Diameter Maintenance and Extensions (DIME)
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2014

S. Donovan
Oracle

February 14, 2014

Diameter Agent Overload
draft-donovan-dime-agent-overload-01.txt

Abstract

This specification documents an extension to the Diameter Overload Control (DOC) base solution. The extension addresses the handling of agent overload.

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 August 18, 2014.

Copyright Notice

Table of Contents

1. Introduction

This document defines the behavior of Diameter nodes when Diameter agents become overloaded.

The base Diameter overload specification [I-D.ietf-dime-ovli] addresses the handling of overload when a Diameter endpoint (a Diameter Client or Diameter server as defined in [RFC6733]) becomes overloaded.

In the base specification, the goal is to react to the overload as close to the generator of the Diameter traffic as is feasible. When possible this is done at the originator of the traffic, generally referred to as a Diameter Client. A Diameter agent can also handle the overload mitigation. For instance, a Diameter agent might handle Diameter overload mitigation when it knows that a Diameter client does not support the DOIC extension.

This document extends the base Diameter endpoint overload specification to address the case when Diameter agents become overloaded. Just as is the case with other Diameter nodes -- clients and servers -- surges in Diameter traffic can cause a Diameter agent to be asked to handle more Diameter traffic than it was configured to handle. For a more detailed discussion of what can cause the overload of Diameter nodes, refer to the Diameter Overload Requirements [RFC7068].

This document builds on the "Loss" overload mitigation algorithm defined in [I-D.ietf-dime-ovli]. The handling of endpoint overload and agent overload is very similar. The primary differences are the following:

- Endpoint overload is handled as close to the originator of the traffic as possible.
- Agent overload is handled by the previous hop that supports this extension.
- Endpoint overload mitigation deals with traffic targeted for a single Diameter application. As such, it is assumed that an overload report impacts just the application implied by the message carrying the overload report.
- Agent overload deals with all traffic targeted for an agent, independent of the application. As such, a single agent overload report can impact multiple applications.

2. Terminology and Abbreviations

Editors note - These definitions need to be made consistent with the base Diameter overload specification defined in [I-D.ietf-dime-ovli].

Diameter Node

A RFC6733 Diameter Client, an RFC6733 Diameter Server, and RFC6733 agent.

Diameter Endpoint

An RFC6733 Diameter Client and RFC6733 Server.

Diameter Overload Endpoint

A Diameter node that supports the Diameter Overload extension defined in [I-D.ietf-dime-ovli].

Diameter Overload Reporting Node

A Diameter overload endpoint that sends and overload report in Diameter answer message.

Diameter Overload Reacting Node

A Diameter overload endpoint that receives and acts on a Diameter overload report.

3. Diameter Agent Overload Use Cases

The agent overload extension must support following use cases.

3.1. Single Agent

This use case is illustrated in Figure 1. In this case, the client sends all traffic through the single agent. If there is a failure in the agent then the client is unable to send Diameter traffic toward the server.

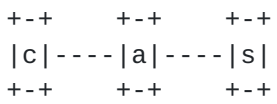

Figure 1

A more likely case for the use of agents is illustrated in Figure 2. In this case, there are multiple servers behind the single agent. The client sends all traffic through the agent and the agent determines how to distribute the traffic to the servers based on local routing and load distribution policy.

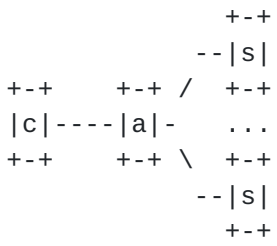

Figure 2

In both of these cases, the occurrence of overload in the single agent must by handled by the client in a similar fashion as if the client were handling the overload of a directly connected server. When the agent becomes overloaded it will insert an agent overload report in answer messages flowing to the client. This overload report will contain a requested reduction in the amount of traffic being sent to the agent. The client will apply overload abatement behavior as defined in the base Diameter overload specification [I-D.ietf-dime-ovli]. This will result in the requested percentage of the requests that would have been sent to the agent being dropped with the appropriate indication given to the service request that resulted in the need for the Diameter transaction.

