in the sequence of transactions should be given more favorable treatment than messages earlier in the sequence. This is because more work has already been done by the Diameter network for those transactions that occur later in the sequence. Rejecting them could result in increasing the load on the network as the transactions earlier in the sequence might also need to be repeated.

~~Stateful~~

Session-based applications:

Overload handling for ~~stateful~~ **session-based** applications must take into consideration the work **load** associated with setting up ~~an~~ **and** maintaining a session. As such, the entity ~~handling overload of a~~ **sending requests towards an overloaded** Diameter entity for a ~~stateful~~ **session-based** application might tend to reject new session requests ~~before~~ **prior to** rejecting intra-session requests. In addition, session ending requests might be given a lower priority of being rejected as rejecting session ending requests could result in session status being out of sync between the Diameter clients and servers. ~~Nodes~~

Application designers that **would decide to** reject mid-session requests will need to consider whether the rejection invalidates the ~~session,~~ **session** and any **resulting** session clean-up ~~that may be required.~~ **procedures.**

3.1.3. Request Transaction Classification

Independent Request:

An independent request is not ~~a part of a Diameter session~~ **correlated to any other requests** and, as such, the lifetime of the session-id is constrained to an individual transaction.

Session-Initiating Request:

A session-initiating request is the initial message that establishes a Diameter session. The ACR message defined in [RFC6733] is an example of a session-initiating request.

Correlated Session-Initiating Request:

There are ~~cases, most~~ **cases when multiple session-initiated requests must be correlated and managed by the same Diameter server. It is** notably **the case** in the 3GPP PCC ~~architecture,~~ **architecture [3GPP.23.203],** where multiple **apparently independent** Diameter **application** sessions are **actually** correlated and must be handled by the same Diameter server. ~~This is a special case of a Session-Initiating Request. Gx CCR-I requests and Rx AAR messages are examples of correlated session-initiating requests.~~

~~[OpenIssue: The previous paragraph needs references.]~~

Intra-Session Request:

An intra session request is a request that uses **the same Session-Id than the one used in** a ~~session id for an already established~~ **previous** request. An intra session request generally needs to be delivered to the server that handled the session creating request for the session. The STR message defined in [RFC6733] is an example of an intra-session requests. ~~CCR-U and CCR-T requests defined in [RFC4006] are further examples of intra-session requests.~~

Pseudo-Session Requests: ~~Pseudo session~~

Pseudo-session requests are independent requests ~~and, as such,~~ **and do not use** the ~~request transactions~~ **same Session-Id but** are ~~not tied together using~~ **correlated by other session-related information contained in** the ~~Diameter session-id.~~ **request.** There ~~exist~~ **exists** Diameter applications that define an expected ordering of transactions. This sequencing of independent transactions results in a pseudo session. The AIR, MAR and SAR requests in the 3GPP defined Cx application are examples of pseudo-session requests.

3.1.4. Request Type Overload Implications

The request classes identified in Section 3.1.3 have implications on decisions about which requests should be throttled first. **The following list of request treatment regarding throttling is provided as guidelines for application designers when implementing the Diameter overload control mechanism described in this document. Exact behavior regarding throttling must be defined per application.**

Independent requests:

Independent requests can be given equal treatment when making

throttling decisions.

~~Session-creating~~

Session-initiating requests: ~~Session-creating~~

Session-initiating requests represent more work than independent or intra-session requests. **Moreover, session-initiated requests are typically followed by other related session-related requests.** As such, **as the main objective of the overload control is to reduce the total number of requests sent to the overloaded entity,** throttling decisions might favor intra-session requests over ~~session-creating~~ **session-initiating** requests. Individual ~~session-creating~~ **session-initiating** requests can be given equal treatment when making throttling decisions.

Correlated ~~session-creating~~ **session-initiating** requests:

A Request that results in a new binding, where the binding is used for routing of subsequent ~~session-creating requests,~~ **session-initiating requests to the same server,** represents more work **load** than other requests. As such, these requests might be throttled more frequently than other request types.

Pseudo-session requests:

Throttling decisions for pseudo-session requests can take where individual requests fit into the overall sequence of requests within the pseudo session. Requests that are earlier in the sequence might be throttled more aggressively than requests that occur later in the sequence.

Intra-session requests

There are two classes of intra-sessions requests. The first ~~is a request~~ **class consists of requests** that ~~ends~~ **terminate** a session. The second ~~is a request~~ **one contains the set of requests** that ~~is~~ **are** used ~~to convey session-related state between~~ **by** the Diameter client and ~~server.~~ **server to maintain the ongoing session state.** Session ~~ending request~~ **terminating requests** should be throttled less aggressively in order to ~~keep session state consistent between~~ **gracefully terminate sessions, allow clean-up of** the ~~client and server,~~ **related resources (e.g. session state)** and ~~possibly reduce~~ **get rid of** the ~~sessions~~ **need for other intra-session requests, reducing the session managements** impact on the overloaded entity. The default handling of other intra-session requests might be to treat them equally when making throttling decisions. There might also be application level considerations whether some request types are favored over others.

3.1.5. Diameter ~~Deployment Scenarios~~

~~This section discusses various~~ **Agent Behaviour**

In the context of the Diameter ~~network deployment scenarios~~ **Overload Indication Conveyance (DOIC)** and **reacting to** the ~~implications~~ **overload information, the functional behaviour** of ~~those deployment models on~~ **Diameter agents in front of servers, especially Diameter proxies, needs to be common. This is important because agents may actively participate in the** handling of **an** overload ~~reports.~~

~~The scenarios vary~~ **conditions. For example, they may make intelligent next hop selection decisions** based on ~~the following:~~

~~o The presence~~ **overload conditions,** or ~~absence of~~ **aggregate overload information to be disseminated downstream.** Diameter agents

~~o Which Diameter entities support~~ **may have other deployment related tasks that are not defined in** the ~~DOC extension~~

~~o The amount~~ **Diameter base protocol [RFC6733]. These include, among other tasks, topology hiding, or agent acting as a Server Front End (SFE) for a farm** of **Diameter servers.**

Since the solution defined in this specification must not break the ~~network topology understood by~~ Diameter ~~clients~~

~~o The complexity of~~ **base protocol [RFC6733] at any time, great care has to be taken not to assume functionality from** the Diameter ~~server deployment for~~ **agents that would break base protocol behavior, or to assume agent functionality beyond the Diameter base protocol. Effectively this means the following from** a Diameter ~~application~~ **agent:**

o ~~Number of Diameter applications supported by Diameter clients and~~ **If a** Diameter ~~servers~~

~~Without consideration for which elements support~~ **agent presents itself as** the ~~DOC extension,~~ **"end node", as an agent acting as an topology hiding SFE,** the ~~following~~ **agent is** ~~a representative list~~ **the final destination** of ~~deployment scenarios:~~

~~o Client --- Server~~

~~o Client --- Multiple equivalent servers~~

~~o Client --- Agent --- Multiple equivalent servers~~

~~o Client --- Agent [--- Agent] --- Partitioned server~~

~~o Client --- Edge Agent [--- Edge Agent] --- { Multiple Equivalent Servers | Partitioned Servers }~~

~~o Client --- Session Correlating Agent --- Multiple Equivalent Servers~~

~~[OpenIssue: Do~~ **requests initiated by Diameter clients,** the ~~"multiple equivalent servers" cases change for session-stateful applications? Do we need to distinguish equivalence~~ **original source** for ~~session-initiation requests vs intra-session requests?]~~

~~The following is~~ **the corresponding answers and server-initiated requests. As a** ~~list~~ **consequence, the DOIC mechanism MUST NOT leak information** of ~~representative DOC deployment scenarios:~~

~~o Direct connection between a DOC client and~~ **the Diameter nodes behind it. This requirement means that such a** ~~DOC server~~

~~o DOC client non DOC agent DOC server~~

~~o DOC client --- DOC agent --- DOC server~~

~~o Non-DOC client --- DOC~~ **Diameter** agent ~~--- DOC server~~

~~o Non-DOC client --- DOC~~ **acts as a back-to-back-agent for DOIC purposes. How the Diameter** agent ~~--- Mix of DOC and non-DOC~~ **in this case appears to the Diameter** servers

~~o DOC client --- agent --- Partitioned/Segmented DOC server~~

~~o DOC client --- agent ---~~ **in the farm, is specific to the implementation and deployment within the realm the Diameter** agent ~~--- Partitioned/Segmented DOC server~~ **is deployed.**

~~o DOC client --- edge agent --- edge agent --- DOC server~~

~~[OpenIssue: In the last 3 list entries, are~~ **If** the ~~agents DOC or non-DOC?]~~

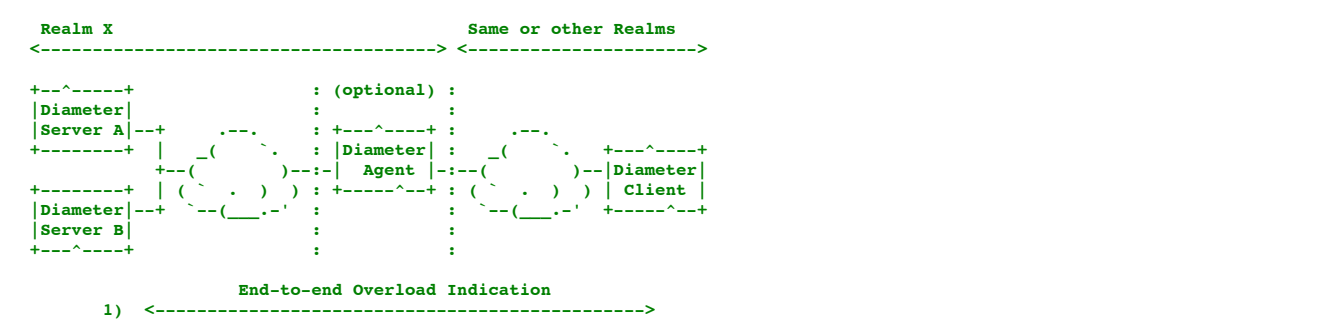
~~3.1.6.~~ Diameter ~~Agent Behaviour~~

~~In~~ **agent does not impersonate** the ~~context of~~ **servers behind it,** the Diameter ~~Overload Indication Conveyance (DOIC)~~ **dialogue is established between clients** and ~~reacting to the~~ **servers and any** overload ~~information,~~ **information received by a client would be from** the ~~functional behaviour of Diameter agents in front of servers, especially Diameter proxies, needs to be common. This is important because agents may actively participate~~ **server identified by the Origin-Host identity contained** in the ~~handling of an overload conditions. For example, they may make intelligent next hop selection decisions based on overload conditions, or aggregate overload information to be disseminated downstream.~~ Diameter ~~agents may have other deployment related tasks that are not defined in~~ **message.**

