   A Generic Bundle Mechanism for the Session Description Protocol (SDP)
                        draft-worley-sdp-bundle-06

Abstract

   This document defines a generic bundle mechanism for the Session
   Description Protocol (SDP) by which the media described by a number
   of media descriptions ("m= lines") are multiplexed and transmitted
   over a single transport association. The single transport
   association is described by an additional media description, allowing
   SDP attributes to be applied to the aggregate, independently of
   attributes applied to the constituents. In offer/answer usage, the
   bundle mechanism is backward compatible with SDP processors that do
   not understand the mechanism. The mechanism is designed to be
   compatible with the limitations of the existing Internet
   infrastructure.

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 September 16, 2013.

Table of Contents

1. Introduction

The central idea of bundling is to multiplex the media that would be
several transport flows into one transport flow, with particular
emphasis on allowing one transport association to carry media that
are presented to the higher, application layer, as multiple transport
flows.

At the interface between the SDP-configured layer and the lower,
transport layer, the media are organized into a single transport
flow. The transport-related properties of the RTP session (e.g.,
transport 5-tuple, encryption, ICE) are described by the transport-
related attributes of a single media description.

At the interface between the SDP-configured layer and the higher,
application layer, the media are organized into several transport
flows. The application-related properties of the transport flow
(e.g., media type and label) are described by the application-related
attributes of separate media descriptions.

(There are some attributes (e.g., bandwidth limitation) that can
apply separately to both the bundled transport flow and the
constituent transport flows.)

However, we do not include the payload type numbers as information
available to the application; only the encoding name and its
parameters are accessible to the application. This gives the bundle
mechanism freedom to place constraints on the use of payload types.

The bundle is signaled in the session description by a "group"
attribute with semantics "BUNDLE". The first media description
listed in the group is the "bundle" media description (MD), whose
transport information describes the transport association via which
the packets will be sent. The remaining (zero or more) media
descriptions listed in the group are the "constituent" MDs. Packets
(either RTP/RTCP/SRTP/SRTCP, SCTP, or SCTP-over-DTLS) received from
the applications for these MDs are sent (unaltered) on the transport
association for the bundle MD. Packets received from the transport
association for the bundle MD are demultiplexed (based on particular
features of the constituent transport flow protocols and the payload
types and SCTP ports specified in the constituent MDs) and sent to
the applications for the constituent MDs.

In offer/answer usage, we must arrange that the bundle mechanism is
backward compatible with entities that do not understand the bundle
mechanism. This requirement drives many features of this solution.
[Section 6.1](#)

In addition, many devices in current usage (especially SBCs) apply
more restrictions on the usage of SDP than one would expect from
abstract consideration of their roles in the network. Some features
of this solution are constructed to avoid these restrictions.
[Section 6.2](#)

[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 [[RFC2119](#)].

The important RFCs in this area use inconsistent terminology. Here,
we use:

media Media is (1) media content, considered in an abstract way,
    that is, without consideration of its particular encoding or the
    framing information around it, and (2) the particular bits and
    octets used to encode and transmit the abstract media content.

media stream (Taken from [[RFC3550](#)].) A media stream is a set of RTP
    packets that are generated by and interpreted by one codec. The
    RTP packets of a media stream are identified by a unique SSRC.

capture (Taken from CLUE's work.) A capture is a set of media
   streams that originate from one (physical or virtual) media source
   and should be composed to provide rendering of that source. For
   example, media streams from one origin including layered
   encodings, forward error correction streams, recovery streams, and
   simulcasted media streams of varying bit rates compose one
   capture.

transport association (Taken from [I-D.alvestrand-mmusic-msid].) A
   transport association is a single data path between two hosts,
   such as a TCP connection, or a pair of UDP ports that send packets
   to each other. A transport association is identified by the
   identity of the protocol being used, the relevant host addresses,
   and the relevant port numbers. In the case of unicast
   communications, these form a "5-tuple", namely, the protocol, the
   host addresses of the two hosts, and the port numbers used on the
   two hosts. In the case of multicast sessions, these form a
   "3-tuple", namely, the protocol, the multicast address, and the
   port number. In SDP, a transport association is specified by the
   address and port of a media description (and possibly the same
   information from the matching offer/answer SDP). If a media
   description specifies multiple addresses or ports, each address or
   port specifies one transport association.

transport flow (Taken from
   [I-D.ietf-avtcore-multi-media-rtp-session].) (This is called an
   "RTP session" by [RFC3264].) A transport flow is the data that
   flows across a transport association.

media description (Taken from [RFC4566].) A media description is
   one group of lines in a session description demarcated by an m=
   line. By synecdoche, a media description is often referred to as
   "an m= line".

transport association group A transport association group is the set
   of transport associations denoted by one media description.
   Usually the m= line specifies only one port and the c= line
   specifies only one address, and so the media description's
   transport association group contains only one transport
   association.

transport flow group A transport flow group is the set of transport
   flows of the transport associations of a transport association
   group.

session description (Taken from [RFC4566] section 2.) A session
   description is an SDP instance.

multimedia session (Taken from [RFC4566] section 2.) A multimedia
   session is the totality of the media that is transmitted/received
   as described by a particular session description.

RTP session (Taken from [RFC3550].) An RTP session is a group of
   media streams which must not have duplicated SSRC values because
   the endpoints share RTCP reporting information. Note that an RTP
   session may encompass more than one multimedia session. RTP
   sessions are not fully described by session descriptions.

3. Desiderata

This section lists desiderata for the bundle mechanism in SDP. (I
use the term "desiderata" -- "things that are desired" -- rather than
"requirements", because we may discover that we can't optimally
satisfy all of these criteria at the same time.) The first section
lists desiderata that are arise from considering the ways
applications may wish to bundling. The second section lists
desiderata that arise from compatibility with existing Internet
infrastructure.

3.1. Feature Desiderata

These desiderata describe features that we would like the bundling
mechanism to provide.

DES F1 For each bundle, there is a group of media descriptions which
   describe the application-level RTP sessions. This specification
   must allow the same granularity of description as when the media
   flows were not multiplexed. This description includes identifiers
   which connect the media flows with the application and with each
   other.

This requirement is taken from [I-D.jennings-mmusic-media-req].

DES F2 For each bundle, there is a media description that describes
   the transport-level RTP session.

DES F1 and DES F2 do not specify whether the transport-level media
description may or may not also be one of the application-level media
descriptions.

DES F3 There must be a uniform way to deal with new SDP parameters,
   so that newly defined SDP parameters do not require a specific
   updating of the bundling procedures.

This desideratum is taken from slides-interim-2013-rtcweb-1-10.pdf.

DES F4 Multiple separate bundles within one SDP must be supported.

DES F5 Bundles may contain other bundles as constituents.

Of course, no bundle may directly or indirectly contain itself. (I
don't expect any current implementation to implement bundles within
bundles, but we should design the mechanism to allow this, as some
day we will likely need it.)

DES F6 A bundle may contain zero constituents.

