

Session Initiation Protocol
Internet-Draft
Expires: April 25, 2005

E. Burger, Ed.
Brooktrout Technology, Inc.
October 25, 2004

A Mechanism for Content Indirection in Session Initiation Protocol
(SIP) Messages
draft-ietf-sip-content-indirect-mech-05

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of [section 3 of RFC 3667](#). 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 become aware will be disclosed, in accordance with [RFC 3668](#).

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/lid-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 25, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004).

Abstract

This document defines an extension to the URL MIME External-Body Access-Type to satisfy the content indirection requirements for SIP. These extensions are aimed at allowing any MIME part in a SIP message to be referred to indirectly via a URI.

Table of Contents

| | | |
|----------------------|---|--------------------|
| 1. | Terminology | 3 |
| 2. | Introduction | 3 |
| 3. | Example Use Cases | 4 |
| 3.1 | Presence Notification | 4 |
| 3.2 | Document Sharing | 5 |
| 4. | Requirements | 6 |
| 5. | Application of RFC2017 to the Content Indirection Problem . | 7 |
| 5.1 | Specifying support for content indirection | 7 |
| 5.2 | Mandatory support for HTTP URI | 7 |
| 5.3 | Rejecting content indirection | 7 |
| 5.4 | Specifying the location of the content via a URI | 8 |
| 5.5 | Marking indirect content optional | 8 |
| 5.6 | Specifying versioning information for the URI | 8 |
| 5.7 | Specifying the lifetime of the URI | 9 |
| 5.8 | Specifying the type of the indirect content | 9 |
| 5.9 | Specifying the size of the indirect content | 10 |
| 5.10 | Specifying the purpose of the indirect content | 10 |
| 5.11 | Specifying multiple URIs for content indirection | 10 |
| 5.12 | Specifying a hash value for the indirect content | 11 |
| 5.13 | Supplying additional comments about the indirect content | 12 |
| 5.14 | Relationship to Call-Info, Error-Info, and Alert-Info Headers | 12 |
| 6. | Examples | 12 |
| 6.1 | Single Content Indirection | 13 |
| 6.2 | Multipart MIME with Content Indirection | 13 |
| 7. | Security Considerations | 14 |
| 8. | IANA Considerations | 16 |
| 9. | Contributions | 16 |
| 10. | Acknowledgements | 16 |
| 11. | References | 17 |
| 11.1 | Normative References | 17 |
| 11.2 | Informative References | 17 |
| | Author's Address | 18 |
| | Intellectual Property and Copyright Statements | 19 |

Internet-Draft

Content Indirection in SIP Messages

October 2004

1. Terminology

[RFC 2119](#) [5] defines the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL".

2. Introduction

The purpose of the Session Initiation Protocol [9] (SIP) is to create, modify, or terminate sessions with one or more participants. SIP messages, like HTTP, are syntactically composed of a start line, one or more headers, and an optional body. Unlike HTTP, SIP is not designed as a general-purpose data transport protocol.

There are numerous reasons why it might be desirable to indirectly specify the content of the SIP message body. For bandwidth limited applications such as cellular wireless, indirection provides a means to annotate the (indirect) content with meta-data which may be used by the recipient to determine whether or not to retrieve the content over the resource limited link.

It is also possible that the content size to be transferred might potentially overwhelm intermediate signaling proxies, thereby unnecessarily increasing network latency. For time-sensitive SIP applications, this may be unacceptable. Indirect content can remedy this by moving the transfer of this content out of the SIP signaling network and into a potentially separate data transfer channel.

There may also be scenarios where the session related data (body) that needs to be conveyed does not directly reside on the endpoint or User Agent. In such scenarios, it is desirable to have a mechanism whereby the SIP message can contain an indirect reference to the desired content. The receiving party would then use this indirect reference to retrieve the content via a non-SIP transfer channel such as HTTP, FTP, or LDAP.

The purpose of content indirection is purely to provide an alternative transport mechanism for SIP MIME body parts. With the exception of the transport mechanism, indirect body parts are equivalent, and should have the same treatment, as in-line body parts.

Previous attempts at solving the content indirection problem made use of the text/uri-list [6] MIME type. While attractive for its simplicity (a list of URIs delimited by end-of-line markers), it failed to satisfy a number of the requirements for a more general-purpose content indirection mechanism in SIP. Most notably lacking is the ability to specify various attributes on a per-URI

basis. These attributes might include version information, the MIME type of the referenced content, etc.