3.1.6. Simplified Example Architecture

Figure 1 illustrates the **simplified architecture for** Diameter ~~base protocol [RFC6733]. These include, among other tasks, topology hiding, and acting as a server front end~~ **overload information conveyance. See Section 5.1** for ~~a server farm of real~~ **more discussion and details how different** Diameter ~~servers.~~

~~Since~~ **nodes fit into** the ~~solution defined in this specification must not break~~ **architecture from** the **DOIC point of view.**



~~[RFC6733] at any time, great care has to be taken not to assume functionality from the Diameter agents that would break~~ **standard** base protocol ~~behavior,~~

Figure 1: Simplified architecture choices for overload indication delivery

In Figure 1, the Diameter overload indication can be conveyed (1) end-to-end between servers and clients or ~~to assume~~ **(2) between servers and**

Diameter agent ~~functionality beyond~~ inside the ~~Diameter base protocol. Effectively this means~~ realm and then between the ~~following from a Diameter agent:~~

~~o If a~~ Diameter agent ~~presents itself as~~ and the ~~"end node", perhaps~~ clients when the Diameter agent acting as ~~an topology hiding SPE,~~ back-to-back-agent for DOIC purposes.

3.2. Conveyance of the ~~DOC mechanism MUST NOT leak information~~ Overload Indication

The following sections describe new Diameter AVPs used for sending overload reports, and for declaring support for certain DOIC features.

3.2.1. DOIC Capability Discovery

Support of DOIC may be specified as part of the functionality supported by a new Diameter ~~nodes behind it. From~~ application. In this way, support of the considered Diameter ~~client point~~ application (discovered during capabilities exchange phase as defined in Diameter base protocol [RFC6733]) indicates implicit support of ~~view~~ the ~~final destination to its requests and~~ DOIC mechanism.

When the ~~original source for~~ DOIC mechanism is introduced in existing Diameter applications, a specific capability discovery mechanism is required. The "DOIC capability discovery mechanism" is based on the ~~answers MUST be~~ presence of specific optional AVPs in the Diameter ~~agent. This requirement means that~~ messages, such ~~a Diameter agent acts~~ as ~~a back-to-back-agent for DOC purposes. How~~ the ~~agent in this case appears to~~ OC-Feature-Vector AVP (see Section 4.1). Although the OC-Feature-Vector AVP can be used to advertise a certain set of new or existing Diameter ~~nodes~~ overload control capabilities, it is ~~representing (i.e. the real Diameter servers), is an implementation and~~ not a ~~deployment specific within~~ versioning solution per se, however, it can be used to achieve the ~~realm~~ same result.

From the Diameter ~~agent~~ overload control functionality point of view, the "Reacting node" is ~~deployed.~~

~~o This requirement also implies that if~~ the ~~Diameter agent does not impersonate~~ requester of the ~~servers behind it,~~ overload report information and the ~~Diameter dialogue~~ "Reporting node" is ~~established between clients and servers and any overload information received by a client would be from a given server identified by~~ the ~~Origin-Host identity.~~

~~[OpenIssue: We've discussed multiple situations where an agent might insert an OLR. I don't think we mean to force them to always perform topology hiding or SPIM~~ provider of the overload report. The OC-Feature-Vector AVP in ~~order to do so. We cannot assume that an OLR~~ the request message is always ~~"from" or "about" the Origin-Host. Also, the section seems to assume that topology hiding agents act~~ interpreted as ~~b2b overload agents, but non-topology hiding agents never do. It don't think that's~~ an announcement of "DOIC supported capabilities". The OC-Feature-Vector AVP in the ~~right abstraction. It's possible that topology hiding agents must do this, but I don't think we can preclude non-topology hiding agents from~~ answer is also ~~doing it,~~ interpreted as a report of "DOIC supported capabilities" and at least ~~some~~ one of supported capabilities MUST be common with the ~~time.]~~

~~3.1.7. Simplified Example Architecture~~

~~Figure 1 illustrates the simplified architecture for~~ "Reacting node" (see Section 4.1).

3.3. Overload Condition Indication

Diameter nodes can request a reduction in offered load by indicating an overload ~~control. See Section 5.1 for more discussion and details~~ condition in the form of an overload report. The overload report contains information about how ~~different~~ much load should be reduced, and may contain other information about the overload condition. This information is conveyed in Diameter ~~nodes fit into~~ Attribute Value Pairs (AVPs).

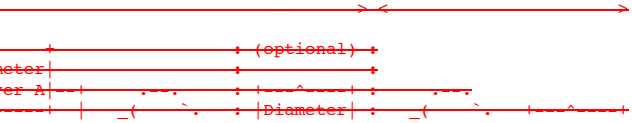
Certain new AVPs may also be used to declare certain DOIC capabilities and extensions.

4. Attribute Value Pairs

This section describes the ~~architecture from~~ encoding and semantics of the Diameter Overload Indication Attribute Value Pairs (AVPs) defined in this document.

4.1. OC-Feature-Vector AVP

The OC-Feature-Vector AVP (AVP code TBD1) is type of Grouped and contains the description of supported DOIC ~~point of view.~~



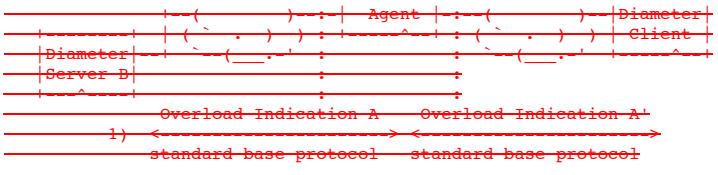


Figure 1: Simplified architecture choices for overload indication delivery

3.2. Conveyance features of the Overload Indication

The following features describe new Diameter AVPs used for AVP sending overload reports, and for declaring support for certain DOC features.

3.2.1. Negotiation and Versioning

Since the Diameter overload control mechanism is also designed to work over existing application (i.e., the piggybacking principle), a proper negotiation is hard to accomplish. node.

```
OC-Feature-Vector ::= < AVP Header: TBD1 >
                      < OC-Sequence-Number >
                      [ OC-Features ]
                    * [ AVP ]
```

The "capability negotiation" OC-Sequence-Number AVP is based on used to indicate whether the existense contents of specific non-mandatory AVP, such as the OC-Feature-Vector AVP (see Section 4.1. Although has changed since last time the OC-Feature-Vector AVP can be used to advertise a certain set of new or existing Diameter overload control capabilities, it emitting node announced its DOIC features and capabilities (see Section 4.4).

The OC-Features AVP is not a versioning solution per se, however, it can be used to achieve announced the same result.

3.2.2. Transmission DOC features supported

by the DOIC endpoint, in the form of a Flag bits field in which each bit announces one feature or capability supported by the Attribute Value Pairs node (see Section 4.2). The Diameter overload control AVPs SHOULD always be sent as an optional AVPs. This requirement stems from absence of the fact OC-Features AVP indicates that piggybacking overload control information on top of existing application cannot really use AVPs with only the M-bit set. However, there are certain exceptions as explained default traffic abatement algorithm described in Section 5.4.

From the Diameter overload control functionality point of view, the "Reacting node" this specification is always supported.

A reacting node includes this AVP to indicate its capabilities to a reporting node. For example, the requester of endpoint (reacting node) may indicate which (future defined) traffic abatement algorithms it supports in addition to the overload report information and default.

During the "Reporting node" is message exchange the provider overload control endpoints express their common set of supported capabilities. The reacting node includes the overload report. OC-Feature-Vector AVP that announces what it supports. The overload report or reporting node that sends the capability information in answer also includes the request message is always interpreted as an announcement of a "capability". OC-Feature-Vector AVP that describe the capabilities it supports. The overload report and set of capabilities advertised by the capability information in reporting node depends on local policies. At least one the answer announced capabilities MUST match mutually. If there is always interpreted no single matching capability the reacting node MUST act as a report if it does not implement DOIC and cease inserting any DOIC related AVPs into any Diameter messages with this specific reacting node.

4.2. OC-Features AVP

The OC-Features AVP (AVP code TBD6) s type of supported commond functionality Unsigned64 and as contains a status report 64 bit flags field of announced capabilities of an overload condition (of a node).

3.3. Overload Condition Indication

Diameter nodes can request a reduction in offered load by indicating an overload condition in the form control endpoint. The value of an overload report. zero (0) is reserved.

The overload report contains information about how much load should be reduced, and may contain other information about following capabilities are defined in this document:

OLR_DEFAULT_ALGO (0x0000000000000001)

When this flag is set by the overload
~~condition. This information is encoded in Diameter Attribute Value~~
~~Pairs (AVPs).~~

~~Certain new AVPs may also be used to declare certain DOIC~~
~~capabilities and extensions.~~

~~4. Attribute Value Pairs~~

~~This section describes~~ control endpoint it means
that the ~~encoding and semantics of Overload~~
~~Indication Attribute Value Pairs (AVPs).~~

~~4.1. OC Feature Vector~~ default traffic abatement (loss) algorithm is supported.

4.3. OC-OLR AVP

The ~~OC Feature Vector~~ OC-OLR AVP (AVP code ~~TBD1)~~ TBD2) is type of ~~Unsigned64~~ Grouped and contains ~~a 64 bit flags field of announced capabil~~
necessary information to convey an overload ~~control endpoint. Sending or receiving the OC Feature~~
~~Vector AVP with the value 0 indicates~~ report. OC-OLR may also
be used to convey additional information about an extension that ~~the endpoint only support~~
~~the capabilities defined~~ is
declared in ~~this specification.~~

~~An overload control endpoint (a reacting node) includes this~~ the OC-Feature-Vector AVP.

