Network Working Group
Internet-Draft
Expires: April 26, 2006

R. Sparks, Ed.
Estacado Systems
A. Hawrylyshen
Ditech Communications Corp.
A. Johnston
Tello Corporation
J. Rosenberg
Cisco Systems
H. Schulzrinne
Columbia University
October 23, 2005

Session Initiation Protocol Torture Test Messages
draft-ietf-sipping-torture-tests-08

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 26, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This informational document gives examples of Session Initiation

Protocol (SIP) test messages designed to exercise and "torture" a SIP implementation.

Table of Contents

1. Overview 4
2. Document Conventions 4
  2.1. Representing Long Lines 4
  2.2. Representing Non-printable Characters 5
  2.3. Representing Long Repeating Strings 6
3. SIP Test Messages 6
  3.1. Parser tests (syntax) 6
    3.1.1. Valid messages 6
      3.1.1.1. A short tortuous INVITE 6
      3.1.1.2. Wide range of valid characters 8
      3.1.1.3. Valid use of the % escaping mechanism 9
      3.1.1.4. Escaped nulls in URIs 10
      3.1.1.5. Use of % when it is not an escape 11
      3.1.1.6. Message with no LWS between display name and < . . 11
      3.1.1.7. Long values in header fields 12
      3.1.1.8. Extra trailing octets in a UDP datagram 14
      3.1.1.9. Semicolon separated parameters in URI user part . 15
      3.1.1.10. Varied and unknown transport types 15
      3.1.1.11. Multipart MIME message 16
      3.1.1.12. Unusual reason phrase 17
      3.1.1.13. Empty reason phrase 18
    3.1.2. Invalid messages 19
      3.1.2.1. Extraneous header field separators 19
      3.1.2.2. Content length larger than message 19
      3.1.2.3. Negative Content-Length 20
      3.1.2.4. Request scalar fields with overlarge values . . . 21
      3.1.2.5. Response scalar fields with overlarge values . . . 22
      3.1.2.6. Unterminated quoted string in display-name 22
      3.1.2.7. <> enclosing Request-URI 23
      3.1.2.8. Malformed SIP Request-URI (embedded LWS) 24
      3.1.2.9. Multiple SP separating Request-Line elements . . . 25
      3.1.2.10. SP characters at end of Request-Line 26
      3.1.2.11. Escaped headers in SIP Request-URI 27
      3.1.2.12. Invalid timezone in Date header field 27
      3.1.2.13. Failure to enclose name-addr URI in <> 28
      3.1.2.14. Spaces within addr-spec 29
      3.1.2.15. Non-token characters in display-name 29
      3.1.2.16. Unknown protocol version 30
      3.1.2.17. Start line and CSeq method mismatch 30
      3.1.2.18. Unknown Method with CSeq method mismatch 30
      3.1.2.19. Overlarge response code 31
  3.2. Transaction layer semantics 31

3.2.1. Missing transaction identifier 31
3.3. Application layer semantics 32
3.3.1. Missing Required Header Fields 32
3.3.2. Request-URI with unknown scheme 33
3.3.3. Request-URI with known but atypical scheme 33
3.3.4. Unknown URI schemes in header fields 34
3.3.5. Proxy-Require and Require 34
3.3.6. Unknown Content-Type 35
3.3.7. Unknown authorization scheme 35
3.3.8. Multiple values in single value required fields . . . 36
3.3.9. Multiple Content-Length values 37
3.3.10. 200 OK Response with broadcast Via header field value 37
3.3.11. Max-Forwards of zero 38
3.3.12. REGISTER with a contact header parameter 38
3.3.13. REGISTER with a url parameter 39
3.3.14. REGISTER with a url escaped header 39
3.3.15. Unacceptable Accept offering 40
3.4. Backward compatibility 41
3.4.1. INVITE with RFC2543 syntax 41
4. Security Considerations 42
5. IANA Considerations 42
6. Acknowledgments 42
7. Informative References 43
Appendix A. Bit-exact archive of each test message 43
A.1. Encoded Reference Messages 44
Authors' Addresses 49
Intellectual Property and Copyright Statements 50

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]. Some messages exercise SIP's use of SDP as described in [RFC3264].

These messages were developed and refined at the SIPIt interoperability test events.

The test messages are organized into several sections. Some stress only a 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.

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 have caused interoperability problems or have particularly unfavorable characteristics if they are handled improperly. This document is a seed for a test plan, not a test plan in itself.

The messages are presented in the text using a set of markup conventions to avoid ambiguity and meet Internet-Draft layout requirements. To resolve any remaining ambiguity, a bit-accurate version of each message is encapsulated in an appendix.

2. Document Conventions

This document contains many example SIP messages. Although SIP is a text-based protocol, many of these examples cannot be unambiguously rendered without additional markup due to the constraints placed on the formatting of RFCs. This document defines and uses the markup defined in this section to remove that ambiguity. This markup uses the start and end tag conventions of XML, but does not define any XML document type.

The appendix contains an encoded binary form of all the messages and the algorithm needed to decode them into files.

2.1. Representing Long Lines

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.

2.2. Representing Non-printable Characters

Several examples contain binary message bodies or header field values containing non-ascii range UTF-8 encoded characters. These are rendered here as a pair of hexadecimal digits per octet between <hex/> tags. This rendering applies even inside quoted-strings.

The following represent the same string of bits:

```
Header-name: value one
```

```
Header-name: value<hex>206F6E</hex>e
```

The following is a Subject header field containing the euro symbol:

```
Subject: <hex>E282AC</hex>
```

2.3. Representing Long Repeating Strings

Several examples contain very large data values created with repeating bit strings. Those will be rendered here using <repeat count=some_integer>value</repeat>. As with <hex> this rendering applies even inside quoted-strings.

For example, the value "abcabcabc" can be rendered as <repeat count=3>abc</repeat>. A display name of "1000000 bottles of beer" could be rendered as

```
To: "1<repeat count=6><hex>30</hex></repeat> bottles of beer"
    <sip:beer.example.com>
```

and a Max-Forwards header field with a value of one google will be rendered here as

```
Max-Forwards: 1<repeat count=100>0</repeat>
```

3. SIP Test Messages

3.1. Parser tests (syntax)

3.1.1. Valid messages

3.1.1.1. A short tortuous INVITE

This short, relatively human-readable message contains:

- o line folding all over
- o escaped characters within quotes
- o an empty subject
- o LWS between colons, semicolons, header field values, and other fields
- o both comma separated and separate listing of header field values
- o mix of short and long form for the same header field name
- o unkown Request-URI parameter
- o unknown header fields
- o unknown header field with a value that would be syntactically invalid if it were defined in terms of generic-param
- o unusual header field ordering
- o unusual header field name character case
- o unknown parameters of a known header field
- o uri parameter with no value
- o header parameter with no value
- o integer fields (Max-Forwards and CSeq) with leading zeros

All elements should treat this as a well-formed request.

The UnknownHeaderWithUnusualValue header field deserves special attention. If this header field were defined in terms of comma separated values with semicolon separated parameters (as many of the existing defined header fields), this would be invalid. However, since the receiving element does not know the definition of the syntax for this field, it must parse it as a header-value. Proxies would forward this header field unchanged. Endpoints would ignore the header field.

Message Details : wsinv

```
INVITE sip:vivekg@chair-dnrc.example.com;unknownparam SIP/2.0
TO :
 sip:vivekg@chair-dnrc.example.com ;   tag    = 1918181833n
from   : "J Rosenberg \\\""       <sip:jdrosen@example.com>
  ;
  tag = 98asjd8
MaX-fOrWaRdS: 0068
Call-ID: wsinv.ndaksdj@192.0.2.1
Content-Length   : 150
cseq: 0009
  INVITE
Via  : SIP  /   2.0
 /UDP
    192.0.2.2;branch=390skdjuw
s :
NewFangledHeader:   newfangled value
 continued newfangled value
UnknownHeaderWithUnusualValue: ;;,,;;,;
Content-Type: application/sdp
Route:
 <sip:services.example.com;lr;unknownwith=value;unknown-no-value>
v:  SIP  / 2.0  / TCP     spindle.example.com   ;
  branch  =   z9hG4bK9ikj8  ,
 SIP  /    2.0   / UDP  192.168.255.111   ; branch=
 z9hG4bK30239
m:"Quoted string \"\"" <sip:jdrosen@example.com> ; newparam =
      newvalue ;
  secondparam ; q = 0.33

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.3
s=-
c=IN IP4 192.0.2.4
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.2. Wide range of valid characters

This message exercises a wider range of characters in several key syntactic elements than implementations usually see. Of particular note:

- o The Method contains non-alpha characters from token. Note that % is not an escape character for this field. A method of IN%56ITE is an unknown method. It is not the same as a method of INVITE
- o The Request-URI contain unusual, but legal, characters
- o A branch parameter contains all non-alphanum characters from token
- o The To header field value's quoted-string contains quoted-pair expansions, including a quoted NULL character
- o The name part of name-addr in the From header field value contains multiple tokens (instead of a quoted string) with all non-alphanum characters from the token production rule. That value also has an unknown header parameter whose name contains the non-alphanum token characters and whose value is a non-ascii range UTF-8 encoded string. The tag parameter on this value contains the non-alphanum token characters
- o The Call-ID header field value contains the non-alphanum characters from word. Notice that in this production:
  - * % is not an escape character. (It is only an escape character in productions matching the rule "escaped")
  - * " does not start a quoted-string. None of ',` or " imply that there will be a matching symbol later in the string
  - * The characters []{}()<> do not have any grouping semantics. They are not required to appear in balanced pairs
- o There is an unknown header field (matching extension-header) with non-alphanum token characters in its name and a UTF8-NONASCII value

If this unusual URI has been defined at a proxy, the proxy will forward this request normally. Otherwise a proxy will generate a 404. Endpoints will generate a 501 listing the methods they understand in an Allow header field.

