

Individual  
Internet-Draft  
Expires: January 4, 2006

E. Burger  
Brooktrout Technology, Inc.  
A. Melnikov  
Isode Ltd.  
July 03, 2005

SMTP Service Extension for Reliable Delivery  
draft-burger-smtp-rdlv-01

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/lid-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 January 4, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

There is an issue with SMTP that [RFC 1047](#) raised in 1988. The time between a SMTP client submitting a mail object and the SMTP server responding to the request can be arbitrarily long. SMTP addresses this issue by a hack, hoping that the SMTP server responds fast enough and the SMTP client waits long enough to find out if the submission was successful.

Internet-Draft

SMTP RDLV

July 2005

Unfortunately, this approach is, by its nature, unreliable. Besides the duplicate delivery problem identified by [RFC 1047](#), this behavior introduces serious human factors problems for submission servers.

This document describes a SMTP Service Extension that enables the SMTP client to reliably know that the SMTP server is alive, well, and working on the message submission.

#### Conventions used in this document

[RFC 2119](#) [[RFC2119](#)] provides the interpretations for the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" found in this document.

#### Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Overview of the Extension . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Formal Description of the Extension . . . . .	<a href="#">4</a>
<a href="#">3.1</a>	Extension Definition . . . . .	<a href="#">4</a>
<a href="#">3.2</a>	Extension Operation . . . . .	<a href="#">5</a>
<a href="#">3.2.1</a>	Server Behavior . . . . .	<a href="#">5</a>
<a href="#">3.2.2</a>	Client Behavior . . . . .	<a href="#">6</a>
<a href="#">4.</a>	Formal Syntax . . . . .	<a href="#">7</a>
<a href="#">5.</a>	IANA Considerations . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Examples . . . . .	<a href="#">7</a>
<a href="#">6.1</a>	Long Submit . . . . .	<a href="#">7</a>
<a href="#">6.2</a>	Unsupported Verb . . . . .	<a href="#">8</a>
<a href="#">6.3</a>	Remote Submit . . . . .	<a href="#">9</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">10</a>
<a href="#">8.</a>	References . . . . .	<a href="#">10</a>
<a href="#">8.1</a>	Normative References . . . . .	<a href="#">10</a>
<a href="#">8.2</a>	Informative References . . . . .	<a href="#">11</a>
	Authors' Addresses . . . . .	<a href="#">12</a>
<a href="#">A.</a>	Acknowledgements . . . . .	<a href="#">12</a>
	Intellectual Property and Copyright Statements . . . . .	<a href="#">13</a>

Internet-Draft

SMTP RDLV

July 2005

## 1. Introduction

The Simple Mail Transfer Protocol, SMTP [[RFC2821](#)], has over twenty years of field experience. SMTP is one of the most widely implemented application protocols in the Internet. Virtually every Internet host of every kind -- mainframe, minicomputer, microcomputer, PDA, or even wristwatch -- interacts with the Internet Mail system.

[RFC 1047](#) [[RFC1047](#)] identified a problem with the message submission protocol in SMTP. Namely, the SMTP server has to guarantee that it has responsibility for the message once it issues the 250 response at the end of the DATA request. However, the time from the final dot indicating the end of the DATA request to the actual 250 response can be arbitrarily long.

From the perspective of the SMTP client, it cannot discern between the following situations:

- o The SMTP server process has crashed, but the TCP connection is still up.
- o The SMTP host has crashed, but the TCP connection has not yet timed-out.
- o The SMTP server is under great load, and will get to the message soon.
- o The message is "big" in some respect, such as message size or number of recipients. Thus the SMTP server, even under normal circumstances, requires a significant amount of time to process the message.

The approach espoused by [RFC 2821](#) is for the server to respond as fast as possible to the request and for the client to wait long enough for the response.

Over the years, this has been the staple approach. [RFC 1123](#) [[RFC1123](#)] reiterates that the server should respond "quickly" and the client should "wait long enough." LMTP [[RFC2033](#)] shares the syntax of SMTP, and shares the submission timeout problem. The advice

offered in LMTP is that if it will take a long time to submit an object, one should not use LMTP. Even [RFC 3464](#) [[RFC3464](#)] on Delivery Status Notifications, points out that duplicate submissions can occur and one should just "live with it."

