SIPPING Working Group
Internet-Draft
Expires: January 5, 2005

M. Garcia-Martin
Nokia
G. Camarillo
Ericsson
July 7, 2004

Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP)
draft-ietf-sipping-uri-list-message-00.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies how to request a SIP URI-list service to send a copy of a MESSAGE to a set of destinations. The client sends a SIP MESSAGE request with a URI-list to the URI-list service, which sends a similar MESSAGE request to each of the URIs included in the list.

Table of Contents

1. Introduction

SIP [2] can carry instant messages in MESSAGE [3] requests. The Advanced Instant Messaging Requirements for SIP [8] mentions the need for sending a MESSAGE request to multiple receipients:

"REQ-GROUP-3: It MUST be possible for a user to send to an ad-hoc group, where the identities of the recipients are carried in the message itself."

To meet this requirement, we allow SIP MESSAGE requests carry URI-lists in "uri-list" body parts, as specified in [4]. A SIP URI-list service, which is a specialized application server, receives the request and sends a similar MESSAGE request to each of the URIs in the list. Each of these MESSAGE requests contains a copy of the body included in the original MESSAGE request.

The UAC (User Agent Client) needs to be configured with the SIP URI of the application server that provides the functionality. Discovering and provisioning of this URI to the UAC is outside the scope of this document.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

'MESSAGE URI-list service': SIP application server that receives a MESSAGE request with a URI-list and sends a similar MESSAGE request to each URI in the list. MESSAGE URI-list services behave effectively as specialised B2BUAs (Back-To-Back-User-Agents). A MESSAGE URI-list service can also offer URI-list services for other methods, although this functionality is outside the scope of this document. In this document we only discuss MESSAGE URI-list services.

'Incoming MESSAGE request': A SIP MESSAGE request that a UAC creates and addresses to a MESSAGE URI-list service. Besides the regular instant message payload, an incoming MESSAGE request contains a URI-list.

'Outgoing MESSAGE request': A SIP MESSAGE request that a MESSAGE URI-list service creates and addresses to a UAS (User Agent Server). It contains the regular instant message payload.

3. Procedures at the UAC

A client that wants to create a multiple-recipient MESSAGE request adds a body part, whose disposition type is "uri-list", which contains a URI-list with the recipients of the MESSAGE.

Multiple-recipient MESSAGE requests typically contain a multipart body that contains the body carrying the list and the actual instant message payload. In some cases, the MESSAGE request may contain bodies other than the text and the list bodies (e.g., when the request is protected with S/MIME [6]).

Typically, the MESSAGE URI-list service will copy all the significant header fields in the outgoing MESSAGE request. However, there might be cases where the SIP UA wants the MESSAGE URI-list service to add a particular header field with a particular value, even if the header field wasn't present in the MESSAGE request sent by the UAC. In this case, the UAC MAY use the "?" mechanism described in Section 19.1.1 of RFC 3261 [2] to encode extra information in any URI in the list. However, the UAC MUST NOT use the special "body" hname (see Section 19.1.1 of RFC 3261 [2]) to encode a body, since the body is present in the MESSAGE request itself.

The following is an example of a URI that uses the "?" mechanism:

sip:bob@example.com?Accept-Contact=*%3bmobility%3d%22mobile%22

The previous URI requests the MESSAGE URI-list service to add the following header field to a MESSAGE request to be sent to bob@example.com:

Accept-Contact: *;mobility="mobile"

As described in [4], the default format for URI-lists in SIP is the XCAP resource list format [5]. Still, specific services need to describe which information clients should include in their URI lists, as described in [4]

UAs generating multiple recipient MESSAGEs SHOULD use flat lists (i.e., no hierarchical lists), SHOULD NOT use any entry's attributes but "uri", and SHOULD NOT include any elements inside entries but "display-name" elements.

A MESSAGE URI-list service receiving a URI-list with more information than what we have just described SHOULD discard all the extra information.

4. Procedures at the MESSAGE URI-List Service

On reception of a MESSAGE request with a URI-list, a MESSAGE URI-list service SHOULD answer to the UAC with a 202 Accepted response. Note that the status code in the response to the MESSAGE does not provide any information about whether or not the MESSAGEs generated by the URI-list service were successfully delivered to the URIs in the list. That is, a 202 Accepted means that the MESSAGE URI-list service has received the MESSAGE and that it will try to send a similar MESSAGE to the URIs in the list. Designing a mechanism to inform a client about the delivery status of an instant message is outside the scope of this document.

