Diameter Maintenance and Extensions (DIME)
Internet-Draft
Updates: RFC7683 (if approved)
Intended status: Standards Track
Expires: September 8, 2017

S. Donovan
Oracle

March 7, 2017

Diameter Agent Overload and the Peer Overload Report
draft-ietf-dime-agent-overload-10.txt

Abstract

This specification documents an extension to RFC 7683 (Diameter Overload Indication Conveyance (DOIC)) base solution. The extension defines the Peer overload report type. The initial use case for the Peer report is the handling of occurrences of overload of a Diameter agent.

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].

Table of Contents

1. Introduction

This specification documents an extension to the Diameter Overload Indication Conveyance (DOIC) [RFC7683] base solution. The extension defines the Peer overload report type. The initial use case for the Peer report is the handling of occurrences of overload of a Diameter agent.

This document defines the behavior of Diameter nodes when Diameter agents enter an overload condition and send an overload report requesting a reduction of traffic. It also defines new overload report type, the Peer overload report type, that is used for handling of agent overload conditions. The Peer overload report type is defined in a generic fashion so that it can also be used for other Diameter overload scenarios.

The base Diameter overload specification [RFC7683] 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 handle abatement of the overload occurrence as close to the source of the Diameter traffic as feasible. When possible this is done at the originator of the traffic, generally referred to as a Diameter Client. A Diameter Agent might 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 -- Diameter Clients and Diameter 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 defines a new overload report type to communicate occurrences of agent overload. This report type works for the "Loss" overload mitigation algorithm defined in [RFC7683] and is expected to work for other overload abatement algorithms defined in extensions to the DOIC solution.

2. Terminology and Abbreviations

Diameter Node

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

Diameter Endpoint

An RFC6733 Diameter Client and RFC6733 Diameter Server.

Diameter Agent

An RFC6733 Diameter Agent.

Reporting Node

A DOIC Node that sends an overload report in a Diameter answer message.

Reacting Node

A DOIC Node that receives and acts on a DOIC overload report.

DOIC Node

A Diameter Node that supports the DOIC solution defined in [RFC7683].

3. Peer Report Use Cases

This section outlines representative use cases for the peer report used to communicate agent overload.

There are two primary classes of use cases currently identified, those involving the overload of agents and those involving overload of Diameter endpoints. In both cases the goal is to use an overload algorithm that controls traffic sent towards peers.

3.1. Diameter Agent Overload Use Cases

The peer report needs to support the following use cases.

3.1.1. 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.

```
                              +-+    +-+    +-+
                              |c|----|a|----|s|
                              +-+    +-+    +-+
```

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.

```
                                            +-+
                                          --|s|
                              +-+    +-+ /  +-+
                              |c|----|a|-   ...
                              +-+    +-+ \  +-+
                                          --|s|
                                            +-+
```

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 overload report in answer messages flowing to the client. This overload report will contain a requested reduction in the amount of traffic sent to the agent. The client will apply overload abatement behavior as defined in the base Diameter overload specification [RFC7683] or the extension draft that defines the indicated overload abatement algorithm. This will result in the throttling of the abated traffic that would have been sent to the agent, as there is no alternative route. The client sends an appropriate error response to the originator of the request.

3.1.2. 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 Diameter 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, under normal circumstances, the client

will use the primary connection for all traffic. The secondary connection is used when there is a failure scenario of some sort.

```
                  +--+   +-+
                --|a1|---|s|
           +-+ /  +--+\ /+-+
           |c|-        x
           +-+ .  +--+/ \+-+
                ..|a2|---|s|
                  +--+   +-+
```

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 local distribution policy to determine the traffic sent through each client.

```
                  +--+   +-+
                --|a1|---|s|
           +-+ /  +--+\ /+-+
           |c|-        x
           +-+ \  +--+/ \+-+
                --|a2|---|s|
                  +--+   +-+
```

Figure 4

In the case where one of the agents in the above scenario becomes 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.

