SIP Early Media

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

Early Media is the ability of two SIP user agents to communicate before a SIP call is actually established. Support for early media is important largely for interoperability with the PSTN. Unfortunately, many SIP devices are providing this capability today without a well-documented, consistent, and complete solution. We define the problem of early media, document and describe the difficulties with the current approach used in SIP UAs, and present a more formal protocol mechanism that resolves these difficulties.

1 Introduction

Early media is the ability of two SIP user agents to communicate before a SIP call is actually established. Typically, this occurs when the called party is a PSTN gateway of some sorts. The gateway
might provide inband tones or announcements before the call is set up, in order to inform the caller of call progress. Early media might even involve the transfer of media from caller to callee. Within the PSTN, forward channels can be established for the purpose of conveying DTMF in order to select a final destination to call. This feature is frequently used for access to IVR systems behind 800 numbers.

Despite the fact that a bidirectional media stream needs to be established, early media cannot be supported by simply answering the call at the called party. The reason is that early media may not be followed by a call acceptance. A 2xx class response to an INVITE both establishes a media session, and indicates acceptance of the call. Early media establishes a media session, but does not answer the call. Since features and applications are driven off of answering or not answering of calls, a 2xx response cannot be used for early media.

In this draft, we discuss the problems in providing early media within SIP, and propose a reasonable solution.

2 Existing Approach

Current implementations support early media through the 183 response code, which was first described in a now-expired Internet Draft. When the called party wishes to send early media to the caller, the called party sends a 183 response to the caller. That response contains SDP. When the caller receives the 183, it suppresses any local alerting of the user (for example, audible ringtones or a pop-up window), and begins playing out media that it receives. The SDP in the 183 provides an address to which RTCP packets can be sent. Some implementations take media from the caller, and send that to the callee as well.

If the call is ultimately rejected, the called party generates a non-2xx final response. When this is received at the caller, it ceases playing out, or sending of media. However, if the call is accepted, the called party generates a 2xx (generally, with the same SDP as was present in the 183), and sends that to the caller. Media transmission continues as before.

This simple approach for early media suffers several serious problems, many of which are related to forking. The known problems with this approach are:

- Early media can’t be declined. If the caller does not wish to receive an early media stream, there is no way to stop one from being sent to it. This is particularly problematic when
the INVITE forks, and reaches multiple UASes, each of which generates early media. If the caller is connected via a low-speed link, the aggregate rate of the resulting media streams may exceed the capacity of the link. This causes excessive packet loss.

- Early media can’t be modified. If the caller wishes to modify some aspect of the early media stream, it cannot. For example, an early media stream might be put on hold. However, since a UAC is not allowed to send an INVITE when an INVITE is in progress, there is no capability for modifying any aspect of the media stream.

- Early media can’t be identified. Early media from caller to callee is sent to the IP address and ports specified in the INVITE. The INVITE might fork and hit multiple UASes, each of which generates early media. Each of those early media streams are sent to the same IP address/port at the caller. The caller can usually separate them based on SSRC. However, it is possible (although unlikely) that both UAS select the same SSRC. Since the two UASes are not in an RTP session together, this collision will not be detected. Even if the SSRC are chosen differently, there is no way to associate a media stream with a 183. That is, if a UAC begins receiving two early media streams as a result of two 183s, it can’t tell which media stream was created from which 183. This will cause problems for user interfaces, and for invocation of features.

- Since the 183 must be received, early media requires the caller to support the reliability of provisional responses extension [1].

- Media and code may not match. Early media can be delivered using any provisional response code, not just 183. Some implementations place SDP in a 180 response code. Unfortunately, the called gateway frequently doesn’t know what the content of the media stream will be. Therefore, some gateways always generate the same response code, 180 for example, when they determine that early media is needed in the reverse direction. The content of the media stream may not agree with the meaning of the provisional response code. For example, the media stream may say something like "The called party is busy", which a 180 response is used, which means "Ringing". This can be confusing for users.

- Sequential searches may not work. Sometimes, the called gateway may not be able to determine that a call failed. This is because the failure indication may come as media inband,
causing the gateway to use early media. The early media might, for example, be a repeating message like "Phone out of service". The gateway may never receive a PSTN signaling message telling it that the call has actually failed. As a result of this, applications in the SIP network will not be able to know that the call failed, in order to try an alternate destination.

