

SIPPING  
Internet-Draft  
Intended status: Informational  
Expires: April 21, 2007

B. Stucker  
Nortel  
October 18, 2006

Coping with Early Media in the Session Initiation Protocol (SIP)  
draft-stucker-sipping-early-media-coping-03

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 April 21, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

Several mechanisms for early media have been proposed in the past, each attacking a different aspect of the problem. A good example of this is [RFC-3960](#) which talks about two models of early media: the gateway model, and the application model. The gateway model uses a series of offer/answer exchanges to control the rendering of early media, but breaks down in the presence of forking (as mentioned in [section 3 of RFC-3960](#)). The application model relies on the UAS to know when it is generating early media and use [RFC-3959](#) to keep early

Internet-Draft

Coping w/ Early Media in SIP

October 2006

media and regular media streams separate to avoid clipping. Even in the presence of the recommendations in [RFC-3960](#) some problems exist within SIP in the area of early media. Although some of these challenges are likely to never be overcome, for example when interworking with a PSTN gateway that does not take into account CPG or ACM messages (in the case of ISUP). However, the potential to improve on what is already there does exist. This document attempts to go into more detail around early media where [RFC-3960](#) left off, what sorts of mechanisms are in use today in existing implementations to deal with the challenges at hand, derives requirements and a possible mechanism to improve upon the current model. In addition, the document goes into other areas that can complicate or be complicated by the presence of early media (especially with forking) such as SRTP keying and media flow authorization.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">5</a>
<a href="#">3.</a>	Types of Early Media . . . . .	<a href="#">5</a>
<a href="#">3.1.</a>	Pre-routing early media . . . . .	<a href="#">5</a>
<a href="#">3.2.</a>	Pre-presentation early media . . . . .	<a href="#">6</a>
<a href="#">3.3.</a>	Post-presentation early media . . . . .	<a href="#">6</a>
<a href="#">3.4.</a>	Non-SDP early media . . . . .	<a href="#">7</a>
<a href="#">4.</a>	Current common coping mechanisms for early media . . . . .	<a href="#">7</a>
<a href="#">4.1.</a>	Problems with current coping mechanisms . . . . .	<a href="#">8</a>
<a href="#">4.1.1.</a>	Proxy-side coping mechanisms . . . . .	<a href="#">8</a>
<a href="#">4.1.1.1.</a>	Proxy SDP stripping . . . . .	<a href="#">8</a>
<a href="#">4.1.1.2.</a>	Proxy SDP weighting . . . . .	<a href="#">9</a>
<a href="#">4.1.2.</a>	Client-side coping mechanisms . . . . .	<a href="#">9</a>
<a href="#">4.1.2.1.</a>	Client detection of forking . . . . .	<a href="#">9</a>
<a href="#">4.1.2.2.</a>	Client slow-start INVITE . . . . .	<a href="#">10</a>
<a href="#">4.1.2.3.</a>	Client Usage of Gateway Model . . . . .	<a href="#">10</a>
<a href="#">4.1.2.4.</a>	Client Usage of Application Server Model . . . . .	<a href="#">10</a>
<a href="#">5.</a>	Requirements . . . . .	<a href="#">10</a>
<a href="#">5.1.</a>	Deprecation of forking . . . . .	<a href="#">11</a>
<a href="#">5.2.</a>	Deprecation of early media . . . . .	<a href="#">11</a>
<a href="#">5.3.</a>	Originating UA's to render early media . . . . .	<a href="#">12</a>
<a href="#">5.4.</a>	Downstream signaling of acceptance . . . . .	<a href="#">12</a>
<a href="#">5.5.</a>	Upstream signaling of importance . . . . .	<a href="#">13</a>
<a href="#">5.6.</a>	Universal backward-compatibility . . . . .	<a href="#">13</a>
<a href="#">5.7.</a>	Recursive forking . . . . .	<a href="#">13</a>

5.8.	Media Gating . . . . .	14
6.	Recommendations . . . . .	14
6.1.	Early Media Classification and Prioritization . . . . .	14
6.1.1.	Overview . . . . .	14
6.1.1.1.	Early-Media Classifications . . . . .	15

Stucker

Expires April 21, 2007

[Page 2]

Internet-Draft

Coping w/ Early Media in SIP

October 2006

6.2.	Early Media Flow Negotiation . . . . .	16
6.2.1.	Overview . . . . .	16
6.2.2.	SDP parameters . . . . .	16
6.2.3.	Usage of emflow with offer/answer . . . . .	17
6.2.3.1.	Meaning of a=emflow:none . . . . .	17
6.2.3.2.	Meaning of a=emflow:send . . . . .	17
6.2.3.3.	Meaning of a=emflow:recv . . . . .	18
6.2.3.4.	Meaning of a=emflow:sendrecv . . . . .	18
6.2.3.5.	Usage of RTP-SSRC-Value . . . . .	18
6.2.4.	Option tag for emflow . . . . .	19
6.2.5.	Example . . . . .	19
6.3.	Early Media and SRTP . . . . .	20
7.	Security Considerations . . . . .	21
8.	IANA Considerations . . . . .	21
9.	References . . . . .	22
9.1.	Normative References . . . . .	22
9.2.	Informational References . . . . .	22
	Author's Address . . . . .	23
	Intellectual Property and Copyright Statements . . . . .	24

