Network Working Group
Request For Comments: 1870
STD: 10
Obsoletes: 1653
Category: Standards Track

J. Klensin, WG Chair
MCI
N. Freed, Editor
Innosoft International, Inc.
K. Moore
University of Tennessee

November 1995

SMTP Service Extension for Message Size Declaration

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

1. Abstract

This memo defines an extension to the SMTP service whereby an SMTP client and server may interact to give the server an opportunity to decline to accept a message (perhaps temporarily) based on the client's estimate of the message size.

2. Introduction

The MIME extensions to the Internet message protocol provide for the transmission of many kinds of data which were previously unsupported in Internet mail. One expected result of the use of MIME is that SMTP will be expected to carry a much wider range of message sizes than was previously the case. This has an impact on the amount of resources (e.g. disk space) required by a system acting as a server.

This memo uses the mechanism defined in [5] to define extensions to the SMTP service whereby a client ("sender-SMTP") may declare the size of a particular message to a server ("receiver-SMTP"), after which the server may indicate to the client that it is or is not willing to accept the message based on the declared message size and whereby a server ("receiver-SMTP") may declare the maximum message size it is willing to accept to a client ("sender-SMTP").

3. Framework for the Size Declaration Extension

The following service extension is therefore defined:

(1) the name of the SMTP service extension is "Message Size Declaration";

(2) the EHLO keyword value associated with this extension is "SIZE";

(3) one optional parameter is allowed with this EHLO keyword value, a decimal number indicating the fixed maximum message size in bytes that the server will accept. The syntax of the parameter is as follows, using the augmented BNF notation of [2]:

```
size-param ::= [1*DIGIT]
```

A parameter value of 0 (zero) indicates that no fixed maximum message size is in force. If the parameter is omitted no information is conveyed about the server's fixed maximum message size;

(4) one optional parameter using the keyword "SIZE" is added to the MAIL FROM command. The value associated with this parameter is a decimal number indicating the size of the message that is to be transmitted. The syntax of the value is as follows, using the augmented BNF notation of [2]:

```
size-value ::= 1*20DIGIT
```

(5) the maximum length of a MAIL FROM command line is increased by 26 characters by the possible addition of the SIZE keyword and value;

(6) no additional SMTP verbs are defined by this extension.

The remainder of this memo specifies how support for the extension affects the behavior of an SMTP client and server.

4. The Message Size Declaration service extension

An SMTP server may have a fixed upper limit on message size. Any attempt by a client to transfer a message which is larger than this fixed upper limit will fail. In addition, a server normally has limited space with which to store incoming messages. Transfer of a message may therefore also fail due to a lack of storage space, but might succeed at a later time.

A client using the unextended SMTP protocol defined in [1], can only be informed of such failures after transmitting the entire message to the server (which discards the transferred message). If, however, both client and server support the Message Size Declaration service extension, such conditions may be detected before any transfer is attempted.

An SMTP client wishing to relay a large content may issue the EHLO command to start an SMTP session, to determine if the server supports any of several service extensions. If the server responds with code 250 to the EHLO command, and the response includes the EHLO keyword value SIZE, then the Message Size Declaration extension is supported.

If a numeric parameter follows the SIZE keyword value of the EHLO response, it indicates the size of the largest message that the server is willing to accept. Any attempt by a client to transfer a message which is larger than this limit will be rejected with a permanent failure (552) reply code.

A server that supports the Message Size Declaration extension will accept the extended version of the MAIL command described below. When supported by the server, a client may use the extended MAIL command (instead of the MAIL command as defined in [1]) to declare an estimate of the size of a message it wishes to transfer. The server may then return an appropriate error code if it determines that an attempt to transfer a message of that size would fail.

5. Definitions

The message size is defined as the number of octets, including CR-LF pairs, but not the SMTP DATA command's terminating dot or doubled quoting dots, to be transmitted by the SMTP client after receiving reply code 354 to the DATA command.

The fixed maximum message size is defined as the message size of the largest message that a server is ever willing to accept. An attempt to transfer any message larger than the fixed maximum message size will always fail. The fixed maximum message size may be an implementation artifact of the SMTP server, or it may be chosen by the administrator of the server.

