

Network Working Group
Internet-Draft
Expires: March 31, 2004

R. Sparks, Ed.
dynamicsoft
A. Hawrylyshen
Jasomi Networks
A. Johnston
MCI
J. Rosenberg
dynamicsoft
H. Schulzrinne
Columbia University
October 2003

Session Initiation Protocol Torture Test Messages
draft-ietf-sipping-torture-tests-02

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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 March 31, 2004.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This informational document gives examples of Session Initiation Protocol (SIP) test messages designed to exercise and "torture" a parser.

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	5
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 <	12
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	S/MIME signed message	16
3.1.1.12	Unusual reason phrase	19
3.1.1.13	Empty reason phrase	20
3.1.2	Invalid messages	20
3.1.2.1	Extraneous header field separators	20
3.1.2.2	Content length larger than message	21
3.1.2.3	Negative Content-Length	22
3.1.2.4	Request scalar fields with overlarge values	22
3.1.2.5	Response scalar fields with overlarge values	23
3.1.2.6	Unterminated quoted string in display-name	24
3.1.2.7	<> enclosing Request-URI	25
3.1.2.8	Malformed SIP Request-URI (embedded LWS)	25
3.1.2.9	Multiple SP separating Request-Line elements	26
3.1.2.10	SP characters at end of Request-Line	27
3.1.2.11	Escaped headers in SIP Request-URI	27
3.1.2.12	Invalid timezone in Date header field	28
3.1.2.13	Failure to enclose name-addr URI in <>	29
3.1.2.14	Spaces within addr-spec	29
3.1.2.15	Non-token characters in display-name	30
3.1.2.16	Unknown protocol version	30
3.1.2.17	Start line and CSeq method mismatch	31
3.1.2.18	Unknown Method with CSeq method mismatch	31
3.1.2.19	Overlarge response code	32
3.2	Transaction layer semantics	32
3.2.1	Missing transaction identifier	32
3.3	Application layer semantics	33
3.3.1	Missing Required Header Fields	33
3.3.2	Request-URI with unknown scheme	33

3.3.3	Request-URI with known but atypical scheme	34
3.3.4	Unknown URI schemes in header fields	34
3.3.5	Proxy-Require and Require	35
3.3.6	Unknown Content-Type	35
3.3.7	Unknown authorization scheme	36
3.3.8	Multiple values in single value required fields	37
3.3.9	Multiple Content-Length values	37
3.3.10	200 OK Response with broadcast Via header field value .	38
3.3.11	Max-Forwards of zero	39
3.3.12	REGISTER with a contact header parameter	39
3.3.13	REGISTER with a url parameter	40
3.3.14	REGISTER with a url escaped header	40
3.3.15	Unacceptable Accept offering	41
3.4	Backward compatibility	42
3.4.1	INVITE with RFC2543 syntax	42
4.	Security Considerations	42
5.	Open Issues and Remaining Work	43
6.	Acknowledgments	43
	Informative References	44
	Authors' Addresses	44
A.	Bit-exact archive of each test message	45
A.1	Encoded Reference Messages	46
	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 "abcbabcabc" 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 or short and long form for the same header field name
- 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: 0ha0isndaksdj@192.0.2.1
Content-Length : 151
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 492170 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: word%ZK-!.*_+'@word`~)(><:\/"[?]{
CSeq: 139122385 !interesting-Method0123456789_+`.%.indeed'~
Max-Forwards: 255
<allOneLine>
extensionHeader-!.%*_+`'~.=
<hex>EFBBBF5A4A7E5819CE99BBB</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: 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: 151
```

```
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 492170 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 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: 39203ndfvkjdasfkq3w4otrQ0adsfdfnvd
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.
%lt;sip:alias2@host2.example.com%gt; 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: 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: 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</count>value;
  longparam<repeat count=25>name</repeat>=shortvalue;
  very<repeat count=25>long</count>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</count>=
unknownheaderparam<repeat count=15>value</count>
;unknownValueless<repeat count=10>paramname</count>
</allOneLine>
Call-ID: 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: 151
```

```
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 492170 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 contained extra octets after the body (which in this case has zero length). Those 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: 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: 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: 151
```

```
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 492170 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: 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: nfc9ehfdfaekijh4akdnaqjkwendsasfdj
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](#) S/MIME signed message

This is a signed INVITE request. The signature is binary encoded. The body 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.

