Internet-Draft M. Chalhoub

Category: Experimental A. Jaffal

Expires: January 07, 2021 July 2020

Time-Cost Stamped Mail Protocol (TCSMP)

Table of Contents

[1. Introduction 2](#_Toc45418945)

[2. The TCSMP model 2](#_Toc45418946)

[2.1. Basic Structure 2](#_Toc45418947)

[2.2 Buffers and state 2](#_Toc45418948)

[2.3 Puzzles 2](#_Toc45418949)

[3. The TCSMP procedures 3](#_Toc45418950)

[3.1. Session initiation 3](#_Toc45418951)

[3.2. Client initiation 3](#_Toc45418952)

[3.3. Mail transactions 3](#_Toc45418953)

[3.3.1. Determining sender information 3](#_Toc45418954)

[3.3.2. Determining receiver information 4](#_Toc45418955)

[3.3.3. Asking for puzzle(s) 4](#_Toc45418956)

[3.3.4 Sending out mail data 4](#_Toc45418957)

[3.3.5. Giving out puzzle(s) solution 5](#_Toc45418958)

[4. SPECIFICATIONS 5](#_Toc45418959)

[4.1. TCSMP commands 5](#_Toc45418960)

[4.1.1. Commands semantics and syntax 5](#_Toc45418961)

[4.2. TCSMP replies 8](#_Toc45418962)

[4.2.1. General syntax 8](#_Toc45418963)

[4.2.2. Reply code severities and theory 8](#_Toc45418964)

[4.2.3. Reply codes by function groups 9](#_Toc45418965)

[4.2.4 Reply code 502 10](#_Toc45418966)

[5. SCENARIO 10](#_Toc45418967)

[5.1. A Typical TCSMP Transaction Scenario 10](#_Toc45418968)

# 1. Introduction

TCSMP is a client-server protocol designed to efficiently transfer mail and allow time consuming transfers.

Although this might sound shocking, it is worth considering the time usage as an expense, i.e. sending a mail actually costs time. The aim is to stop spamming: sending mail to one recipient could take less than 1 sec, while sending mail to 50 could take 30 seconds!

TCSMP now depends on problem solving in order to do so. More specifically, it uses puzzles which grow exponentially in complexity. The server gives the client a puzzle which has to solve it so that the server accepts its mail.

# 2. The TCSMP model

## 2.1. Basic Structure

The TCSMP design can be pictured as:

+----------+ +----------+

| | | |

| | TCSMP | |

| Server - |Commands/Replies| Server- |

+------+ | TCSMP |<-------------->| TCSMP | +------+

| User |<-->| | and Mail | |<-->| User |

+------+ | | | | +------+

+----------+ +----------+

TCSMP client TCSMP server

## 2.2 Buffers and state

The TCSMP sessions are stateful, with both parties holding a shared view of the current state with care. In this document, we model this state on the server by a virtual "buffer" and a "state table" that the client may use to clear the buffer, for example, or "reset the state table," causing the information in the buffer to be discarded and the state returned to some previous state.

## 2.3 Puzzles

TCSMP 's time cost portion is done by solving a Puzzle. Such puzzles can be rectangular in TCSMP, and the way to read one is as follows: from the top left piece to the bottom right piece, from left to right and from top to bottom.

Each piece read as follows: north, east, south, west.

Here is an example. For the following Eternity puzzle:

A Z E

E P P R R L

B X G

B X G

H A A F F S

Q B I

The reading goes like this:

A P B E Z R X P E L G R B A Q H X F B A G S I F

# 3. The TCSMP procedures

This section includes explanations of the procedures used in TCSMP: session initiation, mail transaction, relaying, and opening and closing exchanges. session initiation, mail transaction, relaying, and opening and closing exchanges.

## 3.1. Session initiation

Once a client opens a connection to a server, a TCSMP session is initiated and the server responds with an opening message of 220, acknowledging the client. Additional pieces of information such as identity, software name / version to the message can be included in the server implementations.

A server that is unwilling to recognize the client may reply with a 554 response, providing reasons for doing so in the additional text. The server will close the link in such a situation.

## 3.2. Client initiation

When the 220 welcoming message has been received by the client, the client send registration request REG command, then the client normally sends the TELO command to the server, indicating the client’s identity. This specific command returns the version of the used TCSMP protocol and the options that were supported.

## 3.3. Mail transactions

Sending mails involves several steps:

* Providing sender information through the FROM command
* Providing receivers information through RCPT commands
* Request puzzle(s) via APZL command
* Send mail data via the MAIL command
* Solving puzzles via the PKEY command

These steps are to be taken in order. If a command is issue when it shouldn't, server should reply with a 503 "Bad sequence of commands".

