Network Working Group
Request For Comments: 1845
Category: Experimental

D. Crocker
Brandenburg Consulting
N. Freed
Innosoft International, Inc.
A. Cargille, WG Chair

September 1995

SMTP Service Extension for Checkpoint/Restart

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. This memo does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Abstract

This memo defines an extension to the SMTP service whereby an interrupted SMTP transaction can be restarted at a later time without having to repeat all of the commands and message content sent prior to the interruption.

1. Introduction

Although SMTP is widely and robustly deployed, various extensions have been requested by parts of the Internet community. In particular, when dealing with very large messages over less reliable connections it is possible for substantial resources to be consumed by repeated unsuccessful attempts to transmit the message in its entirety. The original SMTP specification [1] does not provide any means to pick up a partially completed transaction after the underlying TCP connection has been broken and reestablished.

This memo provides a facility by which a client can uniquely identify a particular SMTP transaction. The server then stores this identifying information along with all the information it receives as the transaction proceeds. If the transaction is interrupted during the data transfer phase the SMTP client may establish a new SMTP session at a later time and ask the server to continue the transaction from the point where the server lost its connection with the client. The server then reestablishes the transaction context and tells the client where to resume operations. If this is acceptable the client resumes operations at this point.

This extension may also be used to work around the common timeout problem where a client times out waiting for a response from the server acknowledging that the message has been accepted. However, use of this extension is not an acceptable substitute for proper setting of timeout parameters.

2. Framework for the Checkpointing Extension

The checkpointing extension is laid out as follows:

(1) the name of the SMTP service extension defined here is checkpointing;

(2) the EHLO keyword value associated with the extension is CHECKPOINT;

(3) no parameter is used with the CHECKPOINT EHLO keyword;

(4) one optional parameter using the keyword TRANSID is added to the MAIL FROM command. The value associated with this parameter, coupled with the name of the client taken from EHLO command, forms a globally unique value that identifies this particular transaction and serves to distinguish it from all others. This value is case-sensitive. The syntax of the value is as follows, using the ABNF notation of [2]:

```
transid-value  ::= "<" transid-spec ">"
                       ; transid-value may not be longer than
                       ; 80 characters
transid-spec   ::= transid-local "@" transid-domain
transid-domain ::= transid-token
transid-local  ::= transid-token
transid-token  ::= transid-atom *("." transid-atom)
transid-atom   ::= 1*<any (ASCII) CHAR except SPACE,
                          CTLs, tspecials, or ".">
```

NOTE: tspecials is defined in [3]. The TRANSID is likely to be different from the RFC822 message id, since it must uniquely identify the particular copy of the message being sent over this SMTP link. However, the syntax of transid-value is designed so that any TRANSID is both a legal RFC822 msg-id as well as being a legal esmtp-value [4].

(5) The maximum length of a MAIL FROM command line is increased by 88 characters by the possible addition of the TRANSID keyword and value;

(6) no additional SMTP verbs are defined by this extension; and,

(7) the next section specifies how support for the extension affects the behavior of a server and client SMTP.

3. The checkpointing service extension

When a client SMTP wishes to use checkpointing to eliminate the need to retransmit all message data in its entirety in the event of a session interruption, it first issues the EHLO command to the server SMTP. If the server SMTP responds with code 250 to the EHLO command, and the response includes the EHLO keyword value CHECKPOINT, then the server SMTP is indicating that it supports SMTP checkpointing and will honor requests to restart interrupted SMTP transactions.

The extended MAIL command is issued by a client SMTP when it wishes to enable server checkpointing. The syntax for this command is identical to the MAIL command in [1], except that a TRANSID parameter must appear after the address.

The complete syntax of this extended command is defined in [4], with the esmtp-keyword TRANSID and transid-value parameter as previously defined.

The value associated with the TRANSID parameter must be an identifier that serves to uniquely identify this particular SMTP transaction. Only one TRANSID parameter may be used in a single MAIL command. Care must be used in constructing TRANSID values to simultaneously insure both uniqueness and the ability to reidentify interrupted transactions.

The TRANSID is structured to ensure globally uniqueness without any additional registry. The transid-domain part should be a valid domain name that uniquely identifies the SMTP client. Note that this is usually the same as the domain name given in conjunction with the EHLO command, but not always. The EHLO domain name identifies the specific host the SMTP connection originated from, whereas the transid-domain may refer to a group of hosts that collectively host a multi-homed SMTP client. The transid-local part should be an identifier that distinguishes this SMTP transaction from any other originating from this SMTP client.

Despite the structured nature of the TRANSID the server should treat the value as an opaque, case-sensitive string.

Note that the contents of the RFC822 message-id header typically are NOT appropriate for use as the TRANSID parameter value, since such identifiers may be associated with multiple copies of the same message -- e.g., as it is split during transmission down different network paths -- and hence with multiple distinct SMTP transactions.

A server which supports the checkpointing extension will then retain the transaction identifer as well as the most recent state of the transaction in non-volatile storage. This information should deleted only when the transaction is known to be complete from the client's perspective. Completion is assured only when the client either explicitly aborts the transaction, starts a new transaction, or requests that the connection be closed with a QUIT command.

