

SIPPING
Internet-Draft
Intended status: Informational
Expires: August 16, 2007

B. Stucker
Nortel
February 12, 2007

Early Media INDirection (EMIND) in the Session Initiation Protocol (SIP)
[draft-stucker-sipping-early-media-indirection-00](#)

Status of this Memo

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 becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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 August 16, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes models that can be used to manage multiple early media streams in the Session Initiation Protocol (SIP). The model seeks to allow calling party endpoints to be able to better control the rendering of early media to the user, and distinguish early media from final media.

Although some of early media challenges are likely to never be overcome, (e.g. when interworking with an ISUP PSTN gateway that does

Internet-Draft

Early Media Indirection in SIP

February 2007

not take into account CPG or ACM messages), the potential to improve on what is already there does exist. The recommendations in this document are intended to be an update to [RFC-3960](#) recommendations and to avoid many of the complications outlined in [draft-stucker-sipping-early-media-coping](#).

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Definitions	4
4.	Requirements	4
5.	Early Media Indirection	5
6.	Early Media Indirection Operation	6
6.1.	The Early-Media Header	6
6.2.	User Agent Client Procedures	7
6.2.1.	Indicating support for Early Media Indirection (EMIND)	7
6.2.2.	Accepting indirect early-media streams	7
6.2.3.	Closing indirect early-media streams	8
6.2.4.	Switching from early-media to final-media	8
6.3.	Proxy/User Agent Server Procedures	8
6.3.1.	Offering indirect early media streams	9
6.3.2.	Closing indirect early media streams	9
6.3.3.	Ordering early media streams	10
7.	Example Call Flow	10
8.	Security Considerations	16
9.	IANA Considerations	16
10.	References	17
10.1.	Normative References	17
10.2.	Informational References	17
	Author's Address	17
	Intellectual Property and Copyright Statements	18

1. Introduction

Establishment of one or more media streams within SIP [[RFC3261](#)] using the SDP offer/answer model [[RFC3264](#)] typically involves two phases: early media and final media. Early media is media that is generated prior to a particular SIP session being accepted by a called endpoint. Final media differs from early media in that it is generated only after the SIP session has been accepted by a called endpoint. It is important to note, that although [[RFC3261](#)] does not require that the originator/UAC actually render ANY early media sent to them (just that they be prepared to receive it). However, in order to prevent media clipping (see [section 2 of \[RFC3960\]](#)), UACs nearly universally do try to render early media streams to the originating user in practice.

There are many different reasons that early media may be sent to the calling party. For example: to announce their current account balance, to provide customized ringback, or to present a brief tone identifying the service provider. In each of these cases, an application server may intercept the INVITE and involve a media server into the call. As a result of proxy and/or client interactions (which are explained later in this document) the normal negotiation of SDP between the calling party and the called party's endpoint may be impacted negatively.

Another potentially troublesome scenario arises when the INVITE for the call is forked by one or more proxies. This can create a condition where multiple early media streams arrive at the UAC, possibly prior to the SDP offer/answer exchange being completed. This scenario has been previously discussed in [[RFC3960](#)], and in particular by the definition of the gateway model given in that document. Despite the previous treatment on the topic, there remains room for improvement.

Finally, there is the problem of early media and interworking scenarios. This is typically viewed as a problem with SIP to PSTN

interworking, but the reverse (where a PSTN user is trying to contact a SIP user) can also cause problems. When other protocols are interworked with SIP it may become difficult or impossible to know the reason why a particular early media stream is being generated and what the correct course of action is at the UAC. For example, ISUP has no indication in the CPG or ACM messages as to what is likely in a particular media stream is being sent to the originator for presentation to the calling party.

This document seeks to improve upon each of these scenarios, where possible, by updating the recommendations from [[RFC3960](#)], providing a mechanism to help order early media streams when forking has

occurred, and separate early media negotiation triggered by proxies from SDP negotiation with the called party's end device(s). Further information about the various problems currently caused by existing early media processing mechanisms can be found in [[I-D.stucker-sipping-early-media-coping](#)].

[2.](#) Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#), [RFC 2119](#) [3].