Some of these problems are due to limitations of SIP, and some are fundamental problems that arise due to PSTN interworking (the 5th and 6th items above). Our aim here is to solve only those problems which a protocol extension can fix.

Note that early media is not the same as media that arrives before the 2xx because of near-simultaneous transmission of the 2xx and media at the UAS. This media is sent after the call is accepted. None of the problems described above exist in this case. These media streams can be disconnected through BYE, and modified through re-INVITE.

There still may be a transient situation where media is lost, though. A sends an INVITE that is received at B and C. Both pickup at the same time, generating a 2xx, and both speak a few milliseconds after the 2xx is sent. The UAC will get two media streams, which may result in packet loss. These media streams cannot be stopped until the 2xx’s arrive, and a BYE is sent and received. This will require 1 signaling RTT, which can be a few hundred milliseconds. There is no way to solve this problem without introducing media clipping for an equivalent period of time.

3 Solution Space

To fix these problems, we make the basic observation that these problems don’t exist for regular media established through an INVITE/2xx. The reason is that this media establishment follows a two-pass offer/answer model. One side offers a stream, and the other can accept it, providing their session description, or reject it. At any time, either party can send a new offer to modify characteristics of the session.

Effectively, when the called party decides to send early media, it is choosing to offer an early media stream. For this early media stream to work properly, the caller has to be able to accept or reject, and generate an answer. We need to allow the caller and callee to modify the early media stream at any time through a new offer. Effectively, the same two-pass offer/answer model needs to be used for early media
as well. Specifically:

- If the called party wishes to send early media, it offers an early media stream to the caller.
- The caller wishes to accept the early media stream, it generates an answer.
- At any time, the caller can generate a new offer, which is answered by the called party.
- At any time, the called party can generate a new offer, which is answered by the caller.

The SIP bis-03 specification defines a generalized offer/answer model which operates independently of the pair of SIP messages that deliver the offer and answer. SIP bis-03 allows the offer/answer pair to happen in an INVITE/200 or 200/ACK. To enable early media, we must allow for offers and answers to occur in other messages. Specifically, they must appear in messages from caller to callee, and callee to caller, before the 2xx to the initial INVITE is generated. The entire problem, therefore, is selection of the messages which contain these two SDPs.

We see five reasonable possibilities of mapping offer/answers onto SIP messages:

Solution 1: From called party to caller, the offer is sent in a 1xx provisional response, and the answer is sent in the PRACK. From caller to called party, the offer is sent in a re-INVITE, and the answer in a 1xx.

Solution 2: From called party to caller, the offer is sent in a new request, OFFER_EARLY_MEDIA, and the answer in a 2xx to that request. Similarly, from caller to called party, the offer is sent in an OFFER_EARLY_MEDIA, and the answer in a 2xx.

Solution 3: From called party to caller, the offer for the early media stream is sent in a new INVITE for a new call-leg. This new call leg is associated with the initial call leg through some new header. The answer is sent in a 2xx. Similarly, offers for modification of the early media session are sent in a re-INVITE within this second call leg, and the answer, within a 2xx. The initial INVITE in this new call leg would need to have preloaded routes in order to help assure that it gets to the right party.
Solution 4: A combination of Solution 1 and 2. From the called party to caller, the offer is sent in a 1xx, and the answer in a PRACK. From the caller to called party, the offer is sent in an OFFER_EARLY_MEDIA message, and the answer in a 2xx to that request.

Solution 1 maps naturally to the existing solutions. However, it relies on a re-INVITE from caller to callee. SIP currently forbids sending of a re-INVITE before an initial INVITE completes. This solution would require that restriction to be relaxed. This should not be a major problem. Even though they overlap, each re-INVITE effectively "completes" through the receipt of a 18x followed by PRACK (as opposed to a 2xx followed by ACK). Indeed, the restriction would be imposed that a re-INVITE to modify early media could not take place until a reliable 1xx had been received for that re-INVITE. Solution 1 is elegant since it is almost entirely specified by declaring that 1xx is treated as if it were 2xx, and PRACK treated as if it were ACK. There are some exceptions, of course (the PRACK is not retransmitted on receipt of a duplicate 1xx, and the PRACK will generally carry SDP even when the INVITE and 1xx did. If the INVITE and a 2xx carry SDP, the ACK message does not).