When the client has an active and a standby connection to the two agents then an alternative strategy for responding to an overload report from an agent is to change the standby connection to active and route all traffic through the new active connection.

In the case where both agents are reporting overload, the client may need to start decreasing the total traffic sent to the agents. This would be done in a similar fashion as discussed in Section 3.1.1 The amount of traffic depends on the combined reduction requested by the two agents.

3.1.3. Agent Chains

There are also deployment scenarios where there can be multiple Diameter Agents between Diameter Clients and Diameter Servers. An example of this type of deployment includes when there are Diameter agents between administrative domains.

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.

```
          +---+     +---+   +-+
        --|a11|-----|a21|---|s|
   +-+ /  +---+ \ / +---+\ /+-+
   |c|-          x        x
   +-+ \  +---+ / \ +---+/ \+-+
        --|a12|-----|a22|---|s|
          +---+     +---+   +-+
```

Figure 5

Handling of overload of one or both of agents a11 or a12 in this case is equivalent to that discussed in Section 3.1.2.

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.

The handling of peer overload reports is similar to that discussed in Section 3.1.2. If the overload can be addressed using diversion 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 number of transactions. When throttling requests, an agent uses the same error responses as defined in the base DOIC specification [RFC7683].

3.2. Diameter Endpoint Use Cases

This section outlines use cases for the peer overload report involving Diameter Clients and Diameter Servers.

3.2.1. Hop-by-hop Abatement Algorithms

It is envisioned that abatement algorithms will be defined that will support the option for Diameter Endpoints to send peer reports. For instance, it is envisioned that one usage scenario for the rate algorithm, [I-D.ietf-dime-doic-rate-control], which is being worked on by the DIME working group as this document is being written, will involve abatement being done on a hop-by-hop basis.

This rate deployment scenario would involve Diameter Endpoints generating peer reports and selecting the rate algorithm for abatement of overload conditions.

4. Interaction Between Host/Realm and Peer Overload Reports

It is possible that both an agent and an end-point in the path of a transaction are overloaded at the same time. When this occurs, Diameter entities need to handle both overload reports. In this scenario the reacting node should first handle the throttling of the overloaded host or realm. Any messages that survive throttling due to host or realm reports should then go through abatement for the peer overload report. In this scenario, when doing abatement on the PEER report, the reacting node SHOULD take into consideration the number of messages already throttled by the handling of the HOST/ REALM report abatement.

> Note: The goal is to avoid traffic oscillations that might result from throttling of messages for both the HOST/REALM overload reports and the PEER overload reports. This is especially a concern if both reports indicate the LOSS abatement algorithm.

5. Peer Report Behavior

This section defines the normative behavior associated with the Peer Report extension to the DOIC solution.

5.1. Capability Announcement

5.1.1. Reacting Node Behavior

When sending a Diameter request a DOIC Node that supports the OC_PEER_REPORT (as defined in Section 6.1.1) feature MUST include in the OC-Supported-Features AVP an OC-Feature-Vector AVP with the OC_PEER_REPORT bit set.

When sending a request a DOIC Node that supports the OC_PEER_REPORT feature MUST include a SourceID AVP in the OC-Supported-Features AVP with its own DiameterIdentity.

When a Diameter Agent relays a request that includes a SourceID AVP in the OC-Supported-Features AVP, if the Diameter Agent supports the OC_PEER_REPORT feature then it MUST remove the received SourceID AVP and replace it with a SourceID AVP containing its own DiameterIdentity.

5.1.2. Reporting Node Behavior

When receiving a request a DOIC Node that supports the OC_PEER_REPORT feature MUST update transaction state with an indication of whether or not the peer from which the request was received supports the OC_PEER_REPORT feature.

> Note: The transaction state is used when the DOIC Node is acting as a peer-report reporting node and needs send OC-OLR reports of type peer in answer messages. The peer overload reports are only included in answer messages being sent to peers that support the OC_PEER_REPORT feature.

