

**Media multiplexing with Real-time Transport Protocol (RTP) subsessions  
draft-ejzak-avtcore-rtp-subsessions-02**

Abstract

This document describes a means of multiplexing RTP streams having different media types within a single transport connection and a means of representing this multiplexing option in SDP so that network nodes can easily identify groups of RTP streams and their associated RTCP packets for differential treatment as necessary. This mechanism can be used to complement the BUNDLE multiplexing scheme, which uses the grouping framework to identify all media lines associated with a single transport connection, but provides no means for network nodes to group RTP streams and their RTCP packets for differential treatment. RTP subsessions is an alternative to the SHIM multiplexing proposal; it clearly partitions packets associated with different media lines without changing the format of individual RTP and RTCP packets, but it does not maintain a completely independent SSRC space for each media line, as does SHIM. RTP subsessions avoid SSRC conflicts by construction and can be used in all RTP topologies in which all systems implement RTP subsessions, although SSRC mapping might be needed when forwarding RTP streams from an unpartitioned RTP session into an RTP subsession.

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, 2013.

Copyright Notice

Copyright (c) 2012 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>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Overview of RTP subsessions . . . . .</a>	<a href="#">4</a>
<a href="#">2.1.</a>	<a href="#">RTP procedures . . . . .</a>	<a href="#">4</a>
<a href="#">2.2.</a>	<a href="#">Summary of SSRC bit assignments . . . . .</a>	<a href="#">5</a>
<a href="#">2.3.</a>	<a href="#">SDP procedures . . . . .</a>	<a href="#">5</a>
<a href="#">2.4.</a>	<a href="#">Additional procedures for relays . . . . .</a>	<a href="#">7</a>
<a href="#">2.5.</a>	<a href="#">When a system doesn't support RTP subsessions . . . . .</a>	<a href="#">8</a>
<a href="#">2.6.</a>	<a href="#">Alternative SDP procedures considered . . . . .</a>	<a href="#">8</a>
<a href="#">2.6.1.</a>	<a href="#">Signaling only MSID . . . . .</a>	<a href="#">8</a>
<a href="#">2.6.2.</a>	<a href="#">MSID with implicit SSRC range reservation . . . . .</a>	<a href="#">8</a>
<a href="#">3.</a>	<a href="#">Applicability of RTP subsessions . . . . .</a>	<a href="#">9</a>
<a href="#">4.</a>	<a href="#">Examples . . . . .</a>	<a href="#">10</a>
<a href="#">5.</a>	<a href="#">Evaluation . . . . .</a>	<a href="#">12</a>
<a href="#">5.1.</a>	<a href="#">Comparison to BUNDLE . . . . .</a>	<a href="#">12</a>
<a href="#">5.2.</a>	<a href="#">Comparison to SHIM . . . . .</a>	<a href="#">13</a>
<a href="#">5.3.</a>	<a href="#">Comparison to other SSRC proposals . . . . .</a>	<a href="#">13</a>
<a href="#">6.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">13</a>
<a href="#">7.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">13</a>
<a href="#">8.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">14</a>
	<a href="#">Author's Address . . . . .</a>	<a href="#">15</a>



## **1. Introduction**

The subject of RTP multiplexing has received significant attention within RTCWEB and AVTCORE in the last year. It would be highly desirable to multiplex RTP streams associated with a communications session onto as few transport connections as possible to reduce the messaging required to complete ICE connectivity checks [[RFC5245](#)] and DTLS key exchange [[RFC6347](#)] prior to being able to send media. RTP is specified in [[RFC3550](#)]. This document focuses specifically on how to enable network nodes to provide differential treatment to multiplexed media types within the same transport connection, since means already exist to apply differential treatment to an entire transport connection and RTP is capable of multiplexing RTP streams of the same media type onto a single transport connection using the SSRC field. While an application may be able to request different DSCP markings for these flows, the treatment associated with a particular marking is network-specific and many networks routinely remap packet markings according to local policy unless they can independently verify the requested treatment.

