Internet Engineering Task Force
Internet Draft

SIP WG
G. Camarillo
Ericsson
E. Burger
SnowShore Networks
H. Schulzrinne
Columbia University
A. van Wijk
Viataal

draft-camarillo-sip-deaf-02.txt
February 17, 2003
Expires: August, 2003

Transcoding Services Invocation in the Session Initiation Protocol

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

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

Abstract

This document describes how to discover the need of transcoding services in a session established with SIP and how to invoke those transcoding services. Two models for transcoding services invocation are introduced; the conference bridge model and the third party call control model. Both models meet the requirements for SIP regarding transcoding services invocation to support deaf, hard of hearing and speech-impaired individuals.

Table of Contents

1 Introduction

Two user agents involved in a SIP [1] dialog may find it impossible to establish a media session due to a variety of incompatibilities. Assuming that both user agents understand the same session description format (e.g., SDP), incompatibilities can be found at the user agent level and at the user level. At the user agent level, both terminals may not support any common codec or may not support common media types (e.g., a text-only terminal and an audio-only terminal). At the user level, a deaf person will not be able to understand what it is said over an audio stream.

In order to make communications possible in the presence of incompatibilities, user agents need to introduce intermediaries that provide transcoding services to a session. From the SIP point of view, the introduction of a transcoder is done in the same way to resolve both user level and user agent level incompatibilities. Therefore, the invocation mechanisms described in this document are generally applicable to any type of incompatibility related to how the information that needs to be communicated is encoded.

This document does not describe media server discovery. That is an orthogonal problem that one can address using user agent provisioning or other methods.

All the examples provided in this document use the Session Description Protocol (SDP) [2]. However, other session description formats can be used with the same call flows.

The remainder of this document is organized as follows. Section 2 deals with the discovery of the need of transcoding services for a particular session. Section 3.2 introduces the conference bridge transcoding invocation model, and Section 3.3 introduces the third party call control model. Both models meet the requirements regarding transcoding services invocation in RFC3351 [3] to support deaf, hard of hearing and speech-impaired individuals.

2 Discovery of the Need for Transcoding Services

Following the one-party consent model defined in RFC 3238 [4], transcoding invocation is best performed by one of the end-points involved in the communication. Following the same principle, one of the end-points should be the one detecting that transcoding is needed for a particular session.

In order to decide whether or not transcoding is needed, a user agent needs to know the capabilities of the remote user agent. A user agent acting as an offerer typically obtains this knowledge by downloading

a presence document that includes media capabilities (e.g., Bob is available on a terminal that only supports audio) or by getting an SDP description of media capabilities as defined in RFC 3264 [5]. Presence documents are typically received in a NOTIFY request and SDP media capabilities descriptions are typically received in a 200 (OK) response to an OPTIONS request or in a 488 (Not Acceptable Here) response to an INVITE.

A user agent client acting as an answerer typically gets an offer that it cannot accept. The user agent can send back a media capabilities description hoping that the offerer will invoke some type of transcoding services or it can invoke transcoding services itself.

It is recommended that an offerer does not invoke transcoding services before making sure that the answerer does not support the capabilities needed for the session. Making wrong assumptions about the answerer's capabilities can lead to situations where two transcoders are introduced (one by the offerer and one by the answerer) in a session that would not need any transcoding services at all.

> An example of the situation above is a call between two GSM phones (without using transcoding-free operation). Both phones use a GSM codec, but the speech is converted from GSM to PCM by the originating MSC and from PCM back to GSM by the terminating MSC.

Note that transcoding services can be symmetric (e.g., speech-to-text plus text-to-speech) or asymmetric (e.g., a one-way speech-to-text transcoding for a hearing impaired user that can talk).

3 Transcoding Services Invocation

Once the need for transcoding for a particular session has been identified as described in Section 2, one of the user agents needs to invoke transcoding services.

Invoking transcoding services from a server (T) for a session between two user agents (A and B) involves establishing two media sessions; one between A and T and another between T and B. How to invoke T's services (i.e., how to establish both A-T and T-B sessions) depends on how we model the transcoding service. We have considered two models for invoking a transcoding service. The first is to use a (dial-in and/or dial-out) conference bridge that negotiates the appropriate media parameters on each individual leg (i.e., A-T and T-B). The second is to use third party call control [6], also referred to as 3pcc, to invoke the transcoding service. Section 3.2

describes the conference bridge transcoding invocation model, and Section 3.3 describes the third party call control model.

3.1 Terminology

All the figures in this document follow the naming convention below:

SDP A: A session description generated by A. It contains, among other things, the transport address/es (IP address and port number) where A wants to receive media for each particular stream.

SDP B: A session description generated by B. It contains, among other things, the transport address/es where B wants to receive media for each particular stream.

SDP A+B: A session description that contains, among other things, the transport address/es where A wants to receive media and the transport address/es where B wants to receive media.

SDP TA: A session description generated by T and intended for A. It contains, among other things, the transport address/es where T wants to receive media from A.

SDP TB: A session description generated by T and intended for B. It contains, among other things, the transport address/es where T wants to receive media from B.

SDP TA+TB: A session description generated by T that contains, among other things, the transport address/es where T wants to receive media from A and the transport address/es where T wants to receive media from B.

3.2 Conference Bridge Transcoding Model

A conference server typically establishes an audio stream with each participant of a conference. The server sends over each individual stream the media received over the rest of the streams, typically performing some mixing. The conference server may have to send audio to different participants using different audio codecs. We can think of a transcoding service as a two-party conference server that may change not only the codec in use, but also the format of the media (e.g., audio to text). Using this model, the whole A-T-B session is established in the same way as a conference [7]. Typically, the user agent invoking the transcoding service sets up the media policy at the bridge (possibly using a media policy control protocol) and sends an INVITE to join the conference. The media policy for the session

determines the type of transcoding the bridge will perform.