The declared message size is defined as a client's estimate of the message size for a particular message.

6. The extended MAIL command

The extended MAIL command is issued by a client when it wishes to inform a server of the size of the message to be sent. The extended MAIL command is identical to the MAIL command as defined in [1], except that a SIZE parameter appears after the address.

The complete syntax of this extended command is defined in [5]. The esmtp-keyword is "SIZE" and the syntax for esmtp-value is given by the syntax for size-value shown above.

The value associated with the SIZE parameter is a decimal representation of the declared message size in octets. This number should include the message header, body, and the CR-LF sequences between lines, but not the SMTP DATA command's terminating dot or doubled quoting dots. Only one SIZE parameter may be specified in a single MAIL command.

Ideally, the declared message size is equal to the true message size. However, since exact computation of the message size may be infeasable, the client may use a heuristically-derived estimate. Such heuristics should be chosen so that the declared message size is usually larger than the actual message size. (This has the effect of making the counting or non-counting of SMTP DATA dots largely an academic point.)

NOTE: Servers MUST NOT use the SIZE parameter to determine end of content in the DATA command.

6.1 Server action on receipt of the extended MAIL command

Upon receipt of an extended MAIL command containing a SIZE parameter, a server should determine whether the declared message size exceeds its fixed maximum message size. If the declared message size is smaller than the fixed maximum message size, the server may also wish to determine whether sufficient resources are available to buffer a message of the declared message size and to maintain it in stable storage, until the message can be delivered or relayed to each of its recipients.

A server may respond to the extended MAIL command with any of the error codes defined in [1] for the MAIL command. In addition, one of the following error codes may be returned:

(1) If the server currently lacks sufficient resources to accept a message of the indicated size, but may be able to accept the message at a later time, it responds with code "452 insufficient system storage".

(2) If the indicated size is larger than the server's fixed maximum message size, the server responds with code "552 message size exceeds fixed maximium message size".

A server is permitted, but not required, to accept a message which is, in fact, larger than declared in the extended MAIL command, such as might occur if the client employed a size-estimation heuristic which was inaccurate.

6.2 Client action on receiving response to extended MAIL command

The client, upon receiving the server's response to the extended MAIL command, acts as follows:

(1) If the code "452 insufficient system storage" is returned, the client should next send either a RSET command (if it wishes to attempt to send other messages) or a QUIT command. The client should then repeat the attempt to send the message to the server at a later time.

(2) If the code "552 message exceeds fixed maximum message size" is received, the client should immediately send either a RSET command (if it wishes to attempt to send additional messages), or a QUIT command. The client should then declare the message undeliverable and return appropriate notification to the sender (if a sender address was present in the MAIL command).

A successful (250) reply code in response to the extended MAIL command does not constitute an absolute guarantee that the message transfer will succeed. SMTP clients using the extended MAIL command must still be prepared to handle both temporary and permanent error reply codes (including codes 452 and 552), either immediately after issuing the DATA command, or after transfer of the message.

6.3 Messages larger than the declared size.

Once a server has agreed (via the extended MAIL command) to accept a message of a particular size, it should not return a 552 reply code after the transfer phase of the DATA command, unless the actual size of the message transferred is greater than the declared message size. A server may also choose to accept a message which is somewhat larger than the declared message size.

A client is permitted to declare a message to be smaller than its actual size. However, in this case, a successful (250) reply code is no assurance that the server will accept the message or has sufficient resources to do so. The server may reject such a message after its DATA transfer.

6.4 Per-recipient rejection based on message size.

A server that implements this extension may return a 452 or 552 reply code in response to a RCPT command, based on its unwillingness to accept a message of the declared size for a particular recipient.

(1) If a 452 code is returned, the client may requeue the message for later delivery to the same recipient.

(2) If a 552 code is returned, the client may not requeue the message for later delivery to the same recipient.

7. Minimal usage

A "minimal" client may use this extension to simply compare its (perhaps estimated) size of the message that it wishes to relay, with the server's fixed maximum message size (from the parameter to the SIZE keyword in the EHLO response), to determine whether the server will ever accept the message. Such an implementation need not declare message sizes via the extended MAIL command. However, neither will it be able to discover temporary limits on message size due to server resource limitations, nor per-recipient limitations on message size.