The following drafts provide key background and context, which will not be duplicated here: [[I-D.ietf-rtcweb-rtp-usage](#)], [[I-D.westerlund-avtcore-multiplex-architecture](#)], [[I-D.westerlund-avtcore-max-ssrc](#)] and [[I-D.alvestrand-mmusic-msid](#)]. Note in particular that the use of MSID, mandated in RTCWEB, uses SSRC reservation to assist in stream identification.

The BUNDLE solution [[I-D.ietf-mmusic-sdp-bundle-negotiation](#)] uses SDP [[RFC4566](#)] to assign different sets of payload type values for each media line sharing an RTP session. The SHIM solution [[I-D.westerlund-avtcore-transport-multiplexing](#)] extends the RTP frame with an RTP session identifier to allow multiple RTP sessions to share a single transport connection. Schemes no longer under consideration, based on SSRC, are described in [[I-D.rosenberg-rtcweb-rtpmux](#)] and [[I-D.peterson-rosenberg-avt-rtp-ssrc-demux](#)].

This document describes an optional enhancement to BUNDLE that allows network nodes to identify RTP streams associated with an SDP media line and their associated RTCP packets for differential treatment in the network. Also included is a description of a means of negotiating use of RTP subsessions in SDP, along with some alternatives. This document assumes the use of unicast RTP sessions only and there is no discussion of multicast sessions.



## **2. Overview of RTP subsessions**

### **2.1. RTP procedures**

An RTP subsession is a set of RTP streams and corresponding RTCP packets that use a subset of the entire range of SSRC values. Each RTP subsession has most of the characteristics of a complete RTP session with some exceptions described below. RTP subsessions that are assigned non-overlapping SSRC value ranges can share a single transport connection.

As described in [[RFC3550](#)], RTP topologies can include end systems, translators(relays) and mixers. RTP subsessions are particularly suited to supporting end systems and mixers, since these systems typically remap SSRC values when forwarding RTP streams and so can independently assign SSRC values on each transport connection. The remainder of this section describes use of RTP subsessions by end systems and mixers; a later section describes use of RTP subsessions by relays.

Each RTP subsession sharing a single transport connection is assigned a unique 24-bit SSRC prefix for each direction of transmission, i.e., the first (highest order) bit is reserved to indicate each direction of transmission and the next 23 bits of the 32-bit SSRC field have a fixed value for RTP streams associated with the RTP subsession. Each assigned SSRC prefix is also unique in the first 8 bits to allow for potential connection to relays that forward RTP streams from other sources. The network can use the first 8 bits of the SSRC prefix to filter IP packets on the transport connection to identify those associated with the RTP subsession.

In non-relay topologies, the SDP offerer selects the first 24 bits of the SSRC for RTP streams associated with a media line that it transmits to the SDP answerer, and the SDP answerer selects the other value of the 1st bit and the same values of the remaining 23 bits of the SSRC prefix selected by the SDP offerer for RTP streams associated with the same media line that it transmits in the other direction to the SDP offerer. The final 8 bits of the SSRC are available to specify separate RTP streams within the RTP subsession. The SDP offerer selects the 24 bits of the SSRC prefix randomly for each RTP subsession in such a way that the first 8 bits are also unique across known RTP subsessions in the transport connection, and each endpoint selects the last 8 bits of the SSRC randomly for each RTP stream within an RTP subsession. The SDP offerer is allowed to specify either value for the 1st bit of the SSRC to allow the offerer and answerer to change roles during subsequent session modifications while allowing continued use of already assigned SSRC values.

Ejzak

Expires April 26, 2013

[Page 4]

When concatenating multiple RTP or RTCP packets within a single IP packet, as allowed by existing specifications, RTP systems will only combine packets from the same RTP subsession. RTP subsessions are thus segregated into separate IP packets to allow differential treatment in the network.

## 2.2. Summary of SSRC bit assignments

RTP subsessions supports topologies including end systems, mixers and translators (relays), where relays are classified either as "master" or "slave". These end systems share responsibility for SSRC bit assignments as described in the following table, using the procedure described in the following section. There is a strict precedence order between these systems, where a master relay can override SSRC prefix assignments made by a slave relay, which can override SSRC prefix assignments made by a mixer or end system, but not the other way around.