Once the conference is set up and the invoker has joined it, the remote user has to be added as a participant as well. Users have two options to join a conference. A user can dial-in (i.e., send an INVITE request to the conference bridge) to join a conference, or the conference bridge can dial-out (i.e., send an INVITE request to the user) to add the user to the conference. Both dial-in and dial-out approaches are discussed in the following sections. Section 3.2.1 deals with caller's invocation and Section 3.2.2 deals with callee's invocation of the service.

3.2.1 Caller's Invocation

Once the caller has set up the conference bridge and joined the conference by sending an INVITE to the bridge, it has two options to add the callee to the session; sending a REFER [8] to the bridge (that will instruct the bridge to dial-out) or sending a REFER to the callee (that will instruct the callee to dial-in).

We recommend the first option (i.e., REFER sent to the bridge). The bridge, upon reception of the REFER, generates an INVITE towards the callee. The session description of the INVITE is generated according to the media policy set up by the caller. Figure 1 shows this scenario's message flow.

Note that if the caller chooses to send the REFER directly to the callee (rather than to the bridge) the callee may generate an INVITE with a session description that contained media types the bridge was not configured to handle. In addition to that, some user agents may not support REFER or may not be able to handle out-of-the-blue REFER requests.

3.2.2 Callee's Invocation

Similarly to the situation above, once the callee has set up the conference bridge and joined the conference by sending an INVITE to the bridge, it has two options to add the caller to the session; sending a REFER to the bridge (that will instruct the bridge to dial-out) or sending a REFER to the caller (that will instruct the caller to dial-in).

We recommend the first option (i.e., REFER sent to the bridge). The bridge, upon reception of the REFER, generates an INVITE with a Replaces header field [9] header field towards the callee. The session description of the INVITE is generated according to the media policy set up by the callee. Figure 2 shows this scenario's message

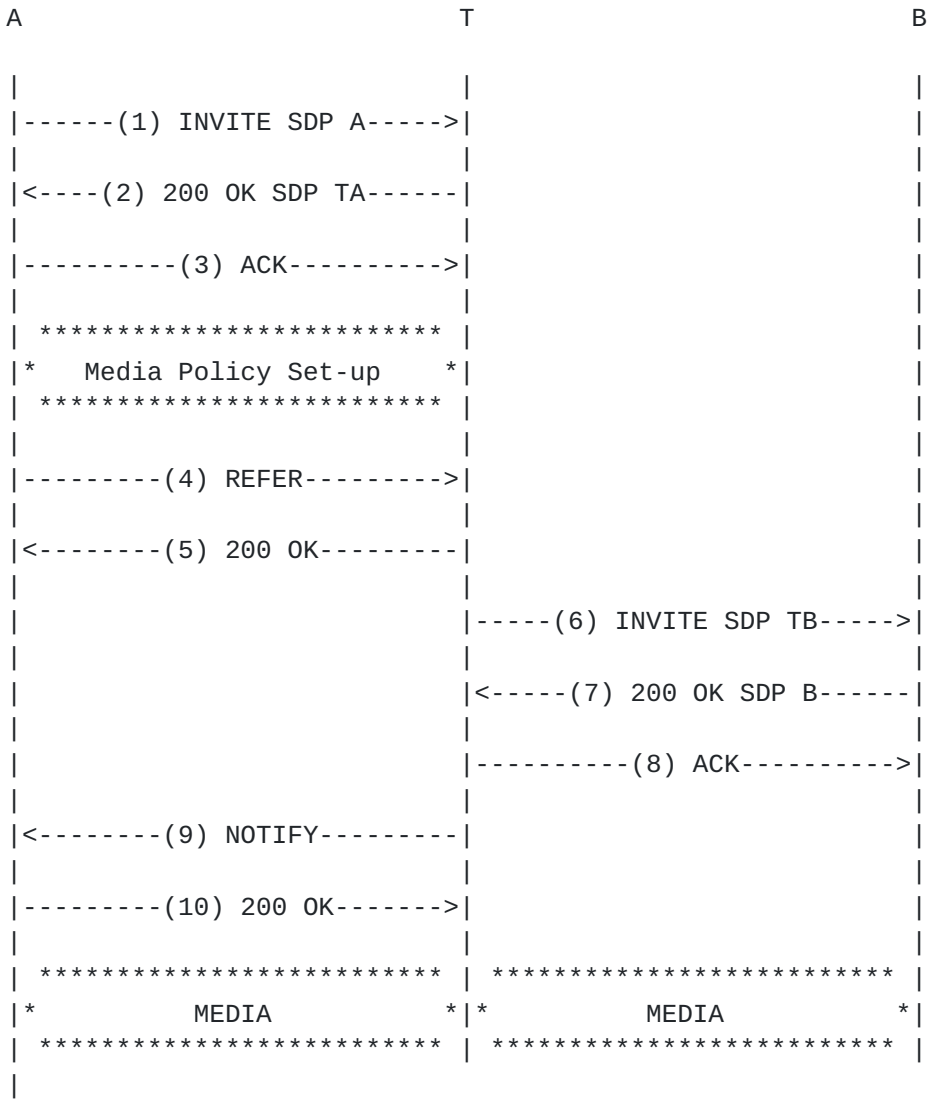

Figure 1: Caller's invocation of a conference bridge

flow.

The flow in Figure 2 requires that the caller supports the Replaces header field. If the caller does not support it, the callee can send a 488 (Not Accpetable Here) for the original INVITE and attempt to establish the session acting as a caller (i.e., sending a new INVITE).

Sending the REFER to the caller (instead of to the bridge) introduces

a number of issues, since there is currently no way for the callee to inform the caller that the newly established session will substitute the original session.

3.3 Third Party Call Control Transcoding Model

If we model T as a transcoding service rather than a special case of a conferencing server, a single INVITE transaction from the invoker of the service provides T with both A's and B's session descriptions. In order to provide in a single session description information about media streams that belong to different entities (A and B), the session description format in use should provide a means to define how these streams should be mapped. For instance, in a session description with two audio streams and one text stream, a possible mapping would be the following; the information received over the first audio stream should be sent over the text stream and over the second audio stream, and the incoming text should be sent only over the first audio stream. SDP [2] can convey this information using the source and sink attributes [10].

