Internet Engineering Task Force
Internet Draft

SIP WG
G. Camarillo
Ericsson
H. Schulzrinne
Columbia University

draft-camarillo-sipping-early-media-01.txt
February 10, 2003
Expires: August, 2003

Early Media and Ringback Tone Generation in the Session Initiation Protocol

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 which inputs need to be taken into consideration to define local policies for ringback tone generation.

Table of Contents

1 Introduction

Early media refers to media (e.g., audio and/or video) that is exchanged before a particular session is accepted by the called user. Early media within a particular SIP dialog takes place from the moment the initial INVITE is sent until the UAS generates a final response. Early media can be unidirectional or bi-directional and can be generated by the caller or/and the callee. Typical examples of early media generated by the callee are ringback tone and announcements (e.g., queuing status). Early media generated by the caller typically consist of voice commands or DTMF tones to drive IVRs.

The basic SIP spec [1] only supports very simple early media. In order to support fully-featured early media, UAs need to implement some extensions in addition to the basic SIP spec. This document describes two models to implement early media and the extensions needed in each model.

Section 2 introduces the gateway model. In this model, the early media session is established using the early dialog established by the original INVITE. Section 2.1, Section 2.2 and Section 2.3 describe the limitation of the gateway model and the scenarios where it is appropriate to use this model. Section 3 introduces the application server model, which resolves some of the issues present in the gateway model. Section 4 discusses the interactions between the Alter-Info header field in both early media models.

2 The Gateway Model

SIP [1] uses the offer/answer model [2] 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 mechanism 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 [3] and UPDATE requests [4] are used, there are many more possible ways to exchange offers and answers.

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 reliable provisional responses, PRACKs and UPDATEs.

2.1 Media Clipping

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.

Media clipping is closely related to the user's expectations. For example, in the PSTN, there usually isn't media clipping in the forward direction because callers are used to wait until the callee answers in order to start speaking. People do not typically start saying "Hello" while they are hearing a ringback tone.

However, callees in the PSTN are used to pick up the phone and start speaking right away. That is why, in the PSTN, when the callee picks up the phone, the media path is already established. It avoids media clipping in the backward direction.

Unlike in the PSTN, there are some situations involving SIP where the callee accepts a session invitation (e.g., picks up a SIP phone) but the media session has not been established yet. This happens, for instance, when the callee's receives an empty INVITE request and sends an offer in a 200 (OK) response. The UAS will not be able to send any media until it receives the answer in the ACK. Everything the callee says during that round trip time will get lost.

However, the situation described above is a general SIP issue, not specifically related to early media. Therefore, it falls outside of the scope of this document. Section 2.1.1 focuses on the scenarios where the gateway model introduces media clipping.

Another form of media clipping (not related to early media either) occurs in the caller->callee direction. If the callee picks up and starts speaking, the UAS will send a 2xx 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 will not be able to send media until the 2xx

response from the UAS arrives. Section 2.1.1 does not deal with this situation either, since it is not early media related.

2.1.1 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 starts sending media right away. Even if the first media packets arrive to the UAS before the 1xx response, the UAS 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.

However, 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 can 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. Therefore, UACs typically pick up one early media session 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.

2.2 Ringback Tone Generation

In the PSTN, telephone switches typically play ringback tones to the caller to indicate that the callee is being alerted. When, where and how these ringback tones are generated has been standardized (i.e., the local exchange of the callee generates a standardized ringback 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 ringback tone to the caller when the callee is being alerted. Such a UAC is supposed to generate ringback 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 ringback tone), the UAC is supposed to play it rather than generating the ringback tone locally.

The problem is that, sometimes, it is not an easy task for a UAC to know whether it should generate local ringback 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 ringback 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 ringback.

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

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

Note, however, that implementing such a policy in a decomposed gateway (media gateway controller and media gateway) can be complex. The media gateway needs to inform the media gateway controller about the presence of incoming media, and based on that information, the media gateway controller needs to control the generation of local ringback in the media gateway. This type of gateway could choose to generate local ringback upon reception of a 180 (Ringing) response, and mix it with any incoming media that happens to arrive (if it does at all).

> 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.

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.

2.3 Applicability of the Gateway Model

Section 2.1 and Section 2.2 described some of the limitations of the gateway model. It produces media clipping in forking scenarios and requires media detection to generate local ringback properly. These issues are addressed by the application server model, described in Section 3, which is the recommended way of generating early media that is not continuous with the regular media that will be 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.

3 The Application Server Model

The application server model consists of having the UAS behave as any other application server in the session [5]. The UAC includes a Join header field in the initial INVITE. In order to send early media, the UAS establishes a new dialog by sending a new INVITE to the URI in the Join header field.

Sending early media using a different dialog than the one used for sending regular media helps avoid media clipping in case of forking. The UAC can reject or mute new invitations 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 achieves a smooth transition from early to regular media.

Having a separate dialog for early media also helps UAs decide whether or not local ringback should be generated. If a new dialog to send early media is established, and that dialog contains at least an audio stream, the UAC can assume that there will be incoming early media and it can then avoid generating local ringback.

> 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 session. We have chosen to establish a new 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 other application server in the network to interact with the UAC.

4 Alert-Info Header Field

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

5 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 and Rohan Mahy provided useful comments and suggestions.

6 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

7 Bibliography

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

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

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

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

[5] J. Rosenberg, "A framework and requirements for application interaction in SIP," internet draft, Internet Engineering Task Force, Nov. 2002. Work in progress.

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement