Internet Engineering Task Force
Internet Draft

SIPPING WG
G. Camarillo
Ericsson
H. Schulzrinne
Columbia University

draft-ietf-sipping-early-media-02.txt
June 1, 2004
Expires: December, 2004

Early Media and Ringing Tone Generation in the Session Initiation Protocol (SIP)

STATUS OF THIS MEMO

Abstract

This document describes how to manage early media in SIP using two models; the gateway model and the application server model. It also describes the inputs one needs to consider to define local policies for ringing tone generation.

Table of Contents

1 Introduction

Early media refers to media (e.g., audio and video) that is exchanged before a particular session is accepted by the called user. Within a dialog, early media occurs from the moment the initial INVITE is sent until the UAS generates a final response. It may be unidirectional or bidirectional, and can be generated by the caller, the callee, or both. Typical examples of early media generated by the callee are ringing tone and announcements (e.g., queuing status). Early media generated by the caller typically consists of voice commands or DTMF tones to drive IVRs.

The basic SIP specification (RFC 3261 [1]) only supports very simple early media mechanisms. These simple mechanisms have a number of problems which relate to forking and security, and do not satisfy the requirements of most applications. This document goes beyond the mechanisms defined in RFC 3261 [1] and describes two models to implement early media using SIP: the gateway model and the application server model.

Although both early media models described in this document are superior to the one specified in RFC 3261 [1], the gateway model still presents a set of issues. In particular, the gateway model does not work well with forking. Nevertheless, the gateway model is needed because some SIP entities (in particular, some gateways) cannot implement the application server model.

The application server model addresses some of the issues present in the gateway model. This model uses the early-session disposition type, which is specified in [2].

The remainder of this document is organized as follows. Section 2 describes the offer/answer model in absence of early media, and Section 3 introduces the gateway model. In this model, the early media session is established using the early dialog established by the original INVITE. Section 3.1, Section 3.2 and Section 3.4

describe the limitations of the gateway model and the scenarios where it is appropriate to use this model. Section 4 introduces the application server model, which, as stated previously, resolves some of the issues present in the gateway model. Section 5 discusses the interactions between the Alter-Info header field in both early media models.

2 Session Establishment in SIP

Before presenting both early media models, we will briefly summarize how session establishment works in SIP. This will let us keep separate features that are intrinsic to SIP (e.g., media being played

before the 200 (OK) to avoid media clipping) from early media operations.

SIP [1] uses the offer/answer model [3] to negotiate session parameters. One of the user agents - the offerer - prepares a session description that is called the offer. The other user agent - the answerer - responds with another session description called the answer. This two-way handshake allows both user agents to agree upon the session parameters to be used to exchange media.

The idea behind the offer/answer model is to decouple the offer/answer exchange from the messages used to transport the session descriptions. For example, the offer can be sent in an INVITE request and the answer can arrive in the 200 (OK) response for that INVITE, or, alternatively, the offer can be sent in the 200 (OK) for an empty INVITE and the answer be sent in the ACK. When reliable provisional responses [4] and UPDATE requests [5] are used, there are many more possible ways to exchange offers and answers.

Media clipping occurs when the user (or the machine generating media) believes that the media session is already established but the establishment process has not finished yet. The user starts speaking (i.e., generating media) and the first few syllables or even the first few words are lost.

When the offer/answer exchange takes place in the 200 (OK) response and in the ACK, media clipping is unavoidable. The called user starts speaking at the same time as the 200 (OK) is sent, but the UAS cannot send any media until the answer from the UAC arrives in the ACK.

On the other hand, media clipping does not appear in the most common offer/answer exchange (an INVITE with an offer and a 200 (OK) with an answer). UACs are ready to play incoming media packets as soon as they send an offer. They do this because they cannot count on the reception of the 200 (OK) to start playing out media for the caller; SIP signalling and media packets typically traverse different paths, and so, media packets may arrive before the 200 (OK) response.

Another form of media clipping (not related to early media either) occurs in the caller->callee direction. When the callee picks up and starts speaking, the UAS sends a 200 (OK) response with an answer and the first media packets in parallel. If the first media packets arrive to the UAC before the answer, and the caller starts speaking as well, the UAC cannot send media until the 2xx response from the UAS arrives.

3 The Gateway Model

SIP uses the offer/answer model to negotiate session parameters (as described in Section 2). An offer/answer exchange that takes place before a final response for the INVITE is sent establishes an "early" media session. Early media sessions terminate when a final response for the INVITE is sent. If the final response is a 2xx, the early media session transitions to a regular media session. If the final response is a non-2xx final response, the early media session is simply terminated.