SSRC bit range	Purpose
1	identifies direction of flow
2-8	subsession id (for QoS)
9-16	allocated by primary relay to slave relays (except one value reserved for self)
17-24	allocated by slave relays to non-relay systems
25-32	allocated by non-relay systems to individual RTP streams (e.g., via MSID)

SSRC bit assignments

## 2.3. SDP procedures

The use of RTP subsessions is negotiated during the SDP offer/answer exchange as follows. When signaled in combination with BUNDLE, BUNDLE defines each group of media lines to share a transport connection, and the SDP attribute for RTP subsessions defines the SSRC prefix for each media line. Use of RTP subsessions without BUNDLE is possible but not defined within this document. This draft assumes that only one port is assigned for transport of RTP streams on each m line, since use of multiple ports, though allowed in [\[RFC4566\]](#), is rarely used. SDP subsessions can be readily enhanced to support multiple media ports per m line if necessary.

BUNDLE specifies how the connection and port information is to be set for each media line in a BUNDLE group. When RTP subsessions is used





with BUNDLE, there is no requirement for the payload type numbers for each media line in the group to be non-overlapping, as required for BUNDLE without RTP subsessions. The payload type field is not needed to identify the media line associated with an RTP stream if the SSRC prefix is known.

A system supporting RTP subsessions will include an "ssrc-prefix" attribute for each media line in a sharing group, where each ssrc-prefix attribute includes a parameter that indicates whether the system is a relay (supporting RTP stream forwarding) and a parameter that is a hex representation of the 24-bit SSRC prefix proposed for use by the RTP subsession associated with the m line. If the endpoint expects to transmit or receive more than 256 RTP streams of the same media type, it should allocate more than one m line for that media type. The system will also include the rtcp-mux attribute [[RFC5761](#)] for each media line sharing the same transport connection. A system supporting RTP subsessions that receives SDP with ssrc-prefix attributes will assume the presence of the rtcp-mux attribute for each associated media line, even in its absence. While rtcp-mux could be made optional, there is no clear use case for supporting RTP subsessions without rtcp-mux.

In addition to including the requisite BUNDLE attributes, the SDP offer includes the ssrc-prefix attribute and the rtcp-mux attribute for each media line in the sharing group. If the SDP answerer is not a relay and agrees to use RTP subsessions (regardless of the role of the SDP offerer), or if the SDP offerer is not a relay and the SDP answerer is a relay that agrees to use RTP subsessions with the value(s) of the SSRC prefix in the SDP offer, then the SDP answerer also includes in the SDP answer the BUNDLE attributes, the ssrc-prefix attribute and the rtcp-mux attribute for each media line in the sharing group, with the same ssrc-prefix parameter values as the corresponding ones from the SDP offer, but with the 1st bit changed, i.e., '0' becomes '1' or '1' becomes '0'. The SDP answerer can disable the use of RTP subsessions by not including the ssrc-prefix attributes in the answer.

If the SDP offerer is not a relay and the SDP answerer is a relay that selects not to use the SSRC prefix in the SDP offer for one or more of the media lines, then it includes in the SDP answer the BUNDLE attributes, the ssrc-prefix attribute and the rtcp-mux attribute for each media line in the sharing group, with its own ssrc-prefix parameter values for those selected media lines and as described in the previous paragraph for the remaining media lines. The SDP offerer must then use the ssrc-prefix values from the selected media lines in the SDP answer with the 1st bit changed as above. Note that for these selected media lines, any SSRC values selected by other SDP attributes (such as ssrc for msid) in the SDP



offer are assumed by the SDP answerer to be remapped to include the ssrc-prefix from the SDP answer with the 1st bit changed and the original final 8 bits from the attribute in the SDP offer. The SDP offerer will also assume the use of the remapped values and include the remapped values in any subsequent SDP messages.

