SIPPING Working Group   G. Camarillo
Internet-Draft   Ericsson
Expires: June 2, 2006   November 29, 2005

The Session Initiation Protocol (SIP) Conference Bridge Transcoding Model
draft-ietf-sipping-transc-conf-01.txt

Status of this Memo

Copyright Notice

Abstract

This document describes how to invoke transcoding services using the conference bridge model. This way of invocation meets the requirements for SIP regarding transcoding services invocation to support deaf, hard of hearing and speech-impaired individuals.

Table of Contents

1. Introduction

The Framework for Transcoding with SIP [9] describes how two SIP [3] UAs (User Agents) can discover imcompatibilities that prevent them from establishing a session (e.g., lack of support for a common codec or for a common media type). When such incompatibilities are found, the UAs need to invoke transcoding services to successfully establish the session. The transcoding framework introduces two models to invoke transcoding services: the 3pcc (third-party call control) model [8] and the conference bridge model. This document specifies the conference bridge model.

In the conference bridge model for transcoding invocation, a transcoding server that provides a particular transcoding service (e.g., speech-to-text) behaves as a B2BUA (Back-to-Back User Agent) between both UAs and is identified by a URI. As shown in Figure 1, both UAs, A and B, exchange signalling and media with the transcoder T. The UAs do not exchange any traffic (signalling or media) directly between them.

```
       +-------+
       |       |**
       |   T   |  **
       |       |\   **
       +-------+ \\   **
         ^   *     \\   **
         |   *       \\   **
         |   *         SIP  **
        SIP  *           \\   **
         |   *             \\   **
         |   *               \\   **
         v   *                 \    **
       +-------+                +-------+
       |       |                |       |
       |   A   |                |   B   |
       |       |                |       |
       +-------+                +-------+


        <-SIP-> Signalling
        ******* Media
```

Figure 1: Conference bridge model

Section 3 and Section 4 specify how the caller A or the callee B, respectively, can use the conference bridge model to invoke transcoding services from T.

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.

3. Caller's Invocation

User agent A needs to perform two operations to invoke transcoding services from T for a session between user agent A and user agent B. User agent A needs to establish a session with T and provide T with user agent B's URI so that T can generate an INVITE towards user agent B.

3.1. Procedures at the User Agent

User agent A uses the procedures for Conference Establishment Using Request-Contained Lists in SIP [11] to provide T with B's URI using the same INVITE that establishes the session between A and T. That is, user agent A adds to the INVITE a body part whose disposition type is recipient-list [10]. This body part consists of a URI-list that MUST contain a single URI: user agent B's URI.

3.2. Procedures at the Transcoder

On receiving an INVITE with a URI-list body, the transcoder follows the procedures in [11] to generate an INVITE request towards the URI contained in the URI-list body. Note that the transcoder acts as a B2BUA, not as a proxy.

Additionally, the transcoder MUST generate the From header field of the outgoing INVITE request using the same value as the From header field included in the incoming INVITE request, subject to the privacy requirements (see [5] and [6]) expressed in the incoming INVITE request. Note that this does not apply to the "tag" parameter.

The session description the transcoder includes in the outgoing INVITE request depends on the type of transcoding service that particular transcoder provides. For example, a transcoder resolving audio codec incompatibilities would generate a session description listing the audio codecs the transcoder supports.

When the transcoder receives a final response for the outgoing INVITE requests, it generates a new final response for the incoming INVITE request. This new final response SHOULD have the same status code as

the one received in the response for the outgoing INVITE request.

If a trancoder receives an INVITE request with a URI-list with more than one URI, it SHOULD return a 488 (Max 1 URI allowed in URI-list) response.

