Internet Engineering Task Force SIP WG
Internet Draft
G. Camarillo
Ericsson
draft-camarillo-sipping-transc-b2bua-01.txt
February 7, 2004
Expires: August, 2004

The Session Initiation Protocol Conference Bridge Transcoding Model

STATUS OF THIS MEMO

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 [1] (draft-ietf-sipping-transc-framework) describes how two SIP UAs 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. Using the conference bridge model is one way to perform such invocation.

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 between both UAs and is identified by a URI.

2 Caller's Invocation

Figure 1 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. The URI in the Request-URI of this INVITE contains a list parameter, as defined in [2] (draft-camarillo-sipping-uri-list-01), with a pointer to a URI list. This URI list contains a single URI: the callee's URI, as shown below:

```
INVITE sip:transcoder@example.com;list=cid:cn35t8@example.com SIP/2.0
Via: SIP/2.0/TCP client.chicago.example.com
    ;branch=z9hG4bKhjhs8ass83
Max-Forwards: 70
To: "Transcoder" <sip:transcoder@example.com>
From: Caller <sip:caller@chicago.example.com>;tag=32331
Call-ID: d432fa84b4c76e66710
CSeq: 1 INVITE
Contact: <sip:caller@client.chicago.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER,
     SUBSCRIBE, NOTIFY
Conten-Type: multipart/mixed;boundary="boundary1"
Content-Length: xxx

--boundary1
Content-Type: application/sdp
Content-Length: xxx

v=0
o=caller 2890844526 2890842807 IN IP4 chicago.example.com
s=Example Subject
c=IN IP4 192.0.0.1
t=0 0
```

```
m=audio 20000 RTP/AVP 0

--boundary1
Content-Type: application/resource-lists+xml
Content-Length: 367
Content-ID: <cn35t8@example.com>

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list name="ad-hoc-1">
    <entry name="1" uri="sip:callee@example2.com" />
  </list>
</resource-lists>
--boundary1--
```

```
 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 1: Successful invocation of a transcoder by the caller

On reception of the INVITE, the transcoder generates a new INVITE towards the callee. The transcoder acts as a B2BUA, so, 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).

The advantage of this message flow is that, for both user agents, is indentical to the flow for establishing a regular session (i.e., without transcoder) between them. Additionaly, the only difference in the message contents is that the caller needs to use a list parameter in the Request-URI of the initial INVITE.

2.1 Unsuccessful Session Establishment

Figure 2 shows a similar message flow as the one in Figure 1. 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) 404 Not Found-----|
 |                            |                            |
 |                            |---------(5) ACK---------->|
 |<----(6) 404 Not Found-----|                            |
 |                            |                            |
 |---------(7) ACK---------->|                            |
 |                            |                            |
```

Figure 2: Unsuccessful session establishment

The problem with this flow is that the caller does not know whether the 404 (Not Found) response means that the initial INVITE (1) did not reach the transcoder or that the INVITE generated by the transcoder (4) did not reach the callee. To resolve this, it is recommended that the caller uses the reliable provisional responses [3] SIP extension.

Figure 3 shows the resulting message flow when the caller requires the use of the reliable provisional responses [3] SIP extension. The repection of the 183 (Session Progress) reliable provisional response

informs the caller that the transcoder was contacted susccessfully. So, the 404 (Not Found) response indicates that the callee could not be reached.

```
 A                                T                            B
 |                                |                            |
 |--------(1) INVITE SDP A------>|                            |
 |                                |                            |
 |<-(2) 183 S. Prog. SDP on hold-|                            |
 |                                |-----(3) INVITE SDP TB---->|
 |                                |                            |
 |-----------(4) PRACK---------->|                            |
 |                                |                            |
 |<----------(5) 200 OK----------|                            |
 |                                |                            |
 |                                |<----(6) 404 Not Found-----|
 |                                |                            |
 |                                |---------(7) ACK---------->|
 |<-------(8) 404 Not Found------|                            |
 |                                |                            |
 |-------------(9) ACK---------->|                            |
 |                                |                            |
```

Figure 3: Invocation using reliable provisional responses

3 Callee's Invocation

If a UA receives an INVITE with an offer that is not acceptable, it can only invoke a transcoder if the caller supports the Replaces [4] extension. This support is indicated by the Supported header field in the INVITE.