As stated previously, the invocation of a transcoding service consists of establishing two sessions; A-T and T-B. How these sessions are established depends on which party, the caller (A) or the callee (B), invokes the transcoding services. However, we have followed a general principle to design our 3pcc flows; a 200 (OK) response from the transcoding service have to be received before contacting the callee. This tries to ensure that the transcoding service will be available when the callee accepts the session.

However, note that the transcoding service does not know the exact type of transcoding it will be performing until the callee accepts the session. Therefore, there are always changes of failing to provide transcoding services after the callee has accepted the session. A system with tough requirements could use preconditions to avoid this situation. When preconditions are used, the callee is not alerted until everything is ready for the session.

3.3.1 Callee's Invocation

In this scenario, B receives an INVITE from A, and B decides to introduce T in the session. Figure 3 shows the call flow for this scenario.

In Figure 3 A can both hear and speak and B is a deaf user with a speech impairment. A proposes to establish a session that consists of an audio stream (1). B wants to send and receive only text, so it invokes a transcoding service T that will perform both speech-to-text

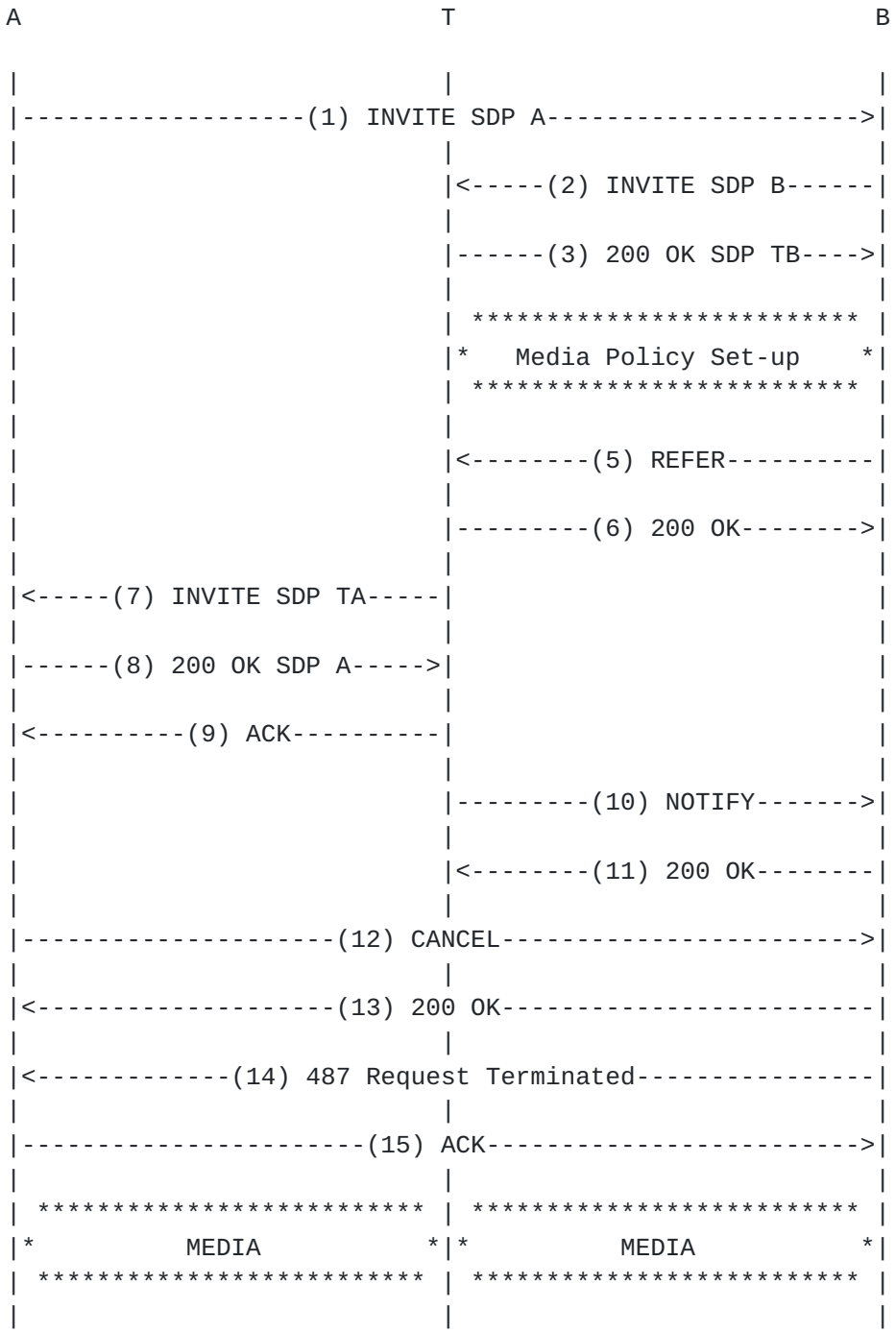
A
T
B
(1) INVITE SDP A
(2) INVITE SDP B
(3) 200 OK SDP TB
Media Policy Set-up
(5) REFER
(6) 200 OK
(7) INVITE SDP TA
(8) 200 OK SDP A
(9) ACK
(10) NOTIFY
(11) 200 OK
(12) CANCEL
(13) 200 OK
(14) 487 Request Terminated
(15) ACK
MEDIA
MEDIA