NOTE: At this time there is a single overload abatement algorithm defined. In the even that multiple algorithms are defined then the abatement logic applied by the client will be based on the behavior indicated in the capability exchange.

Editors note: This might be changing in the base DOIC specification. If this happens then the change will need to be reflected here.

3.2. Redundant Agents

Figure 3 and Figure 4 illustrate a second, and more likely, type of deployment scenario involving agents. In both of these cases, the client has connections to two agents.

Figure 3 illustrates a client that has a primary connection to one of the agents (agent a1) and a secondary connection to the other agent (agent a2). In this scenario, the client will use the primary connection for all traffic. The secondary connection is used when

there is a failure scenario of some sort.

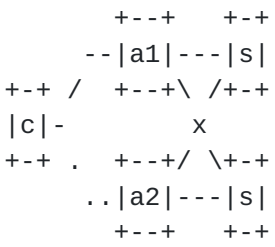

Figure 3

The second case, in Figure 4, illustrates the case where the connections to the agents are both actively used. In this case, the client will have a local distribution policy to determine the percentage of the traffic sent through each client.

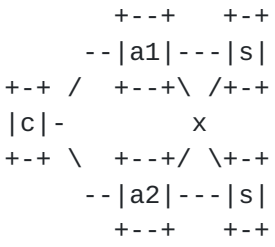

Figure 4

In the case where a single agent in the above scenarios become overloaded, the client should reduce the amount of traffic sent to the overloaded agent by the amount requested. This traffic should instead be routed through the non-overloaded agent. For example, assume that the overloaded agent requests a reduction of 10 percent. The client should send 10 percent of the traffic that would have been routed to the overloaded agent through the non-overloaded agent.

In the case where both agents are reporting overload, the client will need to start decreasing the total traffic sent to the agents. This would be done in a similar fashion as discussed in section 3.1. The amount of traffic depends on the combined reduction requested by the two agents.

3.3. Agent Chains

There are also deployment scenarios where there can be multiple agents between clients and servers. Examples of this type of deployment include when there are edge agents between Diameter networks. Another example of this type of deployment is when there are multiple sets of servers, each supporting a subset of the Diameter traffic.

Figure 5 illustrates one such network deployment case. Note that while this figure shows a maximum of two agents being involved in a Diameter transaction, it is possible that more than two agents could be in the path of a transaction.

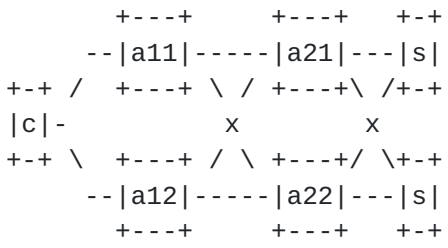

Figure 5

Handling of overload of one or both of agents a11 or a12 in this case is equivalent to that discussed in section 2.2.

Overload of agents a21 and a22 must be handled by the previous hop agents. As such, agents a11 and a12 must handle the overload mitigation logic when receiving an agent overload report from agents a21 and a22.

Editor's note: Probably need to elaborate the reasoning behind the need for the agent overload report being handled by the previous hop agent.

The handling of the overload reports is similar to that discussed in section 2.2. If the overload can be addressed by adjusting the amount of traffic sent to the next hop agents, then this approach should be taken.

If both of the agents have requested a reduction in traffic then the previous hop agent must start throttling the appropriate percentage of transactions. When throttling requests, the agent must use the same mechanism as defined in the base overload specification

[I-D.ietf-dime-ovli].

3.4. Interaction between agent and end-point overload

It is possible that both an agent and a server in the path of a transaction can be overloaded at the same time. When this occurs, Diameter entities will need to handle both overload reports. When this occurs the reacting node should first handle the throttling of the overloaded end-point. Any messages that survive that throttling should then be throttled (or routed) based on the reduction requested in the agent overload report.

4. Agent Overload Report

Editors Note: This section depends upon the base Diameter Overload specification. As such, it cannot be complete until the data model and extension mechanism are finalized in the based DOC specification. Details for any new AVPs or modifications to existing AVPs will be added in a future version of the draft after the base DOIC specification has stabilized.

