Network Working Group
Request for Comments: 1861
Obsoletes: 1645
Category: Informational

A. Gwinn
Southern Methodist University

October 1995

Simple Network Paging Protocol - Version 3 - Two-Way Enhanced

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This RFC suggests a simple way for delivering wireless messages, both one and two-way, to appropriate receiving devices. In its simplest form, SNPP provides a simple way to implement a "shim" between the Internet and a TAP/IXO paging terminal. In its level 3 form, it provides an easy-to-use (and build) method for communicating and receiving end-to-end acknowledgments and replies from two-way messaging devices (such as ReFLEX units).

Gateways supporting this protocol, as well as SMTP, have been in use for well over a year at several commercial paging companies, and private businesses. Client software supporting this protocol has become widespread, and is being integrated into many of the new paging and messaging products being built. In addition to commercial software, email filters and SNPP client software for Unix and Windows (WikiPage) are available at no cost. Please contact the author for more information.

Earlier versions of this specification were reviewed by IESG members and the "822 Extensions" Working Group. They preferred an alternate strategy, as discussed under "Relationship to Other IETF Work", below.

1. Introduction

With all due apologies to the Glenayre engineers (who take offense at the term "nerd") beepers are as much a part of computer nerdom as X-terminals--perhaps, unfortunately, more. The intent of Simple Network Paging Protocol is to provide a standard whereby pages can be delivered to individual paging terminals. The most obvious benefit is the elimination of the need for modems and phone lines to produce alphanumeric pages, and the added ease of delivery of pages to terminals in other cities or countries. The benefits of the Internet

become even more realized when growing towards acknowledgment-based messaging such as ReFLEX paging--where it may be impossible to accurately predict costs associated with telco services such as 1-800 numbers.

2. System Philosophy

Radio paging is somewhat taken for granted, because of the wide availability and wide use of paging products. However, the actual delivery of the page, and the process used (especially in wider area paging) is somewhat complicated. When a user initiates a page, by dialing a number on a telephone, or entering an alphanumeric page through some input device, the page must ultimately be delivered to some paging terminal, somewhere. In most cases, this delivery is made using TAP (Telocator Alphanumeric input Protocol, also known as IXO). This protocol can be a somewhat convoluted, and complicated protocol using older style ASCII control characters and a non-standard checksumming routine to assist in validating the data.

Even though TAP is widely used throughout the industry, there are plans on the table to move to a more flexible "standard" protocol referred to as TME (Telocator Message Entry Protocol). The level two enhancements to SNPP (as described below) are intended for use with this forthcoming standard.

The recently-added level three enhancements have been engineered for use, specifically, with acknowledgment-based paging. With the recent advances in wireless technology, two-way paging is fast approaching reality--therefore creating a need for a workable end-to-end acknowledged protocol. Two-way messaging, however, opens up several new areas of unpredictability. The most pronounced is the subscriber response time. Although deliveries from host to subscriber, and subsequent receipt-acknowledgments happen in a rather predictable manner, it is impossible to know when the subscriber will physically pull the unit out, read the message and respond to it. Therefore, it could well be cost prohibitive to conduct such transactions online using a phone line as medium--especially an 800-number. This makes the Internet an extremely attractive alternative because of its (generally) usage insensitive nature.

However, acknowledging the complexity of task, and flexibility of the current protocols (or the lack thereof), the final user function is quite simple: to deliver a page from point-of-origin to someone’s beeper. That is the simple, real-time function that the base protocol attempts to address.

3. Why not just use Email and SMTP for paging?

Email, while quite reliable, is not always timely. A good example of this is deferred messaging when a gateway is down. Suppose Mary Ghoti (fish@hugecompany.org) sends a message to Zaphod Beeblebrox's beeper (5551212@pager.pagingcompany.com). Hugecompany's gateway to the Internet is down causing Mary's message to be deferred. Mary, however, is not notified of this delay because her message has not actually failed to reach its destination. Three hours later, the link is restored, and (as soon as sendmail wakes up) the message is sent. Obviously, if Mary's page concerned a meeting that was supposed to happen 2 hours ago, there will be some minor administrative details to work out between Mary and Zaphod!

On the other hand, if Mary had used her SNPP client (or simply telnetted to the SNPP gateway), she would have immediately discovered the network problem. She would have decided to invoke plan "B" and call Zaphod's pager on the telephone, ringing him that way.

