

RTCWEB
Internet-Draft
Intended status: Informational
Expires: January 5, 2012

J. Rosenberg
Skype
C. Jennings
Cisco
J. Peterson
Neustar
M. Kaufman
Skype
E. Rescorla
RTFM
T. Terriberry
Mozilla
July 4, 2011

Multiplexing of Real-Time Transport Protocol (RTP) Traffic for Browser
based Real-Time Communications (RTC)
draft-rosenberg-rtcweb-rtpmux-00

Abstract

This document argues that multiplexing of voice and video traffic over a single RTP session should be specified as the baseline mode of operation for multimedia traffic in RTC web.

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 January 5, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal

Internet-Draft

RTP Mux

July 2011

Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	RTP Muxing with SSRC	3
3.	Arguments in Favor of Multiplexing	4
3.1.	NAT Resource Preservation	4
3.2.	Improved Failure Modes	5
3.3.	Setup Time	5
3.4.	Complexity	5
4.	Responding to draft-perkins-rtcweb-rtp-usage	5
4.1.	Requires Additional Signaling	6
4.2.	QoS and Traffic Engineering	6
4.3.	Scalability	7
4.4.	RTP Retransmission	7
4.5.	Forward Error Correction	8
4.6.	RTCP Issues	8
5.	Arguing Against a Shim	9
6.	Conclusion	10
7.	Informative References	10
	Authors' Addresses	11

1. Introduction

The RTCweb working group is chartered to specify a framework and protocols for enabling real-time communications services within a browser, without the need for plugins [[I-D.rosenberg-rtcweb-framework](#)]. It is envisioned that this will enable many use cases [[I-D.ietf-rtcweb-use-cases-and-requirements](#)], the most basic of which is a video call between two users on the web.

In order to enable this functionality, the specifications produced by the IETF will mandate a specific set of protocols that must be implemented within the browser. It is anticipated that these protocols will include the Real-Time Transport Protocol [[RFC3550](#)], and either in full or in part, Interactive Connectivity Establishment (ICE) [[RFC5245](#)].

The usage of RTP raises the question of multiplexing - whether or not RTCP and RTP should run on the same port, and furthermore, whether or not voice, video, and possibly data, should also run on the same port. To provide guidance on this, Perkins et. al. produced [[I-D.perkins-rtcweb-rtp-usage](#)], which recommends that voice and video utilize different RTP sessions, and thus different UDP ports.

This document argues against this conclusion, and advocates that a single transport session (i.e., a single UDP port) is used to carry voice and video traffic, using the SSRC for demux.

2. RTP Muxing with SSRC

This document recommends that all of the associated media content of the call - the voice, video, and RTCP traffic for both the voice and video sessions, utilize a single transport session (i.e., single UDP port). In cases where there are multiple video streams (for example, screen sharing), the single transport session would carry all of the video. Furthermore, that demultiplexing voice and video traffic is

done by assigning a different SSRC to each. This recommendation applies to the case of a single unicast communications session between a pair of endpoints (e.g., this document does not consider the case of running a multi-user service like a gateway).

To enable multiplexing, we propose that the 32-bit SSRC value in the RTP header be broken up into the following sub-fields:

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           Magic Cookie           |Type |      StreamID      |x|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---

```

SSRC Field

The Magic Cookie is two bytes, with a value of 0xf7b3. It is meant to facilitate DPI applications which can use its value to - with high confidence - determine that this RTP packet uses the encoding format defined here. The type is a 3 bit value, corresponding to the top-level MIME type of the media (mapping table TBD). It too is meant to facilitate DPI applications which want to separate voice and video. The streamID is a 12 bit field which represents the unique ID for this stream. It is signaled between participants out of band. The final bit, 'x' is set to zero and is reserved for future usage.

[3.](#) Arguments in Favor of Multiplexing

This section outlines several arguments in favor of multiplexing.

[3.1.](#) NAT Resource Preservation

Today's Internet is full of Network Address Translators (NAT), a situation which is likely to get worse as IPv4 address exhaustion continues. When NAT is in use, the constraint on the number of endpoints behind the NAT is based on the number of parallel transport sessions that need to be supported. If, for example, a NAT has a single external IP address, it can support 64k UDP sessions while having an endpoint-independent mapping behavior [[RFC4787](#)]. Thus, in the presence of NAT, parallel transport sessions becomes the scarce resource.

If rtcweb specifies that audio and video run on a separate port, this will double the number of transport session resources consumed in intervening NATs. While the usage of port as an application layer demux point made sense when RTP was designed back in 1992 (the year the first RTP draft was published), the Internet has changed substantially since then. Continuing to perpetuate this design optimizes preservation of legacy against protection of resources in the modern Internet. We feel that this optimizes in the wrong direction.

Given that we anticipate widespread usage of rtcweb, this design choice may create a non-trivial load on the transport session capacity of the Internet at large. Real-time video communications on the Internet has seen huge growth in recent years. For Skype, approximately 40% of its Skype-to-Skype calls are video based. A recent report by Sandvine reports that Skype alone is the third largest source of upload traffic on the Internet as a whole, largely attributed to Skype video calling. <http://www.sandvine.com/downloads/documents/05-17-2011_phenomena/

Sandvine%20Global%20Internet%20Phenomena%20Spotlight%20-%20Netflix%20Rising.pdf>. The conclusion from this is that the costs of a separate voice and video port cannot be ignored.

Simply put, the usage of transport ports for application demultiplexing should be considered harmful for the Internet.

[3.2.](#) Improved Failure Modes

The usage of separate transport sessions for the audio, video or other content of the call introduces a variety of partial failure modes. The transport session for one type of media might get established; but a NAT capacity problem might cause the transport session for another type of media to fail. Usage of a single transport session means that the conversation succeeds or fails atomically. We consider this a feature.

[3.3.](#) Setup Time

The rtcweb group is considering the usage of ICE to create p2p sessions. ICE provides firewall and NAT traversal in addition to providing a handshake necessary to assure mutual consent for

communications.

Unfortunately, ICE requires time to perform its setup operations. This time grows in proportion to the number of transport sessions which must be opened in order to support the call. By using a different port for video traffic, call setup times will increase. The precise amount of this increase depends on the type of NAT and varies depending on packet loss. However, in a simple, ideal case of no packet loss and direct connectivity between endpoints, this value is XXX [[fill in]].

[3.4.](#) Complexity

ICE is not a simple protocol. One of its significant complexities is its requirement to support calls for multiple media streams, each of which runs on a separate port, and multiple components for each stream (e.g., RTCP). If the concept of streams and components were eliminated, ICE would be a simpler protocol.

If, within rtcweb, a single transport connection was utilized, browsers could implement a simplified version of the ICE protocol.

[4.](#) Responding to [draft-perkins-rtcweb-rtp-usage](#)

[I-D.perkins-rtcweb-rtp-usage] outlines several arguments for

Rosenberg, et al. Expires January 5, 2012 [Page 5]

Internet-Draft RTP Mux July 2011

continuing to use a separate port for audio and video. In this section, we respond to those arguments.

[4.1.](#) Requires Additional Signaling

[I-D.perkins-rtcweb-rtp-usage] argues that multiplexing of voice and video on the same RTP session would require a demux point to be specified (for example, the SSRC), and require additional signaling to be specified to accomplish this.

Firstly, this conclusion is only partly true. For communications sessions between rtcweb users within the same domain, no signaling specifications are required. This is true in general with rtcweb; one of its benefits is that it does not require standardized signaling.