Solution 2 eliminates the problem with overlapping INVITE transactions. However, it introduces a new way to establish sessions, which is very undesirable. It will require the re-specification of many of the procedures already specified for INVITE. It will also not interoperate with the way existing devices are providing early media.

Solution 3 also eliminates the problem with overlapping INVITE transactions. It also uses the existing mechanisms for session establishment, rather than defining additional methods. However, it is likely to confuse many existing systems. That's because the INVITE to establish an early media session will be viewed as a new call attempt, and existing applications within network elements are likely to treat it as such. For example, a call screening app might be invoked which prevents the request from passing to the calling party. In theory, this should be prevented by the pre-loaded routes, but in practice, servers will still apply new call processing to requests with pre-loaded routes. It is also not compatible with the way existing devices are providing early media.

Solution 4 overcomes the problems with overlapping transactions from solution 1, and overcomes the interop problems of solutions 2 and 3. However, it does so at the expense of defining a new mechanism for establishing of sessions, which is the same problem as solution 2.

4 Recommended Solution: Solution 1
Our recommendation is solution 1. The issues with overlapping transactions and extra messaging are minor, in our view. The solution maintains backwards compatibility, which is important. It avoids specification of a parallel call establishment mechanism, which is also important. We believe it is most consistent with existing SIP operation, which uses a re-INVITE to update the existing session parameters.

The following subsections specify the mechanism in more detail.

### 4.1 UAC Behavior

A UAC supporting early media as defined in this specification MUST support the server features extension [2] and MUST include a Supported header in an INVITE request, containing the token "em". A UAC supporting early media MUST also support reliability of provisional responses [1]. It MUST, however, include the "100rel" token in the INVITE request, even though support for em implies support for 100rel.

As specified in bis, once a UAC sends an INVITE with SDP, it MUST be prepared to receive media on the IP addresses and ports placed into the SDP. The use of early media does not change this.

If the UAC receives a provisional response with a Require header containing the tokens "em" and "100rel", the UAS is requesting early media for the call. The UAC MUST generate a PRACK for this provisional response as specified in [1]. This PRACK MUST contain a Require header with the token "em". This PRACK MUST contain an SDP. That SDP MUST be formulated as a valid answer to the offer in the provisional response.

If the UAS wishes to refuse the early media stream (in other words, it doesn’t want any early media), it SHOULD set the port number for each media stream in the answer to zero. This tells the UAS that the UAC does not wish to receive any media before the call is answered.

If the UAS wishes to accept the early media stream, it generates a valid SDP which it can use to receive the offered media streams.

Once the UAC sends the PRACK, it MUST be prepared to receive media according to the information in the the SDP in the PRACK (assuming it didn’t reject the early media streams). It is RECOMMENDED that the SDP in the PRACK to the first 1xx with early media use the same IP addresses and ports as the offered media streams in the INVITE (if there was one). This provides a smooth transition from early media to the final media, in addition to backwards compatibility with older UAS that send early media to the IP addresses and ports in the SDP in the
INVITE. However, the UAC MAY change these addresses in the SDP in the PRACK. In that case, it MUST continue to be prepared to receive media according to the information in the SDP in the original INVITE. Furthermore, a UAC MUST be prepared to receive a 2xx for the call on a new call leg that didn’t use early media. The SDP in that 2xx will always be a response to the SDP offered in the initial INVITE.

The UAC may receive additional provisional responses from the UAS containing SDP. Each of these is treated as an update of the previous SDP on that call leg, and an answer MUST be generated as per [3] which is carried in the PRACK.

The UAC may receive provisional responses from different UASes (known by different call leg identifiers), each of which offers early media. The UAC MAY accept or reject each as it pleases, following the rules here. It is RECOMMENDED that for call legs after the first, the UAC include SDP in the PRACK which contains different IP addresses and ports than those from the INVITE. This allows the UAC to disambiguate the various media streams by IP address/port, and to correlate media streams with call legs. However, in all cases, the UAC MUST continue to listen for media on the IP addresses and ports provided in the INVITE.

