Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 2, 2014

H. Alvestrand
Google

March 31, 2014

Transports for RTCWEB
draft-ietf-rtcweb-transports-03

Abstract

This document describes the data transport protocols used by RTCWEB, including the protocols used for interaction with intermediate boxes such as firewalls, relays and NAT boxes.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 2, 2014.

Table of Contents

1. Introduction

The IETF RTCWEB effort, part of the WebRTC effort carried out in cooperation between the IETF and the W3C, is aimed at specifying a protocol suite that is useful for real time multimedia exchange between browsers.

The overall effort is described in the RTCWEB overview document, [I-D.ietf-rtcweb-overview]. This document focuses on the data transport protocols that are used by conforming implementations.

This protocol suite is designed for WebRTC, and intends to satisfy the security considerations described in the WebRTC security documents, [I-D.ietf-rtcweb-security] and [I-D.ietf-rtcweb-security-arch].

2. 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].

3. Transport and Middlebox specification

3.1. System-provided interfaces

The protocol specifications used here assume that the following protocols are available to the implementations of the RTCWEB protocols:

- UDP. This is the protocol assumed by most protocol elements described.
- TCP. This is used for HTTP/WebSockets, as well as for TURN/SSL and ICE-TCP.

For both protocols, IPv4 and IPv6 support is assumed.

For UDP, this specification assumes the ability to set the DSCP code point of the sockets opened on a per-packet basis, in order to achieve the prioritizations described in [I-D.dhesikan-tsvwg-rtcweb-qos] (see Section 3.4) when multiple media types are multiplexed. It does not assume that the DSCP codepoints will be honored, and does assume that they may be zeroed or changed, since this is a local configuration issue.

Platforms that do not give access to these interfaces will not be able to support a conforming RTCWEB implementation.

This specification does not assume that the implementation will have access to ICMP or raw IP.

3.2. Ability to use IPv4 and IPv6

Web applications running on top of the RTCWEB implementation MUST be able to utilize both IPv4 and IPv6 where available - that is, when two peers have only IPv4 connectivty to each other, or they have only IPv6 connectivity to each other, applications running on top of the RTCWEB implementation MUST be able to communicate.

When TURN is used, and the TURN server has IPv4 or IPv6 connectivity to the peer or its TURN server, candidates of the appropriate types MUST be supported. The "Happy Eyeballs" specification for ICE [I-D.reddy-mmusic-ice-happy-eyeballs] SHOULD be supported.

3.3. Usage of temporary IPv6 addresses

The IPv6 default address selection specification [RFC6724] specifies that temporary addresses [RFC4941] are to be preferred over permanent addresses. This is a change from the rules specified by [RFC3484]. For applications that select a single address, this is usually done by the IPV6_PREFER_SRC_TMP specified in [RFC5014]. However, this rule is not completely obvious in the ICE scope. This is therefore clarified as follows:

When a client gathers all IPv6 addresses on a host, and both temporary addresses and permanent addresses of the same scope are present, the client SHOULD discard the permanent addresses before forming pairs. This is consistent with the default policy described in [RFC6724].

3.4. Usage of Quality of Service - DSCP and Multiplexing

WebRTC implementations SHOULD attempt to set QoS on the packets sent, according to the guidelines in [I-D.dhesikan-tsvwg-rtcweb-qos]. It is appropriate to depart from this recommendation when running on platforms where QoS marking is not implemented.

There exist a number of schemes for achieving quality of service that do not depend solely on DSCP code points. Some of these schemes depend on classifying the traffic into flows based on 5-tuple (source address, source port, protocol, destination address, destination port) or 6-tuple (same as above + DSCP code point). Under differing conditions, it may therefore make sense for a sending application to

choose any of the configurations:

- Each media stream carried on its own 5-tuple
- Media streams grouped by media type into 5-tuples (such as carrying all audio on one 5-tuple)
- All media sent over a single 5-tuple, with or without differentiation into 6-tuples based on DSCP code points

In each of the configurations mentioned, data channels may be carried in its own 5-tuple, or multiplexed together with one of the media flows.

More complex configurations, such as sending a high priority video stream on one 5-tuple and sending all other video streams multiplexed together over another 5-tuple, can also be envisioned.