A bundle with no constituents serves no purpose for the transport of
media, but we are likely to someday need to describe such a bundle.
(Compare that an SDP m= line is syntactically constrained to specify
at least one payload type. When SDP was used only to specify
multicast sessions, this constraint was common sense. But once SDP
offer/answer was invented, when a media description was rejected, the
natural representation would be an m= line with a zero port and no
payload types. But a payload type was syntactically required, so we
now have to provide at least one token payload type in rejected m=
lines.)

DES F7 If an answerer that does understand the bundle mechanism
   processes an offer that contains a bundle, it must be able to (1)
   accept the bundle and selectively accept or reject each
   constituent RTP session within it, (2) reject the bundle as a
   whole, or (3) reject the bundling and selectively accept or reject
   each constituent RTP session as separate RTP sessions.

Presumably answer (3) resembles that which would be produced by an
answerer that does not understand the bundle mechanism. It is a
lower priority that the answerer can distinguish between accepting
the bundle while rejecting all of its constituents, and rejecting the
bundle as a whole. But those two conditions differ conceptually
regarding whether any "framing" actions of the bundle are performed.

DES F8 There must be a reliable way to demultiplex incoming RTP into
   the separate application-level RTP sessions. Similarly, there
   must be a reliable way to demultiplex the associated RTCP
   information.

The RTCP information for each media stream is tagged with the SSRC
about which it reports, and the SSRC is used to correlate the RTCP
reports with the RTP sessions containing media with the same SSRC.
So regarding RTCP, this desideratum appears to be straightforward to
satisfy.

   DES F9 The specification must specify any needed additional
      procedures for handling SSRC collisions between media sources
      within different application-level RTP sessions, as those can now
      collide.

   In the terminology of [RFC3550], the constituent media descriptions
   are now part of one RTP session.

   DES F10 When an offer is constructed, the offerer must not need to
      preallocate TURN relays for constituent media descriptions. When
      both endpoints support bundling, the mechanism must not require
      the offerer to allocate TURN relays for constituent media
      descriptions.

   This desideratum was suggested by Andrew Hutton.

   DES F11 It must be possible to add and remove one way video flows
      within the bundle without requiring an additional offer/answer
      cycle.

   Presumably this can be accomplished as it is now, with a single media
   description carrying multiple video flows that are distinguished only
   by their SSRCs. This desideratum is taken from slides-
   interim-2013-rtcweb-1-10.pdf.

   DES F12 Bundling must not interfere with ICE usage, and in
      particular, ICE's ability to negotiate both IPv4 and IPv6
      addresses simultaneously.

   This desideratum was suggested by Andrew Hutton.

   DES F13 It is desirable for RTP media to appear on the wire as
      (encrypted) SRTP packets and for RTCP reports to appear on the
      wire as (authenticated but not encrypted) SRTCP packets. Thus,
      encryption of RTP/RTCP via DTLS and encapsulation of multiplexed
      packets is not desirable.

   This desideratum was suggested by Martin Thompson (http://
   www.ietf.org/mail-archive/web/mmusic/current/msg10400.html) and Dan
   Wing (http://www.ietf.org/mail-archive/web/mmusic/current/
   msg10408.html).

3.2. Compatibility Desiderata

   These desiderata describe compatibility of the bundling mechanism
   with with non-supporting endpoints or with existing entities in the
   Internet infrastructure.

DES C1 In offer/answer usage, an endpoint using the bundle mechanism
   must interwork correctly with an endpoint that does not understand
   the bundle mechanism.

DES C2 Interworking must continue when SDP endpoints are replaced
   with other endpoints during a sequence of offer/answer exchanges
   (such as happens in 3PCC or call transfers "behind an SBC"),
   including when a supporting endpoint is replaced by a non-
   supporting endpoint or vice-versa.

SDP features (e.g., the codec set and ICE) are generally designed so
that an offerer always offers every facility it is willing to support
in the current situation, regardless of whether it was agreed to by
the answerer in a preceding exchange. Thus, if the current answerer
is a different endpoint than the previous answerer, the new answerer
will negotiate a compatible set of facilities without needing
knowledge of its predecessor's SDP. The offerer will smoothly
transition to the new facilities. This property is required to
support 3PCC situations (e.g., [RFC3725] and
[I-D.worley-service-example]). This desideratum was suggested by
Richard Ejzak.

DES C3 Avoid using media types in m= lines other than audio and
   video unless required for user media, as some SBCs reject SDP that
   uses other media types.

This desideratum was suggested by Hadriel Kaplan.

DES C4 Any additional m= lines prescribed by the bundle mechanism
   should be ordered after the constituent m= lines.

Many devices that have only one audio or video channel accept the
first m= line with that media type and reject any further ones

non-DES C5 SBCs generally pass through attributes that they do not
   understand. SBCs generally pass through codec specifications that
   they do not understand, even if they are configured to transcode
   certain specific codecs.

This non-desideratum was suggested by Hadriel Kaplan.

DES C6 After offer/answer processing is finished, if the exchanged
   SDP is examined by a non-supporting SBC, the set of transport
   associations that it sees being specified for media exchange
   should be the set that are actually used for media transfer.

This is needed because SBCs monitor the packet traffic on the
transport associations and if no media is seen on one of the

associations for a significant period of time, the SBC will tear down
the call. This desideratum was suggested by Hadriel Kaplan.

DES C7 In a session description, no endpoint of a transport
   association may be used multiple times.

Such duplication is not defined by [RFC4566]. Some SBCs do not
support such duplication (ultimately, because it was not supported by
[RFC2327]), and they reject SDP specifying duplicated transport
association endpoints. This desideratum was suggested by Cullen
Jennings.

DES C8 Offer/answer processing between supporting processors must be
   completed in one exchange. When interworking between supporting
   and non-supporting processors, it is less desirable but admissible
   that a second offer/answer exchange may be needed to complete
   configuring the multimedia session.

DES C9 If an intermediate entity provides transcoding between
   codecs, and if the offer/answer does not specify encryption of a
   media stream, the media stream should be able to take advantage of
   the transcoding facility.

The non-encrypted case is not expected to be very common. Encrypted
media can't be transcoded by an intermediate entity.

4. Tutorial Examples

This section is non-normative. (This section was suggested by
Charles Eckel.)

This is an introduction to SDP bundling via a series of examples of
offer/answer processing. Some mandatory SDP lines have been omitted
from the examples for brevity. Long SDP lines have been folded by
using trailing backslashes. Blank lines have been inserted for
clarity.