The UAC MAY decide to update its SDP for a given call leg that is using early media. To do this, it MUST generate a re-INVITE for that call leg, using the procedures specified in [3]. Note, however, that while baseline SIP prohibits the use of multiple outstanding INVITE transactions, this extension softens that restriction (the overlapping is not a problem here, since the 1xx effectively plays the role of a 2xx in completing the transaction, as far as SDP exchanges are concerned). The re-INVITE MUST contain SDP, and that SDP MUST be formulated as a valid offer updating the previous SDP provided by the UAC. The re-INVITE MUST contain a Require header containing the token "em".

Open Issue: should we allow the re-INVITE to not contain SDP? In this case, the UAC is requesting that the UAS generate an offer to update the SDP, and that offer will come in the 1xx. Are there any 3pcc cases where this would be useful?

That re-INVITE may generate a reliable 1xx from the UAS, containing an answer to the offered SDP. This SDP is also an offer for early media, and MUST be answered with SDP in a PRACK. This is neccessary for idempotency; since the initial INVITE transaction operates this way (with the 1xx being an offer, and the PRACK being the answer), so too must the re-INVITE. It is possible that the SDP in the PRACK can
be the same as the SDP in the INVITE, but there are cases where it's not possible. For example, if an offer from the UAC contained a stream as sendrecv, and the answer/offer in the 1xx from the UAS indicated that the stream was sendonly, the answer in the PRACK MUST be marked as recvonly as per bis [3].

Open Issue: Effectively, early media re-INVITE formally introduces a three-way handshake into SIP; the offer comes in an INVITE, the 1xx contains an answer (which is also an offer), and the PRACK contains an answer to the SDP in the 1xx. In principle, we could allow this for baseline SIP by doing the same thing in INVITE/2xx/ACK. However, this would introduce media clipping, which is less acceptable for regular media as it is for early media. That's because early media is always generated by an automata, which can be programmed to wait for a PRACK before sending media. However, you cannot "program" a person to wait for the ACK before talking into the receiver after they answer. Perhaps a better approach is to allow the PRACK to not contain SDP in the re-INVITE case; this way, everything is a two-way handshake, but in the initial INVITE, the SDP is basically "ignored". We would then need a token in the 1xx somewhere which says "I ignored your offer of SDP in the INVITE, this 2xx is an offer, not an answer". This token, perhaps a require token, would be present in the original 1xx but not 1xx in response to re-INVITEs. We could apply this to INVITE/2xx/ACK as well.

The re-INVITE may trigger a response to the previous INVITE. This response may be a 489, which indicates that the UAS has received the updated re-INVITE. Any modifications to the early media will come as reliable provisional responses to the new re-INVITE transaction. Furthermore, it is possible that the previous INVITE is answered with a final response, and the UAC MUST be prepared to handle that. In that case, the INVITE which was just sent will be treated as a re-INVITE for an active call leg, and will be rejected by the UAS, in fact, because it is identified as a "glare" situation, akin to both caller and callee sending INVITE at the same time. Alternatively, the re-INVITE may be rejected, in which case the UAC continues to use the previous SDP in had been using before sending the re-INVITE.

To handle backwards compatibility, a UAC that receives a provisional response with SDP, but which does not contain the token "em" in a Require header, MUST be prepared to handle it. Specifically, the UAC SHOULD listen for media on the ports advertised in the INVITE, and SHOULD generate RTCP towards the IP addresses and ports provided in the 183.
4.2 UAS Behavior

When a UAS receives an INVITE for a new call, it MAY elect to request early media for that call if the INVITE has a Supported header containing the tokens "em" and "100rel". To request early media, it constructs a provisional response, using any response code excepting 100. That provisional response MUST contain a Require header with the tokens "em" and "100rel". The UAS MAY send other reliable provisional responses that have nothing to do, and no impact on, early media. These provisional responses MUST NOT contain the token "em" in a Require header.

Whether or not early media is used, the status code of the provisional response SHOULD reflect known status at the UAS. For example, if early media is used to indicate that the user is queued in a call holding system, the 182 containing SDP should be sent, not a 183. This is so the displays on graphically-enabled clients (which frequently show the response code) are consistent with the media being played to the caller.