The obvious difference here is not page delivery, but the immediate notification of a problem that affects your message. Standard email and SMTP, while quite reliable in most cases, cannot be positively guaranteed between all nodes at all times, making it less desirable for emergency or urgent paging. This inability to guarantee delivery could, whether rightly or wrongly, place the service provider in an uncomfortable position with a client who has just received his or her emergency page, six hours too late.

Another advantage of using a separate protocol for paging delivery is that it gives the sender absolute flexibility over what is sent to the pager. For instance, in the paging arena, where messages are sent to alphanumeric pagers, it is less desirable to send the recipient general header lines from a standard SMTP message. Much of the information is useless, possibly redundant, and a waste of precious RF bandwidth.

Therefore, when implementing an SMTP gateway, the service provider should elect to parse out needed information (such as the sender, and possibly subject) such to maximize the utility of the transmission. Parsing generally means less control over content and format by the message originator. SNPP provides a clean, effective way to send a message, as written, to the recipient's pager.

The other consideration is the relative simplicity of the SNPP protocol for manual telnet sessions versus someone trying to manually hack a mail message into a gateway.

4. The SNPP Protocol

The SNPP protocol is a sequence of commands and replies, and is based on the philosophy of many other Internet protocols currently in use. SNPP has several input commands (the first 4 characters of each are significant) that solicit various server responses falling into the following categories:

2xx - Successful, continue
3xx - Begin DATA input (see "DATA" command)
4xx - Failed with connection terminated
5xx - Failed, but continue session

SNPP version 3 (two-way) adds the following categories:

7xx - UNsuccessful two-way specific transaction, but continue session
8xx - Successful two-way specific transaction, continue
9xx - Successful QUEUED two-way transaction, continue

The first character of every server response code is a digit indicating the category of response. The text portion of the response following the code may be altered to suit individual applications.

The session interaction, especially at SNPP level one, is actually quite simple (hence the name). The client initiates the connection with the listening server. Upon opening the connection, the server issues a "220" level message (indicating the willingness of the server to accept SNPP commands). The client passes pager ID information, and a message, then issues a "SEND" command. The server then feeds the information to the paging terminal, gathers a response, and reports the success or failure to the client.

4.1 Examples of "simple" SNPP Transactions

The following illustrate examples of client-server communication using SNPP.

4.1.1 A Typical Level One Connection

```
            Client                          Server

    Open Connection              -->
                                 <--  220 SNPP Gateway Ready
    PAGE 5551212                 -->
                                 <--  250 Pager ID Accepted
    MESS Your network is hosed   -->
                                 <--  250 Message OK
    SEND                         -->
                                 <--  250 Message Sent OK
    QUIT                         -->
                                 <--  221 OK, Goodbye
```

4.1.2 A Typical Level Two, Multiple Transaction

The following example illustrates a single message sent to two pagers. Using this level protocol, pager-specific options may be selected for each receiver by specifying the option prior to issuing the "PAGEr" command. In this example, an alternate coverage area is selected for the first pager, while delayed messaging is specified for the second.

```
            Client                          Server

    Open Connection              -->
                                 <--  220 SNPP Server Ready
    COVE 2                       -->
                                 <--  250 Alternate Area Selected
    PAGE 5551212 FOOBAR          -->
                                 <--  250 Pager ID Accepted
    HOLD 9401152300 -0600        -->
                                 <--  250 Delayed Message OK
    PAGE 5552323 XYZZY           -->
                                 <--  250 Pager ID Accepted
    SUBJ Seattle Meeting         -->
                                 <--  250 Message Subject OK
    DATA                         -->
                                 <--  354 Begin Input, End With '.'
    Please meet me tomorrow at   -->
    the Seattle office           -->
                                 <--  250 DATA Accepted
    SEND                         -->
                                 <--  250 Message Sent OK
    QUIT                         -->
                                 <--  221 OK, Goodbye
```

4.1.3 A Typical Level Three (two-way) Transaction

Level three transactions are inherently single-unit oriented because of the one-to-one issues surrounding responses. Each transaction begins with the "2WAY" command and terminates with a "SEND" command.

```
      Client                          Server

Open Connection               -->
                              <--  220 SNPP (V3) Gateway Ready
2WAY                          -->
                              <--  250 Two-Way Mode Enabled
NOQUEUE                       -->
                              <--  250 Msg will either be Sent or Rejected
PAGER SHIRLEY                 -->
                              <--  850 Unit online; Don't call me Shirley!
ACKRead 1                     -->
                              <--  250 Read Acknowledgment Requested
DATA                          -->
                              <--  354 Begin Input, End With '.'
Little Bo Binary has lost     -->
her Sparcstation and doesn't  -->
know where to find it. Have   -->
you seen it recently?         -->
                              <--  250 DATA Accepted
RTYPE MULTICHOICE             -->
                              <--  250 Multichoice Responses Enabled
MCRESP 01 In the West Pasture -->
                              <--  250 MCR Code Accepted
MCRESP 02 GoldiFLOCKs has it  -->
                              <--  250 MCR Code Accepted
MCRESP 03 Haven't a clue      -->
                              <--  250 MCR Code Accepted
MCRESP 04 Haven't a life      -->
                              <--  250 MCR Code Accepted
MCRESP 05 Oh, GO AWAY!        -->
                              <--  250 MCR Code Accepted
SEND                          -->
                              <--  860 00321 1234 Message Delivered
QUIT                          -->
                              <--  221 OK, Goodbye
```

4.2 General Response Code Theory

Before discussing specific SNPP transactions, it may be helpful to discuss some of the response codes. As mentioned previously, every response from the SNPP server to the client contains a 3 digit code that categorizes the response. Several of these codes fall into the

"general" category, and may occur more frequently throughout a given SNPP transaction. There are some lesser used (somewhat transaction specific) responses that will be discussed in conjunction with the format of a specific command.

4.2.1 Code 214 - Multi-line "help/info" message

This code prefixes a line of response information (such as in response to the HELP command). It should be terminated with a "250 OK" message. This code is used when the response will take more than one line to display.

4.2.2 Code 218 - Single-line "help/info" message

This code prefixes a single line of response information (such as the request for a single database entry). Unlike the 214 series, it has no "250" series terminator.

4.2.3 Code 250 - Successful Transaction

This code is a general positive acknowledgment from the server indicating that a command was successfully processed. Additionally, code 250 can appear at the end of the response to a HELP command (214 series commands--discussed below).

4.2.4 Code 421 - Fatal Error, Connection Terminated

This code is displayed just prior to the SNPP server terminating a connection with a client for errors. Such a connection termination may occur at any time and for any reason (administrative or technical).

4.2.5 Code 500 - Command Not Implemented

This code is a "fail but continue code" that appears when an illegal command is entered.

4.2.6 Code 503 - Duplicate Command Entry; Already Entered That

This code indicates that the specified information has already been entered. This code would appear, for instance, if the client attempted to enter a MESSage command after specifying a "DATA" sequence.

4.2.7 Codes 550 and 554 - Transaction Failed, but Continue

These codes indicate a failed command, but the session is allowed to continue. A 550 code should be used to indicate a more

"administrative" failure (such as an invalid pager ID, or illegal parameter), while a 554 series indicates a more technical reason (such as a gateway down or equipment failure). In addition to the specified failure codes, additional 550 and 554 failures may be specified as necessary to allow for greater flexibility.

4.2.8 Code 552 - Maximum Entries Exceeded

This code is in response to the entry of the "n+1" item when the server only permits "n" items in a category. As an example, the client would expect to see this message when trying to enter the 6th PAGEr command when the terminal only supported 5.

4.3 Level 1 Commands

Level one commands are designed as a minimum implementation of the protocol. This collection of commands may be used with either TAP/IXO or TME for message delivery to the paging terminal.

4.3.1 PAGEr <Pager ID>

The PAGEr command submits a pager ID (PID) number, for inclusion in the next messaging transaction. The PID used must reside in, and be validated by the paging terminal. Limited validation may optionally be done on the server (such as all numeric, and ID length), or validation can be left up to the terminal at the time the page is sent.

When implementing SNPP, the user may elect to support multiple recipients per message sent. However, be wary that validation-prior-to-sending is not possible with TAP/IXO (and is not an official option of the current TME specification). What this means is that in order to validate a PID, one must generate a message to the pager. The terminal responds favorably or negatively. When reporting failure of a single PID in a sequence, delineating and reporting the failure in a "standard format" may prove to be a challenge.

Possible responses from the SNPP server, with suggested text, in response to a PAGEr command are:

```
 250 Pager ID Accepted
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 550 Error, Invalid Pager ID
 554 Error, failed (technical reason)
```

Both level 2 and level 3 enhancements affect the PAGEr command. Please refer to the appropriate section(s) for details.