4.1. One Audio Stream and One Video Stream

4.1.1. Offer without Bundling

Here is a typical, non-bundled SDP example with both audio and video
media:

```
        o=- 2890844526 2890844526 IN IP4 host.example.com
        c=IN IP4 host.example.com
```

This SDP media description (MD) provides the transport information

about the audio and also identifies the role of the audio from the
application's point of view. In this case, the fact that it is the
first audio m= line suffices to tell the application how to treat it.
In more complex cases, label or content attributes might be used to
communicate the proper handling to the application.

```
m=audio 10000 RTP/AVP 0 8 97
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=candidate:0 1 UDP 2113601791 10.0.1.1 10000 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51000 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 31000 typ relay
```

This MD provides the transport information about the video and also
identifies the role of the video from the application's point of view.

```
m=video 10002 RTP/AVP 31 32
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=candidate:0 1 UDP 2113601791 10.0.1.1 10002 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51002 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 31002 typ relay
```

We call the RTP that is described by each media description (MD) a
transport flow (TF). The audio and video are carried in separate
TFs, which each have a separate transport association (address/port).

4.1.2. Offer with Bundling

With SDP bundling, we add an additional MD to describe a single
"bundle" TF to carry both the audio and video information, and a
group attribute to show the association of the bundle MD with the
constituent MDs:

```
o=- 2890844526 2890844526 IN IP4 host.example.com
c=IN IP4 host.example.com
```

The following group attribute declares which MDs are included in the
multiplexed MD: mid:con1 and mid:con2 are the constituent MDs whose
TFs (from the application point of view) will be carried by the TF of
the first-designed MD, mid:bundle, which is the bundle MD. In order
to allow demultiplexing of the packets on the bundle TF, the
constituent MDs must use disjoint sets of payload types.

         a=group:BUNDLE bundle con1 con2

   This MD provides the application-level description of the audio TF.
   As in the previous example, it is the first audio m= line. It
   includes any attributes which apply to the audio media from the
   application point of view, including the payload type definitions.
   When interpreted by a supporting processor, the transport information
   is ignored. When interpreted by a processor that does not support
   bundling, the transport information sets up the transport association
   for the audio TF. But to avoid the overhead of preallocating TURN
   relays that will probably not be used, ICE relay candidates are not
   provided.

         m=audio 10000 RTP/AVP 0 8 97
         a=mid:con1
         a=rtcp-mux
         a=rtpmap:0 PCMU/8000
         a=rtpmap:8 PCMA/8000
         a=rtpmap:97 iLBC/8000
         a=candidate:0 1 UDP 2113601791 10.0.1.1 10000 typ host
         a=candidate:1 1 UDP 1694194431 198.51.100.32 51000 typ srflx

   This MD provides the application-level description of the video TF.
   As in the previous example, it is the first video m= line. It
   includes any attributes which apply to the video media from the
   application point of view. As in the audio MD, the association
   information is used only by a processor that does not support
   bundling.

         m=video 10002 RTP/AVP 31 32
         a=mid:con2
         a=rtcp-mux
         a=rtpmap:31 H261/90000
         a=rtpmap:32 MPV/90000
         a=candidate:0 1 UDP 2113601791 10.0.1.1 10002 typ host
         a=candidate:1 1 UDP 1694194431 198.51.100.32 51002 typ srflx

   This MD provides the transport information for the bundle TF,
   including any attributes which apply to the transport. We use RTCP
   multiplexing [RFC5761], so only one set of ICE candidates (and only
   one TURN relay) is needed for this MD.

   The MD is artificially given the media type "audio" (which is ugly,
   but it avoids rejection by SBCs) and it is placed after all of the
   constituent MDs so as to not affect their positions as "first audio
   MD", etc. The MD has a proto value of "bundle" to describe the packet
   traffic (which in general can be a mixture of RTP/SRTP/RTCP/SRTCP,
   SCTP, and DTLS). A single, dummy fmt value of 0 is used. The proto

value ensures that an answerer that does not support bundling will not
accept this MD.

The packets sent on this transport flow are the packets provided by
the applications for the constituent transport flows.

```
m=audio 10004 bundle 0
a=mid:bundle
a=candidate:0 1 UDP 2113601791 10.0.1.1 10004 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51004 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 31004 typ relay
```

If this SDP bundle is accepted, RTP provided by the application for
the audio TF will be sent from port 10004. RTP provided by the
application for the video TF will be also be sent from port 10004.

RTP that is received on port 10004 is interpreted according to the
payload type number. Since the payload type numbers used in the two
constituent transport flows are disjoint, incoming RTP packets can be
directed to the proper applications based on payload type number.

4.1.3. Answer from an Answerer that Supports Bundling

If the answerer supports SDP bundling, and desires to accept the
offered bundle and its constituent MDs, the answerer signals that it
accepts the SDP bundling by providing a matching group:BUNDLE
attribute in the answer. As always in offer/answer, the MDs in the
answer correspond to the MDs in the offer by ordinal position.

The answerer provides the necessary transport information for the
bundle MD. The answerer understands that MDs mid:con1 and mid:con2
are incorporated into MD mid:bundle, and ignores their transport
information. It accepts each constituent MD as part of the bundle by
providing an answer MD for each of them that specifies an attribute
"a=bundleaccept" and a port number of 0 (so intermediate entities see
the MD as being rejected).

Note that the constituent MDs, despite having zero port numbers, are
still incorporated in the "a=group:BUNDLE" attribute. This
contravenes [RFC5888] section 9.2, and so this proposal requires an
extension of RFC 5888.

```
o=- 2890844526 2890844526 IN IP4 answer.example.com
c=IN IP4 answer.example.com

a=group:BUNDLE bundle con1 con2
```

```
m=audio 0 RTP/AVP 0 8 97
a=mid:con1
a=bundleaccept
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000

m=video 0 RTP/AVP 31 32
a=mid:con2
a=bundleaccept
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000

m=audio 20000 bundle 0
a=mid:bundle
a=candidate:0 1 UDP 2113601791 10.0.2.1 20000 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.35 51090 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 32000 typ relay
```

4.1.4. Answer from an Answerer that Does Not Support Bundling

   SDP bundling allows for backward compatibility in case the answerer
   does not understand bundling. If the answerer does not understand
   bundling, it ignores the group attribute, and effectively sees the
   offer as:

```
o=- 2890844526 2890844526 IN IP4 host.example.com
c=IN IP4 host.example.com

m=audio 10000 RTP/AVP 0 8 97
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=candidate:0 1 UDP 2113601791 10.0.1.1 10000 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51000 typ srflx

m=video 10002 RTP/AVP 31 32
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=candidate:0 1 UDP 2113601791 10.0.1.1 10002 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51002 typ srflx

m=audio 10004 bundle 0
```

```
         a=candidate:0 1 UDP 2113601791 10.0.1.1 10004 typ host
         a=candidate:1 1 UDP 1694194431 198.51.100.32 51004 typ srflx
         a=candidate:2 1 UDP 1006633215 10.0.100.1 31004 typ relay
```

If the answerer wishes to accept the first audio and video streams, it assembles this answer:

```
         o=- 2890844526 2890844526 IN IP4 answer.example.com
         c=IN IP4 answer.example.com
```

The absence of the group attribute informs the offerer that bundling was rejected.

