Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 22, 2013

B. Campbell
Tekelec
H. Tschofenig
Nokia Siemens Networks
J. Korhonen
Renesas Mobile
A. Roach
Mozilla

February 18, 2013

Diameter Overload Data Analysis
draft-campbell-dime-overload-data-analysis-00

Abstract

When a Diameter server or agent becomes overloaded, it needs to be able to gracefully reduce its load, typically by informing clients to reduce sending traffic for some period of time. Multiple mechanisms have been proposed for transporting overload and load information. While these proposals differ in many ways, they share similar data requirements. This document analyzes the data requirements of each proposal with a view towards proposing a common set of of Diameter Attribute-Value Pairs (AVPs).

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 22, 2013.

Table of Contents

1. Introduction

When a Diameter [RFC6733] server or agent becomes overloaded, it needs to be able to gracefully reduce its load, typically by informing clients to reduce sending traffic for some period of time. The Diameter Overload Control Requirements [I-D.ietf-dime-overload-reqs] describe requirements for overflow control mechanisms.

At the time of this writing, there have been two proposals for Diameter overload control mechanisms. "A Mechanism for Diameter Overload Control" (MDOC) [I-D.roach-dime-overload-ctrl] defines a mechanism that piggybacks overload and load state information over existing Diameter messages. "The Diameter Overload Control Application" (DOCA) [I-D.korhonen-dime-ovl] defines a mechanism that uses a new and distinct Diameter application to communicate similar information. While there are significant differences between the two proposals, they carry similar information. Each proposal includes its own set of Diameter AVPs.

This document is intended as a framework for discussing the data requirements of the two proposals. It includes an analysis of the differences and similarities of their respective data elements, with a view towards rationalizing the AVPs from the two proposals.

> The authors expect that a follow-on effort will eventually specify a common data model for reporting Diameter overload information.

This document assumes that Diameter nodes exchange overload control information via Diameter, rather than via some out-of-band channel. This document does not address the specific difference of either mechanism proposal, except where they impact the AVP definitions.

2. Documentation Conventions

This document uses terms defined in [RFC6733] and [I-D.ietf-dime-overload-reqs].

3. Overload Control Data Usage

A Diameter overload control mechanism based on the overload control requirements [I-D.ietf-dime-overload-reqs] involves the exchange of information between two or more Diameter nodes. The exchanged information serves three distinct purposes:

Negotiation: Diameter nodes need to negotiate support for the overload control mechanism in general. Nodes that support overload control need to advertise the overload control scopes they can support. Finally, they need to select an overload control algorithm.

Communication of Overload State: Nodes need to report that an overload condition is in effect, to what degree they are overloaded, and the scope of the overload condition. We refer to such a communication as an "Overload Report".

Communication of Load: Nodes need to communicate their current load status, even when not in an overloaded state.

Overload Control information may be communicated between adjacent Diameter nodes, or it may cross one or more intervening nodes. Overload Control information can be communicated in either direction; that is, a downstream node can indicate overload to an upstream node, or vice-versa.

Open Issue: There is an ongoing discussion about whether the overload control mechanism should be strictly hop-by-hop, or whether it should support communication between non-adjacent nodes. The results of this discussion may have implications for overload control data elements.

4. Mechanism Differences that Affect Data Structures

While a thorough comparison of the two proposed mechanisms is out of scope for this document, there are a few differences that directly impact the choice of data elements.

4.1. Non-Adjacent Nodes

MDOC only supports hop-by-hop communication of overload information. DOCA allows for the possibility of communication between non-adjacent nodes. For hop-by-hop communication, the originator of an overload report is always the directly connected node. If non-adjacent communication is to be allowed, the data model needs a way to express the identity of the originating node.

4.2. Stateless Negotiation

Both MDOC and DOCA allow overload control parameters to be negotiated at the beginning of a connection, and persist for the duration of the connection. DOCA also allows a "stateless" mode, where the parameters do not persist between overload reports. This requires

the sender of an overload report to restate any relevant parameters for each report. Thus, the DOCA overload report format includes the ability to express all such parameters at any time, not just during negotiation.

