Network Working Group
Request for Comments: 5369
Category: Informational

G. Camarillo
Ericsson

October 2008

Framework for Transcoding with the Session Initiation Protocol (SIP)

Status of This Memo

Abstract

This document defines a framework for transcoding with SIP. This framework includes how to discover the need for transcoding services in a session and how to invoke those transcoding services. Two models for transcoding services invocation are discussed: the conference bridge model and the third-party call control model. Both models meet the requirements for SIP regarding transcoding services invocation to support deaf, hard of hearing, and speech-impaired individuals.

Table of Contents

1. Introduction

Two user agents involved in a SIP [RFC3261] dialog may find it impossible to establish a media session due to a variety of incompatibilities. Assuming that both user agents understand the same session description format (e.g., SDP [RFC4566]), incompatibilities can be found at the user agent level and at the user level. At the user agent level, both terminals may not support any common codec or may not support common media types (e.g., a text-only terminal and an audio-only terminal). At the user level, a deaf person will not understand anything said over an audio stream.

In order to make communications possible in the presence of incompatibilities, user agents need to introduce intermediaries that provide transcoding services to a session. From the SIP point of view, the introduction of a transcoder is done in the same way to resolve both user level and user agent level incompatibilities. So, the invocation mechanisms described in this document are generally applicable to any type of incompatibility related to how the information that needs to be communicated is encoded.

> Furthermore, although this framework focuses on transcoding, the mechanisms described are applicable to media manipulation in general. It would be possible to use them, for example, to invoke a server that simply increases the volume of an audio stream.

This document does not describe media server discovery. That is an orthogonal problem that one can address using user agent provisioning or other methods.

The remainder of this document is organized as follows. Section 2 deals with the discovery of the need for transcoding services for a particular session. Section 3 introduces the third-party call control and conference bridge transcoding invocation models, which are further described in Sections 3.1 and 3.2, respectively. Both models meet the requirements regarding transcoding services invocation in RFC 3351 [RFC3351], which support deaf, hard of hearing, and speech-impaired individuals.

2. Discovery of the Need for Transcoding Services

According to the one-party consent model defined in RFC 3238 [RFC3238], services that involve media manipulation invocation are best invoked by one of the endpoints involved in the communication, as opposed to being invoked by an intermediary in the network. Following this principle, one of the endpoints should be the one detecting that transcoding is needed for a particular session.

In order to decide whether or not transcoding is needed, a user agent needs to know the capabilities of the remote user agent. A user agent acting as an offerer [RFC3264] typically obtains this knowledge by downloading a presence document that includes media capabilities (e.g., Bob is available on a terminal that only supports audio) or by getting an SDP description of media capabilities as defined in RFC 3264 [RFC3264].

Presence documents are typically received in a NOTIFY request [RFC3265] as a result of a subscription. SDP media capabilities descriptions are typically received in a 200 (OK) response to an OPTIONS request or in a 488 (Not Acceptable Here) response to an INVITE.

In the absence of presence information, routing logic that involves parallel forking to several user agents may make it difficult (or impossible) for the caller to know which user agent will answer the next call attempt. For example, a call attempt may reach the user's voicemail while the next one may reach a SIP phone where the user is available. If both terminating user agents have different capabilities, the caller cannot know, even after the first call attempt, whether or not transcoding will be necessary for the session. This is a well-known SIP problem that is referred to as HERFP (Heterogeneous Error Response Forking Problem). Resolving HERFP is outside the scope of this document.

It is recommended that an offerer does not invoke transcoding services before making sure that the answerer does not support the capabilities needed for the session. Making wrong assumptions about the answerer's capabilities can lead to situations where two transcoders are introduced (one by the offerer and one by the answerer) in a session that would not need any transcoding services at all.

> An example of the situation above is a call between two GSM (Global System for Mobile Communications) phones (without using transcoding-free operation). Both phones use a GSM codec, but the speech is converted from GSM to PCM (Pulse Code Modulation) by the originating MSC (Mobile Switching Center) and from PCM back to GSM by the terminating MSC.