4.3.2 MESSage <Alpha or Numeric Message>

The MESSage command specifies a single-line message, into the gateway. Limited validation of the message may be done on the SNPP server (such as length), but type-of-message validation should be done by the paging terminal. Duplicating the MESSage command before SENDing the message should produce an "503 ERROR, Message Already Entered" message, and allow the user to continue.

Possible responses from the SNPP server, with suggested text, in response to a MESSage command are:

250 Message OK
421 Too Many Errors, Goodbye (terminate connection)
421 Gateway Service Unavailable (terminate connection)
503 ERROR, Message Already Entered
550 ERROR, Invalid Message
554 Error, failed (technical reason)

4.3.3 RESEt

The RESEt command clears already entered information from the server session, resetting it to the state of a freshly opened connection. This is provided, primarily, as a means to reset accidentally entered information during a manual session.

Possible responses from the SNPP server, with suggested text, in response to a RESEt command are:

250 RESET OK
421 Too Many Errors, Goodbye (terminate connection
421 Gateway Service Unavailable (terminate connection)

4.3.4 SEND

The SEND command finalizes the current message transaction, and processes the page to the paging terminal. Prior to processing, the PAGEr and MESSage fields (or message DATA when using the level two option) should be checked for the existence of information. Should one of these required fields be missing, the server should respond "503 Error, Incomplete Information" and allow the user to continue. Assuming that the information is complete, the SNPP server should format and send the page to the paging terminal, and await a response.

Possible responses from the SNPP server, with suggested text, in response to a SEND command are:

```
 250 Message Sent Successfully
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 503 Error, Pager ID or Message Incomplete
 554 Message Failed [non-administrative reason]
```

Or, in the case of an illegal or non-existent pager ID, or some other administrative reason for rejecting the page, the server should respond:

```
 550 Failed, Illegal Pager ID (or other explanation)
```

After processing a SEND command, the server should remain online to allow the client to submit another transaction.

Level 3 enhancements to this command allow for other responses. Please refer to the appropriate section for discussion.

4.3.5 QUIT

The QUIT command terminates the current session. The server should simply respond:

```
 221 OK, Goodbye"
```

and close the connection.

4.3.6 HELP (optional)

The optional HELP command displays a screen of information about commands that are valid on the SNPP server. This is primarily to assist manual users of the gateway. Each line of the HELP screen (responses) are preceded by a code "214". At the end of the HELP sequence, a "250" series message is issued.

Possible responses from the SNPP server, with suggested text, in response to a HELP command are:

```
 214 [Help Text]  (repeated for each line of information)
 250 End of Help Information
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 500 Command Not Implemented
```

4.4 Level 2 - Minimum Extensions

This section specifies minimum enhancements to the SNPP protocol for added functionality.

4.4.1 DATA

The DATA command is an alternate form of the MESSage command, allowing for multiple line delivery of a message to the paging terminal. This command's function is similar to the DATA command implemented in SMTP (Internet STD10, RFC821). The SNPP server should only allow one DATA or MESSage command to be issued prior to a SEND.

Possible responses from the SNPP server, with suggested text, in response to a DATA command are:

```
 354 Begin Input; End with <CRLF>'.'<CRLF>
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 503 ERROR, Message Already Entered
 500 Command Not Implemented
 550 ERROR, failed (administrative reason)
 554 ERROR, failed (technical reason)
```

Upon receiving a "354" response, the client begins line input of the message to send to the pager. A single period ("."), in the first position of the line, terminates input. After input, the server may respond:

```
 250 Message OK
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 550 ERROR, Invalid Message (or administrative reason)
 554 ERROR, Failed (technical reason)
```

4.5 Level 2 - Optional Extensions

This section discusses enhancements to the SNPP protocol for more control over paging functions. These are primarily designed to mirror the added functionality built into the Telocator Message Entry (TME) protocol as specified in the TDP protocol suite. These functions may, optionally (as is being done by the author), be integrated into a paging terminal. There is no requirement to implement all of these functions. Requests for invalid functions should return a "500 Function Not Implemented" error.

It is important to note that, at the time of this publication, the TME standard is still not finalized.

4.5.1 LOGIn <loginid> [password]

This command allows for a session login ID to be specified. It is used to validate the person attempting to access the paging terminal.

If no LOGIn command is issued, "anonymous" user status is assumed.

Possible responses from the SNPP server, with suggested text, in response to a LOGIn command are:

```
 250 Login Accepted
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 421 Illegal Access Attempt
 550 Error, Invalid LoginID or Password
 554 Error, failed (technical reason)
```

4.5.2 PAGEr <PagerID> [Password/PIN]

This PAGEr command is an enhancement to the level one specification. The primary difference is the ability to specify a password or PIN for validation or feature access.

Before proceeding, it is important to understand the logical function of the PAGEr command with respect to the LEVEl, COVErage, HOLDtime, and ALERt commands (option parameters as described below). Each time a PAGEr command is issued, it should be thought of as the last step in a multiple step transaction.

When the PAGEr command is processed, the pager ID (and password) is submitted to the paging terminal with LEVEl, COVErage, HOLDtime, and ALERt. If these parameters have not been altered, then their defaults are assumed for the transaction. After the next PAGEr command has been processed, these option parameters are reset their defaults. Using this type of "option-option-option-go" scheme, it is possible to specify a different priority level for "Jeff," and an alternate coverage area for "Kathy," while sending the same message to each.

Possible responses from the SNPP server, with suggested text, in response to a PAGEr command are:

```
 250 Pager ID Accepted
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 550 Error, Invalid Pager ID or Password
 552 Max Recipients Exceeded
 554 Error, failed (technical reason)
```

4.5.3 LEVEl <ServiceLevel>

The LEVEl function is used to specify an optional alternate level of service for the next PAGEr command. Ideally, "ServiceLevel" should

be an integer between 0 and 11 inclusive. The TME protocol specifies ServiceLevel as follows:

```
 0 - Priority
 1 - Normal (default)
 2 - Five minutes
 3 - Fifteen minutes
 4 - One hour
 5 - Four hours
 6 - Twelve hours
 7 - Twenty Four hours
 8 - Carrier specific '1'
 9 - Carrier specific '2'
10 - Carrier specific '3'
11 - Carrier specific '4'
```

The choice on how to implement this feature, or to what level it should be implemented, should be optional and up to the discretion of the carrier.

Possible responses from the SNPP server, with suggested text, in response to a LEVEl command are:

```
 250 OK, Alternate Service Level Accepted
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 500 Command Not Implemented
 550 Error, Invalid Service Level Specified
 554 Error, failed (technical reason)
```

4.5.4 ALERt <AlertOverride>

The optional ALERt command may be used to override the default setting and specify whether or not to alert the subscriber upon receipt of a message. This option, like the previous command, alters the parameters submitted to the paging terminal using the PAGEr command. The TME protocol specifies AlertOverride as either 0-DoNotAlert, or 1-Alert.

Possible responses from the SNPP server, with suggested text, in response to a ALERt command are:

```
 250 OK, Alert Override Accepted
 421 Too Many Errors, Goodbye (terminate connection)
 421 Gateway Service Unavailable (terminate connection)
 500 Command Not Implemented
 550 Error, Invalid Alert Parameter
 554 Error, failed (technical reason)
```

4.5.5 COVErage <AlternateArea>

The optional COVErage command is used to override the subscriber's default coverage area, and allow for the selection of an alternate region. This option, like the previous command, alters the parameters submitted to the paging terminal using the PAGEr command. AlternateArea is a designator for one of the following:

- A subscriber-specific alternate coverage area
- A carrier-defined region available to subscribers

As an example, Mary Ghoti is a subscriber having local service in Chicago, Illinois (Mary's region '1'). Her account has been set up in such a manner as to allow Mary's pager to be paged nationwide upon demand (Mary's region '2'). Specifying "COVErage 2" prior to issuing the appropriate "PAGEr" command allows the default Chicago area to be overridden, and Mary's pager to be messaged nationally for that transaction. It is assumed that the carrier providing Mary's service will keep track of how many pages have been sent to her pager in this manner, and will bill her accordingly.

Possible responses from the SNPP server, with suggested text, in response to a COVErage command are:

250 Alternate Coverage Selected
421 Too Many Errors, Goodbye (terminate connection)
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
550 Error, Invalid Alternate Region
554 Error, failed (technical reason)

4.5.6 HOLDuntil <YYMMDDHHMMSS> [+/-GMTdifference]

The HOLDuntil command allows for the delayed delivery of a message, to a particular subscriber, until after the time specified. The time may be specified in local time (e.g. local to the paging terminal), or with an added parameter specifying offset from GMT (in other words, "-0600" specifies Eastern Standard Time). This option, like the previous command, alters the parameters submitted to the paging terminal using the PAGEr command.