## 1. Introduction

One of the mechanisms within SIP [[RFC3261](#)] that has caused much consternation (and interesting service scenarios) is forking, especially forking of INVITE requests. This is where a SIP INVITE request sent to a SIP proxy is resolved into two or more destinations which are signaled in parallel or sequentially by the proxy. When this occurs, multiple downstream parties will receive similar INVITE requests to initiate a SIP session from a given originating SIP user agent (UA). This creates the possibility of race conditions where the ordering of the provisional and final responses to this request, as observed by the originating SIP UA, may potentially arrive in any order, or not at all.

Another mechanism in SIP that looks simple, but causes difficult interactions, was introduced to handle SIP to PSTN interworking. Because the PSTN has a specific set of behaviors which require that only one endpoint in the PSTN network (typically the last PSTN switch reached) may generate media back to the originator of a PSTN call, generation of early media (media produced prior to the intended terminator of a call answering the call) is relatively straightforward. In SIP, this PSTN interaction with early media was handled by allowing any endpoint that has received an SDP offer as part of setting up a session to be able to immediately generate media back to the SDP offerer. Further, the SDP offerer was obligated to be prepared to render any media received at the location specified in the SDP offer at any time as long as the session was in a setup or stable state.

Each of these mechanisms, taken separately, can create complex signaling flows and difficult service interactions to resolve. Together, however, they compound the effects of one another to create an area of study that has been open within the SIP design community for some time. Several extensions to [\[RFC3261\]](#) have been proposed to handle some of the various effects that early media suffers from, most notably [\[RFC3959\]](#) and [\[RFC3960\]](#). However, none have fully attacked a few key areas of interest:

- o Controlling the order and timing of early media stream rendering at the originating SIP UAC.
- o Knowing under what general conditions early media flows are potentially being sent to the originating SIP UAC.

This document seeks to capture the salient requirements for these areas, and propose a mechanism for handling these early media interactions in a more predictable manner.

## [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.](#) Types of Early Media

Not all early media is created equal, some types are more problematic than others. There are four generic types of early media within SIP:

1. Pre-routing early media - This is early media that is conveyed via SDP and is presented to the originator by a proxy before routing on the URI is started.
2. Pre-presentation early media - This is early media that is presented to the originator, conveyed via SDP, by a proxy after the URI has been routed upon, but before any forwarding of the INVITE request has occurred.
3. Post-presentation early media - This is early media that is presented to the originator, conveyed via SDP, by either a forking proxy or any subsequent hop after the INVITE request has been forwarded from the proxy.

4. Non-SDP early media - This is early media that may be presented to the originator at any time through means other than SDP, such as the Alert-Info header as defined in [[RFC3261](#)]

### [3.1.](#) Pre-routing early media

Pre-routing early media is typically generated and characterized by a proxy that has an associated media resource. An example of this type of early media would be a brief 'branding' message that is played to the originator thanking them for using the service provider associated with the originator's local outbound proxy. When the message ends, the media resource signals this to the proxy and routing of the request continues per [[RFC3261](#)]

This type of early media typically does not pose the originator's local outbound proxy any issues unless the client is using one of the mechanisms defined in [Section 4.1.2](#) or something similar. This is because the proxy is in complete control over the pace at which the terminator will be routed to relative to the media stream being presented. If the proxy attempting to present pre-routing early media to the originator is a subsequent proxy from the originator's local outbound proxy, then the service may not work due to upstream proxies employing one of the mechanisms described in [Section 4.1.1](#)

### [3.2.](#) Pre-presentation early media

Pre-presentation early media is similar to pre-routing early media except that it may take into account the routes that the proxy is about to route the INVITE request to in its decision of what to play. This may allow the proxy to employ one of the proxy-side early media coping mechanisms defined in [Section 4.1.1](#). Likewise, the proxy may inject its own SDP answer into the signaling to the originator to kick off services like colorful ringback tone (CRBT) where the originator is hearing a recording (typically music) selected by the terminator while the network attempts to reach the terminating party.

Pre-presentation early media also differs from non-SDP early media in that the proxy or proxies are manipulating the SDP offer/answer rather than SIP headers such as Alert-Info (as defined in [[RFC3261](#)])

to signify what media the originator should be rendering. There are several potential reasons why the Alert-Info header is not used in this case: the service may be interactive, requiring two-way media in order to work (such as digit collection for a credit card number), or may not want to rely on the originator's ability to render the information in the Alert-Info header to the end user (such as a call originating from the PSTN through a SIP gateway).

### [3.3.](#) Post-presentation early media

Post-presentation early media is most typically characterized by the ugly interactions that arise between it and forking. Since this is early media that has come about after the proxy has potentially caused multiple endpoints to be contacted, and therefore the possibility that multiple early media streams may have been triggered, it is commonly considered to be the worst-case scenario with early media.

