            Session Initiation Protocol (SIP) Torture Test Messages for
                        Internet Protocol Version 6 (IPv6)

Status of This Memo

Abstract

   This document provides examples of Session Initiation Protocol (SIP)
   test messages designed to exercise and "torture" the code of an
   IPv6-enabled SIP implementation.

Table of Contents

1. Overview

   This document is informational, and is *not normative* on any aspect
   of SIP.

   This document contains test messages based on the current version
   (2.0) of the Session Initiation Protocol as defined in [RFC3261].

   This document is expected to be used as a companion document to the
   more general SIP torture test document [RFC4475], which does not
   include specific tests for IPv6 network identifiers.

   This document does not attempt to catalog every way to make an
   invalid message, nor does it attempt to be comprehensive in exploring
   unusual, but valid, messages. Instead, it tries to focus on areas
   that may cause interoperability problems in IPv6 deployments.

2. Document Conventions

   This document contains many examples of SIP messages with IPv6
   network identifiers. The appendix contains an encoded binary form
   containing the bit-exact representation of all the messages and the
   script needed to decode them into separate files.

   The IPv6 addresses used in this document correspond to the 2001:
   DB8::/32 address prefix reserved for documentation [RFC3489].
   Likewise, the IPv4 addresses used in this document correspond to the
   192.0.2.0/24 address block as described in [RFC3330].

   Although SIP is a text-based protocol, some of these examples cannot
   be unambiguously rendered without additional markup due to the
   constraints placed on the formatting of RFCs. This document uses the
   markup convention established in [RFC4475] to avoid
   ambiguity and meet the Internet-Draft layout requirements. For the
   sake of completeness, the text defining this markup from Section 2.1
   of [RFC4475] is reproduced in its entirety below:

      Several of these examples contain unfolded lines longer than 72
      characters. These are captured between <allOneLine/> tags. The
      single unfolded line is reconstructed by directly concatenating
      all lines appearing between the tags (discarding any line feeds or
      carriage returns). There will be no whitespace at the end of
      lines. Any whitespace appearing at a fold-point will appear at
      the beginning of a line.

      The following represent the same string of bits:

      Header-name: first value, reallylongsecondvalue, third value

         <allOneLine>
         Header-name: first value,
          reallylongsecondvalue
         , third value
         </allOneLine>

         <allOneLine>
         Header-name: first value,
          reallylong
         second
         value,
          third value
         </allOneLine>

      Note that this is NOT SIP header-line folding, where different
      strings of bits have equivalent meaning.

3. SIP and IPv6 Network Configuration

   System-level issues like deploying a dual-stack proxy server,
   populating DNS with A and AAAA Resource Records (RRs), zero-
   configuration discovery of outbound proxies for IPv4 and IPv6
   networks, when a dual-stack proxy should Record-Route itself, and
   media issues also play a major part in the transition to IPv6. This
   document does not, however, address these issues. Instead, a
   companion document [sip-trans] provides more guidance on these
   issues.

4. Parser Torture Tests

   The test messages are organized into several sections. Some stress
   only the SIP parser and others stress both the parser and the
   application above it. Some messages are valid and some are not.
   Each example clearly calls out what makes any invalid messages
   incorrect.

   Please refer to the complete Augmented Backus-Naur Form (ABNF) in
   [RFC3261] on representing IPv6 references in SIP messages. IPv6
   references are delimited by a "[" and "]". When an IPv6 reference is
   part of a SIP Uniform Resource Identifier (URI), RFC 3261 mandates
   that the "IPv6reference" production rule be used to recognize tokens
   that comprise an IPv6 reference. More specifically, the ABNF states
   the following:

```
     SIP-URI        = "sip:" [ userinfo ] hostport
                      uri-parameters [ headers ]
     hostport       = host [ ":" port ]
     host           = hostname / IPv4address / IPv6reference
     IPv4address    = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
     IPv6reference  = "[" IPv6address "]"
     IPv6address    = hexpart [ ":" IPv4address ]
     hexpart        = hexseq / hexseq "::" [ hexseq ] / "::" [ hexseq ]
     hexseq         = hex4 *( ":" hex4)
     hex4           = 1*4HEXDIG
```

4.1. Valid SIP Message with an IPv6 Reference

   The request below is well-formatted according to the grammar in
   [RFC3261]. An IPv6 reference appears in the Request-URI (R-URI), Via
   header field, and Contact header field.

