

RTCWEB  
Internet-Draft  
Intended status: Standards Track  
Expires: April 26, 2012

J. Lennox  
Vidyo  
J. Rosenberg  
Skype  
October 24, 2011

Multiplexing Multiple Media Types In a Single Real-Time Transport  
Protocol (RTP) Session  
draft-lennox-rtcweb-rtp-media-type-mux-00

## Abstract

This document describes mechanisms and recommended practice for transmitting media streams of multiple media types (e.g., audio and video) over a single Real-Time Transport Protocol (RTP) session, primarily for the use of Real-Time Communication for the Web (rtcweb).

## 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 April 26, 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 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
3.	Transmitting multiple types of media in a single RTP session . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	Optimizations . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Backward compatibility . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Signaling . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Protocols with SSRC semantics . . . . .	<a href="#">8</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">8.</a>	IANA Considerations . . . . .	<a href="#">9</a>
<a href="#">9.</a>	References . . . . .	<a href="#">9</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">9</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">9</a>
	Authors' Addresses . . . . .	<a href="#">10</a>

## 1. Introduction

Classically, multimedia sessions using the Real-Time Transport Protocol (RTP) [[RFC3550](#)] have transported different media types (most commonly, audio and video) in different RTP sessions, each with its own transport flow. At the time RTP was designed, this was a reasonable design decision, reducing system variability and adding flexibility ([[RFC3550](#)] discusses the motivation for this design decision in [section 5.2](#)).

However, the de facto architecture of the Internet has changed substantially since RTP was originally designed, nearly twenty years ago. In particular, Network Address Translators (NATs) and firewalls are now ubiquitous, and IPv4 address space scarcity is becoming more severe. As a consequence, the network resources used up by an application, and its probability of failure, are directly proportional to the number of distinct transport flows it uses.

Furthermore, applications have developed mechanisms (notably Interactive Connectivity Establishment (ICE) [[RFC5245](#)]) to traverse NATs and firewalls. The time such mechanisms need to perform the traversal process is proportional to the number of distinct transport flows in use.

As a result, in the modern Internet, it is advisable and useful to revisit the transport-layer separation of media in a multimedia session. Fortunately, the architecture of RTP allows this to be done in a straightforward and natural way: by placing multiple sources of different media types in the same RTP session.

Since this is architecturally somewhat different from existing RTP deployments, however, this decision has some consequences that may be non-obvious. Furthermore, it is somewhat complex to negotiate such flows in signaling protocols that assumed the older architecture, most notably the Session Description Protocol (SDP) [[RFC4566](#)]. The rest of this document discusses these issues.

## 2. Terminology

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](#)] and indicate requirement levels for compliant implementations.

## 3. Transmitting multiple types of media in a single RTP session

RTP [[RFC3550](#)] supports the notion of multiple sources within a session. Historically, this was typically used for distinct users within a group to send media of the same type. Each source has its own synchronization source (SSRC) value and has a distinct sequence number and timestamp space. This document specifies that this same mechanism is used to allow sources of multiple media types in the same RTP session, even if they come from the same user. For example, in a call containing audio and video between two users, each sending a single audio and a single video source, there would be a single RTP session containing two sources (one audio, one video) from each user, for a total of four sources (and thus four SSRC values) within the RTP session.

Transmitting multiple types of media in a single RTP [[RFC3550](#)] session is done using the same RTP mechanisms as are used to transmit multiple sources of the same media type on a session. Notably:

- o Each stream (of every media type) is a distinct source (distinct stream of consecutive packets to be sent to a decoder) and is given a distinct synchronization source ID (SSRC), and has its own distinct timestamp and sequence number space.
- o Every media type (full media type and subtype, e.g. video/h264 or audio/pcmu) has a distinct payload type value. The same payload type value mappings apply across all sources in the session.
- o RTP SSRCs, initial sequence numbers, and initial timestamps are chosen at random, independently for each source (of each media type).
- o RTCP bandwidth is five percent of the total RTP session bandwidth.

- o RTP session bandwidth and RTCP bandwidth are divided among all the sources in the session.
- o RTCP sender report (SR) or receiver report (RR) packets, and source description (SDS) packets, are sent periodically for every source in the session.

In other words, no special RTP mechanisms are specifically needed for senders of multiplexed media. The only constraint is that senders sources MUST NOT change the top-level media type (e.g. audio or video) of a given source. (It remains valid to change a source's subtype, e.g. switching between audio/pcmu and audio/g729.)

For a receiver, the primary complexity of multiplexing is knowing how to process a received source. Without multiplexing, all sources in an RTP session can (in theory) be processed the same manner; e.g., all audio sources can be fed to an audio mixer, and all video sources displayed on a screen. With multiplexing, however, receivers must apply additional knowledge.

