Network Working Group
Request for Comments: 1568
Category: Informational

A. Gwinn
Southern Methodist University

January 1994

# Simple Network Paging Protocol - Version 1(b)

## 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 both alphanumeric and numeric pages (one-way) to radio paging terminals. Gateways supporting this protocol, as well as SMTP, have been in use for several months in one nationwide paging firm. One other paging firm is in the process of adopting it.

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

## 1. Introduction

Beepers are as much a part of computer nerdom as X-terminals (perhaps, unfortunately, more). The intent of Simple Network Paging Protocol (SNPP) 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 to produce alphanumeric pages, and the added ease of delivery of pages to terminals in other cities or countries. Additionally, automatic page delivery should be somewhat more simplified.

## 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. One note: even though the TAP protocol allows for a password for sending simple pages, they are rarely used (especially in commercial markets), and therefore support for them has not been implemented in this version of the protocol.

Even though TAP is widely used throughout the industry, there are plans on the table to move to a more flexible "standard" protocol (the proposal for which is actually more convoluted than most Internet RFC's). However, acknowledging the complexity 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 this protocol attempts to address. Validation of the paging information is left completely up to the TAP/IXO paging terminal, making an SNPP gateway a direct "shim" between a paging terminal and the Internet.

## 3. Why not just use Email and SMTP?

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. 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 Future of SNPP

While the current form of the SNPP protocol is designed for use with TAP/IXO, it is intended to provide a porting base for use with the newer TME (TDP) protocol. In addition, future releases of SNPP will allow for multiple recipient messages with individual "envelope" options and specifications as allowed by TME. For example, the protocol should allow the user to specify delivery of an urgent message to Zaphod in Denver, while carbon-copying Mary in Des Moines at a lower priority.

## 5. The 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 six input commands (the first 4 characters of each are significant) that solicit various server responses falling into three categories: (1) successful, (2) failed-but-continue, and (3) failed-with-connection-terminated. The first character of every server response code is a digit indicating the category of response: '2xx', '5xx', and '4xx' respectfully. The text portion of the response following the code may be altered to suit individual applications.

The session interaction is actually quite simple (hence the name). The client initiates the connection with the listening server. Upon opening the connection, the server issues a greeting followed by "250 READY" (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 TAP paging terminal, gathers a response, and reports the success or failure to the client.

### 6.1 A Typical Successful Connection

```
          Client                          Server

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

## 6.2 Commands

### 6.2.1 PAGEr <Pager ID>

The PAGEr command sets the pager ID (PID) number, for the transaction, into the gateway. The PID used must reside in the TAP terminal (and there is where it should be validated). Limited validation may optionally be done on the server (such as all numeric, and ID length), or it can all be done by the TAP terminal at the time the page is sent. Duplicating the PAGEr command before SENDing the message should produce an "503 ERROR, Already Entered" message, and allow the user to continue.

In the future, a series of PAGEr commands may be specified to allow for multiple recipients of the same message. Right now, however, TAP/IXO only validates the PID at the time the message is accepted by the paging terminal. This makes "pre" validation of PID's currently difficult.

### 6.2.2 MESSage <Alpha or Numeric Message>

The MESSage command sets the numeric or alphanumeric message for the transaction, 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 TAP/IXO paging terminal. Duplicating the MESSage command before SENDing the message should produce an "503 ERROR, Already Entered" message, and allow the user to continue.

### 6.2.3 RESEt

The RESEt command clears the PAGEr and MESSage fields, and allows the client to start over. This is provided, primarily, as a means to reset accidentally entered information during a manual session. Upon a successful reset, the server should respond "250 RESET OK".

### 6.2.4 SEND

The SEND command processes the page to the TAP terminal. Prior to processing, the PAGEr and MESSage fields 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 all of the fields are filled in, the SNPP server should format and send the page to the TAP terminal, and await a response. Upon receiving a reply, the server should respond as follows:

```
 250 Page Sent          - successful delivery
 554 Failed, <reason>   - unsuccessful, and gives a 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 enter another page.

### 6.2.5 QUIT

The QUIT command terminates the current session. The server should respond "221 OK, Goodbye" and close the connection.

### 6.2.6 HELP

The HELP command (optional) 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 OK" is issued.

## 6.3 Illegal Commands

Should the client issue an illegal command, the server should respond "421 ERROR, Goodbye" and close the connection immediately. Optionally, the server may respond "502 Command Error, try again" should it be desirable to leave the connection open.

## 6.4 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.

## 6.5 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 should one desire. The following is a hunk of C code that is used currently in an SNPP gateway. The only response that has not been discussed is "421 SERVER DOWN, Goodbye" and is used when the gateway is administratively down, or when there are communication problems with the TAP/IXO paging terminal.

```
/* SNPP Client Commands */

#define PAGER        "PAGE"
#define MESSAGE      "MESS"
#define SEND         "SEND"
#define QUIT         "QUIT"
#define RESET        "RESE"
#define HELP         "HELP"

/* Responses from SNPP server to client */

#define SNPP_OK      "250 OK"
#define SNPP_RESET   "250 Reset OK"
#define SNPP_SENT    "250 Page Sent"
#define SNPP_BADPIN  "550 Failed,"
#define SNPP_NOTSENT "554 Failed,"
#define SNPP_ENTERR  "503 Error, Already Entered"
#define SNPP_ERRINC  "503 Error, Incomplete Info"
#define SNPP_OKCLOS  "221 OK, Goodbye"
#define SNPP_TIMEOUT "421 Timeout, Goodbye"
#define SNPP_ERRCLOS "421 ERROR, Goodbye"
#define SNPP_DOWN    "421 SERVER DOWN, Goodbye"
```

## 7. 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.

When a bad pager ID message (IXO/TAP administrative failure was received from the paging terminal, a 554 series (general failure) was returned. This has been changed to a 550 failure code allowing a distinction to be made.

## 8. 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 [3] 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 [4] 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.

## 9. 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, February 1993.

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

[4] 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.

## 10. Security Considerations

Security issues are not discussed in this memo.

## 11. 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@sulaco.lonestar.org