

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2013

H. Alvestrand
Google
October 15, 2012

Cross Session Stream Identification in the Session Description Protocol
draft-alvestrand-mmusic-msid-01

Abstract

This document specifies a grouping mechanism for RTP media streams that can be used to specify relations between media streams within different RTP sessions.

This mechanism is used to signal the association between the RTP concept of SSRC and the WebRTC concept of "media stream" / "media stream track" using SDP signalling.

This document is an input document for discussion. It should be discussed in the MMUSIC WG list, mmusic@ietf.org.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 18, 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

1. Introduction	3
1.1. Structure Of This Document	3
1.2. Why A New Mechanism Is Needed	3
1.3. Application to the WEBRTC MediaStream	4
2. The Msid Mechanism	4
3. The Msid-Semantic Attribute	5
4. Applying Msid to WebRTC Media Streams	6
4.1. Handling of non-signalled tracks	7
5. IANA Considerations	8
6. Security Considerations	9
7. Acknowledgements	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Appendix A. Design considerations, open questions and and alternatives	10
Appendix B. Change log	11
B.1. Changes from rtcweb-msid-00 to -01	11
B.2. Changes from rtcweb-msid-01 to -02	11
B.3. Changes from rtcweb-msid-02 to mmusic-msid-00	11
B.4. Changes from mmusic-msid-00 to -01	12
Author's Address	12

1. Introduction

1.1. Structure Of This Document

This document extends the SSRC grouping framework [RFC5888] by adding a new grouping relation that can cross RTP session boundaries.

Section 1.2 gives the background on why a new mechanism is needed.

Section 2 gives the definition of the new mechanism.

Section 4 gives the application of the new mechanism for providing necessary semantic information for the association of MediaStreamTracks to MediaStreams in the WebRTC API .

1.2. Why A New Mechanism Is Needed

When media is carried by RTP [RFC3550], each RTP media stream is distinguished inside an RTP session by its SSRC; each RTP session is distinguished from all other RTP sessions by being on a different transport association (strictly speaking, 2 transport associations, one used for RTP and one used for RTCP, unless RTCP multiplexing [RFC5761] is used).

There exist cases where an application using RTP and SDP needs to signal some relationship between RTP media streams that may be carried in either the same RTP session or different RTP sessions. For instance, there may be a need to signal a relationship between a video track in one RTP session and an audio track in another RTP session. In traditional SDP, it is not possible to signal that these two tracks should be carried in one session, so they are carried in different RTP sessions.

The SSRC grouping mechanism ("a=ssrc-group") [RFC5576] can be used to associate RTP media streams when those RTP media streams are part of the same RTP session. The semantics of this mechanism prevent the association of RTP media streams that are spread across different RTP sessions.

The SDP grouping framework [RFC5888] can be used to group RTP sessions. When an RTP session carries one and only one RTP media stream, it is possible to associate RTP media streams across different RTP sessions. However, if an RTP session has multiple RTP media streams, using multiple SSRCS, the SDP grouping framework cannot be used for this purpose.

There are use cases (some of which are discussed in [I-D.westerlund-avtcore-multiplex-architecture]) where neither of

these approaches is appropriate; In those cases, a new mechanism is needed.

In addition, there is sometimes the need for an application to specify some application-level information about the association between the SSRC and the group. This is not possible using either of the frameworks above.

1.3. Application to the WEBRTC MediaStream

The W3C WebRTC API specification [W3C.WD-webrtc-20120209] specifies that communication between WebRTC entities is done via MediaStreams, which contain MediaStreamTracks. A MediaStreamTrack is generally carried using a single SSRC in an RTP session (forming an RTP media stream. The collision of terminology is unfortunate.) There might possibly be additional SSRCs, possibly within additional RTP sessions, in order to support functionality like forward error correction or simulcast. This complication is ignored below.