A reliable provisional response to established early media MUST contain SDP. This SDP MUST be a valid offer as specified in [3], as it constitutes an offer for early media, that will be answered by the UAC. This SDP has no relationship to the SDP in the INVITE (i.e., it is not necessarily a valid answer to the SDP in the INVITE). The provisional response MUST contain a tag in the To field. This capability is optional in SIP but the strength is increased for this extension. The provisional response MUST mirror any Record-Route headers present in the request. This capability is optional in SIP but the strength is increased for this extension. The provisional response MUST contain a Contact header. This capability is optional in SIP but the strength is increased for this extension.

Open Issue: We are effectively ignoring the SDP in the INVITE. Is that OK?

If the UAS is only interested in one-way media from the caller (UAC) to it, the media streams in the SDP it generates MUST be marked as sendonly, if possible. This may not be possible if the UAC offered a stream as sendonly. In that case, the answer from the UAS MUST be placed on hold.

If the UAS is interested in bidirectional early media, the media streams in the SDP it generates MUST be send-receive, if possible (since this is the default, the a=sendrecv attribute MAY be omitted).

Once the UAS sends the provisional response, it MUST be prepared to
receive media on any streams that were sendonly or sendrecv in the SDP in its offer. The UAS MUST NOT send any media until it receives an answer from the UAC, which will arrive in the PRACK for that provisional response.

This restriction may result in clipping of early media. That is unavoidable if we wish to allow the UAC to generate an answer to the offered early media. Doing so is needed to allow early media to be refused or directed to a different port.

The UAS MAY update its media information by sending an additional reliable provisional response. However, it MUST NOT do so until it has received a PRACK for any previous reliable provisional response that contained SDP. The SDP in subsequent PRACKs MUST be constructed as if the 1xx were a re-INVITE, and the SDP in the previous 1xx was a 2xx response to the INVITE, following the rules specified for this case in [3]. In other words, this SDP is another offer that updates the last SDP provided by the UA.

The UAS MAY generate other reliable (and unreliable) provisional responses that have nothing to do with early media. These provisional responses MUST NOT contain the token "em" in a Require header. These responses have no impact on early media, and are ignored for purposes of this specification.

Open Issue: Do we want to force all reliable provisional responses afterwards to contain SDP, and use em, if for no other purpose than to "refresh" the current SDP? This is consistent with INVITE behavior.

The UAS may receive a re-INVITE on the call leg, even though the UAS has not generated a final response for that call leg. In this case, the UAC is requesting an update to the early media streams. If the update is not acceptable, the UAS MUST respond to the re-INVITE with a 488 response. As with normal re-INVITEs, this does not terminate the call, but simply means that both users continue to use the previous SDP. Any subsequent reliable provisional responses, used to update the media session, are sent on the previous INVITE transaction, which is still in progress.

If the update is acceptable, the UAS MUST generate a final response to the previous INVITE transaction. This response MUST use a 489 response code, which has the default reason phrase "Request Updated". This response tells that UAC that the request was rejected because it has a new request on which the actual final response will
be sent. The UAS MAY generate a final response to this updated INVITE request, or, if it wishes to continue with early media on the call leg, it MUST generate a reliable provisional response containing an answer to the SDP in the re-INVITE. This SDP will also constitute an offer, and will be answered by an SDP in the PRACK, in the same fashion as is done for the initial INVITE. The rules for handling this are identical to those for the initial INVITE.

The result is that there is always one in-progress INVITE transaction, which will be used to send further provisional responses, or to generate the final response.

At any time, the UAS may elect to generate a final response to the in-progress INVITE transaction. If the final response is a 2xx, it MUST contain SDP. If the original INVITE contained SDP, the SDP in the 2xx MUST contain a valid answer to the SDP in the most recent PRACK. If the INVITE didn’t contain SDP, the SDP in the 2xx contains an offered SDP, and this SDP SHOULD be a valid update to the last SDP offered by the UAS in reliable provisional response. An answer to this SDP will then arrive in the ACK.

Open Issue: are we sure? If the INVITE had SDP, an alternative is for the 2xx to be an updated offer, followed by an answer in the ACK. However, besides the inconsistency with bis, the bigger problem is that the UAS would need to wait for the ACK before sending media, and therefore clipping would be introduced. The above processing does mean that the SDP in the 2xx may not be a valid answer to the SDP in the INVITE, because intermediate 1xx/PRACK may have added media streams, for example. Is that OK?