[3.](#) Definitions

This document uses the following terms:

- o End-to-end early media - Media generated by the called party's end client device prior to the call being answered at that device. For example, colorful ringback tones streamed to the originator that are generated by the terminating end user's phone.
- o End-to-middle early media - Media that is generated by an application server that is intended to cease upon arrival of the 200 OK to the INVITE by a device controlled by the called party. For example, account balance information, branding, colorful ringback tones, etc.
- o Transitional media - End-to-end final media generated by the UAS after the 200 OK, but arrives at the UAC before the 200 OK. In

effect the signaling is racing with the media to get to the UAC. For example, the called party has answered the call, and is saying "Hello", but the UAC doesn't yet know that the call has been answered due to the decoupled signaling and media paths having different latencies.

- o End-to-end in-band signaling - Signaling that is done end-to-end in the media path instead of through SIP. For example, ICE connectivity checks, or in-band SRTP handshake messages.
- o Non-SDP early media - Ringback information sent in 180 messages to the UAC per [[RFC3261](#)].

[4.](#) Requirements

The following requirements are considered to be the starting point in more formally discussing improvements to SIP for early media interactions:

- R1: A mechanism should exist by which a UAC can signal to a particular UAS that it is now willing to accept early media, without increasing the likelihood of transitional media clipping.
- R2: A mechanism should exist by which a proxy or UAS can signal to a particular UAC what type of early media (if known) it wishes to present to the UAC. Additionally the proxy or UAS should be able to denote if it requires one-way, or two-way media flows, and the relative importance of the early media.
- R3: Any mechanism proposed should try, to the greatest extent possible, be simple to implement, and reuse the existing design patterns within [[RFC3261](#)], [[RFC3264](#)] and [[RFC3960](#)].
- R4: The mechanism must be able to deal with recursive forking scenarios. This is where an INVITE passes two or more proxies that both choose to fork the request to two or more endpoints at each proxy in parallel.
- R5: The mechanism must not require exchange of packets on the media path to identify or coordinate early media streams as this may not interoperate with common network media gating mechanisms.

[5.](#) Early Media Indirection

In order to satisfy all of the requirements outlined in this document and [[I-D.stucker-sipping-early-media-coping](#)], a new mechanism is proposed to manage early media negotiation and rendering: early media indirection. Instead of negotiating both early and final media parameters within the same SIP dialog, proxies and endpoints that wish to generate early-media include a special Early-Media header in any 18x response to the original INVITE if the UAC indicated support for this mechanism. The UAC may then choose the time and conditions under which it will contact the URI specified in the Early-Media header to establish a separate (but related) dialog to handle the "early" media. This separate dialog is called an early-media dialog. The original INVITE dialog is called the final-media dialog.

By doing so, several benefits accrue to the UAC and the proxy/UAS:

- o The UAC has total control over when/if it will initiate the early media dialog.
- o Because the SDP negotiation is separate from final media the early media may have different directionality from the final-media. For instance, the final-media may be recvonly, but the early media may be sendrecv.
- o The establishment of a separate early-media dialog reuses the same mechanism on the UAC that generates any other INVITE transaction. No new methods or complex negotiation mechanisms are required.

- o Forking is no longer an issue for early media. The UAC controls the timing and ordering of any early media dialogs explicitly. It can also know which endpoint is generating what media through subtle manipulation of its SDP parameters.
- o No negotiation is required in the media path. Everything is handled in the open on the signaling path where proxies may enforce network policies regarding early media.
- o The mechanism is entirely compatible with existing mechanisms. If no Early-Media header is received by the UAC, it follows [[RFC3960](#)] rules for early media rendering.
- o SRTP negotiation is simplified. The final-media offer may include SRTP whereas early-media dialogs may contain an offer that uses regular RTP to conserve resources.

[6.](#) Early Media Indirection Operation

[6.1.](#) The Early-Media Header

In order to allow proxies and UASs to specify a location that the UAC should contact for early-media, a new header is introduced. This is done instead of adding SDP attributes to ease the ability of a proxy to manage these contacts. For instance, if an Early-Media header is seen coming from an untrusted endpoint, the proxy may strip the header from the message. Likewise, if multiple headers are received on various forked call legs, it is possible that the proxy may store and forward these header values at a later time to facilitate ordering of early media streams to the calling party.