Message Details : smime01

```
INVITE sip:receiver@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.org;branch=z9hG4bK923rnasdkl3
To: <sip:receiver@example.com>
From: <sip:sender@example.org>;tag=2390234seiu3
Call-ID: afnkjeriuoqeiuavnklafejkj34iu43uawe
CSeq: 282398492 INVITE
Max-Forwards: 70
Contact: <sip:sender@host5.example.org>
Content-Length: 3134
Content-Type: multipart/signed;
               protocol="application/pkcs-7-signature";
               micalg=sha1;
               boundary="----34CF59C076641DD0879594444157C7EB"
```


-----34CF59C076641DD0879594444157C7EB

Content-Type: message/sip

INVITE sip:receiver@example.com SIP/2.0
Via: SIP/2.0/UDP host5.example.org;branch=z9hG4bK923rnasdkl3
To: <sip:receiver@example.com>
From: <sip:sender@example.org>;tag=2390234seiu3
Call-ID: afnkjeriuoqeiuvnklafejkq34iu43uawe
CSeq: 282398492 INVITE
Max-Forwards: 70
Contact: <sip:sender@host5.example.org>
Accept: application/sdp, application/pkcs7-mime,
 multipart/mixed, multipart/signed,
 message/sip, message/sipfrag
Content-Type: application/sdp
Content-Length: 149

v=0
o=sender 29739 7272939 IN IP4 192.0.2.1
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 492170 RTP/AVP 0 12
m=video 3227 RTP/AVP 31
a=rtpmap:31 LPC

-----34CF59C076641DD0879594444157C7EB

Content-Type: application/pkcs-7-signature; name="smime.p7s"
Content-Transfer-Encoding: binary
Content-Disposition: attachment; filename="smime.p7s"

<hex>3082088806092A86
4886F70D010702A082087930820875020101310B300906052B0E03021A050030
0B06092A864886F70D010701A082067A30820339308202A2A003020102020800
90008902240001300D06092A864886F70D01010505003070310B300906035504
0613025553311330110603550408130A43616C69666F726E69613111300F0603
550407130853616E4A6F7365310E300C060355040A1305736970697431293027
060355040B135369706974546573744365727469666963617465417574686F72
697479301E170D3033313032313134343332355A170D31333130313831343433
32355A3062310B3009060355040613025553311330110603550408130A43616C
69666F726E69613111300F0603550407130853616E4A6F7365310E300C060355
040A13057369706974311B30190603550403141273656E646572406578616D70
6C652E6F726730819F300D06092A864886F70D0101050003818D0030818902
818100CB8302060F12C8FA2D1786922CA173DCEB80BF1B1B8AF74A310C6975A5
56A7630FB6E044D9E994DCD49AFF7976C462D7A8E74ECBF98723AEBF2796EDDD
6263577C6C2B77DC7C300B533DEDB5FB8EB3827FD6FC9B37B9A0DE829F1B1081
D632A8AD9FB00A860928E88F87E0B979BA65294AC7D6D2D18A78C86B4FA73387
4E230203010001A381E93081E6301D0603551D1104163014811273656E646572