In searching for a replacement for the text/uri-list MIME type, [RFC2017](#) defines a strong candidate. [RFC2017](#) [1] defines an extension to the message/external-body MIME type originally defined in [RFC2046](#) [3]. The extension that [RFC2017](#) makes is to allow a generic URI to specify the location of the content rather than protocol specific parameters for FTP, etc. as originally defined in [RFC2046](#). While providing most of the functionality needed for a SIP content indirection mechanism, [RFC2017](#) by itself is not a complete solution. This document specifies the usage of [RFC2017](#) necessary to fulfill the requirements outlined for content indirection.

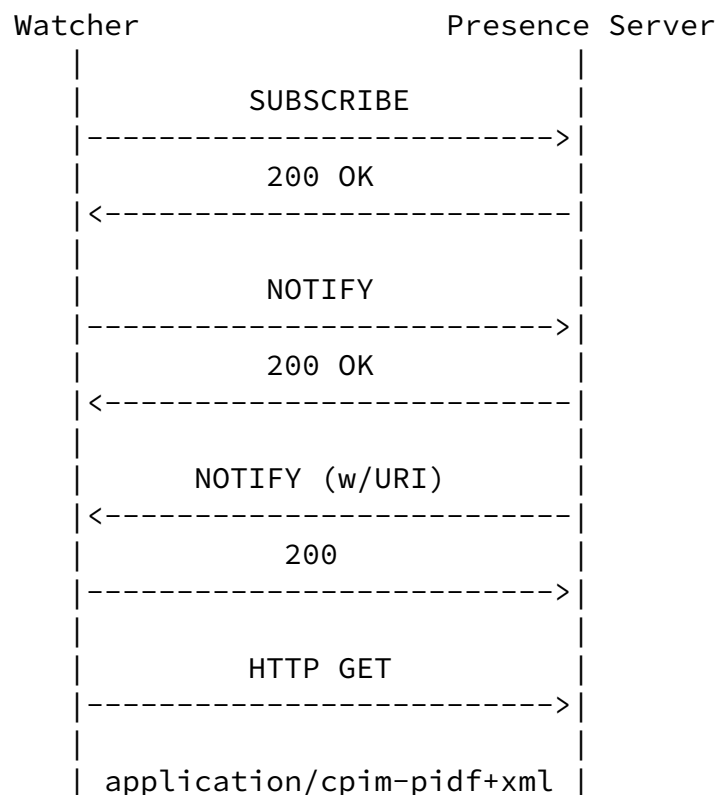
The requirements can be classified as applying either to the URI, which indirectly references the desired content, or to the content itself. Where possible, existing MIME parameters and entity headers are used to satisfy those requirements. MIME (Content-Type) parameters are the preferred manner of describing the URI while entity headers are the preferred manner of describing the (indirect) content. See [RFC 2045](#) [2] for a description of most of these entity headers and MIME parameters.

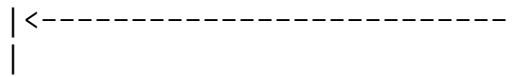
[3.](#) Example Use Cases

There are several example users of such a content indirection mechanism. These are examples only and are not intended to limit the scope or applicability of the mechanism.

[3.1](#) Presence Notification

The information carried in a presence document could potentially exceed the recommended size for a SIP (NOTIFY) request, particularly if the document carries aggregated information from multiple endpoints. In such a situation, it would be desirable to send the NOTIFY request with an indirect pointer to the presence document which could then be retrieved by, for example, HTTP.

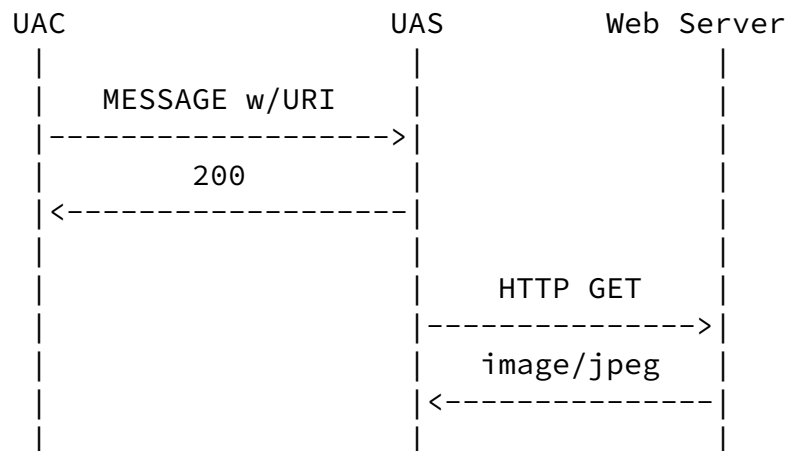




In this example, the presence server returns an HTTP URI pointing to a presence document on the presence server which the watcher can then fetch using an HTTP GET.

3.2 Document Sharing

During an instant messaging conversation, a useful service is document sharing wherein one party sends an IM (MESSAGE request) with an indirect pointer to a document which is meant to be rendered by the remote party. Carrying such a document directly in the MESSAGE request is not appropriate for most documents. Furthermore, the document to be shared may reside on a completely independent server from the originating party.



In this example, a user wishes to exchange a JPEG image that she has stored on her web server with another user she has an IM conversation with. She intends to render the JPEG inline in the IM conversation. The recipient of the MESSAGE request launches a HTTP GET request to the web server to retrieve the JPEG image.

4. Requirements

- o It MUST be possible to specify the location of content via a URI. Such URIs MUST conform with [RFC2396](#) [7] or its successors, such as [13].
- o It MUST be possible to specify the length of the indirect content.
- o It MUST be possible to specify the type of the indirect content.
- o It MUST be possible to specify the disposition of each URI independently.
- o It MUST be possible to label each URI to identify if and when the content referred to by that URI has changed. Applications of this mechanism may send the same URI more than once. The intention of this requirement is to allow the receiving party to determine if the content referenced by the URI has changed without having to actually retrieve that content. Example ways the URI could be labeled include a sequence number, timestamp, version number, etc. When used with HTTP, the entity-tag (ETAG) mechanism as defined in [RFC2068](#) [4] may be appropriate. Note that we are not labeling the URI itself, but the content to which the URI refers, and that the label is therefore effectively "metadata" of the content itself.
- o It MUST be possible to specify the time span for which a given URI is valid. This may or may not be the same as the lifetime for the content itself.
- o It MUST be possible for the UAC and the UAS to indicate support of this content indirection mechanism. A fallback mechanism SHOULD be specified in the event that one of the parties is unable to support content indirection.

- o It MUST be possible for the UAC and UAS to negotiate the type of the indirect content when using the content indirection mechanism.
- o It MUST be possible for the UAC and UAS to negotiate support for URI scheme(s) to be used in the content indirection mechanism. This is in addition to the ability to negotiate the content type.
- o It SHOULD be possible to ensure the integrity and confidentiality of the URI when it is received by the remote party.

- o It MUST be possible to process the content indirection without human intervention.
- o It MUST allow for indirect transference of content in any SIP message that would otherwise carry that content as a body.

[5.](#) Application of [RFC2017](#) to the Content Indirection Problem

The following text describes the application of [RFC2017](#) to the requirements for content indirection.

[5.1](#) Specifying support for content indirection

A UAC/UAS indicates support for content indirection by including the message/external-body MIME type in the Accept header. The UAC/UAS MAY supply additional values in the Accept header to indicate the content types that it is willing to accept, either directly or through content indirection. User-Agents supporting content indirection MUST support content indirection of the application/sdp MIME type.

For example:

Accept: message/external-body, image/*, application/sdp

[5.2](#) Mandatory support for HTTP URI

Applications which use this content indirection mechanism MUST support the HTTP URI scheme. Additional URI schemes MAY be used, but a UAC/UAS MUST support receiving a HTTP URI for indirect content if it advertises support for content indirection.

The UAS MAY advertise alternate access schemes in the schemes parameter of the Contact header in the UAS response to the UAC's session establishment request (e.g., INVITE, SUBSCRIBE, etc.), as described in Application Interaction [\[11\]](#).

[5.3](#) Rejecting content indirection

If a UAS receives a SIP request which contains a content indirection

payload, and the UAS cannot or does not wish to support such a content type, it MUST reject the request with a 415 Unsupported Media Type response as defined in [section 21.4.13](#) of SIP [9]. In particular, the UAC should note the absence of the message/external-body MIME type in the Accept header of this response to indicate that the UAS does not support content indirection, or the absence of the particular MIME type of the requested comment to indicate that the UAS does not support the particular media type.

[5.4](#) Specifying the location of the content via a URI

The URI for the indirect content is specified in a "URI" parameter of the message/external-body MIME type. An access-type parameter indicates that the external content is referenced by a URI. HTTP URI specifications MUST conform to [RFC2396](#) [7] or its successors [13].