The format of the Early-Media header is given as the following (building off of ABNF definitions given in [section 25.1 of \[RFC3261\]](#)):

```
Early-Media = "Early-Media" HCOLON ( contact-param * (COMMA
contact-param))
```

The value of the early media header is one (or more) URIs for the UAC to contact for early-media. In the case that a single URI is specified and the UAC fails to establish a dialog, that early media stream will fail to be rendered. If multiple URIs are specified, the UAC may contact the next URI in the list if it fails to establish an early media stream using a prior entry in the list. Once an early media dialog is established, the remainder of the list (if any) is discarded.

[6.2.](#) User Agent Client Procedures

The following sections describe the procedures that the UAC uses in order to support early media indirection.

[6.2.1.](#) Indicating support for Early Media Indirection (EMIND)

Upon generation of an initial INVITE request, the UAC MUST include

the 'emind' option tag in a Supported header as part of the INVITE request being sent. This lets the UAS know that the UAC supports early media indirection and will process the Early-Media header information sent back in any 18x responses. Failure to include this value in the supported header may cause endpoints wishing to generate early media to not use the procedures defined in this document (i.e. revert to baseline behavior). If the initial INVITE request does not include an SDP offer, then the next SIP request message that contains an SDP offer prior to the 200 OK to the INVITE being received MUST include the 'emind' option tag in a Supported header.

6.2.2. Accepting indirect early-media streams

The following section specifies the mechanism that a UAC uses to accept an indirect (EMIND negotiated) early-media stream. If the UAC is currently rendering any non-EMIND negotiated early-media it MUST stop rendering that early-media. If the UAC is currently rendering any EMIND negotiated early-media it MUST stop rendering that early-media and complete the procedures given in section [Section 6.2.3](#).

In order to accept indirect early-media, the UAC creates a separate SIP dialog to handle the negotiation of this early media. This dialog, known as an early-media dialog, allows the UAC to receive early media from endpoints other than the called party's device(s) such as a network media server, does not suffer from the problems of forking (since each indirect early-media session creates a separate early media dialog), allows for the UAC to specify a different RTP payload type (to facilitate discrimination of early-media versus final-media), and gives both the UAC and UAS the ability to explicitly start and stop the early media stream without complicating the SDP negotiations for the final media.

1. Create a new "early-media" INVITE request copying information from the original INVITE request.
2. Copy the URI information from the 18x Early-Media header into the request URI of the new "early-media" INVITE.
3. Create a new SDP offer (may be a copy or subset of the offer in the original INVITE). For each codec offered, specify a new dynamic RTP payload type specified for the offered codecs. The payload number MUST be unique such that it does not duplicate a

- dynamic RTP payload type is called the early-media payload type.
4. Send the new "early-media" INVITE to the same first-hop as the original INVITE was sent to.
 5. Render any media received with the matching the early-media payload type.

[6.2.3.](#) Closing indirect early-media streams

The following section specifies the mechanism that a UAC uses to close an indirect (EMIND negotiated) early-media stream. These procedures may be executed in parallel to accepting a new indirect early-media stream or any SIP signaling in relation to the original INVITE request sent by the UAC to initiate the dialog.

1. Stop rendering any media matching the early-media payload type used for this indirect early-media stream.
2. Send the UAS generating the early media a BYE request to close the dialog for this early-media stream.

[6.2.4.](#) Switching from early-media to final-media

The procedure for knowing when early-media packets should be replaced with final-media packets is straight-forward due to the utilization of unique dynamic RTP payload types for the early-media dialog(s). When the UAC detects media being received containing valid RTP payload types that do not match those of a dynamic RTP payload type specified in an early-media dialog, it simply stops rendering the early-media stream(s) and begins rendering the final-media for the original INVITE dialog. This eliminates problems coordinating signaling between early-media generators and a called-party device answering the original INVITE.

Upon receiving a final-media RTP payload type, the UAC SHOULD execute the procedures in section [Section 6.2.3](#) in a timely fashion for any early-media dialog(s) it may have created.

[6.3.](#) Proxy/User Agent Server Procedures

If a proxy or UAS receives an INVITE, and wishes to generate end-to-middle early media towards the UAC, it should first look to see if the UAC has indicated support for early media indirection by the presence of the 'emind' option tag in the Supported header of the INVITE. If it has, then it may use the procedures specified in this section.