The OC-OLR AVP does not contain explicit information to
~~indicate its capabilities to the other overload control endpoint (the~~
~~reporting node). For example, the endpoint (reacting node) may~~
~~indicate~~ which ~~(future defined) traffic abatement algorithms~~
application it
~~supports in addition~~ applies to and who inserted the ~~default.~~

~~During~~ AVP or whom the ~~message exchange~~
specific OC-OLR AVP concerns to. Both these information is
implicitly learned from the ~~overload control endpoints express~~
~~their common set of supported capabilities.~~ encapsulating Diameter message/command.
The ~~endpoint sending a~~
~~request (the reacting node) includes~~ application the ~~OC Feature Vector AVP with~~
~~those flags set that correspond what it supports. The endpoint that~~
~~sends the answer (the reporting node) also includes the OC Feature~~
~~Vector AVP with flags set to describe the capabilities it both~~
~~supports and agrees with the request sender (e.g., based on the local~~
~~policy and/or configuration). The answer sending endpoint (the~~
~~reporting node) does not need to advertise those capabilities it is~~
~~not going to use with the request sending endpoint (the reacting~~
~~node).~~

~~This specification does not define any additional capability flag.~~
~~The implicity capability (all flags set to zero) indicates the~~
~~support for this specification only.~~

~~4.2. OC-OLR AVP~~

~~The OC-OLR AVP (AVP code TBD2) is type of Grouped and contains the~~
~~necessary information to convey an overload report. OC-OLR may also~~
~~be used to convey additional information about an extension that is~~
~~declared in the OC-Feature-Vector AVP.~~

~~The OC-OLR AVP does not contain explicit information to which~~
~~application it applies to and who inserted the AVP or whom the~~
~~specific OC-OLR AVP concerns to. Both these information is~~
~~implicitly learned from the encapsulating Diameter message/command.~~
~~The application the OC-OLR~~ OC-OLR AVP applies to is the same as the
Application-Id found in the Diameter message header. The identity
the OC-OLR AVP concerns is determined from the Origin-Host AVP (and
Origin-Realm AVP as well) found from the encapsulating Diameter
command.

```
OC-OLR ::= < AVP Header: TBD2 >
           < ~~TimeStamp~~ OC-Sequence-Number >
           [ ~~Reduction-Percentage~~ OC-Report-Type ]
           [ ~~ValidityDuration~~ OC-Reduction-Percentage ]
           [ ~~ReportType~~ OC-Validity-Duration ]
         * [ AVP ]
```

The ~~TimeStamp~~ Sequence-Number AVP indicates ~~when~~ the ~~original OC-OLR AVP with~~ "freshness" of the
~~current content was created.~~ OC-OLR AVP.
It is possible to replay the same ~~OC-~~
~~OLR~~ OC-OLR AVP multiple times between
the overload control endpoints, however, when the OC-OLR AVP content
changes or ~~the other information~~ sending endpoint wants the receiving endpoint to update
its overload control information, then the ~~TimeStamp~~ iOC-Sequence-Number AVP
MUST contain a new ~~value.~~

~~[OpenIssue: Is this necessarily a timestamp, or is it just~~ greater value than the previously received. The
receiver SHOULD discard an OC-OLR AVP with a sequence number that ~~can be implemented as~~ is
less than previously received one.

Note that if a ~~timestamp? Is this timestamp~~
~~used~~ Diameter command were to ~~calculate expiration time? (propose no.). We should also~~
~~consider whether either a timestamp or sequence number is needed for~~
~~protection against replay attacks.]~~

~~4.3. TimeStamp~~ contain multiple OC-OLR AVPs

they all MUST have different OC-Report-Type AVP value.

4.4. OC-Sequence-Number AVP

The ~~TimeStamp~~ OC-Sequence-Number AVP (AVP code TBD3) is type of ~~Time.~~ Unsigned64. Its usage in the context of the overload control is described in ~~Section 4.2.~~ Sections 4.1 and 4.3. From the functionality point of view, the ~~TimeStamp~~ OC-Sequence-Number AVP ~~is merely~~ MUST be used as a non-volatile increasing counter between two overload control endpoints.

~~4.4. ValidityDuration~~ The sequence number is only required to be unique between two overload control endpoints and does not need to be monotonically increasing.

[Editor's note: how to handle overflows? With time stamps that would be "trivial" since the sequence number would have a structure and we would also know the "validity window" from the life time of the OC-OLR.]

4.5. OC-Validity-Duration AVP

The ~~ValidityDuration~~ OC-Validity-Duration AVP (AVP code TBD4) is type of Unsigned32 and describes the number of seconds the OC-OLR AVP and its content is valid since the ~~creation~~ reception of the new OC-OLR AVP (as indicated by the ~~TimeStamp~~ OC-Sequence-Number AVP). The default value for the OC-Validity-Duration AVP value is 5 (i.e., 5 seconds). When the OC-Validity-Duration AVP is not present in the OC-OLR AVP, the default value applies.

A timeout of the overload report has specific concerns that need to be taken into account by the endpoint acting on the earlier received overload report(s). Section ~~4.6~~ 4.7 discusses the impacts of timeout in the scope of the traffic abatement algorithms.

As a general guidance for implementations it is RECOMMENDED never to let any overload report to timeout. ~~Rather,~~ Following to this rule, an overload endpoint should explicitly ~~signal, e.g.~~ signal the end of overload ~~condition.~~ condition and not rely on the expiration of the validity time of the overload report in the reacting node. This leaves no need for the ~~other overload endpoint~~ reacting node to reason or guess the overload condition of the ~~other endpoint is at.~~

~~4.5. ReportType~~ reporting node.

4.6. OC-Report-Type AVP

The ~~ReportType~~ OC-Report-Type AVP (AVP code TBD5) is type of Enumerated. The value of the AVP describes what the overload report concerns. The following values are initially defined:

0 Reserved.

1 ~~Destination-Host~~ A host report. The overload treatment should apply to requests that the ~~sender~~ reacting node knows will reach the overloaded ~~server.~~ node. For example, requests with a Destination-Host AVP indicating the ~~server.~~ endpoint. The reacting node learns the "host" implicitly from the Origin-Host AVP of the received message that contained the OC-OLR AVP.

2 ~~Realm (aggregated)~~ A realm report. The overload treatment should apply to all requests bound for the overloaded realm. The reacting node learns the "realm" implicitly from the Origin-Realm AVP of the received message that contained the OC-OLR AVP.

The default value of the OC-Report-Type AVP is 1 (i.e. the host report).

The ReportType AVP is envisioned to be useful for situations where a reacting node needs to apply different overload treatments for different "types" of overload. For example, the reacting node(s) might need to throttle differently requests ~~that are targeted~~ sent to a specific server (identified by the ~~presence of a~~ Destination-Host AVP ~~than for~~ in the request) and requests that can be handled by any server in a realm. The example in Appendix ~~C.3~~ B.1 illustrates this usage.

~~[OpenIssue: There is an ongoing discussion about whether the ReportType AVP is the right way to solve that issue, and whether it's needed at all.]~~

~~4.6. Reduction-Percentage~~

4.7. OC-Reduction-Percentage AVP

The ~~Reduction-Percentage~~ OC-Reduction-Percentage AVP (AVP code TBD8) is type of Unsigned32 and describes the percentage of the traffic that the sender is requested to reduce, compared to what it otherwise would have sent. The OC-Reduction-Percentage AVP applies to the default (loss like) algorithm specified in this specification. However, the AVP can be reused for future abatement algorithms, if its semantics fit into the new algorithm.

The value of the Reduction-Percentage AVP is between zero (0) and one hundred (100). Values greater than 100 are interpreted as 100. The value of 100 means that no traffic is expected, i.e. the ~~sender of the information~~ reporting

node is under a severe load and ceases to process any new messages.
The value of 0 means that the ~~sender of the information~~ reporting node is in a stable state and
has no requests to the other endpoint to apply any traffic abatement.

~~[Open Issue: We should consider an algorithm independent way to end
an overload condition. Perhaps setting the validitytime to zero?
Counter comment; since~~
The default value of the ~~abatement~~ OC-Reduction-Percentage AVP is ~~based on a specific
algorithm, it~~ 0. When the
OC-Reduction-Percentage AVP is ~~natural to indicate that from~~ not present in the ~~abatement
algorithm point of view status quo has been reached.]~~ overload report,
the default value applies.

If an overload control endpoint comes out of the 100 percent traffic
reduction as a result of the overload report timing out, the
following concerns are RECOMMENDED to be applied. The ~~endpoint~~ reacting node
sending the traffic should be conservative and, for example, first
send few "probe" messages to learn the overload condition of the
~~other endpoint~~
overloaded node before converging to any traffic amount/rate decided
by the sender. Similar concerns actually apply in all cases when the
overload report times out unless the previous overload report stated
0 percent reduction.

~~[Open Issue: It is still open whether we need an AVP to indicate the
exact used traffic abatement algorithm. Currently it assumed that
the reacting node is able to figure out what to do based on the
Reducttion-Percentage AVP and possible other embedded information
inside the OC-OLR AVP.]~~

~~4.7.~~

4.8. Attribute Value Pair flag rules

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules MUST	AVP flag rules MUST NOT
OC-Feature-Vector	TBD1	x.x	~~Unsigned64~~ Grouped		V
OC-OLR	TBD2	x.x	Grouped		V
~~TimeStamp~~ OC-Sequence-Number	TBD3	x.x	~~Time~~ Unsigned64		V
~~ValidityPeriod~~ OC-Validity-Period	TBD4	x.x	Unsigned32		V
~~ReportType~~ OC-Report-Type	TBD5	x.x	Enumerated		V
~~Reduction~~ OC-Reduction -Percentage	TBD8	x.x	Unsigned32		V
OC-Features	TBD6	x.x	Unsigned64		V

As described in the Diameter base protocol [RFC6733], the M-bit
setting for a given AVP is relevant to an application and each
command within that application that includes the AVP.