In the RTP specification, media streams are identified using the SSRC field. Streams are grouped into RTP Sessions, and also carry a CNAME. Neither CNAME nor RTP session correspond to a MediaStream. Therefore, the association of an RTP media stream to MediaStreams need to be explicitly signalled.

The marking needs to be on a per-SSRC basis, since one RTP session can carry media from multiple MediaStreams, and one MediaStream can have media in multiple RTP sessions. This means that the [RFC4574] "label" attribute, which is used to label RTP sessions, is not usable for this purpose.

The marking needs to also carry the unique identifier of the RTP media stream as a MediaStreamTrack within the media stream; this is done using a single letter to identify whether it belongs in the video or audio track list, and the MediaStreamTrack's position within that array.

This usage is described in Section 4.

2. The Msid Mechanism

This document extends the Source-Specific Media Attributes framework [RFC5576] by adding a new "msid" attribute that can be used with the "a=ssrc" SDP attribute. This new attribute allows endpoints to associate RTP media streams that are carried in different RTP sessions, as well as allowing application-specific information to the association.

The value of the "msid" attribute consists of an identifier and optional application-specific data, according to the following ABNF [RFC5234] grammar:

```
; "attribute" is defined in RFC 4566.
; This attribute should be used with the ssrc-attr from RFC 5576.
attribute =/ msid-attr
msid-attr = "msid:" identifier [ " " appdata ]
identifier = 1*64 ("0".."9" / "a".."z" / "-")
appdata = 1*64 ("0".."9" / "a".."z" / "-")
```

An example MSID value for the SSRC 1234 might look like this:
a=ssrc:1234 msid:examplefoo v1

The identifier is a string of ASCII characters chosen from 0-9, a-z, A-Z and - (hyphen), consisting of between 1 and 64 characters. It MUST be unique among the identifier values used in the same SDP session. It is RECOMMENDED that is generated using a random-number generator.

Application data is carried on the same line as the identifier, separated from the identifier by a space.

The identifier uniquely identifies a group within the scope of an SDP description.

There may be multiple msid attributes on a single SSRC. There may also be multiple SSRCS that have the same value for identifier and application data.

Endpoints can update the associations between SSRCS as expressed by msid attributes at any time; the semantics and restrictions of such grouping and ungrouping are application dependent.

3. The Msid-Semantic Attribute

In order to fully reproduce the semantics of the SDP and SSRC grouping frameworks, a session-level attribute is defined for signalling the semantics associated with an msid grouping.

This OPTIONAL attribute gives the group identifier and its group semantic; it carries the same meaning as the ssrc-group-attr of RFC 5576 section 4.2, but uses the identifier of the group rather than a list of SSRC values.

The ABNF of msid-semantic is:

```
attribute =/ msid-semantic-attr
msid-semantic-attr = "msid-semantic:" " " identifier token
token = <as defined in RFC 4566>
```

The semantic field may hold values from the IANA registries "Semantics for the "ssrc-group" SDP Attribute" and "Semantics for the "group" SDP Attribute".

An example msid-semantic might look like this:

```
a=msid-semantic: examplefoo LS
```

4. Applying Msid to WebRTC Media Streams

This section creates a new semantic for use with the framework defined in Section 2, to be used for associating SSRCs representing media stream tracks with media streams as defined in [W3C.WD-webrtc-20120209].

The semantic token for WebRTC Media Streams is "WMS".

The value of the msid corresponds to the "id" attribute of a `MediaStream`. (note: as of Jan 11, 2012, this is called "label". The word "label" means many other things, so the same word should not be used.)

In a WebRTC-compatible SDP description, all SSRCs intending to be sent from one peer will be identified in the SDP generated by that entity.

The appdata for a `WebRTC MediaStreamTrack` consists of the track type and the track number; the track type is encoded as the single letter "a" (audio) or "v" (video), and the track number is encoded as a decimal integer with no leading zeros. The first track is track zero, and is identified as "a0" for audio, and "v0" for video.