If the streams being multiplexed are simply audio and video, this processing can decision can be made based simply on a source's payload type. For more complex situations (for example, simultaneous live-video and shared-application sources, both sent as video), signaling-level descriptions of sources would be needed, using a mechanism such as SDP Source Descriptions [[RFC5576](#)].

Additionally, due to the large difference in typical bitrate between different media (video can easily use a bit rate an order of magnitude or more larger than audio), some complications arise with RTCP timing. Because RTCP bandwidth is shared evenly among all sources in a session, the RTCP for an audio source can end up being sent significantly more frequently than it would in a non-multiplexed session. (The RTCP for video will, correspondingly, be sent slightly less frequently; this is not nearly as serious an issue.)

For RTP sessions that use RTP's recommended minimum fixed timing interval of 5 seconds, this problem is not likely to arise, as most sessions' bandwidth is not so low that RTCP timing exceeds this limit. The RTP/AVP [[RFC3551](#)] or RTP/SAVP [[RFC3711](#)] profiles use this minimum interval by default, and do not have a mechanism in SDP to negotiate an alternate interval.

For sessions using the RTP/AVPF [[RFC4585](#)] and RTP/SAVPF [[RFC5124](#)] profiles, however, endpoints SHOULD set the minimum RTCP regular reporting interval trr-int to 5000 (5 seconds), unless they explicitly need it to be lower. This minimizes the excessive RTCP bandwidth consumption, as well as aiding compatibility with AVP endpoints. Since this value only affects regular RTCP reports, not RTCP feedback, this does not prevent AVPF feedback messages from being sent as needed.

### [3.1.](#) Optimizations

For multiple sources in the same session, several optimizations are possible. (Most of these optimizations also apply to multiple sources of the same type in a session.) In all cases, endpoints MUST be prepared for their peers to be using these optimizations.

An endpoint sending multiple sources MAY, as needed, reallocate media bandwidth among the RTP sources it is sending. This includes adding or removing sources as more or less bandwidth becomes available.

An endpoint MAY choose to send multiple sources' RTCP messages in a single compound RTCP packet (though such compound packets SHOULD NOT exceed the path MTU, if avoidable and if it is known). This will reduce the average compound RTCP packet size, and thus increase the frequency with which RTCP messages can be sent. Regular (non-

feedback) RTCP compound packets MUST still begin with an SR or RR packet, but otherwise may contain RTCP packets in any order. Receivers MUST be prepared to receive such compound packets.

An endpoint SHOULD NOT send reception reports from one of its own sources about another one ("cross-reports"). Such reports are useless (they would always indicate zero loss and jitter) and use up bandwidth that could more profitably be used to send information about remote sources. Endpoints receiving reception reports MUST be prepared that their peers might not be sending reception reports about their own sources. (A naive RTCP monitor might think that there is a network disconnection between these sources; however, architecturally it is very unclear if such monitors actually exist, or would care about a disconnection of this sort.)

Similarly, an endpoint sending multiple sources SHOULD NOT send reception reports about a remote source from more than one of its local sources. Instead, it SHOULD pick one of its local sources as the "reporting" source for each remote source, which sends full report blocks; all its other sources SHOULD be treated as if they were disconnected, and never saw that remote source. An endpoint MAY choose different local sources as the reporting source for different remote sources (for example, it could choose to send reports about remote audio sources from its local audio source, and reports about remote video sources from its local video source), or it MAY choose a single local source for all its reports. If the reporting source leaves the session (sends BYE), another reporting source MUST be chosen. This "reporting" source SHOULD also be the source for any AVPF feedback messages about its remote sources, as well. Endpoints interpreting reception reports MUST be prepared to receive RTCP SR or RR messages where only one remote source is reporting about its sources.

#### 4. Backward compatibility

In some circumstances, the offerer in an offer/answer exchange [[RFC3264](#)] will not know whether the peer which will receive its offer supports media type multiplexing.

In scenarios where endpoints can rely on their peers supporting Interactive Connectivity Establishment (ICE) [[RFC5245](#)], even if they might not support multiplexing, this should not be a problem. An endpoint could construct a list of ICE candidates for its single session, and then offer that list, for backward compatibility, toward each of the peers; it would disambiguate the flows based on the ufrag fields in the received ICE connectivity checks. (This would result in the chosen ICE candidates participating in multiple RTP sessions,

in much the same manner as following a forked SIP offer.) For RTCWeb, it is currently anticipated that ICE will be required in all cases, for consent verification.