4.1. OC-Supported-Features AVP

This extension adds a new feature to the OC-Feature-Vector AVP. This feature indication shows support for handling of peer overload reports. Peer overload reports are used by agents to indicate the need for overload abatement handling but the agents peer.

OLR_PEER_REPORT (0x0000000000000010)

> When this flag is set by a reacting endpoint it indicates that the endpoint supports the peer overload report type and, as a result, that the endpoint supports handling of agent overload.

A supporting node must also include the OC-Peer-Node-ID AVP in the OC-Supported-Features capability advertisemnt AVP.

This AVP contains the Diameter Identity of the node that supports the PEER overload report type. This AVP is used to determine if support for the peer overload report is in an adjectent node. The value of this AVP should be the same Diameter identity used as part of the CER/CEA base Diameter capabilities exchange.

This extension makes no change to the OC-Sequence-Number AVP in the OC-Supported-Features AVP.

4.1.1. OC-Peer-Node

The OC-Reporting-Node AVP (AVP code TBD) is of type DiameterIdentity and is inserted by the reporting node. It contains the Diameter Identity of the inserting node. This is used by the reacting node to determine if the peer report came from a true peer. Behavior associated with this AVP is discussed in Section 5.3

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules: MUST	MUST NOT
OC-Peer-Node	TBD1	x.x	Unsigned64		V

4.2. OC-OLR AVP

This extension makes no changes to the SequenceNumber or ValidityDuration AVPs in the OC-OLR AVP. These AVPs must also be used in peer overload reports.

The agent overload function extends the base Diameter overload specification by defining a new overload report type of "peer". See section [4.5] in [I-D.ietf-dime-ovli] for a description of the overload report type AVP.

The following extension is proposed for the ReportType AVP.

2 Peer. The overload treatment should apply to all requests bound for the peer identified in the overload report. If the peer identified in the overload report is not a peer to the reacting endpoint then the overload report should be stripped and not acted upon.

The overload report must also include the Diameter identity of the agent that generated the report. This is necessary to handle the case where there is a non supporting agent between the reporting node and the reacting node. Without the indication of the agent that generated the overload request, the reacting node could erroneously assume that the report applied to the non supporting node. This could, in turn, result in unnecessary traffic being either redistributed or throttled.

This extension adds the Reporting-Node AVP.

4.2.1. OC-Reporting-Node

The OC-Reporting-Node AVP (AVP code TBD) is of type DiameterIdentity and is inserted by the reporting node. It contains the Diameter Identity of the inserting node. This is used by the reacting node to determine if the peer report came from a true peer. Behavior associated with this AVP is discussed in Section 5.3

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules: MUST	AVP flag rules: MUST NOT
OC-Reporting-Node	TBD1	x.x	Unsigned64		V

5. Agent Overload Behavior

5.1. Capability Advertisement

Diameter nodes that support this extension must include the OLR_PEER_REPORT capability in all OC-Feature-Vector AVPs sent in Diameter reqeuest messages.

Diameter nodes that support this extension must also inlcude the OC-Peer-Node-ID AVP in the OC-Supported-Features AVP. The value of the OC-Peer-Node-ID AVP must be the same as the Diameter identity used in the base Diameter CER/CEA capabilities exchange for the connection upon which the message carrying the OC-Feature-Vector AVP is to be sent.

An agent that supports this extension must insert a separate OC-Supported-Features AVP in all request messages traversing the agent.

A Diameter node that receives an OC-Supported-Features AVP that indicates support for the peer report type must determine if the support came from a true peer. If the value of the OC-Peer-Node-ID AVP matches the Diameter identity of the previous hop Diameter node then the receiving node knows that the peer node supports this extension.

When an agent receives a request that contains an OC-Supported-Features AVP containing an indication for support of the peer overload report type, the agent must remove that instance of the OC-Supported-Features AVP from the request. The agent must insert it's own OC-Supported-Features AVP, with an OC-Peer-Node-ID AVP containing its own Diameter ID, in the request.