The Diameter overload control AVPs SHOULD always be sent with the
M-bit cleared when used within existing Diameter applications to
avoid backward compatibility issues. Otherwise, when reused in newly
defined Diameter applications, the DOC related AVPs SHOULD have the
M-bit set.

5. Overload Control Operation

5.1. Overload Control Endpoints

The overload control solution can be considered as an overlay on top
of an arbitrary Diameter network. The overload control information
is exchanged over on a "DOIC association" between two ~~communicatin~~ communication
endpoints. The endpoints, namely the "reacting node" and the
"reporting node" do not need to be adjacent Diameter peer nodes, nor
they need to be the end-to-end Diameter nodes in a typical "client-
server" deployment with multiple intermediate Diameter agent nodes in
between. The overload control endpoint are the two Diameter nodes
that decide to exchange overload control information between each
other. How the endpoints are determined is specific to a deployment,
a Diameter node role in that deployment and local configuration.

The following diagrams illustrate the concept of Diameter Overload
End-Points and how they differ from the standard [RFC6733] defined
client, server and agent Diameter nodes. The following is the key to
the elements in the diagrams:

C Diameter client as defined in [RFC6733].

S Diameter server as defined in [RFC6733].

A Diameter agent, in either a relay or proxy mode, as defined in

[RFC6733].

DEP Diameter Overload End-Point as defined in this document. In the following figures a DEP may terminate two different DOIC associations being a reporter and reactor at the same time.

Diameter Session A Diameter session as defined in [RFC6733].

DOIC Association A DOIC association exists between two Diameter Overload End-Points. One of the end-points is the overload reporter and the other is the overload reactor.

Figure 2 illustrates the most basic configuration where a client is connected directly to a server. In this case, the session and association are both between the client and server.

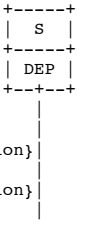


Figure 2: Basic DOIC deployment

In Figure 3 there is an agent that is not participating directly in the exchange of overload reports. As a result, the DOIC association is still between the client and the server.

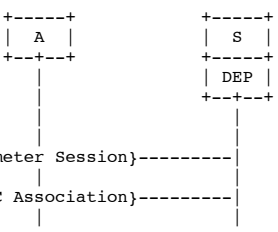


Figure 3: DOIC deployment with non participating agent

Figure 4 illustrates the case where the client does not support Diameter overload. In this case, the DOIC association is between the agent and the server. The agent handles the role of the reactor for overload reports generated by the server.

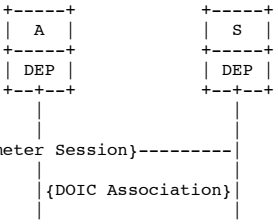


Figure 4: DOIC deployment with non-DOIC client and DOIC enabled agent

In Figure 5 there is a DOIC association between the client and the agent and a second DOIC association between the agent and the server. One use case requiring this configuration is when the agent is serving as a ~~SFE/SFIM~~ **SFE** for a set of servers.

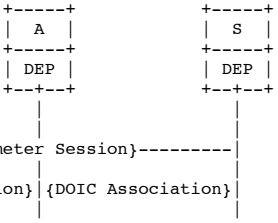
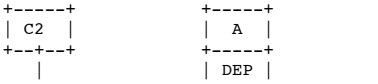


Figure 5: A deployment where all nodes support DOIC

Figure 6 illustrates a deployment where some clients support Diameter overload control and some do not. In this case the agent must support Diameter overload control for the non supporting client. It might also need to have a DOIC association with the server, as shown here, to handle overload for a server farm and/or for managing Realm overload.



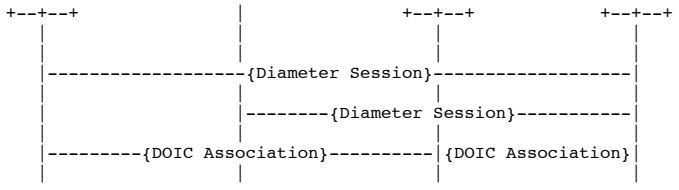


Figure 6: A deployment with DOIC and non-DOIC supporting clients

Figure 7 illustrates a deployment where some agents support Diameter overload control and others do not.

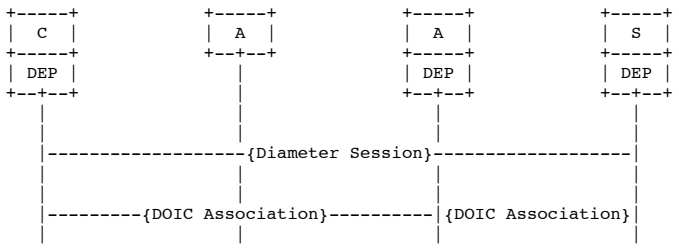


Figure 7: A deployment with DOIC and non-DOIC supporting agents

5.2. Piggybacking Principle

The overload control solution defined AVPs are essentially piggybacked on top of existing application message exchanges. This is made possible by adding overload control top level AVPs, the OC-OLR AVP and the OC-Feature-Vector AVP into existing commands (this has an assumption that the application CCF allows adding new AVPs into the Diameter messages.

In a case of newly defined Diameter applications, it is RECOMMENDED to add and defined how overload control mechanisms works on that application. using OC-Feature-Vector and OC-OLR AVPs in a non-mandatory manner is intended only existing applications.

Note that the overload control solution does not have fixed server and client roles. The endpoint role is determined based on the sent message type: whether the message is a request (i.e. sent by a "reacting node") or an answer (i.e. send by a "reporting node"). Therefore, in a typical "client-server" deployment, the "client" MAY report its overload condition to the "server" for any server initiated message exchange. An example of such is the server requesting a re-authentication from a client.

5.3. Capability Announcement

Since the overload control solution relies on the piggybacking principle for the overload reporting and the overload control endpoint are likely not adjacent peers, finding out whether the other endpoint supports the overload control or what is the common traffic abatement algorithm to apply for the traffic. The approach defined in this specification for the end-to-end capability capability announcement relies on the exchange of the OC-Feature-Vector between the endpoints. The feature announcement solution also works when carried out on existing applications. For the newly defined application the negotiation can be more exact based on the application specification. The announced set of capabilities MUST NOT change during the life time of the Diameter session (or transaction in a case of non-session maintaining applications).

5.3.1. Request Message Initiator Endpoint Considerations

The basic principle is that the request message initiating endpoint (i.e. the "reacting node") announces its support for the overload control mechanism by including in the request message the OC-Feature-Vector AVP with those ~~capability flag bits set that~~ **capabilities** it supports and is willing to use for this Diameter session (or transaction in a case of a non-session state maintaining ~~applications). In a case of session maintaining applications the request message initiating endpoint does not need to do the capability announcement more than once~~ **applications, see Section 3.1.2** for ~~the lifetime of the~~ **more details on** Diameter ~~session. In a case of non-session maintaining applications, it~~ **sessions). It** is RECOMMENDED that the request message initiating endpoint includes the capability announcement into every request regardless it has had prior message exchanges with the give remote endpoint.

~~[OpenIssue: We need to think about the lifetime of a capabilities declaration. It's probably not~~

Once the ~~same as for a session. We have had proposals~~ **endpoint** that **initiated** the ~~feature vector needs to go into every~~ request ~~sent by an OC node. For peer to peer cases, this can be associated with connection lifetime, but it's more complex for non adjacent OC support.]~~

~~Once the endpoint that initiated the request message receives~~ **message receives** an answer message from the remote endpoint, it can detect from the

received answer message whether the remote endpoint supports the overload control solution and in a case it does, what features are supported. The support for the overload control solution is based on the presence of the OC-Feature-Vector AVP in the Diameter answer for existing application. For the newly defined applications the support for the overload control MAY already be part of the application specification. Based on capability knowledge the request message initiating endpoint can select the preferred common traffic abatement algorithm and act accordingly for the subsequent message exchanges.

5.3.2. Answer Message Initiating Endpoint Considerations

When a remote endpoint (i.e. a "reporting node") receives a request message in can detect whether the request message initiating endpoint has support for the overload control solution based on the presence of the OC-Feature-Vector AVP. For the newly defined applications the overload control solution support can be part of the application specification. Based on the content of the OC-Feature-Vector AVP the request message receiving endpoint knows what overload control functionality the other endpoint supports and then act accordingly for the subsequent answer messages it initiates. ~~It is RECOMMENDED that the~~ **The** answer message initiating endpoint ~~selects one common traffic abatement algorithm even if~~ **MAY announce as many supported capabilities as** it ~~would support multiple.~~ **has (the announced set is a subject to local policy and configuration). However, at least one of the announced capabilities MUST be the same as received in the request message.**

The answer message initiating endpoint MUST NOT include any overload control solution defined AVPs into its answer messages if the request message initiating endpoint has not indicated support at the beginning of the the created session (or transaction in a case of non-session state maintaining applications). **The same also applies if none of the announced capabilities match between the two endpoints.**

5.4. Protocol Extensibility

The overload control solution can be extended, e.g. with new traffic abatement algorithms or new functionality. The new features and algorithms MUST be registered with the IANA and for the ~~ppossible~~ **possible** use with the OC-Feature-Vector for announcing the support for the new features (see Section 7 for the required procedures).

It should be noted that [RFC6733] defined Grouped AVP extension mechanisms also apply. This allows, for example, defining a new feature that is mandatory to understand even when piggybacked on an existing applications. More specifically, the sub-AVPs inside the OC-OLR AVP MAY have the M-bit set. However, when overload control AVPs are piggybacked on top of an existing applications, setting M-bit in sub-AVPs is NOT RECOMMENDED.

5.5. Overload Report Processing

5.5.1. ~~Sender Endpoint Considerations~~

~~5.5.2. Receiver Endpoint Considerations~~

~~[OpenIssue: did we now agree that e.g.~~ **Overload Control State**

Both reacting and reporting nodes MUST maintain an overload condition state for each endpoint (a host or a ~~server can refrain sending OLR in answers based on some magical algorithm? (Note: We seem to~~ **realm) they communicate with and both endpoints** have ~~consensus that a server MAY repeat OLRs in subsequent messages, but~~ **announced support for DOIC. See Sections 4.1 and 5.3 for discussion how the support for DOIC** is ~~not required to do so, based on local policy.)]~~

~~6. Transport Considerations~~

~~In order to reduce~~ **determined. The** overload ~~control introduced additional AVP and message processing it might~~ **condition state SHOULD** be ~~desirable/beneficial~~ **able** to ~~signal whether the Diameter command carries overload control information~~ **make a difference between a realm and a specific host in** that ~~should be of interest of an~~ **realm.**

The overload ~~aware Diameter node.~~

~~Should such indication be~~ **condition state SHOULD** include ~~is not part of this specification. It has not either been concluded~~ at ~~what layer such possible indication should be. Obvious candidates include transport layer protocols (e.g., SCTP PPID or TCP flags)~~ **least the following information (per host** or ~~Diameter command header flags.~~

~~7. IANA~~ **realm):**

- **The endpoint information (realm and/or DiameterIdentity, application identifier, etc)**
- **Reduction percentage**
- **Validity period timer**
- **Sequence number**
- **Supported/selected traffic abatement algorithm**

5.5.2. Reacting Node Considerations

~~7.1. AVP codes~~

~~New AVPs defined by this specification are listed in Section 4. All~~

Once a reacting node receives an OC-OLR AVP ~~codes allocated~~ from **a reporting node, it applies** the ~~'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.~~

~~7.2. New registries~~

~~Three new registries are needed under~~ **traffic abatement based on** the ~~'Authentication, Authorization,~~ **commonly supported algorithm with the reporting node** and ~~Accounting (AAA) Parameters' registry.~~

~~Section 4.1 defines a new "Overload Control Feature Vector" registry including~~ the ~~initial assignments. New values can be added into~~ **current overload condition. The reacting node learns** the ~~registry using~~ **reporting node supported abatement algorithms directly from** the ~~Specification Required policy [RFC5226].~~

~~Section 4.5 defines a new "Overload Report Type" registry~~ **received answer message containing the OC-Feature-Vector AVP or indirectly remembering the previously used traffic abatement algorithm** with ~~its initial assignments. New types can be added using~~ the ~~Specification Required policy [RFC5226].~~

~~8. Security Considerations~~

~~This mechanism gives Diameter nodes~~ **given reporting node.**

The received OC-Feature-Vector AVP does not change the ~~ability to request that downstream nodes send fewer Diameter requests. Nodes do this by exchanging~~ **existing** overload ~~reports~~ **condition and/or traffic abatement algorithm settings if the SequenceNumber AVP contains a value** that ~~directly affect this reduction. This exchange~~ is ~~potentially subject to multiple methods of attack, and has~~ **equal than** the ~~potential to be used as a Denial-of-Service (DoS) attack vector.~~

~~Overload reports may contain information about~~ **previously received/recorded one. If** the ~~topology and current status of a Diameter network. This information~~ **OC-Feature-Vector AVP** is ~~potentially sensitive. Network operators may wish to control disclosure of overload reports to unauthorized parties to avoid its use~~ **received** for ~~competitive intelligence or to target attacks.~~

~~Diameter does not include features to provide end-to-end authentication, integrity protection,~~ **the first time for the reporting node** or ~~confidentiality. This may cause complications when sending overload reports between non-adjacent nodes.~~

~~8.1. Potential Threat Modes~~

~~The Diameter protocol involves transactions in~~ the ~~form of requests and answers exchanged between clients and servers. These clients~~ **SequenceNumber value is less than the previously received/recorded one, then either the sequence number is stale (e.g. an intentional or unintentional replay)** and ~~servers may~~ **SHOULD** be ~~peers, that is,they may share~~ **silently discarded. A sequence number value less than** a ~~direct transport (e.g. TCP or SCTP) connection, or the messages may traverse~~ **previous** one ~~or~~ **MAY also indicate an overflow of the sequence number. [Editor's note: do we need** more ~~intermediaries, known as Diameter Agents. Diameter nodes use TLS, DTLS, or IPSec to authenticate peers, and~~ to ~~provide confidentiality and integrity protection of traffic between peers. Nodes can make authorization decisions based~~ **say here** on the ~~peer identities authenticated at~~ **overflow?]**

The OC-OLR AVP contains the ~~transport layer.~~

~~When agents are involved, this presents an effectively hop-by-hop trust model. That is, a Diameter client or server can authorize an agent for certain actions, but it must trust that agent~~ **necessary information of the overload condition on the reporting node. Similarly** to ~~make appropriate authorization decisions about its peers, and so on.~~

~~Since confidentiality and integrity protection occurs at~~ the ~~transport layer. Agents can read,~~ **OC-Feature-Vector's sequence numbering, the OC-OLD AVP also has the SequenceNumber AVP** and ~~perhaps modify, any part of a Diameter message, including an overload report.~~

~~There are several ways an attacker might attempt~~ **its handling is similar** to ~~exploit~~ the **one in the OC-Feature-Vector AVP. The reacting node MUST update its** overload ~~control mechanism. An unauthorized third party might inject an overload report into~~ **condition state whenever receiving** the ~~network. If this third party is upstream of an agent, and that agent fails to apply proper authorization policies, downstream nodes may mistakenly trust~~ **OC-OLR AVP for** the ~~report. This attack is at least partially mitigated by~~ **first time or** the ~~assumption that nodes include overload reports in Diameter answers but not~~ **SequenceNumber sub-AVP indicates a change** in ~~requests. This requires an attacker to have knowledge~~ **the OC-OLR AVP.**

Each OC-OLR AVP also contains the validity duration of the ~~original request~~

~~in order to construct a response. Therefore, implementations SHOULD~~
~~validate that an answer containing~~ **overload**
information either explicitly or implicitly. The reacting node MUST
maintain an overload ~~report is a properly~~
~~constructed response to a pending request prior to acting on~~ **condition validity timer for each reporting node**
(or realm) it communicates with. Once the validity duration times
out, the reacting node MUST assume the overload ~~report.~~

~~A similar attack involves an otherwise authorized Diameter~~ **condition has ended**
with the given reporting node ~~that~~
~~sends~~ **and discard all overload condition**
state information with the reporting node. The ValidityDuration AVP
with value 0 indicates an ~~inappropriate~~ **explicit expiration of the** overload ~~report. For example, a server~~
condition state for a giver "ReportType" (see next paragraph).

From the ReportType AVP the reacting node learns whether the ~~realm "example.com" might send an~~ overload
condition report ~~indicating that~~ **concerns** a ~~competitor's realm "example.net" is overloaded. If other nodes act~~
~~on~~ **specific host (as identified by** the ~~report, they may falsely believe that "example.net" is~~
~~overloaded, effectively reducing that realm's capacity. Therefore,~~
~~it's critical that nodes validate that an overload report received~~
~~from a peer actually falls within that peer's responsibility before~~
~~acting on~~
Origin-Host AVP of the ~~report~~ **answer message containing the OC-OLR AVP)** or ~~forwarding~~
the ~~report to other peers. For~~
~~example, an overload report from an peer that applies to a~~ **entire** realm ~~not~~
~~handled~~ **(as identified** by ~~that peer is suspect.~~

~~An attacker might use~~ the ~~information in an overload report to assist~~
~~in certain attacks. For example, an attacker could~~ **Origin-Realm AVP of the answer**
message containing the OC-OLR AVP). The reacting node SHOULD use
this information
~~about current overload conditions to time a DoS attack for maximum~~
~~effect, or use subsequent overload reports~~ as ~~a feedback mechanism to~~
~~learn~~ **an input for its traffic abatement algorithm.**

The idea is that the ~~results~~ **reacting node apply different handling** of **the**
traffic abatement, whether sent request messages are targeted to a ~~previous~~
specific host or ~~ongoing attack.~~

~~8.2. Denial of Service Attacks~~

~~Diameter overload reports can cause a node~~ to ~~cease sending some or~~
~~all Diameter requests for an extended period. This makes them a~~
~~tempting vector for DoS tacks. Furthermore, since Diameter is almost~~
~~always used~~ **any host** in ~~support of other protocols,~~ a ~~DoS attack on Diameter~~
~~is likely to impact those protocols as well. Therefore, Diameter~~
~~nodes~~ **realm.**

In the context of this specification and the default traffic
abatement algorithm, the Reduction-Percentage AVP value MUST ~~NOT honor or forward~~ **be**
interpreted in the following way:

value == 0

Indicates explicitly the end of overload ~~reports from unauthorized or~~
~~otherwise untrusted sources.~~

~~8.3. Non-Compliant Nodes~~

~~When a Diameter~~ **condition and the**
reacting node ~~sends an~~ **should not apply the traffic abatement algorithm**
procedures any more for the given reporting node (or realm). The
reacting node MAY still preserve the overload ~~report, it cannot assume~~ **condition state**
information with the given reacting node (or realm).

value == 100

Indicates that
~~all nodes will comply. A non compliant~~ **the reporting** node ~~might continue~~ **(or realm) does not want to** ~~send~~
~~requests with no reduction in load. Requirement 28~~
~~[I-D.ietf-dime-overload-reqs] indicates that~~
receive any traffic from the ~~overload control~~
~~solution cannot assume that~~ **reacting node for the application the**
report concerns. The reacting node MUST do all ~~Diameter nodes in a network are~~
~~necessarily trusted, and that malicious nodes~~ **measure** not ~~be allowed~~ to ~~take~~
~~advantage of~~
send traffic to the reporting node (or realm) as long as the
overload ~~control mechanism~~ **condition changes or expires.**

0 < value < 100

Indicates that the reporting node urges the reacting node to ~~get more than their~~
~~fair share of service.~~

~~In~~
reduce its traffic by a given percentage. For example if the ~~absence~~
reacting node has been sending 100 packets per second to the
reporting node, then a reception of ~~an overload control mechanism, Diameter nodes need~~ **Reduction-Percentage value of**
10 would mean that from now on the reacting node MUST only send 90
packets per second. How the reacting node achieves the "true
reduction" transactions leading to ~~implement strategies~~ **the sent request messages is up**
to ~~protect themselves~~ **the implementation. The reacting node MAY simply drop every**
10th packet from ~~floods of~~
~~requests,~~ **its output queue** and **let the generic application**
logic try to ~~make sure~~ **recover from it.**

5.5.3. Reporting Node Considerations

[OpenIssue: did we now agree that e.g. a ~~disproportionate load from one source does not prevent other sources from receiving service. For example, a Diameter~~ server ~~might reject a certain percentage of requests from sources that exceed certain limits. Overload control~~ can ~~be thought of as an optimization for such strategies, where downstream nodes never send the excess requests~~ refrain sending OLR in ~~the first place. However, the presence of an overload control mechanism does not remove the need for these other protection strategies.~~

~~8.4. End-to End-Security Issues~~

~~The lack of end-to-end security features makes it far more difficult~~ answers based on some magical algorithm? (Note: We seem to ~~establish~~ adjacent nodes. ~~Any agents in the message path may insert or modify overload reports. Nodes must trust~~ have consensus that ~~their adjacent peers perform proper checks on overload reports from their peers, and so on,~~ ~~creating~~ a ~~transitive trust requirement extending for potentially long chains of nodes. Network operators must determine if this transitive trust requirement~~ server MAY repeat OLRs in subsequent messages, but is ~~acceptable for their deployments. Nodes supporting Diameter overload control MUST give operators the ability~~ not required to ~~select which peers are trusted~~ do so, based on local policy.)]