```
Message Details : intmeth

<allOneLine>
!interesting-Method0123456789_*+`.%indeed'~
 sip:1_unusual.URI~(to-be!sure)&isn't+it$/crazy?,/;;*
:&it+has=1,weird!*pas$wo~d_too.(doesn't-it)
@example.com SIP/2.0
</allOneLine>
Via: SIP/2.0/TCP host1.example.com;branch=z9hG4bK-.!%66*_+`'~
<allOneLine>
To: "BEL:\<hex>07</hex> NUL:\<hex>00</hex> DEL:\<hex>7F</hex>"
 <sip:1_unusual.URI~(to-be!sure)&isn't+it$/crazy?,/;;*
@example.com>
</allOneLine>
<allOneLine>
From: token1~` token2'+_ token3*%!.- <sip:mundane@example.com>
;fromParam''~+*_!.-%=
"<hex>D180D0B0D0B1D0BED182D0B0D18ED189D0B8D0B9</hex>"
;tag=_token~1'+`*%!-.
</allOneLine>
Call-ID: intmeth.word%ZK-!.*_+'@word`~)(><:\/"][?}{
CSeq: 139122385 !interesting-Method0123456789_*+`.%indeed'~
Max-Forwards: 255
<allOneLine>
extensionHeader-!.%*+_`'~:
<hex>EFBBBFE5A4A7E5819CE99BBB</hex>
</allOneLine>
Content-Length: 0
```

3.1.1.3. Valid use of the % escaping mechanism

This INVITE exercises the % HEX HEX escaping mechanism in several places. The request is syntactically valid. Interesting features include:

- o The request-URI has sips:user@example.com embedded in its userpart. What that might mean to example.net is beyond the scope of this document.
- o The From and To URIs have escaped characters in their userparts.
- o The Contact URI has escaped characters in the URI parameters. Note that the "name" uri-parameter has a value of "value%41" which is NOT equivalent to "valueA". Per [RFC2396], unescaping URI components is never performed recursively.

A parser must accept this as a well-formed message. The application using the message must treat the % HEX HEX expansions as equivalent to the character being encoded. The application must not try to interpret % as an escape character in those places where % HEX HEX

("escaped" in the grammar) is not a valid part of the construction. In [RFC3261], "escaped" only occurs in the expansions of SIP-URI, SIPS-URI, and Reason-Phrase.

```
Message Details : esc01

INVITE sip:sips%3Auser%40example.com@example.net SIP/2.0
To: sip:%75se%72@example.com
From: <sip:I%20have%20spaces@example.net>;tag=938
Max-Forwards: 87
i: esc01.239409asdfakjkn23onasd0-3234
CSeq: 234234 INVITE
Via: SIP/2.0/UDP host5.example.net;branch=z9hG4bKkdjuw
C: application/sdp
Contact:
  <sip:cal%6Cer@host5.example.net;%6C%72;n%61me=v%61lue%25%34%31>
Content-Length: 150

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.4. Escaped nulls in URIs

This register request contains several URIs with nulls in the userpart. The message is well formed - parsers must accept this message. Implementations must take special care when unescaping the Address-of-Record (AOR) in this request to not prematurely shorten the username. This request registers two distinct contact URIs.

```
Message Details : escnull

REGISTER sip:example.com SIP/2.0
To: sip:null-%00-null@example.com
From: sip:null-%00-null@example.com;tag=839923423
Max-Forwards: 70
Call-ID: escnull.39203ndfvkjdasfkq3w4otrq0adsfdfnavd
CSeq: 14398234 REGISTER
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bKkdjuw
Contact: <sip:%00@host5.example.com>
Contact: <sip:%00%00@host5.example.com>
L:0
```

3.1.1.5. Use of % when it is not an escape

Most of the places % can appear in a SIP message, it is not an escape character. This can surprise the unwary implementor. The following well-formed request has these properties:

- o The request method is unknown. It is NOT equivalent to REGISTER
- o The display-name portion of the To and From header fields is "%Z%45". Note that this is not the same as %ZE
- o This message has two Contact header field values, not three. <sip:alias2@host2.example.com> is a C%6Fntact header field value

A parser should accept this message as well formed. A proxy would forward or reject the message depending on what the Request-URI meant to it. An endpoint would reject this message with a 501.

Message Details : esc02

```
RE%47IST%45R sip:registrar.example.com SIP/2.0
To: "%Z%45" <sip:resource@example.com>
From: "%Z%45" <sip:resource@example.com>;tag=f232jadfj23
Call-ID: esc02.asdfnqwo34rq23i34jrjasdcnl23nrlknsdf
Via: SIP/2.0/TCP host.example.com;branch=z9hG4bK209%fzsnel234
CSeq: 29344 RE%47IST%45R
Max-Forwards: 70
Contact: <sip:alias1@host1.example.com>
C%6Fntact: <sip:alias2@host2.example.com>
Contact: <sip:alias3@host3.example.com>
l: 0
```

3.1.1.6. Message with no LWS between display name and <

This OPTIONS request is not valid per the grammar in RFC 3261 since there is no LWS between the quoted string in the display name and < in the From header field value. This has been identified as a specification bug that will be removed when RFC 3261 is revised. Elements should accept this request as well formed.

Message Details : lwsdisp

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: "caller"<sip:caller@example.com>;tag=323
Max-Forwards: 70
Call-ID: lwsdisp.1234abcd@funky.example.com
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP funky.example.com;branch=z9hG4bKkdjuw
l: 0
```

3.1.1.7. Long values in header fields

This well-formed request contains header fields with many values and values that are very long. Features include:

- o The To header field has a long display name, and long uri parameter names and values
- o The From header field has long header parameter names and values, in particular a very long tag
- o The Call-ID is one long token

Message Details : longreq

```
INVITE sip:user@example.com SIP/2.0
<allOneLine>
To: "I have a user name of
 <repeat count=10>extreme</repeat> proportion"
<sip:user@example.com:6000;
unknownparam1=very<repeat count=20>long</repeat>value;
longparam<repeat count=25>name</repeat>=shortvalue;
very<repeat count=25>long</repeat>ParameterNameWithNoValue>
</allOneLine>
<allOneLine>
F: sip:
<repeat count=5>amazinglylongcallername</repeat>@example.net
;tag=12<repeat count=50>982</repeat>424
;unknownheaderparam<repeat count=20>name</repeat>=
unknowheaderparam<repeat count=15>value</repeat>
;unknownValueless<repeat count=10>paramname</repeat>
</allOneLine>
Call-ID: longreq.one<repeat count=20>really</repeat>longcallid
CSeq: 3882340 INVITE
<allOneLine>
Unknown-<repeat count=20>Long</repeat>-Name:
 unknown-<repeat count=20>long</repeat>-value;
 unknown-<repeat count=20>long</repeat>-parameter-name =
 unknown-<repeat count=20>long</repeat>-parameter-value
</allOneLine>
Via: SIP/2.0/TCP sip33.example.com
v: SIP/2.0/TCP sip32.example.com
V: SIP/2.0/TCP sip31.example.com
Via: SIP/2.0/TCP sip30.example.com
ViA: SIP/2.0/TCP sip29.example.com
VIa: SIP/2.0/TCP sip28.example.com
VIA: SIP/2.0/TCP sip27.example.com
via: SIP/2.0/TCP sip26.example.com
viA: SIP/2.0/TCP sip25.example.com
vIa: SIP/2.0/TCP sip24.example.com
vIA: SIP/2.0/TCP sip23.example.com
```

```
V :  SIP/2.0/TCP sip22.example.com
v :  SIP/2.0/TCP sip21.example.com
V  : SIP/2.0/TCP sip20.example.com
v  : SIP/2.0/TCP sip19.example.com
Via : SIP/2.0/TCP sip18.example.com
Via  : SIP/2.0/TCP sip17.example.com
Via: SIP/2.0/TCP sip16.example.com
Via: SIP/2.0/TCP sip15.example.com
Via: SIP/2.0/TCP sip14.example.com
Via: SIP/2.0/TCP sip13.example.com
Via: SIP/2.0/TCP sip12.example.com
Via: SIP/2.0/TCP sip11.example.com
Via: SIP/2.0/TCP sip10.example.com
Via: SIP/2.0/TCP sip9.example.com
Via: SIP/2.0/TCP sip8.example.com
Via: SIP/2.0/TCP sip7.example.com
Via: SIP/2.0/TCP sip6.example.com
Via: SIP/2.0/TCP sip5.example.com
Via: SIP/2.0/TCP sip4.example.com
Via: SIP/2.0/TCP sip3.example.com
Via: SIP/2.0/TCP sip2.example.com
Via: SIP/2.0/TCP sip1.example.com
<allOneLine>
Via: SIP/2.0/TCP
 host.example.com;received=192.0.2.5;
branch=very<repeat count=50>long</repeat>branchvalue
</allOneLine>
Max-Forwards: 70
<allOneLine>
Contact: <sip:
<repeat count=5>amazinglylongcallername</repeat>
@host5.example.net>
</allOneLine>
Content-Type: application/sdp
l: 150

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.8. Extra trailing octets in a UDP datagram

This message contains a single SIP REGISTER request, which ostensibly arrived over UDP in a single datagram. The packet contains extra octets after the body (which in this case has zero length). The extra octets happen to look like a SIP INVITE request, but (per section 18.3 of [RFC3261]) they are just spurious noise that must be ignored.

A SIP element receiving this datagram would handle the REGISTER request normally and ignore the extra bits that look like an INVITE request. If the element is a proxy choosing to forward the REGISTER, the INVITE octets would not appear in the forwarded request.

```
Message Details : dblreq

REGISTER sip:example.com SIP/2.0
To: sip:j.user@example.com
From: sip:j.user@example.com;tag=43251j3j324
Max-Forwards: 8
I: dblreq.0ha0isndaksdj99sdfafnl3lk233412
Contact: sip:j.user@host.example.com
CSeq: 8 REGISTER
Via: SIP/2.0/UDP 192.0.2.125;branch=z9hG4bKkdjuw23492
Content-Length: 0


INVITE sip:joe@example.com SIP/2.0
t: sip:joe@example.com
From: sip:caller@example.net;tag=141334
Max-Forwards: 8
Call-ID: dblreq.0ha0isnda977644900765@192.0.2.15
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;branch=z9hG4bKkdjuw380234
Content-Type: application/sdp
Content-Length: 150

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.15
s=-
c=IN IP4 192.0.2.15
t=0 0
m=audio 49217 RTP/AVP 0 12
m =video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.9. Semicolon separated parameters in URI user part

This request has a semicolon-separated parameter contained in the "user" part of the Request-URI (whose value contains an escaped @ symbol). Receiving elements will accept this as a well formed message. The Request-URI will parse such that the user part is "user;par=u@example.net".