Besides the obvious problem of an unreliable protocol, in the Lemonade working group we have run into a human factors problem made worse by this time out issue. There are extensions being proposed, such as BURL [[burl](#)], which can suffer from the same lack of protocol support for long requests. This impacts the user's experience, especially when they have intermittent connectivity, as experienced on wide-area wireless networks. The user has no idea if a submission

request is taking a long time because the server needs a long time or if they have lost their connection.

What we need is a method for the SMTP server to inform the SMTP client that the server is alive and still working on the submission.

## [2.](#) Overview of the Extension

This document introduces the RDLV extension. When indicated, SMTP servers implementing the RDLV extension periodically issue a 120 response. The SMTP client then issues a CONT command to continue the operation.

DISCUSSION POINT: Ideally, the response should be truly preliminary and not change the SMTP state machine. If the server is under load, the last thing it needs is more handshake protocol actions to perform. I could see having the server simply periodically issuing 120 responses. However, in this case neither RSET to cancel nor DETA to detach would work. Moreover, the proposal in this document follows the VERB-RESULT model of SMTP. Periodic emission of 120's changes the model to VERB-InterimRESULT-InterimRESULT-FinalRESULT. On the other hand, This extension adds sub-states to potentially all long commands, which is a major change, as well.

## [3.](#) Formal Description of the Extension

### [3.1](#) Extension Definition

1. The text name of this SMTP service extension is "Reliable

- Delivery".
2. The EHLO keyword associated with this SMTP service extension is "RDLV".
  3. The RDLV EHLO keyword has one parameter, mintimer. Mintimer specifies the minimum amount of time in integer seconds the server will wait before issuing a 120 response to a long command.
  4. This extension doesn't add any new parameters to MAIL or RCPT commands.
  5. This document defines a new SMTP verb, RDLV. The RDLV verb indicates that the following transmission verb, such as DATA, BDAT [[RFC3030](#)], and so on is to use the procedures described in this document for generating 120 responses.

DISCUSSION POINT: We are really describing an adverb here, not a verb. On the one hand, this should be a new parameter to each of the "takes a long time" verbs. On the other hand, it is much easier to describe the behavior if we keep them separate.

Conversely, we could define RDLV to be for the entire session, not just the following command. What do we think of that?

6. The RDLV verb takes a single parameter, maxtimer. Maxtimer specifies the maximum amount of time in integer seconds the client will wait for a response from the server before assuming the server is not available.
7. This service extension is appropriate for the standard SMTP [[RFC2821](#)], SMTP submission protocol [[RFC2476](#)] and LMTP [[RFC2033](#)].

## [3.2](#) Extension Operation

### [3.2.1](#) Server Behavior

Upon receiving a RDLV verb, the server examines the maxtimer parameter. If maxtimer is less than mintimer, the server SHOULD respond with a 501 (5.5.4 Bad Parameter). The server MAY elect to honor the shorter timer. However, especially under load, the server should not accept undue burdens from clients.

If there is no maxtimer parameter, the server MUST respond with a 501 (5.5.2 Missing parameter).

If the maxtimer parameter is acceptable, the server MUST respond with a 250 response code and set the server's state to issue 120 responses to the following long command, at least every maxtimer seconds.

If, when the client issues the next command, the server determines it is not able to issue 120 responses to the particular command, the server MUST respond with a 520 (5.3.3 System not capable of selected features [Is this correct?]) response code. The server MUST terminate both the RDLV and the last received command request.

The server MUST set its interval timer to something less than maxtimer to ensure that the time from the client's perspective, including transit delays, is less than maxtimer. A reasonable amount of time to reduce maxtimer is 7 seconds.

If the server receives a EHLO, HELO, or RSET, it MUST clear the server's state, including terminating the sending 120 responses.

If the server completes processing the long command, the server responds to the client with the appropriate response code. If the server has not completed processing in maxtimer seconds since the last response, the server MUST issue a 120 response.

When the server issues the 120 response, the server waits for the client to issue a CONT request. The server SHOULD continue processing while waiting for the CONT request.

When the server receives the CONT request, the server MUST continue processing, and the server MUST issue another 120 response (now to the CONT request) within maxtimer seconds.

If the server does not receive a CONT request within maxtimer seconds, the server MUST consider the client gone.

After the long transaction completes or the client abandons the transaction, the server MUST set its state to not issue 120 responses, unless requested again by a new RDLV request.