To compound the basic issue at play, the presence of forking can confuse the type of early media being presented to the originator. A downstream proxy that has received a forked request may not be aware that the INVITE has forked as a B2BUA may have forked the request. As a result, the proxy may be acting as if it is the only proxy to handle the request from the originator, and operate in a pre-routing, pre-presentation, or non-SDP early media mode despite the fact that the early media reaching the originator is post-presentation. Therefore, unless the proxy is the originator's edge proxy, it cannot necessarily determine what kind of early media it may actually be sending to the originator.

### [3.4.](#) Non-SDP early media

Non-SDP early media is typically characterized by the presence of an Alert-Info header [[RFC3261](#)]. The Alert-Info header specifies a URI that the originator may go to in order to receive a file or stream that contains information (such as a wave recording) about the ringback tone the terminator wishes the originator to hear. It is somewhat simpler in that it is not part of the offer/answer model,

and that it is not trying to create a two-way media stream. The interaction between inband ringback, client generated (local) ringback, and other forms of early media is spelled out in [[RFC3960](#)]. It is worth noting that rendering the Alert-Info header contents should only be done when the origin of the header is trusted (per [[RFC3261](#)]), so this may limit its usefulness to a considerable degree. The remainder of this document assumes that the UAC and UAS follows the advice in [[RFC3960](#)] with respect to interactions with early media.

Although non-SDP early media is for future study, it is envisioned that this document would clarify the behavioral interactions between non-SDP early media and other types of early media.

#### [4.](#) Current common coping mechanisms for early media

A number of mechanisms exist for coping with early media. They all rely, generally, on 'fixing' the early media problem by 'breaking' the behaviors specified in other RFCs (or at least bending the spirit of them to some extent):

1. Proxy SDP stripping - If a proxy detects that it is about to fork an INVITE, it keeps track of this fact in its processing state for the INVITE transaction. Any SDP answers in provisional responses are stripped before being forwarded upstream. The SDP answer may be added into a 200 response upstream from last provisional SDP answer received if SDP is not already present in the message to ensure that the offer/answer exchange is completed. This effectively turns off early media.
2. Proxy SDP weighting - If a proxy detects that it had previously forked the INVITE to which it is now receiving a provisional response it may allow a particular provisional response to retain the SDP answer in the message body and strip other SDP answers in provisional responses per the proxy SDP stripping methodology. This mechanism is used to favor SDP that the proxy may have some control over. For instance, if the proxy knows that one forked leg is to a media server streaming CRBT media to the originator, it may allow that SDP answer to flow back, but block all other SDP answers on other legs in the meantime.

3. Client detection of forking - Clients may start out playing local



ringback to the originator until the first SDP answer is received. When the first SDP answer is received, the client may switch to playing the media for that SDP answer. However, upon detecting that the INVITE forked through subsequent provisionals being received (reception of two or more distinct SDP answers or [\[RFC3261\]](#) 'T0' header tags) the client may irrevocably return to playing local ringback. At this point, the client is likely to continue to playing local ringback until the call is answered, or an error condition arises.

4. Client slow-start - Clients may wish to simply not include any SDP in their initial INVITE message in order to accumulate a set of SDP offers from their prospective terminating endpoints. Such INVITEs are known as 'slow-start' INVITEs, because the SDP offer/answer exchange gets off to a 'slow start'. These may also be used in protocol interworkings (notably H.323 to SIP) with no intent as to managing early media. The client can either use PRACK or UPDATE to respond to offers received in provisional responses at the point in time the originating client wishes to stage early media streams.

#### [4.1.](#) Problems with current coping mechanisms

##### [4.1.1.](#) Proxy-side coping mechanisms

###### [4.1.1.1.](#) Proxy SDP stripping

This is a very common mechanism, perhaps second only to the two client mechanisms mentioned above. When a proxy employs this mechanism, it remembers when forking has occurred and removes any SDP in provisional responses as a result. This means that if the originator supports reliable provisional responses (100rel) as defined in [\[RFC3262\]](#), that this option tag must be removed by the proxy before forwarding the INVITE to each forked leg. Otherwise it may be forced to potentially handle SDP in negotiation within a PRACK transaction for the originating client with little or no information about the originating client's capabilities. In the case that the originator requires support for PRACK the proxy may have to fail the call setup, handle very complex negotiation signaling in the case that the call forks, or simply not fork the call.

Additionally, this mechanism also completely breaks any early media services or announcements, some of which may be critical to proper completion or billing disposition of the call upon answer. For instance, the call may fork to a PSTN gateway that is trying to tell the originator that it is about to bill them \$500 to complete the current call. With proxy SDP stripping this announcement would not be heard by the originator.

#### [4.1.1.2.](#) Proxy SDP weighting

Proxy weighting of SDP can be useful in situations where the proxy knows what is going on with the call routing for each leg. However, lack of information as to why downstream elements are sending SDP in provisional responses can cause proxies to weight the SDP incorrectly. Further, if multiple proxies are traversed, the SDP that is accepted for delivery to the originating UA may not be the SDP selected at any given proxy. There is no indication to downstream network clients as to what has happened with their SDP as it traverses proxys back upstream towards the originator. Likewise, the \$500 warning announcement presented in the previous section may or may not be heard.