Possible responses from the SNPP server, with suggested text, in response to a HOLDuntil command are:

250 Delayed Messaging Selected
421 Too Many Errors, Goodbye (terminate connection)
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented

550 Error, Invalid Delivery Date/Time
554 Error, failed (technical reason)

4.5.7 CALLerid <CallerID>

The CALLerid function is a message-oriented function (as opposed to the subscriber-oriented functions just described). This allows for the specification of the CallerIdentifier function as described in TME. This parameter is optional, and is at the discretion of the carrier as to how it should be implemented or used.

Possible responses from the SNPP server, with suggested text, in response to a CALLerid command are:

250 Caller ID Accepted
421 Too Many Errors, Goodbye (terminate connection)
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
550 Error, Invalid Caller ID
554 Error, failed (technical reason)

4.5.8 SUBJect <MessageSubject>

The SUBJect function allows is a message-oriented function that allows the sender to specify a subject for the next message to be sent. This parameter is optional and is at the discretion of the carrier as to how it should be implemented or used.

Possible responses from the SNPP server, with suggested text, in response to a SUBJect command are:

250 Message Subject Accepted
421 Too Many Errors, Goodbye (terminate connection)
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
550 Error, Invalid Subject Option
554 Error, failed (technical reason)

4.6 Level 3 - Two-Way Extensions

This section specifies enhancements to the SNPP protocol to support acknowledgment-based paging (2-way). One of the more powerful features of ReFLEX-style paging, in addition to confirmed message delivery, is the ability to "seed" a message with multiple-choice type responses. After the recipient views the message, she can reply with one of the seeded messages. In addition to the multiple-choice responses (MCR’s), the sender may elect to receive confirmation when the message is first viewed by the recipient.

4.6.1 2WAY

The 2WAY command prefaces each two-way transaction (see previous example). This places the server in the mode to receive and process a single 2-way transaction. The server returns to "non-2WAY" mode upon the completion of a SEND command or a RESEt command. In 2WAY mode, it is, however, possible to do multiple MSTAtus commands (to check responses from field message units). Possible responses are:

250 OK, Beginning 2-Way Transaction
550 Error, Standard Transaction Already Underway, use RESEt
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.2 PING <PagerID | Alias>

This command localizes (finds) the field message unit on the system and returns its location and/or status. Because of the sensitive nature of location information, the subscriber may elect to have a generic "pager located" message (ACLU mode) rather than to return her actual location. Possible responses are:

820 <Locus_Code> Unit On System, This Area
821 Unit On System, No Location Information Available (ACLU mode)
750 Unit Valid But Not Online At This Time
920 Unit Not Online, But Can Queue Message for Later Delivery
550 Can't PING; Unit NOT 2-way capable
550 Unknown or Illegal ID
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.3 EXPTag <hours>

Changes the default expiry time for a queued message delivery. If the message is not delivered in the specified number of hours, then it is deleted and the MSTAtus tag is updated to reflect the inability to deliver (code 760). Possible responses are:

250 Message Expiry Time Changed to 'nnn' Hours
550 Cannot Change Expiry Time
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.4 NOQUEUEing

Specifies that the server should not allow message queuing for this 2WAY transaction. In this mode, if a pager is not online, the client will receive a "750" series response to a PAGEr command. This command must be specified prior to a PAGEr command. Possible responses are:

250 Queuing Disabled, This Transaction
550 Can't Disable Queueing
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.5 ACKRead <0|1>

Activates or deactivates message "read" acknowledgment. When activated, instructs the field message unit to return a message when the subscriber actually views the received message. This feature is independent of the actual reply. Possible responses are:

250 Read Acknowledgment <Enabled|Disabled>
550 Cannot modify Read Acknowledgment
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.6 RTYPe <Reply_Type_Code>

Changes the type of reply expected from the field message unit that is acceptable to the client program. Initial appropriate reply type codes are:

NONE - (default) No Reply Permitted
YESNO - Seeds a simple "Yes" or "No reply
SIMREPLY - Only pre-coded replies from providers's reply base
MULTICHOICE - Allows full multiple choice replies
TEXT - Allows full text replies (generated by field unit)