#### **2.4. Additional procedures for relays**

RTP subsessions will work without SSRC collisions for RTP system configurations consisting of a single master relay directly connected to up to 256 slave relays, where each relay may be directly connected to up to 256 end systems or mixers. Note that a master relay incorporates the function of at least one slave relay so that it can connect to non-relay systems. The only requirements are that the master relay initiate all connections to separate slave relays using SDP offer/answer procedures and that all participating systems in the RTP session comprising the RTP subsessions support SDP offer/answer procedures to negotiate RTP subsessions. The master relay picks the first 8 bits of the SSRC for every RTP subsession (used for filtering in the network), thus allowing up to 128 RTP subsessions within an RTP session. Each RTP subsession multiplexes traffic that is to receive the same QoS treatment throughout the RTP session. The master relay picks the first 16 bits of the SSRC that a slave relay can use with a particular RTP subsession and the slave relay picks the first 24 bits of the SSRC that an end system or mixer can use with a particular RTP subsession. The non-relay systems are free to allocate the last 8 bits of the SSRC to multiplex RTP streams on an RTP subsession.

The previous sections already describe how SDP offer/answer occurs between relay and non-relay systems to ensure that the relay determines the 24-bit SSRC prefix that the non-relay system uses for an RTP subsession, regardless of which system initiates the SDP offer/answer procedure.

Once a relay system chooses the role of a master relay by selecting an SSRC prefix for an RTP subsession, it refuses to accept an SDP offer from any other relay. When sending an SDP offer to another system, without knowing if it is another relay, it selects a 24-bit SSRC prefix for each media line as already discussed, ensuring that the first 16 bits of the SSRC have not been assigned to any other transport connection. If the SDP answerer is not a relay, then the 24-bit SSRC prefix is reserved for the non-relay system to use for the RTP subsession. If the SDP answerer is another relay (a slave relay), it can accept the RTP subsession in the same manner as a non-relay system by responding with the same ssrc-prefix with the first bit changed. A slave relay is not allowed to respond to a master relay with a different SSRC prefix except for the changed first bit.



Since the SDP offerer and answerer both signal that they are relays, they both understand that the master relay will reserve the first 16 bits of the SSRC to the slave relay, thus allowing the master relay to delegate SSRC prefix assignment to up to 256 slave relays and allowing each slave relay to select the 24-bit SSRC prefix for the RTP subsessions for up to 256 non-relay systems.

## **[2.5.](#) When a system doesn't support RTP subsessions**

When a non-relay system fails to negotiate the use of RTP subsessions with a peer, it will be difficult for the network to identify flows for differential treatment in the presence of BUNDLE type multiplexing. Once a relay has committed to RTP subsessions with one or more other systems, it has only two choices if a new system being added to the RTP session does not support RTP subsessions: it can re-negotiate with all existing systems to disable RTP subsessions; or it can perform SSRC mapping with the non-conformant system.

## **[2.6.](#) Alternative SDP procedures considered**

Since [[I-D.alvestrand-mmusic-msid](#)] already describes an SSRC reservation procedure for SDP, it is appropriate to consider the relationship to RTP subsessions and whether MSID can be used to simplify the signaling of RTP subsessions in SDP.

### **[2.6.1.](#) Signaling only MSID**

If we consider using MSID without change to signal the use of RTP subsessions, there are two limitations to consider. MSID provides for explicit identification of the SSRC used for each stream, so a traffic filter could be constructed to search for RTP packets with particular SSRC values, but this is quite cumbersome once there is any significant amount of multiplexing per media line. Furthermore, MSID includes no SSRC collision resolution procedure, thus making the use of SDP offer/answer re-negotiation necessary for collision resolution. While SSRC collisions are fairly rare if SSRC assignment is completely random, collisions will occur and this form of resolution is expensive and preferably avoided.

### **[2.6.2.](#) MSID with implicit SSRC range reservation**

RTP subsession negotiation could be easily added to MSID as a mandatory feature, as follows. If selection of an SSRC value for MSID implicitly reserves SSRC ranges according to the conventions described earlier in the document, then there would be no need to explicitly signal the SSRC prefix. The only aspect missing is to create a convention to negotiate precedence (master relay, slave relay, or other) to provide for a simple method of resolving any SSRC



or SSRC range collisions. Details are left for further study if this proposal for enhancing MSID is agreed.