3.3. Example

Figure 2 shows the message flow for the caller's invocation of a transcoder T. The caller A sends an INVITE (1) to the transcoder (T) to establish the session A-T. Following the procedures in [11], the caller A adds a body part whose disposition type is recipient-list [10].

```
A                            T                            B
|                            |                            |
|-----(1) INVITE SDP A------>|                            |
|                            |                            |
|<-(2) 183 Session Progress-|                            |
|                            |-----(3) INVITE SDP TB---->|
|                            |                            |
|                            |<-----(4) 200 OK SDP B-----|
|                            |                            |
|                            |---------(5) ACK---------->|
|<----(6) 200 OK SDP TA-----|                            |
|                            |                            |
|---------(7) ACK---------->|                            |
|                            |                            |
| ************************* | ************************* |
|**         Media         **|**         Media         **|
| ************************* | ************************* |
|                            |                            |
```

Figure 2: Successful invocation of a transcoder by the caller

The following example shows an INVITE with two body parts: an SDP [14] session description and a URI-list.

```
INVITE sip:transcoder@example.com SIP/2.0
Via: SIP/2.0/TCP client.chicago.example.com
    ;branch=z9hG4bKhjhs8ass83
Max-Forwards: 70
To: Transcoder <sip:transcoder@example.org>
From: A <sip:A@chicago.example.com>;tag=32331
Call-ID: d432fa84b4c76e66710
CSeq: 1 INVITE
Contact: <sip:A@client.chicago.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
     SUBSCRIBE, NOTIFY
Allow-Events: dialog
Accept: application/sdp, message/sipfrag
Require: recipient-list-invite
Conten-Type: multipart/mixed;boundary="boundary1"
Content-Length: 556

--boundary1
Content-Type: application/sdp

v=0
o=example 2890844526 2890842807 IN IP4 chicago.example.com
s=-
c=IN IP4 192.0.2.1
t=0 0
m=audio 20000 RTP/AVP 0
a=rtpmap:0 PCMU/8000

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

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:B@example.org" />
  </list>
</resource-lists>
--boundary1--
```

On receiving the INVITE, the transcoder generates a new INVITE towards the callee. The transcoder acts as a B2BUA, not as a proxy. Therefore, this new INVITE (3) belongs to a different transaction than the INVITE (1) received by the transcoder.

When the transcoder receives a final response (4) from the callee, it generates a new final response (6) for INVITE (1). This new final

response (6) has the same status code as the one received in the response from the callee (4).

3.4. Unsuccessful Session Establishment

Figure 3 shows a similar message flow as the one in Figure 3. Nevertheless, this time the callee generates a non-2xx final response (4). Consequently, the transcoder generates a non-2xx final response (6) towards the caller as well.

```
A                            T                            B
|                            |                            |
|-----(1) INVITE SDP A----->|                            |
|                            |                            |
|<-(2) 183 Session Progress-|                            |
|                            |-----(3) INVITE SDP TB---->|
|                            |                            |
|                            |<----(4) 603 Decline-------|
|                            |                            |
|                            |---------(5) ACK---------->|
|<----(6) 603 Decline-------|                            |
|                            |                            |
|---------(7) ACK---------->|                            |
|                            |                            |
```

Figure 3: Unsuccessful session establishment

The ambiguity in this flow is that, if the provisional response (2) gets lost, the caller does not know whether the 603 (Decline) response means that the initial INVITE (1) was rejected by the transcoder or that the INVITE generated by the transcoder (4) was rejected by the callee. The use of the "History-Info" header field [12] between the transcoder and the caller resolves the previous ambiguity.

Callers that do not support the "History-Info" header field can, alternatively, require the use of the reliable provisional responses [4] SIP extension. If the caller receives a response reporting a reachability problem, the caller can also send an OPTIONS request to the transcoder to check whether or not the transcoder is reachable. If the transcoder is reachable, the party that could not be reached was the callee.

Note that this ambiguity problem could also have been resolved by having transcoders act as a pure conference bridge. The transcoder would respond with a 200 (OK) the INVITE request from the caller and generate an outgoing INVITE request towards the callee. The caller

would get information about the result of the latter INVITE request by subscribing to the conference event package [15] at the transcoder. Nevertheless, while this flow would have resolved the ambiguity problem without requiring support for the "History-Info" header field, it is more complex, requires a higher number on messages, and introduces higher session setup delays. That is why it was not chosen to implement transcoding services.