The peer supports the OC_PEER_REPORT feature if the received request contains an OC-Supported-Features AVP with the OC-Feature-Vector with the OC_PEER_REPORT feature bit set and with a SourceID AVP with a value that matches the DiameterIdentity of the peer from which the request was received.

When an agent relays an answer message, a reporting node that supports the OC_PEER_REPORT feature MUST strip any SourceID AVP from the OC-Supported-Features AVP.

When sending an answer message, a reporting node that supports the OC_PEER_REPORT feature MUST determine if the peer to which the answer is to be sent supports the OC_PEER_REPORT feature.

If the peer supports the OC_PEER_REPORT feature then the reporting node MUST indicate support for the feature in the OC-Supported-Features AVP.

If the peer supports the OC_PEER_REPORT feature then the reporting node MUST insert the SourceID AVP in the OC-Supported-Features AVP in the answer message.

If the peer supports the OC_PEER_REPORT feature then the reporting node MUST insert the OC-Peer-Algo AVP in the OC-Supported-Features AVP. The OC-Peer-Algo AVP MUST indicate the overload abatement algorithm that the reporting node wants the reacting nodes to use should the reporting node send a peer overload report as a result of becoming overloaded.

5.2. Peer Overload Report Handling

This section defines the behavior for the handling of overload reports of type peer.

5.2.1. Overload Control State

This section describes the Overload Control State (OCS) that might be maintained by both the peer-report reporting node and the peer-report reacting node.

This is an extension of the OCS handling defined in [RFC7683].

5.2.1.1. Reporting Node Peer Report OCS

A DOIC Node that supports the OC_PEER_REPORT feature SHOULD maintain Reporting Node OCS, as defined in [RFC7683] and extended here.

If different abatement specific contents are sent to each peer then the reporting node MUST maintain a separate reporting node peer report OCS entry per peer to which a peer overload report is sent.

> Note: The rate overload abatement algorithm allows for different rates to be sent to each peer.

5.2.1.2. Reacting Node Peer Report OCS

In addition to OCS maintained as defined in [RFC7683], a reacting node that supports the OC_PEER_REPORT feature maintains the following OCS per supported Diameter application:

> A peer-type OCS entry for each peer to which it sends requests.

A peer-type OCS entry is identified by the pair of Application-ID and the peer's DiameterIdentity.

The peer-type OCS entry include the following information (the actual information stored is an implementation decision):

> Sequence number (as received in the OC-OLR AVP).
>
> Time of expiry (derived from OC-Validity-Duration AVP received in the OC-OLR AVP and time of reception of the message carrying OC-OLR AVP).
>
> Selected abatement algorithm (as received in the OC-Supported-Features AVP).

Input data that is abatement algorithm specific (as received in the OC-OLR AVP -- for example, OC-Reduction-Percentage for the loss abatement algorithm).

5.2.2. Reporting Node Maintenance of Peer Report OCS

All rules for managing the reporting node OCS entries defined in [RFC7683] apply to the peer report.

5.2.3. Reacting Node Maintenance of Peer Report OCS

When a reacting node receives an OC-OLR AVP with a report type of peer it MUST determine if the report was generated by the Diameter peer from which the report was received.

If a reacting node receives an OC-OLR AVP of type peer and the SourceID matches the DiameterIdentity of the Diameter peer from which the response message was received then the report was generated by a Diameter peer.

If a reacting node receives an OC-OLR AVP of type peer and the SourceID does not match the DiameterIdentity of the Diameter peer from which the response message was received then the reacting node MUST ignore the overload report.

> Note: Under normal circumstances, a Diameter node will not add a peer report when sending to a peer that does not support this extension. This requirement is to handle the case where peer reports are erroneously or maliciously inserted into response messages.

If the peer report was received from a Diameter peer then the reacting node MUST determine if it is for an existing or new overload condition.