```
Message Details : semiuri

OPTIONS sip:user;par=u%40example.net@example.com SIP/2.0
To: sip:j_user@example.com
From: sip:caller@example.org;tag=33242
Max-Forwards: 3
Call-ID: semiuri.0ha0isndaksdj
CSeq: 8 OPTIONS
Accept: application/sdp, application/pkcs7-mime,
        multipart/mixed, multipart/signed,
        message/sip, message/sipfrag
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bKkdjuw
l: 0
```

3.1.1.10. Varied and unknown transport types

This request contains Via header field values with all known transport types and exercises the transport extension mechanism. Parsers must accept this message as well formed. Elements receiving this message would process it exactly as if the 2nd and subsequent header field values specified UDP (or other transport).

```
Message Details : transports

OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: <sip:caller@example.com>;tag=323
Max-Forwards: 70
Call-ID:  transports.kijh4akdnaqjkwendsasfdj
Accept: application/sdp
CSeq: 60 OPTIONS
Via: SIP/2.0/UDP t1.example.com;branch=z9hG4bKkdjuw
Via: SIP/2.0/SCTP t2.example.com;branch=z9hG4bKklasjdhf
Via: SIP/2.0/TLS t3.example.com;branch=z9hG4bK2980unddj
Via: SIP/2.0/UNKNOWN t4.example.com;branch=z9hG4bKasd0f3en
Via: SIP/2.0/TCP t5.example.com;branch=z9hG4bK0a9idfnee
l: 0
```

3.1.1.11. Multipart MIME message

This MESSAGE request contains two body parts. The second part is binary encoded and contains null (0x00) characters. Receivers must take care to properly frame the received message.

Parsers must accept this message as well formed, even if the application above the parser does not support multipart/signed.

Additional examples of multipart/mime messages, in particular S/MIME messages, are available in the security call flow examples draft [I-D.jennings-sip-sec-flows].

```
Message Details : mpart01

MESSAGE sip:kumiko@example.org SIP/2.0
<allOneLine>
Via: SIP/2.0/UDP 127.0.0.1:5070
;branch=z9hG4bK-d87543-4dade06d0bdb11ee-1--d87543-;rport
</allOneLine>
Max-Forwards: 70
Route: <sip:127.0.0.1:5080>
<allOneLine>
Identity: r5mwreLuyDRYBi/0TiPwEsY3rEVsk/G2WxhgTV1PF7hHuL
IK0YWVKZhKv9Mj8UeXqkMVbnVq37CD+813gvYjcBUaZngQmXc9WNZSDN
GCzA+fWl9MEUHWIZo1CeJebdY/XlgKeTa0Olvq0rt70Q5jiSfbqMJmQF
teeivUhkMWYUA=
</allOneLine>
Contact: <sip:fluffy@127.0.0.1:5070>
To: <sip:kumiko@example.org>
From: <sip:fluffy@example.com>;tag=2fb0dcc9
Call-ID: 3d9485ad0c49859b@Zmx1ZmZ5LW1hYy0xNi5sb2NhbA..
CSeq: 1 MESSAGE
Content-Transfer-Encoding: binary
Content-Type: multipart/mixed;boundary=7a9cbec02ceef655
Date: Sat, 15 Oct 2005 04:44:56 GMT
User-Agent: SIPimp.org/0.2.5 (curses)
Content-Length: 553

--7a9cbec02ceef655
Content-Type: text/plain
Content-Transfer-Encoding: binary

Hello
--7a9cbec02ceef655
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

```
<hex>
3082015206092A86
4886F70D010702A08201433082013F02
01013109300706052B0E03021A300B06
092A864886F70D010701318201203082
011C020101307C3070310B3009060355
04061302555331133011060355040813
0A43616C69666F726E69613111300F06
03550407130853616E204A6F7365310E
300C060355040A130573697069743129
3027060355040B132053697069742054
657374204365727469666961746520
417574686F72697479020801950071 02
330113300706052B0E03021A300D0609
2A864886F70D01010105000481808EF4
66F948F0522DD2E5978E9D95AAE9F2FE
15A06659716292E8DA2AA8D8350A68CE
FFAE3CBD2BFF1675DDD5648E593DD647
28F26220F7E941749E330D9A15EDABDB
93D10C42102E7B7289D29CC0C9AE2EFB
C7C0CFF9172F3B027E4FC027E1546DE4
B6AA3ABB3E66CCCB5DD6C64B8383149C
B8E6FF182D944FE57B65BC99D005
</hex>
--7a9cbec02ceef655--
```

3.1.1.12. Unusual reason phrase

This 200 response contains a reason phrase other than "OK". The reason phrase is intended for human consumption, and may contain any string produced by

```
Reason-Phrase   =  *(reserved / unreserved / escaped
                    / UTF8-NONASCII / UTF8-CONT / SP / HTAB)
```

This particular response contains unreserved and non-ASCII UTF-8 characters. This response is well formed. A parser must accept this message.

```
Message Details : unreason

<allOneLine>
SIP/2.0 200 = 2**3 * 5**2 <hex>D0BDD0BE20D181D182
D0BE20D0B4D0B5D0B2D18FD0BDD0BED181D182D0BE20D0B4
D0B5D0B2D18FD182D18C202D20D0BFD180D0BED181D182D0
BED0B5</hex>
</allOneLine>
Via: SIP/2.0/UDP 192.0.2.198;branch=z9hG4bK1324923
Call-ID: unreason.1234ksdfak3j2erwedfsASdf
CSeq: 35 INVITE
From: sip:user@example.com;tag=11141343
To: sip:user@example.edu;tag=2229
Content-Length: 154
Content-Type: application/sdp
Contact: <sip:user@host198.example.com>

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.198
s=-
c=IN IP4 192.0.2.198
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.1.13. Empty reason phrase

This well formed response contains no reason phrase. A parser must accept this message. The space character after the reason code is required. If it were not present, this message could be rejected as invalid (a liberal receiver would accept it anyway).

```
Message Details : noreason

SIP/2.0 100<hex>20</hex>
Via: SIP/2.0/UDP 192.0.2.105;branch=z9hG4bK2398ndaoe
Call-ID: noreason.asndj203insdf99223ndf
CSeq: 35 INVITE
From: <sip:user@example.com>;tag=39ansfi3
To: <sip:user@example.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.example.com>
```

3.1.2. Invalid messages

This section contains several invalid messages reflecting errors seen at interoperability events and exploring important edge conditions that can be induced through malformed messages. This section does not attempt to be a comprehensive list of all types of invalid messages.

3.1.2.1. Extraneous header field separators

The Via header field of this request contains additional semicolons and commas without parameters or values. The Contact header field contains additional semicolons without parameters. This message is syntactically invalid.

An element receiving this request should respond with a 400 Bad Request error.

```
Message Details : badinv01

INVITE sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=134161461246
Max-Forwards: 7
Call-ID: badinv01.0ha0isndaksdjasdf3234nas
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;;,;,,
Contact: "Joe" <sip:joe@example.org>;;;;
Content-Length: 152
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.15
s=-
c=IN IP4 192.0.2.15
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.2. Content length larger than message

This is a request message with a Content Length that is larger than the actual length of the body.

When sent over UDP (as this message ostensibly was), the receiving element should respond with a 400 Bad Request error. If this message arrived over a stream-based transport such as TCP, there's not much

the receiving could do but wait for more data on the stream and close the connection if none is forthcoming in a reasonable period of time.

Message Details : clerr

```
INVITE sip:user@example.com SIP/2.0
Max-Forwards: 80
To: sip:j.user@example.com
From: sip:caller@example.net;tag=93942939o2
Contact: <sip:caller@hungry.example.net>
Call-ID: clerr.0ha0isndaksdjweiafasdk3
CSeq: 8 INVITE
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bK-39234-23523
Content-Type: application/sdp
Content-Length: 9999

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.155
s=-
c=IN IP4 192.0.2.155
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.3. Negative Content-Length

This request has a negative value for Content-Length.

An element receiving this message should respond with an error. This request appeared over UDP, so the remainder of the datagram can simply be discarded. If a request like this arrives over TCP, the framing error is not recoverable and the connection should be closed. The same behavior is appropriate for messages that arrive without a numeric value in the Content-Length header field such as:

```
Content-Length: five
```

Implementors should take extra precautions if the technique they choose for converting this ascii field into an integral form can return a negative value. In particular, the result must not be used as a counter or array index.

Message Details : ncl

```
INVITE sip:user@example.com SIP/2.0
Max-Forwards: 254
To: sip:j.user@example.com
From: sip:caller@example.net;tag=32394234
Call-ID: ncl.0ha0isndaksdj2193423r542w35
CSeq: 0 INVITE
Via: SIP/2.0/UDP 192.0.2.53;branch=z9hG4bKkdjuw
Contact: <sip:caller@example53.example.net>
Content-Type: application/sdp
Content-Length: -999

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.53
s=-
c=IN IP4 192.0.2.53
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.4. Request scalar fields with overlarge values

This request contains several scalar header field values outside their legal range.

- o the CSeq sequence number is >2**32-1.
- o the Max-Forwards value is >255.
- o the Expires value is >2**32-1.
- o the Contact expires parameter value is >2**32-1.

An element receiving this request should respond with a 400 Bad Request due to the CSeq error. If only the Max-Forwards field were in error, the element could choose process the request as if the field were absent. If only the expiry values were in error, the element could treat them as if they contained the default values for expiration (3600 in this case).

Other scalar request fields that may contain aberrant values include, but are not limited to, the Contact q value, the Timestamp value, and the Via ttl parameter.

```
Message Details : scalar02

REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/TCP host129.example.com;branch=z9hG4bK342sdfoi3
To: <sip:user@example.com>
From: <sip:user@example.com>;tag=239232jh3
CSeq: 36893488147419103232 REGISTER
Call-ID: scalar02.23o0pd9vanlq3wnrlnewofjas9ui32
Max-Forwards: 300
Expires: 1<repeat count=100>0</repeat>
Contact: <sip:user@host129.example.com>
  ;expires=280297596632815
Content-Length: 0
```

3.1.2.5. Response scalar fields with overlarge values

This response contains several scalar header field values outside their legal range.