   Message Details: ipv6-good

      REGISTER sip:[2001:db8::10] SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Max-Forwards: 70
      Contact: "Caller" <sip:caller@[2001:db8::1]>
      CSeq: 98176 REGISTER
      Content-Length: 0

4.2. Invalid SIP Message with an IPv6 Reference

   The request below is not well-formatted according to the grammar in
   [RFC3261]. The IPv6 reference in the R-URI does not contain the
   mandated delimiters for an IPv6 reference ("[" and "]").

   A SIP implementation receiving this request should respond with a 400
   Bad Request error.

   Message Details: ipv6-bad

      REGISTER sip:2001:db8::10 SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Max-Forwards: 70
      Contact: "Caller" <sip:caller@[2001:db8::1]>
      CSeq: 98176 REGISTER
      Content-Length: 0

4.3. Port Ambiguous in a SIP URI

   IPv6 uses the colon to delimit octets. This may lead to ambiguity if
   the port number on which to contact a SIP server is inadvertently
   conflated with the IPv6 reference. Consider the REGISTER request
   below. The sender of the request intended to specify a port number
   (5070) to contact a server, but inadvertently, inserted the port
   number inside the closing "]" of the IPv6 reference. Unfortunately,
   since the IPv6 address in the R-URI is compressed, the intended port
   number becomes the last octet of the reference.

   From a parsing perspective, the request below is well-formed.
   However, from a semantic point of view, it will not yield the desired
   result. Implementations must ensure that when a raw IPv6 address
   appears in a SIP URI, then a port number, if required, appears
   outside the closing "]" delimiting the IPv6 reference. Raw IPv6
   addresses can occur in many header fields, including the Contact,
   Route, and Record-Route header fields. They also can appear as the
   result of the "sent-by" production rule of the Via header field.
   Implementers are urged to consult the ABNF in [RFC3261] for a
   complete list of fields where a SIP URI can appear.

   Message Details: port-ambiguous

      REGISTER sip:[2001:db8::10:5070] SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "Caller" <sip:caller@[2001:db8::1]>
      Max-Forwards: 70
      CSeq: 98176 REGISTER
      Content-Length: 0

4.4. Port Unambiguous in a SIP URI

   In contrast to the example in Section 4.3, the following REGISTER
   request leaves no ambiguity whatsoever on where the IPv6 address ends
   and the port number begins. This REGISTER request is well formatted
   per the grammar in [RFC3261].

   Message Details: port-unambiguous

      REGISTER sip:[2001:db8::10]:5070 SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "Caller" <sip:caller@[2001:db8::1]>
      Max-Forwards: 70
      CSeq: 98176 REGISTER
      Content-Length: 0

4.5. IPv6 Reference Delimiters in Via Header

   IPv6 references can also appear in Via header fields; more
   specifically in the "sent-by" production rule and the "via-received"
   production rule. In the "sent-by" production rule, the sequence of
   octets comprising the IPv6 address is defined to appear as an
   "IPv6reference" non-terminal, thereby mandating the "[" and "]"
   delimiters. However, this is not the case for the "via-received"
   non-terminal. The "via-received" production rule is defined as
   follows:

      via-received = "received" EQUAL (IPv4address / IPv6address)

   The "IPv6address" non-terminal is defined not to include the
   delimiting "[" and "]". This has led to the situation documented
   during the 18th SIP Interoperability Event [Email-SIPit]:

      Those testing IPv6 made different assumptions about enclosing
      literal v6 addresses in Vias in []. By the end of the event, most
      implementations were accepting either. Its about 50/50 on what
      gets sent.

   While it would be beneficial if the same non-terminal
   ("IPv6reference") was used for both the "sent-by" and "via-received"
   production rules, there has not been a consensus in the working group
   to that effect. Thus, the best that can be suggested is that
   implementations must follow the Robustness Principle [RFC1122] and be
   liberal in accepting a "received" parameter with or without the

   delimiting "[" and "]" tokens. When sending a request,
   implementations must not put the delimiting "[" and "]" tokens.

   The two test cases below are designed to stress this behavior. A SIP
   implementation receiving either of these messages must parse them
   successfully.

   The request below contains an IPv6 address in the Via "received"
   parameter. The IPv6 address is delimited by "[" and "]". Even
   though this is not a valid request based on a strict interpretation
   of the grammar in [RFC3261], robust implementations must nonetheless
   be able to parse the topmost Via header field and continue processing
   the request.