### [3.2.2](#) Client Behavior

To receive notifications from the server that the server is still working on a long transaction, the client MUST issue a RDLV request

before the long transaction request, such as DATA, BDAT, or BURL. The RDLV request MUST have a maxtimer parameter indicating the maximum number of seconds the server can wait before issuing a 120 request. Maxtimer MUST NOT be less than the mintimer parameter issued by the server in the EHLO keyword.

The client then issues the long transaction request as normal. The server will respond with the normal result code for the request or a 120 response.

To continue the long transaction, the client MUST issue a CONT request. Issuing other requests, or ignoring the 120 response, is not defined, except for the client issuing a new EHLO, HELO, or RSET requests. Those requests reset the state of the server, as specified in SMTP [[RFC2821](#)].

The client MUST wait a reasonable amount of time, in addition to maxtimer seconds, for the 120 response. This is to account for congested transit delays between the client and server. A reasonable amount of time to wait before abandoning the transaction is 7 seconds.

Server generation of 120 responses is per-transaction. Thus, if the client wishes to receive 120 responses on a new long request, the client MUST issue a new RDLV request before issuing the long transaction request.

If the server is unable to comply with the RDLV request for a particular verb, the server will respond with a 520 (5.3.3 System not capable of selected features) response code to the verb. In this case the server has rejected both the RDLV and the verb request. The client is free to make the verb request again without preceding it with a RDLV request.

If the server also supports PIPELINING [[RFC3030](#)] SMTP extension, then the RDLV command MUST be the last command in a group of pipelined commands.

#### [4.](#) Formal Syntax

Formal syntax is defined using ABNF [[ABNF](#)]. Elements not defined here can be found in the [[ABNF](#)] and SMTP [[RFC2821](#)].

```
seconds      = 1*DIGIT
               ; 31-bit positive integer
               ; (1 <= n < 2,147,483,648)

rdlv-param    = "mintimer" "=" seconds
               ; parameter to RDLV EHLO keyword.
               ; matches "ehlo-keyword"

rdlv-cmd      = "RDLV" SP "maxtimer" "=" seconds CRLF

cont-cmd      = "CONT"
```

## [5.](#) IANA Considerations

Please register the RDLV service extension in the Simple Mail Transfer Protocol (SMTP) Service Extensions table as follows.

### Service Extension

Keyword: RDLV  
Description: Reliable Delivery  
Reference: RFCXXXX

### Service Extension Parameters

Service Extension: Reliable Delivery  
EHLO Keyword: RDLV  
Parameters: maxtimer  
Reference: RFCXXXX

## [6.](#) Examples

These examples are informative in nature. For any discrepancies between behavior here and the normative behavior, the normative behavior is correct.

### [6.1](#) Long Submit

This addresses the classic long submit problem. The client sets the RDLV to 600 seconds, the classic 10-minute timeout. Rather than

giving up at 10 minutes, the client watches for the 120. Upon



receiving the 120, it knows the server is still alive. The client requests a continuation of the request, and after 22 minutes, the submission succeeds. Compare that to the classic way, where the submission would fail.

```
S: 220 example.net SMTP XYZ Ready
C: EHLO example.com
S: 250-example.net greets example.com
S: 250-8BITMIME
S: 250-BURL imap
S: 250-RDLV mintimer=30
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<smith@example.com>
S: 250 OK
C: RCPT TO:<jones@example.net>
S: 250 OK
C: RDLV maxtimer=600
S: 200 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: [... lots and lots of data ...]
C:
C: .
[about 10 minutes pass]
S: 120 DATA processing in progress
C: CONT
[about 10 minutes pass]
S: 120 DATA processing in progress
C: CONT
[about 2 minutes pass]
S: 250 OK
C: QUIT
S: 221 example.net Service closing transmission channel
```

## [6.2](#) Unsupported Verb

The client requests in-process notifications for a method that the server does not support in-process notifications for the following verb, BURL in this case.

```
S: 220 example.net SMTP XYZ Ready
C: EHLO example.com
S: 250-example.net greets example.com
S: 250-8BITMIME
S: 250-BURL imap
S: 250-RDLV mintimer=30
S: 250-DSN
S: 250-ENHANCEDSTATUSCODES
S: 250 HELP
C: MAIL FROM:<smith@example.com>
S: 250 2.5.0 OK
C: RCPT TO:<jones@example.net>
S: 250 2.1.5 OK
C: RDLV maxtimer=600
S: 200 2.5.0 OK
C: BURL imap://handset@host.example.com/outbox
    ;uidvalidity=a9j230r932/;uid=32
    ;authid=fred;urlauth=submit+handset
    :internal:91354a473744909de610943775f92038 LAST
S: 520 5.3.3 This server does not support RDLV for BURL
C: QUIT
S: 221 2.0.0 example.net Service closing transmission channel
```