Note that stateless negotiation does not mean that no state may ever be saved. Nodes may use implementation-specific methods of remembering certain parameters, or out-of-band configuration methods to do the same.

4.3. Overload Scopes

As described in [I-D.ietf-dime-overload-reqs], it's possible for a Diameter node to experience overload that impacts some subset of potential traffic. For example, a Diameter agent might route traffic to different servers based on realm. If the server for one realm experienced an outage or overload condition, the agent report that it is overloaded for that realm, but can process traffic for other realms normally. We use the term "overload scope", or simply "scope", to refer to the set of potential messages affected by an overload report.

MDOC includes a richer (and therefore more complex) concept of overload scopes. A node may include multiple scopes in an overload report. Each scope entry indicates both the type of scope, and the value of the scope, where the value is interpreted according to the type.

DOCA also allows a node to include multiple scopes in a report. But DOCA's current set of scope types only affect the interpretation of the originating node identity. Therefore the DOCA scope entries do not include a value.

4.4. Hard or Soft Overload State

MDOC assumes that overload information is soft state. That is, it expires if not refreshed within a stated interval. DOCA also treats most overload information as soft state, but there are situations where it may be treated as hard-state. For example, if the OC-Level is set to "Hold", the expiration time is not honored.

5. Naming Conventions

MDOC and DOCA use somewhat different naming conventions for their respective AVPs. DOCA prefixes each AVP name with "OC". (for example, "OC-Scope"). MDOC prefixes AVPs that can appear in the root of messages with "Overload", and leaves those that occur inside an

overload related grouped AVP to be identified by context. (For example, "Overload Info" and "Supported Scopes"). The working group should consider picking one approach or the other.

6. Data Element Comparison

6.1. Data Elements for Connection Establishment and Negotiation

The following sections describe data elements used for initial negotiation.

6.1.1. Supported Scope Selection

- DOCA: OC-Scope : Bitmap of scopes supported by the sender. Currently defined values are "Host scope", "Realm Scope", "Only origin realm", "Application Information", "Node Utilization Information", and "Application Priorities".

- MDOC: Supported-Scopes : Bitmap of scopes supported by the sender. Currently defined values are "Destination-Realm", "Application-ID", "Destination-Host", "Host", "Connection", "Session-Group", and "Session".

DOCA uses OC-Scope both to declare supported scopes, and to list the scopes associated with a particular overload report. MDOC uses separate dedicated AVPs for the two purposes. DOCA overloads OC-Scope to include indicators that load information and priority information may be included.

6.1.2. Algorithm Selection

- DOCA: OC-Algorithm : Bitmap of supported algorithms. Currently defined values are "Drop", "Throttle", and "Prioritize". Multiple values allowed.

- MDOC: Overload-Algorithm: Enumeration of supported algorithms. Multiple instances allowed in negotiation. Currently, there is one algorithm described, namely "loss".

Both mechanisms support algorithm extensibility. MDOC only allows Overload-Algorithm to occur in a CER or CEA message, and negotiates a single algorithm for the duration of the connection. DOCA allows the algorithm to be selected at report time. (Open Issue: what does it mean to indicate multiple algorithms in a congestion report?)

6.1.3. Application Selection

- DOCA: OC-Applications: Indications of the applications that are of interest.

- MDOC: MDOC assumes that overload reports can apply to any and all applications, and does not negotiate the list upfront. The "application" scope is used to select one or more applications on a per-report basis.

Open Issue: Are there use cases for the up front negotiation of applications of interest?

6.1.4. Frequency of Reports

- DOCA: OC-Tocl: Indicates how frequent reports shall be sent.

- MDOC: N/A

Since MDOC piggybacks overload reports in existing messages, the rate of overload reports is the same as the overall message rate. This may have advantage of giving more rapid and precise feedback as load increases.

Open Issue: We need further discussion about the appropriate rate(s) for overload reporting, regardless of which mechanism may be selected.

6.1.5. Grouping

- DOCA: n/a - negotiation AVPs included at message root.

- MDOC: Load-Info: Grouped AVP acting as a container for the other AVPs used for negotiation.

