Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 28, 2015

B. Campbell
Oracle

October 25, 2014

Architectural Considerations for Diameter Load Information
draft-campbell-dime-load-considerations-00

Abstract

RFC 7068 describes requirements for Overload Control in Diameter. This includes a requirement to allow Diameter nodes to send "load" information, even when the node is not overloaded. The Diameter Overload Information Conveyance (DOIC) solution describes a mechanism meeting most of the requirements, but does not currently include the ability to send load. This document explores some architectural considerations for a mechanism to send load information.

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 April 28, 2015.

Table of Contents

1. Introduction 2
2. Differences between Load and Overload information 3
3. How is Load Information Used? 4
4. Piggy-Backing vs a Dedicated Application. 4
5. Which Nodes Exchange Load Information? 5
6. Scope of Load Information 6
7. Load Information Semantics 7
8. Is Negotiation of Support Needed? 7
9. Security Considerations 8
10. IANA Considerations 8
11. References 8
11.1. Normative References 8
11.2. Informative References 8
Author's Address 9

1. Introduction

[RFC7068] describes requirements for Overload Control in Diameter [RFC6733]. At the time of this writing, the DIME working group is working on the Diameter Overload Information Conveyance (DOIC) mechanism. As currently specified, DOIC fulfills some, but not all, of the requirements.

In particular, DOIC does not fulfill Req 24, which requires a mechanism where Diameter nodes can indicate their current load, even if they are not currently overloaded. DOIC also does not fulfill Req 23, which requires that nodes that diverts traffic away from overloaded nodes be provided with sufficient information to select targets that are most likely to have sufficient capacity.

There are several other requirements in RFC 7068 that mention both overload and load information that are only partially fulfilled by DOIC.

The DIME working group explicitly chose not to fulfill these requirements in DOIC due to several reasons. A principal reason was that the working group did not agree on a general approach for conveying load information. It chose to progress the rest of DOIC, and defer load information conveyance to a DOIC extension or a separate mechanism.

This document describes some high level architectural decisions that the working group will need to consider in order to solve the load-related requirements from RFC 7068.

At the time of this writing, there have been several attempts to create mechanisms for conveyance of both load and overload control information that were not adopted by the DIME working group. While these drafts are not expected to progress, they may be instructive when considering these decisions.

- [I-D.tschofenig-dime-dlba] proposed a dedicated Diameter application for exchanging load balancing information.

- [I-D.roach-dime-overload-ctrl] described a strictly peer-to-peer exchange of both load and overload information in new AVPs piggy-backed on existing Diameter messages.

- [I-D.korhonen-dime-ovl] described a dedicated Diameter application for exchanging both load and overload information.

2. Differences between Load and Overload information

Previous discussions of how to solve the load-related requirements in [RFC7068] have shown that people do not have an agreed-upon concept of how "load" information differs from "overload" information. The two concepts are highly interrelated, and so far the working group has not defined a bright line between what constitutes load information and what constitutes overload information.

In the author's opinion, there are two primary differences. First, a Diameter node always has a load. At any given time that load maybe effectively zero, effectively fully loaded, or somewhere in between. In contrast, overload is an exceptional condition. A node only has overload information when it in an overloaded state. Furthermore, the relationship between a node's load level and overload state at any given time may be vague. For example, a node may normally operate at a "fully loaded" level, but still not be considered overloaded. Another node may declare itself to be "overloaded" even though it might not be fully "loaded".

Second, Overload information, in the form of a DOIC Overload Report (OLR) [I-D.ietf-dime-ovli] indicates an explicit request for action on the part of the reacting node. That is, the OLR requests that the reacting node reduce the offered load by an indicated amount or to an indicated level. Effectively, DOIC provides a contract between the reporting node and the reacting node.

In contrast, load is informational. That is, load information can be considered a hint to the recipient node. That node may use the load information for load balancing purposes, as an input to certain overload abatement techniques, to make inferences about the likelihood that the sending node becomes overloaded in the immediate future, or for other purposes.

None of this prevents a Diameter node from deciding to reduce the offered load based on load information. The fundamental difference is that an overload report requires that reduction.