The peer report is for an existing overload condition if the reacting node has an OCS that matches the received peer report. For a peer report, this means it matches the Application-ID and the peer's DiameterIdentity in an existing OCS entry.

If the peer report is for an existing overload condition then it MUST determine if the peer report is a retransmission or an update to the existing OLR.

If the sequence number for the received peer report is greater than the sequence number stored in the matching OCS entry then the reacting node MUST update the matching OCS entry.

If the sequence number for the received peer report is less than or equal to the sequence number in the matching OCS entry then the reacting node MUST silently ignore the received peer report. The matching OCS MUST NOT be updated in this case.

If the received peer report is for a new overload condition then the reacting node MUST generate a new OCS entry for the overload condition.

For a peer report this means it creates an OCS entry with a DiameterIdentity from the SourceID AVP in the received OC-OLR AVP.

If the received peer report contains a validity duration of zero ("0") then the reacting node MUST update the OCS entry as being expired.

The reacting node does not delete an OCS when receiving an answer message that does not contain an OC-OLR AVP (i.e. absence of OLR means "no change").

The reacting node sets the abatement algorithm based on the OC-Peer-Algo AVP in the received OC-Supported-Features AVP.

5.2.4. Peer-Report Reporting Node Behavior

When there is an existing reporting node peer report OCS entry, the reporting node MUST include an OC-OLR AVP with a report type of peer using the contents of the reporting node peer report OCS entry in all answer messages sent by the reporting node to peers that support the OC_PEER_REPORT feature.

> The reporting node determines if a peer supports the OC_PEER_REPORT feature based on the indication recorded in the reporting node's transaction state.

The reporting node MUST include its DiameterIdentity in the SourceID AVP in the OC-OLR AVP. This is used by DOIC Nodes that support the OC_PEER_REPORT feature to determine if the report was received from a Diameter peer.

The reporting agent must follow all other overload reporting node behaviors outlined in the DOIC specification.

5.2.5. Peer-Report Reacting Node Behavior

A reacting node supporting this extension MUST support the receipt of multiple overload reports in a single message. The message might

include a host overload report, a realm overload report and/or a peer overload report.

When a reacting node sends a request it MUST determine if that request matches an active OCS.

In all cases, if the reacting node is an agent then it MUST strip the Peer Report OC-OLR AVP from the message.

If the request matches an active OCS then the reacting node MUST apply abatement treatment to the request. The abatement treatment applied depends on the abatement algorithm indicated in the OCS.

For peer overload reports, the preferred abatement treatment is diversion. As such, the reacting node SHOULD attempt to divert requests identified as needing abatement to other peers.

If there is not sufficient capacity to divert abated traffic then the reacting node MUST throttle the necessary requests to fit within the available capacity of the peers able to handle the requests.

If the abatement treatment results in throttling of the request and if the reacting node is an agent then the agent MUST send an appropriate error response as defined in [RFC7683].

In the case that the OCS entry validity duration expires or has a validity duration of zero ("0"), meaning that if the reporting node has explicitly signaled the end of the overload condition then abatement associated with the OCS entry MUST be ended in a controlled fashion.

6. Peer Report AVPs

6.1. 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 by the agent's peer.

A supporting node must also include the SourceID AVP in the OC-Supported-Features capability AVP.

This AVP contains the DiameterIdentity of the node that supports the OC_PEER_REPORT feature. This AVP is used to determine if support for the peer overload report is in an adjacent node. The value of this AVP should be the same Diameter identity used as part of the Diameter Capabilities Exchange procedure defined in [RFC7683].

This extension also adds the OC-Peer-Algo AVP to the OC-Supported-Features AVP. This AVP is used by a reporting node to indicate the abatement algorithm it will use for peer overload reports.

```
 OC-Supported-Features ::= < AVP Header: 621 >
                           [ OC-Feature-Vector ]
                           [ SourceID ]
                           [ OC-Peer-Algo]
                         * [ AVP ]
```

6.1.1. OC-Feature-Vector AVP

The peer report feature defines a new feature bit for the OC-Feature-Vector AVP.