#### [4.1.2.](#) Client-side coping mechanisms

##### [4.1.2.1.](#) Client detection of forking

This mechanism is where a client may play audible ringback while waiting for an initial provisional or final response to an INVITE message it originated. When the first provisional response with SDP is received, it may switch from playing audible ringback to rendering the media stream defined in the SDP. If a subsequent provisional response is received from a different endpoint (identifiable by a different to-tag in the 'TO' header as defined in [[RFC3261](#)]) it stops rendering any early media media packets it is receiving and typically returns to audible ringback. Upon receiving a non-3xx final response, the UA switches media appropriately to the response. For 3xx responses, the client continues to play audible ringback if that was what is currently being rendered, or switches (typically) to ringback again if it was rendering media packets. This mechanism is used by client devices for a number of reasons:

- o What gets presented to the end user is predictable.
- o Does not rely on the set of proxies handling any given INVITE request to do anything special.
- o Is easy to implement.

The problem with this approach is that it often causes early media to break altogether. If a leg that the call was forked to is awaiting media from the originating client (such as prompting for digit collection, like a credit card number or extension) that leg's early media may fail due to other provisional responses sent to the originator by other call legs.

Network and terminating services that utilize early media are likely

to fail or work erratically (due to race conditions between messages) when an originating client behaves in this manner. What's worse, is

that there's no indication as to what the originating client is doing to the downstream network elements.

Due to the client switching to locally generated playback, and ignoring early media RTP streams prior to receiving a final response to the INVITE, there is the opportunity for clipping to occur in the SIP signaling path latency lags the media path latency.

#### [4.1.2.2](#). Client slow-start INVITE

Slow-start INVITEs circumvent the problem of having to immediately render media packets from an unknown set of terminating endpoints by not giving those endpoints anywhere to send the media to. However, this mechanism has some serious drawbacks, most notably guaranteed clipping (potentially severe if the SDP offer is not received from the other end until a 200 response is received) and the potential for an increased number of messaging round-trips to setup a call.

Due to some service designs and protocol interworking slow-start INVITEs will continue to be seen, but due to the clipping problems associated with slow-start INVITEs this coping mechanism is considered to be incomplete.

#### [4.1.2.3](#). Client Usage of Gateway Model

Clients typically do not use the [\[RFC3960\]](#) gateway model because of the limitations presented in the RFC around early media and forking with the gateway model in [section 3.1](#).

#### [4.1.2.4](#). Client Usage of Application Server Model

The application server model defined in [\[RFC3960\]](#), along with [\[RFC3959\]](#) define an improved mechanism over the gateway model in that early media is negotiated separately from regular media to reduce media clipping issues. However, there still are problems with UASs that generate early media packets upon receiving the SDP offer from the UAC that cannot currently be distinguished from other media in all situations, and the UAC has no feedback from the various UASs that are generating early media as to which ones are of importance or

otherwise. UASs typically do adhere to the request in [\[RFC3960\]](#) [section 4.1](#) that they not generate superfluous early media streams to assist the UAC with early media rendering.

## 5. Requirements

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

Stucker

Expires April 21, 2007

[Page 10]

---

Internet-Draft

Coping w/ Early Media in SIP

October 2006

interactions:

- R1: Deprecation of forking within the [\[RFC3261\]](#) is considered to be out-of-scope of the possible solutions (sorry Dean).
- R2: Deprecation of early media from within [\[RFC3261\]](#) is considered to be out-of-scope of the possible solutions (sorry again, Dean).
- R3: SIP UAs that are attempting to create a new SIP dialog using the INVITE method should no longer be obligated to blindly render media packets that are delivered to them as a result of an SDP offer sent in the INVITE.
- R4: A mechanism should exist by which an originating SIP UA can signal to a downstream SIP endpoint that it is now willing to accept media packets.
- R5: A mechanism should exist by which a terminating SIP UA can signal to an upstream SIP endpoint what type of early media (if known) it wishes to present to the originating UA, if it requires one-way, or two-way media flows, and the relative importance of the early media.
- R6: Universal backwards-compatibility is a secondary goal. Where possible, backwards-compatibility with clients that do not implement recommendations in this draft should be preserved.
- R7: 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.
- R8: 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.

### [5.1.](#) Deprecation of forking

Deprecation of forking from SIP [\[RFC3261\]](#) is considered to be out of

scope. This is due to the heavy deployment of forking in existing implementations for key routing services. Changes of this nature are considered by the author (and others) to be of too large a scope relative to the problem at hand and are subsequently excluded from this draft in favor of searching for less radical solutions.

## [5.2.](#) Deprecation of early media

Deprecation of early media from SIP [[RFC3261](#)] is considered to be out of scope. Early media is required in order to handle certain PSTN interactions as defined in [RFC-3398](#) [[RFC3398](#)] and elsewhere. In addition, the desire to provide announcements and other media prior to the terminating party answering the call is considered desirable and must therefore use some form of "pre-answer" media (currently known as early media).