It is possible that between the transmission of a final response to the INVITE transaction, and the reception of the ACK for it, a re-INVITE to update the early media is received by the UAS. In this case, the UAS MUST reject the re-INVITE with a 400 class response and include a Retry-After header with a random value between 0 and 10 seconds. This is identical to the handling of INVITE "glare" as specified in bis.

It is possible that between the transmission of a reliable provisional response that updates the early media session, and the reception of the PRACK for it, a re-INVITE to update the early media is received by the UAS. In this case, the UAS MUST reject the re-INVITE with a 400 class response and include a Retry-After header with a random value between 0 and 10 seconds. This is identical to the handling of INVITE "glare" as specified in bis.
5 Call Flow Examples

In this section we detail some call flow examples.

5.1 One-Way Media through ISUP Gateway

In this call flow, a UAC initiates a call that terminates in a PSTN gateway. The gateway wishes to establish a reverse channel towards the caller to play an announcement. So, it sends a 183 with SDP, and includes its offer for early media, which is a one-way PCMU stream from the GW to the caller. The caller generates its SDP as the answer in the PRACK. Shortly afterwards, the call is answered. The call flow is depicted in Figure 1.

The SIP messages are:

(1) UAC -> GW

INVITE sip:+19739525000@gw.com;user=phone SIP/2.0
Via: SIP/2.0/UDP pc.col.edu
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays
To: <sip:+19739525000@gw.com;user=phone>
Contact: sip:ustudent@pc.col.edu
Supported: em, 100rel
Call-ID: 123456@pc.col.edu
CSeq: 98760 INVITE
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 2890844526 2890844526 IN IP4 pc.col.edu
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

(4) GW -> UAC

SIP/2.0 183 Proceeding
Via: SIP/2.0/UDP pc.col.edu;received=100.101.102.103
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhgff
Contact: sip:gw4.gw.com  
Require: em, 100rel  
Call-ID: 123456@pc.col.edu  
CSeq: 98760 INVITE  
Content-Type: application/sdp  
Content-Length: ...

v=0  
o=        UserA  890844526  890844526 IN IP4  gw4.gw.com  
s=Session SDP  
c=IN IP4  1.2.3.4  
t=0 0  
m=audio 45442 RTP/AVP 0  
a=rtpmap:0 PCMU/8000  
a=sendonly

(5) UAC -> GW

PRACK sip:gw4.gw.com SIP/2.0  
Via: SIP/2.0/UDP pc.col.edu  
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays  
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhggff  
Require: em  
Call-ID: 123456@pc.col.edu  
CSeq: 98761 PRACK  
Content-Type: application/sdp  
Content-Length: ...

v=0  
o=        UserA  2890844526  2890844527 IN IP4  pc.col.edu  
s=Session SDP  
c=IN IP4  100.101.102.103  
t=0 0  
m=audio 49172 RTP/AVP 0  
a=rtpmap:0 PCMU/8000

(6) GW -> UAC

SIP/2.0 200 OK  
Via: SIP/2.0/UDP pc.col.edu;received=100.101.102.103  
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays  
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhggff
Figure 1: One-way media through ISUP gateway

Call-ID: 123456@pc.col.edu
CSeq: 98761 PRACK

Jonathan Rosenberg
(8) GW -> UAC

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc.col.edu;received=100.101.102.103
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhggff
Contact: sip:gw4.gw.com
Call-ID: 123456@pc.col.edu
CSeq: 98760 INVITE
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 890844526 890844526 IN IP4 gw4.gw.com
s=Session SDP
c=IN IP4 1.2.3.4
t=0 0
m=audio 45442 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly

(9) UAC -> GW

ACK sip:gw4.gw.com SIP/2.0
Via: SIP/2.0/UDP pc.col.edu
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhggff
Call-ID: 123456@pc.col.edu
CSeq: 98760 ACK

5.2 UAC Rejects Early Media

In this flow, the UAC calls a gateway that sends a reliable provisional response to establish early media flow for an announcement. The UAC does not wish to receive the early media, so it refuses the stream. The call flow is shown in Figure 2.

