

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: March 3, 2007

T. Asveren  
Ulticom Inc.  
August 30, 2006

**Diameter Duplicate Detection Cons.  
draft-asveren-dime-dupcons-00.txt**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 3, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

Diameter transport mechanism relies on storing data about received requests to detect duplicate requests. This document discusses implementation and deployment considerations regarding this functionality.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Reasons for Duplicate Requests . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	Restart of Client . . . . .	<a href="#">3</a>
<a href="#">2.2.</a>	Restart of Intermediate Node . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Arrival of Retransmission Before Original Request . . . . .	<a href="#">4</a>
<a href="#">4.</a>	Duplicate Detection Implementation Guidelines . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	Buffering of Requests with T-bit not Set . . . . .	<a href="#">4</a>
<a href="#">4.2.</a>	Buffering of Requests with T-bit Set . . . . .	<a href="#">6</a>
<a href="#">4.3.</a>	End-to-End Id Selection . . . . .	<a href="#">6</a>
<a href="#">4.4.</a>	Retransmission of First Request in a Session . . . . .	<a href="#">7</a>
<a href="#">5.</a>	IANA Considerations . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">7</a>
<a href="#">7.</a>	Acknowledgments . . . . .	<a href="#">7</a>
<a href="#">8.</a>	Normative References . . . . .	<a href="#">7</a>
	Author's Address . . . . .	<a href="#">7</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">8</a>



## 1. Introduction

Diameter Base Protocol[1] defines the transport mechanism to be used for sending/receiving requests/answers. The capability to detect duplicate requests is also included in this mechanism to prevent multiple processing of the same request. This capability relies on storing data about received requests on the server. Origin-Host AVP and End-to-End Identifiers of received messages need to be stored for duplicate detection. If the application is unable to regenerate the exact answer which was sent for the initial request, the answer message itself needs to be stored as well.

## 2. Reasons for Duplicate Requests

Duplicate requests may be received due to client or intermediate node restarts.

### 2.1. Restart of Client

When a client fails, it may retransmit requests, which were sent before the failure but for which no corresponding answer has been received yet. This may cause a server to receive the same request twice.

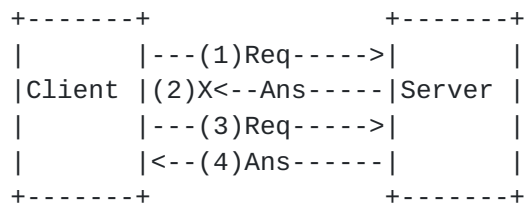


Figure 1: Retransmission of Request After Client Restart

- (1) Client sends a request, request is received by server.
- (2) Client goes down but before this is detected by the server, server sends back the corresponding answer.
- (3) Client restarts, and resends the request with T-bit set.

### 2.2. Restart of Intermediate Node

When an intermediary node in the path from client to server fails, the node before it needs to retransmit requests, for which no corresponding answer has been received yet.



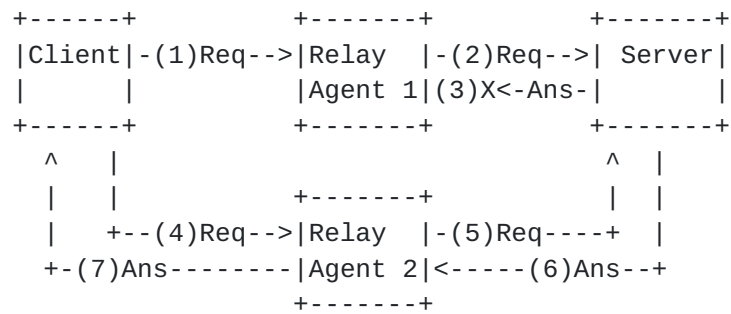


Figure 2: Retransmission of Request After Relay Agent Failure

- (1) Client sends the request to Relay Agent 1.
- (2) Relay Agent 1 forwards the request to server.
- (3) Relay Agent 1 goes down and before the server can detect it, the server sends the answer.
- (4) Client detects that Relay Agent 1 went down and retransmits the request to Relay Agent 2.
- (5) Relay Agent 2 forwards the request to the server.
- (6) Server sends the answer message to Relay Agent 2.
- (7) Relay Agent 2 forwards the answer to the client.

### 3. Arrival of Retransmission Before Original Request

A retransmitted request may arrive to the server before the corresponding original request. This may happen due to requests taking different paths in the diameter or IP networks. Because of the latter, even a client and server, which are directly connected from diameter point of view may observe retransmitted requests arriving before the original ones, if the client restarts.

## 4. Duplicate Detection Implementation Guidelines

### 4.1. Buffering of Requests with T-bit not Set

Origin-Host AVP and End-to-End Identifier for all requests received by a server MUST be saved, until it is guaranteed that no corresponding retransmission will be received. If the server is unable to regenerate the exact answer which was sent as response to the original request, this answer message MUST be saved as well.

Diameter base protocol does not provide a mechanism, by which a server can detect that an answer message has been received by the client, which sent the corresponding request message. This would have indicated the server that buffered information for that request could be deleted because from that moment on no retransmissions for



that request are possible.