For example:

```
Content-Type: message/external-body;  
             access-type="URL";  
             URL="http://www.example.com/the-indirect-content"
```

[5.5](#) Marking indirect content optional

Some content is not critical to the context of the communication if there is a fetch or conversion failure. The content indirection mechanism uses the Critical-Content mechanism described in [RFC3459](#) [10]. In particular, if the UAS is unable to fetch or render an optional body part, then the server MUST NOT return an error to the UAC.

[5.6](#) Specifying versioning information for the URI

In order to determine whether or not the content indirectly referenced by the URI has changed, a Content-ID entity header is used. The syntax of this header is defined in [RFC2045](#) [2]. Changes in the underlying content referred to by a URI MUST result in a change in the Content-ID associated with that URI. Multiple SIP messages carrying URI that refer to the same content SHOULD reuse the same Content-ID to allow the receiver to cache this content and avoid unnecessary retrievals. The Content-ID is intended to be globally unique and SHOULD be temporally unique across SIP dialogs.

For example:

Content-ID: <4232423424@www.example.com>

[5.7](#) Specifying the lifetime of the URI

The URI supplied by in Content-Type header is not required to be accessible or valid for an indefinite period of time. Rather, the supplier of the URI MUST specify the time period for which this URI is valid and accessible. This is done through an "EXPIRATION" parameter of the Content-Type. The format of this expiration parameter is a [RFC1123](#) date-time value. This is further restricted in this application to use only GMT time, consistent with the Date: header in SIP. This is a mandatory parameter. Note that the date-time value can range from minutes to days or even years.

For example:

```
Content-Type: message/external-body;
             expiration="Mon, 24 June 2002 09:00:00 GMT"
```

[5.8](#) Specifying the type of the indirect content

To support existing SIP mechanisms for the negotiation of content types, a Content-Type entity header SHOULD be present in the entity (payload) itself. If the protocol (scheme) of the URI supports its own content negotiation mechanisms (e.g. HTTP), this header may be omitted. The sender MUST however be prepared for the receiving party to reject content indirection if the receiver is unable to negotiate an appropriate MIME type using the underlying protocol for the URI scheme.

For example:

```
Content-Type: message/external-body; access-type="URL";
             expiration="Mon, 24 June 2002 09:00:00 GMT";
             URL="http://www.example.com/the-indirect-content"
<CRLF>
Content-Type: application/sdp
Content-Disposition: session
```

<CRLF>

[5.9](#) Specifying the size of the indirect content

When known in advance, the size of the indirect content SHOULD be supplied via a size parameter on the Content-Type header. This is an extension of [RFC2017](#) but in line with other access types defined for the message/external-body MIME type in [RFC2046](#). The content size is useful for the receiving party to make a determination about whether or not to retrieve the content. As with directly supplied content, a UAS may return a 513 error response in the event the content size is too large. This is an optional parameter.

For example:

```
Content-Type: message/external-body; access-type="URL";
             expiration="Mon, 24 June 2002 09:00:00 GMT";
             URL="http://www.example.com/the-indirect-content";
             size=4123
```

[5.10](#) Specifying the purpose of the indirect content

A Content-Disposition entity header MUST be present for all indirect content.

For example:

```
Content-Type: message/external-body; access-type="URL";
             expiration="Mon, 24 June 2002 09:00:00 GMT";
             URL="http://www.example.com/the-indirect-content"
<CRLF>
Content-Type: image/jpeg
Content-Disposition: render
```

[5.11](#) Specifying multiple URIs for content indirection

If there is a need to send multiple URIs for the purpose of content indirection, an appropriate multipart MIME type [\[3\]](#) should be used. Each URI MUST be contained in a single entity. Indirect content may be mixed with directly supplied content. This is particularly useful with the multipart/alternative MIME type.

NOTE: This specification does not change the meanings of the various multipart flavors, particularly multipart/related, as described in [RFC2387](#) [\[12\]](#).