OC_PEER_REPORT (0x0000000000000010)

When this flag is set by a DOIC Node it indicates that the DOIC Node supports the peer overload report type.

6.1.2. OC-Peer-Algo AVP

The OC-Peer-Algo AVP (AVP code TBD1) is of type Unsigned64 and contains a 64 bit flags field of announced capabilities of a DOIC Node. The value of zero (0) is reserved.

Feature bits defined for the OC-Feature-Vector AVP and associated with overload abatement algorithms are reused for this AVP.

6.2. OC-OLR AVP

This extension makes no changes to the OC_Sequence_Number or OC_Validity_Duration AVPs in the OC-OLR AVP. These AVPs are also be used in peer overload reports.

The OC_PEER_REPORT feature extends the base Diameter overload specification by defining a new overload report type of "peer". See section [7.6] in [RFC7683] for a description of the OC-Report-Type AVP.

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 report, 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 diverted or throttled.

The SourceID AVP is used in the OC-OLR AVP to carry this DiameterIdentity.

```
OC-OLR ::= < AVP Header: 623 >
           < OC-Sequence-Number >
           < OC-Report-Type >
           [ OC-Reduction-Percentage ]
           [ OC-Validity-Duration ]
           [ SourceID ]
         * [ AVP ]
```

6.2.1. OC-Report-Type AVP

The following new report type is defined for the OC-Report-Type AVP.

PEER_REPORT 2   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.

6.3. SourceID AVP

The SourceID AVP (AVP code TBD2) is of type DiameterIdentity and is inserted by a Diameter node to indicate the source of the AVP in which it is a part.

In the case of peer reports, the SourceID AVP indicates the node that supports this feature (in the OC-Supported-Features AVP) or the node that generates an overload with a report type of peer (in the OC-OLR AVP).

It contains the DiameterIdentity of the inserting node. This is used by other Diameter nodes to determine the node that inserted the enclosing AVP that contains the SourceID AVP.

6.4. Attribute Value Pair Flag Rules

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules: MUST	AVP flag rules: MUST NOT
OC-Peer-Algo	TBD1	6.1.2	Unsigned64		V
SourceID	TBD2	6.3	DiameterIdentity		V

7. IANA Considerations

7.1. AVP Codes

New AVPs defined by this specification are listed in Section 6.4. All AVP codes are allocated from the 'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.

One new OC-Report-Type AVP value is defined in Section 6.2.1

7.2. New Registries

There are no new IANA registries introduced by this document.

The values used for the OC-Peer-Algo AVP are the subset of the "OC-Feature-Vector AVP Values (code 622)" registry. Only the values in that registry that apply to overload abatement algorithms apply to the OC-Peer-Algo AVP.

8. Security Considerations

Agent overload is an extension to the base Diameter overload mechanism. As such, all of the security considerations outlined in [RFC7683] 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, a peer 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.

9. 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.

10. References

10.1. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC7068] McMurry, E. and B. Campbell, "Diameter Overload Control Requirements", RFC 7068, DOI 10.17487/RFC7068, November 2013, <http://www.rfc-editor.org/info/rfc7068>.

10.2. Normative References

[I-D.ietf-dime-doic-rate-control] Donovan, S. and E. Noel, "Diameter Overload Rate Control", draft-ietf-dime-doic-rate-control-03 (work in progress), March 2016.

[RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <http://www.rfc-editor.org/info/rfc6733>.

[RFC7683] Korhonen, J., Ed., Donovan, S., Ed., Campbell, B., and L. Morand, "Diameter Overload Indication Conveyance", RFC 7683, DOI 10.17487/RFC7683, October 2015, <http://www.rfc-editor.org/info/rfc7683>.

Author's Address

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

Email: srdonovan@usdonovans.com