- o the CSeq sequence number is >2**32-1.
- o The Retry-After field is unreasonably large (note that RFC 3261 does not define a legal range for this field).
- o The Warning field has a warning-value with more than 3 digits

An element receiving this response will simply discard it.

```
Message Details : scalarlg

SIP/2.0 503 Service Unavailable
<allOneLine>
Via: SIP/2.0/TCP host129.example.com
;branch=z9hG4bKzzxdiwo34sw
;received=192.0.2.129
</allOneLine>
To: <sip:user@example.com>
From: <sip:other@example.net>;tag=2easdjfejw
CSeq: 9292394834772304023312 OPTIONS
Call-ID: scalarlg.noase0of0234hn2qofoaf0232aewf2394r
Retry-After: 949302838503028349304023988
Warning: 1812 overture "In Progress"
Content-Length: 0
```

3.1.2.6. Unterminated quoted string in display-name

This is a request with an unterminated quote in the display name of the To field. An element receiving this request should return an 400 Bad Request error.

An element could attempt to infer a terminating quote and accept the message. Such an element needs to take care that it makes a reasonable inference when it encounters

```
To: "Mr J. User <sip:j.user@example.com> <sip:realj@example.net>
```

```
Message Details : quotbal

INVITE sip:user@example.com SIP/2.0
To: "Mr. J. User <sip:j.user@example.com>
From: sip:caller@example.net;tag=93334
Max-Forwards: 10
Call-ID: quotbal.aksdj
Contact: <sip:caller@host59.example.net>
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.59:5050;branch=z9hG4bKkdjuw39234
Content-Type: application/sdp
Content-Length: 152

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.15
s=-
c=IN IP4 192.0.2.15
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.7. <> enclosing Request-URI

This INVITE request is invalid because the Request-URI has been enclosed within in "<>".

It is reasonable to always reject a request with this error with a 400 Bad Request. Elements attempting to be liberal with what they accept may choose to ignore the brackets. If the element forwards the request, it must not include the brackets in the messages it sends.

Message Details : ltgtruri

```
INVITE <sip:user@example.com> SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=39291
Max-Forwards: 23
Call-ID: ltgtruri.1@192.0.2.5
CSeq: 1 INVITE
Via: SIP/2.0/UDP 192.0.2.5
Contact: <sip:caller@host5.example.net>
Content-Type: application/sdp
Content-Length: 159

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=3149328700 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.8. Malformed SIP Request-URI (embedded LWS)

This INVITE has illegal LWS within the Request-URI.

An element receiving this request should respond with a 400 Bad Request.

An element could attempt to ignore the embedded LWS for those schemes (like sip) where that would not introduce ambiguity.

Message Details : lwsruri

```
INVITE sip:user@example.com; lr SIP/2.0
To: sip:user@example.com;tag=3xfe-9921883-z9f
From: sip:caller@example.net;tag=231413434
Max-Forwards: 5
Call-ID: lwsruri.asdfasdoeoi2323-asdfwrn23-asd834rk423
CSeq: 2130706432 INVITE
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKkdjuw2395
Contact: <sip:caller@host1.example.net>
Content-Type: application/sdp
Content-Length: 159

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=3149328700 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.9. Multiple SP separating Request-Line elements

This INVITE has illegal multiple SP characters between elements of the start line.

It is acceptable to reject this request as malformed. An element that is liberal in what it accepts may ignore these extra SP characters while processing the request. If the element forwards the request, it must not include these extra SP characters in the messages it sends.

Message Details : lwsstart

```
INVITE  sip:user@example.com  SIP/2.0
Max-Forwards: 8
To: sip:user@example.com
From: sip:caller@example.net;tag=8814
Call-ID: lwsstart.dfknq234oi243099adsdfnawe3@example.com
CSeq: 1893884 INVITE
Via: SIP/2.0/UDP host1.example.com;branch=z9hG4bKkdjuw3923
Contact: <sip:caller@host1.example.net>
Content-Type: application/sdp
Content-Length: 150

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.10. SP characters at end of Request-Line

This OPTIONS request contains SP characters between the SIP-Version field and the CRLF terminating the Request-Line.

It is acceptable to reject this request as malformed. An element that is liberal in what it accepts may ignore these extra SP characters while processing the request. If the element forwards the request, it must not include these extra SP characters in the messages it sends.

Message Details : trws

```
OPTIONS sip:remote-target@example.com SIP/2.0<hex>2020</hex>
Via: SIP/2.0/TCP host1.examle.com;branch=z9hG4bK299342093
To: <sip:remote-target@example.com>
From: <sip:local-resource@example.com>;tag=329429089
Call-ID: trws.oicu34958239neffasdhr2345r
Accept: application/sdp
CSeq: 238923 OPTIONS
Max-Forwards: 70
Content-Length: 0
```

3.1.2.11. Escaped headers in SIP Request-URI

This INVITE is malformed as the SIP Request-URI contains escaped headers.

It is acceptable for an element to reject this request with a 400 Bad Request. An element could choose to be liberal in what it accepts and ignore the escaped headers. If the element is a proxy, the escaped headers must not appear in the Request-URI of forwarded request (and most certainly must not be translated into the actual header of the forwarded request).

```
Message Details : escruri

INVITE sip:user@example.com?Route=%3Csip:example.com%3E SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=341518
Max-Forwards: 7
Contact: <sip:caller@host39923.example.net>
Call-ID: escruri.23940-asdfhj-aje3br-234q098w-fawerh2q-h4n5
CSeq: 149209342 INVITE
Via: SIP/2.0/UDP host-of-the-hour.example.com;branch=z9hG4bKkdjuw
Content-Type: application/sdp
Content-Length: 150

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.12. Invalid timezone in Date header field

This INVITE is invalid as it contains a non GMT time zone in the SIP Date header field.

It is acceptable to reject this request as malformed (though an element shouldn't do that unless the contents of the Date header field were actually important to its processing). An element wishing to be liberal in what it accepts could ignore this value altogether if it wasn't going to use the Date header field anyhow. Otherwise, it could attempt to interpret this date and adjust it to GMT.

RFC 3261 explicitly defines the only acceptable timezone designation as "GMT". "UT", while synonymous with GMT per [RFC2822], is not

valid. "UTC" and "UCT" are also invalid.

Message Details : baddate

```
INVITE sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=2234923
Max-Forwards: 70
Call-ID: baddate.239423mnsadf3j23lj42--sedfnm234
CSeq: 1392934 INVITE
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKkdjuw
Date: Fri, 01 Jan 2010 16:00:00 EST
Contact: <sip:caller@host5.example.net>
Content-Type: application/sdp
Content-Length: 150

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.13. Failure to enclose name-addr URI in <>

This REGISTER request is malformed. The SIP URI contained in the Contact Header field has an escaped header, so the field must be in name-addr form (which implies the URI must be enclosed in <>).

It is reasonable for an element receiving this request to respond with a 400 Bad Request. An element choosing to be liberal in what it accepts could infer the angle brackets since there is no ambiguity in this example. In general, that won't be possible.

Message Details : regbadct

```
REGISTER sip:example.com SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=998332
Max-Forwards: 70
Call-ID: regbadct.k345asrl3fdbv@10.0.0.1
CSeq: 1 REGISTER
Via: SIP/2.0/UDP 135.180.130.133:5060;branch=z9hG4bKkdjuw
Contact: sip:user@example.com?Route=%3Csip:sip.example.com%3E
l: 0
```

3.1.2.14. Spaces within addr-spec

This request is malformed since the addr-spec in the To header field contains spaces. Parsers receiving this request must not break. It is reasonable to reject this request with a 400 Bad Request response. Elements attempting to be liberal may ignore the spaces.

Message Details : badaspec

```
OPTIONS sip:user@example.org SIP/2.0
Via: SIP/2.0/UDP host4.example.com:5060;branch=z9hG4bKkdju43234
Max-Forwards: 70
From: "Bell, Alexander" <sip:a.g.bell@example.com>;tag=433423
To: "Watson, Thomas" < sip:t.watson@example.org >
Call-ID: badaspec.sdf0234n2nds0a099u23h3hnnw009cdkne3
Accept: application/sdp
CSeq: 3923239 OPTIONS
l: 0
```

3.1.2.15. Non-token characters in display-name

This OPTIONS request is malformed since the display names in the To and From header fields contain non-token characters but are unquoted.

It is reasonable to always reject this kind of error with a 400 Bad Request response.

An element may attempt to be liberal in what it receives and infer the missing quotes. If this element were a proxy, it must not propagate the error into the request it forwards. As a consequence, if the fields are covered by a signature, there's not much point in trying to be liberal - the message should be simply rejected.

Message Details : baddn

```
OPTIONS sip:t.watson@example.org SIP/2.0
Via:     SIP/2.0/UDP c.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards:      70
From:    Bell, Alexander <sip:a.g.bell@example.com>;tag=43
To:      Watson, Thomas <sip:t.watson@example.org>
Call-ID: baddn.31415@c.example.com
Accept: application/sdp
CSeq:    3923239 OPTIONS
l: 0
```

3.1.2.16. Unknown protocol version

To an element implementing [RFC3261], this request is malformed due to its high version number.

The element should respond to the request with a 505 Version Not Supported error.

```
Message Details : badvers

OPTIONS sip:t.watson@example.org SIP/7.0
Via:     SIP/7.0/UDP c.example.com;branch=z9hG4bKkdjuw
Max-Forwards:     70
From:    A. Bell <sip:a.g.bell@example.com>;tag=qweoiqpe
To:      T. Watson <sip:t.watson@example.org>
Call-ID: badvers.31417@c.example.com
CSeq:    1 OPTIONS
l: 0
```

3.1.2.17. Start line and CSeq method mismatch

This request has mismatching values for the method in the start line and the CSeq header field. Any element receiving this request will respond with a 400 Bad Request.

```
Message Details : mismatch01

OPTIONS sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch01.dj0234sxdfl3
CSeq: 8 INVITE
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKkdjuw
l: 0
```

3.1.2.18. Unknown Method with CSeq method mismatch

This message has an unknown method in the start line, and a CSeq method tag which does not match.