A sending implementation MUST be able to multiplex all media and data on a single 5-tuple (fully bundled), MUST be able to send each media stream and data on their own 5-tuple (fully unbundled), and MAY choose to support other configurations.

NOTE IN DRAFT: is there a need to place the "group by media type, with data multiplexed on the video" as a MUST or SHOULD configuration?

A receiving implementation MUST be able to receive media and data in all these configurations.

3.5. Middle box related functions

The primary mechanism to deal with middle boxes is ICE, which is an appropriate way to deal with NAT boxes and firewalls that accept traffic from the inside, but only from the outside if it's in response to inside traffic (simple stateful firewalls).

ICE [RFC5245] MUST be supported. The implementation MUST be a full ICE implementation, not ICE-Lite.

In order to deal with situations where both parties are behind NATs which perform endpoint-dependent mapping (as defined in [RFC5128] section 2.4), TURN [RFC5766] MUST be supported.

Configuration of STUN and TURN servers, both from browser configuration and from an applicaiton, MUST be supported.

In order to deal with firewalls that block all UDP traffic, TURN

using TCP between the client and the server MUST be supported, and TURN using TLS over TCP between the client and the server MUST be supported. See [RFC5766] section 2.1 for details.

In order to deal with situations where one party is on an IPv4 network and the other party is on an IPv6 network, TURN extensions for IPv6 [RFC6156] MUST be supported.

TURN TCP candidates [RFC6062] MAY be supported.

However, such candidates are not seen as providing any significant benefit. First, use of TURN TCP would only be relevant in cases which both peers are required to use TCP to establish a PeerConnection. Secondly, that use case is anyway supported by both sides establishing UDP relay candidates using TURN over TCP to connect to the relay server. Thirdly, using TCP only between the endpoint and its relay may result in less issues with TCP in regards to real-time constraints, e.g. due to head of line blocking.

ICE-TCP candidates [RFC6544] MAY be supported; this may allow applications to communicate to peers with public IP addresses across UDP-blocking firewalls without using a TURN server.

If TCP connections are used, RTP framing according to [RFC4571] MUST be used, both for the RTP packets and for the DTLS packets used to carry data channels.

The ALTERNATE-SERVER mechanism specified in [RFC5389] (STUN) section 11 (300 Try Alternate) MUST be supported.

Further discussion of the interaction of RTCWEB with firewalls is contained in [I-D.hutton-rtcweb-nat-firewall-considerations]. This document makes no requirements on interacting with HTTP proxies or HTTP proxy configuration methods.

NOTE IN DRAFT: This may be added.

3.6. Transport protocols implemented

For transport of media, secure RTP is used. The details of the profile of RTP used are described in "RTP Usage" [I-D.ietf-rtcweb-rtp-usage].

For data transport over the RTCWEB data channel [I-D.ietf-rtcweb-data-channel], RTCWEB implementations MUST support SCTP over DTLS over ICE. This encapsulation is specified in [I-D.ietf-tsvwg-sctp-dtls-encaps]. Negotiation of this transport in SDP is defined in [I-D.ietf-mmusic-sctp-sdp]. The SCTP extension for

NDATA, [I-D.ietf-tsvwg-sctp-ndata], MUST be supported.

The setup protocol for RTCWEB data channels is described in [I-D.jesup-rtcweb-data-protocol].

RTCWEB implementations MUST support multiplexing of DTLS and RTP over the same port pair, as described in the DTLS_SRTP specification [RFC5764], section 5.1.2. All application layer protocol payloads over this DTLS connection are SCTP packets.

4. IANA Considerations

This document makes no request of IANA.

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

5. Security Considerations

Security considerations are enumerated in [I-D.ietf-rtcweb-security].

6. Acknowledgements

This document is based on earlier versions embedded in [I-D.ietf-rtcweb-overview], which were the results of contributions from many RTCWEB WG members.

Special thanks for reviews of earlier versions of this draft go to Magnus Westerlund, Markus Isomaki and Dan Wing; the contributions from Andrew Hutton also deserve special mention.

7. References

7.1. Normative References

[I-D.dhesikan-tsvwg-rtcweb-qos]
Dhesikan, S., Druta, D., Jones, P., and J. Polk, "DSCP and other packet markings for RTCWeb QoS", draft-dhesikan-tsvwg-rtcweb-qos-06 (work in progress), March 2014.