The more difficult case is if an offerer cannot rely on its potential peers supporting any features beyond baseline RTP (i.e., neither ICE nor multiplexing). In this case, it would either need to be prepared to use only a single media type (e.g., audio) with such a

peer, or else will need to do the pre-offer steps to set up all the non-multiplexed sessions. Notably, this would include opening local ports, and doing ICE address gathering (collecting candidate addresses from STUN and/or TURN servers) for each session, even if it is anticipated that in most cases backward compatibility is not going to be necessary.

If the signaling protocol in use supports sending additional ICE candidates for an ongoing ICE exchange, or updating the destination of a non-ICE RTP session, it is instead possible for an offerer to do such gathering lazily, e.g. opening only local host candidates for the non-default RTP sessions, and gathering and offering additional candidates or public relay addresses once it becomes clear that they are needed. (With SIP, sending updated candidates or RTP destinations prior to the call being answered is possible only if both peers support the SIP 100rel feature [[RFC3262](#)], i.e. PRACK and UPDATE; otherwise, the initial offer cannot be updated until after the 200 OK response to the initial INVITE.)

## [5.](#) Signaling

There is a need to signal multiplexed media in the Session Description Protocol (SDP) [[RFC4566](#)] -- for inter-domain federation in the case of RTCWeb, as well as for "pure" SIP endpoints that also want to use media-multiplexed sessions.

To signal multiplexed sessions, two approaches seem to present themselves: either using the SDP grouping framework [[RFC5888](#)], as in [[I-D.holmberg-mmusic-sdp-bundle-negotiation](#)], or directly representing the multiplexed sessions in SDP.

Directly encoded multiplexed sessions would have some grammar issues in SDP, as the syntax of SDP mixes together top-level media types and transport information in the m= line, splitting media types to be partially described in the m= line and partially in the a=rtpmap attribute. New SDP attributes would need to be invented to describe the top-level media types for each source.

```
a=mediamap:96 video
a=rtpmap:96 H264/90000
a=mediamap:97 audio
a=rtpmap:97 pcmu/8000
```

Figure 1: Hypothetical syntax for describing multiplexed media lines in SDP

If single-pass backward compatibility is (ever) a goal, directly encoding multiplexed sessions in SDP `m=` lines becomes much more complex, as it would require SDP Capability Negotiation [[RFC5939](#)] in order to offer both the legacy and the multiplexed streams.

Using SDP grouping seems to rule out the possibility of non-backward-compatible multiplexed streams. Other than that, however, it seems that it would be the easier path to signal multiplexed sessions.

## [6.](#) Protocols with SSRC semantics

There are some RTP protocols that impose semantics on SSRC values. Most notably, there are several protocols (for instance, FEC [[RFC5109](#)], layered codecs [[RFC5583](#)], or RTP retransmission [[RFC4588](#)]) have modes that require that sources in multiple RTP sessions have the same SSRC value.

When multiplexing, this is impossible. Fortunately, in each case, there are alternative ways to do this, by explicitly signaling RTP SSRC values [[RFC5576](#)]. Thus, when multiplexing, these modes need to be used instead.

It is unclear how to signal this in a backward-compatible way (falling back to session-multiplexed modes) if SDP grouping semantics are used to describe multiplexed sources in SDP.

## [7.](#) Security Considerations

The security considerations of a muxed stream appear to be similar to those of multiple sources of the same media type in an RTP session.

Notably, it is crucial that SSRC values are never used more than once with the same SRTP keys.

## 8. IANA Considerations

The IANA actions required depend on the decision about how muxed streams are signaled.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", [RFC 4585](#), July 2006.

### 9.2. Informative References

- [I-D.holmberg-mmusic-sdp-bundle-negotiation] Holmberg, C. and H. Alvestrand, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", [draft-holmberg-mmusic-sdp-bundle-negotiation-00](#) (work in progress), October 2011.
- [RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", [RFC 3262](#), June 2002.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, [RFC 3551](#), July 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", [RFC 3711](#), March 2004.

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

- [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.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", [RFC 5124](#), February 2008.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", [RFC 5576](#), June 2009.
- [RFC5583] Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", [RFC 5583](#), July 2009.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", [RFC 5888](#), June 2010.
- [RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", [RFC 5939](#), September 2010.

#### Authors' Addresses

Jonathan Lennox  
Vidyo, Inc.  
433 Hackensack Avenue  
Seventh Floor  
Hackensack, NJ 07601  
US

Email: jonathan@vidyo.com

Lennox & Rosenberg

Expires April 26, 2012

[Page 10]

---

Internet-Draft

Multiplexing Media Types in RTP

October 2011

Jonathan Rosenberg  
Skype

Email: [jdrosen@skype.net](mailto:jdrosen@skype.net)

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