The audio MD is accepted. Transport information is provided, including ICE candidates.

```
         m=audio 20000 RTP/AVP 0 8 97
         a=rtcp-mux
         a=rtpmap:0 PCMU/8000
         a=rtpmap:8 PCMA/8000
         a=rtpmap:97 iLBC/8000
         a=candidate:0 1 UDP 2113601791 10.0.2.1 20000 typ host
         a=candidate:1 1 UDP 1694194431 198.51.100.128 52000 typ srflx
```

The video MD is accepted. Transport information (using a different port) is provided.

```
         m=audio 20002 RTP/AVP 31 32
         a=rtcp-mux
         a=rtpmap:31 H261/90000
         a=rtpmap:32 MPV/90000
         a=candidate:0 1 UDP 2113601791 10.0.2.1 20002 typ host
         a=candidate:1 1 UDP 1694194431 198.51.100.128 52002 typ srflx
```

The bundle MD is rejected by the answerer because the proto value is "bundle", and the answerer does not implement it.

```
         m=audio 0 bundle 0
```

Because the group attribute is not present in the response, the offerer knows that the answerer does not support bundling (or does not want to consider the offered bundle). The offerer knows that the answerer wants to establish one audio TF and one video TF, and formally, that has been done. But if transport requires ICE relay candidates on the offerer's side, the offerer must send an updated offer containing those ICE candidates for the constituent MDs:

```
        o=- 2890844526 2890844527 IN IP4 host.example.com
        c=IN IP4 host.example.com
```

No group attribute is included, to ensure that this update only
revises transport attributes, and does not trigger bundle-supporting
behavior if the answering entity has changed in the meantime.

Provide ICE relay candidates for the audio MD.

```
        m=audio 10000 RTP/AVP 0 8 97
        a=mid:con1
        a=rtcp-mux
        a=rtpmap:0 PCMU/8000
        a=rtpmap:8 PCMA/8000
        a=rtpmap:97 iLBC/8000
        a=candidate:0 1 UDP 2113601791 10.0.1.1 10000 typ host
        a=candidate:1 1 UDP 1694194431 198.51.100.32 51000 typ srflx
        a=candidate:2 1 UDP 1006633215 10.0.100.1 31000 typ relay
```

Provide a separate TURN relay for the video MD.

```
        m=video 10002 RTP/AVP 31 32
        a=mid:con2
        a=rtcp-mux
        a=rtpmap:31 H261/90000
        a=rtpmap:32 MPV/90000
        a=candidate:0 1 UDP 2113601791 10.0.1.1 10002 typ host
        a=candidate:1 1 UDP 1694194431 198.51.100.32 51002 typ srflx
        a=candidate:2 1 UDP 1006633215 10.0.100.1 31002 typ relay
```

The bundle MD must still be listed, but it is disabled.

```
        m=audio 0 bundle 0
        a=mid:bundle
```

The answerer provides the same answer as before.

The ICE renegotiation proceeds, the transport associations are
established, and RTP flows.

4.1.5. Fast-Start Offer

The baseline procedure requires the offerer to update its offer when
it discovers that the answerer does not support SDP bundling if TURN
relays are needed to support the constituent MDs. The offerer can
avoid this delay by providing TURN relays for the constituent MDs as
well as for the bundle MD. The penalty is that the offerer must

preallocate TURN relays for both the constituent MDs as well as the
bundle MD.

```
o=- 2890844526 2890844526 IN IP4 host.example.com
c=IN IP4 host.example.com

a=group:BUNDLE bundle con1 con2

m=audio 10000 RTP/AVP 0 8 97
a=mid:con1
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=candidate:0 1 UDP 2113601791 10.0.1.1 10000 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51000 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 31000 typ relay

m=video 10002 RTP/AVP 31 32
a=mid:con2
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=candidate:0 1 UDP 2113601791 10.0.1.1 10002 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51002 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 31002 typ relay

m=audio 10004 bundle 0
a=mid:bundle
a=rtcp-mux
a=candidate:0 1 UDP 2113601791 10.0.1.1 10004 typ host
a=candidate:1 1 UDP 1694194431 198.51.100.32 51004 typ srflx
a=candidate:2 1 UDP 1006633215 10.0.100.1 31004 typ relay
```

If the answerer understands bundling and accepts the bundle, it
accepts the constituent MDs within the bundle (with "a=bundleaccept"
and port 0) and accepts the bundle MD. If the answerer does not
understand bundling, it accepts the constituent MDs and rejects the
bundle MD. In either case, ICE relay candidates are in place and ICE
negotiation proceeds.

4.2. Two Audio Streams and Two Video Streams

In this example, a presentation involves four media roles: the
speaker's audio, the floor microphone, the video of the speaker, and
the video of the speaker's slides. We use separate MDs for each
media stream because each TF has a different role; the application
will handle each of them in distinctly different ways.

```
        o=- 2890844526 2890844526 IN IP4 host.example.com
        c=IN IP4 host.example.com

        a=group:BUNDLE b c1 c2 c3 c4

        m=audio 10000 RTP/AVP 0 8 97
        a=mid:c1
        a=label:speaker-audio
        a=rtcp-mux
        a=rtpmap:0 PCMU/8000
        a=rtpmap:8 PCMA/8000
        a=rtpmap:97 iLBC/8000
```

Note that different constituent MDs must use different payload types
(even for the same codec), because incoming RTP is demultiplexed based
on payload type.

```
        m=audio 10002 RTP/AVP 98 99 100
        a=mid:c2
        a=label:floor-mic
        a=rtcp-mux
        a=rtpmap:98 PCMU/8000
        a=rtpmap:99 PCMA/8000
        a=rtpmap:100 G722

        m=video 10004 RTP/AVP 101 102
        a=mid:c3
        a=label:speaker-video
        a=rtcp-mux
        a=rtpmap:101 H261/90000
        a=rtpmap:102 MPV/90000

        m=video 10006 RTP/AVP 103 104
        a=mid:c4
        a=label:slides
        a=rtcp-mux
        a=rtpmap:103 H261/90000
        a=rtpmap:104 MPV/90000

        m=multipart 10008 bundle 0
        a=mid:b
```

4.3. **Virtual Classroom with One Audio Stream, Two Video Streams, and a Group of Video Streams**

   This example is the teacher's connection to a virtual classroom
   server. The media descriptions are tagged using the "content"
   attribute. [RFC4796] The media comprises:

   1. 1. one audio channel, for sending the teacher's voice and
       receiving the voice of a selected student

   2. 2. one video channel, for sending the teacher's presentation

   3. 3. one video channel, for sending the teacher's face

   4. 4. one video channel, for receiving a dynamically varying set of
       students' faces

   The fourth TF (for students' faces) contains a large and dynamically
   varying set of video captures. These can be handled by a single TF
   because they all have essentially similar roles -- the application
   will process them as a set. As Adam Roach would say, "no control
   surfaces are necessary to talk about and/or manipulate the individual
   streams". In particular, this allows a large number of captures to
   be handled without mentioning them in the SDP, at the expense of not
   allowing the SDP to describe any of them individually. Similarly,
   the number of captures can vary without having to renegotiate the
   SDP.