Figure 2: Conference bridge transcoding model

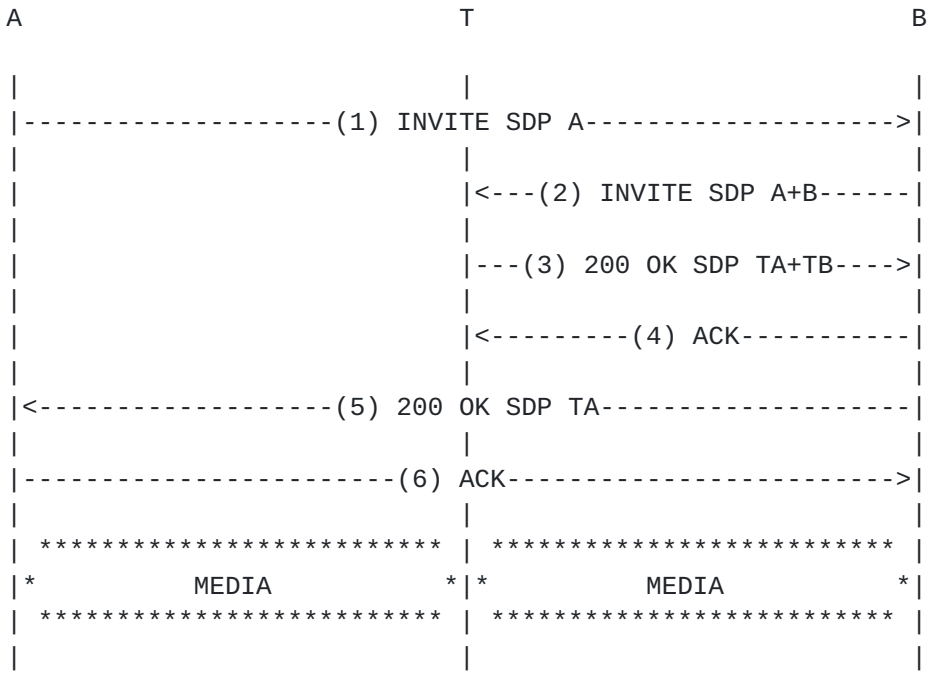

Figure 3: Callee's invocation of a transcoding service

and text-to-speech conversions (2). The session descriptions of Figure 3 are partially shown below.

(1) INVITE SDP A

```
m=audio 20000 RTP/AVP 0
c=IN IP4 A.domain.com
```

(2) INVITE SDP A+B

```
m=audio 20000 RTP/AVP 0
c=IN IP4 A.domain.com
a=source:1
a=sink:2
m=text 40000 RTP/AVP 96
c=IN IP4 B.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

(3) 200 OK SDP TA+TB

```
m=audio 30000 RTP/AVP 0
c=IN IP4 T.domain.com
a=source:1
a=sink:2
m=text 30002 RTP/AVP 96
c=IN IP4 T.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

(5) 200 OK SDP TA

```
m=audio 30000 RTP/AVP 0
c=IN IP4 T.domain.com
```

Four media streams (i.e., two bi-directional streams) have been established at this point:

1. Audio from A to T.domain.com:30000
2. Text from T to B.domain.com:40000
3. Text from B to T.domain.com:30002
4. Audio from T to A.domain.com:20000

When either A or B decide to terminate the session, B will send a BYE to T indicating that the session is over.

If the first INVITE (1) received by B is empty (no session description), the call flow is slightly different. Figure 4 shows the messages involved.

B may have different reasons for invoking T before knowing A's session description. B may want to hide its capabilities, and therefore it wants to return a session description with all the codecs B supports plus all the codecs T supports. Or T may provide recording services (besides transcoding), and B wants T to record the conversation, regardless of whether or not transcoding is needed.

This scenario (Figure 4) is a bit more complex than the previous one.