Secondly, it is not yet clear that rtcweb will be able to interoperate with existing VoIP endpoints without a media intermediary to terminate ICE traffic. It is our position that interoperability without media intermediary only be provided for basic voice services, and even then, only when RTCP is supported. In the case of basic voice endpoints, where there is no video, RTP multiplexing of voice and video is irrelevant, and thus no signaling complexity is introduced.

Thirdly, the primary place where there will be a need for signaling enhancements is for inter-domain calling between rtcweb endpoints in different domains. In such a case, an SDP extension is required, and one can be specified. It is trivial to do so.

Finally, this document does recommend that it be possible to utilize a separate transport session for voice and for video, and that, in the worst case, this mode can be used for calls between an rtcweb endpoint and a legacy endpoint.

[4.2.](#) QoS and Traffic Engineering

[I-D.perkins-rtcweb-rtp-usage] argues that multiplexing of voice and video on the same RTP session would mean that it would not be possible to apply QoS techniques separately for voice and video which rely on the 5-tuple.

Firstly, the public Internet lacks any QoS mechanism, so this argument is moot on the public Internet.

Secondly, private enterprise networks which do provide QoS most often use diffserv. Diffserv is compatible with utilization of a common port for voice and video traffic. Typically, different DSCPs are

used for voice and video (Cisco recommends EF for audio and AF41 for video in enterprise telephony deployments), and this practice is compatible with usage of the same port - each packet would be marked appropriately. It is also possible to use the same DSCP for voice and video.

Carrier networks, such as mobile operator networks, typically provide QoS through traffic engineering, using a combination of MPLS tunnels

and diffserv markings. MPLS tunnels do use 5-tuples as classifiers to determine which traffic to put in what kind of tunnel. If there is a need for using separate MPLS tunnels for voice and video, the DSCP codepoint itself can be used as a differentiator.

It is true that it would not be possible to utilize RSVP to separately establish QoS treatment for the voice and the video traffic. However, there is very little real deployment of RSVP. None within the public Internet and relatively little within corporate networks. As such, this argument is mostly theoretical.

Finally, DPI is used within some operator networks to perform traffic classification. It would always be possible to use DPI to assign different treatment to voice and video traffic.

[4.3.](#) Scalability

[I-D.perkins-rtcweb-rtp-usage] argues that multiplexing of voice and video on the same RTP session would mean that layered coding using multicast for each layer would not be possible.

Firstly, most layered coding today uses unicast and a switch or mixer of some sort to discard layers. That architecture is completely compatible with the usage of a single transport session for voice and video. The limitation applies only to the use of IP multicast for real-time communications. The usage of multicast on the Internet has substantially diminished over time. There is some usage today in private networks but primarily for streaming media distribution. The usage for real-time communications is quite rare. As such, we find this to be a theoretical corner case.

[4.4.](#) RTP Retransmission

[I-D.perkins-rtcweb-rtp-usage] argues that multiplexing of voice and video on the same RTP session would not be interoperable with endpoints doing RTP retransmission per [\[RFC4588\]](#).

As pointed out above, interoperability with existing endpoints without the usage of a media intermediary is not a given at this point, and we argue it should only be supported for the common case -

a basic, voice-only RTP-capable endpoint. There is, to our

knowledge, relatively little deployment of [RFC4588](#), at least for real-time communications. It is certainly not a common feature in basic RTP endpoints and never a baseline requirement for interoperability. Consequently, if there is a need to interoperate with an endpoint supporting [RFC4588](#), and it is desired to avoid a media intermediary, [RFC4588](#) can just be turned off for the session.

As such, we find the interoperability argument here not compelling.

[4.5.](#) Forward Error Correction

[I-D.perkins-rtcweb-rtp-usage] argues that multiplexing of voice and video on the same RTP session will limit the applicability of FEC [[RFC5109](#)] to when the RTP packets are half of the path MTU.

There are two cases to consider - interoperability with existing endpoints and usage for calls between rtcweb endpoints.

For interoperability with existing endpoints, we argue the same thing here as for retransmits. FEC is not commonly used in legacy voice endpoints, and if it is supported, is never a required feature. Consequently, if present, its usage can be disabled when interoperating with an rtcweb endpoint. If FEC is included as part of the rtcweb specifications, the lower bandwidth of voice means that FEC packets could be sent on the same port, using [[RFC2198](#)], without approaching the path MTU.

For communications between rtcweb endpoints, this is only an issue if FEC is included as part of the rtcweb specification. If the group decides to do that (there is some value for real-time video), it should define a mechanism which allows for FEC packets to be sent using a separate SSRC.

[4.6.](#) RTCP Issues

[I-D.perkins-rtcweb-rtp-usage] argues that multiplexing of voice and video on the same RTP session will introduce complications in the usage of RTCP, primarily when considering RTCP extensions.