   (In contrast, the third TF (the teacher's face) is a separate TF
   because it is processed in a different role than that of the
   students' faces.)

   In unbundled usage, there would be one transport association for the
   fourth TF. Incoming RTP from that association would be demultiplexed
   by the application based on the SSRC values, which would be unique
   for each student. With bundling, once the single transport TF is
   demultiplexed based on the RTP payload type, packets destined for the
   fourth TF (index = 3) would be further demultiplexed by their SSRC
   values. The demultiplexing by SSRC is considered to be an
   application layer function in the context of SDP bundling.

   The offered SDP is:

           o=- 2890844526 2890844526 IN IP4 host.example.com
           c=IN IP4 host.example.com

           a=group:BUNDLE b c1 c2 c3 c4

The audio channel is send/receive.

```
m=audio 10000 RTP/AVP 0 8 97
a=mid:c1
a=label:speaker-audio
a=content:speaker
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
```

The teacher's face and presentation are send-only.

```
m=video 10002 RTP/AVP 103 104
a=mid:c2
a=label:speaker-video
a=content:speaker
a=sendonly
a=rtcp-mux
a=rtpmap:103 H261/90000
a=rtpmap:104 MPV/90000

m=video 10004 RTP/AVP 105 106
a=mid:c3
a=label:presentation
a=content:slides
a=sendonly
a=rtcp-mux
a=rtpmap:105 H261/90000
a=rtpmap:106 MPV/90000
```

The student video input is receive-only and is limited to 24 simultaneous SSRCs.

```
m=video 10006 RTP/AVP 107 108
a=mid:c4
a=label:student-thumbnails
a=recvonly
a=max-recv-ssrc:* 24
a=rtcp-mux
a=rtpmap:107 H261/90000
a=rtpmap:108 MPV/90000

m=audio 10000 RTP/AVP 127
a=mid:b
a=rtcp-mux
a=rtpmap:127 bundle
a=candidate:0 1 UDP 2113601791 10.0.1.1 10000 typ host
```

```
        a=candidate:1 1 UDP 1694194431 198.51.100.32 51000 typ srflx
```

4.4. One Audio Stream and Two SCTP Streams

   This example contains one audio MD and two SCTP MDs, which are used
   for Webrtc datachannels.

```
        o=- 2890844526 2890844526 IN IP4 host.example.com
        c=IN IP4 host.example.com

        a=group:BUNDLE bundle con1 con2 con3

        m=audio 10000 RTP/AVP 0 8 97
        a=mid:con1
        a=rtcp-mux
        a=rtpmap:0 PCMU/8000
        a=rtpmap:8 PCMA/8000
        a=rtpmap:97 iLBC/8000
```

   These MDs provides the the SCTP TFs. Because packets are not
   encapsulated, the two SCTP TFs must use different (nominal) SCTP ports
   to allow their packets to be distinguished.

```
        m=application 10002 DTLS/SCTP 5000
        a=mid:con2
        a=sctpmap:5000 webrtc-datachannel 16

        m=application 10004 DTLS/SCTP 5001
        a=mid:con3
        a=sctpmap:5001 webrtc-datachannel 16

        m=audio 10006 bundle 0
        a=mid:bundle
```

5. Syntax and Semantics

   TBD (Here lies the real description.)

5.1. Constructing an Offer

   TBD

5.2. Constructing an Answer

   TBD

5.3. Offer/Answer Considerations

   TBD

5.4. RTCP, SSRC, and RTP Sessions

   TBD

5.5. ICE considerations

   TBD

5.6. Encryption considerations

   TBD Need to discuss here how the encryption associations are set up.
   For SRTP/SRTCP, it would be possible to have either one association
   for all multiplexed streams, or one for each constituent MD, because
   SRTP preserves the PTs. (Have to verify that and check whether SRTCP
   preserves SSRCs.) But SCTP-over-DTLS can't be demultiplexed before
   it's decrypted, so there can be only one DTLS crypto association.

6. Compatibility Considerations

6.1. Backward Compatibility during Offer/Answer

   TBD

6.2. Backward Compatibility with Existing Devices

   TBD

7. Design Features and Comparison

   Key:

         x = feature present in proposal
         - = feature not present in proposal
         . = feature not discussed in proposal
       N/A = feature is not relevant because of another feature choice

                 worley-sdp-bundle-06
                 | ietf-mmusic-sdp-bundle-negotiation-03 (BUNDLE)
                 | | holmberg-mmusic-sdp-mmt-negotiation-00 (MMT)
                 | | | alvestrand-one-rtp-02 (TOGETHER)
                 | | | | ejzak-mmusic-bundle-alternatives-01
                 | | | | | Roach alternative 1a
                 | | | | | |(roach-mmusic-mlines-00)
                 | | | | | | Roach alternative 1b

```
                        |  |  |  |  |  |  |  Roach alternative 2
                        |  |  |  |  |  |  |  |   westerlund-avtcore-
                        |  |  |  |  |  |  |  |  |transport-multiplexing-05
                        |  |  |  |  |  |  |  |  |(SHIM)
                        V  V  V  V  V  V  V  V  V
   MD grouping:
       one              -  -  -  -  -  -  X  -  -
       per type         -  -  -  -  -  X  -  -  -
       none             x  x  x  x  x  -  -  x  x

   Separate bundle MD:
       no               -  x  -  x  x  x  x  x  x
       m=anymedia       -  -  x  -  -  -  -  -  -
       m=audio          x  -  -  -  -  -  -  -  -
       m=multipart      -  -  -  -  -  -  -  -  -

   Immediate update:
       none             -  -  x  x  x  x  x  x  x
       for support      -  x  -  -  -  -  -  -  -
       for compat.      x  -  -  -  -  -  -  -  -

   Constituent MD ports after establishment:
       N/A              -  -  -  -  -  X  X  -  -
       same             -  x  -  x  -  -  -  x  x
       different        -  -  -  -  -  -  -  -  -
       null             -  -  -  -  -  -  -  -  -
       rejected         x  -  x  -  x  -  -  -  -

   Bundle MD payload types:
       N/A              x  -  -  -  -  X  X  -  -
       one MD           -  x  -  x  .  -  -  x  x
       all MDs          -  -  x  -  .  -  -  -  -
       one value        -  -  -  -  .  -  -  -  -

   Constituent MD payload types:
       N/A              -  -  -  -  -  X  X  -  -
       overlapping      -  -  -  -  .  -  -  x  x
       distinct         x  x  x  x  .  -  -  -  -

   Demultiplexing based on:
       N/A              -  -  -  -  -  X  X  -  -
       PT               x  x  x  x  .  -  -  x  -
       encap.           -  -  -  -  .  -  -  -  x

   Rejection of bundle MD based on:
       N/A              -  x  -  x  x  x  x  x  x
       media type       -  -  x  -  -  -  -  -  -
       proto            x  -  -  -  -  -  -  -  -
```

```
      codec       -   -   -   -   -   -   -   -   -
```

Addresses/ports in constituent MDs in offer:

```
    N/A           -   -   -   -   -   x   x   -   -
    NA,ZP         -   -   -   -   -   -   -   -   -
    NA,NZP        -   -   -   -   x   -   -   -   -
    RA,ZP         -   -   -   -   -   -   -   -   -
    RA,NZP,U      x   x   x   -   x   -   -   -   -
    RA,NZP,S      -   x   -   x   -   -   -   x   x
```

NA = null address, RA = real address

ZP = zero port, NZP = non-zero port

U = unique ports, S = shared port

7.1. Aggregation of Constituent Media Descriptions

Are the constituent media descriptions combined into grouped media descriptions?

o   All media are combined into one media description.

o   All media of each media type are combined into one media description (which has that type).

o   Each constituent media description is separate in the session description.

This proposal does not aggregate constituent MDs so that attributes can be provided directly for each constituent MD.

7.2. Presence of a Bundle Media Description and Its Media Type

Is there a separate bundle media description, and if so, what media type does it have?

o   There is no separate bundle media description.

o   There is a separate bundle media description of type "anymedia".

o   There is a separate bundle media description of type "audio".

o   There is a separate bundle media description of type "multipart".

This proposal has a separate bundle MD so that attributes can be provided for the bundle MD independently of any constituent MD.

7.3. Immediate Update

Is an immediate updated offer/answer used during session
establishment?

o No.

o Yes, when establishing a session using bundling.

o Yes, when establishing a session not using bundling.

This proposal can require immediate updates for non-bundle-supporting
answers to provide ICE TURN candidates, if the offerer has not
preallocated them.

7.4. Effective Media Description Ports after Session Establishment

What are the effective port numbers in MDs after the session is
established?

o There are not multiple media descriptions because constituent MDs
   are combined.

o Port numbers in the MDs are the same.

o Port numbers in the MDs are different.

o All but one MD have null addresses.

o All but one MD have a zero port number (and thus are formally
   rejected).

This proposal uses a zero port number for constituent MDs to prevent
intermediate entities from expecting to see media for the constituent
transport associations.

7.5. Payload Types in the Bundled Media Description

What payload types are listed for the bundled MD?

o There is no MD describing the bundle as a whole.

o The bundle MD lists the payload types of one constituent MD.

o The bundle MD lists the payload types of all constituent MDs.

o The bundle MD lists one payload type.

This proposal uses a distinct proto value that does not use payload
type numbers.

7.6. Relationship of Payload Types of Constituent Media Descriptions

   What is the relationship between the payload types of the constituent
   MDs?

   o There are not multiple media descriptions because constituent MDs
      are combined.

   o Different constituent MDs may have overlapping payload type
      numbers.

   o Different constituent MDs may not have overlapping payload type
      numbers.

   This proposal requires constituent MDs to use distinct payload types
   in order to demultiplex the constituent MDs.

7.7. Basis of Demultiplexing

   What is the basis for the demultiplexing of RTP?

   o Demultiplexing is not done because incoming RTP is not attributed
      to specific constituent MDs (possibly because constituent MDs are
      combined).

   o Demultiplexing is done based on payload type numbers.

   o Demultiplexing is done based on data carried in an encapsulation.

   This proposal does demultiplexing based on information in the
   encapsulated payload format.

7.8. Basis of Rejection of the Bundle MD

   We must ensure that the bundle MD is rejected by non-supporting
   endpoints. What method is used to ensure rejection?

   o There is no bundle MD.

   o The bundle MD uses a special media type value.

   o The bundle MD uses a special proto value.

   o The bundle MD uses (only) a special codec name.

   This proposal uses a distinct proto type in the bundle media
   description to ensure that answerers that do not implement bundling
   reject this MD.

7.9. How Constituent MDs Are Offered

There are a number of alternative ways that the offerer can configure
the constituent media descriptions.

Method	1	2	3	4	5	6	7	8
Coded in chart as	NA, ZP	NA,N ZP	RA,Z P	RA,NZ P,U	RA,NZ P,U	RA,N ZP,U	RA,N ZP,U	RA,N ZP,S
Offered address	null	null	real	real	real	real	real	real
Offered port	zero	non-zero	zero	non-zero, unique	non-zero, unique	non-zero, unique	non-zero, unique	non-zero, shared
TURN candidates?	N/A	N/A	N/A	no	yes	no	yes	yes
Supporting answerer accepts?	yes	yes	yes	yes	yes	no	no	yes
Update needed for supporting answerer?	no	no	possibly	yes	yes	no	no	no
Non-supporting answerer accepts?	no	probably	no	yes	yes	yes	yes	yes
Update needed for non-supporting answerer?	yes	yes	yes	yes	no	yes	no	no
Disadvantages	ACE	CD	ABCE	BC	BG	C	G	BFG

Offered address This is the address offered for the MD. There are
   two choices, a null address (0.0.0.0 for IPv4 or "invalid." for
   IPv6 or mixed IPv4/IPv6) or a real address of the offerer.

Offered port This is the port that is offered. It can be either
   non-zero or zero (which indicates a stream that is not being
   offered). If the offered port is non-zero and the offered address
   is real, the port can be either unique, or shared with the other
   media descriptions in the bundle.

TURN candidates? If the offered address is real and the offered port
   is non-zero, are TURN candidates for the address provided (if they
   are needed)?

Supporting answerer accepts? If the answerer supports bundling, does
   it accept the constituent media description (with a non-zero port
   in the answer)? If not, the answerer must have some other method
   of indicating acceptance of the constituent of the bundle.

Update needed for supporting answerer? Is an SDP update needed to
   complete session setup if the answerer supports bundling? If an
   update is needed, it is needed to inform intermediaries that there
   will be no media sent based on the connection information in this
   media description; the update is not needed to establish
   communications and does not delay the application.

Non-supporting answerer accepts? If the answerer does not support
   bundling, does it accept this media description (without a further
   update)?

Update needed for non-supporting answerer? Is an SDP update needed
   to complete session setup if the answerer does not support
   bundling? If an update is needed, in all cases, it is needed to
   establish communication.

Flaws The disadvantages of each alternative:

   A   Media descriptions that are rejected (have zero ports) are
         not allowed to be members of a group (in the
         offer).[RFC5888]

   B   An SDP update is needed for a supporting answerer to prevent
         intermediaries from timing out the multimedia session.

   C   An SDP update is needed for a non-supporting answerer to
         establish communications.

   D   Although a media description with a non-zero port but a null
         address is formally offered (although shown as on-hold via
         the old method), it is possible that the answerer will not
         consider it to be offered and many not accept it even if it
         otherwise wood.

       E   If the offered port is zero, the media description is not
              formally offered and the answerer should not accept it.

       F   SDP offers with multiple media descriptions that use the
              same port number (for the same real address) may be rejected
              by intermediaries.