   Message Details: via-received-param-with-delim

      BYE sip:[2001:db8::10] SIP/2.0
      To: sip:user@example.com;tag=bd76ya
      From: sip:user@example.com;tag=81x2
      <allOneLine>
      Via: SIP/2.0/UDP [2001:db8::9:1];received=[2001:db8::9:255];
      branch=z9hG4bKas3-111
      </allOneLine>
      Call-ID: SSG9559905523997077@hlau_4100
      Max-Forwards: 70
      CSeq: 321 BYE
      Content-Length: 0

   The OPTIONS request below contains an IPv6 address in the Via
   "received" parameter without the adorning "[" and "]". This request
   is valid according to the grammar in [RFC3261].

   Message Details: via-received-param-no-delim

      OPTIONS sip:[2001:db8::10] SIP/2.0
      To: sip:user@example.com
      From: sip:user@example.com;tag=81x2
      <allOneLine>
      Via: SIP/2.0/UDP [2001:db8::9:1];received=2001:db8::9:255;
      branch=z9hG4bKas3
      </allOneLine>
      Call-ID: SSG95523997077@hlau_4100
      Max-Forwards: 70
      Contact: "Caller" <sip:caller@[2001:db8::9:1]>
      CSeq: 921 OPTIONS
      Content-Length: 0

4.6. SIP Request with IPv6 Addresses in Session Description Protocol
      (SDP) Body

   This request below is valid and well-formed according to the grammar
   in [RFC3261]. Note that the IPv6 addresses in the SDP [RFC4566] body
   do not have the delimiting "[" and "]".

   Message Details: ipv6-in-sdp

      INVITE sip:user@[2001:db8::10] SIP/2.0
      To: sip:user@[2001:db8::10]
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::20];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "Caller" <sip:caller@[2001:db8::20]>
      CSeq: 8612 INVITE
      Max-Forwards: 70
      Content-Type: application/sdp
      Content-Length: 268

      v=0
      o=assistant 971731711378798081 0 IN IP6 2001:db8::20
      s=Live video feed for today's meeting
      c=IN IP6 2001:db8::20
      t=3338481189 3370017201
      m=audio 6000 RTP/AVP 2
      a=rtpmap:2 G726-32/8000
      m=video 6024 RTP/AVP 107
      a=rtpmap:107 H263-1998/90000

4.7. Multiple IP Addresses in SIP Headers

   The request below is valid and well-formed according to the grammar
   in [RFC3261]. The Via list contains a mix of IPv4 addresses and IPv6
   references.

   Message Details: mult-ip-in-header

      BYE sip:user@host.example.net SIP/2.0
      Via: SIP/2.0/UDP [2001:db8::9:1]:6050;branch=z9hG4bKas3-111
      Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bKjhja8781hjuaij65144
      <allOneLine>
      Via: SIP/2.0/TCP [2001:db8::9:255];branch=z9hG4bK451jj;
      received=192.0.2.200
      </allOneLine>
      Call-ID: 997077@lau_4100
      Max-Forwards: 70
      CSeq: 89187 BYE
      To: sip:user@example.net;tag=9817--94
      From: sip:user@example.com;tag=81x2
      Content-Length: 0

4.8. Multiple IP Addresses in SDP

   The request below is valid and well-formed according to the grammar
   in [RFC3261]. The SDP contains multiple media lines, and each media
   line is identified by a different network connection address.

   Message Details: mult-ip-in-sdp

      INVITE sip:user@[2001:db8::10] SIP/2.0
      To: sip:user@[2001:db8::10]
      From: sip:user@example.com;tag=81x2
      Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "Caller" <sip:caller@[2001:db8::9:1]>
      Max-Forwards: 70
      CSeq: 8912 INVITE
      Content-Type: application/sdp
      Content-Length: 181

      v=0
      o=bob 280744730 28977631 IN IP4 host.example.com
      s=
      t=0 0
      m=audio 22334 RTP/AVP 0
      c=IN IP4 192.0.2.1
      m=video 6024 RTP/AVP 107
      c=IN IP6 2001:db8::1
      a=rtpmap:107 H263-1998/90000

4.9. IPv4-Mapped IPv6 Addresses

   An IPv4-mapped IPv6 address is usually represented with the last 32
   bits appearing as a dotted-decimal IPv4 address; e.g., ::ffff:
   192.0.2.1. A SIP implementation receiving a message that contains
   such a mapped address must be prepared to parse it successfully. An
   IPv4-mapped IPv6 address may appear in signaling, or in the SDP
   carried by the signaling message, or in both. If a port number is
   part of the URI represented by the IPv4-mapped IPv6 address, then it
   must appear outside the delimiting "]" (cf. Section 4.4).