406578616D706C652E6F726730090603551D1304023000301D0603551D0E0416
041440FF1C0C1BB8684CA917839D70E97DF8DD5B60D130819A0603551D230481
9230818F80146B461714EA94762580546E1354DAA1E35414A1B6A174A4723070
310B3009060355040613025553311330110603550408130A43616C69666F726E
69613111300F0603550407130853616E4A6F7365310E300C060355040A130573
6970697431293027060355040B13536970697454657374436572746966696361
7465417574686F72697479820100300D06092A864886F70D0101050500038181
006FFE1A3B5CE807C3DD2CFDF6E9787F491C84DBF7DCD11DB2D6A8887D2FE3F2
2E9C6894994282E50AA0DFFE1CBD4EC2C20217831FC2AD360FF1C0DE1DE1E870
102CFA99EE504C7DC0D8752A63294AC748DDDEFAD55C6D051F1CD54CFE7C153
278962A53CEF61B875C1FD3C74E972242CBA0131B3B8C607BF95B378212CA9A7
5E30820339308202A2A00302010202080090008902240001300D06092A864886
F70D01010505003070310B300906035504061302555331133011060355040813
0A43616C69666F726E69613111300F0603550407130853616E4A6F7365310E30
0C060355040A1305736970697431293027060355040B13536970697454657374
4365727469666963617465417574686F72697479301E170D3033313032313134
343332355A170D3133313031383134343332355A3062310B3009060355040613
025553311330110603550408130A43616C69666F726E69613111300F06035504
07130853616E4A6F7365310E300C060355040A13057369706974311B30190603
550403141273656E646572406578616D706C652E6F726730819F300D06092A86
4886F70D0101050003818D0030818902818100CB8302060F12C8FA2D178692
2CA173DCEB80BF1B1B8AF74A310C6975A556A7630FB6E044D9E994DCD49AFF79
76C462D7A8E74ECBF98723AEBF2796EDDD6263577C6C2B77DC7C300B533DEDB5
FB8EB3827FD6FC9B37B9A0DE829F1B1081D632A8AD9FB00A860928E88F87E0B9
79BA65294AC7D6D2D18A78C86B4FA733874E230203010001A381E93081E6301D
0603551D1104163014811273656E646572406578616D706C652E6F7267300906
03551D1304023000301D0603551D0E0416041440FF1C0C1BB8684CA917839D70
E97DF8DD5B60D130819A0603551D2304819230818F80146B461714EA94762580
546E1354DAA1E35414A1B6A174A4723070310B30090603550406130255533113
30110603550408130A43616C69666F726E69613111300F060355040713085361
6E4A6F7365310E300C060355040A1305736970697431293027060355040B1353
69706974546573744365727469666963617465417574686F7269747982010030
0D06092A864886F70D0101050500038181006FFE1A3B5CE807C3DD2CFDF6E978
7F491C84DBF7DCD11DB2D6A8887D2FE3F22E9C6894994282E50AA0DFFE1CBD4E
C2C20217831FC2AD360FF1C0DE1DE1E870102CFA99EE504C7DC0D8752A63294A
C748DDDEFAD55C6D051F1CD54CFE7C153278962A53CEF61B875C1FD3C74E972
242CBA0131B3B8C607BF95B378212CA9A75E318201D6308201D2020101307C30
70310B3009060355040613025553311330110603550408130A43616C69666F72
6E69613111300F0603550407130853616E4A6F7365310E300C060355040A1305
736970697431293027060355040B135369706974546573744365727469666963
617465417574686F7269747902080090008902240001300906052B0E03021A05
00A081B1301806092A864886F70D010903310B06092A864886F70D010701301C
06092A864886F70D010905310F170D3033313032323135323930325A30230609
2A864886F70D010904311604144A2FD5856B6006413209FA56A0C1D85179DBCB
5F305206092A864886F70D01090F31453043300A06082A864886F70D0307300E
06082A864886F70D030202020080300D06082A864886F70D0302020140300706
052B0E030207300D06082A864886F70D0302020128300D06092A864886F70D01
01010500048180C1C3193CF4A8BE1278B5529ACFA1C51DDEDECF0D3DC4C18FC5


```

9A5B120E6D559F4953A3C3C7C97B4EAD8388F1508F7AD2FC71CC7B9ED2844789
60A3ECF87984E25A15B4AB63F150C30570B6315A2327E381EE11E866DC1405DA
29E74CC20201816F1516DD893332D9A8E26FBAEC237C494F3EFAEF4EBCD2122C
DE7D57DECD</hex>-----34CF59C076641DD0879594444157C7EB--

```

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: 0384840201234ksdfak3j2erwedfsASdf
CSeq: 35 INVITE
From: sip:user@example.com;tag=11141343
To: sip:user@example.edu;tag=2229
Content-Length: 159
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 192.0.2.198/127
t=0 0
m=audio 492170 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: adsn2309jasndj203insdf99223ndf
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 and header field of this request contains contain 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: 0ha0isndaksdjasdf3234nas
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;;;,
Contact: "Joe" <sip:joe@example.org>;;;
Content-Length: 153
```


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 492170 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 length of the body.

When sent UDP (as this message ostensibly was), the receiving element should respond with a 400 Bad Request error. If this message were received over a stream-based transport such as TCP, there's not much you can 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: 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 492170 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: 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 492170 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: asdnw3qjr23o0pd9vanlq3wnrlnewofjas9ui32
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 : scalar1g


```
SIP/2.0 503 Service Unavailable
Via: SIP/2.0.TCP host129.example.com;branch=z0hG4bKzzxdiwo34sw
To: <sip:user@example.com>
From: <sip:other@example.net>;tag=2easdjfejw
CSeq: 9292394834772304023312 OPTIONS
Call-ID: nvoao34irnoase0of0234hn2qofoaf0232aewf2394r
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: 0ha0isndaksdj
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: 153
```

```
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 492170 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: 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: 160
```

```
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 492170 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: asdfasdoijweoi2323-asdfwern23-asd8ia0sw34rk423
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: 160
```

```
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 492170 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: 2304u0qwsdfknq234oi243099adsdfnawe3@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: 151
```

```
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 492170 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: afewroicu34958239neffasdhr2345r
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: 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: 151
```

```
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 492170 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: 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: 151
```

```
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 492170 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: 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: 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;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: 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: 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: 0ha0isndaksdj0234sxdf13
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, and a CSeq method tag which does not match it.