       G   A TURN relay must be allocated for the constituent media
              description before the offer is sent.

   In my estimation, the worst disadvantages are A (zero port in offer),
   E (acceptance of offer with zero port), and F (duplicate port
   numbers), because they involve violations of [RFC4566] or are known
   to trigger limitations of large numbers of intermediate devices.
   Disadvantage D (offering a MD with a null address) is nearly as
   severe, as we expect it to cause undesired behavior in many non-
   supporting answerers. Disadvantages C (update needed to communicate
   with non-supporting answerer) and G (TURN relay must be preallocated)
   are moderate, and disadvantage B (updated needed to prevent
   intermediaries from timing out) is the least severe (because it never
   delays the establishment of communication).

   Applying these priorities to the possible solutions, methods 6 and 7
   (offer real address, non-zero unique port, with/without TURN
   candidates, answer has zero port for constituent MDs) are tied for
   the best choices, with the choice made based on the relative
   importance of minimizing preallocation of TURN relays compared to
   quickly establishing communication with non-supporting answerers.

8. Demultiplexing Considerations

   This section discusses the constraints regarding demultiplexing
   datagrams from multiple protocols that are presented on one transport
   flow. This is an expansion of the analysis in [RFC5764] section
   5.1.2.

   The first octets of datagrams generated by particular protocols are:

Protocol	First octet	Second octet	Third octet	Fourth octet
STUN	0x00, 0x01	0x00, 0x01		
RTP	0x80 to 0xBF	0x00 to 0xC7, 0xCD to 0xFF		

```
| RTCP          | 0x80 to   | 0xC8 to 0xCC |           |            |
|               | 0xBF      |              |           |            |
|               |           |              |           |            |
| RTP/RTCPv3    | 0xC0 to   |              |           |            |
|               | 0xFF      |              |           |            |
|               |           |              |           |            |
| DTLS          | 0x14 to   | 0x03         | 0x03      |            |
|               | 0x17      |              |           |            |
|               |           |              |           |            |
| SCTP          | source    | source port  | dest.     | dest. port |
|               | port high | low          | port high | low        |
+-------------+-----------+-------------+----------+------------+
```

TBD RFC 5764 specifies that the first octet of a DTLS packet is in
the range 0x14 to 0x3F. RFC 5246 and RFC 6374 together specify the
first octet is a "ContentType", with the range 0x14 to 0x17. Are
additional octet values reserved for expansion? What is the range
that should be reserved in practice?

The most generalized stack of protocols we consider is this:

```
Layer 6:  ... application interfaces ...
          |||       |||       |||       |||
           V         V         V         V
           |         |         |         |
           |         |         |         |
Layer 5:   |         |         |        SCTP
           |         |         |         |
           |         |         |         |
          RTP       SRTP       |         |
Layer 4:  RTCP      SRTCP     SCTP      DTLS    [ STUN ]
             \       |         |         |       /
              --------------- ----------------
                          V
                          |
                          |
Layer 3:            [ encapsulation     STUN ]
                    [         \             /   ]
                    [          ---- ---------    ]
                    [               V            ]
                                    |
                                    |
Layer 2:                    [ DTLS              STUN ]
                            [    \                 /   ]
                            [     ------- ---------     ]
                            [               V           ]
                                            |
```

```
                                        |
   Layer 1:                          [ TURN ]
                                        |
                                        |
   Layer 0:                            UDP
```

Layer 0: UDP This is the base layer of this analysis, where packets
     are carried by UDP.

Layer 1: TURN (optional) If a packet arrives from a TURN relay for
     which we are a client, the TURN encapsulation must be removed
     before further processing. This need can be detected because the
     packet arrives from the client-facing address/port of a TURN
     server of which this entity is a client.

Layer 2: DTLS applied to the bundle transport flow
(optional)
     If DTLS is applied to the bundle transport flow as a whole, that
     use of DTLS will have been negotiated. However, before
     deciphering, STUN packets need to be separated. STUN packets can
     be distinguished from DTLS packets by their first or second
     octets.

Layer 3: Decapsulation (depending on the bundling method If the
     bundling method uses encapsulation, decapsulation is done at this
     point in the protocol stack. However, before decapsulating, STUN
     packets need to be separated. The encapsulation method must allow
     encapsulated packets to be distinguished from STUN packets, if
     STUN packets have not been demultiplexed at layer 2 (due to DTLS
     encryption of the entire transport stream).

     All proposed encapsulation techniques ensure the first octet of
     the encapsulation is in the range allowed for the first octet of
     RTP, 0x80 to 0xBF, and thus is distinguishable from STUN.

Layer 4: Core demultiplexing At this layer, most of the protocols
     are demultiplexed. RTP/SRTP, SRTP/SRTCP, DTLS, and STUN are
     distinguished by the values of the first two octets of the packet.
     SRTP/SRTCP is distinguished from RTP/RTCP by negotiation with the
     other endpoint -- SRTP/SRTCP is never multiplexed with RTP/RTCP.

     SCTP can be distinguished from the other protocols if the other
     endpoint agrees to use only SCTP port numbers in the range 0x4000
     to 0x7FFF. This restriction can be specified here, because this
     limitation is only needed when bundling is implemented, which
     happens only when when both endpoints support bundling. (This
     particular range is chosen to avoid collision with a future RTP/
     RTCP version 3. The range of SCTP ports can be chosen arbitrarily

because the SCTP ports are not used to route packets to this
entity, as they are encapsulated in UDP.)

Layer 5: SCTP over DTLS If the layer 4 protocol is DTLS, the
protocol above it will always be SCTP.

Layer 6: Application interface demultiplexing The method used to
demultiplex the various application interface streams varies
depending whether encapsulation is used and if not, on the layer 4
/5 protocol. If the layer 4 protocol is encapsulated, the
encapsulation determines which constituent media description is to
be assigned to this packet, and then application demultiplexing is
done as normal for for that particular media description.
Otherwise, the constituent media description must be determined by
a method that is specific to the layer 4/5 protocol:

RTP/SRTP An RTP/SRTP packet is assigned to the constituent media
description which specifies its payload type number. (If
encapsulation is not used, the constituent media
descriptions must specify distinct payload type numbers.)

RTCP/SRTCP An RTCP/SRTCP sub-packet is dispatched to the
application which receives the RTP media stream containing
the SSRC that is carried in the RTCP sub-packet.

SCTP An SCTP packet is assigned to the constituent media
description which specifies its destination port number.
(If encapsulation is not used, the constituent media
descriptions must specify distinct SCTP port numbers.)

9. Security Considerations

If an SBC wishes to prevent positively the transport of certain media
types or codecs, and enforces that by examining the content of RTP
packets, the use of kumquat encoding may defeat the examination.