Stucker

Expires April 21, 2007

[Page 11]

---

Internet-Draft

Coping w/ Early Media in SIP

October 2006

## [5.3.](#) Originating UA's to render early media

Currently, [section 5.1](#) of the offer/answer model [[RFC3264](#)] states that the offerer in an SDP offer/answer exchange must be prepared to receive media from media streams described in the offer as being 'recvonly' or 'sendrecv'. Further, in [section 6.1 of](#) [[RFC3264](#)] it states that the answerer in an SDP offer/answer exchange may immediately send media to media streams that are described in the answer as being 'sendrecv' (note: [[RFC3264](#)] does not explicitly state as much, but it is assumed that media streams that are 'sendonly' in the SDP answer can also have media immediately sent to them by the SDP offerer).

These statements, taken together, create an obligation upon the originating UA to render any early media sent to them by anyone to whom their SDP offer was delivered (unless the media stream was defined to be 'sendonly' or 'inactive'). This is useful in resolving the PSTN interactions in [[RFC3398](#)], especially as noted in the example call flows and ACM message processing in [section 7](#) of that document. This obligation on the part of the originating UA has subsequently been used in the absence of actual PSTN interworking to provide services that mimic the PSTN network (such as providing far-end announcements), or provide other services such as colorful ringback tones (CRBT) in which media is streamed to the originator while the terminator is being located/alerted.

The argument can, and has, been made that simply because a service exists in the PSTN world, that it does not mean that it must exist within SIP. However, given the prevalence of services that utilize early media, and the number of RFCs that talk about dealing with various aspects of early media, this particular train appears to have long ago left the station. It is not the intent of this document to pass judgement upon these services, but to find a way to cope with them in a more robust manner than currently is available.

The obvious downside to this property of [\[RFC3264\]](#) is that while the offerer may have limited control over the delivery of their SDP offer, they have an obligation to render anything sent to them. This severely restricts the policies that the offerer (as the originator) may use to decide to render early media, which needs to be augmented.

#### [5.4.](#) Downstream signaling of acceptance

An INVITE with SDP should serve two simple purposes: establish the path by which all signaling shall follow to/from the originator and the set of terminating clients, and to let each terminating party know what sort of communications the originator can and will engage in. Currently, SDP offers also imply tacit acceptance of any and all

media that might be generated in the reverse path upstream towards the originator. This should not necessarily always be the case, and a mechanism whereby the originator may assert that it is further ready to receive media packets is needed. The originator may wish to imply a combination of early and final media acceptance or denial in order to prevent unruly early media interactions and clipping of final media.

#### [5.5.](#) Upstream signaling of importance

A provisional response from a terminating party currently implies that the terminating party is listening to the SIP signaling it is receiving, and (if an SDP answer is present) the type of communications that the terminator wishes to engage in (if any). What is missing is a way for the terminating party to tell upstream entities what sort of demands it has upon the originator for rendering of its early media, and the relative importance associated with the media that it generates towards the originator. This helps

the originator decide what is important and what is not when choosing which media stream it should render (if it wishes to, see [Section 4.1.2.1](#)).

## [5.6.](#) Universal backward-compatibility

There are scenarios in which there is no way to cope appropriately with early media streams. An example would be a call that forks to an ISUP PSTN gateway as defined in [[RFC3398](#)] that is ignorant of the content of early media it is generating. There is no reliable indication in ISUP CPG or ACM messages as to what the other end might be doing for early media. It is possible that a cause code is present in the CPG in some ISUP to legacy platform interworking scenarios, but these are not present generally in ISUP signaling flows, and therefore cannot be relied upon. Mechanisms to deal with these types of devices is currently for future study and not explored further here at this time.

## [5.7.](#) Recursive forking

The mechanism should be able to deal with recursive forking scenarios. This would be where two or more independent proxies fork a given INVITE request from an originating client. In this case, the proxies are normally not coordinated in their operations. As a result, the mechanism proposed should be robust enough to allow for both end-to-middle and end-to-end negotiation of early media.

## [5.8.](#) Media Gating

In many network environments, it is common for the media flow to be 'gated' in some way. Gating refers to a practice whereby an element in the network is examining the signaling (SIP and SDP) being exchanged by UAs and is sending instructions to a middlebox as to when media packets are authorized to flow between UAs. This gating behavior is typically used to prevent theft of service. As a result of this gating behavior, any mechanism used to identify or coordinate early media should not employ media packet exchanges. It is allowable for early media itself to be marked as such in the media

packets, however, because gating behavior does not interact negatively with such a mechanism. Operations that require early media packet behavior by the UAC may fail in the presence of gating.

## [6.](#) Recommendations

The following sections include recommendations that create a framework that is capable of both identifying/prioritizing the type of early media being presented to the originator, and giving the originating client a means by which it can control the order in which early media flows are presented to it.