6. Transport Considerations

In order to ~~deliver~~ reduce overload ~~reports,~~ control introduced additional AVP and ~~whether they are trusted~~ message processing it might be desirable/beneficial to ~~forward~~ signal whether the Diameter command carries overload ~~reports from non-adjacent nodes.~~

~~[OpenIssue: This requires~~ control information that ~~a responding node~~ should be ~~able to tell a peer-generated OLR from one generated by a non-adjacent node. One way~~ of ~~doing this would~~ interest of an overload aware Diameter node.

Should such indication be ~~to~~ include ~~the identity~~ is not part of ~~the node that generated the report as part of the OLR]~~

~~[OpenIssue: Do we need further language about~~ this specification. It has not either been concluded at what ~~rules an agent~~ layer such possible indication should ~~apply before forwarding an OLR?]~~

~~The lack of end to end protection creates a tension between two requirements~~ be. Obvious candidates include transport layer protocols (e.g., SCTP PPID or TCP flags) or Diameter command header flags.

7. IANA Considerations

7.1. AVP codes

New AVPs defined by this specification are listed in Section 4. All AVP codes allocated from the ~~overload control requirements document. [I-D.ietf-dime-overload-reqs] Requirement 34 requires the ability to send overload reports across intermediaries (i.e. agents) that do not support overload control mechanism. Requirement 27 forbids~~ 'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.

7.2. New registries

Three new registries are needed under the ~~mechanism from adding~~ 'Authentication, Authorization, and Accounting (AAA) Parameters' registry.

Section 4.2 defines a new ~~vulnerabilities or increasing~~ "Overload Control Feature Vector" registry including the ~~severity of existing ones. A non supporting agent will most likely forward overload reports without inspecting them or applying any sort of validation or authorization. This makes~~ initial assignments. New values can be added into the ~~transitive trust issue considerably more of~~ registry using the Specification Required policy [RFC5226]. See Section 4.2 for the initial assignment in the registry.

Section 4.6 defines a ~~problem. Without~~ new "Overload Report Type" registry with its initial assignments. New types can be added using the Specification Required policy [RFC5226].

8. Security Considerations

This mechanism gives Diameter nodes the ability to ~~authenticate and integrity protect~~ request that downstream nodes send fewer Diameter requests. Nodes do this by exchanging overload reports ~~across a non-supporting agent,~~ that directly affect this reduction. This exchange is potentially subject to multiple methods of attack, and has the ~~mechanism cannot comply with both requirements.~~

~~[OpenIssue: What do we want~~ potential to ~~do about this? Req27 is~~ be used as a ~~normative MUST, while Req34 is "merely"~~ Denial-of-Service (DoS) attack vector.

Overload reports may contain information about the topology and current status of a ~~SHOULD.~~ Diameter network. This ~~would seem to imply that 27 has~~ information is potentially sensitive. Network operators may wish to ~~take precedent. Can we say that~~ control disclosure of overload reports ~~MUST NOT be sent~~ to ~~and/or accepted from non-supporting~~

~~agents until such time we can~~ unauthorized parties to avoid its use for competitive intelligence or to target attacks.

Diameter does not include features to provide end-to-end ~~security.]~~ authentication, integrity protection, or confidentiality. This may cause complications when sending overload reports between non-adjacent nodes.

8.1. Potential Threat Modes

The ~~lack of end-to-end confidentiality protection means that any~~ Diameter ~~agent~~ protocol involves transactions in the ~~path of an overload r~~ ~~contents~~ form of ~~that report. In addition to the requirement to select~~ ~~which peers are trusted to send overload reports, operators MUST~~ requests and answers exchanged between clients and servers. These clients and servers may be ~~able to select which peers are authorized to receive reports. A node~~ ~~MUST not send an overload report to a peer not authorized to receive~~ ~~it. Furthermore, an agent MUST remove any overload reports~~ peers, that ~~might have been inserted by other nodes before forwarding~~ is,they may share a direct transport (e.g. TCP or SCTP) connection, or the messages may traverse one or more intermediaries, known as Diameter ~~message~~ Agents. Diameter nodes use TLS, DTLS, or IPSec to ~~a peer that is not authorized~~ authenticate peers, and to ~~receive overload reports.~~

~~At the time~~ provide confidentiality and integrity protection of ~~this writing,~~ traffic between peers. Nodes can make authorization decisions based on the ~~DIME working group is studying~~ ~~requirements~~ peer identities authenticated at the transport layer.

When agents are involved, this presents an effectively hop-by-hop trust model. That is, a Diameter client or server can authorize an agent for ~~adding end-to-end security~~ ~~[I-D.ietf-dime-e2e-sec-req] features~~ certain actions, but it must trust that agent to ~~Diameter. These features,~~ ~~when they become available, might~~ make ~~it easier~~ appropriate authorization decisions about its peers, and so on.

Since confidentiality and integrity protection occurs at the transport layer. Agents can read, and perhaps modify, any part of a Diameter message, including an overload report.

There are several ways an attacker might attempt to ~~establish~~ ~~trust in non-adjacent nodes for~~ exploit the overload control ~~purposes.~~ ~~Readers should be reminded, however,~~ mechanism. An unauthorized third party might inject an overload report into the network. If this third party is upstream of an agent, and that agent fails to apply proper authorization policies, downstream nodes may mistakenly trust the report. This attack is at least partially mitigated by the assumption that nodes include overload ~~control~~ ~~mechanism encourages~~ reports in Diameter ~~agents~~ answers but not in requests. This requires an attacker to ~~modify AVPs in, or insert~~ ~~additional AVPs into, existing messages~~ have knowledge of the original request in order to construct a response. Therefore, implementations SHOULD validate that ~~are originated by~~ ~~other nodes. If end-to-end security is enabled, there~~ an answer containing an overload report is a ~~risk~~ properly constructed response to a pending request prior to acting on the overload report.

A similar attack involves an otherwise authorized Diameter node that ~~such modification could violate integrity protection. The~~ ~~details of using any future Diameter end-to-end security mechanism~~ ~~with~~ sends an inappropriate overload ~~control will require careful consideration, and are~~ ~~beyond~~ report. For example, a server for the ~~scope of this document.~~

~~9. Contributors~~

~~The following people contributed substantial ideas, feedback, and~~ ~~discussion~~ realm "example.com" might send an overload report indicating that a competitor's realm "example.net" is overloaded. If other nodes act on the report, they may falsely believe that "example.net" is overloaded, effectively reducing that realm's capacity. Therefore, it's critical that nodes validate that an overload report received from a peer actually falls within that peer's responsibility before acting on the report or forwarding the report to ~~this document:~~

~~o Eric McMurry~~

~~o Hannes Tschofenig~~

~~o Ulrich Wiehe~~

~~o Jean-Jacques Trottin~~

~~o Lionel Morand~~

~~o Maria Cruz Bartolome~~

~~o Martin Dolly~~

~~o Nirav Salot~~

~~o Susan Shishufeng~~

~~10. Acknowledgements~~

~~...~~

~~11. References~~

~~11.1. Normative References~~

~~[RFC2119] Bradner, S., "Key words for~~ **other peers. For example, an overload report from an peer that applies to a realm not handled by that peer is suspect.**

An attacker might use **the information** in ~~RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.~~

~~[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing~~ an ~~IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.~~

~~[RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.~~

~~11.2. Informative References~~

~~[I-D.ietf-dime-e2e-sec-req] Tschofenig, H., Korhonen, J., Zorn, G., and K. Pillay, "Diameter AVP Level Security: Scenarios and Requirements", draft-ietf-dime-e2e-sec-req-00 (work~~ **overload report to assist** in ~~progress), September 2013.~~

~~[I-D.ietf-dime-overload-reqs] McMurry, E. and D. Campbell, "Diameter Overload Control Requirements", draft-ietf-dime-overload-reqs-13 (work in progress), September 2013.~~

~~[RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, "Diameter Credit-Control Application", RFC 4006, August 2005.~~

~~Appendix A. Issues left for future specifications~~

~~The base solution for the overload control does not cover all possible~~ **certain attacks. For example, an attacker could** use ~~cases. A number of solution aspects were intentionally left for future specification and protocol work.~~

~~A.1. Additional traffic abatement algorithms~~

~~This specification describes only means for a simple loss based algorithm. Future algorithms can be added using the designed solution extension mechanism. The new algorithms need~~ **information about current overload conditions** to ~~be registered with IANA. See Sections 4.1 and 7~~ **time a DoS attack** for ~~the required IANA steps.~~

~~A.2. Agent Overload~~

~~This specification focuses on Diameter end-point (server~~ **maximum effect,** or ~~client) overload. A separate extension will be required~~ **use subsequent overload reports as a feedback mechanism** to ~~outline the handling~~ **learn** the ~~case~~ **results** of ~~agent overload.~~

~~A.3. DIAMETER_TOO_BUSY clarifications~~

~~The current [RFC6733] behaviour in~~ a ~~case~~ **previous or ongoing attack.**

8.2. Denial of ~~DIAMETER_TOO_BUSY is somewhat underspecified. For example, there is no information how long the specific~~ **Service Attacks**

Diameter **overload reports can cause a** node ~~is willing~~ to ~~be unavailable. A specification updating [RFC6733] should clarify the handling of DIAMETER_TOO_BUSY from the error answer initiating~~ **cease sending some or all** Diameter ~~node point of view and from the original request initiating~~ **requests for an extended period. This makes them a tempting vector for DoS tacks. Furthermore, since** Diameter ~~node point~~ **is almost always used in support** of ~~view. Further, the inclusion of possible additional information providing AVPs should be discussed and possible be recommended to be used.~~

~~Appendix B. Conformance to Requirements~~

~~The following section analyses, which~~ **other protocols, a DoS attack on** Diameter ~~Overload Control requirements [I-D.ietf-dime-overload-reqs] are met by this specification.~~

~~Key:~~

~~S - Supported~~

~~P - Partial~~

~~N - Not supported~~

~~+ + + +~~
~~| Rqmt | S/ | Notes |~~
~~| # | P/ | |~~

	N	
REQ 1	P	The DOIC solution only addresses overload information. Load information is left likely to impact those protocols as future work. In addition, the DOIC solution does not address agent overload scenarios.
REQ 2	P	The DOIC solution supports well. Therefore, Diameter nodes MUST NOT honor or forward overload reports from unauthorized or otherwise untrusted sources.