A minimal server that employs this service extension may simply use the SIZE keyword value to inform the client of the size of the largest message it will accept, or to inform the client that there is no fixed limit on message size. Such a server must accept the extended MAIL command and return a 552 reply code if the client's declared size exceeds its fixed size limit (if any), but it need not detect "temporary" limitations on message size.

The numeric parameter to the EHLO SIZE keyword is optional. If the parameter is omitted entirely it indicates that the server does not advertise a fixed maximum message size. A server that returns the SIZE keyword with no parameter in response to the EHLO command may not issue a positive (250) response to an extended MAIL command containing a SIZE specification without first checking to see if sufficient resources are available to transfer a message of the declared size, and to retain it in stable storage until it can be relayed or delivered to its recipients. If possible, the server should actually reserve sufficient storage space to transfer the message.

8. Example

The following example illustrates the use of size declaration with some permanent and temporary failures.

```
S: <wait for connection on TCP port 25>
C: <open connection to server>
S: 220 sigurd.innosoft.com -- Server SMTP (PMDF V4.2-6 #1992)
C: EHLO ymir.claremont.edu
S: 250-sigurd.innosoft.com
S: 250-EXPN
S: 250-HELP
S: 250 SIZE 1000000
C: MAIL FROM:<ned@thor.innosoft.com> SIZE=500000
S: 250 Address Ok.
C: RCPT TO:<ned@innosoft.com>
S: 250 ned@innosoft.com OK; can accomodate 500000 byte message
C: RCPT TO:<ned@ymir.claremont.edu>
S: 552 Channel size limit exceeded: ned@YMIR.CLAREMONT.EDU
C: RCPT TO:<ned@hmcvax.claremont.edu>
S: 452 Insufficient channel storage: ned@hmcvax.CLAREMONT.EDU
C: DATA
S: 354 Send message, ending in CRLF.CRLF.
 ...
C: .
S: 250 Some recipients OK
C: QUIT
S: 221 Goodbye
```

9. Security Considerations

The size declaration extensions described in this memo can conceivably be used to facilitate crude service denial attacks. Specifically, both the information contained in the SIZE parameter and use of the extended MAIL command make it somewhat quicker and easier to devise an efficacious service denial attack. However, unless implementations are very weak, these extensions do not create any vulnerability that has not always existed with SMTP. In addition, no issues are addressed involving trusted systems and possible release of information via the mechanisms described in this RFC.

10. Acknowledgements

This document was derived from an earlier Working Group work in progess contribution. Jim Conklin, Dave Crocker, Neil Katin, Eliot Lear, Marshall T. Rose, and Einar Stefferud provided extensive comments in response to earlier works in progress of both this and the previous memo.

11. References

[1] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, USC/Information Sciences Institute, August 1982.

[2] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, August 1982.

[3] Borenstein, N., and N. Freed, "Multipurpose Internet Mail Extensions", RFC 1521, Bellcore, Innosoft, September 1993.

[4] Moore, K., "Representation of Non-ASCII Text in Internet Message Headers", RFC 1522, University of Tennessee, September 1993.

[5] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extensions", STD 11, RFC 1869, MCI, Innosoft International, Inc., Dover Beach Consulting, Inc., Network Management Associates, Inc., Brandenburg Consulting, November 1995.

[6] Partridge, C., "Mail Routing and the Domain System", STD 14, RFC 974, BBN, January 1986.

12. Chair, Editor, and Author Addresses

John Klensin, WG Chair
MCI
2100 Reston Parkway
Reston, VA 22091

Phone: +1 703 715-7361
Fax: +1 703 715-7436
EMail: klensin@mci.net

Ned Freed, Editor
Innosoft International, Inc.
1050 East Garvey Avenue South
West Covina, CA 91790
USA

Phone: +1 818 919 3600
Fax: +1 818 919 3614
EMail: ned@innosoft.com

Keith Moore
Computer Science Dept.
University of Tennessee
107 Ayres Hall
Knoxville, TN 37996-1301
USA

EMail: moore@cs.utk.edu