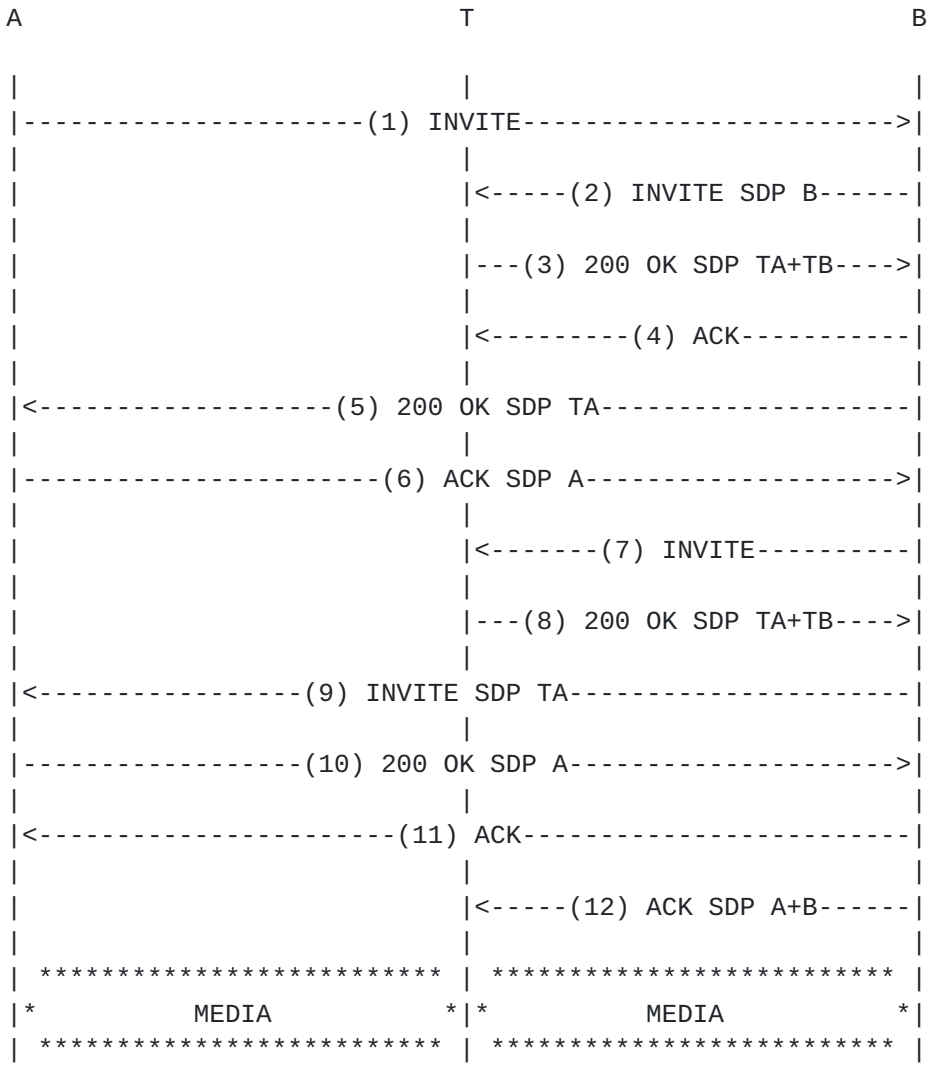

Figure 4: Callee's invocation after initial INVITE without SDP

In INVITE (2), B still does not have SDP A, so it cannot provide T with that information. When B finally receives SDP A in (6), it has to send it to T. B sends an empty INVITE to T (7) and gets a 200 OK with SDP TA+TB (8). In general, this SDP TA+TB can be different than the one that was sent in (3). That is why B needs to send the updated SDP TA to A in (9). A then sends a possibly updated SDP A (10) and B sends it to T in (12). However, if T happens to return the same SDP TA+TB in (8) as in (3), B can skip messages (9), (10) and (11). Therefore, implementors of transcoding services are encouraged to return the same session description in (8) as in (3) in this type of scenario. The session descriptions of this flow are shown below:

(2) INVITE SDP A+B

```
m=audio 20000 RTP/AVP 0
c=IN IP4 0.0.0.0
a=source:1
a=sink:2
m=text 40000 RTP/AVP 96
c=IN IP4 B.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

(3) 200 OK SDP TA+TB

```
m=audio 30000 RTP/AVP 0
c=IN IP4 T.domain.com
a=source:1
a=sink:2
m=text 30002 RTP/AVP 96
c=IN IP4 T.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

(5) 200 OK SDP TA

```
m=audio 30000 RTP/AVP 0
c=IN IP4 T.domain.com
```

(6) ACK SDP A

```
m=audio 20000 RTP/AVP 0
c=IN IP4 A.domain.com
```

(8) 200 OK SDP TA+TB

```
m=audio 30004 RTP/AVP 0
c=IN IP4 T.domain.com
a=source:1
a=sink:2
```

```
m=text 30006 RTP/AVP 96
c=IN IP4 T.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

(9) INVITE SDP TA

```
m=audio 30004 RTP/AVP 0
c=IN IP4 T.domain.com
```

(10) 200 OK SDP A

```
m=audio 20002 RTP/AVP 0
c=IN IP4 A.domain.com
```

(12) ACK SDP A+B

```
m=audio 20002 RTP/AVP 0
c=IN IP4 A.domain.com
a=source:1
a=sink:2
m=text 40000 RTP/AVP 96
c=IN IP4 B.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

Four media streams (i.e., two bi-directional streams) have been established at this point:

1. Audio from A to T.domain.com:30004
2. Text from T to B.domain.com:40000
3. Text from B to T.domain.com:30006
4. Audio from T to A.domain.com:20002

3.3.2 Caller's Invocation

In this scenario, A wishes to establish a session with B using a transcoding service. A uses 3pcc to set up the session between T and B. The call flow we provide here is slightly different than the ones in [6]. In [6], the controller establishes a session between two user agents, being the user agents the ones deciding the characteristics of the streams. Here, A wants to establish a session between T and B, but A wants to decide how many and which types of streams are established. That is why A sends its session description in the first INVITE (1) to T, as opposed to the media-less initial INVITE recommended by [6]. Figure 5 shows the call flow for this scenario.

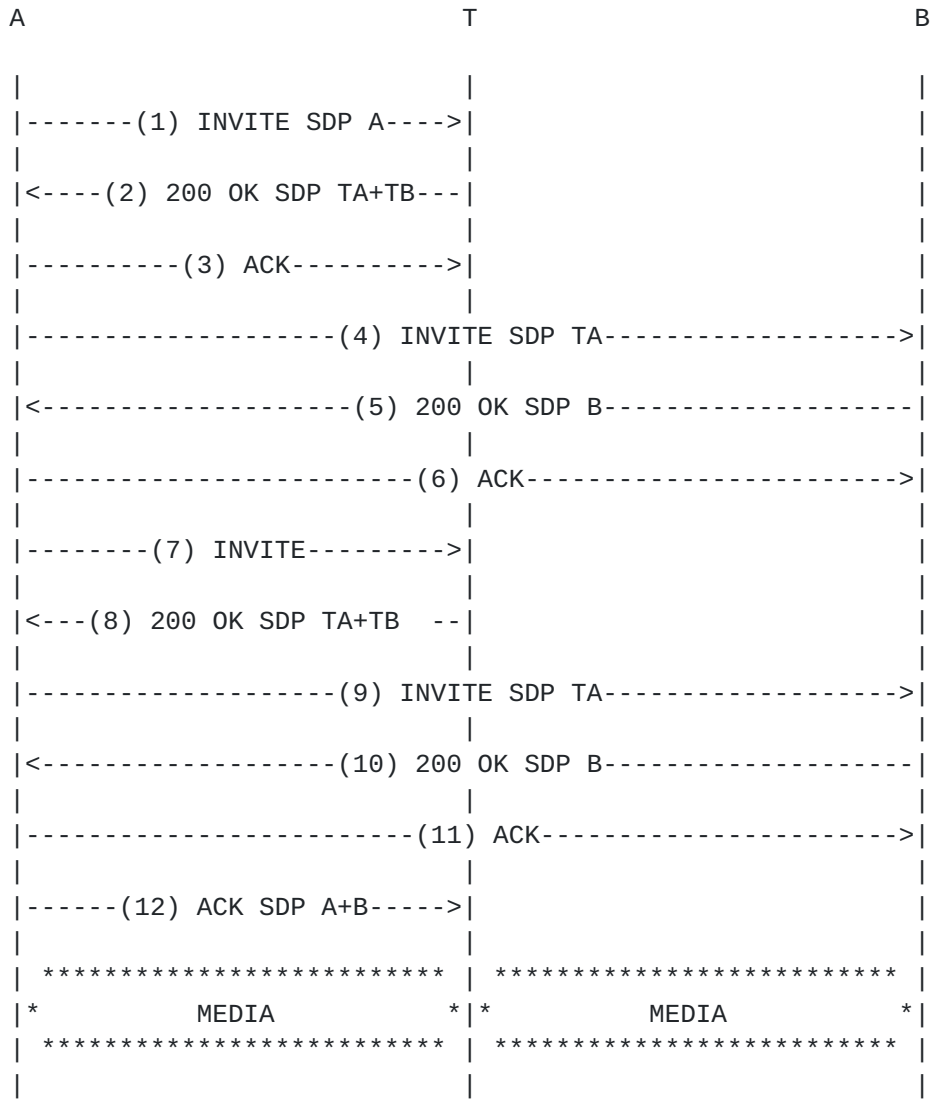

Figure 5: Caller's invocation of a transcoding service

We do not include the session descriptions of this flow, since they are very similar to the ones in Figure 4. In this flow, if T returns the same SDP TA+TB in (8) as in (2), messages (9), (10) and (11) can be skipped.