Open Issue: There is something interesting in this flow. As per the discussion above, the SDP in the 2xx has to be an answer to the last SDP sent by the UAC, which in this case was the SDP in the PRACK refusing the media stream.
However, if an SDP is offered with a disabled stream, the answer also has to have port zero! Thus, disabling an early media stream means we can’t turn it back on in the 2xx.
doesn’t allow turning back on of disabled media streams, which is a separate issue in any case (instead of refusing, we can hold the stream). It may be that the only sensible solution is that, once early media is used, the 2xx always contains an offer, and the ACK contains the answer. This means we may have SDP in the INV/200/ACK, which I was trying to avoid. Maybe it doesn’t matter.

The SIP messages are:

(1) UAC -> GW

INVITE sip:+19739525000@gw.com;user=phone SIP/2.0
Via: SIP/2.0/UDP pc.col.edu
From: U. Student <sip:ustudent@col.edu>;tag=7ahhsays
To: <sip:+19739525000@gw.com;user=phone>
Contact: sip:ustudent@pc.col.edu
Supported: em, 100rel
Call-ID: 123456@pc.col.edu
CSeq: 98760 INVITE
Content-Type: application/sdp
Content-Length: ...

v=0
o=UserA 2890844526 2890844526 IN IP4 pc.col.edu
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

(4) GW -> UAC

SIP/2.0 183 Proceeding
Via: SIP/2.0/UDP pc.col.edu;received=100.101.102.103
From: U. Student <sip:ustudent@col.edu>;tag=7ahhsays
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhgff
Contact: sip:gw4.gw.com
Require: em, 100rel
Call-ID: 123456@pc.col.edu
CSeq: 98760 INVITE
Content-Type: application/sdp
Content-Length: ...
v=0
c=IN IP4 1.2.3.4
m=audio 45442 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly

(5) UAC -> GW

PRACK sip:gw4.gw.com SIP/2.0
Via: SIP/2.0/UDP pc.col.edu
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhggff
Require: em
Call-ID: 123456@pc.col.edu
CSeq: 98761 PRACK
Content-Type: application/sdp
Content-Length: ...

v=0
c=IN IP4 2890844526 2890844527 IN IP4 pc.col.edu
m=audio 0 RTP/AVP 0
a=rtpmap:0 PCMU/8000

(6) GW -> UAC

SIP/2.0 200 OK
Via: SIP/2.0/UDP pc.col.edu;received=100.101.102.103
From: U. Student <sip:ustudent@col.edu>;tag=7ahhhsays
To: <sip:+19739525000@gw.com;user=phone>;tag=jjhhggff
Call-ID: 123456@pc.col.edu
CSeq: 98761 PRACK

(8) GW -> UAC
<table>
<thead>
<tr>
<th>(1) INVITE</th>
<th>(2) IAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------&gt;</td>
<td>---------------&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(3) ACM</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;--------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(4) 183</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;--------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(5) PRACK</th>
<th>reverse channel opened to UAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(6) 200 PRACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;------------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(7) 183</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;--------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(8) PRACK</th>
<th>video channel opened to UAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------&gt;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(9) 200 PRACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;------------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(10) REL</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;--------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(11) 486</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;--------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(12) ACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UAC</th>
<th>ISUP</th>
<th>PSTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>Switch</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4: UAS Updates Early Media**

### 5.4 Call Forking

In this call flow, a UAC A makes a call that forks at a proxy. The call is forked to two PSTN gateways, B and C. B first sends a 183 to open a reverse media channel. C then sends a 183 to open a reverse
media channel. The UAC doesn’t want to reject the early media streams, but it wants to be able to put one on hold so it can listen to the other. Specifically, when the early media stream from C comes, the UAC wants to put the media stream from B on hold, and listen to C. After the announcement is played, the call is accepted at C. A then sends a CANCEL to terminate the call leg with B, since A effectively "forked" when it performed a re-INVITE to B in order to put its early media stream on hold.

The call flow is shown in Figure 3 and 5.

6 Security Considerations

This mechanism doesn’t introduce any security considerations beyond those already present in SIP. However, this requires further investigation.

7 Acknowledgements

Thanks to Jon Peterson and Gonzalo Camarillo for their detailed reviews and comments.

8 Authors Addresses

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
First Floor
East Hanover, NJ 07936
e-mail: jdrosen@dynamicsoft.com

9 Bibliography


Figure 5: Call Forking: Second half