3. How is Load Information Used?

[RFC7068] contemplates two primary uses for load information. Req 23 discusses how load information might be used when performing diversion as an overload abatement technique, as described in [I-D.ietf-dime-ovli]. When a reacting node diverts traffic away from an overloaded node, it needs load information for the other candidates for that traffic in order to effectively load balance the diverted load between potential candidates. Otherwise, diversion has a greater potential to drive other nodes into overload.

Req 24 discusses how a Diameter information might be used when no overload condition currently exists. Diameter nodes can use the load information to make decisions to try to avoid overload conditions in the first place. Normal load-balancing falls into this category. A node might also take other proactive steps to reduce offered load based on load information, so that the loaded node never goes into overload in the first place.

If the loaded nodes are Diameter servers (or clients in the case of server-to-client transactions), both of these uses are most effectively accomplished by a Diameter node that performs server selection. Typically, server selection is performed by a node (a client or an agent) that is an immediate peer of the server. However a client or proxy that is not an immediate peer to the selected server can enforce server selection by inserting a Destination-Host AVP.

4. Piggy-Backing vs a Dedicated Application.

[I-D.roach-dime-overload-ctrl] imbeds load and overload information onto messages of existing applications. This is known as a "piggy-back" approach. Such an approach has the advantage of not requiring new messages to carry load information. It has an additional advantage of scaling with load; that is, the more the transaction load, the more opportunities to send load information.

DOIC [I-D.ietf-dime-ovli] also uses a piggy-backed approach to send OLRs. Given the potentially tight connection between load and overload information, there may be advantages to maintaining consistency with DOIC.

[I-D.tschofenig-dime-dlba] used a dedicated application to carry load information. This application has quasi-subscription semantics, where a client requests updates according to a cadence. The server can send unsolicited updates if the load level changes between updates in the cadence.

[I-D.korhonen-dime-ovl] also used a dedicated application, but allowed nodes to send unsolicited reports containing load and overload information. The mechanism has an issue that the sender of load information may not know which other nodes need it. It may be possible to infer that information from the primary Diameter applications.

Another potential approach is that of a dedicated Diameter application with a slightly different subscription semantic than that of [I-D.tschofenig-dime-dlba]. In such an application, a node that consumes load information sends a Diameter request to the source of the load information. This request indicates that the consumer wishes to receive load information for some period of time. The load source would send periodic Diameter requests indicating the current load level, until such time that the subscription period expired, or the subscribe explicitly unsubscribed. After the initial notification, the sender would only send updates when the load level changed.

5. Which Nodes Exchange Load Information?

Previous load related efforts have made different assumptions about which Diameter nodes exchange load information.

[I-D.roach-dime-overload-ctrl] operated in a strictly peer-to-peer mode. Each node would only learn the load (and overload) information from its immediate peers.

[I-D.korhonen-dime-ovl] and [I-D.tschofenig-dime-dlba] are each effectively any-to-any. That is, they each allowed any node to send load information to any other node that supported the dedicated overload or load application, respectively.

In the latter case, load is effectively sent between clients and servers of the dedicated application, but those roles may not match the client and server roles for the "main" Diameter applications in use. For example, a pair of adjacent diameter agents might be

"client" and "server" for the dedicated "load" application, effectively creating a peer-to-peer relationship similar to that of [I-D.roach-dime-overload-ctrl].

Each approach has advantages. Since server selection is typically done by immediate peers to the servers, peer-to-peer transmission covers most cases. Additionally, selection of non-terminal nodes is exclusively done on a peer-to-peer basis. If the loaded node is an agent, for example, the load information is only useful to immediate peers. Peer-to-peer transmission is the easiest to negotiation. (See Section 8)

Any-to-Any transmission offers more flexibility, and could potentially cover the case where server selection is done by nodes that are not peers to the candidate servers.

6. Scope of Load Information

The "scope" of load information defines what the load indication applies to. For example, load could apply to a whole Diameter node, or a node could report different load for different application. It might be possible to have a load value for a whole realm, or a group of nodes.

[I-D.roach-dime-overload-ctrl] has a very expressive concept of scope, which applies both to load and overload information. It defines the scopes of "Destination-Realm", "Application-ID", "Destination-Host", "Host", "Connection", "Session", and "Session-Group". Scopes can be combined.

[I-D.tschofenig-dime-dlba] does not have an explicit concept of scope. Load information describes the load of a server for all Diameter purposes.