3.3.3 Receiving the Original Stream

Sometimes, as pointed out in the requirements for SIP in support of deaf, hard of hearing and speech-impaired individuals [3], a user wants to receive both the original stream (e.g., audio) and the transcoded stream (e.g., the output of the speech-to-text conversion). There are various possible solutions for this problem. One solution consists of using the SDP group attribute with FID semantics [11]. FID allows requesting that a stream is sent to two different transport addresses in parallel, as shown below:

```
a=group:FID 1 2
m=audio 20000 RTP/AVP 0
c=IN IP4 A.domain.com
a=mid:1
m=audio 30000 RTP/AVP 0
c=IN IP4 T.domain.com
a=mid:2
```

The problem with this solution is that the majority of the SIP user agents do not support FID. And even if FID is supported, many user agents do not support sending simultaneous copies of the same media stream at the same time. In addition to that, both copies of the stream need to use the same codec.

Therefore, we recommend that T (instead of a user agent) replicates the media stream. The following session description requests T to perform speech-to-text and text-to-speech conversions between the first audio stream and the text stream. In addition, it requests T to copy of the first audio stream to the second audio stream and send it to A.

```
m=audio 40000 RTP/AVP 0
c=IN IP4 B.domain.com
a=source:1
a=sink:2
m=audio 20000 RTP/AVP 0
c=IN IP4 A.domain.com
a=recvonly
a=sink:1
m=text 20002 RTP/AVP 96
```

```
c=IN IP4 A.domain.com
a=rtpmap:96 t140/1000
a=source:2
a=sink:1
```

3.3.4 Transcoding Services in Parallel

Transcoding services sometimes consist of human relays (e.g., a person performing speech-to-text and text-to-speech conversions for a session). If the same person is involved in both conversions (i.e., from A to B and from B to A), he or she has access to all the conversation. In order to provide some degree of privacy, sometimes two different persons are allocated to do the job (i.e., one person handles A->B and the other B->A). This type of disposition is also useful for automated transcoding services, where one machine converts text to synthetic speech (text-to-speech) and a different machine performs voice recognition (speech-to-text).

The scenario just described involves four different sessions; A-T1, T1-B, B-T2 and T2-A. Figure 6 shows the call flow where A invokes T1 and T2.

(1) INVITE SDP AT1

```
m=text 20000 RTP/AVP 96
c=IN IP4 A.domain.com
a=rtpmap:96 t140/1000
a=sendonly
a=source:1
m=audio 20000 RTP/AVP 0
c=IN IP4 0.0.0.0
a=recvonly
a=sink:1
```

(2) INVITE SDP AT2

```
m=text 20002 RTP/AVP 96
c=IN IP4 A.domain.com
a=rtpmap:96 t140/1000
a=recvonly
a=sink:1
m=audio 20000 RTP/AVP 0
c=IN IP4 0.0.0.0
```

```
a=sendonly
a=source:1
```

(3) 200 OK SDP T1A+T1B

```
m=text 30000 RTP/AVP 96
c=IN IP4 T1.domain.com
a=rtpmap:96 t140/1000
a=recvonly
a=source:1
m=audio 30002 RTP/AVP 0
c=IN IP4 T1.domain.com
a=sendonly
a=sink:1
```

(5) 200 OK SDP T2A+T2B

```
m=text 40000 RTP/AVP 96
c=IN IP4 T2.domain.com
a=rtpmap:96 t140/1000
a=sendonly
a=sink:1
m=audio 40002 RTP/AVP 0
c=IN IP4 T2.domain.com
a=recvonly
a=source:1
```

(7) INVITE SDP T1B+T2B

```
m=audio 30002 RTP/AVP 0
c=IN IP4 T1.domain.com
a=sendonly
m=audio 40002 RTP/AVP 0
c=IN IP4 T2.domain.com
a=recvonly
```