### [6.1.](#) Early Media Classification and Prioritization

#### [6.1.1.](#) Overview

Regardless of the mechanism that is used to control the presentation of early media, if at any point more than one endpoint is attempting to stream early media to the originator a few problems arise:

- o Nobody upstream of the device attempting to stream early media to the originator is aware of what exactly it is that the early media generator is generating. Is it advertising? Is it an important message? Who knows. This is important not only for the originating client (see [Section 4.1.2.1](#)), but proxies as well since they may be employing a weighting mechanism as described in [Section 4.1.1.2](#).
- o The device generating the early media may have no idea how many other devices that are peer to it or downstream from it are also trying to generate early media. Again, this is important if the client is using the client-side detection of forking mechanism defined in [Section 4.1.2.1](#).
- o Multiple streams may be included in the offer, not all of which are suitable or intended for early media. For instance, an offer may include video and audio streams. Early media may only be streamed to the audio port during call setup. Another example

would be the inclusion of RTP and SRTP streams where only the RTP stream is intended for early media. Therefore, the UAC may not wish to apply early-media coping mechanisms to all streams offered.



In order to rectify this situation, proper classification of the possible early media to be sent after completion of the SDP exchange is needed and a specific linkage of that classification to particular streams is highly desirable. This can be handled either by inclusion of SIP headers in the message carrying SDP sent towards the originator or by inclusion in the SDP itself. If the classification is handled in the SDP itself, this limits the ability of intermediaries to use this information to update the SDP as the message body may be covered by an integrity protection mechanism or may be otherwise unavailable (for example, the SDP could be encrypted). If the classification is handled in the SIP headers, then it may be unclear as to which SDP stream the classification applies to. If classification is handled via a SIP header (previous revisions of this document referred to an 'Early-Media-Class' header), then it is recommended that the SIP header only apply to SDP covered by an Early-Session content disposition as defined in [\[RFC3959\]](#). This allows the UAC to clearly understand which streams the classification applies to. In either case, via SIP or SDP, upon answer of the INVITE, all processing of media streams and SDP should revert to [\[RFC3261\]](#) rules as the call is answered and no media from this point on should be considered 'early'.

#### [6.1.1.1](#). Early-Media Classifications

The following list is given to show a possible set of common early-media classifications. Each class is given in increasing order of importance.

1. [RFC-3264](#) - The default behavior defined in [RFC-3264](#) is requested.
2. Advertisement - A non-critical advertisement.
3. Warning - A non-critical announcement.
4. Two-way - The endpoint presenting early media wishes to establish a two-way early media session before completing the call.
5. Critical - A critical announcement, such as: "We're about to bill you for \$10k".
6. Unknown - The nature of the early media being presented to the originator is unknown (such as from a PSTN gateway receiving a generic announcement.)

Early media classified as "Unknown" must unfortunately be considered of the highest importance: there's no indication given that qualifies it to be of lower importance. It is recommended that unclassified early media would be treated as [RFC-3264](#). This is to prevent network

elements that do not classify their early media from overriding elements that are more forthcoming. An additional q-value, such as that defined in [section 20.10 of \[RFC3261\]](#), can be used to break ties between classifications.

## [6.2.](#) Early Media Flow Negotiation

The following sections take the requirements from [Section 5](#) and tries to create a mechanism that can satisfy them. This mechanism is built along similar lines as the SIP preconditions framework [\[RFC3312\]](#).

### [6.2.1.](#) Overview

A simple mechanism is introduced that tells terminators what the originator expects to have happen with respect to early media. This information may also be of use to intermediate nodes that also wish to generate early media. The mechanism differs from the SDP [\[RFC2327\]](#) 'a=recvonly', 'a=sendonly', 'a=sendrecv', and 'a=inactive' attributes in that the final media flow mode can be negotiated and ready upon answer without further messaging, and from the preconditions [\[RFC3959\]](#) SDP attributes in that QoS can be negotiated separately as well.

### [6.2.2.](#) SDP parameters

The following media-level parameters are defined:

```
early-media-flow-status = "a=emflow:" direction-tag [ COMMA rtp-ssrc-value ]
direction-tag = ("none" | "send" | "recv" | "sendrecv")
rtp-ssrc-value = 1 * 8hex
```

The early-media-flow-status 'a=emflow' denotes two things:

- o The current state of the early media from the perspective of the originating party of the call as specified by the direction-tag.
- o The RTP SSRC for a given early media stream (as defined in [section 8 of \[RFC3550\]](#)) to facilitate correlation of RTP packets with a particular early media session. It is possible for this value to be the same for two different early media stream. The intent of this is to give the UAC a starting point to work from.
  - ISSUE: What should the UAC do if it sees that the RTP SSRC in two or more early media flows collides?
  - ISSUE: How stable are RTP SSRC values during call setup?

It is expected that the directionality indicators defined in [\[RFC2327\]](#) as 'a=sendrecv', 'a=sendonly', 'a=recvonly', and 'a=inactive' are otherwise unaffected. Likewise, preconditions, as

defined in [\[RFC3312\]](#) are likewise unaffected. The emflow values may

be changed in subsequent offer/answer exchanges to allow the originator to properly stage multiple early media streams according to the Early-Media-Class header values. For example, an originator may specify 'a=emflow:none' initially to suppress all early media flows, and then send an UPDATE with a new SDP offer to an endpoint the originator received an early media indication from with 'a=emflow:recv' to denote that the originator is now willing to receive early media.