TBD

10. IANA Considerations

TBD

11. Acknowledgments

Many people have provided input for this proposal regarding both the
technical aspects and the organization of the presentation. Chief
among them are the authors of the predecessor proposals
([I-D.alvestrand-one-rtp] (TOGETHER),

[I-D.holmberg-mmusic-sdp-mmt-negotiation] (MMT), and
[I-D.ietf-mmusic-sdp-bundle-negotiation] (BUNDLE)): Harald
Alvestrand, Jonathan Lennox, and Christer Holmberg. In addition,
input was provided by Charles Eckel, Andrew Hutton, Cullen Jennings,
Hadriel Kaplan, Paul Kyzivat, Adam Roach, and Robert Sparks.

12. Revision History

Note to RFC Editor: Please remove this section before publication.

12.1. draft-worley-sdp-bundle-00

Initial version.

12.2. Changes from draft-worley-sdp-bundle-00 to draft-worley-sdp-bundle-01

Thoroughly revise the text and structure of the document.

12.3. Changes from draft-worley-sdp-bundle-01 to draft-worley-sdp-bundle-02

Heavily revise Terminology regarding media flows.

Revise Desiderata, including adding that multiple separate bundles
must be possible, and noninterference with ICE negotiation.

Add section on ICE considerations.

Change "fusion" to "bundle".

Use a=rtcp-mux in examples to be more realistic (and to shorten the
examples).

Correct the use of ICE in answers; ICE candidates are not provided if
an offered MD does not contain ICE candidates.

Add an example of a fast-start offer.

12.4. Changes from draft-worley-sdp-bundle-02 to draft-worley-sdp-bundle-03

Add design comparison Section 7.

Use bibxml references.

Add DES C9, regarding continued usage of transcoding facilities
offered by intermediate entities.

12.5.   **Changes from draft-worley-sdp-bundle-03 to draft-worley-sdp-bundle-04**

Add demultiplexing considerations Section 8.

12.6.   **Changes from draft-worley-sdp-bundle-04 to draft-worley-sdp-bundle-05**

Change recommendation for SCTP port numbers from 0xC000-0xFFFF to 0x4000-0x7FFF to avoid collision with a future RTP/RTCP version 3.

Add the transport flow index to the KUMQUAT encapsulation.

Add section on choices for offering constituent MDs Section 7.9. Revise the examples to show offering "real address, non-zero port, no ICE candidates".

Add note to DES C9 (support intermediate transcoding facilities) saying that intermediate transcoding facilities are not expected to be very useful, given that encryption will be the normal use case.

Add an exampleSection 4.4 with SCTP MDs.

Add a section for encryption considerations.Section 5.6

Revise generalized demultiplexing diagram to make explicit the optional RTP encapsulation layer.

Update comparison chartSection 7 for draft-ejzak-mmusic-bundle-alternatives-01[I-D.ejzak-mmusic-bundle-alternatives] and draft-westerlund-avtcore-transport-multiplexing-05[I-D.westerlund-avtcore-transport-multiplexing].

Update comparison chartSection 7 to discuss alternative address/port/candidate policies for offering constituent MDs.

12.7.   **Changes from draft-worley-sdp-bundle-05 to draft-worley-sdp-bundle-06**

Add desideratum that we want the RTP media to be visible on the wire as SRTP.

Remove the encapsulation.

Change acceptance-within-bundle in an answer to use a zero port number with a=bundleaccept attribute. Revise the the table Section 7.9 to include this method.

Change c= lines in examples to use host names, which is what would be
done in a dual-stack environment. (ICE candidates carry the actual
addresses used.)

Add TURN ICE candidates to the examples with ICE candidates, as
described by the existing text.

13. References

13.1. Normative References

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

[RFC3264]   Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model
            with Session Description Protocol (SDP)", RFC 3264, June
            2002.

[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.

[RFC5888]   Camarillo, G. and H. Schulzrinne, "The Session Description
            Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

13.2. Informative References

[RFC2327]   Handley, M. and V. Jacobson, "SDP: Session Description
            Protocol", RFC 2327, April 1998.

[RFC3725]   Rosenberg, J., Peterson, J., Schulzrinne, H., and G.
            Camarillo, "Best Current Practices for Third Party Call
            Control (3pcc) in the Session Initiation Protocol (SIP)",
            BCP 85, RFC 3725, April 2004.

[RFC4796]   Hautakorpi, J. and G. Camarillo, "The Session Description
            Protocol (SDP) Content Attribute", RFC 4796, February
            2007.

   [RFC4960]   Stewart, R., "Stream Control Transmission Protocol", RFC
               4960, September 2007.

   [RFC5761]   Perkins, C. and M. Westerlund, "Multiplexing RTP Data and
               Control Packets on a Single Port", RFC 5761, April 2010.

   [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.

   [I-D.alvestrand-mmusic-msid]
               Alvestrand, H., "Cross Session Stream Identification in
               the Session Description Protocol", draft-alvestrand-
               mmusic-msid-02 (work in progress), December 2012.

   [I-D.alvestrand-one-rtp]
               Alvestrand, H., "SDP Grouping for Single RTP Sessions",
               draft-alvestrand-one-rtp-02 (work in progress), September
               2011.

   [I-D.ejzak-mmusic-bundle-alternatives]
               Ejzak, R., "Alternatives to BUNDLE", draft-ejzak-mmusic-
               bundle-alternatives-01 (work in progress), February 2013.

   [I-D.holmberg-mmusic-sdp-mmt-negotiation]
               Holmberg, C., Alvestrand, H., and J. Lennox, "Multiplexed
               Media Types (MMT) Using Session Description Protocol (SDP)
               Port Numbers", draft-holmberg-mmusic-sdp-mmt-
               negotiation-00 (work in progress), October 2012.

   [I-D.ietf-avtcore-multi-media-rtp-session]
               Westerlund, M., Perkins, C., and J. Lennox, "Multiple
               Media Types in an RTP Session", draft-ietf-avtcore-multi-
               media-rtp-session-02 (work in progress), February 2013.

   [I-D.ietf-mmusic-sdp-bundle-negotiation]
               Holmberg, C., Alvestrand, H., and C. Jennings,
               "Multiplexing Negotiation Using Session Description
               Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-
               bundle-negotiation-03 (work in progress), February 2013.

   [I-D.jennings-mmusic-media-req]
               Jennings, C., Uberti, J., and E. Rescorla, "Requirements
               from various WG for MMUSIC", draft-jennings-mmusic-media-
               req-00 (work in progress), February 2013.

   [I-D.roach-mmusic-mlines]

          Roach, A., "Thoughts on syntax for representing multiple
          media streams", draft-roach-mmusic-mlines-00 (work in
          progress), January 2013.

   [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-05 (work in progress), February
          2013.

   [I-D.worley-service-example]
          Worley, D., "Session Initiation Protocol Service Example
          -- Music on Hold", draft-worley-service-example-11 (work
          in progress), February 2013.

Author's Address

   Dale R. Worley
   Ariadne Internet Services, Inc.
   738 Main St.
   Waltham, MA 02451
   US

   Phone: +1 781 647 9199
   Email: worley@ariadne.com