8.3. Non-Compliant Nodes

When a Diameter node sends an overload report, it cannot assume that all nodes will comply. A non-compliant node might continue to send requests with no reduction in load. Requirement 28 [RFC7068] indicates that implicitly indicate the application impacted by the report. The DOIC overload control solution does cannot assume that all Diameter nodes in a network are necessarily trusted, and that malicious nodes not support reporting load information. The DOIC solution is thought be allowed to support graceful behavior. Allowing an application specific capabilities negotiation mechanism violates application-independence. Suggested different wording: The DOIC solution supports take advantage of the overload reports that are applicable control mechanism to any get more than their fair share of service.

In the absence of an overload control mechanism, Diameter application. The DOIC solution does not support reporting load information. The DOIC solution allows nodes need to support graceful behavior; this will be enhanced when the Load information will be defined. Comment: Can we removed the words "is thought"?

REQ 3	S	The DOIC solution is thought implement strategies to address this requirement. Comment: Can we removed the words "is thought"?
REQ 4	P	The DOIC solution protect themselves from floods of requests, and to make sure that a disproportionate load from one source does allow for both both not prevent other sources from receiving service. For example, a Diameter server and might reject a Diameter client to certain percentage of requests from sources that exceed certain limits. Overload control can be thought of as an optimization for such strategies, where downstream nodes never send overload reports. The DOIC solution only addresses Diameter end-point (server and client) overload. Agent overload is being addressed the excess requests in a separate draft.
REQ 5	S	The DOIC solution the first place. However, the presence of an overload control mechanism does not depend on how remove the end-points are discovered. Comment: it might be worth working through at least one use case showing DNS based dynamic peer discovery to make sure we haven't missed anything.
REQ 6	?	Need to update text as some configuation is required. Need need for these other protection strategies.

8.4. End-to End-Security Issues

The lack of end-to-end security features makes it far more difficult to determin if establish trust in overload reports that originate from non-adjacent nodes. Any agents in the current discussion message path may insert or modify overload reports. Nodes must trust that their adjacent peers perform proper checks on overload application id increases the amount of configuration which would change this to reports from their peers, and so on, creating a N.

REQ 7	S	The DOIC solution supports the loss algorithm, which is expected to address transitive-trust requirement extending for potentially long chains of nodes. Network operators must determine if this requirement. There transitive trust requirement is concern about acceptable for their deployments. Nodes supporting Diameter overload control MUST give operators the ability to address oscillations. Wording is included for how a reacting node starts select which peers are trusted to increase traffic after an deliver overload report expires reports, and whether they are trusted to address this concern. Suggested different wording: forward overload reports from non-adjacent nodes.

The DOIC solution supports a baseline mechanism relying on traffic reduction percentage lack of end-to-end confidentiality protection means that is a loss algorithm, which allows to address this requirement. Oscillations are avoided or quite minimized by sending successive OLR reports with any Diameter agent in the values to converge path of an overload report can view the

contents of that report. In addition to the optimal traffic or requirement to
smoothly come back select
which peers are trusted to normal traffic conditions when
send overload decreases and ends.
REQ ? The DOIC solution supports a timestamp reports, operators MUST be
able to select which is meant
8 peers are authorized to serve as a receive reports. A node
MUST not send an overload report version indication to address
this requirement. Comment: The use of the timestamp
is under discussion.
REQ ? The DOIC solution uses a piggybacking strategy for
9 carrying overload reports, which scales lineraly with
the amount of traffic. As such, the first part of
the requirement is addressed. The DOIC solution does
peer not support a mechanism for sending authorized to receive
it. Furthermore, an agent MUST remove any overload reports
over that
might have been inserted by other nodes before forwarding a quiescent transport connections or, more
generally, to Diameter nodes
message to a peer that are is not producing
traffic. Suggested different wording: The DOIC
solution uses a piggybacking strategy for carrying
authorized to receive overload reports. As such,

At the first part time of this writing, the
requirement DIME working group is addressed. For a connection that has
studying
requirements for adding end-to-end security
[I-D.ietf-dime-e2e-sec-req] features to Diameter. These features,
when they become quiescent due available, might make it easier to OLRs with a 100% traffic
reduction, establish trust
in non-adjacent nodes for overload control purposes. Readers should
be reminded, however, that the validity timer allows overload control mechanism encourages
Diameter agents to handle this
case. Other cases of quiescent connections modify AVPs in, or insert additional AVPs into,
existing messages that are
outside the scope originated by other nodes. If end-to-end
security is enabled, there is a risk that such modification could
violate integrity protection. The details of using any future
Diameter end-to-end security mechanism with overload (e.g. their
handling may be done through control will
require careful consideration, and are beyond the watch dog scope of the
Diameter base protocol).
REQ S this
document.

9. Contributors

The DOIC solution supports two methods following people contributed substantial ideas, feedback, and
discussion to this document:

- Eric McMurry
- Hannes Tschofenig
- Ulrich Wiehe
- Jean-Jacques Trottin
- Lionel Morand
- Maria Cruz Bartolome
- Martin Dolly
- Nirav Salot
- Susan Shishufeng

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for managing
10 the length of an overload condition. First, all
overload reports must contain a duration indication,
after which the node reacting use in RFCs to the report can
consider the overload condition as ended. Secondly,
the solution supports the method Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for the node
originating the overload report to explicitly
communicate that the condition has ended. This
latter mechanism depends on traffic to be sent from
the reacting node and, as such, can not be depended
upon Writing an
IANA Considerations Section in all circumstances.
REQ ? The DOIC solution works well for small network
11 configurations RFCs", BCP 26, RFC 5226,
May 2008.

[RFC6733] Fajardo, V., Arkko, J., Loughney, J., and for network configurations with a
single Diameter agent hop. More analysis is required

to determine how well G. Zorn,
"Diameter Base Protocol", RFC 6733, October 2012.

10.2. Informative References

[3GPP.23.203]
3GPP, "Policy and charging control architecture", 3GPP TS 23.203 10.9.0, September 2013.

[3GPP.29.229]
3GPP, "Cx and Dx interfaces based on the DOIC solution handles very large Diameter network with partitioned or segmented server farms requiring multiple hops through Diameter agents.

REQ P The DOIC solution focuses
protocol; Protocol details", 3GPP TS 29.229 10.5.0, March 2013.

[3GPP.29.272]
3GPP, "Evolved Packet System (EPS); Mobility Management Entity (MME) and Serving GPRS Support Node (SGSN) related interfaces based on Diameter end point

12 overload protocol", 3GPP TS 29.272
10.8.0, June 2013.

[I-D.ietf-dime-e2e-sec-req]
Tschofenig, H., Korhonen, J., Zorn, G., and meets this requirement for those Diameter nodes. The DOIC solution does not address Diameter Agent overload K. Pillay,
"Diameter AVP Level Security: Scenarios and does not meet this requirement for those Diameter nodes.

REQ ? The DOIC solution requires including of the overload
13 report in all answer messages in some situations. It is not agreed, however, that this constitutes substantial work. This can also be mitigated by the sender of the overload report keeping state to record who has received overload reports. It is left to implementation decisions as to which approach is taken send in all messages or send once with a record of who has received the report. Another way is to let the request sender (reacting node) insert information in the request to say whether a throttling is actually performed. The reporting node then can base its decision on information received in the request; no need for keeping state to record who has received overload reports. The DOIC solution also requires capabilities negotiation Requirements",
draft-ietf-dime-e2e-sec-req-00 (work in every request progress),
September 2013.

[RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and response message, which increases the baseline work required for any node supporting the DOIC solution. Suggested additional text: It does not, however, require that the information be recalculated or updated with each message. The update frequency is up to the implementation, and each implementation can make decisions on balancing the update of overload information along with its other priorities. It is expected that using a periodically updated OLR report added to all messages sent to overload control endpoints will not add substantial additional work. Piggyback base transport also does not require composition, sending, or parsing of new Diameter messages for the purpose of conveying overload control information. There is still discussion on the substantial additional work due to have OLR in each answer message.

REQ S The DOIC solution uses the piggybacking method to
14 deliver overload report, which scales lineraly with the amount of traffic. This allows for immediate feedback to any node generating traffic toward another overloaded node.

REQ S The DOIC solution does not interfere with transport
15 protocols.

REQ ? The DOIC solution allows for a mixed network of
16 supporting and non supporting Diameter end-points. It isn't clear how realm overload is handled in a network with agents that do not support the DOIC solution. Suggested additional wording: Evaluation of Realm overload may require a DA supporting DOIC, if the realm overload is not evaluated by the client. Realm overload handling is still under discussion.