If the caller (A) does not support Replaces, the callee (B) can always reject the session and attempt to establish a new session with A following the procedures in Section 2. This way, B would act as a caller and, consequently, it would follow the procedures for caller's invocation of transcoders.

Assuming that the caller (A) supports Replaces, the callee (B) follows the steps shown in Figure 4 to invoke a transcoder. The callee sends a 183 (Session Progress) response (2) to the caller. This response carries a tag in the To header field. The caller needs to receive this To tag so that this early dialog can be replaced later in (5). So, the callee SHOULD use the reliable provisional

responses [3] SIP extension. The SDP in the 183 (Session Progress) response may put the media streams on hold. If the caller did not support this extension, the callee MAY send a 200 (OK) putting the media streams on hold.

OPEN ISSUE: can we use 0.0.0.0 instead of hold here?

After returning a response with a To tag to the caller, the callee sends an INVITE (2) to the Transcoder. The URI in the Request-URI of this INVITE contains a list parameter, as defined in [2] (draft-camarillo-sipping-uri-list-01), with a pointer to a URI list. This URI list contains a single URI: the URI received in the Contact header field of the initial INVITE (1) with an escaped Replaces header field, as shown in the following example:

```
sip:caller@client.chicago.example.com?Replaces=40d432fa84b4c76e66710;
                ;from-tag=32331
                ;to-tag=12dr45
```

We recommend the use of the reliable provisional responses between the callee and the transcoder so that the callee is able to distinguish between problems with the transcoder and problems with the caller, as we described in Section 2.1.

When A receives this INVITE (5), it replaces the original dialog (1) with this new dialog. The caller sends a CANCEL (10) to cancel the original dialog (1) and receives a 487 (Request Terminated) response (11) from the callee.

4 Security Considerations

TBD.

5 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.

6 OPEN ISSUES

In SIP, the Route header field is used to traverse proxies, but is seems that using it for traversing B2BUAs would be stretching its semantics too much.

```
    A                               T                              B
    |                               |                              |
    |----------------------(1) INVITE SDP A----------------------->|
    |                               |                              |
    |<-------------(2) 183 Session Progress SDP on hold------------|
    |                               |                              |
    |--------------------------(3) PRACK-------------------------->|
    |                               |                              |
    |<-------------------------(4) 200 OK--------------------------|
    |                               |                              |
    |                               |<----(5) INVITE SDP TB--------|
    |                               |                              |
    |                               |-(6) Session Progress SDP TB->|
    |                               |                              |
    |                               |<---------(7) PRACK-----------|
    |                               |                              |
    |                               |----------(8) 200 OK--------->|
    |                               |                              |
    |<------(9) INVITE SDP TA-------|                              |
    |                               |                              |
    |-------(10) 200 OK SDP A------>|                              |
    |                               |                              |
    |<-----------(11) ACK-----------|                              |
    |                               |---------(12) 200 OK--------->|
    |                               |                              |
    |                               |<----------(13) ACK-----------|
    |                               |                              |
    |-------------------------(14) CANCEL------------------------->|
    |                               |                              |
    |<------------------------(15) 200 OK--------------------------|
    |                               |                              |
    |<------------------(16) 407 Request Terminated----------------+
    |                              |                               |
    |---------------------------(17) ACK-------------------------->|
    |                               |                              |
    | ***************************** | **************************** |
    |**          Media            **|**            Media         **|
    | ***************************** | **************************** |
```

   Figure 4: Callee's invocation of a transcoder

7 Authors' Addresses

   Gonzalo Camarillo
   Ericsson

Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland
electronic mail: Gonzalo.Camarillo@ericsson.com

8 Bibliography

[1] G. Camarillo, "Framework for transcoding with the session initiation protocol," Internet Draft draft-camarillo-sipping-transc-framework-00, Internet Engineering Task Force, Aug. 2003. Work in progress.

[2] G. Camarillo, "Providing a session initiation protocol (SIP) application server with a list of URIs," Internet Draft draft-camarillo-sipping-uri-list-00, Internet Engineering Task Force, Nov. 2003. Work in progress.

[3] J. Rosenberg and H. Schulzrinne, "Reliability of provisional responses in session initiation protocol (SIP)," RFC 3262, Internet Engineering Task Force, June 2002.

[4] B. Biggs, R. W. Dean, and R. Mahy, "The session inititation protocol (SIP) Engineering Task Force, Aug. 2003. Work in progress.