(8) 200 OK SDP BT1+BT2

```
m=audio 50000 RTP/AVP 0
c=IN IP4 B.domain.com
```

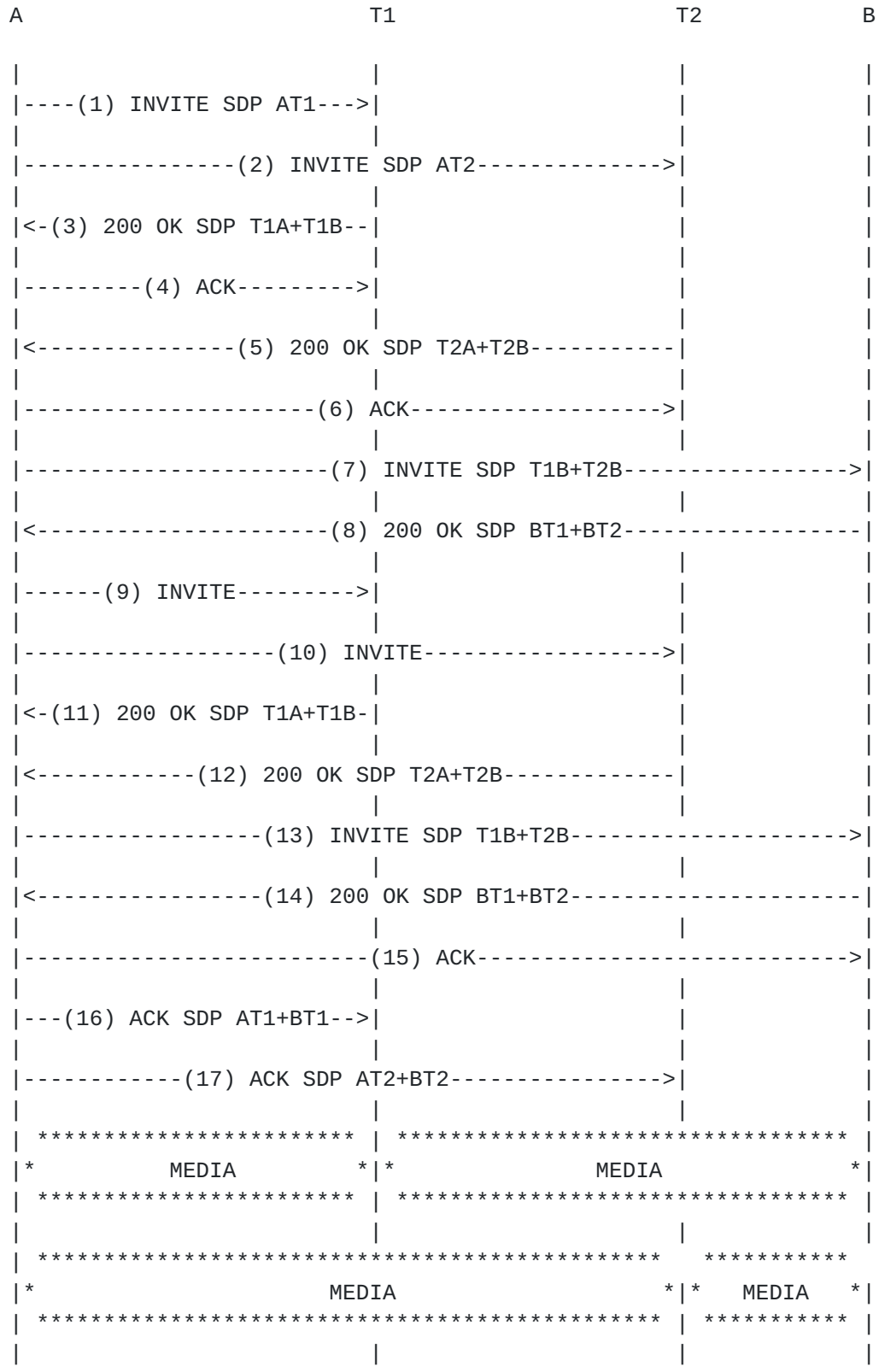
A
T1
T2
B
(1) INVITE SDP AT1
(2) INVITE SDP AT2
(3) 200 OK SDP T1A+T1B
(4) ACK
(5) 200 OK SDP T2A+T2B
(6) ACK
(7) INVITE SDP T1B+T2B
(8) 200 OK SDP BT1+BT2
(9) INVITE
(10) INVITE
(11) 200 OK SDP T1A+T1B
(12) 200 OK SDP T2A+T2B
(13) INVITE SDP T1B+T2B
(14) 200 OK SDP BT1+BT2
(15) ACK
(16) ACK SDP AT1+BT1
(17) ACK SDP AT2+BT2
MEDIA
MEDIA
MEDIA
MEDIA

Figure 6: Transcoding services in parallel

```
a=recvonly
m=audio 50002 RTP/AVP 0
c=IN IP4 B.domain.com
a=sendonly
```

(11) 200 OK SDP T1A+T1B

```
m=text 30000 RTP/AVP 96
c=IN IP4 T1.domain.com
a=rtpmap:96 t140/1000
a=recvonly
a=source:1
m=audio 30002 RTP/AVP 0
c=IN IP4 T1.domain.com
a=sendonly
a=sink:1
```

(12) 200 OK SDP T2A+T2B

```
m=text 40000 RTP/AVP 96
c=IN IP4 T2.domain.com
a=rtpmap:96 t140/1000
a=sendonly
a=sink:1
m=audio 40002 RTP/AVP 0
c=IN IP4 T2.domain.com
a=recvonly
a=source:1
```

Since T1 have returned the same SDP in (11) as in (3) and T2 has returned the same SDP in (12) as in (5), messages (13), (14) and (15) can be skipped.

(16) ACK SDP AT1+BT1

```
m=text 20000 RTP/AVP 96
c=IN IP4 A.domain.com
a=rtpmap:96 t140/1000
a=sendonly
a=source:1
m=audio 50000 RTP/AVP 0
c=IN IP4 B.domain.com
```

```
a=recvonly
a=sink:1
```

(17) ACK SDP AT2+BT2

```
m=text 20002 RTP/AVP 96
c=IN IP4 A.domain.com
a=rtpmap:96 t140/1000
a=recvonly
a=sink:1
m=audio 50002 RTP/AVP 0
c=IN IP4 B.domain.com
a=sendonly
a=source:1
```

Four media streams have been established at this point:

1. Text from A to T1.domain.com:30000

2. Audio from T1 to B.domain.com:50000

3. Audio from B to T2.domain.com:40002

4. Text from T2 to A.domain.com:20002

Note that B, the user agent server, needs to support two media streams; one sendonly and the other recvonly. At present, some user agents, although they support a single sendrecv media stream, they do not support a different media line per direction. Implementers are encouraged to build support for this feature.

3.3.5 Transcoding Services in Serial

In a distributed environment, a complex transcoding service (e.g., English text to Spanish speech) is often provided by several servers. For example, one server performs English text to Spanish text translation, and its output is feed into a server that performs text-to-speech conversion. The flow in Figure 7 shows how A invokes T1 and T2.

4 Security Considerations

This document describes how to use the REFER method and third party

call control to invoke transcoding services. It does not introduce new security considerations besides the ones discussed in [8] and [6].

5 TODO List

We need to see whether or not it is possible to use the media policy work in the 3pcc model as well (instead of source/sink).