Possible responses to an RTYPe command are:

250 Reply Type Accepted
550 Illegal Reply Type
503 Already Entered That
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.7 MCREsponse <2-byte_Code> Response_Text

This command is issued prior the the SEND command, and "seeds" the transaction with an acceptable multiple choice response. Each response is specific to the current message. The number of acceptable responses may be limited by the SNPP server as desired by the provider. Examples of MCREsponse(s) are:

```
 MCREsponse 1E2C Here is one response
 MCREsponse 0002 This is another response
```

Responses from the SNPP server to the client are:

```
 250 Response Added to Transaction
 502 Error! Would Duplicate Previously Entered MCResponse
 550 Invalid MCResponse Code
 550 MCResponses Not Enabled
 552 Too Many MCResponses Entered
 421 Gateway Service Unavailable (terminate connection)
 500 Command Not Implemented
 554 Error, failed (technical reason)
```

4.6.8 PAGEr

In 2WAY mode, the following enhanced responses are available:

```
 850 Two-Way Unit Online and Available; Transaction Accepted
 950 Unit NOT Online; Message Will be Queued for Later Delivery
 750 Two-Way Unit NOT Online; Transaction Denied
 550 Error, Pager Not 2WAY Capable
 550 Error, RTYPe Mode Invalid for This Unit
 503 Already Selected PAGEr
 421 Gateway Service Unavailable (terminate connection)
 554 Error, failed (technical reason)
```

4.6.9 SEND

Instructs the SNPP server to "launch" the message (plus attached response codes) to the field message unit. A successful SEND command will return, to the client, a "Message_Tag" number and a "Pass_Code" for periodic status checking. The client then uses the MSTAtus command to check the progression of the transaction. The "Message_Tag" functions as a "record locator," while the "Pass_Code" should be a randomly generated "PIN" code to authorize checking of the "Message_Tag."

Response codes to a SEND command, as well as the MSTAtus command, indicate the degree of "finality" to the transaction. Based on the

delivery process, there are four categories. Together with their response code prefixes, these are:

86x - Initial message delivered, awaiting requested action(s)
87x - Intermediate processing completed, awaiting closure
88x - Transaction concluded (final)
96x - Queued transaction

These prefixes make a multi-tiered transaction relatively simple to follow to closure. When an 88x series response code is received from the server, all requested portions of the transaction have been processed, and no further status changes will take place.

The SEND command should reply with the first tier of message processing. Following this, the status of the message in the system is checked, periodically, using the MSTAtus command.

Possible responses to a SEND command are:

860 <Message_Tag> <Pass_Code> Delivered, Awaiting Read Ack
861 <Message_Tag> <Pass_Code> Delivered, Awaiting Reply (MCR)

880 <Message_Tag> <Pass_Code> Message Delivered

960 <Message_Tag> <Pass_Code> OK, Message QUEUED for Delivery

550 Delivery Failed! Message destroyed.
421 Gateway Service Unavailable (terminate connection)
500 Command Not Implemented
554 Error, failed (technical reason)

4.6.10 MSTAtus <Message_Tag> <Pass_Code>

This is used by a client program to periodically check the status of delivery and response of a given message. The SEND command returns the "Message_Tag" and "Pass_Code" required to check the status. A "Message_Tag" may be (should be) expired by the SNPP server after an appropriate amount of time has passed. Expiration of these tags is vendor dependent, and may accelerate after the first check after final disposition of the message (such as after a client program has successfully received the field unit’s response code).

The tag record contains a "Sequence" number which is an incremental counter that rises as the record’s status changes (such as from a delivery acknowledgment to a reply). In addition, date and time of the current transaction should be kept in the following format:

YYMMDDHHMMSS+GMT (example: 950925143501+7)

Because of the tiered structure of replies, possible responses to an MSTAtus command are:

```
 860 <Sequence> <Date&Time> Delivered, Awaiting Read Confirmation
 861 <Sequence> <Date&Time> Delivered, Awaiting Reply (MCR)

 870 <Sequence> <Date&Time> Delivered, Read, Awaiting Reply (MCR)

 880 <Sequence> <Date&Time> Message Delivered (No Reply Pending)
 881 <Sequence> <Date&Time> Message Delivered and Read by Recipient
 888 <Sequence> <Date&Time> <Reply_Code> MCR Reply Received
 889 <Sequence> <Date&Time> <Full_Text_Response>

 960 <Sequence> <Date&Time> Message Queued; Awaiting Delivery

 780 <Sequence> <Date&Time> MESSAGE EXPIRED Before Delivery!

 550 Unknown or Illegal Message_Tag or Pass_Code
 421 Gateway Service Unavailable (terminate connection)
 500 Command Not Implemented
 554 Error, failed (technical reason)
```

After a closure-series (88x) command has been returned to the client, acceleration of message tag deletion may be desired to maximize use of resources on the server.

```
KTAG <Message_Tag> <Pass_Code>
```

Used to "kill" the message tag after final reading (or when no further responses are desired). This is more of a courtesy feature that allows the client to "clean up" rather than wait for the SNPP server to expire the tag.

4.7 Illegal Commands

Should the client issue an illegal command, the server may respond in one of the two following ways:

```
 421 Too Many Errors, Goodbye (terminate connection)
 500 Command Not Implemented, Try Again
```

The number of illegal commands allowed before terminating the connection should be at the discretion of the operator of the SNPP server. The only response that has not been discussed is:

```
 421 SERVER DOWN, Goodbye
```

This is used to refuse or terminate connections when the gateway is administratively down, or when there is some other technical or administrative problem with the paging terminal.

4.8 Timeouts

The SNPP server can, optionally, have an inactivity timeout implemented. At the expiration of the allotted time, the server responds "421 Timeout, Goodbye" and closes the connection.

4.9 Rigidity of Command Structure

The commands from client to server should remain constant. However, since the first character of the response indicates success or failure, the text of the server responses could be altered to suit the tastes of the operator of the SNPP server. It is suggested that the response codes mirror SMTP response codes as closely as possible.

5. Revision History

Originally, when proposed, the author employed POP2 style result/response codes. The Internet community suggested that this '+' and '-' style theory be altered to provide numeric response codes -- similar to those used in other services such as SMTP. The protocol has been altered to this specification from the first proposed draft.

Administrative errors (Illegal Pager ID, for example) have been separated from technical errors (out-of-space on disk, for example). Administrative failures are generally preceded with a 550 series response, while technical failures bear a 554 series code.

Level two enhancements to the protocol have been added in preparation for TME deployment.

Level three enhancements to the protocol have been added in preparation for acknowledgment-based messaging.

Error code "502 Command not implemented" was changed to a general "500 Command not recognized" failure result to closer follow SMTP.

6. Relationship to Other IETF Work

The strategy of this specification, and many of its details, were reviewed by an IETF Working Group and three IESG members. They concluded that an approach using the existing email infrastructure was preferable, due in large measure to the very high costs of deploying a new protocol and the advantages of using the Internet's

most widely-distributed applications protocol infrastructure. Most reviewers felt that no new protocol was needed at all because the special "deliver immediately or fail" requirements of SNPP could be accomplished by careful configuration of clients and servers. The experimental network printing protocol [4] was identified as an example of an existing infrastructure approach to an existing problem. Other reviewers believed that a case could be made for new protocol details to identify paging clients and servers to each other and negotiate details of the transactions, but that it would be sensible to handle those details as extensions to SMTP [1, 2] rather than deploying a new protocol structure.

The author, while recognizing these positions, believes that there is merit in a separate protocol to isolate details of TAP/IXO and its evolving successors from users and, indeed, from mail-based approaches that might reach systems that would act as SMTP/MIME [3] to SNPP gateways. Such systems and gateways are, indeed, undergoing design and development concurrent with this work. See the section "Why not just use Email and SMTP?" for additional discussion of the author’s view of the classical electronic email approach.

7. References

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

[2] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extensions", United Nations University, Innosoft, Dover Beach Consulting, Inc., Network Management Associates, Inc., The Branch Office, RFC 1425, February 1993.

[3] Borenstein, N., and N. Freed, "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, Bellcore, Innosoft, September 1993.

[4] Rose, M., and C. Malamud, "An Experiment in Remote Printing", RFC 1486, Dover Beach Consulting, Inc., Internet Multicasting Service, July 1993.

8. Security Considerations

Security issues are not discussed in this memo.

9. Author's Address

R. Allen Gwinn, Jr.
Associate Director, Computing Services
Business Information Center
Southern Methodist University
Dallas, TX 75275

Phone: 214/768-3186
EMail: allen@mail.cox.smu.edu or allen@radio.net