[I-D.korhonen-dime-ovl] defines several scopes for overload information. However, load information applies to the a whole node.

The author's opinion is that the load level of a Diameter node will usually apply to the whole node. Thus, the working group should consider a single "whole node" scope for load information. Alternatively, a "per-connection" scope could simulate "whole node" scope without requiring the recipient to pay attention to whether multiple transport connections terminate at the same peer.

7. Load Information Semantics

Both [I-D.tschofenig-dime-dlba] and [I-D.korhonen-dime-ovl] define load level to be a range between zero and some maximum value, where zero means no load at all and the max value means fully loaded. The former uses a range of 0-10, while the later uses 0-100

[I-D.roach-dime-overload-ctrl] treats load information as a strictly relative weighting factor. The weight is only meaningful when load-balancing across multiple destinations. That is, a maximum load value does not necessarily imply that the node is cannot handle more traffic. The load level scale is zero to 65535. That scale was chosen to match the resolution of the weight field from a DNS SRV record, [RFC2782]

8. Is Negotiation of Support Needed?

The working group should discuss whether a load conveyance mechanism requires negotiation or declaration of support. Several considerations apply to this discussion.

If load information is treated as a hint, it can be safely ignored by nodes that don't understand it. However, security considerations may apply if load information is accidentally leaked across a non-supporting node to a node that is not authorized to receive it.

If load information is conveyed using a dedicated Diameter application, the normal mechanisms for negotiation support for Diameter applications apply. However, the Diameter Capabilities Exchange [RFC6733] mechanism is inherently peer-to-peer. If there is an need to convey load information across a node that does not understand the mechanism, the standard Diameter mechanism would involve probing by support by sending load requests and watching for error answers with a result code of DIAMETER_APPLICATION_UNSUPPORTED. If the probe request also includes load information, there is again a potential for leaking load information to unauthorized parties.

If load information was treated in a strictly peer-to-peer fashion, there would be no need to probe to see if non-adjacent nodes support the mechanism. However, there would still be a need to control whether a non-supporting node would leak load information. Such a leak could be prevented if adjacent peers declared support, and never sent load information to a peer that did not declare support.

A peer-to-peer mechanism would also need a way to make sure that, if load information leaked across a non-supporting node, the receiving node would not mistakenly think the information came from the non-supporting node. This could be mitigated with a mechanism to declare

support as in the previous paragraph, or with a mechanism to identify the origin of the load information. In the latter case, the receiving node would treat any load information as invalid if the origin of that information did not match the identity of the peer node.

9. Security Considerations

Load information may be sensitive information in some cases. Depending on the mechanism. an unauthorized recipient might be able to infer the topology of a Diameter network from load information. Load information might be useful in identifying targets for Denial of Service (DoS) attacks, where a node known to be already heavily loaded might be a tempting target. Load information might also be useful as feedback about the success of an ongoing DoS attack.

Any load information conveyance mechanism will need to allow operators to avoid sending load information to nodes that are not authorized to receive it. Since Diameter currently only offers authentication of nodes at the transport level, any solution that sends load information to non-peer nodes might require a transitive-trust model.

10. IANA Considerations

This document makes no requests of IANA.

11. References

11.1. Normative References

[I-D.ietf-dime-ovli]
Korhonen, J., Donovan, S., Campbell, B., and L. Morand, "Diameter Overload Indication Conveyance", draft-ietf-dime-ovli-03 (work in progress), July 2014.

[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.

11.2. Informative References

[I-D.korhonen-dime-ovl]
Korhonen, J. and H. Tschofenig, "The Diameter Overload Control Application (DOCA)", draft-korhonen-dime-ovl-01 (work in progress), February 2013.

[I-D.roach-dime-overload-ctrl]
Roach, A. and E. McMurry, "A Mechanism for Diameter Overload Control", draft-roach-dime-overload-ctrl-03 (work in progress), May 2013.

[I-D.tschofenig-dime-dlba]
Tschofenig, H., "The Diameter Load Balancing Application (DLBA)", draft-tschofenig-dime-dlba-00 (work in progress), July 2013.

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

Author's Address

Ben Campbell
Oracle
7460 Warren Parkway # 300
Frisco, Texas 75034
USA

Email: ben@nostrum.com