In the event of an interruption prior to completing a transaction this preserved state will remain for some period of time defined by the operational policies of the server administrator. It is recommended that transaction state information be preserved for at least 48 hours, although no specific time is required.

When a client detects that a transaction has been interrupted, it then must wait for some period before reconnecting. This period must be long enough for server connections to time out and for the transaction state associated with such connections to be released for use by a new connection. The Internet Host Requirements [5] also impose restriction on how quickly reconnection attempts can be made (section 5.3.1.1).

Once the necessary period has elapsed the client first checks the DNS as described in [6] and determine the set of acceptable IP addresses the message can be transferred to. If the IP address used to connect to the original server is still on this list it should be tried first, since this server is most likely to be capable of restarting the transaction. If this connection attempt fails the client must then proceed as described in [6] to try all the remaining IP addresses and restart the transaction there. If the attempt to restart fails on one of the other servers the client is required to retransmit the transaction in its entirety at that point. Waiting for a server with an interrupted transaction state to come back online is not acceptable.

Note: Multi-homed SMTP servers do exist, which means that it is entirely possible for a transaction to restart on a different server host.

Once the connection is made the client issues the same MAIL command with exactly the same transaction identifier. If the transaction was interrupted during or at the end of the transfer of actual message

data, the server first reestablishes its context to a point close as possible to the point of interruption and then responds with the status message:

355 octet-offset is the transaction offset

The actual status text can vary. However the octet-offset field is required to be the first thing on the first line of the reply, it must be separated from any following text by linear whitespace, and it is structured as follows:

octet-offset ::= 1*DIGIT

The octet-offset represents an offset, counting from zero, to the particular octet in the actual message data the server expects to see next. (This is also a count of how many octets the server has received and stored successfully.) This offset does NOT account for envelope data, i.e., MAIL FROM and RCPT TO commands. A value of 0 would indicate that the client needs to start sending the message from the beginning, a value of 1 would indicate that the client should skip one octet, and so on.

The SMTP canonical format for messages is used when this offset is computed. Any octets added by any SMTP data-stuffing algorithm do not count as part of this offset. In the case of data transferred with the DATA command the offset must also correspond to the beginning of a line.

Once this context is reestablished the client issues another data transfer command (e.g., DATA) and sends the remaining message data. Once this data is terminated the transaction completes in the normal fashion and the server deletes the transaction context from non-volatile storage.

Note that the semantics of the octet-offset immediately suggest a particularly simple implementation strategy, where the client retransmits the message data as it normally would but suppresses output of the first octet-offset octets of material. The semantics used here are intentionally designed to make such implementation possible, but care must be taken to insure that such an implementation strategy does not impose a significant performance penalty on the client.

5. Usage Example

The following dialogue illustrates the use of the checkpointing service extension:

```
S: <wait for connection on TCP port 25>
C: <open connection to server>
S: 220 dbc.mtview.ca.us SMTP service ready
C: EHLO ymir.claremont.edu
S: 250-dbc.mtview.ca.us says hello
S: 250 CHECKPOINT
C: MAIL FROM:<ned@ymir.claremont.edu> TRANSID=<12345@claremont.edu>
S: 250 <ned@ymir.claremont.edu>... Sender and TRANSID ok
C: RCPT TO:<mrose@dbc.mtview.ca.us>
S: 250 <mrose@dbc.mtview.ca.us>... Recipient ok
C: DATA
S: 354 Send checkpointed message, ending in CRLF.CRLF
<some amount of message data transmitted>
<session is interrupted and TCP connection is broken>

Some time later a new connection is established:
S: <wait for connection on TCP port 25>
C: <open connection to server>
S: 220 dbc.mtview.ca.us SMTP service ready
C: EHLO ymir.claremont.edu
S: 250-dbc.mtview.ca.us says hello
S: 250 CHECKPOINT
C: MAIL FROM:<ned@ymir.claremont.edu> TRANSID=<12345@claremont.edu>
S: 355 6135 is the transaction offset
C: DATA
S: 354 Send previously checkpointed message starting at octet 6135
C: <message data minus first 6135 octets sent>
C: .
S: 250 OK
C: QUIT
S: 221 Goodbye
```

6. Security Considerations

This RFC does not discuss security issues and is not believed to raise any security issues not already endemic in electronic mail and present in fully conforming implementations of [1].

7. 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] Rose, M., Stefferud, E., Crocker, D., Klensin, J., and N. Freed, "SMTP Service Extensions", RFC 1651, Dover Beach Consulting, Inc., Network Management Associates, Inc., Silicon Graphics, Inc., MCI, Innosoft, July 1994.

[5] Braden, R., Editor, "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, USC/Information Sciences Institute, October 1989.

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

8. Authors' Addresses

Dave Crocker
Brandenburg Consulting
675 Spruce Dr.
Sunnyvale, CA 94086 USA
USA

Phone: +1 408 246 8253
Fax: +1 408 249 6205
EMail: dcrocker@mordor.stanford.edu

Ned Freed
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