Any element receiving this response should respond with a 501 Not Implemented. A 400 Bad Request is also acceptable, but choosing a 501 (particularly at proxies) has better future-proof characteristics.

```
Message Details : mismatch02

NEWMETHOD sip:user@example.com SIP/2.0
To: sip:j.user@example.com
From: sip:caller@example.net;tag=34525
Max-Forwards: 6
Call-ID: mismatch02.dj0234sxdfl3
CSeq: 8 INVITE
Contact: <sip:caller@host.example.net>
Via: SIP/2.0/UDP host.example.net;branch=z9hG4bKkdjuw
Content-Type: application/sdp
l: 138

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
c=IN IP4 192.0.2.1
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.1.2.19. Overlarge response code

This response has a response code larger than 699. An element receiving this response should simply drop it.

```
Message Details : bigcode

SIP/2.0 4294967301 better not break the receiver
Via: SIP/2.0/UDP 192.0.2.105;branch=z9hG4bK2398ndaoe
Call-ID: bigcode.asdof3uj203asdnf3429uasdhfas3ehjasdfas9i
CSeq: 353494 INVITE
From: <sip:user@example.com>;tag=39ansfi3
To: <sip:user@example.edu>;tag=902jndnke3
Content-Length: 0
Contact: <sip:user@host105.example.com>
```

3.2. Transaction layer semantics

This section contains tests that exercise an implementation's parser and transaction layer logic.

3.2.1. Missing transaction identifier

This request indicates support for RFC 3261-style transaction identifiers by providing the z9hG4bK prefix to the branch parameter, but it provides no identifier. A parser must not break when receiving this message. An element receiving this request could

reject the request with a 400 Response (preferably statelessly, as other requests from the source are likely to also have a malformed branch parameter), or it could fall back to the RFC 2543 style transaction identifier.

Message Details : badbranch

```
OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.org;tag=33242
Max-Forwards: 3
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bK
Accept: application/sdp
Call-ID: badbranch.sadonfo23i420jv0as0derf3j3n
CSeq: 8 OPTIONS
l: 0
```

3.3. Application layer semantics

This section contains tests that exercise an implementation's parser and application layer logic.

3.3.1. Missing Required Header Fields

This request contains no Call-ID, From, or To header fields.

An element receiving this message must not break because of the missing information. Ideally, it will respond with a 400 Bad Request error.

Message Details : insuf

```
INVITE sip:user@example.com SIP/2.0
CSeq: 193942 INVITE
Via: SIP/2.0/UDP 192.0.2.95;branch=z9hG4bKkdj.insuf
Content-Type: application/sdp
l: 152

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.95
s=-
c=IN IP4 192.0.2.95
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.3.2. Request-URI with unknown scheme

This OPTIONS contains an unknown URI scheme in the Request-URI. A parser must accept this as a well-formed SIP request.

An element receiving this request will reject it with a 416 Unsupported URI Scheme response.

Some early implementations attempt to look at the contents of the To header field to determine how to route this kind of request. That is an error. Despite the fact that the To header field and the Request URI frequently look alike in simplistic first-hop messages, the To header field contains no routing information.

Message Details : unkscm

```
OPTIONS nobodyKnowsThisScheme:totallyopaquecontent SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=384
Max-Forwards: 3
Call-ID: unkscm.nasdfasser0q239nwsdfasdkl34
CSeq: 3923423 OPTIONS
Via: SIP/2.0/TCP host9.example.com;branch=z9hG4bKkdjuw39234
Content-Length: 0
```

3.3.3. Request-URI with known but atypical scheme

This OPTIONS contains an Request-URI with an IANA registered scheme that does not commonly appear Request-URIs of SIP requests. A parser must accept this as a well-formed SIP request.

If an element will never accept this scheme as meaningful in a request-URI, it is appropriate to treat it as unknown and return a 416 Unsupported URI Scheme response. If the element might accept some URIs with this scheme, then a 404 Not Found is appropriate for those URIs it doesn't accept.

Message Details : novelsc

```
OPTIONS soap.beep://192.0.2.103:3002 SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=384
Max-Forwards: 3
Call-ID: novelsc.asdfasser0q239nwsdfasdkl34
CSeq: 3923423 OPTIONS
Via: SIP/2.0/TCP host9.example.com;branch=z9hG4bKkdjuw39234
Content-Length: 0
```

3.3.4. Unknown URI schemes in header fields

This message contains registered schemes in the To, From and Contact header fields of a request. The message is syntactically valid. Parsers must not fail when receiving this message.

Proxies should treat this message as they would any other request for this URI. A registrar would reject this request with a 400 Bad Request response since the To: header field is required to contain a SIP or SIPS URI as an AOR.

```
Message Details : unksm2

REGISTER sip:example.com SIP/2.0
To: isbn:2983792873
From: <http://www.example.com>;tag=3234233
Call-ID: unksm2.daksdj@hyphenated-host.example.com
CSeq: 234902 REGISTER
Max-Forwards: 70
Via: SIP/2.0/UDP 192.0.2.21:5060;branch=z9hG4bKkdjuw
Contact: <name:John_Smith>
l: 0
```

3.3.5. Proxy-Require and Require

This request tests proper implementation of SIP's Proxy-Require and Require extension mechanisms.

Any element receiving this request will respond with a 420 Bad Extension response containing an Unsupported header field listing these features from either the Require or Proxy-Require header field depending on the role in which the element is responding.

```
Message Details : bext01

OPTIONS sip:user@example.com SIP/2.0
To: sip:j_user@example.com
From: sip:caller@example.net;tag=242etr
Max-Forwards: 6
Call-ID: bext01.0ha0isndaksdj
Require: nothingSupportsThis, nothingSupportsThisEither
Proxy-Require: noProxiesSupportThis, norDoAnyProxiesSupportThis
CSeq: 8 OPTIONS
Via: SIP/2.0/TLS fold-and-staple.example.com;branch=z9hG4bKkdjuw
Content-Length: 0
```

3.3.6. Unknown Content-Type

This INVITE request contains a body of unknown type. It is syntactically valid. A parser must not fail when receiving it.

A proxy receiving this request would process it just like any other INVITE. An endpoint receiving this request would reject it with a 415 Unsupported Media Type error.

```
Message Details : invut

INVITE sip:user@example.com SIP/2.0
Contact: <sip:caller@host5.example.net>
To: sip:j.user@example.com
From: sip:caller@example.net;tag=8392034
Max-Forwards: 70
Call-ID: invut.0ha0isndaksdjadsfij34n23d
CSeq: 235448 INVITE
Via: SIP/2.0/UDP somehost.example.com;branch=z9hG4bKkdjuw
Content-Type: application/unknownformat
Content-Length: 40

<audio>
 <pcmu port="443"/>
</audio>
```

3.3.7. Unknown authorization scheme

This REGISTER request contains an Authorization header field with an unknown scheme. The request is well-formed. A parser must not fail when receiving it.

A proxy will treat this request as any other REGISTER. If it forwards the request, it will include this Authorization header field unmodified in the forwarded messages.

A registrar that does not care about challenge-response authentication will simply ignore the Authorization header field, processing this registration as if the field were not present. A registrar that does care about challenge-response authentication will reject this request with a 401, issuing a new challenge with a scheme it understands.

Endpoints choosing not to act as registrars will simply reject the request. A 405 Method Not Allowed is appropriate.

```
Message Details : regaut01

REGISTER sip:example.com SIP/2.0
To: sip:j.user@example.com
From: sip:j.user@example.com;tag=87321hj23128
Max-Forwards: 8
Call-ID: regaut01.0ha0isndaksdj
CSeq: 9338 REGISTER
Via: SIP/2.0/TCP 192.0.2.253;branch=z9hG4bKkdjuw
Authorization: NoOneKnowsThisScheme opaque-data=here
Content-Length:0
```

3.3.8. Multiple values in single value required fields

The message contains a request with multiple Call-ID, To, From, Max-Forwards and CSeq values. An element receiving this request must not break.

An element receiving this request would respond with a 400 Bad Request error.

```
Message Details : multi01

INVITE sip:user@company.com SIP/2.0
Contact: <sip:caller@host25.example.net>
Via: SIP/2.0/UDP 192.0.2.25;branch=z9hG4bKkdjuw
Max-Forwards: 70
CSeq: 5 INVITE
Call-ID: multi01.98asdh@192.0.2.1
CSeq: 59 INVITE
Call-ID: multi01.98asdh@192.0.2.2
From: sip:caller@example.com;tag=3413415
To: sip:user@example.com
To: sip:other@example.net
From: sip:caller@example.net;tag=2923420123
Content-Type: application/sdp
l: 154
Contact: <sip:caller@host36.example.net>
Max-Forwards: 5

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.25
s=-
c=IN IP4 192.0.2.25
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
```

```
a=rtpmap:31 LPC
```

3.3.9. Multiple Content-Length values

Multiple conflicting Content-Length header field values appear in this request.

From a framing perspective, this situation is equivalent to an invalid Content-Length value (or no value at all).

An element receiving this message should respond with an error. This request appeared over UDP, so the remainder of the datagram can simply be discarded. If a request like this arrives over TCP, the framing error is not recoverable and the connection should be closed.

Message Details : mcl01

```
OPTIONS sip:user@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.net;branch=z9hG4bK293423
To: sip:user@example.com
From: sip:other@example.net;tag=3923942
Call-ID: mcl01.fhn2323orihawfdoa3o4r52o3irsdf
CSeq: 15932 OPTIONS
Content-Length: 13
Max-Forwards: 60
Content-Length: 5
Content-Type: text/plain
```

There's no way to know how many octets are supposed to be here.

3.3.10. 200 OK Response with broadcast Via header field value

This message is a response with a 2nd Via header field value's sent-by containing 255.255.255.255. The message is well formed - parsers must not fail when receiving it.

Per [RFC3261] an endpoint receiving this message should simply discard it.

If a proxy followed normal response processing rules blindly, it would forward this response to the broadcast address. To protect against this being used as an avenue of attack, proxies should drop such responses.

```
Message Details : bcast

SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.198;branch=z9hG4bK1324923
Via: SIP/2.0/UDP 255.255.255.255;branch=z9hG4bK1saber23
Call-ID: bcast.0384840201234ksdfak3j2erwedfsASdf
CSeq: 35 INVITE
From: sip:user@example.com;tag=11141343
To: sip:user@example.edu;tag=2229
Content-Length: 154
Content-Type: application/sdp
Contact: <sip:user@host28.example.com>

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.198
s=-
c=IN IP4 192.0.2.198
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.3.11. Max-Forwards of zero

This is a legal SIP request with the Max-Forwards header field value set to zero.

A proxy should not forward the request and respond 483 (Too Many Hops). An endpoint should process the request as if the Max-Forwards field value were still positive.

