Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 08, 2013

A. B. Roach
Mozilla
May 07, 2013

An Approach for Adding RTCWEB Media Streams without Glare
draft-roach-rtcweb-glareless-add-00

Abstract

One of the ongoing challenges in dealing with the massive number of streams that RTCWEB implementations may wish to instantiate and manipulate is the ability to add and remove streams in a way that avoids the condition known as "glare." This document describes a non-normative set of behaviors that RTCWEB implementations can implement to completely avoid inducing a glare condition.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

One of the ongoing challenges in dealing with the massive number of streams that RTCWEB implementations may wish to instantiate and manipulate is the ability to add and remove streams in a way that avoids the condition known as "glare."

In the offer-answer model used by RTCWEB, "glare" arises when offer messages "cross on the wire" (that is, both parties in a session attempt to change the session at the same time). When that happens, both participants must "back off" and wait before attempting to effect their proposed changes. When this happens mid-session, user experience can be negatively impacted.

In a nutshell, the approach eliminates offers "crossing on the wire" by ensuring that only one party ever initiates offers. When a call is established, the RTCWEB application designates one of the two parties as the persistent offerer, and the other as the persistent answerer. These roles are assigned at the very beginning of the session: whichever party makes the first offer will be the persistent offerer.

It is critical to keep in mind that RTCWEB applications, lacking a standardized signaling protocol, can take these actions unilaterally without any further standardization necessary. The purpose of this document is to explain a technique that can be employed by such applications; it defines no normative behavior.

Note that this approach does not eliminate the need for RTCWEB implementations to implement glare handling; such code will be necessary, at the very least, to deal with an attempt to simultaneously initiate two sessions between two endpoints in opposite directions. This technique simply allows implementations to guarantee that such handling is never invoked during an ongoing session.

2. Session Manipulation

The following sections describe how an application would effect various types of session manipulations.

2.1. Offerer Updates Session

If the party that has been designated the "persistent offerer" wishes to update the session, it simply sends a new offer describing the desired session state. The persistent answerer generates an answer and sends it back. This operates exactly like it does in any straightforward application of [RFC3264].

2.2. Answerer Updates Session, no Contention

If the party that has been designated the "persistent answerer" wishes to update the session, it sends a message to the "persistent offerer" indicating that it wishes to update the session. For the purposes of the present document, we will refer to this new kind of message as a "solicitation" from this point forward. If the "persistent answerer" needs to add m-line sections to the session, it includes information in the solicitation to indicate the number and kind of m-line sections it requires. So, for example, if the "persistent answerer" wishes to add a new audio-visual element, it sends a solicitation to the "persistent offerer" indicating that it requires one new audio m-line section and one new video m-line section.

The "persistent offerer", upon receipt of this solicitation, generates a new offer for the session and sends it to the persistent answerer (assuming it has no offer outstanding; see Section 2.3). If additional m-line sections were requested, the offer will contain these new m-line sections, each containing "a=recvonly" attributes. The "persistent answerer", upon receipt of this offer, populates any new m-line sections with the information regarding the streams it wishes to establish, and otherwise updates the session in the answer according to any other changes it needs to perform.

2.3. Answerer Updates Session, with Contention

It is possible that the "persistent offerer" and "persistent answerer" attempt to update the session nearly simultaneously. This is the situation that would have previously resulted in an unfavorable "glare" condition. In such an interaction, the "persistent offerer" will send an offer, and then receive a solicitation prior to receiving the answer for the outstanding offer. From the "persistent answerer" perspective, the message sequence will be that it has send a solicitation, and then (as expected) received an offer. However, the offer may or may not satisfy the solicitation's request for additional m-line sections. We consider these two situations separately below.

2.4. Outstanding Offer Satisfies Solicitation

When the offerer detects the contention situation (i.e., receives a solicitation with an offer outstanding), it examines the outstanding offer to determine whether it satisfies the solicitation. For example: consider a session that contained two m-line sections (one audio, one video) prior to the most recent offer. The persistent offerer has sent a new offer that adds one m-line sections for each kind of media (i.e., contains two audio and two video). The offerer then receives a solicitation to add one audio and one video section to the session. Since the outstanding offer satisfies the

solicitation, the offerer discards the solicitation. The answerer then uses the new m-line sections in the offer to describe the new streams it wished to add.

Note that this kind of contention is most likely to arise due to a synchronizing event of some kind (e.g., the humans involved in the call decide, through their social interaction, that a new stream is warranted for the ongoing session, and both initiate actions to add such a stream).

2.5. Outstanding Offer Does Not Satisfy Solicitation

In the case that a solicitation arrives with an outstanding offer, and the outstanding offer does not contain enough additional m-line sections to satisfy the solicitation, then the offerer queues the solicitation for processing after it has received an answer. The answerer, for its own part, recognizes that the offer does not contain the requested additional sections, and so produces an answer equivalent to what it would have sent in the case that it had no outstanding solicitation. Then, when the following offer arrives -- which is guaranteed to contain the requested additional sections -- the answerer can take whatever actions it desires regarding the new media streams.

3. Future Work: Inter-Application Interoperation

The author anticipates that inter-application use of RTCWEB technologies will play an important role in the future of the web, and that the Session Initiation Protocol (SIP) will play a key role in such inter-operation. While a more formal definition of using the glareless add technique described in this document would require its own specification, doing so is a straightforward exercise:

o The use of a persistent offerer mechanism would be indicated in the first offer of the session (using a Supported header field plus a flag indicating activation of the mechanism). If the answer contains an acceptance of the mechanism, then it is a promise from the answerer that it will not, for the duration of this session, send an offer.

o The solicitation message is conveyed using an SIP INFO message (with a new INFO package). The body of such a message will contain a trivial document (probably XML) indicating how many new sections of each media type are required.

4. Acknowledgements

Thanks to Cullen Jennings and Eric Rescorla for early review of this document.

5. Security Considerations

The technique described in this document is not believed to introduce any changes in the fundamental security properties of RTCWEB clients.

6. IANA Considerations

This document makes no request of the IANA.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

7.2. Informative References

[webrtc-api]
Bergkvist, Burnett, Jennings, Narayanan, , "WebRTC 1.0: Real-time Communication Between Browsers", October 2011.

Available at http://dev.w3.org/2011/webrtc/editor/webrtc.html

Author's Address

Adam Roach
Mozilla
Dallas, TX
US

Phone: +1 650 903 0800 x863
Email: adam@nostrum.com