Media exchanged within an early media session is, not surprisingly, referred to as early media. The gateway model consists of managing early media sessions using offer/answer exchanges in reliable provisional responses, PRACKs, and UPDATEs.

The gateway model presents serious limitations in presence of forking, as described in Section 3.1. Therefore, its use in only acceptable when the UA cannot distinguish between early and regular media, as described in Section 3.4. In any other situation (the majority of UAs), it is strongly recommended that the application server model described in Section 4 is used instead.

3.1 Forking

In the absence of forking, assuming that the initial INVITE contains an offer, the gateway model does not introduce media clipping. Following normal SIP procedures, the UAC is ready to play any incoming media as soon as it sends the initial offer in the INVITE. The UAS sends the answer in a reliable provisional response and can send media as soon as there is media to send. Even if the first media packets arrive to the UAC before the 1xx response, the UAC will play them.

> Note that, in some situations, the UAC does need to receive the answer before being able to play any media. UAs in such a situation (e.g., QoS, media authorization or media encryption is required) use preconditions to avoid media clipping.

On the other hand, if the INVITE forks, the gateway model may introduce media clipping. This happens when the UAC receives different answers to its offer in several provisional responses from different UASs. The UAC has to deal with bandwidth limitations and early media session selection.

If the UAC receives early media from different UASs, it needs to present it to the user. If the early media consists of audio, playing several audio streams to the user at the same time may be confusing. Other media types (e.g., video), on the other hand, can be presented

to the user at the same time. The UAC can, for example, build a mosaic with the different inputs.

However, even with media types that can be played at the same time to the user, if the UAC has limited bandwidth, it will not be able to receive early media from all the different UASs at the same time. Therefore, many times, the UAC needs to choose a single early media session and "mute" the rest of them sending UPDATE requests.

> It is difficult to decide which early media session carry more important information from the caller's perspective. In fact, in some scenarios, the UA cannot even correlate media packets with their particular SIP early dialog. Therefore, UACs typically pick up one early dialog randomly and mute the rest.

If one of the early media sessions that was muted transitions to a regular media session (i.e., the UAS sends a 2xx response), media clipping is likely to appear. The UAC typically sends an UPDATE with a new offer (upon reception of the 200 OK for the INVITE) to unmute the media session. The UAS cannot send any media until it receives the offer from the UAC. Therefore, if the caller starts speaking before the offer from the UAC is received, his words will get lost.

> Having the UAS send the UPDATE to unmute the media session (instead of the UAC) does not avoid media clipping in the backward direction and it causes possible race conditions.

3.2 Ringing Tone Generation

In the PSTN, telephone switches typically play ringing tones to the caller to indicate that the callee is being alerted. When, where and how these ringing tones are generated has been standardized (i.e., the local exchange of the callee generates a standardized ringing tone while the callee is being alterted). A standardized approach to provide this type of feedback for the user makes sense in a homogeneous environment such as the PSTN, where all the terminals have a similar user interface.

This homogeneity is not found among SIP user agents. SIP user agents have different capabilities, different user interfaces and may be used to establish sessions that do not involve audio at all. Because of this, the way a SIP UA provides the user with information about the progress of session establishment is a matter of local policy. For example, a UA with a GUI may choose to display a message on the screen when the callee is being alerted while another UA may choose to show a picture of a phone ringing instead. Many SIP UAs choose to imitate the user interface of the PSTN phones. They provide a ringing

tone to the caller when the callee is being alerted. Such a UAC is supposed to generate ringing tones locally for its user as long as no early media is received from the UAS. If the UAS generates early media (e.g., an announcement or a special ringing tone), the UAC is supposed to play it rather than generating the ringing tone locally.

The problem is that, sometimes, it is not an easy task for a UAC to know whether it should generate local ringing or it will be receiving

early media. A UAS can send early media without using reliable provisional responses (very simple UASs do that) or it can send an answer in a reliable provisional response without any intention of sending early media (this is the case when preconditions are used). Therefore, by only looking at the SIP signalling, a UAC cannot be sure whether or not there will be early media for a particular session. The UAC needs to check if media packets are arriving at a given moment.

> An implementation could even choose to look at the contents of the media packets, since they could carry only silence or comfort noise.

With this in mind, a UAC should develop its local policy regarding local ringing generation. For example, a POTS-like SIP UA could implement the following local policy:

1. Unless a 180 (Ringing) response is received, never generate local ringing.

2. If a 180 (Ringing) has been received but there are no incoming media packets, generate local ringing.

3. If a 180 (Ringing) has been received and there are incoming media packets, play them and do not generate local ringing.