Any element receiving this response will 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: 0ha0isndaksdj0234sxdf13
CSeq: 8 INVITE
Contact: <sip:caller@host.example.net>
Via: SIP/2.0/UDP host.example.net;branch=z9hG4bKkdjuw
Content-Type: application/sdp
l: 139
```

```
v=0
o=mhandley 29739 7272939 IN IP4 192.0.2.1
c=IN IP4 192.0.2.1
m=audio 492170 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: 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: 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=z9hG4bKkdjuw
Content-Type: application/sdp
l: 153
```

```
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 492170 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: 2340923nasdfasser0q239nwsdfasdkl34
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: 2340923nasdfasser0q239nwsdfasdkl34
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: 0ha0isndaksdj@hyphenated-host.example.com
CSeq: 234902 REGISTER
Max-Forward: 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-Forward: 6
Call-ID: 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: 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: 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: 98asdh@192.0.2.1
CSeq: 59 INVITE
Call-ID: 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: 155
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 492170 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: 234asdfhn2323orihawfdoa3o4r52o3irsdf
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: 0384840201234ksdfak3j2erwedfsASdf
CSeq: 35 INVITE
From: sip:user@example.com;tag=11141343
To: sip:user@example.edu;tag=2229
Content-Length: 159
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 192.0.2.198/127
t=0 0
m=audio 492170 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: 2304sadjfasdlfnm2o2l43r5u0asdfas
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 part of the 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: 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: 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: 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 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: 0ha0isndaksdj9342dasdd
Accept: text/nobodyKnowsThis
CSeq: 8 INVITE
Via: SIP/2.0/UDP 192.0.2.15;branch=z9hG4bKkdjuw
Content-Length: 151
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 492170 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.

- o There is no branch parameter at all on the Via header field value
- o There is no From tag
- o There is no explicit Content-Length (The body is assumed to be all octets in the datagram after the null-line)
- o 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@iftg.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: 1717@iftg.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 492170 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. Open Issues and Remaining Work

1. All of the messages in this document should be considered new. They are either new additions or major revisions of the previous versions. They all need to be carefully reviewed by the working group
2. Are the header field values in [Section 3.1.1.7](#) long enough? Should we push each field over 256 octets or even longer? Where is the threshold of reason?
3. Is this really possible to recover from embedded whitespace in a SIP Request-URI as suggested in [Section 3.1.2.8](#)?
4. Can we modify the example in [Section 3.1.2.15](#) such that it is not obvious where to infer quotes?
5. Can we modify the example in [Section 3.1.2.13](#) such that it is not obvious (due to ambiguity perhaps) where to infer angle brackets?
6. Is the message at [Section 3.3.11](#) sufficiently tortuous to include in this document?

6. Acknowledgments

The authors wish to thank the following individuals for their participation in the review of earlier versions of this document: 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 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 WorldCom 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 WorldCom, 3Com, Cisco, Lucent and Nortel.

Informative References

- [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.

Authors' Addresses

Robert J. Sparks (editor)
dynamicsoft
5100 Tennyson Parkway
Suite 1200
Plano, TX 75024

EMail: rsparks@dynamicsoft.com

Alan Hawrylyshen
Jasomi Networks
2033 Gateway Place
Suite 500
San Jose, CA 95110

EMail: alan@jasomi.com

Alan Johnston
MCI
100 South 4th Street
St. Louis, MO 63102

EMail: alan.johnston@mci.com

Jonathan Rosenberg
dynamicsoft
600 Lanidex Plaza
Parsippany, NJ 07052

Phone: +1 973 952 5000
EMail: jdrosen@dynamicsoft.com
URI: <http://www.jdrosen.net>

Henning Schulzrinne
Columbia University
Department of Computer Science
1214 Amsterdam Ave.
New York, NY 10027
USA

EMail: schulzrinne@cs.columbia.edu

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 -xzf -").

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

H4sIAKxjuj8CA+xdy3PbSHq3PZnHaktJbc0tJ1g1XM9YItXobrwoUSOPrdnR
20NxwbKnarMpT4tokCDBBgWAouVUnMSHrewl1XtymEq1KtfnLamtSv4Fzx+x
h5xyzDXd4AsEQBKsXyc9aJsiCXx4dB0/r793nxCT+G1aLZkkuDanBmQAVBVf
A7xpsXfRFFm7BjRNUbEqqwrg9EhF4JoEri2gdfyAeJJ0zWv4beI1/Ul0s/a/
pe3bR8eH3z48kny7Xe741Nunz0mr7dCS69Wko8NH27AE1tee2qQ8+Lb95N4j
qe76AS4NaKtuq6wAFeyceIRV65UXRv1X+OR+02x0MIIIr699Q54Xv3S9LvFM
vyxp/JRfevwgaeML6jhb0h2Hn4qZ1NuQdsWdkFKtdML37EeusLcTkFoFI4Qh
Wl87dvnB35HAD9mWdFx3W8Tnx4bdCErdcPtYV/bW1+4Sxyke3itLvmkBf1cM
MtMHBbHBG6I6qjPWbCComk1G+QXuVKu0HZQ10m47dpUEtsu2fbPNT3NET8sS
MiDvmSH1x299zSlLvFdv3e9/QszejjZHBjAL/0jDcfxrGsrXv1T8c8SN8C/Q
lkYxgLHYV+XwGucfIWARghjGGQBKYSmywT+VYEm0cZFPUBwimpqus1yIbAxB
4wwQH3BmYqEGYgPA6nGovoVgnQ/+OfDpPKf/WfiXgaze8Y8xzvG/iHb480nh
8cHVw5/RIIQ/5B0tISbspAQwRC+fR/mc3mIcxRyyEDkNDItFn5oWa4XSQw/B
Mp9zDYS13i1PEEqiMkmKONJdX7vHH/ey9KVnb0lAlr4mTIL8EZRktQwA/y8d
HB3zK7osIFX0c3YjFRPnV0qRHu71CCkLisfnbZrGofq7H1BWC+q8D4osOM9Z
hfffrbTqX0hx6LKEDY3LEhrUeAcN3kHp8BEEmkRlfc2vFNfXqpXkjQACBC9r
VUjHtF2JD7WsAenx8aPt008fSbxXU0w8s03qSghCbbgL8fsgFS9ot0i7jGTP
wa070UP8ifJ/Nlfun0H/Awn+L6Nc/lu4/Jeq0I3rgKJFWW41iw7YjXP/sI2U
QN5iauBsLbA3KYVtXAvsHZrWlagKiGQsK/tjdz9T5ePtXdh6xvBvszMgL9P+
o0GQkP9UNcf/ysl/jdIlJED+Y8qqzH9aiNWEGBiBJKgTYPvMJE3fbBCfi4LC
RkP8kQI3UfAbqo7Kzs7WztZWRH7b+Nq1faNSw6Vj7GCHtzQJDC2U6i4owMkT
JTh56SIcx/8Z9fxl6n8AIzm0fwVo0f5XZv7XEvo/ljb/Z536x2b+06Vw8p81
4592qWuftmlk3j8u9af+7F0+Fp/yhz07/A7N6RfCf5VwpXmp8j/fixLyvyLn
+F9E60+jEgRA+vb+tLnV0GP4lhHsmXYSx0BFKUve8eN8ckI9ceBo7kc61jGA
Q0ZzPp//LdJEDU9LjUt/86RaQ3dLspQChgJHnGhpCd2yBzvaKaoJMio2enb
p6CRJgQYmUw7IxNReHJhIIJ6lMPsXVxYMPSJ0oKhb8tQy60+ebs6/k+fB/PV
/jLYf9Qk/9dgzv8XLf9lUACfXcYFgCENvDFBsCypkzS/9bXH9LRje5z1Mjeo
26x21Gm3XS/wj+u2v5W28cA06pRf4JHnPg8vRg4XG2zq92kHx3v33DvsPLkv
xVM4NrMdPziSLNcxi5yNF/1TE04js6Tf+MyapLliV2ruuZS/X9AAVH/vxza
f5Rc/1uo/IehqQ1VQ0CWtmgQUE8ATTrxKGLKHF2SR6vUPhMomywhgrikB5Gh
c1i7NAJ24puuhToNCBD/yCzEr9zhn+oW8RGth5Yf4hv2S0JD2MAxqW83jV31
FEVkeOZbd1/u200T/HqEBoANZrImRakonSDe8U4m5Lu3+fevco7tLdn/oylK
HP8Q5PP/6th/x004+ptahA3h7+d/XDjJ0d5hNe887mlPFR61CacX5hNlMFO
P07AT87ZReHfWUWIFIgu7No3eLuEaXiKbXgRxuEqf6hJa7n+H6yBxPyv5faf
hbTHB786PD0+eBziNRX8CRT5J0h47FK035H1V1wuZq8dWG7uNa2ntUbTC77p
Eurdvz8w+qYfEmEMGtBksJ+8vQFngNKgtXg+I866KRsaVBQF8rZf68pRzrPT
YU3mdlMik3cvdrCPf7hU/CtqEv+5//enin8d6XyqvCLMozTM714I9HvvcSsw
eeJ49HSp9j8RIBDHP0J5/0+K4D+bqJ/c2w/UgorcQA0EE01A+vraYUyYNwyh
/FvMQU4T8kdAhrGJun+VeKDvSPQfgX2ymQIqaawqjFN0tQSIfxE1KRpGMjZQ
g3sc358lQgbzBz5tfJIqj6Fx4GBD4EfZj0aRXCRGJqX3SAe9S0v5hzNfQTSM
lFnjyb1bMxr1q/N2/8z2/yMY4/8wz/9avP2Hv/wCuiM4bAGDCAuLMqzkvFDQ
FJ8WNJjG9EJB67AA0f86o/yND2GV+thZ9W2xSE+wP219ze7lhgAjNAo3G00G
kcv4F1BEkcQQKFJccVazj+C66a6aKU7+9TVpaKEqqHfTkkF2+HY+DDusoMot
Wjnjb06Hd1opIFxA8t4Vcc+JzDMPC8jb5fg/XLL9X1UT8r8ICcv5/0Lk/wLW

uMhcwEpPB/BozfYDj3ilidrARuHXnLwfVe1R3+141TGZd2+Y3j+TMmT/FkSw
QUyrMRYUJpg+O+26CHunENkINzzhHawyByLm0U3mi7iwcRf93ZlZgBAYBeuF
z6gTnUEMhLEUHYvUhmUX+wFxb0LLPadgzCV4t6B+mSCFvfCw0GnynCgkRDHC
0Zkg0P5Zx3GW6/9HWhz/Csrtfyum/4vHpFgAoCg+TNJwJxL1LXuGAXv106ak
AyMDAsRM66zZMIlvNU9RF7uBdwqI6VucJZAzc5gSjJGhC9lvit4/y+kXCdQZ
AZF3YT9x4F4K1STCB+W3w2DI8e91PHu5+f8p+Me5/2/R+l/cgPf5Y7cT0EoB
3Y3xhgI6uJLaAAhzDUhPywmb1H0f8o9JcQGhqlgUUK09USQNik68IucNp8DQ
u0WldKlXh6fF0mbKiH1wVs0nfjhddyy6VjGo02Kdiy/ZI/7maULLlcC8XU2z
md+xlqz/4ZT4Lzn3/yyb/494fJ9bhnFbsz0MhnIpxuj0c28vyAuNie4EI6+P
kgX/QYsG9aXKf7ImJ/W/HP8LaTf5A0A96gc2qxW/4U+Ca4ZJeIqq6caz25vf
lwo2Myk1b70MmYT8rMM6foc4pSePD19+GrjFE3rT73j0s1/aPrsVbNrBJ9tV
j7w4/3xre2fndvmXdrBZJ35F3upS2zNv3m4T/50u+9J8Frhu6VPTpeKwoh18
tj87/GRg35GnBXGwbhZU9fazze9vvezbq744eFD+zYfSwyf87Zp0T3z7m75R
6qL9STVzBW6TMvn1970P8Nbms94ndLtws1TsXajVYSZhMe0XxQ9/JGJMbt16
uXn7GScuVDZ+/NvXv3/976//68dXr3//4z/8+LvX//H6PzdCgflZeNaX8q3N
7/mZi6WI8Nt1PbPw6/vFmyXe81v74uv3Lz/7dG+3/Jvtjb/8i8//+q8iZbRk
CJGuSBf47eNC0hTBqfQ55+c+Z+BfUWJSj1+7cHvzGR/2yv/+4b//+K//9se/
++f/+ac/TPCp56LXivD/M6hgtFT7n6qqyfj/3P63aPnvCZf/vtjPFgNoW0Gt
W5rk8d2UEUCKosi87XPSUtWx0QMjy7DjV6u06y6jeyLlr8oZVjE00PTpkbib
nRYxTa/i+3Fr+8ACsSmSRwcBfVGy8ctFmCYXPLS0+4uEFCnqUOC92qIwb1DV
7yqZJ8d/J1i2/w8k5D8Ic/yvkP6XtRrmG6UF6aHhH0/1DYwXiTJ9y26IQt7I
HMWCKBhPCYPz3RbNvid0Mtz7wcGW67VIkBRvcCjf7IbY5Wmi7barrY4ksowr
GxijjW2+cXd7sHvJv7/jstq8A4Bn4F+FY/G/sKf/5fm/q4P/UIk6lEQql0Qk
QScx0qKSa0lcB/Boi77hm9T2XAEQjq6N1NzessoflLGofLlyRr3zXY+2KQmk
qtthQWCPfE8726HX/f0iN0h02JLeIS45Xm/Kn6dd6N34ZT7U8buL9T8KFfB
Hvk37+yg/tB9Ko4UemWPU5IwecGVM+dcHNZjm+Iq89ycjE6Ghr6K/zHEgwei
Huqf4Y+ceCBEnwYDXunRTySXld4zM6AfnD/8VRzq+zFysDd8sAaHROyqLmt6
lH85X/zfwU9qd+dFpAs/ORh0jE96PSS+4JRX+SqKZ7ks9QeuKG7kKl/FHrTm
d/72AJPFkMNVFnGlsE8pli70ARAaV0r0khRwn0JpkkK0UaRdB8Rp7iRooBGj
OUyeZ7z6laBJOY8W61PK/UA1TpNyHiVGk3Y/OE6TcP7YGD+VylKCJjbKZ2k0
8XGwp0S1QPw8SRrZSPxeKUR6kiiFSpv908tqBholAw30QIMY0MAMNBkeaRnM
pjFmk+izSTIMcoYxzjDEGUY4wwBnGd/pJAKfql8dxqwMjRgDnUoIQVfN0pf1
6nWpz61nhocuQXS7xAIVTh6B8pNuTlAL5h0A0NP+N7b+V6j/a0q+/t8i9f/0
glpXE+VnQEN0+A+jgf7yfsT23XdSzg4zUea4So8Krtiej2RsIKhrYNVYo9P1
TdtvL9X/h7Sk/U/J838W0q5m/b+NHvY2d1P4wKgy34y4fxF6QE6q5r7Fde7z
NGecCiaU5hR8IXFUu1k/X/wvjv/lx/9jkMC/mtv/l23/35EcbzYL6EH7uUWL
Bp/NdB0VXxhwlqrAqFefPeHvU2IpgKJgqN0Qyz+ItZfC+P4u13d6H3WbAL/L
EPaaYVpR3wkoI6ABFaMM0aryx0JFEBnTZA58TLGdK1rZWWMvK00/xcMMFjq
+l8aSup/Of9fJP9P1/8mVoB9I4VQ12U8lrYFcAeccj3EarJTLgNyRo8RMAxi
igRw0qVoP0UwLHUD6fqMqh/yLhkQGYNkr3Ph8atvWwtVnWXX/wGKpsbxD+Q8
/30F9L8L19QRBRUgysYmXLF6Q5rZSKQbjTEKHCZ3MiEHup5dJ13LdAlysadA
F9meP1onSFa4KDRSFhPATGiikKhSKXHAB/R5sN12iM0ErI/5bdNbvsRcquv0
pcCVhKeYD01XahF2LrnVgAa+RDwq+WKVCZ+aguiESuLA0moooS3bb5GgWp8n
E5hp/1Hi9l/Mt+X4Xz37z6Xi0hFwoJLA26SgTlGH0H9uWk7Wgu65yedK8A+X
h38NWct+c/vvQtrDg+++0Tj+6tt7bwcHmCipxwT16ZxicgnA2c5yZFxGpE+T
5lfcQtPq0IE9Zw1gZv6vmvD/IjVf/2Up9l806DYXXjPmf0BlFu4GzztUstaH
7+ddjSA/YBKGLpaJ2o8gqE9qzKSFU5jU0IYtzNGiBPFknWwWJ6GxZLF3h5Wv
RH5ttpicaaZnpMaGPWFbvyCHgh091/DKiyjcktgqNVZ15mz9ySD/4bj9F6o4