   The message below is well-formed according to the grammar in
   [RFC3261]. The Via list contains two Via headers, both of which
   include an IPv4-mapped IPv6 address. An IPv4-mapped IPv6 address
   also appears in the Contact header and the SDP. The topmost Via
   header includes a port number that is appropriately delimited by "]".

   Message Details: ipv4-mapped-ipv6

      INVITE sip:user@example.com SIP/2.0
      To: sip:user@example.com
      From: sip:user@east.example.com;tag=81x2
      Via: SIP/2.0/UDP [::ffff:192.0.2.10]:19823;branch=z9hG4bKbh19
      Via: SIP/2.0/UDP [::ffff:192.0.2.2];branch=z9hG4bKas3-111
      Call-ID: SSG9559905523997077@hlau_4100
      Contact: "T. desk phone" <sip:ted@[::ffff:192.0.2.2]>
      CSeq: 612 INVITE
      Max-Forwards: 70
      Content-Type: application/sdp
      Content-Length: 236

      v=0
      o=assistant 971731711378798081 0 IN IP6 ::ffff:192.0.2.2
      s=Call me soon, please!
      c=IN IP6 ::ffff:192.0.2.2
      t=3338481189 3370017201
      m=audio 6000 RTP/AVP 2
      a=rtpmap:2 G726-32/8000
      m=video 6024 RTP/AVP 107
      a=rtpmap:107 H263-1998/90000

4.10. IPv6 Reference Bug in RFC 3261 ABNF

   It is possible to follow the IPv6reference production rule of RFC
   3261 ABNF -- the relevant portion of which is reproduced at the top
   of Section 4 -- and arrive at the following construct:

   [2001:db8:::192.0.2.1]

   Note the extra colon before the IPv4 address in the above construct.
   The correct construct, of course, is:

   [2001:db8::192.0.2.1]

   The ABNF pertaining to IPv6 references in RFC 3261 was derived from
   RFC 2373 [RFC2373], which has been obsoleted by RFC 4291 [RFC4291].
   The specific behavior of inserting an extra colon was inherited from
   RFC 2373, and has been remedied in RFC 4291. However, following the
   Robustness Principle [RFC1122], an implementation must tolerate both
   of the above constructs.

   The message below includes an extra colon in the IPv6 reference. A
   SIP implementation receiving such a message may exhibit robustness by
   successfully parsing the IPv6 reference (it can choose to ignore the
   extra colon when parsing the IPv6 reference. If the SIP
   implementation is acting in the role of a proxy, it may additionally
   serialize the message without the extra colon to aid the next
   downstream server).

   Message Details: ipv6-bug-abnf-3-colons

      OPTIONS sip:user@[2001:db8:::192.0.2.1] SIP/2.0
      To: sip:user@[2001:db8:::192.0.2.1]
      From: sip:user@example.com;tag=810x2
      Via: SIP/2.0/UDP lab1.east.example.com;branch=z9hG4bKas3-111
      Call-ID: G9559905523997077@hlau_4100
      CSeq: 689 OPTIONS
      Max-Forwards: 70
      Content-Length: 0

   The next message has the correct syntax for the IPv6 reference in the
   R-URI.

   Message Details: ipv6-correct-abnf-2-colons

      OPTIONS sip:user@[2001:db8::192.0.2.1] SIP/2.0
      To: sip:user@[2001:db8::192.0.2.1]
      From: sip:user@example.com;tag=810x2
      Via: SIP/2.0/UDP lab1.east.example.com;branch=z9hG4bKas3-111
      Call-ID: G9559905523997077@hlau_4100
      CSeq: 689 OPTIONS
      Max-Forwards: 70
      Content-Length: 0

5. Security Considerations

   This document presents examples of SIP messages with IPv6 references
   contained in the signaling headers and SDP payload. While this
   document may clarify the behavior of SIP elements processing a

   message with IPv6 references, it does not normatively change the base
   SIP [RFC3261] specification in any way. Consequently, all security
   considerations in [RFC3261] apply.

   Parsers must carefully consider edge conditions and malicious input
   as part of their design. Attacks on many Internet systems use
   crafted input to cause implementations to behave in undesirable ways.
   Many of the messages in this document are designed to stress a parser
   implementation at points traditionally used for such attacks. This
   document does not, however, attempt to be comprehensive. It contains
   some common pitfalls that the authors have discovered while parsing
   IPv6 identifiers in SIP implementations.