5.2. Agent Overload Reporting Node Behavior

An agent that supports this specification must have the ability to determine when it is appropriate to send an overload report. The method used to determine when to request overload abatement handling is an implementation decision but is likely to be based on usage characteristics like CPU utilization or the total number of Diameter transactions being handled over a unit of time.

Once the agent determines that there is need to request a reduction in traffic then it SHOULD include the overload report in answer messages handled by the agent for transactions where the agent believes the previous hop supports the peer overload report type.

The overload report must include a type of peer.

The amount of reduction requested MUST be included in the overload report.

The requested duration of the report MUST be included in the overload report.

The overload report must include a sequence number as specified in the based DOIC specification.

Editor's note: These statements might turn out to be repeats of normative requirements in the DOC baseline specification. If this is so then they likelly can be removed from this document.

The overload report must include the DiameterIdentity of the reporting node in the OC-Reporting-Node AVP. This is used by DOC end-points to determine if the report came from a true peer or from a non adjacent reporting node.

The reporting agent must follow all other overload reporting node behaviors outlined in the base overload specification. This includes sending a report with a reduction of zero when the need for a reduction has been abated. It also includes sending a new overload report, with a new sequence number, to refresh the abatement duration.

5.3. Agent Overload Reacting Node Behavior

A reacting node supporting this extension must support the receipt of three overload reports in a single message. The message might inlude both a host overload report, a realm overload report and a peer overload report.

A DOC reacting node receiving an overload report of type "peer" must first verify that the report came from an adjacent node. This can be achieved by comparing the OC-Reporting-Node AVP value with the Diameter identity of the node on the other end of the connection upon which the message is received.

If the report came from a non-adjacent reporting node then the reacting node must strip the overload report and take no other action as a result of the report.

If the peer report came from an adjacent node then the reacting node should attempt to adjust the distribution of subsequent traffic through available routes, with a reduction of the amount of traffic sent to the reporting node. The reasoning behind re-distributing the requests through other routes is the general thought that it is best to attempt to complete requests when there is capacity in the network. In the case of agent overload, the targetted servers will not necessarily be overloaded. As such, re-distributed requests are likely to be successfully handled.

If there is not sufficient capacity to route offered traffic through the available routes then the reacting node must throttle traffic.

If the reacting node is throttling traffic then it must select the throttled traffic using the loss algorithm defined in [I-D.ietf-dime-ovli].

If the Diameter node is a Diameter end-point then the throttling action results in the Diameter request not being sent and presenting the appropriate application level response to the request that caused the need for the Diameter transaction.

If the Diameter node is a Diameter agent then the throttling action involves generating the error response in an answer message for the throttled transactions. The error response must be the same as defined for agent throttling actions in [I-D.ietf-dime-ovli].

6. IANA Considerations

Editors note: This section will be completed once the base overload

document has finished the definition of extension IANA requirements.

7. Security Considerations

Agent overload is an extension to the based Diameter overload mechanism. As such, all of the security considerations outlined in [I-D.ietf-dime-ovli] apply to the agent overload scenarios.

It is possible that the malicious insertion of an agent overload report could have a bigger impact on a Diameter network as agents can be concentration points in a Diameter network. Where an end-point report would impact the traffic sent to a single Diameter server, for example, an agent overload report could throttle all traffic to the Diameter network.

This impact is amplified in an agent that sits at the edge of a Diameter network that serves as the entry point from all other Diameter networks.

8. Acknowledgements

Adam Roach and Eric McMurry for the work done in defining a comprehensive Diameter overload solution in draft-roach-dime-overload-ctrl-03.txt.

Ben Campbell for his insights and review of early versions of this document.

9. Normative References

[I-D.ietf-dime-ovli] Korhonen, J., "Diameter Overload Indication Conveyance", October 2013.

[RFC2119] Bradner, S., "Key words for use 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.

[RFC7068] McMurry, E. and B. Campbell, "Diameter Overload Control

Requirements", RFC 7068, November 2013.

Author's Address

Steve Donovan
Oracle
7460 Warren Parkway, Suite 300
Frisco, Texas 75034
United States

Email: srdonovan@usdonovans.com