1/+W7f+bZP2FCn5TeTA07SA8SQKEcmg+8hQMu2gYEwQyxAShLMX9xu9KQW9m
5C1eZt0vUehmQrgQWmgEJXM9SnyXLXX9HwgT/h+Q+38W0gbrf/KfSHRztT1N
n0EEjAbhSBYrfNqiRq9hQCjjeU5cwz1fzX0J8797Rh2/utT4P87qk/m/+fy/
WPuvS9qlE0rb5e3tEehRGQEArYMMWE9E+qBx5w7gGhLrRfzw04NTzmFYtxcA
1HRGtbpRr4TwpEW6+xWijCyeX7z663PPu51230CELLf+t4YT/l8F5/bf1ZH/
w/z/b7yS9HVJEnV5erNoUvTfy7T4b8paV/LEGh/TgvyNu0CedQ0sxSgrQAEZ
2ELmQA+0jHwWrKAF8GiNdIilr/+Lo/W/UYh/mON/IW306//pGoJyvQGRDPVM
C9z1MR8imbOKiSv6iXl+ZK0coPbf6QR117NfhMgtSw/dbxm9z9yuf1y3/aNq
XVQecdvktEMl/viTMC4jAe5323XM8X9CzGqw3PV/QRz/Ks7z/1YM/9PQn4p9
w9ARglNTfpoIK8T3HGSZJ2f7MiiJf/IoCXDacp5IKck6p0bihaYsQTy2guj0
RQ74qzS+0MEgeuQdxr9YAmS5+MdKHP+5/e/tx79+EejHnbpXuLLPNxMsjL0g
/xat4nP55n0FinjzXQJwpv6PcEL+B3n894rgP70A05xmYEMY+qblTjbbR20F
U6z/UGjjsFEfBoIhVtCQFgkkGpYnSXAwghE0Ec0bZF102vAgckHbNM4Ic05R
l3k0o13XahDf6NhJ2QQB3t+D523bo6FRYgFtsmtifIRFIVFph/burQJ1AA1N
MVQVQV1W3qdEeg//Tm2Z+FeUBP7VPP5rsf4/BSDpiHpndpVKTxg5I7ZDTpxY
VbrSLPiDEP4vXjw3bbFup9/NzAASIVV9Dka5lBsWbXSHRgEYJoboCGuaSB0D
ECE5muox4AHszCX8HmyPucSnwLVEXGmdwVPXcon4AgntWuJcniHhHnJnxTtW
QD1+CwwgAHWk80ER7+KruI6hc5Hm0+Ixm9VEAhq/rHtGvaDjUwnjkeMPLfG
0elvvEVLHvhme+n5XyLZM4Z/kNf/XjH7f2jke5ZUAibnTsZjQz0UBJhaDGDM
RigChEzOG8z1tTvVKm0H/ews5p645vnQyHcBp4A8Y9mqsaT0lVkl4E0dAT5t
2Uuu/wEUNsn/5/W/FtPi+v87vJOVTgGDCDgvzhQmQt31ar1oAAQxnBYPk0oQ
0Eft/AD0Mehtjw1oN6u+VmzZLbolx0deCzMeeC+D7Zb9nJpbkQ2+XWN8S4SW
z+akRvk0fubIF8sjtWms503JQ/PF6Cw3/0MDyfkfKbn8v+j5v1/E9XIp4ALY
4w89F9M91ovdiWgBaRcZ1wR8ysxxjje0BoSZYdTuRBkFsVizQT27457yPeSM
NR1i0WbjlIv+HYw6pEuHZYF0IcPzGXQoDMwq4tq/l0RX95JiAZKTcQNxrIz
4iy91vbcwK26TmUjzraKwLEcQ4RysZE4rsVJnVrFrXM5se/EFsvNeeevjSJv