Burger

Expires April 25, 2005

[Page 10]

Internet-Draft

Content Indirection in SIP Messages

October 2004

For example:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=boundary42

--boundary42
Content-Type: text/plain; charset=us-ascii

The company announcement for June, 2002 follows:
--boundary42
Content-Type: message/external-body;
    access-type="URL";
    expiration="Mon, 24 June 2002 09:00:00 GMT";
    URL="http://www.example.com/announcements/07242002";
    size=4123

Content-Type: text/html
Content-Disposition: render

--boundary42--
```

[5.12](#) Specifying a hash value for the indirect content

If the sender knows the specific content being referenced by the indirection, and the sender wishes the recipient to be able to validate that this content has not been altered from that intended by

the sender, the sender includes a SHA-1 [8] hash of the content. If included, the hash is encoded by extending the MIME syntax [3] to include a "hash" parameter for the content type "message/external-body", the value of which is a hexadecimal encoding of the hash.

For example:

```
Content-Type: message/external-body;
    access-type="URL";
    expiration="Mon, 24 June 2002 09:00:00 GMT";
    URL="http://www.example.com/the-indirect-content.au";
    size=52723;
    hash=10AB568E91245681AC1B
<CRLF>
Content-Disposition: render
```

[5.13](#) Supplying additional comments about the indirect content

One MAY use the Content-Description entity header to provide optional, freeform text to comment on the indirect content. This text MAY be displayed to the end user but MUST NOT be used by other elements to determine disposition of the body.

For example:

```
Content-Type: message/external-body;
    access-type="URL";
    expiration="Mon, 24 June 2002 09:00:00 GMT";
    URL="http://www.example.com/the-indirect-content";
    size=52723
<CRLF>
Content-Description: Multicast gaming session
Content-Disposition: render
```

[5.14](#) Relationship to Call-Info, Error-Info, and Alert-Info Headers

SIP [9] defines three headers that supply additional information with

regard to a session, a particular error response, or alerting. All three of these headers allow the UAC or UAS to indicate additional information through a URI. They may be considered a form of content indirection. The content indirection mechanism defined in this document is not intended as a replacement for these headers. Rather, the headers defined in SIP MUST be used in preference to this mechanism where applicable because of the well-defined semantics of those headers.

[6.](#) Examples

[6.1](#) Single Content Indirection

```
INVITE sip:boromir@example.com SIP/2.0
From: <sip:gandalf@example.net>;tag=347242
To: <sip:boromir@example.com>
Call-ID: 3573853342923422@example.net
CSeq: 2131 INVITE
Accept: message/external-body application/sdp
Content-Type: message/external-body;
    ACCESS-TYPE=URL;
    URL="http://www.example.net/party/06/2002/announcement";
    EXPIRATION="Sat, 20 Jun 2002 12:00:00 GMT";
    size=231
Content-Length: ...
```

Content-Type: application/sdp
Content-Disposition: session
Content-ID: <4e5562cd1214427d@example.net>

[6.2](#) Multipart MIME with Content Indirection

MESSAGE sip:boromir@example.com SIP/2.0
From: <sip:gandalf@example.net>;tag=34589882
To: <sip:boromir@example.com>
Call-ID: 9242892442211117@example.net
CSeq: 388 MESSAGE
Accept: message/external-body, text/html, text/plain,
image/*, text/x-emoticon
MIME-Version: 1.0

Content-Type: multipart/mixed; boundary=zz993453

--zz993453

Content-Type: message/external-body;
access-type="URL";
expiration="Mon, 24 June 2002 09:00:00 GMT";
URL="http://www.example.net/company_picnic/image1.png";
size=234422

Content-Type: image/png
Content-ID: <9535035333@example.net>
Content-Disposition: render
Content-Description: Kevin getting dunked in the wading pool

--zz993453

Content-Type: message/external-body;
access-type="URL";
expiration="Mon, 24 June 2002 09:00:00 GMT";
URL="http://www.example.net/company_picnic/image2.png";
size=233811

Content-Type: image/png
Content-ID: <1134299224244@example.net>
Content-Disposition: render
Content-Description: Peter on his tricycle

--zz993453--

[7.](#) Security Considerations

Any content indirection mechanism introduces additional security concerns. By its nature, content indirection requires an extra processing step and information transfer. There are a number of potential abuses of a content indirection mechanism:

- o Content indirection allows the initiator to choose an alternative protocol with weaker security or known vulnerabilities for the

content transfer. For example, asking the recipient to issue an

- HTTP request that results in a Basic authentication challenge.
- o Content indirection allows the initiator to ask the recipient to consume additional resources in the information transfer and content processing, potentially creating an avenue for denial of service attacks. For example, an active FTP URL consuming 2 connections for every indirect content message.
 - o Content indirection could be used as a form of port scanning attack where the indirect content URL is actually a bogus URL pointing to an internal resource of the recipient. The response to the content indirection request could reveal information about open (and vulnerable) ports on these internal resources.
 - o A content indirection URL can disclose sensitive information about the initiator such as an internal user name (as part of an HTTP URL) or possibly geolocation information.

Fortunately, all of these potential threats can be mitigated through careful screening of both the indirect content URIs that are received as well as those that are sent. Integrity and confidentiality protection of the indirect content URI can prevent additional attacks as well.

For confidentiality, integrity, and authentication, this content indirection mechanism relies on the security mechanisms outlined in [RFC3261](#). In particular, the usage of S/MIME as defined in [section 23 of RFC3261](#) provides the necessary mechanism to ensure integrity, protection, and confidentiality of the indirect content URI and associated parameters.

Securing the transfer of the indirect content is the responsibility of the underlying protocol used for this transfer. If HTTP is used, applications implementing this content indirection method SHOULD support the HTTPS URI scheme for secure transfer of content and MUST support the upgrading of connections to TLS using starttls. Note that a failure to complete HTTPS or starttls (for example, due to cert or encryption mismatch) after having accepted the indirect content in the SIP request is not the same as rejecting the SIP request, and may require additional user-user communication for correction.

Note that this document does not advocate the use of transitive trust. That is, just because the UAS receives a URI from a UAC that the UAS trusts, the UAS SHOULD NOT implicitly trust the object referred to by the URI without establishing its own trust relationship with the URI provider.

Access control to the content referenced by the URI is not defined by this specification. Access control mechanisms may be defined by the

protocol for the scheme of the indirect content URI.

If the UAC knows the content in advance, the UAC SHOULD include a hash parameter in the content indirection. The hash parameter is a hexadecimal-encoded SHA-1 [8] hash of the indirect content. If a hash value is included, the recipient MUST check the indirect content against that hash and indicate any mismatch to the user.

In addition, if the hash parameter is included, and the target URI involves setting up a security context using certificates, the UAS MUST ignore the results of the certificate validation procedure, and instead verify that the hash of the (canonicalized) content received matches the hash presented in the content-indirection hash parameter.

If the hash parameter is NOT included, the sender SHOULD use only schemes that offer message integrity (such as https:). When the hash parameter is not included and security using certificates is used, the UAS MUST verify any server certificates using the UAS's list of trusted top-level certificate authorities.

If hashing of indirect content is not used, the possibility exists that the content returned to the recipient by exercise of the indirection has been altered from that intended by the sender.

8. IANA Considerations

This document raises no new IANA considerations.

9. Contributions

Sean Olson, seanol@microsoft.com, provided the vast majority of the content of this document, including editorship through the first IESG review.

Dean Willis touched it next.

Eric Burger edited the document and addressed IESG comments, including the access protocol negotiation mechanism.

10. Acknowledgements

Cullen Jennings and Nancy Greene provided a thorough review and valuable comments and suggestions.

11. References

11.1 Normative References

- [1] Freed, N. and K. Moore, "Definition of the URL MIME External-Body Access-Type", [RFC 2017](#), October 1996.
- [2] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.
- [3] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- [4] Fielding, R., Gettys, J., Mogul, J., Nielsen, H. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2068](#), January 1997.
- [5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [6] Daniel, R., "A Trivial Convention for using HTTP in URN Resolution", [RFC 2169](#), June 1997.
- [7] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", [RFC 2396](#), August 1998.
- [8] Eastlake, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", [RFC 3174](#), September 2001.
- [9] 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.
- [10] Burger, E., "Critical Content Multi-purpose Internet Mail Extensions (MIME) Parameter", [RFC 3459](#), January 2003.
- [11] Rosenberg, J., "A Framework for Application Interaction in the Session Initiation Protocol (SIP)",

[draft-ietf-sipping-app-interaction-framework-02](#) (work in progress), July 2004.

[11.2](#) Informative References

- [12] Levinson, E., "The MIME Multipart/Related Content-type", [RFC 2387](#), August 1998.
- [13] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform

Burger

Expires April 25, 2005

[Page 17]

Internet-Draft

Content Indirection in SIP Messages

October 2004

Resource Identifier (URI): Generic Syntax",
[draft-fielding-uri-rfc2396bis-06](#) (work in progress), July 2004.

Author's Address

Eric Burger (editor)
Brooktrout Technology, Inc.

EMail: eburger@brooktrout.com

URI: <http://www.brooktrout.com>

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 (2004). 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.