Note that transcoding services can be symmetric (e.g., speech-to-text plus text-to-speech) or asymmetric (e.g., a one-way speech-to-text transcoding for a hearing-impaired user that can talk).

3. Transcoding Services Invocation

Once the need for transcoding for a particular session has been identified as described in Section 2, one of the user agents needs to invoke transcoding services.

As stated earlier, transcoder location is outside the scope of this document. So, we assume that the user agent invoking transcoding services knows the URI of a server that provides them.

Invoking transcoding services from a server (T) for a session between two user agents (A and B) involves establishing two media sessions; one between A and T and another between T and B. How to invoke T's services (i.e., how to establish both A-T and T-B sessions) depends on how we model the transcoding service. We have considered two models for invoking a transcoding service. The first is to use third-party call control [RFC3725], also referred to as 3pcc. The second is to use a (dial-in and dial-out) conference bridge that negotiates the appropriate media parameters on each individual leg (i.e., A-T and T-B).

Section 3.1 analyzes the applicability of the third-party call control model, and Section 3.2 analyzes the applicability of the conference bridge transcoding invocation model.

3.1. Third-Party Call Control Transcoding Model

In the 3pcc transcoding model, defined in [RFC4117], the user agent invoking the transcoding service has a signalling relationship with the transcoder and another signalling relationship with the remote user agent. There is no signalling relationship between the transcoder and the remote user agent, as shown in Figure 1.

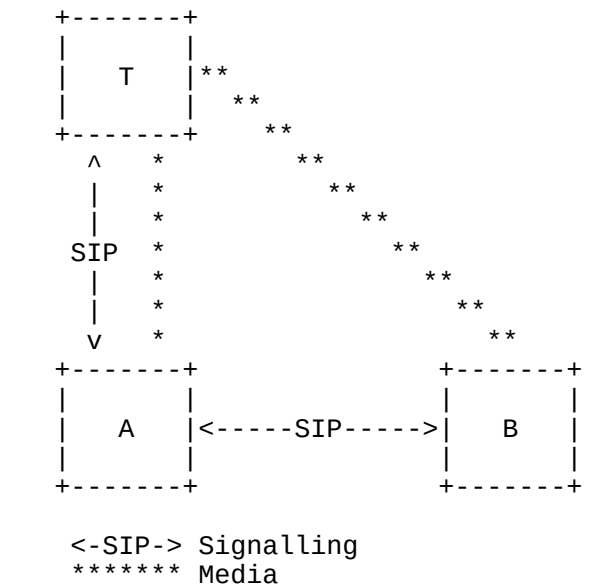

Figure 1: Third-Party Call Control Model

This model is suitable for advanced endpoints that are able to perform third party call control. It allows endpoints to invoke transcoding services on a stream basis. That is, the media streams that need transcoding are routed through the transcoder while the streams that do not need it are sent directly between the endpoints. This model also allows invoking one transcoder for the sending direction and a different one for the receiving direction of the same stream.

Invoking a transcoder in the middle of an ongoing session is also quite simple. This is useful when session changes occur (e.g., an audio session is upgraded to an audio/video session) and the endpoints cannot cope with the changes (e.g., they had common audio codecs but no common video codecs).

The privacy level that is achieved using 3pcc is high, since the transcoder does not see the signalling between both endpoints. In this model, the transcoder only has access to the information that is strictly needed to perform its function.

3.2. Conference Bridge Transcoding Model

In a centralized conference, there are a number of media streams between the conference server and each participant of a conference. For a given media type (e.g., audio) the conference server sends,

over each individual stream, the media received over the rest of the streams, typically performing some mixing. If the capabilities of all the endpoints participating in the conference are not the same, the conference server may have to send audio to different participants using different audio codecs.

Consequently, we can model a transcoding service as a two-party conference server that may change not only the codec in use, but also the format of the media (e.g., audio to text).

Using this model, T behaves as a B2BUA (Back-to-Back User Agent) and the whole A-T-B session is established as described in [RFC5370]. Figure 2 shows the signalling relationships between the endpoints and the transcoder.