Implementations MAY configure a value for the maximum time, after which no retransmission of a request will arrive, e.g. maximum expected downtime for any client + maximum network delay. Although such a value could be only a guess and needs to be configured generously to prevent non-detection of retransmissions, it still MAY be used to decide when buffered information can be deleted.

Client failures could be hardware related, where replacement of equipment may be necessary. Such cases could result downtimes of a few hours. This would cause buffering of large amounts of data on servers. For example consider a server which handles 1000 messages per second, which can't regenerate answers:

length of End-to-End Identifier: 4 bytes

average answer length: 280 bytes

average Origin-Host AVP length: 16 bytes

maximum buffering time: 2 hours

amount of data to be buffered:  $1000 * 7200(4+16+280) \sim 2 \text{ GBytes}$

Especially with larger answer messages, amount of data to be buffered can get much bigger. Usually, that type of memory requirement is considered undesirable. Implementation MAY choose to store this information in non-volatile memory but frequent writes to non-volatile memory can cause a significant performance penalty.

Applications MAY use new requests arriving from a peer as indirect acknowledgements to decrease the amount of data buffered for duplicate detection, if a value can be configured as the maximum end-to-end delay in the Diameter network. Each new request MAY be interpreted as that answers sent 2\*maximum end-to-end delay ago are received by the originator of the request, and buffered data associated with the corresponding request can be deleted. This technique could decrease memory requirements for duplicate detection significantly but it should be noted that it MAY cause failure to detect duplicates, if maximum end-to-end delay is not chosen carefully.

Applications MAY try to guess end-to-end delay between two peers dynamically. This can be achieved by sending an invalid message to other peers and measuring the time difference between sending the message and receiving corresponding error answer. By considering multiple measurements and providing a generous buffer, the calculated value can be utilized while using requests as implicit acknowledgements.





#### **4.2. Buffering of Requests with T-bit Set**

Information related with requests with T-bit set MUST be buffered as well, if the original request is not received yet, because it is possible for a retransmission to arrive before the corresponding original request. In such a case, the original request MUST be treated as a duplicate.

Information buffered for requests with T-bit SHOULD be buffered as long as the expected maximum network delay. Usually this value could be around a few seconds and considering that requests with T-bit set are rare, it is not expected that memory requirements will be high.

#### **4.3. End-to-End Id Selection**

End-to-End Identifier is important from duplicate detection point of view because it uniquely identifies requests sent by a specific peer.

Diameter base protocol mandates that End-to-End Id must be unique at least for a period of 4 minutes. This MAY cause false duplicate detections, if a client goes down for more than 4 minutes, because a retransmission of a request from the previous boot-cycle and a new request MAY have the same End-to-End Id.

Considering that End-to-End Id is 32-bits, the duration of its uniqueness can be generated as a function of average number of messages per second and minimum restart time. Enough bits need to be allocated to distinguish between each message in a boot cycle and between boot cycles.

The uniqueness period  $t_u$  MUST satisfy the following inequality:  
$$32 \geq \text{ceiling}[\log_2(\text{msg\_rate} * t_u)] + \text{ceiling}[\log_2(t_u / \text{min\_restart})]$$

For example:

message rate: 500 msg/sec min\_restart: 1 sec

A uniqueness period of 1035 seconds could be guaranteed:

$$\text{ceiling}[\log_2(500 * 1035)] + \text{ceiling}[\log_2(1035)] =$$
$$19 + 11 = 30 < 32$$

Even if uniqueness of End-to-End Id is guaranteed for more than 4 minutes, as long as uniqueness period is less than the maximum expected downtime, false duplicate detections MAY occur but longer uniqueness periods statistically will decrease the probability of that to happen.

Implementations MAY consider Session-Id as well to decrease the possibility of false duplicate detections, in addition to End-to-End Id and Origin-Host AVP.



#### **4.4. Retransmission of First Request in a Session**

First requests in a session are different than the subsequent ones, because the first requests MAY NOT contain Destination-Host AVP. In such a case, the request is routed based on Destination-Realm AVP and Application-Id.

Considering that information about request are buffered at the server where they have been sent, retransmission of a request SHOULD be sent to the same server so that duplicate detection can be performed. To guarantee this type of behavior, all Diameter nodes SHOULD guarantee that all requests with the same End-to-End Id are sent to the same next hop.

#### **5. IANA Considerations**

This document does not require any action from IANA.

#### **6. Security Considerations**

This document does not introduce new security considerations and the considerations given in [RFC3588](#) [1] do apply.

#### **7. Acknowledgments**

The author would like to thank David Lehmann for his invaluable comments.

#### **8. Normative References**

- [1] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", [RFC 3588](#), September 2003.

#### **Author's Address**

Tolga Asveren  
Ulticom Inc.  
1020 Briggs Road  
Mount Laurel, NJ, 08054  
USA

Email: [asveren@ulticom.com](mailto:asveren@ulticom.com)



## Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