6 Authors' Addresses

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

Eric W. Burger
SnowShore Networks, Inc.
Chelmsford, MA
USA
electronic mail: eburger@snowshore.com

Henning Schulzrinne
Dept. of Computer Science
Columbia University 1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA
electronic mail: schulzrinne@cs.columbia.edu

Arnoud van Wijk
Viataal
Research & Development
Afdeling RDS
Theerestraat 42
5271 GD Sint-Michielsgestel
The Netherlands
electronic mail: a.vwijk@viataal.nl

7 Bibliography

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol," RFC 3261, Internet Engineering Task Force, June 2002.

[2] M. Handley and V. Jacobson, "SDP: session description protocol,"

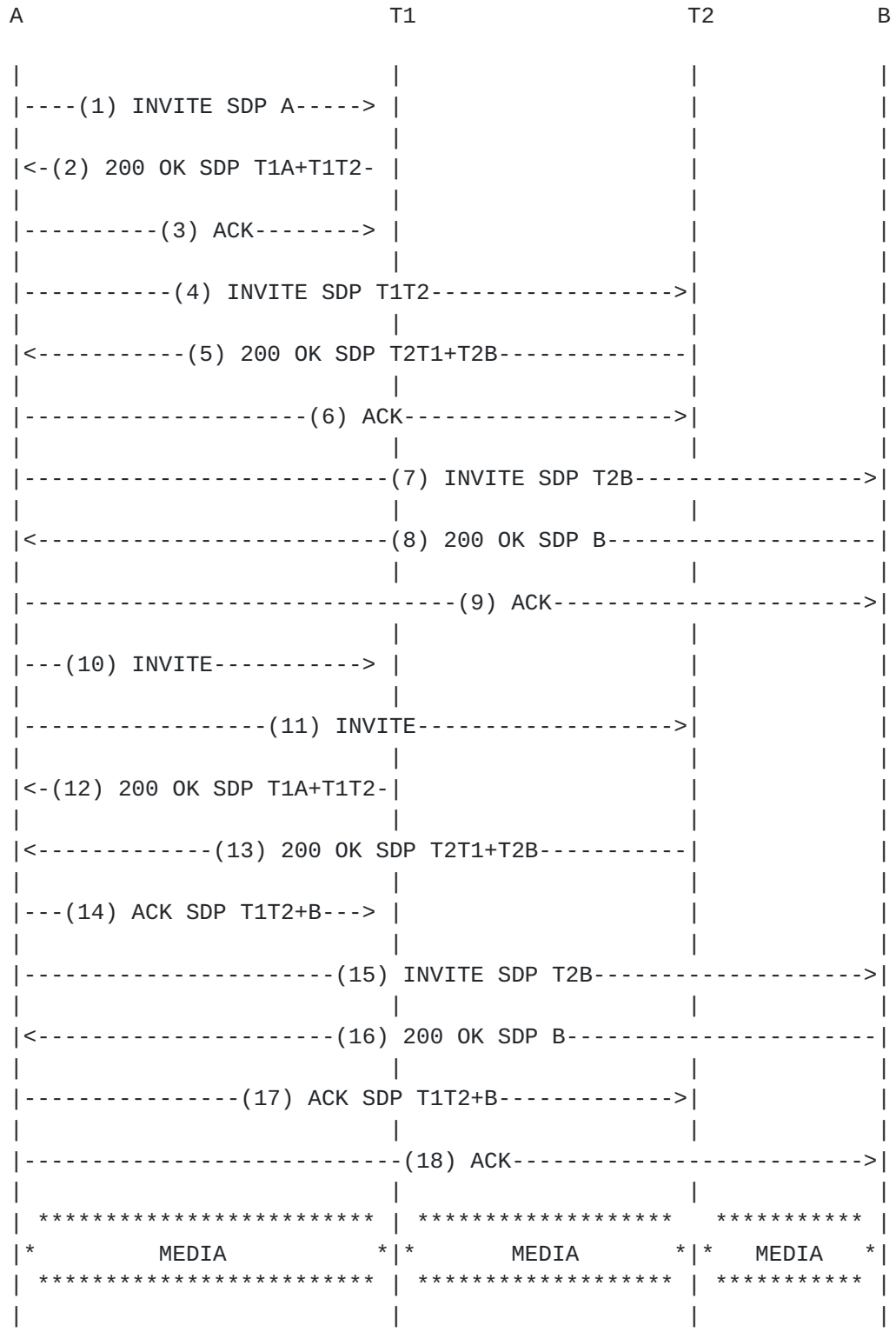

Figure 7: Transcoding services in serial

RFC 2327, Internet Engineering Task Force, Apr. 1998.

[3] N. Charlton, M. Gasson, G. Gybels, M. Spanner, and A. van Wijk,

RFC 3351, Internet Engineering Task Force, Aug. 2002.

[4] S. Floyd and L. Daigle, "IAB architectural and policy considerations for open pluggable edge services," RFC 3238, Internet Engineering Task Force, Jan. 2002.

[5] J. Rosenberg and H. Schulzrinne, "An offer/answer model with session description protocol (SDP)," RFC 3264, Internet Engineering Task Force, June 2002.

[6] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, "Best current practices for third party call control in the session initiation protocol," internet draft, Internet Engineering Task Force, June 2002. Work in progress.

[7] J. Rosenberg, "A framework for conferencing with the session initiation protocol," internet draft, Internet Engineering Task Force, Nov. 2002. Work in progress.

[8] R. Sparks, "The SIP refer method," internet draft, Internet Engineering Task Force, Dec. 2002. Work in progress.

[9] B. Biggs, R. Dean, and R. Mahy, "The session inititation protocol (SIP)," internet draft, Internet Engineering Task Force, May 2002. Work in progress.

[10] G. Camarillo, H. Schulzrinne, and E. Burger, "The source and sink attributes for the session description protocol," internet draft, Internet Engineering Task Force, Sept. 2002. Work in progress.

[11] G. Camarillo, J. Holler, G. Eriksson, and H. Schulzrinne, "Grouping of m lines in SDP," internet draft, Internet Engineering Task Force, Feb. 2002. Work in progress.

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 assigns.

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.