REQ ? Suggested wording: The DOIC solution addresses this
17 requirement through the loss algorithm (DOIC baseline mechanism) with the following possibilities. A DA supporting DOIC can act on behalf of clients not supporting DOIC. A reporting node is also aware of the nodes not supporting the DOIC as there is no advertisement of the DOIC support. It may then apply a particular throttling of the requests coming from

~~these non supporting DOIC clients.~~

~~REQ 18 ? It isn't clear yet that if this requirement is addressed. There has been a proposal to mark messages that survived overload throttling as one method for an overloaded node to address fairness but this proposal is not yet part of the solution. It is also possible that the overloaded node could use state gathered as part of the capability advertisement mechanism to know if the sending node supports the DOIC solution and if not, to apply a particular throttling of the requests coming from these non supporting DOIC clients.~~

~~REQ 19 S The DOIC solution supports the ability for the overloaded node and the reacting node to be in different administrative domains.~~

~~REQ 20 ? This mechanism is still under discussion. Comment 1: I think this is a "S". OLRs are clearly distinguishable from any error code. The fact that an agent would need to send errors if it throttles is not an overload indication per se. It needs to do that even without DoC. OTOH, if we apply some DOC related fix to TOO_BUSY, we probably need a new code. Comment 2: New AVPs conveys overload control information, and this is transported on existing answer messages, so distinguishable from Diameter errors.~~

~~REQ 21 S The inability for a node to send overload reports will result in equivalent through put to a network that does not support the DOIC solution.~~

~~REQ 22 S The DOIC solution gives this node generating the overload report the ability to control the amount of throttling done by the reacting node using the reduction percentage parameter in the overload report.~~

~~REQ 23 ? Initial text: The DOIC mechanism supports two abatement strategies by reacting nodes, routing to an alternative node or dropping traffic. The routing to an alternative node will be enhanced when the Load extension is defined. Comment: This is a N. There's no good way to determine which nodes are likely to have sufficient capacity without some sort of load metric for non-overloaded nodes.~~

~~REQ 24 N The DOIC solution does not address delivering load information.~~

~~REQ 25 S The DOIC solution contains some guideance.~~

~~REQ S~~ J. Loughney, "Diameter Credit-Control Application", RFC 4006, August 2005.

[RFC7068] McMurry, E. and B. Campbell, "Diameter Overload Control Requirements", RFC 7068, November 2013.

Appendix A. Issues left for future specifications

The ~~DOIC~~ base solution for the overload control does not ~~constrain a nodes ability to determine which requests are trottled.~~

~~REQ 26~~

~~REQ ? Initial text: The DOIC~~ cover all possible use cases. A number of solution ~~does add~~ aspects were intentionally left for future specification and protocol work.

A.1. Additional traffic abatement algorithms

This specification describes only means for a simple loss based algorithm. Future algorithms can be added using the designed solution extension mechanism. The new ~~line~~ algorithms need to be registered with IANA. See Sections 4.1 and 7 for the required IANA steps.

~~REQ 27~~

A.2. Agent Overload

This specification focuses on Diameter end-point (server or client) overload. A separate extension will be required to outline the handling the case of ~~attack~~ agent overload.

A.3. DIAMETER_TOO_BUSY clarifications

The current [RFC6733] behaviour in ~~the ability for~~ a ~~malicious entity to insert overload reports that would reduce or eliminate traffic. This, however,~~ case of DIAMETER_TOO_BUSY is ~~no worse than an attacker that would assert erroneous error responses such as a TOO_BUSY response. It~~ somewhat under specified. For example, there is ~~recognized that the end to end security solution currently being worked on by~~ no information how long the ~~DIME working group~~

specific Diameter node is needed willing to close these types of vulurabilities. Comment: Sending a malicious OLR with a type be unavailable. A specification updating [RFC6733] should clarify the handling of "realm" will have considerably more impact than a TOO_BUSY. Personally, I don't think we can achieve this requirement without either being hop-by-hop or requiring e2e security. We probably need further analysis DIAMETER_TOO_BUSY from the error answer initiating Diameter node point of view and from the security implications original request initiating Diameter node point of view. Further, the capabilities negotiation as well. Suggested inclusion of possible additional verbage: An OLR only relates information providing AVPs should be discussed and possible be recommended to be used.

Appendix B. Examples

B.1. Mix of Destination-Realm routed requests and Destination-Host routed requests

Diameter allows a client to optionally select the traffic between destination server of a reporting node request, even if there are agents between the client and the server. The client does this using the Destination-Host AVP. In cases where the client does not care if a reacting node and specific server receives the request, it can omit Destination-Host and route the request using the Destination-Realm and Application Id, effectively block letting an agent select the traffic from server.

Clients commonly send mixtures of Destination-Host and Destination-Realm routed requests. For example, in an application that uses user sessions, a client typically won't care which would be an important impact. Nevertheless OLRs are regularly sent in all answers, so a malicious OLR will have a short transient effect, as quickly overridden by a new OLR. To have a significant impact would require a continuous flow of answers with malicious OLRs. There server handles a session-initiating requests. But once the session is initiated, the exception of client will send all subsequent requests in that session to the OLR same server. Therefore it would send the initial request with no Destination-Host AVP. If it receives a successful answer, the client would copy the Origin-Host value of 100% reduction traffic which has from the answer message into a higher vulnerability and Destination-Host AVP in each subsequent request in the use of which should be avoided when possible. In addition such malicious OLRs must be session.

An agent has very limited options in answers, which means applying overload abatement to requests that contain Destination-Host AVPs. It typically cannot route the capability request to insert a different server than the malicious OLR one identified in an existing answer rather than Destination-Host. It's only remaining options are to throttle such requests locally, or to create send an answer which overload report back towards the client so the client can throttle the requests. The second choice is much less easy than usually more efficient, since it prevents any throttled requests from being sent in the first place, and removes the agent's need to create a send errors back to the client for each dropped request. To have a network wide applicability would request

On the other hand, an agent has much more leeway to generate malicious OLRs messages towards apply overload abatement for requests that do not contain Destination-Host AVPs. If the agent has multiple servers in its peer table for the given realm and application, it can route such requests to other, less overloaded servers.

If the overload severity increases, the agent may reach a point where there is not sufficient capacity across all reacting nodes. It servers to handle even realm-routed requests. In this case, the realm itself can be considered that overloaded. The agent may need the baseline mechanism offer a relevant level of security. Further analysis with a security expertise would client to throttle realm-routed requests in addition to Destination-Host routed requests. The overload severity may be beneficial.

REQ 28	?	- See REQ 18 Guidance may	different for each server, and REQ 27. Suggested additional verbage: the severity for the realm at is likely to be provided different than for detection of non compliant/abnormal use

any specific server. Therefore, an agent may need to forward, or originate, multiple overload reports with differing ReportType and Reduction-Percentage values.

Figure 8 illustrates such a mixed-routing scenario. In this example, the servers S1, S2, and S3 handle requests for the realm "realm". Any of OLRs, not only by endpoints but also by intermediate DA that the three can be aware handle requests that are not part of OLRs, an example being edge DAs with external networks. Further analysis with a security expertise would be beneficial. user

session (i.e. routed by Destination-Realm). But once a session is established, all requests in that session must go to the same server.

Figure 8: Mix of Destination-Host and Destination-Realm Routed Requests

1. The client sends a request with no Destination-Host AVP (that is, a Destination-Realm routed request.)

2. The agent follows local policy to select a server from its peer table. In this case, the agent selects S2 and forwards the request.

3. S1 is overloaded. It sends a answer indicating success, but also includes an overload report. Since the overload report only applies to S1, the ReportType is "Destination-Host".

4. The agent sees the overload report, and records that S1 is overloaded by the value in the Reduction-Percentage AVP. It begins diverting the indicated percentage of realm-routed traffic from S1 to S2 and S3. Since it can't divert Destination-Host routed traffic, it forwards the overload report to the client. This effectively delegates the throttling of traffic with Destination-Host:S1 to the client.

5. The client sends another Destination-Realm routed request.

6. The agent selects S2, and forwards the request.

7. It turns out that S2 is also overloaded, perhaps due to all that traffic it took over for S1. S2 returns an successful answer containing an overload report. Since this report only applies to S2, the ReportType is possible) or drop/reject requests. "Destination-Host".

8. The agent sees that S2 is also overloaded by the value in Reduction-Percentage. This DOIC baseline solution value is a loss algorithm and DOIC should not require further specification. probably different than the value from S1's report. The answer agent diverts the remaining traffic to REQ32 indicates S3 as best as it can, but it calculates that the possibility remaining capacity across all three servers is no longer sufficient to add other algorithms on top handle all of the DOIC baseline solution. The DOIC solution currently defines this loss algorithm as realm-routed traffic. This means the default algorithm. It realm itself is still under discussion to make it as mandatory to implement.

REQ 34	P	overloaded.	The ability to communicate realm's overload reports between supporting Diameter nodes does not require agents to support the DOIC solution. Load information exchange percentage is not currently defined.

Table 1

Appendix C. Examples

C.1. 3GPP S6a interface overload indication

[TBD: Would cover S6a MME-HSS communication with several topology choices (such as with most likely different than that for either S1 or without DRA, S2. The agent forward's S2's report back to the client in the Diameter answer. Additionally, the agent generates a new report for the realm of "realm", and inserts that report into the answer. The client throttles requests with "generic" agents).]

C.2. 3GPP PCC interfaces overload indication

[TBD: Would cover Gx/Rx and maybe S9..]

C.3. Mix of Destination Realm routed Destination-Host:S1 at one rate, requests with Destination-Host:S2 at another rate, and Destination-Host rerouted requests

[TBD: Add example showing the use of with no Destination-Host type OLRs and Realm type OLRs.] AVP at yet a third rate. (Since S3 has not indicated overload, the client does not throttle requests with Destination-Host:S3.)

Authors' Addresses

Jouni Korhonen (editor)
Broadcom
Porkkalankatu 24
Helsinki FIN-00180
Finland

Email: jouni.nospam@gmail.com

Steve Donovan
Oracle
17210 Campbell Road
Dallas, Texas 75254
United States

Email: srdonovan@usdonovans.com

Ben Campbell
Oracle
17210 Campbell Road
Dallas, Texas 75254
United States

Email: ben@nostrum.com