6. Acknowledgments

   The authors thank Jeroen van Bemmel, Dennis Bijwaard, Gonzalo
   Camarillo, Bob Gilligan, Alan Jeffrey, Larry Kollasch, Erik Nordmark,
   Kumiko Ono, Pekka Pessi, Jon Peterson, and other members of the SIP-
   related working groups for input provided during the construction of
   the document and discussion of the test cases.

   This work is being discussed on the sipping@ietf.org mailing list.

   A.B. Nataraju and A.C. Mahendran provided working group last call
   comments.

   Mohamed Boucadair and Brian Carpenter suggested new test cases for
   inclusion in the document.

7. References

7.1. Normative References

   [RFC1122]   Braden, R., Ed., "Requirements for Internet Hosts -
                Communication Layers", STD 3, RFC 1122, October 1989.

   [RFC3261]   Rosenberg, J., Schulzrinne, H., Camarillo, G.,
                Johnston, A., Peterson, J., Sparks, R., Handley, M.,
                and E. Schooler, "SIP: Session Initiation Protocol",
                RFC 3261, June 2002.

   [RFC3330]   IANA, "Special-Use IPv4 Addresses", RFC 3330, September
                  2002.

   [RFC3489]   Rosenberg, J., Weinberger, J., Huitema, C., and R.
                  Mahy, "STUN - Simple Traversal of User Datagram
                  Protocol (UDP) Through Network Address Translators
                  (NATs)", RFC 3489, March 2003.

   [RFC4475]   Sparks, R., Ed., Hawrylyshen, A., Johnston, A.,
                  Rosenberg, J., and H. Schulzrinne, "Session Initiation
                  Protocol (SIP) Torture Test Messages", RFC 4475, May
                  2006.

   [RFC4566]   Handley, M., Jacobson, V., and C. Perkins, "SDP:
                  Session Description Protocol", RFC 4566, July 2006.

7.2. Informative References

   [RFC2373]   Hinden, R. and S. Deering, "IP Version 6 Addressing
                  Architecture", RFC 2373, July 1998.

   [RFC4291]   Hinden, R. and S. Deering, "IP Version 6 Addressing
                  Architecture", RFC 4291, February 2006.

   [sip-trans]   Camarillo, G., El Malki, K., and V. Gurbani, "IPv6
                  Transition in the Session Initiation Protocol (SIP)",
                  Work in Progress, August 2007.

   [Email-SIPit] Sparks, R., "preliminary report: SIPit 18", Electronic
                  Mail archived at http://www1.ietf.org/mail-archive/web/
                  sip/current/msg14103.html, April 2006.

Appendix A. Bit-Exact Archive of Each Test Message

   The following text block is an encoded, gzip compressed TAR archive
   of files that represent each of the example messages discussed in
   Section 4.

   To recover the compressed archive file intact, the text of this
   document may be passed as input to the following Perl script (the
   output should be redirected to a file or piped to "tar -xzvf -").

```
#!/usr/bin/perl
use strict;
my $bdata = "";
use MIME::Base64;
while(<>) {
  if (/-- BEGIN MESSAGE ARCHIVE --/ .. /-- END MESSAGE ARCHIVE --/) {
        if ( m/^\s*[^\s]+\s*$/) {
            $bdata = $bdata . $_;
        }
  }
}
print decode_base64($bdata);
```

   Alternatively, the base-64 encoded block can be edited by hand to
   remove document structure lines and fed as input to any base-64
   decoding utility.