6.2. Data Elements for Overload and Load reporting

6.2.1. Scope of Report

- DOCA: OC-Scope (See Section 6.1.1)

- MDOC: Load-Info-Scope: Octet-String giving the scope of the overload report. The string contains a type indicator and a value. One or more instances required.

MDOC has a richer and more complex concept of scopes. Multiple scopes can be combined for a given overload report. Allowable scope combinations are described in [I-D.roach-dime-overload-ctrl].

6.2.2. Overload Severity

- DOCA: OC-Level: OctetString(1): Values 1-6 define discreet overload levels of increasing severity, with 1 meaning no overload condition, and 6 meaning clients should switch to a different server.

- DOCA: OC-Sending-Rate: Float32: Used when the "throttle" algorithm is in effect to indicate the maximum desired Diameter message rate.

- MDOC: Overload-Metric (Unsigned32): A numeric representation of load. The meaning is up to the interpretation of the selected algorithm, with the exception that a value of zero always means that no overload abatement is in effect. For the "Loss" algorithm, Overload Metric is a numeric value in the range of zero through 100, indicating the percentage of traffic reduction requested.

The Overload-Metric AVP used by MDOC is more general than OC-Level, in that it's interpretation is left to the algorithm. The meaning of the OC-Level values appear to be fixed regardless of algorithm choice. the OC-Level meanings could be used in MDOC by defining a new algorithm that interpreted Overload-Metric values 1-6 in the same way as defined for OC-Level.

Since MDOC does not define an algorithm similar to "throttle", it has no built in analog to OC-Sending-Rate. However, since MDOC allows algorithm extensibility, one could define a similar algorithm, and if necessary, add an extension AVP to state sending-rate.

6.2.3. Report Algorithm

- DOCA: OC-Algorithm (See Section 6.1.2)

- MDOC: The overload control algorithm is set during negotiation, and doesn't change for the duration of the connection.

Open Issue: DOCA's reuse of the OC-Algorithm AVP seems to allow more than one algorithm to be assigned to a single overload report. It's not clear what that would mean.

6.2.4. Report Expiration

- DOCA: OC-Best-Before: (Time) Time of report expiration.

- MDOC: Period-Of-Validity (Unsigned32)- Number of seconds until expiration.

DOCA defines expiration to be a point in time. MDOC uses a duration, i.e. number of seconds until expiration. The DOCA approach seems to require clock synchronization.

DOCA contains an open issue about whether to allow reports to expire vs. requiring explicit signaling.

6.2.5. Current Load

- DOCA: OC-Utilization: Indicates the overall load situation as a value between 0 and 100.

- MDOC: Load: The load situation in terms of 0 - 65535.

Current load indicates the existing load on an otherwise non-overloaded node. MDOC's range of 0-65535 was selected to harmonize with the DNS service location (SRV) [RFC2782] record's "Weight" field.

6.2.6. Applications covered by a Report

- DOCA: OC-Applications: Indications what applications are of interest for load reporting.

- MDOC does not use a separate AVP for this purpose. Rather, one or more applications can be indicated using the application scope type.

6.2.7. Report Action

- DOCA: OC-Action: Indicates the start, interim, and end of an overload period.

- MDOC: MDOC does not have a separate AVP to indicate the start and stop of an overload condition. Rather, a report with a non-zero Overload-Metric value starts the condition, and a report with a zero value, or the expiration of the Period-of-Validity value, indicate an end. Subsequent reports with non-zero Overload-Metric values serve the same purpose as a DOCA report with an OC-Action value of "interim".

Open Issue: Is OC-Action redundant? DOCA also has the ability to express a non-overload condition in OC-Level, so an approach similar to that of MDOC should be workable.

6.2.8. Priority

- DOCA: OC-Priority: Unsigned32: When used in an OC-Information AVP, sets the relative priority of applications listed in OC-Applications. As specified, may also be used to set the priority of a given Diameter message. [Open Issue: Is OC-Priority only in effect when the "Prioritize" algorithm is in effect?]