CN/9UjHucrypWL53D+iaoRiYN1nR7moHX2wIhlQsZiFN9G7ED8VJ8ifpkk/S
KkxlF413wdGQ917fVqWsyWUf52kI3AnX06lshLNLqa1FVfxjT4QoU694wKqu
GZoGTmzGATgiuWf7bde3e1EIJOCPRB3Ft+9Ilu3Q5JnX18Crj/7+g5/d/u1X
v/2/9esf3vjh1UfnfFPnxvXr8s/Bzz54f/NP37vx5+9fAz8fEV3/4dUHL8Cr
9wzw6sa//PDejes3bnx07R+v/e7GJ9eug/Uh3fX3+WHT8CzVPfmTDz6+8eRI
/hj8Qnz56G0BBdtyPWT+Rfgz8TGDz/+6Iiwr12f/n971wLUXnGGkQBTE2MS
TExTTLuFdhxTBHsPGZ0wGVKnSUXdYVvYZBxP40N3QofEHb47IePYE4SdOM6k
ZSbjdCbuw2bcjMl03Gnj20N4mjjYeXXcjBHT0jyApvG0Nm0e406nSWke3b0T
IAkJsNuIib0fD+l2b0+rf/ff3f/f//+XWgwX4ZTcJdmo20g6tQIux9c3LHHj
y0ZR01eLqi55MRHFqOmF3gm/WpQHGRqSFMwyDG3fXJRHoUvKEb2Ezf+XyhTD
W/B1ZmHBDk6H4Y0xFLBkZ2SGf5ABw/us4XDGuD3WBfkFr47bih56rKJXG/57
1+ni4kc+rKcWScEnNx3x5J94+443xvYP//7AZ50dLzb/se+i69y/95b98vTy
H7030uxp2hn4Vm4J7zBXfve8f/88Fj3A0Mf/7j62U0j3QeLbwwP0X1HD/4q
96GFt17q2fv2s52nxBX1rwwNRh7Z/qq/4Qiz11VmzbRkWA6Hx2D4r7AEfYOS
m7KWwsJwsm+BiVSyJMSKM8w7F2ctzSqs+2zZouKTvrVPFe3+afvYro9G7t0a
geED0L8sK/wYDPd0FfrvLCr82/60b3Y1yksa3+y90FjYe6JX/7ma1n7Qbc1I
6IaZiPrKp1+p2XIp5+xIxSf/Gtv+wJple976cDhS8vRQ3807qi78o/Invv1P
fLv7L7mH/vTpsudcZ85Yi3Z/7czRlfmXXxgt+f0l9hsrxp94f93aXS+8Hiz3
INLePTI6Prx5YH1l19rPH+x3718X/OTqz7gTwb7P1mlj6nfmqDhloY6dfDnn
90PHtn+94qkj3ycMQxiGMMwVMAzVbRmC3ZZBNBxBnWmlRwx3Ts2Ah8K/hl+e
JNLCTCp2RoTLpnKyqfwoW9GUnUbrv82wbCo3i0I9o77qDw/6ty64nV44vulQ
/+vr0986dx/cMHVTPvUduBrmLviSmZCZAXdPvrDarRldqLmmri11MMesZk58

- - END MESSAGE ARCHIVE - -

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in [BCP-11](#). Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgment

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