```
Message Details : zeromf

OPTIONS sip:user@example.com SIP/2.0
To: sip:user@example.com
From: sip:caller@example.net;tag=3ghsd41
Call-ID: zeromf.jfasdlfnm2o2l43r5u0asdfas
CSeq: 39234321 OPTIONS
Via: SIP/2.0/UDP host1.example.com;branch=z9hG4bKkdjuw2349i
Max-Forwards: 0
Content-Length: 0
```

3.3.12. REGISTER with a contact header parameter

This register request contains a contact where the 'unknownparam' parameter must be interpreted as being a contact-param and not a url-param.

This REGISTER should succeed. The response must not include "unknownparam" as a url-parameter for this binding. Likewise, "unknownparam" must not appear as a url-parameter in any binding during subsequent fetches.

Behavior is the same, of course, for any known contact-param parameter names.

Message Details : cparam01

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP saturn.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards: 70
From: sip:watson@example.com;tag=DkfVgjkrtMwaerKKpe
To: sip:watson@example.com
Call-ID: cparam01.70710@saturn.example.com
CSeq: 2 REGISTER
Contact: sip:+19725552222@gw1.example.net;unknownparam
l: 0
```

3.3.13. REGISTER with a url parameter

This register request contains a contact where the URI has an unknown parameter.

The register should succeed and a subsequent retrieval of the registration must include "unknownparam" as a url-parameter.

Behavior is the same, of course, for any known url-parameter names.

Message Details : cparam02

```
REGISTER sip:example.com SIP/2.0
Via: SIP/2.0/UDP saturn.example.com:5060;branch=z9hG4bKkdjuw
Max-Forwards: 70
From: sip:watson@example.com;tag=838293
To: sip:watson@example.com
Call-ID: cparam02.70710@saturn.example.com
CSeq: 3 REGISTER
Contact: <sip:+19725552222@gw1.example.net;unknownparam>
l: 0
```

3.3.14. REGISTER with a url escaped header

This register request contains a contact where the URI has an escaped header.

The register should succeed and a subsequent retrieval of the registration must include the escaped Route header in the contact URI for this binding.

Message Details : regescrt

```
REGISTER sip:example.com SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=8
Max-Forwards: 70
Call-ID: regescrt.k345asrl3fdbv@192.0.2.1
CSeq: 14398234 REGISTER
Via: SIP/2.0/UDP host5.example.com;branch=z9hG4bKkdjuw
M: <sip:user@example.com?Route=%3Csip:sip.example.com%3E>
L:0
```

3.3.15. Unacceptable Accept offering

This request indicates the response must contain a body in an unknown type. In particular, since the Accept header field does not contain application/sdp, the response may not contain an SDP body. The recipient of this request could respond with a 406 Not Acceptable with a Warning/399 indicating that a response cannot be formulated in the formats offered in the Accept header field. It is also appropriate to respond with a 400 Bad Request since all SIP User-Agents (UAs) supporting INVITE are required to support application/sdp.

Message Details : sdp01

```
INVITE sip:user@example.com SIP/2.0
To: sip:j_user@example.com
Contact: <sip:caller@host15.example.net>
From: sip:caller@example.net;tag=234
Max-Forwards: 5
Call-ID: sdp01.ndaksdj9342dasdd
Accept: text/nobodyKnowsThis
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;branch=z9hG4bKkdjuw
Content-Length: 150
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=0 0
m=audio 49217 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC
```

3.4. Backward compatibility

3.4.1. INVITE with RFC2543 syntax

This is a legal message per RFC 2543 (and several bis versions) which should be accepted by RFC 3261 elements which want to maintain backwards compatibility.

- There is no branch parameter at all on the Via header field value
- There is no From tag
- There is no explicit Content-Length (The body is assumed to be all octets in the datagram after the null-line)
- There is no Max-Forwards header field

Message Details : inv2543

```
INVITE sip:UserB@example.com SIP/2.0
Via: SIP/2.0/UDP iftgw.example.com
From: <sip:+13035551111@ift.client.example.net;user=phone>
Record-Route: <sip:UserB@example.com;maddr=ss1.example.com>
To: sip:+16505552222@ss1.example.net;user=phone
Call-ID: inv2543.1717@ift.client.example.com
CSeq: 56 INVITE
Content-Type: application/sdp

v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.5
s=-
c=IN IP4 192.0.2.5
t=0 0
m=audio 49217 RTP/AVP 0
```

4. Security Considerations

This document presents NON NORMATIVE examples of SIP session establishment. The 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 draft are designed to stress a parser implementation at points traditionally used for such attacks. This document does not, however, attempt to be comprehensive. It should be considered a seed to stimulate thinking and planning, not simply a set of tests to be passed.

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgments

The final detailed review of this document was performed by: Diego Besprosvan, Vijay Gurbani, Shashi Kumar, Derek MacDonald, Gautham Narasimhan, Nils Ohlmeier, Bob Penfield, Reinaldo Penno, Marc Petit-Huguenin, Richard Sugarman, and Venkatesh Venkataramanan.

Earlier versions of this document were reviewed by: Aseem Agarwal, Rafi Assadi, Gonzalo Camarillo, Ben Campbell, Cullen Jennings, Vijay

Gurbani, Sunitha Kumar, Rohan Mahy, Jon Peterson, Marc Petit-Huguenin, Vidhi Rastogi, Adam Roach, Bodgey Yin Shaohua and Tom Taylor.

Thanks to Cullen Jennings and Eric Rescorla for their contribution to the multipart/mime sections of this document and their work constructing S/MIME examples in [I-D.jennings-sip-sec-flows]. Thanks to Neil Deason for contributing several messages and Kundan Singh for performing parser validation of messages in earlier versions.

The following individuals provided significant comments during the early phases of the development of this document: Jean-Francois Mule, Hemant Agrawal, Henry Sinnreich, David Devanatham, Joe Pizzimenti, Matt Cannon, John Hearty, the whole MCI IPOP Design team, Scott Orton, Greg Osterhout, Pat Sollee, Doug Weisenberg, Danny Mistry, Steve McKinnon, and Denise Ingram, Denise Caballero, Tom Redman, Ilya Slain, Pat Sollee, John Truetken, and others from MCI, 3Com, Cisco, Lucent and Nortel.

7. Informative References

[I-D.jennings-sip-sec-flows]
Jennings, C. and K. Ono, "Example call flows using SIP security mechanisms", draft-jennings-sip-sec-flows-03 (work in progress), July 2005.

[RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[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.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

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 3.

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);
```

Figure 58

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 --
H4sIAMUmXEMCA+xdW2zc2Hm2nexNG6UN3LRF0QdaiOJdyxwdnkMOhyOPVrIt
27It22tdvHYTeM8MDzWc4ZAjkqORvK2bbIAAedmHtEHRdlvkoUCLFAjSlyLF
9rJPLYoWrTdAg6JFHwp0i+5D0SIoEAQFuj2HnAuH5GgoW3PxmgcazYU/b4f/
//3Xc04Rq9ipk1JGxe6xITVAW1YUvXc5K/W8syYheEygP0lIECWJ0gkSkMAx
DhwbQWs4LrY57phdcerYrjr96AZtb91L5/0paTdvbazevLHOOXo933CIvUT2
cK1ukIxlb3Prq7fmYQZMT23pON/+Nr958RZXthxXzLRpS1YtL4EsWCja2CyV
Cw+U8mWxeK2qVhoigkicnlrDe/wly25iW3XynEwPecmmO3Ez54lhnOWWDXoo
UyX2DHeOXQnObGeKdMtS4AyLCy7eLogIiRBNT21YdOc72HUs8yy3UbZq2KH7
erfhZpre7z23sjg9dQEbBr96Mc8V2zzvqBqgl2dCU3UABorSgKiMyqbZBEAp
qVWT0DMtl0qk7uY5XK8begm7umXOO2qdHm+d7OQ5pEB6iwrX6sjpKSPP0dub
nno6nj/tC/+hDREABsm/CLJh+YcCTOV/rPJPJa4r/0za4ijaYsy2lah49eKH
J7AIQRGGAQDFQIqg0E8ZmBFCKHKQBAYkusXFDlYtU7Mg0kUIKrsAO4DCioYq
yGxLbO5pldUhyT8VfDJM9T9I/gUAxWPsnygCAWURlX8xm031/0ja6o2t1Y2V
oxd/k7ie+EOqXxWmsKMWQFB6PR6kapTq9ppJZZgKLERGRYQ87xBVM2ueFeHL
r0BVroJEzr/0PsZJ0DaJMUua01MX6Snz3CVbP8sBgbuKTQ5SVuSEbB4A+set
rG/QM1qmi0sUe84F7pEdX8oE7nTRJySmy2/s10kcUrU2Xyfmtlum9yB5uLNb
oG9WoVamxo9B9jmoyNSUkKFMb1ChN8it3hI7wChNTzkFfnqqVIhucAuAIVmt
gBuqbnG0ywWZu71xa3556xZHbwqybbu6SiwOQdjdhITpKVyw3XoN1/NI4K7f
```