### 3. Applicability of RTP subsessions

Each RTP subsession is able to provide most of the capabilities of a full RTP session with some limitations associated with constraints on SSRC assignment. No changes are required to RTP or SRTP packet formats to realize RTP subsessions. The RTP streams associated with an individual media line can be easily identified for separate handling (e.g., to provide separate QoS treatment) by filtering on the first 8 bits of the SSRC field, in addition to the five-tuple for the transport connection. RTP systems can enable successful filtering on the port and SSRC fields, which are above the IP layer, by adhering to MTU limits to avoid IP fragmentation.

[RFC3550] describes three kinds of systems that can use RTP: end systems, translators (relays) and mixers. RTP subsessions can be freely used by end systems and mixers since these systems can freely assign any SSRC value to each RTP stream and thus are free to detect and resolve SSRC conflicts. For interworking with systems not supporting RTP subsessions in particular, a relay can act as a mixer to reassign the SSRC value associated with an RTP stream before forwarding. When performing SSRC mapping rather than SRC forwarding, RTCP is handled differently and SRTP forwarding is more complex, requiring the decrypting and re-encrypting of each packet, as discussed in [[I-D.westerlund-avtcore-multiplex-architecture](#)]. While there is some additional processing needed to change SSRC when forwarding SRTP, there seems to be no consensus in RTCWEB to require support for forwarding with SSRC preservation. Systems that provide additional media processing functions before forwarding RTP streams need to decrypt SRTP packets anyway.

The relay procedures for RTP subsessions allow support of most relay topologies as long as all systems in an RTP session support RTP subsessions. This is a limitation in the short term when interworking with existing systems, but if RTP subsession support is made available early in the deployment of RTCWEB systems, then new applications where relay systems, multiplexing of different media types, and differential treatment of media flows are desirable can require support for RTP subsessions.

Some applications require the ability to allocate the same SSRC value across multiple ports or media lines. RTP streams in different RTP subsessions are considered to use identical SSRC values if they match on the last 24 bits of the SSRC, so RTP subsessions can also be used for these applications.





Most RTP capabilities and extensions depend primarily on the use of a different SSRC for each RTP stream. Since RTP subsessions retain this characteristic, they introduce no new compatibility issues. Examples of such extensions include FEC, interleaving and RTP retransmissions.

Another short term limitation of RTP subsessions is that most networks capable of assigning differential treatment do so with the granularity of a five-tuple. The availability of a simple filter to identify flows for differential treatment allows deployment of DPI or custom gateways to enforce or verify DSCP markings in the short term. In the longer term, network policy control systems can be enhanced to perform SSRC prefix filtering.

#### 4. Examples

In the first example, a non-relay SDP offerer desiring to use a single transport connection for an audio m line and a video m line might construct the following SDP offer. BUNDLE attributes are present but not shown.

```
C=...
m=audio 49170 RTP/AVP 96 97
a=rtcp-mux
a=ssrc-prefix: non-relay 0x111111
...
m=video 49170 RTP/AVP 98 99
a=rtcp-mux
a=ssrc-prefix: non-relay 0x222222
...
```

The non-relay SDP answerer agrees to use SDP subsessions as described in the SDP offer and accepts the SSRC prefixes for the two m lines as shown below. BUNDLE attributes are present but not shown.

```
C=...
m=audio 36008 RTP/AVP 96 97
a=rtcp-mux
a=ssrc-prefix: non-relay 0x911111
...
m=video 36008 RTP/AVP 98 99
a=rtcp-mux
a=ssrc-prefix: non-relay 0xa22222
...
```

The endpoints will establish one audio RTP subsession and one video RTP subsession associated with the SDP offer/answer exchange. These



SDP subsessions and their corresponding RTCP will each share a single transport connection using ports 49170 and 36008, respectively. Media flows associated with each SDP subsession can be identified by filtering on the first 8 bits of the SSRC field as necessary for differential handling, e.g., to assign separate QoS treatment. The SDP offerer will send RTP streams on the two RTP subsessions for the audio and video media m lines using 24-bit SSRC prefixes 0x111111 and 0x222222, respectively. The SDP answerer will send RTP streams using 24-bit SSRC prefixes 0x911111 and 0xa22222, respectively. The network can filter on 8-bit SSRC prefixes 0x11 and 0x22 for packets in the five-tuple from the SDP offerer and the network can filter on 8-bit SSRC prefixes 0x91 and 0xa2 for packets in the five-tuple sent to the SDP offerer.

