RTCWeb Working Group H. Alvestrand
Internet-Draft Google
Intended status: Standards Track U. Rauschenbach
Expires: July 24, 2016 Nokia Networks
January 21, 2016

WebRTC Gateways
draft-ietf-rtcweb-gateways-02

Abstract

This document describes interoperability considerations for a class of WebRTC-compatible endpoints called "WebRTC gateways", which interconnect between WebRTC endpoints and devices that are not WebRTC endpoints.

Requirements Language

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 RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The WebRTC model described in [I-D.ietf-rtcweb-overview] is focused on direct browser to browser communication as its primary use case. Nevertheless, it is clearly interesting to have WebRTC endpoints connect to other types of devices, including but not limited to SIP phones, legacy phones, CLUE-based teleconferencing systems, XMPP-based conferencing systems, and entirely proprietary devices or systems.

WebRTC gateways are middle boxes which enable the exchange of media streams between WebRTC endpoints on one side, and the other types of devices mentioned above on the other side. To a WebRTC endpoint, the gateway appears as a WebRTC-compatible endpoint.

This document describes the requirements that need to be placed on such gateways, both the requirements on WebRTC endpoints that can be relaxed and the additional requirements that need to be applied.

A WebRTC gateway appears as a WebRTC-compatible endpoint, and will thus not be conformant with all requirements for a WebRTC endpoint (it does not do everything a WebRTC endpoint does), but is able to interoperate with WebRTC endpoints.

NOTE IN DRAFT: There is still not a WG consensus called on whether this document is Informational or standards-track. If it becomes informational, the use of RFC 2119 language is used to call attention to features where non-conformance will render a gateway unable to interoperate with WebRTC-based endpoints.

1.1. Implications of the gateway environment

A gateway will be limited in the functionality it can offer by the system or class of devices it is gatewaying to. For instance, a gateway into the telephone system will not be able to relay data or video, no matter how much it is required. Therefore, a number of functions that are mandatory to support in WebRTC endpoints are not mandatory on gateways; the requirement on the gateway is that it is able to negotiate those features away correctly.

1.2. Signalling model

The WebRTC model is that signalling is outside the scope of the specification. This document does not change that.

Nevertheless, any practical gateway needs to deal with signalling. For that, this document assumes that the overall system consists of an application running in the WebRTC browser, possibly one or more signalling relays that mediate signalling and thereby enable communication between the application and the gateway, and the actual gateway that is responsible for handling the media flows.

The application, the signalling relays (if any) and the gateway together need to be able to:

- o adhere to the offer/answer semantics

- o deal with the description of configuration coming from the browser; this is specified in SDP format in the WebRTC browser API o generate the information that is needed by the browser to set up the session, and express that information in the form of SDP.

The shorthand notation "The gateway MUST/SHOULD/MAY support <SDP function xxx>" used below means that an application running in the Web browser, the signalling relays, and the gateway together MUST/SHOULD/MAY support this functionality; it is not a requirement that this happens at the media gateway itself.

2. WebRTC non-browser requirements that can be relaxed

WebRTC gateways are intended to communicate with WebRTC endpoints[I-D.ietf-rtcweb-overview]. Some features that typical WebRTC endpoints are required to support may be meaningless or unneccesary for WebRTC gateways; some such things are noted in this section. This lack of conformance means that a gateway is considered a WebRTC-compatible endpoint, not a WebRTC endpoint (unless a particular gateway claims to be a WebRTC endpoint, which it is of course allowed to do).

A WebRTC gateway which is expected to be deployed where it can be reached with a static IP address (as seen from the client) does not need to support full ICE; it therefore MAY implement ICE-Lite only.

ICE-Lite implementations do not send consent checks, so a gateway MAY choose not to send consent checks too, but MUST respond to consent checks it receives.

A gateway with a static IP address is expected to not need to hide its location, so it does not need to support functionality for operating only via a TURN server; instead it MAY choose to produce Host ICE candidates only.

If a gateway serves as a media relay into another RTP domain, it MAY choose to support only features available in that network. This means that it MAY choose to not support Bundle and any of the RTP/RTCP extensions related to it, RTCP-Mux, or Trickle Ice. However, the gateway MUST support DTLS-SRTP, since this is required for interworking with WebRTC endpoints.

Note that non-support of BUNDLE means that "bundle-only" tracks are not supported. This means that applications using an RTCBundlePolicy other than "max-compat" ([I-D.ietf-rtcweb-jsep] section 4.1.1) can only use one track of each media type.

If a gateway serves as a media relay into a network or to devices not implementing the WebRTC Datachannel, it MAY choose to not support the Datachannel.

3. Additional WebRTC gateway requirements

(nothing yet)

4. Considerations for SDP-using networks

Some networks that are gatewayed into, such as SIP networks, will also use SDP to represent the media configurations. Gateways will, however, need to inspect and probably modify the SDP passed between the SDP-using network and the WebRTC endpoints to achieve maximum interoperability.

Considerations include:

- If a correspondent does not offer the features WebRTC depends on, connections will not complete. The support for dtls-srtp, shown by the "fingerprint" attribute, is the most obvious example. The gateway is probably better off either ending such calls early or acting as a full B2BUA (as defined in [RFC3261]) with media gatewaying.

- If a correspondent makes an offer using features that are not required by JSEP, these may not be understood by the WebRTC implementation. The gateway may choose to strip out some such features.

o Certain ancient practices (such as using port 0 to place a media section on hold with the intent of resuming it later) are not conformant with the SDP offer/answer spec ([RFC3264] section 8.2). Since WebRTC implementations are expected to be SDP offer/answer conformant, such practices may need to be stripped out by the gateway

[NOTE IN DRAFT: This section may need expanding.]

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

A WebRTC gateway may operate in two security modes: Security-context termination and security-context relaying.

Relaying is only possible where signed and encrypted content can be passed through unchanged, and where keys can be exchanged directly between the endpoints.

When the gateway terminates the security context, it means that the WebRTC user has to place trust in the gateway to perform all verification of identity and protection of content in the realm on the other side of the gateway; there is no way the end-user can detect a man-in-the-middle attack, an identity spoofing attack or a recording done at the gateway. For many scenarios, this is not going to be seen as a problem, but needs to be considered when one decides to use a gatewayed service.

7. Acknowledgements

Several comments from Christer Holmberg and Andrew Hutton were included.

8. Change history

Changes from draft-alvestrand-rtcweb-gateways-00

o Aligned terminology with draft-rtcweb-overview-12

o Rewrote text on signaling to improve clarity

o Editorial nits

Changes from draft-alvestrand-rtcweb-gateways-01

o Aligned terminology with draft-rtcweb-overview-13 ("non-browser")

o Nits

Changes from draft-alvestrand-rtcweb-gateways-02

o Re-submitted as WG draft

o Addressed a comment from Andrew Hutton that deployment in open internet is an option, not a fact.

Changes from draft-ietf-rtcweb-gateways-00

o Added note about impllications of non-support of BUNDLE

o Added "Considerations for SDP-using networks" section

Changes from draft-ietf-rtcweb-gateways-01: None, this is a keepalive update.

9. References

9.1. Normative References

[I-D.ietf-rtcweb-jsep]
Uberti, J., Jennings, C., and E. Rescorla, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-09 (work in progress), March 2015.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-13 (work in progress), November 2014.

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

9.2. Informative References

[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.

Authors' Addresses

Harald Alvestrand
Google

Email: harald@alvestrand.no

Uwe Rauschenbach
Nokia Networks

Email: uwe.rauschenbach@nokia.com