### [6.3](#) Remote Submit

This addresses the problem raised by the Lemonade WG. The client issues a very small request, a BURL. However, the object referenced by the BURL is quite large. The client sets the RDLV to 60 seconds, so it keeps aware of the progress of message.

Internet-Draft

SMTP RDLV

July 2005

```
S: 220 example.net SMTP XYZ Ready
C: EHLO example.com
S: 250-example.net greets example.com
S: 250-8BITMIME
S: 250-BURL imap
S: 250-RDLV mintimer=30
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<handset@example.com>
S: 250 OK
C: RCPT TO:<jones@example.net>
S: 250 OK
C: RDLV maxtimer=60
S: 200 OK
C: BURL imap://handset@host.example.com/outbox
    ;uidvalidity=a9j230r932/;uid=32
    ;authid=fred;urlauth=submit+handset
    ;internal:91354a473744909de610943775f92038 LAST
[about a minute passes]
S: 120 DATA processing in progress
C: CONT
[about a minute passes]
S: 120 DATA processing in progress
C: CONT
[about a minute passes]
S: 120 DATA processing in progress
C: CONT
[about a minute passes]
S: 120 DATA processing in progress
C: CONT
S: 250 OK
C: QUIT
S: 221 example.net Service closing transmission channel
```

## 7. Security Considerations

Malicious clients can amplify server load by issuing unreasonably high notification rates in addition to sending large, complex

documents. Servers MUST ensure to deny client requests for unduly short inter-notification times.

## 8. References

### 8.1 Normative References

- [ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

Burger & Melnikov Expires January 4, 2006 [Page 10]

---

Internet-Draft SMTP RDLV July 2005

#### [PIPELINING]

Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, [RFC 2920](#), September 2000.

- [RFC2033] Myers, J., "Local Mail Transfer Protocol", [RFC 2033](#), October 1996.

- [RFC2034] Freed, N., "SMTP Service Extension for Returning Enhanced Error Codes", [RFC 2034](#), October 1996.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

- [RFC2476] Gellens, R. and J. Klensin, "Message Submission", [RFC 2476](#), December 1998.

- [RFC2821] Klensin, J., "Simple Mail Transfer Protocol", [RFC 2821](#), April 2001.

- [RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", [RFC 3464](#), January 2003.

### 8.2 Informative References

- [RFC1047] Partridge, C., "Duplicate messages and SMTP", [RFC 1047](#), February 1988.

- [RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), October 1989.

- [RFC3030] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", [RFC 3030](#),

December 2000.

- [RFC3464] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", [RFC 3464](#), January 2003.
- [burl] Newman, C., "Message Submission BURL Extension", Internet Draft [draft-ietf-lemonade-burl-00.txt](#), July 2004.

Burger & Melnikov

Expires January 4, 2006

[Page 11]

---

Internet-Draft

SMTP RDLV

July 2005

#### Authors' Addresses

Eric Burger  
Brooktrout Technology, Inc.  
18 Keewaydin Dr.  
Salem, NH 03079  
USA

Email: [e.burger@ieee.org](mailto:e.burger@ieee.org)

Alexey Melnikov  
Isode Ltd.  
5 Castle Business Village  
36 Station Road  
Hampton, Middlesex TW12 2BX  
GB

Email: [alexey.melnikov@isode.com](mailto:alexey.melnikov@isode.com)

#### [Appendix A](#). Acknowledgements

The need for notification that the SMTP server is still doing work came from the lemonade interim meeting in May of 2004 in Richardson, TX, USA.

#### Intellectual Property Statement

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).

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

## Disclaimer of Validity

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.

## Copyright Statement

Copyright (C) The Internet Society (2005). 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.

Burger & Melnikov	Expires January 4, 2006	[Page 13]
-------------------	-------------------------	-----------

---

Internet-Draft	SMTP RDLV	July 2005
----------------	-----------	-----------

## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