If two different SSRCs have the same value for identifier and appdata, it means that these two SSRCs are both intended for the same `MediaStreamTrack`. This may occur if the sender wishes to use simulcast or forward error correction, or if the sender intends to switch between multiple codecs on the same `MediaStreamTrack`.

When an SDP description is updated, a specific msid continues to refer to the same media stream; an msid value MUST NOT be reused for another media stream within a `PeerConnection`'s lifetime.

The following are the rules for handling updates of the list of SSRCs and their msid values.

- o When a new msid value occurs in the description, the recipient can signal to its application that a new media stream has been added.
- o When a description is updated to have more SSRCs with the same msid value, the recipient can signal to its application that new media stream tracks have been added to the media stream.
- o When a description is updated to no longer list the msid value on a specific ssrc, the recipient can signal to its application that the corresponding media stream track has been closed.
- o When a description is updated to no longer list the msid value on any ssrc, the recipient can signal to its application that the media stream has been closed.

OPEN ISSUE: Exactly when should the recipient signal that the track is closed? When the msid value disappears from the description, when the SSRC disappears by the rules of [RFC3550] section 6.3.4 (BYE packet received) and 6.3.5 (timeout), any of the above, or some combination of the above?

4.1. Handling of non-signalled tracks

Pre-WebRTC entities will not send msid. This means that there will be some incoming RTP packets with SSRCs where the recipient does not know about a corresponding MediaStream id.

Handling will depend on whether or not any SSRCs are signalled in the relevant RTP session. There are two cases:

- o No SSRC is signalled with an msid attribute. The SDP session is assumed to be a backwards-compatible session. All incoming SSRCs, on all RTP sessions that are part of the SDP session, are assumed to belong to a single media stream. The identifier of this media stream is "default".
- o Some SSRCs are signalled with an msid attribute. In this case, the session is WebRTC compatible, and the newly arrived SSRCs are either caused by a bug or by timing skew between the arrival of the media packets and the SDP description. These packets MAY be discarded, or they MAY be buffered for a while in order to allow immediate startup of the media stream when the SDP description is updated. The arrival of media packets MUST NOT cause a new MediaStreamTrack to be signalled.

Note: This means that it is wise to include at least one `a=ssrc:` line with an `msid` attribute, even when no media streams are yet attached to the session. (Alternative: Mark the RTP session explicitly as "I will signal the media stream tracks explicitly").

It follows from the above that media stream tracks in the "default" media stream cannot be closed by signalling; the application must instead signal these as closed when the SSRC disappears according to the rules of RFC 3550 section 6.3.4 and 6.3.5.

5. IANA Considerations

This document requests IANA to register the "msid" attribute in the "att-field (source level)" registry within the SDP parameters registry, according to the procedures of [RFC5576]

The required information is:

- o Contact name, email: IETF, contacted via `rtcweb@ietf.org`, or a successor address designated by IESG
- o Attribute name: `msid`
- o Long-form attribute name: Media stream group Identifier
- o The attribute value contains only ASCII characters, and is therefore not subject to the `charset` attribute.
- o The attribute gives an association over a set of SSRCs, potentially in different RTP sessions. It can be used to signal the relationship between a WebRTC `MediaStream` and a set of SSRCs.
- o The details of appropriate values are given in RFC XXXX.

This document requests IANA to register the "WMS" semantic within the "Semantics for the "ssrc-group" SDP Attribute" registry within the SDP parameters registry.

The required information is:

- o Description: WebRTC Media Stream, as given in RFC XXXX.
- o Token: WMS
- o Standards track reference: RFC XXXX

IANA is requested to replace "RFC XXXX" with the RFC number of this

document upon publication.

6. Security Considerations

An adversary with the ability to modify SDP descriptions has the ability to switch around tracks between media streams. This is a special case of the general security consideration that modification of SDP descriptions needs to be confined to entities trusted by the application.