> Note that a 180 (Ringing) response means that the callee is being alerted, and a UAS should send such a response if the callee is being alerted, regardless of the status of the early media session.

At first sight, such a policy may look difficult to implement in decomposed UAs (i.e., media gateway controller and media gateway), but this policy is the same as the one described in Section 2, which must be implemented by any UA. That is, any UA should play incoming media packets (and stop local ringing tone generation if it was being performed) in order to avoid media clipping, even if the 200 (OK) response has not arrived. So, the tools to implement this early media policy are available already to any UA that uses SIP.

Note that, while it is not desirable to standardize a common local

policy to be followed by every SIP UA, a particular subset of more or less homogeneous SIP UAs could use the same local policy by convention. Examples of such subsets of SIP UAs may be "all the PSTN/SIP gateways" or "every 3G IMS terminal". However, defining the particular common policy that such groups of SIP devices may use is outside the scope of this document.

3.3 Absence of an Early Media Indicator

SIP, as opposed to other signalling protocols, does not provide an early media indicator. That is, there is no information about the presence or absence of early media in SIP. Such an indicator could be potentially used to avoid generation of local ringing tone by the UAC when UAS intends to provide in-band ringing tone or some type of announcement. However, due to the way SIP works, such an indicator would, in the majority of the cases, be of little use.

One important reason that would limit the benefit of a potential early media indicator is the loose coupling between SIP signalling and the media path. SIP signalling traverses a different path than the media. The media path is typically optimized to reduce the end-to-end delay (e.g., minimum number of intermediaries) while the SIP signalling path typically traverses a number of proxies providing different services for the session. Due to that reason, it is very likely that the media packets with early media reach the UAC before any SIP message which could contain an early media indicator.

Nevertheless, sometimes, SIP responses arrive at the UAC before any media packet. There are situations when the UAS intends to send early media but cannot do it straight away. For example, UAs using ICE [6] may need to exchange several STUN messages before being able to exchange media. In this situations, an early media indicator would keep the UAC from generating local ringing tone during this time. However, while the early media is not arriving to the UAC, the user would not be aware of the fact that the remote user is being alerted, even though a 180 (Ringing) had been received. Therefore, a better solution would be to apply local ringing tone until the early media packets could be sent from the UAS to the UAC. This solution does not require any early media indicator.

> Note that migrations from local ringing tone to early media at the UAC happen in the presence of forking as well; one UAS sends a 180 (Ringing) response, and later, another UAS starts sending early media.

3.4 Applicability of the Gateway Model

Section 3 described some of the limitations of the gateway model. It produces media clipping in forking scenarios and requires media detection to generate local ringing properly. These issues are addressed by the application server model, described in Section 4, which is the recommended way of generating early media that is not continuous with the regular media generated during the session.

The gateway model is, therefore, acceptable in situations where the UA cannot distinguish between early media and regular media. A PSTN gateway is an example of this type of situation. The PSTN gateway receives media from the PSTN over a circuit, and sends it to the IP network. The gateway is not aware of the contents of the media, and it does not exactly know when the transition from early to regular media takes place. From the PSTN perspective, the circuit is a continuous source of media.

4 The Application Server Model

The application server model consists of having UAS behave as an application server to establish early media sessions with the UAC. The UAC indicates support for the early-session disposition type (defined in [2]) using the early-session option tag. This way, UASs know that they can keep offer/answer exchanges for early media (early-session disposition type) and for regular media (session disposition type) separate.

Sending early media using a different offer/answer exchange than the one used for sending regular media helps avoid media clipping in case of forking. The UAC can reject or mute new offers for early media without muting the sessions that will carry media when the original INVITE is accepted. The UAC can give priority to media received over the latter sessions. This way, the application server model transitions from early to regular media at the right moment.

Having a separate offer/answer exchange for early media also helps UACs decide whether or not local ringing should be generated. If a new early session is established and that early session contains at least an audio stream, the UAC can assume that there will be incoming early media and it can then avoid generating local ringing.

> An alternative model would consist of adding a new stream labeled as "early media" to the original session between the UAC and the UAS using an UPDATE, instead of establishing a new early session. We have chosen to establish a new early session to be coherent with the

mechanism used by application servers that are NOT co-located with the UAS. This way, the UAS uses the same

mechanism as any application server in the network to interact with the UAC.