uvAso+Gzif/mUNE/gf8noJD9JyJJSvF/1PZfrOPU6wOyFoTaUhIfsBlGf691
nUDaQm7gYC/QV0pe6/UC/V3jbiXkAlK2p/wmSEs99zDQ36Otj8v3tMq/bu4C
YZzxHxmBsP0nA5TK/8TZf5XMY1iAAhKFLLXoBShmI2Zgr0j6nAjKGOiOqeKq
o1awQ21BFqPBTtd/62v5dXxIaWHh7MLZswEDbuaqRVrRpYpFenBhgbY4Ew0O
NOsOacEJfU04YVw2HO31XWI74/T/gCgG9b/kyT/KpvI/Mfpfjuh/OU7/J1X9
PZp/OeMp/0Eaf6dJLH2nTgJ6fyPTUv2JVb7H6Uzpy2Gl39HtwjMWHCqWMHWW
x2r/U9GPxH9QNpX/kbSW9uQgANzNawepVCUXkm8BQT+0E9kHSlIm8Arv5+Ai
sdmOXdn0uBCgnJgTAQQCVfhU+Wu4iiqQ2E2ias7yuqp1ci5SxwTomh9h08Q3
PgQq7ajtLkTIiNpoRamgEmcBiIkCO90AkXdwFh6CuSC+LB7eUlByfU0FtimN
96TtaPCf7LnD9f4SxH9kKWz/iVBI8X/y8n+V+4+TAhAhce2wIZgNwr/PhD2O
3/TUbbLT0G0KuqbllnVze71Rr1u262yUdeds3I8rulsm9ES3bGtvnw/szn7Q
idOibe9vX7SWzf3otpg8YY+G27i+zmmWofIUyHnKPZ4iGWQFh3XLpFiWRX27
ZKljzf8BSRAj/l8a/xmt/SdCRVSyMgIClUbXJTYTMK5oE1zlqFRxNikRfZdJ
V38LEYQtPYiUHBVniwRlvcVw2FEtDTUqECD60dQQvYIG/VTWsINI2Qv8YEfR
uzYfEhUxZPedi4Mt32FECjYdTW9ZfufiTD+fUAGwYqpmldX6xEhpHwOP3mzE
wnsa9X+JIrY95vyPnIVh/48a7an8T0z8t1dz5540IqywPD/9Z8F+CfaGuW3v
hzPsbQTxWbbHWGgSHVO0UKsoQZC4N30f1dg8y++IPEQSRIdO7Cu0PUZc+IDA
8HAjwyXK1Lg23vyPBISo/pdT+R9Fu71yeXV9Y+W2J6+xwh+RHwe7Ddt8rMRv
N/LLTheK17ZjNher2tZ2pWq7a01M7GvX2kHf+F2CwNDmZco9AliKXmcbHCDX
vu0AALHDzwmKDCkLQtqWtptCEIIWGmbVtJqmd5JPTHS41WdwnPIvZuWI/MNU
/z+r8p9DOaokDyfzcKDMoziZP3cooV/8BOaE1KJhk52xxv8EJGXD9r+U1n9N
ivwnM/WjW1uFWlASKqiCYGQQUG56ajXPtfivx5pXFOb7a6aBjCpESBRgSE23
Thau8+3a/l1R7x+tgFIcYnnlyn3CdfSCu95SsIikp7/a19i7PUmhDHV6UVw3
dcAu3FmKLFO5UjxpWgqWkhymUCamE1AO+PXWwy9qfvKSGC6x55NmuqKNOKVh
p38G5/+RHMZ/mMZ/Rh7/oS9nFi0zaJ0VQQC7gkgV1QuzsuSQWRnGoZ1nX63O
QopYu4S+0S4sESd4vFYMFuUiuCdPT+l5zudPNjIEKF5IuFqpmhBZJv0CeBQY
FgLZwFAxadiHgW58ouaAJP/0FNeJU81mL8QNBVmgv9POWDBns0KNFHbpm9Gg
ty7NInEWCYtHhJp9QTMtDUjbYfEfjjn+n436/1SSU/wfjf0/K8rUVp4VJd8H
sMm27rg2tjN9vYGZ2XuUvFVMbRPHatilHmN3sTO8fyClB/8aRLCCVa3SUxTm
syYDfXOnaSHR3oFIR2LFZrnBkmlAZNpG1XRYXVhvgv7CwDGAECiz2gPHJEZQ
gyhIFLlgj8QOW+wJHmBDx47gpwRDCcELs9lLEVLol4eFSaPHRB4hChEeefyB
drLZMIzx5v/FaPwvm47/nzD/n7EJPwsAzz70c237ErUie4oC/fk7DhgO3GZJ
pECATFXbrVZU7GjVHdQULdfeAVh1NAoKeFftDAkWkZJj1t8Bjv+gtF+gUKcr
ivRWliI7LsZQ9SO8np/seCHtbLth6+Md/y9G4n+ylOr/Uft/4QDea7ethksK
s+hCCBtm0cqRzA2AREEScnFjwvqNuvfwo19dQJuVPWeRZ3ZDucLjCkFFm6fY
sAOUXJPXcJPYZbjDl0VT6sIHhRqq/OHB3iNvabxbJnyZmjHJK/6GGTxL3cC0
PVnTTaehjdn/E7OR8b9imv8dO/53Mb6Fkl7d1uCcghKTU8h4fDYQFI3WsNtD
4qDSN4mgpHOjDJJ/t0bc8ljtP9hT/5X17b/U/xtJO0UZgNjEcXVzm1+jnGCp
3vA7KSvnlPtn5t7MzOqmSoh6+qEHEsL9htlwGtjIbN5effiKa/FFcspp2OTV
L+qOedqd090vzJds/GD/tbPzCwtn8l/U3bkydgrC2SbRbfXUmTp2vtC0Hqr3
XcvKvKJahO3G6+6rS4PLT9qRHeGg8s3Mqdls9sz9uTdPP2zFq86vXM9/6QXu
xiZ9O8ZdZN9+tRWUOuz9xIa5XKtKTOHhm/4HeHruvv8JnZk9leH9E9UaporN
UPBLo7vfYqUlp08/nDtznxLPFmY++Mqj7z36o0d//sHbj773wTsffOPR9x/9
yYxnMN/3jvpQOD33Jj0ynwkYv205blq2OnvvGn8qQ3vg9BL7+ubDV19ZPJf/
0vzMl3/ptV95KzCNlgAhykncIXggbKxDVp5K9iimOxTErxCsEpuee/bM3H3a
/fn/eu8vPvy93//wq+9+9NvvTe4YmGfb/tuFkojGGv/LeuP/QvZfOv5v1Pbf
JrX/zi8lqwHUNXe7memX8Z0TEECSJAm0LVHSTMnQqeD3FtbRsxXqZcski2yo
X4kCFe8FHFrHiFzNQg2rql1wnHCcvR2BmBOyEmjX8QXJek/XA5o+8wuyIMdd

Z6CmSMp2DN+jnRPm8Wf1OxLspF3QcMed/wNy2P4T0/k/J8n/SzoL5hMNC8p5
AX/xwNyAz669M0SpjqZX2CzeSO3WgkiieED5m2PVSLJZQvsLe6syWLPsGnaj
xo3oWTfnPMmlPcOdq5dqDY6NMS7MiCKamac/nptvbx7b8zcsc3vYBcAD5D+b
lVEk/p/m/yZI/j0napVjRVwc5hgdZ+Ia4SyNo7a/TWrkCd+4um0x0aByNRM7
pjefpX3aU4wvFHaJvc/Y9yhfu9hokAX2yTsJu8thvwpOmd65f+Jh3FLcy/M5
CXX6btC3O7pbvmFtsQtgHq2P0biGH1B30PAuxwdsdrHD/DlaEA2V3CT+iVBs
s2LZ83iHwisF/xSBM3g8cjT/2tfvPXWDOE7nFo78Q0B7t5UNtcJtQn/cH/3/
NsvpHVsB5VjlAOgYC5t+z/DXKeVRvngma3mu1fH8UYs07z/X4R2/3sYM3sP+
wijO5N1TTAyQIhRCvW7abpQC9lJsRSmEEEXceUCYZjlCA5UQzWr0OL0zgjGa
mOPIoXuKuR6YDdPEHEcK0cRdjximiTlOqI+3uDwXoQn18m4cTbifOS56LhA+
TpRGUCLPK4YoFyWKoZIHP3ohm4BGSkAjJqBBCWhgApoELC2AwTTKYJLcYJIE
nZygjxN0cYIeTtDBSfp3AEnEq2xNnKMWOnGdtqM5KmNvFC//llpwPbBmdgy2
5WOs2WE8C0U5hrvtDrsAcOD8P5H5XyUhrf8bqf8fP5HW0VT5KVARInnDYKF/
hweFpUDwu5WlHFxvIg1zmR7liAP6SBAVBHMymBAgMJqOqjv18c7/HlP/Iafx
/5G0o1n/b8YXuZlzMTjQnZFvQN1/mxVZ6QEultQljXqY+3HJuCzoMzUng4XI
XvGB/XTpv7b8j7/+H4Xz/xIA6fwf447/L3CGPRgCfNHe0wivUCWWyyH+gaIl
mRUY+TOzR/J9Ui8keNzpTwaqWsTS2dJLXnF/0zb9Tzkk2lVvUFEr+ScgalJm
RZSgVlXoO3URRMpBhoUwesPiYMdi0gyLtD0t+M8A0B2r/yfDiP+X2n8jxf94
+6/vDLBP5BDmcoLYi/EtDtSq5g61/SjGiwgoClbZ0G/cJGgpxgYUcgrK5QbM
9yEMsgOR0p7jdTgoP/Hho1rJGPf8P0CSs2H/L53/YaL8v0NPpsOmUoAoGUxY
bNWGuLARG24UAAqfVbWyyUxAy9bLuKmpFkaWaEvQQrrtdFcIovYVNf86TmJE
MCOeaBZEqaSwwLtkz52vG1g3mVhv0Msmpx3OtLgm3udci2P5UNo1Ta6GzX3O
KrnEdThsE85hq0s4RGVERcKxHTOT4XzWdKeG3VJ5mCAwSP6RJIbkX4Jp/eck
xn8eq64TiRKUDlr+JciDFTbzoLOnakbSqdzTUM+RyD8cn/zLXvy3x/6nX1L5
H0W7sXJnbWXjys2Lk4EAcAAC9LXUQ4b6wUjRf/K/wRlhNlPh4U36OGt+ImI0
NcrUw14AboD8Q1Fg/j+CMn3yCHjzP0mp/h9NW1tZX1++7AeAq42aXrUiq//G
ewAClCkjAy+CKocjqLyakyUR8aKKVQKyKiiqRUEghBf49qYFmxV8xyWEgqPA
gmfJASrYqyqVT93dz3O2VGva5Hpj/+Ltu+f1ebCh32quOHeRvbLlVOcvwzt7
5e2NLeHWJbl8pXF99Rq4e2fr2r3ytV1lrZLbJG/sVNe2iubWDpIvXJzLCWh7
926ldH4T3zO3X6+9UVLu3Li3fvHG5QsPlue0O4aytrJ55c7qPUu4QK6Sonp3
/g1j+xrZwOCmsbsDbFcGr0sVfV0r7qxdrb1+ySVE390sV9fu3N1cLoSBSzMa
mra/1NuHi4GlqqKPYrFnhF3rAJEcG9SKQC2VlAC2IlURcxJWQUlUcpJSXLpX
2xPu1e5J1+8I5bv7YO+GLjlFeKNcXM5kumn3FmMEMNFmC2oRm18xS5aqm9ts
LS8T2/th2Kw1DFdnsDJf0/eIulC02NBne78gY6VUJCUAS4RoWTZy+CJmT3od
u2epx8bdLLlsHVqJA2JeFPNSlru8tjE9xUYC8svb9AQeB+q1OuuPeS+lzr1S
atgOcV6N8d4kxJCa56On7e/VJbjZ6akrxDCsJEcOKhDPG+Qd1yZs9ZBE5wFv
H7/9/Etnvn7l6z+ePv7CiW+/ffwC/em1E8ePCy+BF55/bu6znzrxC+DlLslx
4e3jHCX5RUoCfhnUhZfBS89/avPTz588sbkunASfY19ePMl4Q9cs29Sx8Dnw
U+zHF06+uI5N7ip1UoXPgs+wn6ZOPkc5TXeFV8Fp9v3lk9w6+85tEMflLhDb
1TV2c4Rbbrhl6o+7ydePP6tYzsn0PGT3cub7lze8ePPHfv0V7/yzv9oP7ny
37f5H3z4G+/8zre+89GP/u/z39bu7hS/+R//dOYP/lGaKv/dx98992dzH/9s
41/+QX3nbuGHl1/5UZH78UfL7u+i6d/6/H/+4T//2gefOf/Tmbfsb/zg3ff/
+ruZ//2r9//+Jz83v3Di4c33T//rRu3f/vg7+fcWtb/9my//8C+vfe1rP/Pu
9//945/nf/3mh2+RP/3NR8/FPTief1a9A09Yx6v/BQiC479EX/+n6z+NJf9L
NUkdm/sJx39CaZDd3bZ341dbiK0H8cdbd03+jpPQYlUlx5aJXApY0q1dlMT7
wAOclk5Om6Wn2SIE/WOY7S2RCGaS/Lc3EyabZyNZIap40LR42dBjiGTUD+mx
wL5FbPCoJ1RKwzJjbWbJGHL2J0H8p2f+Tw//QTad/20s+J9g/U/IsOjJ4kFe
agcFs8CMD3uG9UPBSyHZkgibqFMODBKUA6MkE/v2XpmEnizRyz/Oqp/MP+lT

KYxGNlTAtKhD4ljmWNf/gzCQ/5G9+g+Uyv9IWnv9b9r/3BOv7d1hJkylmK3s
rbPZ+RUFQjaPd3cl73QV78nR/9YuMZzSeOv/oRSSf1FO9f+I878WrmeKhNTz
8/NdoUd5BAA8mmFAuUilL+qBDp8P/UpfeniwQxHGbPqFv1Wju0oH8pcQ6DMC
oD0GU0lS+SWmcxLuNCy3iMe7/ocsSmH7X07X/5wg+9+b/2fNznBXMxyLxvta
NGr6LybAAQXFLHIpBMcBtVnScwMOGtunhI32pKteSkpeAhJIAAuJCz3hGFa+
PIocsU22cWPICeBk87/36n8xnf9zJG3I6//mZASFcgUiAeYOWtm2w4Y9MYC2
SFPM6LumL1P43eBkH9+/laF64Ilwnrth3TTJNdNqOhtl3VkvldkUZFYd7zQI
T+UAF1iBZkTKP5GmAe34IlZL7njlH4X9f5DmfyZN/g+S/ljZV5QcQvDAIb8d
7qsiUcKObSBNLe4uCcAvi+hWIxy0oDeSMkKOUiP2Qn0H9IXWED94tSP6yvSu
ePQJLSOlT4CtmzRe+Zey0fhfWv/5tMt/bpDo+4wXEv1wVvcIl/b7//aOZrdt
G3wv0HcgAuwSxClFSrbkzEWLYcPWrcnWrEsPBQrFomz5h3JIyWpy2noesHcZ
1uO2V0jeaPxE2ZVkW05S1E5WsUkdU6Ik09//7/MVpsZ1qH8P2vjdekilULni
07YAXqv/U3NB/ic1/t8R/F/egIVUGdioSaTnh6vN9nlbQYX1n4A2Tgb9eSA4
bdoONSGBVCmIDsQMU5IjDHPiModqQkM88Zypy0dnNOFixFkS+gNXOnGwKJtQ
rD7v128ngWCpUWIDY7VrorjDUEIcHTD9bB1iY+K0LKfZpMQ2rI8wY+qdGvW2
if9Nnf9BSBODopjK/7X9b7P+PwtTdMzENOgy9JK7UzcYuacjdiv0v7h46wXQ
sVsmi+UP1dJrU4WFuKqMLDBXegOfDZK5gYCk2aI2NVstQrEJiQRGPv+zSBgU
uPPQlQyHPiSb9Dk5C/3QhTfEZYkP1xLQlSQS542nfsSEuoXpUExsaqudgld4
C/dxbCXnnLiCp8G7hq1uG06ZiGLB0M53HP0owp5CWblzN30N0ptsPf+7ZS3E
/xBa6/93y/6fGvneLCoBq2snlGNDr1ESqLIYkAbVzDQIwUGeIgPewwdPu102
ibI4fh6eht753LZ3A6eAYVXnhBWKOtyVHkAf7QiQbBxsuf4Xtlo5+b+Jtf+/
WeP/JkY5//tAfchO/IWJc8h5c6KwEtVD0dPRAJSYpCoeYAaXS/0B9gfOPkP+
EgruFSYmw65sNcbBmO2BGK1HKU1pLzchgx5XM7lzFQN3e0wdUFfOvfGF26si
KXc/Hz1i8lO3/1pv/8vXf9H4T1q1/2/T/F9JxvyJUo3DcZClNxwhhVjlWY3b
wh1N+kWkf+6+avhH4sR94R0rCMfNvIMvhbMCMudNfa/OLy7OG0Sdpng2Uj9K
8Ec8Hp8yoVZypCME9Fm0Pefl6URBUVdPwVwtivthiE7VV6u+NnjqMIQ3In1Z
wtTt6/L0SPdJ7aSJG/ff/hdBAiJk4sot1n9otkr4b1m1/X87/P929X8/qu4v
ysHgMBj0TXfocfdsMEwY96QrfWD8K/j8tSoCry8DV1p0/NXPahWpWjVy5cDr
+2X7yA/HKKIV64hj45h78ImKj3n4/eHRySGKzIrFSt3BPmV8iVGm0v2BXSfw
fM7YEtEjEoncNv+3cBn/TYpr/N84/gs2DiPWUPvRWy70I1TZj30FxIOqjp28
L2DljYrmv1GoSEpDMBnGosuWERbHJA6289UOUngOg25MTceyIYSY+RBB3BfQ
0FysJSSE2k4+uHh5L5P/T8RwzIeyO/60FGCt/G/SMv43a/zfLP6X7GY6Jq4d
hRE0zdOhcV0N+JvIBsigktfZAJvA/zHZMv5TUsZ/i9T6/x3x/wOWB/KUt5Xs
SlsOsVt0zqX7UQQJQ0mS7C8T+xVilrE6hbVU/e+fT/qMuxHzGuUyjh+Yseng
vIN/kRuvLjhhXCcS8EtoDtZ+Fvb5m+NxEPUff27VImO+9fxfAxtl/59l0Lr+
20bGTLInGKMOIru7FO0ia3eXoMu/L/9BV79dvVMvl39dvr/88+oPmCtNXb27
+h010OW/V7/Ojl2+r7KJO3YJJw1KTIcUCUUGlNAKaAhMf0gHhImEeb58elyR
SrwyGNEwdLuRFVIL82LtgiQQnrDo8rteStCyMB7HXsgwvlmakGOvzBOCQ3ev
pWA97tFIZMCn243/MFqWUY7/UAJgTf837P+ZBlM27D3p9t1ANDwuugWlKev2
nfbmzgmHR6j98MH65egAIaQILDhTO4qAGTb8o5Q/fOAL6DSC2mjnGXoRSsZP
meih169f7+xk3ledbeoJOPhkISYS/oMrdxAU+Rp49hpPlIb4JU6oItVPHymN
9ejKtPIMxg7cKhdHkrWVRuiROjndDgSsTruN53LojNtRB8tM/JSwaYcs+cbl
vRHzvmWuByFmCHGW+HoOZb10EWjdAY/VzOLBrFW9Xn+ixNeXPJaxO/oFDrfR
wcHenvo9WMu7dMVVdbN0p6UOA5SFr38kZhCQqPt0dLP5WQd4HupO7Y/TNuyz
PUkNho8Q6OAw5CQAdleAiuzr0zsEkIFQJhQ4wXBgIwQ++PkW6yuqP0GeSDfY
aNr7xLL2FXeHa2UX6qhFs0hkiNBTHLC981McKlUDyUgEXIHXDoDXSsBSl1K7
rWG9MwsDUDPpp9TPLMEe4ulTDtCZena8T+mN2fvKGkDmZlj7BVPo529Z/yd4
0f5X6//3yP9XZenr9aVnGjkCnIHcAEx5I5+PSUhGJhVWjLW5r2Dao8So8Oxd

q8cTWBGCsu0A16U/6lGPenzm4z/w/fIfAPAAAA==
-- END MESSAGE ARCHIVE --

Authors' Addresses

Robert J. Sparks (editor)
Estacado Systems

Email: RjS@estacado.net

Alan Hawrylyshen
Ditech Communications Corp.
602 - 11 Ave SW
Suite 310
Calgary, Alberta T2R 1J8
Canada

Phone: +1 403 561 7313
Email: ahawrylyshen@ditechcom.com

Alan Johnston
Tello Corporation
999 Baker Way, Suite 250
San Mateo, CA 94404

Email: ajohnston@tello.com

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07052

Phone: +1 973 952 5000
Email: jdrosen@cisco.com
URI: http://www.jdrosen.net

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7042
Email: hgs@cs.columbia.edu
URI: http://www.cs.columbia.edu

Intellectual Property Statement

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.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.