### 3.3.1. Determining sender information

FROM «reverse-path» «CRLF»

This command informs the TCSMP server that they are beginning a new mail transaction. The reverse path between "«" and "»" must be given and contains only the source mailbox. The server returns a "250 OK" reply, if accepted. If not, the server will state the reason for non-acceptance to the client, and whether it is temporary and permanent. The Specifications section includes potential return codes.

### 3.3.2. Determining receiver information

RCPT «forward-path1» «CRLF»

This command specifies the mail to be sent to the specified forward-paths, which defines target mailboxes, to the TCSMP server. The server returns a "250 OK" reply, if accepted.

If the receiver is not known to be a deliverable address, the TCSMP server will return a 550 response, usually with a string such as "no such user-" and the name of the mailbox (other circumstances and reply codes are possible).

The Requirements section includes potential return codes.

### 3.3.3. Asking for puzzle(s)

APZL «CRLF»

The client uses this to notify the server it's waiting for puzzle(s). It will then get 215 response codes if all goes well. It will get:

215-domain w,h PUZZLE «CRLF»

Domain is the domain that asks for a solution to the given puzzle. "W" and "h" are 2 numbers, which indicate the height and width of the puzzle matrix. PUZZLE is a puzzle to solve.

### 3.3.4 Sending out mail data

MAIL «CRLF»

This command is used to mark the start of data transfer. Upon receipt, the server returns an intermediate code of 354 indicating that it is now waiting for the data itself, which will contain all lines except the end of data line.

The end of the data is signaled by a line with only a dot ".". Parsers should search for « CRLF». «CRLF » which will indicate the completion of the data transmission when read. The server will reply with a "250 OK" message after reading the end of the data line.

A transparency procedure is used to prevent user text from being interfered with by the end of line statement (But I didn’t implement transparency in this version).

For each APZL relayed the server should cache mail data as long as the PKEY commands have not been received. The server should wait PKEY commands before deleting the mail data, unless connection is broken.

### 3.3.5. Giving out puzzle(s) solution

PKEY domain h,l ABCDEFG... «CRLF»

This command serves to provide a puzzle solution for a given domain to the server.

"h" and "l" are 2 numbers indicating the height and width of the matrix representing the puzzle.

"ABCDEF ..." represents the puzzle, with 4 characters describing each part, parts being read from top left to bottom right.

If the PKEY command is issued and the server does not have the mail data to be sent, it will return via a 517 message, indicating that it is waiting for the data first because it has not. The client will be obliged to issue the PKEY command again after sending the data.

Two cases can happen here:

* solution was found correct: server responds with a 216 response code
* solution was found incorrect: server responds with a 516 response code

# 4. SPECIFICATIONS

## 4.1. TCSMP commands

### 4.1.1. Commands semantics and syntax

The TCSMP commands define the user requesting mail transfer or the original mail system function. TCSMP commands are strings of characters terminated with a « CRLF ». The commands themselves are four alphanumeric characters that are terminated by a "SPACE" if parameters follow or "CRLF," otherwise they must always be specified in uppercase. The TCSMP commands do not exceed 1024 bytes in size, including the sequence « CRLF ». TCSMP receivers are encouraged to tolerate white space trailing before terminating the « CRLF ». These are now discussed below for each TCSMP command.

#### 4.1.1.1 REG command

This command is used to Register a new user in the TSCMP server.

Syntax: REG USERNAME PASSWORD «CRLF»

#### 4.1.1.1. TELO command

This command is used to identify the TCSMP client to the TCSMP server. The argument field should contain the fully-qualified domain name of the TCSMP client if one is available. A client TCSMP should always start a session by issuing this TELO command. If the TCSMP server supports the TCSMP service extensions it will give a successful response, a failure response or an error response.

Syntax: TELO [«SPACE» FQDN] «CRLF»

A successful TCSMP server response would be a multi-line response. The first one should be a "200 «SPACE» OK «CRLF»" followed by a response list corresponding to the server-supported protocols and additional features. Such an answer should follow the syntax "200 « SPACE » KEYWORD « CRLF »". For example, if a TCSMP server implements this RFC, it must return the following sequence when it receives a TELO command.

200-TCSMPV1

200 OK

Like commands, keywords must always be specified in uppercase.

#### 4.1.1.2. FROM command

This command is used to identify the issuer of the mail through his reverse-path. Reverse-path is the identifier of the sender and it must be given between "«" and "»".

There must be just only one.