Regardless of the value of this parameter, both endpoints may immediately begin exchanging media packets upon answer according to [\[RFC3261\]](#), [\[RFC3264\]](#) and [\[RFC2327\]](#). Intermediate proxies should honor this indication, and adjust their behavior accordingly, potentially causing them to divert from their normal early media coping mechanisms.

### [6.2.3.](#) Usage of emflow with offer/answer

#### [6.2.3.1.](#) Meaning of a=emflow:none

If the emflow value of 'none' is set in an the SDP offer, it indicates that the endpoint generating the offer will not accept early media and that anyone accepting this SDP offer MUST NOT send early media. If the emflow value in the SDP offer was 'none', then the emflow value in the SDP answer MUST be set to 'none' as well.

If the emflow value of 'none' is set in an SDP answer, it indicates that the endpoint generating the answer will not generate early media. The SDP offeror can take this indication to mean that they should not expect early media packets from this endpoint per [\[RFC3264\]](#), and that any received prior to answer from this source MAY be discarded.

#### [6.2.3.2.](#) Meaning of a=emflow:send

If the emflow value of 'send' is set in an the SDP offer, it indicates that the endpoint generating the offer may send early media packets, but will not accept early media. Anyone accepting this SDP offer MUST NOT send early media, but SHOULD process received early media packets if it is appropriate to the device receiving packets to

do so. If the emflow value in the SDP offer was 'send', then the emflow value in the SDP answer MUST be set to 'none' or 'recv' depending on whether the application intends to process the early media packets that the offeror may send to it.

If the emflow value of 'send' is set in an SDP answer, it indicates that the endpoint generating the answer may generate early media but will not process any sent to it. Any early media sent to it per

[RFC3264] MAY be discarded. The SDP offeror can take this indication to mean that they should expect early media packets from this endpoint and behave appropriately.

#### [6.2.3.3](#). Meaning of a=emflow:recv

If the emflow value of 'recv' is set in an the SDP offer, it indicates that the endpoint generating the offer may be sent early media packets, but will not generate early media. Anyone accepting this SDP offer MAY send early media, but SHOULD NOT expect to receive early media from the SDP offeror, and that any media packets received prior to answer from this the offeror may safely be discarded. If the emflow value in the SDP offer was 'recv', then the emflow value in the SDP answer MUST be set to 'none' or 'send' depending on whether the application intends to send the early media packets to the offeror or not.

If the emflow value of 'recv' is set in an SDP answer, it indicates that the endpoint generating the answer will accept early media but will not generate any. The SDP offeror can take this indication to mean that they should not expect early media packets from this endpoint and may safely discard any received prior to answer.

#### [6.2.3.4](#). Meaning of a=emflow:sendrecv

If the emflow value of 'sendrecv' is set in an the SDP offer, it indicates that the endpoint generating the offer may send and receive early media packets. Anyone accepting this SDP offer MAY send early media, and SHOULD process received early media packets if it is appropriate to the device receiving packets to do so. If the emflow value in the SDP offer was 'sendrecv', then the emflow value in the SDP answer MAY be set to any value. The value set in the SDP answer depends on if the endpoint answering the SDP offer intends to send

and/or receive early media packets.

If the emflow value of 'sendrecv' is set in an SDP answer, it indicates that the endpoint generating the answer may generate and receive early media and behave appropriately.

#### [6.2.3.5.](#) Usage of RTP-SSRC-Value

The RTP-SSRC value is useful in helping endpoints correlate incoming RTP packets with SDP offer/answer exchanges. The value used in this tag is the SSRC value used in the header portion of an RTP packet as defined in [\[RFC3550\]](#). The SSRC value in an RTP packet is used to define a means for an endpoint to synchronize RTP packets sent from a particular source. As such, the SSRC value must be unique for a given RTP stream.

Stucker

Expires April 21, 2007

[Page 18]

---

Internet-Draft

Coping w/ Early Media in SIP

October 2006

The worst-case expectation for uniqueness of an SSRC value during the offer/answer SDP phase of RTP resource allocation is given in [Section 8 of \[RFC3550\]](#) as  $10^{-4}$  if there are 1000 different RTP streams being offered. As the number of RTP streams typically used in a call setup, even with significant forking involved, is likely to be 0(10) or fewer, the likelihood of each RTP stream getting a unique SSRC number early on is good. If a collision is detected, then [\[RFC3550\]](#) defines a mechanism for detecting this and reselecting a unique SSRC value. This re-selection does not require another SDP exchange today, but if necessary, an SDP exchange could be initiated through a target refresh of the INVITE dialog to update the SDP offer and/or answer with the update SSRC value in the emflow parameter.