In order to offer an indirect early media stream, the proxy/UAS creates a secondary dialog. This dialog is called the "early-media

dialog". By creating a separate dialog, the negotiation of early-media is separated from the negotiation of final media.

Because the early-media is then handled on a separate dialog, there is no reason for it to be "early" anymore. The early-media dialog that is created can be answered or rejected immediately. When either end wishes to discontinue playback of the "early" media stream they simply end the session by sending a BYE request for the early-media dialog.

It should be noted that there is no restriction that says that the UAS handling the original INVITE request must also handle the early-media dialog. This allows UASs and proxies to specify media servers which may be available in the network as sources for early-media content. As a result, restrictions placed upon client devices that prevent streaming of media prior to answering the call (i.e. gating of their media) do not interfere with the ability to provide early media services such as colorful ringback.

[6.3.1.](#) Offering indirect early media streams

If a proxy/UAS wishes to offer an early-media stream, they may do so by including the following in a 18x response to the original INVITE request:

1. A Requires header that includes the tag value 'emind'.
2. Include an Early-Media header that specifies the URI that the originator should contact for early media. This URI may be a SIP URI (possibly a GRUU) that points to the resource that is controlling the early media to be generated. This may be the proxy/UAS itself, or another network element such as a media server. The URI may contain parameters that provide contextual information about what early media is to be played.

[6.3.2.](#) Closing indirect early media streams

If the proxy/UAS wishes to close an early media stream, they may do so by sending a BYE request on the early-media dialog. Closing early-media streams by a proxy/UAS that are being served by third-party network elements is outside the scope of this specification as it is assumed that there is some form of coordination (such as an H.248 interface to the media server from the proxy server) present that makes this possible.

Any early-media dialogs that are controlled by the proxy/UAS MUST be closed after sending a 3xx or higher response code to the original INVITE request. If any early-media dialogs were created that are

handled by third-party network elements, it is assumed that the UAC will close these dialogs upon receipt of the 3xx or higher response

code from the proxy/UAS.