A.1. Encoded Reference Messages

   -- BEGIN MESSAGE ARCHIVE --
   H4sICPujD0cAA21zZy50YXIA7Vpbc6M2GPUzv0Ldl74UWzckIHUnbXY39XS760ncz
   HQ6mY5sFBuvDRSwN+mvrwAb303c2GQ34byAjYSEpHO+i1Rv1E4OCCnkEKorRJyl1+
   R2dk1RQ6oE4RhxRNT/CCHGa8bpu1arTaJYhKrJ6ef+3nJ+PJDhnufzD8ku+LidPB3
   qDTeYUn0sgkA6urpnx28DIggZpbvmHyFOF/NPWTL/FFFcg8fvyiZe+fy3Pt60Ou9A
   5Ab2JJLhubwX42Ak6z1/DK5b7QauQ63j21sLaO9Df7z8SERxfen5WSz6TRPdY+3GF
   fb8dY0/3rbBX7Z9p2AjS/1Tx3UEb9W9iclZNxReb9D81xpc0u5v3QGyimvj27VqIi
   K60hDtQoxGeuutqn19aRmGZUHDwMSyOOT8fDASk7+pWpvahe/Fohfb4E2nDhwZfQb
   BwPfkG/Bj8m2xdM43W/xJu7iW/9iAIQyyQdR+F/f6ez/8IkInsgHP3iu9WO88BNIG
   imIjtydi1/cakRPkTz9Irx8PbIAJ07RpE2p+U0SRq9alFwOLI06UKiLCTW6Z0EQAq
   vZAq83Aep+0qJl8MBhLEPm+9wNQ8yAi+Z3Wa+6qETcJISY1ETItQAhPGIoh0sZNMX
   FcHzC1lsFVp934+aYNsCaaYRworbAxuOSY6QQ3TFVCFZ+6jkyKY5oXV5ReVFA/wK+
   YqWmxLLNhJRzRnnvtV5jpP9O7wjldGwX6DyklSv8Z5AZEmPNE/7FBWKX/JeDq3WXr
   uvPuKlVxrEbedrqmreh6uPo/TvgXbVg2eqJubxXcTMiTN8hwpuC99Mf5Utso12/LV
   GsSzIdhQ5Sh9rJlasb/vu+fTgCK+W8s+I9pyn9OKv+vDKzwf5kg8LZSgFegADP+u5
   6uXNITtVEU/0GO5/zHkKKX2X7m8vOJ/CViP/x4jAatlnqwCGB4tfCvgvGppTnrziHE
   bMw+L25Y7pGK2D+5Ugix+upPSAXd+CGLfEQ/fRyqUk7Hr9RcR3ErdKnqr8ETUG+PJ
   KNbdIDEBAymcvSL3/1Dk/6l1l+s/wjDN/xECK/0vAb/8uST+A38pgefJOJf/IifOZ
   tCAO0R8o26e81urMBwMhclNNBhOhDtkBqJ0tXLnYq1hbBjrpoMaaDg8C2VPKlV1mn
   mmKzETc2syMyB7nMjMRFjI5EAN0HYHWI1Pat8S91HXLfooO/jVOZcr/D+RC1jEf85
   Zzn+MMv9PWc6K/yXgK/D/nh4FPtoBtNKwbzffc5fwMA8QmWjuAXb9LsAm5JRyAtWd
   pRY3QZnnR8GKwCYRdNRUThwEMHfZMCZk4YTBueNHF6q5213b4iSiIh+u3gj8MNbFu
   Ov2J/4kOsUaK8z/GLn9R4Rl9l+NYMX/ErA7/2MbkH8bSaCDcj47yP9ak0Az/k+8Ey
   rAIfynGKX8p8So+F8C9uR/UwGo+P/S+T91hT6Pl/RAhGKse77uyJE7PlIbhfxni/1
   fg6X7Pwzzav+nDHxqd1qfPl4/3/ZPHqqvBfabkrAuB0fdDrKWN4QwArNxefFCsJX/
   X9x4cEQFKOQ/Xth/I4v/GcMV/8vAPP93IPdTgncdzh7EkWWgKMH35A3ilOJEUTzJ7
   L10ehdifv5r0tdF17vTid7zR7531CigmP/Z+W/MGUvPfSUygKvzX2Vg2f6vJ/cWp3
   OLE4FLZYsFAW5ThJHoovrGEeIC8u8NC7LzuaaVG/OdG70L+j/3fJSNGf97fqgUOM4
   0AB9ZAwr5j1jOf+UFpPZfSUDF/xKwj/8H0L9if4UKFSp8Y/gPJmWg1AA6AAA=
   -- END MESSAGE ARCHIVE --

Authors' Addresses

   Vijay K. Gurbani
   Bell Laboratories, Alcatel-Lucent
   2701 Lucent Lane
   Rm 9F-546
   Lisle, IL 60532
   USA

   Phone: +1 630 224 0216
   EMail: vkg@alcatel-lucent.com

   Chris Boulton
   Ubiquity Software Corporation
   Building 3
   West Fawr Lane
   St Mellons
   Cardiff, South Wales CF3 5EA

   EMail: cboulton@ubiquitysoftware.com

   Robert J. Sparks
   Estacado Systems

   EMail: RjS@estacado.net

Intellectual Property

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at
   ietf-ipr@ietf.org.