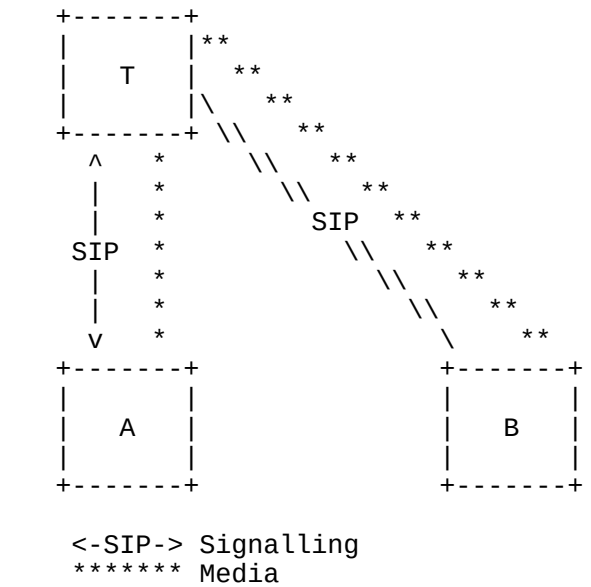

Figure 2: Conference Bridge Model

In the conferencing bridge model, the endpoint invoking the transcoder is generally involved in less signalling exchanges than in the 3pcc model. This may be an important feature for endpoints using low-bandwidth or high-delay access links (e.g., some wireless accesses).

On the other hand, this model is less flexible than the 3pcc model. It is not possible to use different transcoders for different streams or for different directions of a stream.

Invoking a transcoder in the middle of an ongoing session or changing from one transcoder to another requires the remote endpoint to support the Replaces [RFC3891] extension. At present, not many user agents support it.

Simple endpoints that cannot perform 3pcc and thus cannot use the 3pcc model, of course, need to use the conference bridge model.

4. Security Considerations

The specifications of the 3pcc and the conferencing transcoding models discuss security issues directly related to the implementation of those models. Additionally, there are some considerations that apply to transcoding in general.

In a session, a transcoder has access to at least some of the media exchanged between the endpoints. In order to avoid rogue transcoders getting access to those media, it is recommended that endpoints authenticate the transcoder. TLS [RFC5246] and S/MIME [RFC3850] can be used for this purpose.

To achieve a higher degree of privacy, endpoints following the 3pcc transcoding model can use one transcoder in one direction and a different one in the other direction. This way, no single transcoder has access to all the media exchanged between the endpoints.

The fact that transcoders need to access media exchanged between the endpoints implies that endpoints cannot use end-to-end media security mechanisms. Media encryption would not allow the transcoder to access the media, and media integrity protection would not allow the transcoder to modify the media (which is obviously necessary to perform the transcoding function). Nevertheless, endpoints can still use media security between the transcoder and themselves.

5. Contributors

This document is the result of discussions amongst the conferencing design team. The members of this team include Eric Burger, Henning Schulzrinne, and Arnoud van Wijk.

6. References

6.1. Normative References

[RFC3238] Floyd, S. and L. Daigle, "IAB Architectural and Policy Considerations for Open Pluggable Edge Services", RFC 3238, January 2002.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3265] Roach, A.B., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[RFC3351] Charlton, N., Gasson, M., Gybels, G., Spanner, M., and A. van Wijk, "User Requirements for the Session Initiation Protocol (SIP) in Support of Deaf, Hard of Hearing and Speech-impaired Individuals", RFC 3351, August 2002.

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[RFC3850] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling", RFC 3850, July 2004.

[RFC3891] Mahy, R., Biggs, B., and R. Dean, "The Session Initiation Protocol (SIP) "Replaces" Header", RFC 3891, September 2004.

[RFC4117] Camarillo, G., Burger, E., Schulzrinne, H., and A. van Wijk, "Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc)", RFC 4117, June 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5370] Camarillo, G., "The Session Initiation Protocol (SIP) Conference Bridge Transcoding Model", RFC 5370, October 2008.

6.2. Informative References

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

Author’s Address

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Full Copyright Statement

Intellectual Property