[6.3.3.](#) Ordering early media streams

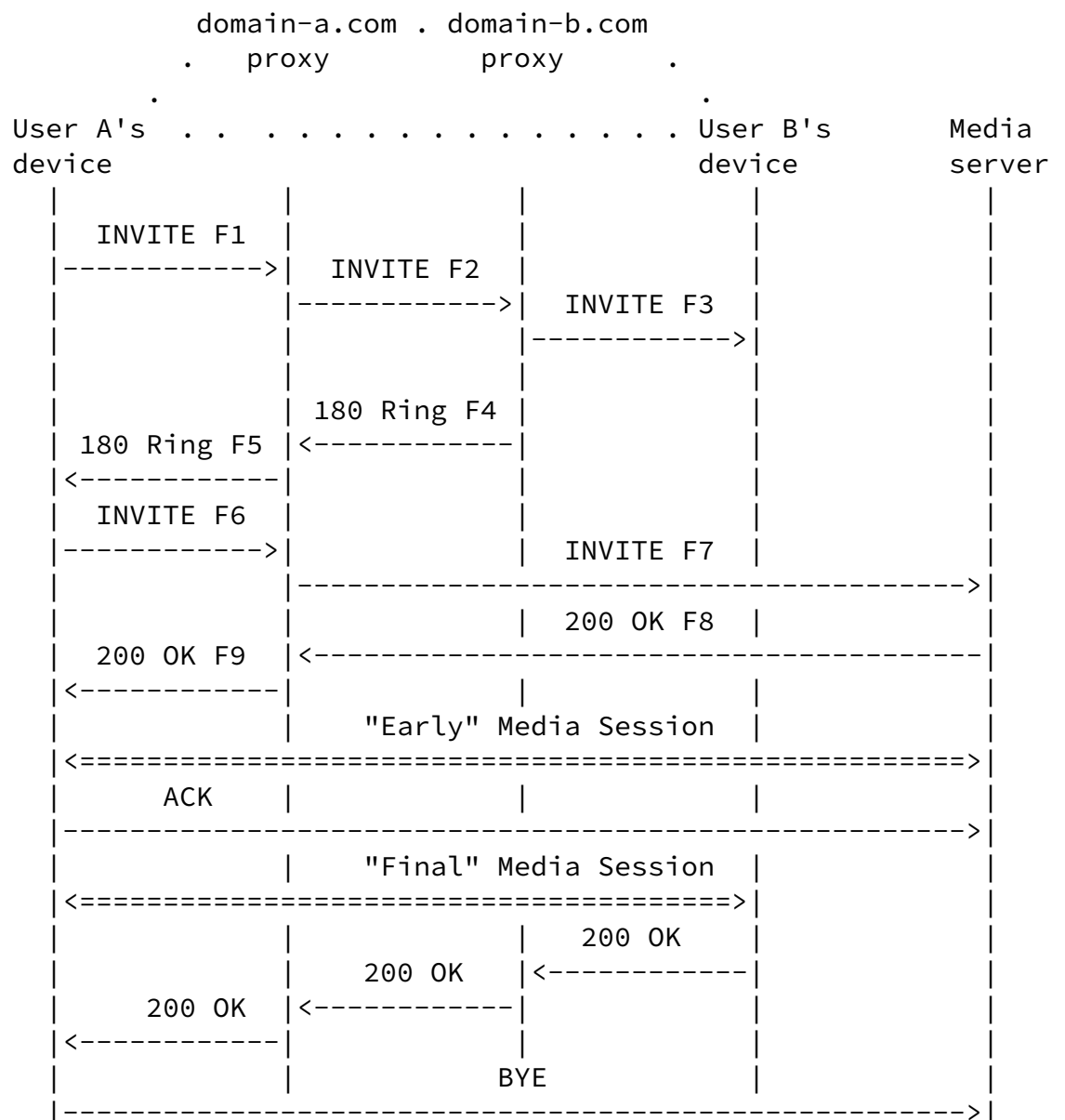
With the current specifications for early media, issues exist when multiple early media streams are sent to the calling party simultaneously. Because the media path may be decoupled from the signaling path, proxies may have little or no control over the ordering of early-media being sent to the calling party. Because devices generating early-media may not be aware of forking of the SIP signaling, they may not know that a conflict exists at the calling party that prevents proper rendering of early media being generated.

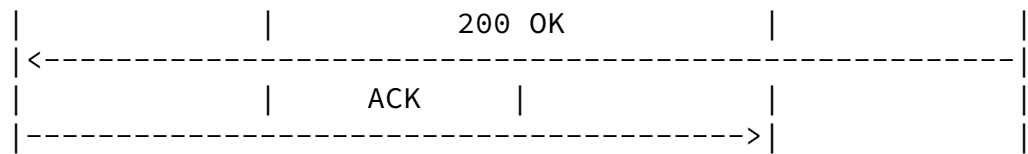
Correct ordering of early-media is facilitated by early-media indirection due to the requirement that the calling party explicitly create a separate SIP dialog to handle negotiations. Further, because the passing of indirection information is handled in a SIP header instead of in a message body, it is possible for proxies to inspect requests for early media and facilitate proper ordering.

[7.](#) Example Call Flow

The following call flow is given to show how the early-media indirection procedures operate. In this example User A calls User B. User B has previously instructed their proxy that they wish to have a custom ringtone streamed to the originator. Rather than trying to manage this as an inline operation, the proxy uses early media indirection to handle the media interactions.

The call flow for this scenario would be as follows:





Call flow diagram

The detailed call flow is as follows:

F1: INVITE (User A to Proxy A):

```

INVITE sip:userb@domain-b.com SIP/2.0
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: UserB <userb@domain-b.com>
From: UserA <usera@domain-a.com>;tag=d7t6d7ftsdf7t6
Call-ID: 843817637684230@998sdasdh09
CSeq: 1 INVITE
Supported: emind
Contact: <sip:usera@devicea.domain-a.com:5060>
Content-Length: ...
Content-Type: application/sdp
  
```

```

v=0
o=usera 53655765 2353687637 IN IP4 devicea.domain-a.com
s=-
t=0 0
c=IN IP4 devicea.domain-a.com
m=audio 3456 RTP/AVP 0 1 3 99
a=rtpmap:0 PCMU/8000
a=sendrecv
  
```

User A's proxy forwards the request to User B's proxy.

F2: INVITE (Proxy A to Proxy B):

INVITE sip:userb@domain-b.com SIP/2.0
Via: SIP/2.0/UDP proxy.domain-a.com:5060;branch=z9hG4bKaf88df7
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: UserB <userb@domain-b.com>
From: UserA <usera@domain-a.com>;tag=d7t6d7ftsdf7t6
Call-ID: 843817637684230@998sdasdh09
CSeq: 1 INVITE
Supported: emind
Contact: <sip:usera@devicea.domain-a.com:5060>
Content-Length: ...
Content-Type: application/sdp

v=0
o=usera 53655765 2353687637 IN IP4 devicea.domain-a.com
s=-
t=0 0
c=IN IP4 devicea.domain-a.com
m=audio 3456 RTP/AVP 0 1 3 99
a=rtpmap:0 PCMU/8000
a=sendrecv

User B's proxy forwards the request from User A to User B's device.

F3: INVITE Proxy B to User B):

INVITE sip:userb@deviceb.domain-b.com SIP/2.0
Via: SIP/2.0/UDP proxy.domain-b.com:5060;branch=z9hG4bK87yfgd8g
Via: SIP/2.0/UDP proxy.domain-a.com:5060;branch=z9hG4bKaf88df7
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: UserB <userb@domain-b.com>
From: UserA <usera@domain-a.com>;tag=d7t6d7ftsdf7t6
Call-ID: 843817637684230@998sdasdh09
CSeq: 1 INVITE
Supported: emind
Contact: <sip:usera@devicea.domain-a.com:5060>
Content-Length: ...
Content-Type: application/sdp

v=0
o=usera 53655765 2353687637 IN IP4 devicea.domain-a.com
s=-

```
t=0 0
c=IN IP4 devicea.domain-a.com
m=audio 3456 RTP/AVP 0 1 3 99
a=rtpmap:0 PCMU/8000
a=sendrecv
```

Meanwhile, user B's proxy wishes to invoke early media from a media server in domain-b.com to User A's device to provide custom ringback:

F4: 180 Ringing (Proxy B to Proxy A):

```
SIP 2.0 180 Ringing
Via: SIP/2.0/UDP proxy.domain-a.com:5060;branch=z9hG4bKaf88df7
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: UserB <userb@domain-b.com>;tag=s12td12tr3t2
From: UserA <usera@domain-a.com>;tag=d7t6d7ftsdf7t6
Call-ID: 843817637684230@998sdasdh09
CSeq: 1 INVITE
Required: emind
Early-Media: <sip:service@media-server.domain-b.com:5060
;service=ringback;user=userb>
Content-Length: 0
```

User A's proxy validates that User B's proxy is trusted for early media and leaves the Early-Media header alone. The 180 is forwarded to User A's device:

F5: 180 Ringing (Proxy A to User A):

```
SIP 2.0 180 Ringing
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: UserB <userb@domain-b.com>;tag=s12td12tr3t2
From: UserA <usera@domain-a.com>;tag=d7t6d7ftsdf7t6
Call-ID: 843817637684230@998sdasdh09
CSeq: 1 INVITE
Required: emind
Early-Media: <sip:service@media-server.domain-b.com:5060
;service=ringback;user=userb>
```

Content-Length: 0

User A's device sees that an Early-Media header is present in the 180 response and executes the procedures in section [Section 6.2.2](#) to initiate the requested early-media dialog. It selects the dynamic RTP payload type of 96 for the early-media payload type:

F6: INVITE (User A to Proxy A):

```
INVITE sip:service@media-server.domain-b.com:5060
      ;service=ringback;user=userb SIP/2.0
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345aaa
To: Service <service@media-server.domain-b.com>
From: UserA <usera@domain-a.com>
Call-ID: 843230@998sdasdh09
CSeq: 1 INVITE
Contact: <sip:usera@devicea.domain-a.com:5060>
Content-Length: ...
Content-Type: application/sdp
```

```
v=0
o=usera 53655765 2353687637 IN IP4 devicea.domain-a.com
s=-
t=0 0
c=IN IP4 devicea.domain-a.com
m=audio 3456 RTP/AVP 96
a=rtpmap:96 PCMU/8000
a=recvonly
```