#### [6.2.4.](#) Option tag for emflow

The option tag "emflow" is defined for use in the Require and Supported header fields [\[RFC3261\]](#). An offerer MAY include this tag in a Require header if they wish to ensure that any endpoint reached supports this extension (typically when 'a=emflow:' is not set to 'sendrecv'). Then if the party generating an SDP offer or answer supports this extension it MUST include this tag in a Supported header if it is not already in a Require header of any message containing SDP. This allows the other party or parties involved in the signaling flow to know that the other end is processing their emflow values.

#### [6.2.5.](#) Example

The following figures show a simple offer/answer exchange in which the UAC does not wish to receive early media automatically. The UAS then answers indicating that it has a warning announcement it would like to play as early media. The UAC then updates the emflow value to allow the warning announcement to proceed.

```
v=0
o=alice 2890844526 2890844526 IN IP4 uac.anywhere.com
s=
c=IN IP4 uac.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=emflow: none, 1e4f381
```

Figure 1: An SDP offer with no early media allowed and SSRC 1e4f381.

```
...
Early-Media-Class: Warning; q=1.0
```

```
v=0
o=bob 2890844730 2890844730 IN IP4 uas.example.com
s=
c=IN IP4 uas.example.com
t=0 0
m=audio 49920 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=emflow: none, 23a73c01
```

Figure 2: A SIP early media answer to the offer with SSRC of 23a73c01.

```
v=0
o=alice 2890844526 2890844526 IN IP4 uac.anywhere.com
s=
```

```
c=IN IP4 uac.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=emflow: recv, 1e4f381
```

Figure 3: An SDP offer with early media allowed towards the UAC.

```
v=0
o=bob 2890844730 2890844730 IN IP4 uas.example.com
s=
c=IN IP4 uas.example.com
t=0 0
m=audio 49920 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=emflow: send, 23a73c01
```

Figure 4: An SDP answer to the offer acknowledging that early media will be sent using RTP SSRC 23a73c01.

### [6.3.](#) Early Media and SRTP

One of the challenges in dealing with SRTP is the initial key exchanges required to support it. The draft [\[I-D.wing-rtpsec-keying-eval\]](#) discusses a number of keying mechanisms, concentrating on their interaction with SIP. Given the amount of effort likely involved to establish a secure media flow, it is undesirable to require that early media be secure. After all, an

attacker can likely surmise from a 180 Ringing response to an INVITE that the originator is probably hearing ringback. It is the conversation that typically seeks to be protected, therefore securing early media in many situations is likely wasteful.

Additionally, there may be issues where the early media coping mechanisms mentioned in [Section 4](#) are employed that prevents SRTP keying exchanges from taking place in a timely manner. This can cause a number of potentially poor outcomes, especially when SDP is stripped or otherwise manipulated during call setup by a network element.

Finally, network elements that wish to generate early media typically serve many endpoints simultaneously. This means that they do not have the computational power available to support key exchange and encryption without an undesirable reduction in the amount of traffic that they can handle. Therefore, it is recommended that if a client is offering an SRTP stream, that they also offer a regular RTP stream as well for purposes of early media. This gives the network a separate playground to work with for purposes of establishing early media to the UAC. If the SDP for the early media stream were separated in the SDP offer (possibly using [[RFC3959](#)]) it is conceivable that network elements that employ the mechanisms described in [Section 4](#) would simply leave the SRTP portion of a UAC's offer alone, thereby improving the observed behavior of SRTP and early media by the user and SIP network administrator.

ISSUE: The interaction between the mechanisms outlined in this draft and SRTP clearly warrants more investigation.

## [7.](#) Security Considerations

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

## [8.](#) IANA Considerations

This document defines the SDP media type of "emflow" and the direction-tag values of "none", "send", "recv", and "sendrecv" which will require IANA registration.

## [9.](#) References

### [9.1.](#) 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.

## [9.2.](#) 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.
- [RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", [RFC 3262](#), June 2002.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [RFC3312] Camarillo, G., Marshall, W., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", [RFC 3312](#), October 2002.
- [RFC3398] Camarillo, G., Roach, A., Peterson, J., and L. Ong, "Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping", [RFC 3398](#), December 2002.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [RFC3959] Camarillo, G., "The Early Session Disposition Type for the Session Initiation Protocol (SIP)", [RFC 3959](#), December 2004.
- [RFC3960] Camarillo, G. and H. Schulzrinne, "Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)", [RFC 3960](#), December 2004.

[I-D.wing-rtppsec-keying-eval]

Audet, F. and D. Wing, "Evaluation of SRTP Keying with SIP", [draft-wing-rtpsec-keying-eval-01](#) (work in progress), June 2006.

#### Author's Address

Brian Stucker  
Nortel  
2201 Lakeside  
Richardson, TX 75082  
US

Phone: +1 972 685 7724  
Email: [bstucker@nortel.com](mailto:bstucker@nortel.com)  
URI: <http://www.nortel.com/>

Internet-Draft

Coping w/ Early Media in SIP

October 2006

## Full Copyright Statement

Copyright (C) The Internet Society (2006).

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 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](mailto:ietf-ipr@ietf.org).

## Acknowledgment

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