On reception of a MESSAGE request with a URI-list, a MESSAGE URI-list service SHOULD create as many new MESSAGE requests as URIs the list contains, except when two of those URIs are equivalent (section 19.1.4 of RFC 3261 [2] defines equivalent URIs), in which case the MESSAGE URI-list service SHOULD create only one outgoing MESSAGE request per URI.

When creating the body of each of the outgoing MESSAGE requests, the MESSAGE URI-list service tries to keep the relevant bodies of the incoming MESSAGE request and copies them to the outgoing MESSAGE request. The following guidelines are provided:

- The incoming MESSAGE request typically contains a URI-list body [4] with the actual list of recipients. The MESSAGE URI-list service need not copy the URI-list body to each of the outgoing MESSAGE requests, although it MAY do it.
  NOTE: This document does not provide any semantics associated to a URI-list body included in an outgoing MESSAGE request. Future extensions may indicate actions at a UAS when it receives that body.
- A MESSAGE request received at a MESSAGE URI-list service can contain one or more security bodies encrypted with the public key of the MESSAGE URI-list service. These bodies are deemed to be read by the URI-list service rather than the recipient of the outgoing MESSAGE request (which will not be able to decrypt them). Therefore, a MESSAGE URI-list service MUST NOT copy any security body (such as an S/MIME encrypted body) addressed to the MESSAGE URI-list service to the outgoing MESSAGE request. This includes bodies encrypted with the public key of the URI-list service.
- An exception to this rule is the URI-list itself: as mentioned in Section 4, a MESSAGE URI-list service need not, but MAY, copy the URI-list into each of the outgoing MESSAGE requests; on doing so, a MESSAGE URI-list service SHOULD use S/MIME [6] to encrypt the URI-list with the public key of the receiver.

- The MESSAGE URI-list service SHOULD copy all the rest of the message bodies (e.g., text messages, images, etc.) to the outgoing MESSAGE request.
- If there is only one body left, the MESSAGE URI-list service MUST remove the multipart/mixed wrapper in the outgoing MESSAGE request.

The rest of the MESSAGE request corresponding to a given URI in the list MUST be created following the rules in Section 19.1.5 "Forming Requests from a URI" of RFC 3261 [2]. In particular, Section 19.1.5 of RFC 3261 [2] states:

"An implementation SHOULD treat the presence of any headers or body parts in the URI as a desire to include them in the message, and choose to honor the request on a per-component basis."

SIP allows to append a "method" parameter to a URI. Therefore, it is legitimate that an the "uri" attribute of the "entry" element in the XCAP resource list contains a "method" parameter. MESSAGE URI-list services MUST generate only MESSAGE requests, regardless of the "method" parameter that the URIs in the list indicate. Effectively, MESSAGE URI-list services MUST ignore the "method" parameter in each of the URIs present in the URI list.

It is RECOMMENDED that the MESSAGE URI-list service copies the From header field of the incoming MESSAGE into the outgoing MESSAGE requests (note that this does not apply to the "tag" parameter). The MESSAGE URI-list service SHOULD also copy into the outgoing MESSAGE request any P-Asserted-Identity header fields present in the incoming MESSAGE request.

For each given outgoing MESSAGE request, the MESSAGE URI-list service SHOULD generate a new To header field value which, according to the procedures of RFC 3261 Section 8.1.1.1, should be equal to the Request-URI of the outgoing MESSAGE request.

For each given outgoing MESSAGE request, the MESSAGE URI-list service SHOULD initialize the values of the Call-ID, CSeq and Max-Forwards header fields. The MESSAGE URI-list service should also include its own value in the Via header field.