In another example to highlight how a relay may override an SSRC prefix selected by a non-relay system, a non-relay SDP offerer desiring to use a single transport connection for an audio m line and a video m line might construct the following SDP offer. BUNDLE attributes are present but not shown.

```
C=...
m=audio 49170 RTP/AVP 96 97
a=rtcp-mux
a=ssrc-prefix: non-relay 0x111111
...
m=video 49170 RTP/AVP 98 99
a=rtcp-mux
a=ssrc-prefix: non-relay 0x222222
...
```

The SDP answerer performing as an RTP relay agrees to use SDP subsessions as described in the SDP offer. It accepts the SSRC prefix for the audio m line but selects an alternate SSRC prefix for the video m line as shown below. BUNDLE attributes are present but not shown.

```
C=...
m=audio 36008 RTP/AVP 96 97
a=rtcp-mux
a=ssrc-prefix: relay 0x911111
...
m=video 36008 RTP/AVP 98 99
a=rtcp-mux
a=ssrc-prefix: relay 0x333333
...
```

The endpoints will establish one audio RTP subsession and one video RTP subsession associated with the SDP offer/answer exchange. These



SDP subsessions and their corresponding RTCP will each share a single transport connection using ports 49170 and 36008, respectively. Media flows associated with each SDP subsession can be identified by filtering on the first 8 bits of the SSRC field as necessary for differential handling, e.g., to assign separate QoS treatment. The non-relay SDP offerer will send RTP streams on the two RTP subsessions for the audio and video media m lines using 24-bit SSRC prefixes 0x111111 and 0xb33333, respectively. The SDP answerer performing as an RTP relay will send RTP streams (forwarded from other systems) using 8-bit SSRC prefixes 0x91 and 0x33, respectively. The network can filter on 8-bit SSRC prefixes 0x11 and 0xb3 for packets in the five-tuple from the SDP offerer and the network can filter on 8-bit SSRC prefixes 0x91 and 0x33 for packets in the five-tuple sent to the SDP offerer. Note that the 32-bit prefix for any ssrc values selected for RTP streams associated with the video m line are assumed to be changed from 0x222222 to 0xb33333 while ssrc values selected for the audio line can remain unchanged.

## **5. Evaluation**

### **5.1. Comparison to BUNDLE**

BUNDLE [[I-D.ietf-mmusic-sdp-bundle-negotiation](#)] effectively merges RTP sessions together in relay topologies, thus slightly increasing the probability of SSRC collisions, although the impact is minor. In the context of RTCWEB, an SSRC collision, though rare, will require the use of an additional SDP offer/answer transaction to change msid/ssrc mappings, so is clearly undesirable. This can be avoided with RTP subsessions, since SDP offer/answer is used to pre-allocate SSRC ranges for each m line, thus completely avoiding SSRC collisions. It must also be assured that different RTP streams do not share the same SSRC, even though they use non-overlapping payload type values, so that each RTCP packet can be uniquely associated with a particular RTP stream. It is difficult to define a filter to allow the network to identify the RTP streams associated with different media lines with BUNDLE since this requires filtering on sets of payload type values. RTP subsessions can enhance BUNDLE for this purpose, since it is only necessary to screen for a single value in the first 8 bits of the SSRC. It is also not possible with BUNDLE to associate RTP streams from different m lines using a single SSRC value (as it is possible to do using the last 24 bits of the SSRC with RTP subsessions), due to the need to separate the RTCP messages for each RTP stream.

The author proposes that RTP subsessions be adopted to augment BUNDLE to eliminate the restrictions described above.