User A's proxy forwards the request to the media server in domain-b.com:

F7: INVITE (Proxy A to Media Server):

```
INVITE sip:service@media-server.domain-b.com:5060
      ;service=ringback;user=userb SIP/2.0
```

Via: SIP/2.0/UDP proxy.domain-a.com:5060;branch=z9hG4bkdfsd8fysdf
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345aaa
To: Service <service@media-server.domain-b.com>
From: UserA <usera@domain-a.com>;tag=d87fysdf8sdfy
Call-ID: 843230@998sdasdh09
CSeq: 1 INVITE
Contact: <sip:usera@devicea.domain-a.com:5060>
Content-Length: ...
Content-Type: application/sdp

v=0
o=usera 53655765 2353687637 IN IP4 devicea.domain-a.com
s=-
t=0 0
c=IN IP4 devicea.domain-a.com
m=audio 3456 RTP/AVP 96
a=rtpmap:96 PCMU/8000
a=recvonly

The media server in domain-b accepts the incoming request for media and accepts the offer from User A's device.

F8: 200 OK (Media Server to Proxy A):

SIP 2.0 200 OK
Via: SIP/2.0/UDP proxy.domain-a.com:5060;branch=z9hG4bkdfsd8fysdf
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: Service <service@media-server.domain-b.com>
From: UserA <usera@domain-a.com>;tag=d87fysdf8sdfy
Call-ID: 843230@998sdasdh09
CSeq: 1 INVITE
Content-Length: ...
Content-Type: application/sdp

v=0
o=media 53655765 2353687637 IN IP4 media-server.domain-b.com
s=-
t=0 0
c=IN IP4 media-server.domain-b.com
m=audio 7890 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly

User A's proxy forwards the response from User B's media server to

User A's device.

F9: 200 OK (Proxy A to User A):

```
SIP 2.0 200 OK
Via: SIP/2.0/UDP usera.domain-a.com:5060;branch=z9hG4bK12345
To: Service <service@media-server.domain-b.com>
From: UserA <usera@domain-a.com>;tag=d87fysdf8sdfy
Call-ID: 843230@998sdasdh09
CSeq: 1 INVITE
Content-Length: ...
Content-Type: application/sdp
```

```
v=0
o=media 53655765 2353687637 IN IP4 media-server.domain-b.com
s=-
t=0 0
c=IN IP4 media-server.domain-b.com
m=audio 7890 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly
```

Early media is now flowing from User B's media server to User A's device. Some time later, User B answers the phone and begins sending RTP packets to User A's device. These packets arrive ahead of the 200 OK to the INVITE sent to User B's device. User A's device detects packets with an RTP payload type of other than 96 (the selected early media payload type). Accordingly, it sends a BYE to the early-media dialog (not shown). The remaining messaging on the original INVITE dialog follows the normal SIP call setup procedures (not shown).

[8.](#) Security Considerations

This document is a work in progress. Security considerations will be added as various recommendations become more concrete.

[9.](#) IANA Considerations

This document defines the option-tag of "emind" which will require IANA registration.

[10.](#) References

Internet-Draft

Early Media Indirection in SIP

February 2007

[10.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

[10.2.](#) Informational References

- [RFC2327] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", [RFC 2327](#), April 1998.
- [RFC3261] 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.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [RFC3960] Camarillo, G. and H. Schulzrinne, "Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)", [RFC 3960](#), December 2004.
- [I-D.stucker-sipping-early-media-coping]
Stucker, B., "Coping with Early Media in the Session Initiation Protocol (SIP)",
[draft-stucker-sipping-early-media-coping-03](#) (work in progress), October 2006.

Author's Address

Brian Stucker
Nortel
2201 Lakeside
Richardson, TX 75082
US

Phone: +1 972 685 7724
Email: bstucker@nortel.com
URI: <http://www.nortel.com/>

Stucker

Expires August 16, 2007

[Page 17]

Internet-Draft

Early Media Indirection in SIP

February 2007

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

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.

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, THE IETF TRUST 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.

Intellectual Property

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.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).