4. Callee's Invocation

If a UA receives an INVITE with a session description that is not acceptable, it can redirect it to the transcoder by using a 302 (Moved Temporarily) response. The Contact header field of the 302 (Moved Temporarily) response contains the URI of the transcoder plus a "?body=" parameter. This parameter contains a recipient-list body with B's URI. Note that some escaping (e.g., for Carriage Returns and Line Feeds) is needed to encode a recipient-list body in such a parameter. Figure 4 shows the message flow for this scenario.

```
A                            T                            B
|                            |                            |
|-------------------(1) INVITE SDP A------------------->|
|                            |                            |
|<---------------(2) 302 Moved Temporarily---------------|
|                            |                            |
|-----------------------(3) ACK------------------------->|
|                            |                            |
|-----(4) INVITE SDP A----->|                            |
|                            |                            |
|<-(5) 183 Session Progress-|                            |
|                            |-----(6) INVITE SDP TB---->|
|                            |                            |
|                            |<-----(7) 200 OK SDP B-----|
|                            |                            |
|                            |---------(8) ACK---------->|
|<----(9) 200 OK SDP TA-----|                            |
|                            |                            |
|--------(10) ACK---------->|                            |
|                            |                            |
| ************************* | ************************* |
|**         Media          **|**         Media          **|
| ************************* | ************************* |
```

Figure 4: Callee's invocation of a transcoder

Note that A does not necessarily need to be the one performing the

recursion on the 302 (Moved Temporarily) response. Any proxy in the path between A and B may perform such a recursion.

5. Security Considerations

Transcoders implementing this specification behave as a URI-list service as described in [11]. Therefore, the security considerations for URI-list services discussed in [10] apply here as well.

In particular, the requirements related to list integrity and unsolicited requests are important for transcoding services. User agents SHOULD integrity protect URI-lists using mechanisms such as S/MIME [7] or TLS [2], which can also provide URI-list confidentiality if needed. Additionally, transcoders MUST authenticate and authorize users and MAY provide information about the identity of the original sender of the request in their outgoing requests by using the SIP identity mechanism [13].

The requirement in [10] to use opt-in lists (e.g., using the Framework for Consent-Based Communications in SIP [16]) deserves special discussion. The type of URI-list service implemented by transcoders following this specification does not produce amplification (only one INVITE request is generated by the transcoder on receiving an INVITE request from a user agent) and does not involve a translation to a URI that may be otherwise unknown to the caller (the caller places the callee's URI in the body of its initial INVITE request). Additionally, the identity of the caller is present in the INVITE request generated by the transcoder. Therefore, there is no requirement for transcoders implementing this specification to use opt-in lists.

6. IANA Considerations

This document does not contain any IANA actions.

7. Contributors

This document is the result of discussions amongst the conferencing design team. The members of this team include Eric Burger, Henning Schulzrinne, and Arnoud van Wijk.

8. References

8.1. Normative References

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

[2] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[3] 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.

[4] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[5] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[6] 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.

[7] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling", RFC 3850, July 2004.

[8] Camarillo, G., Burger, E., Schulzrinne, H., and A. van Wijk, "Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc)", RFC 4117, June 2005.

[9] Camarillo, G., "Framework for Transcoding with the Session Initiation Protocol", draft-camarillo-sipping-transc-framework-00 (work in progress), August 2003.

[10] Camarillo, G. and A. Roach, "Framework and Security Considerations for Session Initiation Protocol (SIP) Uniform Resource Identifier (URI)-List Services", draft-ietf-sipping-uri-services-04 (work in progress), October 2005.

[11] Camarillo, G. and A. Johnston, "Conference Establishment Using Request-Contained Lists in the Session Initiation Protocol (SIP)", draft-ietf-sipping-uri-list-conferencing-04 (work in progress), October 2005.

[12] Barnes, M., "An Extension to the Session Initiation Protocol for Request History Information", draft-ietf-sip-history-info-06 (work in progress), January 2005.

[13] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

8.2. Informational References

[14] Handley, M., "SDP: Session Description Protocol", draft-ietf-mmusic-sdp-new-25 (work in progress), July 2005.

[15] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Conference State", draft-ietf-sipping-conference-package-12 (work in progress), July 2005.

[16] Rosenberg, J., "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)", draft-ietf-sipping-consent-framework-03 (work in progress), October 2005.

Author's Address

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.