## **5.2. Comparison to SHIM**

SHIM [[I-D.westerlund-avtcore-transport-multiplexing](#)] is the most successful scheme in preserving the SSRC space of each RTP session when multiplexing those RTP sessions onto a single transport connection. SHIM changes RTP so it can potentially impact many different middleboxes. It also slightly increases the size of each media packet, which can be of concern in some bandwidth-limited deployments.

The author recommends RTP subsessions over SHIM to better address two key requirements: to support flow identification for differential treatment; and to minimize impact on the RTP packet format. Given the increasing use of pre-selected SSRC values to identify individual RTP streams, which is inconsistent with the idea of random SSRC assignment and the use of SSRC collision detection and resolution schemes, the author also recommends the use of an effective SSRC collision avoidance scheme based on SSRC range reservation, such as the one defined for RTP subsessions.

## **5.3. Comparison to other SSRC proposals**

Two expired drafts, [[I-D.rosenberg-rtcweb-rtpmux](#)] and [[I-D.peterson-rosenberg-avt-rtp-ssrc-demux](#)], propose other means of multiplexing based on the SSRC field. [[I-D.rosenberg-rtcweb-rtpmux](#)] uses a portion of the SSRC to identify the media type for the RTP stream. This scheme redefines the SSRC field and has significant limitations when multiple m lines share the same media type, since some other mechanism is still needed to separate them. [[I-D.peterson-rosenberg-avt-rtp-ssrc-demux](#)] assumes a single SSRC value per m line, so is not consistent with current RTP multiplexing requirements.

Neither earlier SSRC-based scheme fully addresses the requirements for multiplexing agreed in the working groups.

## **6. IANA Considerations**

To be completed.

## **7. Security Considerations**

To be completed.





## 8. Informative References

- [I-D.alvestrand-mmusic-msid]  
Alvestrand, H., "Cross Session Stream Identification in the Session Description Protocol", [draft-alvestrand-mmusic-msid-01](#) (work in progress), October 2012.
- [I-D.ietf-mmusic-sdp-bundle-negotiation]  
Holmberg, C. and H. Alvestrand, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", [draft-ietf-mmusic-sdp-bundle-negotiation-00](#) (work in progress), February 2012.
- [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-02](#) (work in progress), March 2012.
- [I-D.peterson-rosenberg-avt-rtp-ssrc-demux]  
Peterson, J. and J. Rosenberg, "A Multiplexing Mechanism for the Real-Time Protocol (RTP)", [draft-peterson-rosenberg-avt-rtp-ssrc-demux-00](#) (work in progress), July 2004.
- [I-D.rosenberg-rtcweb-rtmux]  
Rosenberg, J., Jennings, C., Peterson, J., Kaufman, M., Rescorla, E., and T. Terriberry, "Multiplexing of Real-Time Transport Protocol (RTP) Traffic for Browser based Real-Time Communications (RTC)", [draft-rosenberg-rtcweb-rtmux-00](#) (work in progress), July 2011.
- [I-D.westerlund-avtcore-max-ssrc]  
Westerlund, M., Burman, B., and F. Jansson, "Multiple Synchronization sources (SSRC) in RTP Session Signaling", [draft-westerlund-avtcore-max-ssrc-01](#) (work in progress), April 2012.
- [I-D.westerlund-avtcore-multiplex-architecture]  
Westerlund, M., Burman, B., and C. Perkins, "RTP Multiplexing Architecture", [draft-westerlund-avtcore-multiplex-architecture-01](#) (work in progress), March 2012.
- [I-D.westerlund-avtcore-transport-multiplexing]  
Westerlund, M. and C. Perkins, "Multiple RTP Sessions on a



Single Lower-Layer Transport",  
[draft-westerlund-avtcore-transport-multiplexing-02](#) (work  
in progress), March 2012.

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", [RFC 5761](#), April 2010.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", [RFC 6347](#), January 2012.

#### Author's Address

Richard Ejzak  
Alcatel-Lucent  
1960 Lucent Lane  
Naperville, Illinois 60563-1594  
US

Phone: +1 630 979 7036  
Email: [richard.ejzak@alcatel-lucent.com](mailto:richard.ejzak@alcatel-lucent.com)