5. Examples

The following is an example of an incoming MESSAGE request which carries a URI list in its body.

```
MESSAGE sip:list-service.example.com SIP/2.0
Via: SIP/2.0/TCP uac.example.com
    ;branch=z9hG4bKhjhs8ass83
Max-Forwards: 70
To: MESSAGE URI-List Service <sip:list-service.example.com>
From: Carol <sip:carol@example.com>;tag=32331
Call-ID: d432fa84b4c76e66710
CSeq: 1 MESSAGE
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: 440

--boundary1
Content-Type: text/plain

Hello World!

--boundary1
Content-Type: application/resource-lists+xml
Content-Disposition: uri-list

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@example.com" />
    <entry uri="sip:joe@example.org" />
    <entry uri="sip:ted@example.net" />
  </list>
</resource-lists>
--boundary1--
```

Figure 3: Multiple recipient incoming MESSAGE request

The following is an example of one of the outgoing MESSAGE requests that the MESSAGE URI-list service creates.

```
MESSAGE sip:bill@example.com SIP/2.0
Via: SIP/2.0/TCP list-service.example.com
    ;branch=z9hG4bKhjhs8as34sc
Max-Forwards: 70
To: <sip:bill@example.com>
From: Carol <sip:carol@uac.example.com>;tag=210342
Call-ID: 39s02sdsl20d9sj2l
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 13

Hello World!
```

Figure 4: Outgoing MESSAGE request

6. Security Considerations

The Security Considerations Section of the Requirements and Framework for SIP URI-List Services [7] discusses issues related to SIP URI-list services. Implementations of MESSAGE URI-list services MUST follow the security-related rules in [7]. These rules include mandatory authentication and authorization of clients, and opt-in lists.

If the contents of the instant message needs to be kept private, the user agent client SHOULD use S/MIME [6] to prevent a third party from viewing this information. In this case, the user agent client SHOULD encrypt the instant message body with a content encryption key. Then, for each receiver in the list, the UAC SHOULD encrypt the content encryption key with the public key of the receiver, and attach it to the MESSAGE request.

7. Acknowledgements

Duncan Mills supported the idea of having 1 to n MESSAGEs. Ben Campbell, Paul Kyzivat, and Cullen Jennings provided helpful comments.

8. Change control

8.1 Changes from draft--sipping-message-exploder-00.txt to draft-ietf-sipping-uri-list-message-00.txt

Clarified that the MESSAGE exploder should not distribute a body that has been encrypted with the public key of the exploder. The exception is the URI list, which can be distributed by the exploder, providing that is encrypted with the public key of the receiver.

The security considerations section describes how to encrypt the list and how to encrypt the instant message payload.

Terminology aligned with the requirements and the framework for URI-list services (e.g., the term "exploder" has been deprecated).

8.2 Changes from draft-garcia-simple-message-exploder-00.txt to draft-garcia-sipping-message-exploder-00.txt

The MESSAGE exploder may or may not copy the URI list body to the outgoing MESSAGE request. This allows to extend the mechanism with a Reply-to-all feature.

It is clarified that the MESSAGE exploder must not include a list in the outgoing MESSAGE requests. This avoids loops or requires a MESSAGE exploder functionality in the next hop.

The MESSAGE exploder must remove the multipart/mixed wrapper if there is only one body left in the outgoing MESSAGE request.

Filename changed due to focus on the SIPPING WG.

9. References

9.1 Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] 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.

[3] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C. and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[4] Camarillo, G., "Providing a Session Initiation Protocol (SIP) Application Server with a List of URIs", draft-camarillo-sipping-uri-list-01 (work in progress), February 2004.

[5] Rosenberg, J., "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Presence Lists", draft-ietf-simple-xcap-list-usage-02 (work in progress), February 2004.

[6] Ramsdell, B., "S/MIME Version 3.1 Message Specification",

draft-ietf-smime-rfc2633bis-07 (work in progress), February 2004.

[7] Camarillo, G., "Requirements for Session Initiation Protocol (SIP) Exploder Invocation", draft-camarillo-sipping-exploders-02 (work in progress), February 2004.

9.2 Informational References

[8] Rosenberg, J., "Advanced Instant Messaging Requirements for the Session Initiation Protocol (SIP)", draft-rosenberg-simple-messaging-requirements-01 (work in progress), February 2004.

[9] Peterson, J., "SIP Authenticated Identity Body (AIB) Format", draft-ietf-sip-authid-body-02 (work in progress), July 2003.

[10] Jennings, C., Peterson, J. and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

Authors' Addresses

Miguel A. Garcia-Martin
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland

EMail: miguel.an.garcia@nokia.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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