Syntax: FROM «SPACE» «X@BINIOU» «CRLF»

The server should analyze the reverse-path when it receives this command and should respond to the client.

#### 4.1.1.3. RCPT command

This command is used to identify an individual receiver of the message data. The argument field contains a forward-path between "«" and "»". When this command reaches its ultimate destination, it is inserted into the destination message box by the TCSMP server in accordance with the name specified, ending server response replies with a 200 OK message if forward-path address exists and with an error otherwise.

Syntax: RCPT «Y@POUET» «CRLF»

#### 4.1.1.4. RSET command

This command takes up no parameters. It specifies that the current transaction to TCSMP will be aborted. Any sender, receivers, puzzle and mail data stored MUST be discarded by servers. The receiver MUST send a response to a successful RSET command by "250 OK". A command of RSET may be issued at any time during the transaction.

A TCSMP server MUST NOT close the connection as the result of receiving a RSET, that is the purpose of the QUIT command.

Since TELO implies some additional processing by the server, a RSET command will normally be more efficient than reissuing a TELO one even if the end result is the same.

Syntax: RSET «CRLF»

#### 4.1.1.5. APZL command

This command is used to request the puzzle(s) to be resolved, so that a message can be sent to the different recipients. This command is issued just before the MAIL command is sent out. Each relay MUST generate a new APZL command for each domain found in its receiver buffer upon receiving this command.

Syntax: APZL

When this command reaches its ultimate destination, TCSMP server in the response data with the puzzle. It will look like reply from ending server:

215 domain w,h ABCDEFGH

The domain corresponds to the domain identifying the end server, w and h correspond to the width and height of the puzzle, the puzzle data followed by these two numbers.

#### 4.1.1.6. PKEY command

This command is issued by the client to send the solved puzzle. Before sending this command, a MAIL command SHOULD have been issued. The TCSMP server MUST respond with an error code 517 if it is not the case. When a client has solved the puzzle, this command sends the solution immediately.

Syntax: PKEY domain w,h PUZZLE-SOL

Command parameters are the same as the reply given when the APZL command is received by server. The last argument corresponds to the puzzle solution, and it must be the unique parameter that differs from the original one found in the APZL reply.

This command is relayed by TCSMP relay to the well ending server which replies with a "216 solving OK" in case the solved puzzle is good and with a "516 solving ERROR" otherwise.

#### 4.1.1.7. MAIL command

This command is issued without parameters by the TCSMP client and to send the message data to the TCSMP receiver.

The TCSMP receiver normally responds to MAIL with a 354 response, then treats the command-following lines as sender mail data. This command causes attachment of mail data to the mail data buffer. The mail data may contain any of the 128 ASCII character codes.

The message data is terminated by a line containing only a period, that is, the character sequence "«CRLF».«CRLF»". This is the end of message data indication. Note that the first «CRLF» of this terminating sequence is also the «CRLF» that ends the final line of the data (message text) or, if there was no data, ends the MAIL command itself. An extra «CRLF» MUST NOT be added, as that would cause an empty line to be added to the message.

When receiver receives the ending «CRLF».«CRLF» which means that the message data is terminated, it MUST start the processing of the message data. If this processing is successful, the receiver MUST send an OK reply. If the processing fails the receiver MUST send a failure reply.

When the TCSMP server accepts a message either for relaying or for final delivery, it inserts a trace record (also referred to interchangeably as a "time stamp line" or "Received" line) at the top of the mail data. This trace record indicates the identity of the host that sent the message, the identity of the host that received the message (and is inserting this time stamp), and the date and time the message was received.

Syntax: MAIL «CRLF»

#### 4.1.1.8. QUIT command

This command specified that the receiver MUST send an OK response and then close the message. Receiver or sender MUST NOT intentionally close the communication without receiving or sending a QUIT command, even when an error occurred. Furthermore, the issuer must wait for QUIT reply before closing the connection.

The QUIT command may be issued at any time.

Syntax: QUIT «CRLF»

## 4.2. TCSMP replies

### 4.2.1. General syntax

Replies to TCSMP commands serve to ensure the synchronization of requests and actions issued by sender or receiver and to guarantee that the TCSMP client always know the state of the server. Every command MUST generate at least one reply.

An TCSMP reply consists of a three digit number transmitted as numeric characters at the beginning and followed by some text. The number is for use by automate to determine what state to enter next, the text, in most case, is for the human user. The three digits contain enough encoded information that the TCSMP client don't need to parse text to take the appropriate action. In the general case, the text may be receiver dependent and context dependent, so there are likely to be varying texts for each reply code. Formally a reply is defined to be the sequence:

xyz «SPACE» one-text-line «CRLF»

Only the TELO, APZL and HELP commands are expected to result in multi-line replies. In multi-line replies, the last reply has the normal form (xyz «SPACE» one-text-line «CRLF» whereas other replies (before the last reply) are defined by replacing the «SPACE» after the digits by a «HIPHEN» ("-") for the first replies. For example, here is a sample multi-line reply issued when server receives TELO command.