[I-D.ietf-mmusic-sctp-sdp]
Loreto, S. and G. Camarillo, "Stream Control Transmission Protocol (SCTP)-Based Media Transport in the Session

Description Protocol (SDP)", draft-ietf-mmusic-sctp-sdp-06 (work in progress), February 2014.

[I-D.ietf-rtcweb-data-channel]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-07 (work in progress), February 2014.

[I-D.ietf-rtcweb-rtp-usage]
Perkins, C., Westerlund, M., and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", draft-ietf-rtcweb-rtp-usage-12 (work in progress), February 2014.

[I-D.ietf-rtcweb-security]
Rescorla, E., "Security Considerations for WebRTC", draft-ietf-rtcweb-security-06 (work in progress), January 2014.

[I-D.ietf-rtcweb-security-arch]
Rescorla, E., "WebRTC Security Architecture", draft-ietf-rtcweb-security-arch-09 (work in progress), February 2014.

[I-D.ietf-tsvwg-sctp-dtls-encaps]
Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "DTLS Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-dtls-encaps-03 (work in progress), February 2014.

[I-D.ietf-tsvwg-sctp-ndata]
Stewart, R., Tuexen, M., Loreto, S., and R. Seggelmann, "A New Data Chunk for Stream Control Transmission Protocol", draft-ietf-tsvwg-sctp-ndata-00 (work in progress), February 2014.

[I-D.reddy-mmusic-ice-happy-eyeballs]
Reddy, T., Patil, P., and P. Martinsen, "Happy Eyeballs Extension for ICE", draft-reddy-mmusic-ice-happy-eyeballs-06 (work in progress), February 2014.

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

[RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, July 2006.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC6062] Perreault, S. and J. Rosenberg, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations", RFC 6062, November 2010.

[RFC6156] Camarillo, G., Novo, O., and S. Perreault, "Traversal Using Relays around NAT (TURN) Extension for IPv6", RFC 6156, April 2011.

[RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, March 2012.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

7.2. Informative References

[I-D.hutton-rtcweb-nat-firewall-considerations] Stach, T., Hutton, A., and J. Uberti, "RTCWEB Considerations for NATs, Firewalls and HTTP proxies", draft-hutton-rtcweb-nat-firewall-considerations-03 (work in progress), January 2014.

[I-D.ietf-rtcweb-overview] Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-09 (work

in progress), February 2014.

[I-D.jesup-rtcweb-data-protocol]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel Protocol", draft-jesup-rtcweb-data-protocol-04 (work in progress), February 2013.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, September 2007.

[RFC5128] Srisuresh, P., Ford, B., and D. Kegel, "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)", RFC 5128, March 2008.

Appendix A. Change log

A.1. Changes from -00 to -01

- Clarified DSCP requirements, with reference to -qos-
- Clarified "symmetric NAT" -> "NATs which perform endpoint-dependent mapping"
- Made support of TURN over TCP mandatory
- Made support of TURN over TLS a MAY, and added open question
- Added an informative reference to -firewalls-
- Called out that we don't make requirements on HTTP proxy interaction (yet

A.2. Changes from -01 to -02

- Required support for 300 Alternate Server from STUN.
- Separated the ICE-TCP candidate requirement from the TURN-TCP requirement.
- Added new sections on using QoS functions, and on multiplexing considerations.

- Removed all mention of RTP profiles. Those are the business of the RTP usage draft, not this one.
- Required support for TURN IPv6 extensions.
- Removed reference to the TURN URI scheme, as it was unnecessary.
- Made an explicit statement that multiplexing (or not) is an application matter.

.

A.3. Changes from -02 to -03

- Added required support for draft-ietf-tsvwg-sctp-ndata
- Removed discussion of multiplexing, since this is present in rtp-usage.
- Added RFC 4571 reference for framing RTP packets over TCP.
- Downgraded TURN TCP candidates from SHOULD to MAY, and added more language discussing TCP usage.
- Added language on IPv6 temporary addresses.
- Added language describing multiplexing choices.
- Added a separate section detailing what it means when we say that an RTCWEB implementation MUST support both IPv4 and IPv6.

Author's Address

Harald Alvestrand
Google

Email: harald@alvestrand.no