It is our belief that normal RTCP operation as defined in the RTCP specification will work fine with multiplexed voice and video traffic. SRs and RRs are already generated per SSRC to handle multiple senders, and RTCP in general supports feedback for multiple SSRC within a session. These mechanisms work as defined when each SSRC happens to represent a different media stream instead of a different user.

The only complication that arises is for RTCP extensions which are defined to be media dependent. [I-D.perkins-rtcweb-rtp-usage] points out, as an example, the usage of RTCP extended report blocks (XR) [RFC3611]. However, XR works fine in conjunction with multiplexing of voice and video within the same port. Each of the seven report blocks defined in [RFC3611] include the SSRC of the source as part of the block, and thus will work. [I-D.perkins-rtcweb-rtp-usage] indicates that "SSRC purpose tagging needs not only to be on the media side, but also on the RTCP reporting". However, we do not believe this to be accurate. Since the XR blocks report the SSRC source already, the specifications provide all that is needed. The XR report is merely included when it is relevant.

Furthermore, the discussion around XR assumes that we need to support them for interoperability with existing VoIP endpoints, or we are utilizing it for rtcweb itself. As with FEC and retransmissions, in the case of interoperability, if there is an issue, XR can simply be disabled in these cases. [RFC3611] does specify that XR can be sent without prior signaling. In the worst case XR are received by an rtcweb endpoint which are then discarded. In terms of usage of RTCP XR for communications between rtcweb endpoints, we would argue that a much more flexible solution would be to provide Javascript APIs which allow the application to have access to the same data used to generate the XR, and then the application itself can use this data as it sees fit, including sending it back to the sender through some kind of application data packet.

5. Arguing Against a Shim

It has been proposed on the mailing list that an alternative approach for multiplexing on the same port would be to specify a new multiplexing protocol that has a small shim, which could then be used to separate voice and video traffic as a layer between UDP and RTP. Such a shim could then also be used to enable non-RTP data traffic as well.

We believe that such a shim would be a mistake, for the same reason that shims have been avoided in the multiplexing of RTCP, STUN, and DTLS on the same port as RTP:

- o The shim would break interoperability with a great deal of existing network inspection gear – firewalls, packet sniffers, traffic analyzers, and so on – which know how to extract, parse, and process RTP packets.

- o The shim would add complexity through yet another layer of multiplexing.

- o The shim would increase packet overhead further.
- o A shim is a mistake which cannot be undone later. If multiplexing on a single port truly causes interoperability issues, clients can fall back to using multiple ports, possibly even in the preponderance of cases. However, once a shim is inserted, interoperability will always require an intermediary to strip it out, forever.

[6.](#) Conclusion

In conclusion, we feel that benefits of multiplexing of voice and video on a single RTP session (and thus single transport connection), outweigh the drawbacks. The primary benefit is the impact on NAT capacity, which is becoming an important issue in the modern Internet. Furthermore, the unique nature of backwards compatibility for rtcweb lessens many of the interoperability concerns, and the traditional arguments around multicast and RSVP are simply no longer relevant and those technologies have faded from use.

[7.](#) Informative References

[I-D.perkins-rtcweb-rtp-usage]

Perkins, C., Westerlund, M., and J. Ott, "RTP Requirements for RTC-Web", [draft-perkins-rtcweb-rtp-usage-01](#) (work in progress), June 2011.

[I-D.rosenberg-rtcweb-framework]

Rosenberg, J., Kaufman, M., Hiie, M., and F. Audet, "An Architectural Framework for Browser based Real-Time Communications (RTC)", [draft-rosenberg-rtcweb-framework-00](#) (work in progress), February 2011.

[I-D.ietf-rtcweb-use-cases-and-requirements]

Holmberg, C., Hakansson, S., and G. Eriksson, "Web Real-Time Communication Use-cases and Requirements", [draft-ietf-rtcweb-use-cases-and-requirements-01](#) (work in

progress), July 2011.

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time

Rosenberg, et al.

Expires January 5, 2012

[Page 10]

Internet-Draft

RTP Mux

July 2011

Applications", STD 64, [RFC 3550](#), July 2003.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", [BCP 127](#), [RFC 4787](#), January 2007.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", [RFC 4588](#), July 2006.

[RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", [RFC 5109](#), December 2007.

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", [RFC 3611](#), November 2003.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", [RFC 2198](#), September 1997.

Authors' Addresses

Jonathan Rosenberg
Skype

Email: jdrosen@skype.net

URI: <http://www.jdrosen.net>

Cullen Jennings
Cisco

Email: fluffy@cisco.com

Jon Peterson
Neustar

Email: jon.peterson@neustar.biz

Rosenberg, et al. Expires January 5, 2012 [Page 11]

Internet-Draft RTP Mux July 2011

Matthew Kaufman
Skype

Email: matthew.kaufman@skype.net

Eric Rescorla
RTFM

Email: ekr@rtfm.com

Tim Terriberry
Mozilla

Email: tterriberry@mozilla.com