4.1 In-Band Versus Out-of-Band Session Progress Information

Note that, even when the application server model is used, a UA will have to choose which early media sessions are muted and which ones are rendered to the user. In order to make this choice easier to UAs, it is strongly recommended that information that is not essential for the session is not transmitted using early media. For instance, UAs should not use early media to send special ringing tones. SIP already provides a means to inform the remote user about session establishment progress which does not cause any of the problems associated with early media; the status code and the reason phrase in provisional responses.

5 Alert-Info Header Field

The Alert-Info header field allows specifying an alternative ringing content, such as ringing tone, to the UAC. This header field tells the UAC which tone should be played in case local ringing is generated, but it does not tell the UAC when to generate local ringing. A UAC should follow the rules described above for ringing tone generation in both models. If, after following those rules, the UAC decides to play local ringing, it can then use the Alert-Info header field to generate it.

6 Security Considerations

SIP uses the offer/answer model [3] to establish early sessions in both the gateway and the application server models. User Agents (UAs) generate a session description, which contains the transport address (i.e., IP address plus port) where they want to receive media, and send it to their peer in a SIP message. When media packets arrive at this transport address, the UA assumes that they come from the receiver of the SIP message carrying the session description. Nevertheless, attackers may attempt to gain access to the contents of the SIP message and send packets to the transport address contained in the session description. To prevent this situation, UAs SHOULD

encrypt their session descriptions (e.g., using S/MIME).

Still, even if a UA encrypts its session descriptions, an attacker may try to guess the transport address used by the UA and send media packets to that address. Guessing such a transport address is sometimes easier than it may seem because many UAs always pick up the same initial media port. To prevent this situation, UAs SHOULD use media-level authentication mechanisms (e.g., SRTP [7]). In addition, UAs that wish to keep their communications confidential SHOULD use

media-level encryption mechanisms (e.g, SRTP [7]).

Attackers may attempt to make a UA send media to a victim as part of a DoS attack. This can be done by sending a session description with the victim's transport address to the UA. To prevent this attack, the UA SHOULD engage in a handshake with the owner of the transport address received in a session descriptions (just verifying willingness to receive media) before sending a large amount of data to the transport address. This check can be performed by using a connection oriented transport protocol, by using STUN [8] in an end-to-end fashion, or by the key exchange in SRTP [7].

In any event, note that the previous security considerations are not early media specific, but apply to the usage of the offer/answer model in SIP to establish sessions in general.

Additionally, an early media-specific risk (roughly speaking, an equivalent to forms of "toll fraud" in the PSTN) attempts to exploit the different charging policies some operators apply to early and to regular media. When UAs are allowed to exchange early media for free, but are required to pay for regular media sessions, rogue UAs may try to establish a bidirectional early media session and never send a 2xx response for the INVITE.

On the other hand, some application servers (e.g., Interactive Voice Response systems) use bidirectional early media to obtain information from the callers (e.g., the PIN code of a calling card). So, we do not recommend that operators disallow bidirectional early media. Instead, operators should consider a remedy of charging early media exchanges that last too long, or stopping them at the media level (according to the operator's policy).

7 Acknowledgments

Jon Peterson provided useful ideas on the separation between the gateway model and the application server model.

Paul Kyzivat, Christer Holmberg, Bill Marshall, Francois Audet, John Hearty, Adam Roach, Eric Burger, Rohan Mahy, and Allison Mankin provided useful comments and suggestions.

8 Authors' Addresses

Gonzalo Camarillo
Ericsson
Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland

---

electronic mail: Gonzalo.Camarillo@ericsson.com

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

9 Normative References

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

[2] G. Camarillo, "The early session disposition type for the session initiation protocol (SIP)," Internet Draft draft-ietf-sipping-early-disposition-01, Internet Engineering Task Force, Jan. 2004. Work in progress.

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

10 Informative References

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

[5] J. Rosenberg, "The session initiation protocol (SIP) UPDATE method," RFC 3311, Internet Engineering Task Force, Oct. 2002.

[6] J. Rosenberg, "Interactive connectivity establishment (ICE): a methodology for nettwork address translator (NAT) traversal for the session initiation protocol (SIP)," internet draft, Internet Engineering Task Force, July 2003. Work in progress.

[7] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The secure real-time transport protocol (SRTP)," RFC 3711, Internet Engineering Task Force, Mar 2004.

[8] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy, "STUN - simple traversal of user datagram protocol (UDP) through network address translators (nats)," RFC 3489, Internet Engineering Task Force, Mar. 2003.

Intellectual Property Statement

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.

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.