- MDOC: N/A

MDOC does not have an explicit priority data element. Relative priority between applications can be managed using the "Application" scope. This is not exactly the same as stating inter-application priority explicitly, but it may be possible to accomplish similar behavior.

6.2.9. Session Groups

- DOCA: N/A

- MDOC: Session-Group: UTF8String: Session-Group allows a node to assign a session to a named group. Overload Reports can refer to all sessions in a group using the Session-Group AVP.

A common application for Session-Group is when a Diameter agent load balances Diameter sessions across a set of servers. If the agent assigns all of the sessions assigned to a particular server to a group, and that server later becomes overloaded, the agent can send one overload report that applies to all sessions in the group, but does not apply to sessions assigned to other, non-overloaded, servers.

DOCA may be able to do something similar using by using the OC-Origin AVP to identify the overloaded server. However, the server-group approach can work even if the Diameter agent performs topology hiding.

6.3. Result Codes

DOCA defines the following Diameter result codes:

- DIAMETER_NO_COMMON_SCOPE (Permanent Failure): The Diameter peers are unable to negotiate one or more scopes in common.

- DIAMETER_NO_COMMON_ALGORITHM (Permanent Failure): The Diameter peers are unable to negotiate one or more algorithms in common.

- DIAMETER_TOCL_TOO_SMALL (Permanent Failure): The peer included an OC-TOCL AVP with an unacceptably low value.

- DIAMETER_TOCL_TOO_BIG (Permanent Failure): The peer included an OC-TOCL AVP with an unacceptably high value.

- DIAMETER_RATE_TOO_BIG (Permanent Failure): The peer included an OC-SENDING-RATE AVP with an unacceptably high value.

A failure to negotiate Overload Control support does not cause a connection failure in MDOC. Instead, overload control is just not invoked on the connection.

MDOC defines the following result codes:

- DIAMETER_PEER_IN_OVERLOAD (Transient Failure): When a Diameter node drops a request due to overload, it responds with this result code. This is primarily used when the peer does not support overload control, and therefore fails to reduce load as it would be expected to do so if it supported overload control.

DIAMETER_PEER_IN_OVERLOAD may be of value to both mechanisms. The Overload Control Requirements [I-D.ietf-dime-overload-reqs] argues that the result codes in the Diameter base protocol are insufficient for reporting failures due to congestion.

7. IANA Considerations

This draft makes no requests of IANA. The authors expect that a follow-on effort will specify a common set of Overload Control AVPs.This may introduce additional IANA considerations.

8. Security Considerations

This document compares the data elements used by "DOCA [I-D.korhonen-dime-ovl] and MDOC [I-D.roach-dime-overload-ctrl]. It introduces no security considerations beyond those in the respective documents.

The authors expect that a follow-on effort will specify a common set of Overload Control AVPs. This may introduce additional security considerations.

The authors made no attempt to analyze the security considerations in the DOCA and MDOC specifications for completeness.

9. References

9.1. Normative References

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

[I-D.ietf-dime-overload-reqs] McMurry, E. and B. Campbell, "Diameter Overload Control Requirements", draft-ietf-dime-overload-reqs-03 (work in progress), January 2013.

[I-D.roach-dime-overload-ctrl] Roach, A., "A Mechanism for Diameter Overload Control", draft-roach-dime-overload-ctrl-01 (work in progress), October 2012.

[I-D.korhonen-dime-ovl] Korhonen, J., "Diameter Overload Control Application", draft-korhonen-dime-ovl-00 (work in progress), October 2012.

9.2. Informative References

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

Appendix A. Contributors

Eric McMurry made significant contributions to the analysis in this draft.

Authors' Addresses

Ben Campbell
Tekelec
17210 Campbell Rd.
Suite 250
Dallas, TX 75252
US

Email: ben@nostrum.com

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Email: Hannes.Tschofenig@nsn.com

Jouni Korhonen
Renesas Mobile
Porkkalankatu 24
Helsinki FIN-00180
Finland

Email: jouni.nospam@gmail.com

Adam Roach
Mozilla
Dallas, TX

Email: adam@nostrum.com