200-TCSMPV1

200 OK

### 4.2.2. Reply code severities and theory

Each digit of the server reply have a special significance. The first digit denotes whether the response is good, bad or incomplete. A very simple TCSMP client, or one that receives an unexpected reply, will be able to determine its appropriate next action by examining the first digit of the reply. A sophisticated TCSMP client may examine the second digit to know what kind of error occurred. The third digit and any supplemental information that may be present is reserved for the finest gradation of information.

There are four values for the first digit of the reply code:

2yz   Positive Completion reply

The requested action has been successfully completed. A new request may be initiated.

3yz Positive Intermediate reply

The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The SMTP client should send another command specifying this information. This reply is used in command sequence groups (i.e., in MAIL).

4yz Transient Negative Completion reply

The command was not accepted, and the requested action did not occur. However, the error condition is temporary and the action may be requested again. The sender should return to the beginning of the command sequence (if any).

5yz Permanent Negative Completion reply

The command was not accepted and the requested action did not occur. The TCSMP client is discouraged from repeating the exact request (in the same sequence).

The second digit encodes responses in specific categories:

x0z Syntax: These replies refer to syntax errors, syntactically correct commands that do not fit any functional category, and unimplemented or superfluous commands.

x1z Information: These are replies to requests for information, such as status or help.

x2z Connections: These are replies referring to the transmission channel.

x3z Unspecified.

x4z Unspecified.

x5z Message system: These replies indicate the status of the receiver TCSMP system vis-a-vis the requested transfer or other mail system action.

The third digit gives a finer gradation of meaning in each category specified by the second digit.

The reply text may be longer than a single line.

### 4.2.3. Reply codes by function groups

500 Syntax error, command unrecognized. This may include errors such as command line too long.

501 Syntax error in parameters or arguments

502 Command not implemented (see section 4.2.2)

503 Bad sequence of commands

504 Command parameter not implemented

515 Puzzle generation has failed

516 Puzzle solution in incorrect

517 No message has been recorder

211 System status, or system help reply

214 Help message (Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user)

215 Puzzle reply

216 Puzzle solution is correct

220 «domain» Service ready

221 «domain» Service closing transmission channel

421 «domain» Service not available, closing transmission channel (This may be a reply to any command if the service knows it must shut down)

250 Requested mail action okay, completed

251 User not local; will forward to «forward-path»

450 Requested message action not taken: messagebox unavailable (e.g., messagebox busy)

550 Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or command rejected for policy reasons)

451 Requested action aborted: error in processing

551 User not local

452 Requested action not taken: insufficient system storage

552 Requested mail action aborted: exceeded storage allocation

553 Requested action not taken: mailbox name not allowed

354 Start data input; end with «CRLF».«CRLF»

554 Transaction failed

### 4.2.4 Reply code 502

Reply code 502 SHOULD be used when the command is actually

recognized by the TCSMP server, but not implemented.  If the

command is not recognized, code 500 SHOULD be returned.

# 5. SCENARIO

## 5.1. A Typical TCSMP Transaction Scenario

This TCSMP example shows mail sent by Albert at host home.com to Bernard at host paris.com. Here we assume that host home.com contacts host paris.com directly. The mail is accepted for Bernard.

S: 220 paris.com Time Control Stamped Mail Protocol

C: TELO paris.com

S: 250-paris.com greets home.com

S: 250 TCSMPv1

C: FROM «albert@home.com»

S: 250 OK

C: RCPT «bernard@paris.com»

S: 250 OK

C: APZL

S: 215 paris.com 2,2 DCAIHEECBGDFGBFD

C: MAIL

S: 354 Start mail input; end with «CRLF».«CRLF»

C: Date: Fri, 10 April 2009 02:15:29 +0100

C: From: Albert A. «albert@home.com»

C: To: Bernard Guilé «bernard@paris.com»

C: Subject: Need of Camembert

C: Hi everybody,

C: I'm going to France next week, would you be

C: interested in some creamy camembert?

C: Let me know! Albert.

C: .

S: 250 OK

C: PKEY paris.com 2,2 ABCDEFGBCDEFGHID

S: 216 your mail has been kept!

C: QUIT

S: 221 paris.com See you next time!