No attacks that are relevant to the browser's security have been identified that depend on this mechanism.

7. Acknowledgements

This note is based on sketches from, among others, Justin Uberti and Cullen Jennings.

Special thanks to Miguel Garcia for his work in reviewing this draft, with many specific language suggestions.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.
- [W3C.WD-webrtc-20120209] Bergkvist, A., Burnett, D., Narayanan, A., and C. Jennings, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120209, February 2012, <<http://www.w3.org/TR/2012/WD-webrtc-20120209>>.

8.2. Informative References

- [I-D.westerlund-avtcore-multiplex-architecture]
Westerlund, M., Burman, B., Perkins, C., and H. Alvestrand, "Guidelines for using the Multiplexing Features of RTP",
draft-westerlund-avtcore-multiplex-architecture-02 (work in progress), July 2012.
- [RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

Appendix A. Design considerations, open questions and alternatives

This appendix should be deleted before publication as an RFC.

One suggested mechanism has been to use CNAME instead of a new attribute. This was abandoned because CNAME identifies a synchronization context; one can imagine both wanting to have tracks from the same synchronization context in multiple media streams and wanting to have tracks from multiple synchronization contexts within one media stream.

Another suggestion has been to put the msid value within an attribute of RTCP SR (sender report) packets. This doesn't offer the ability to know that you have seen all the tracks currently configured for a media stream.

There has been a suggestion that this mechanism could be used to mute tracks too. This is not done at the moment.

The special value "default" and the reservation of "example*" seems bothersome; apart from that, it's a random string. It's uncertain whether "example" has any benefit.

An alternative to the "default" media stream is to let each new media stream track without a msid attribute create its own media stream. Input on this question is sought.

Discarding of incoming data when the SDP description isn't updated yet (section 3) may cause clipping. However, the same issue exists

when crypto keys aren't available. Input sought.

There's been a suggestion that acceptable SSRCs should be signalled in a response, giving a recipient the ability to say "no" to certain SSRCs. This is not supported in the current version of this document.

This specification reuses the ssrc-group semantics registry for this semantic, on the argument that the WMS purpose is more similar to an SSRC grouping than a session-level grouping, and allows values from both registries, on the argument that some semantics (like LS) are well defined for MSID. Input sought.

Appendix B. Change log

This appendix should be deleted before publication as an RFC.

B.1. Changes from rtcweb-msid-00 to -01

Added track identifier.

Added inclusion-by-reference of draft-lennox-mmusic-source-selection for track muting.

Some rewording.

B.2. Changes from rtcweb-msid-01 to -02

Split document into sections describing a generic grouping mechanism and sections describing the application of this grouping mechanism to the WebRTC MediaStream concept.

Removed the mechanism for muting tracks, since this is not central to the MSID mechanism.

B.3. Changes from rtcweb-msid-02 to mmusic-msid-00

Changed the draft name according to the wishes of the MMUSIC group chairs.

Added text indicting cases where it's appropriate to have the same appdata for multiple SSRCs.

Minor textual updates.

B.4. Changes from mmusic-msid-00 to -01

Increased the amount of explanatory text, much based on a review by Miguel Garcia.

Removed references to BUNDLE, since that spec is under active discussion.

Removed distinguished values of the MSID identifier.

Author's Address

Harald Alvestrand
Google
Kungsbron 2
Stockholm, 11122
Sweden

Email: harald@alvestrand.no

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 15, 2013

H. Alvestrand
Google
December 12, 2012

Cross Session Stream Identification in the Session Description Protocol
draft-alvestrand-mmusic-msid-02

Abstract

This document specifies a grouping mechanism for RTP media streams that can be used to specify relations between media streams within different RTP sessions as well as within a single RTP session.

This mechanism is used to signal the association between the RTP concept of SSRC and the WebRTC concept of "media stream" / "media stream track" using SDP signaling.

This document is an input document for discussion. It should be discussed in the MMUSIC WG list, mmusic@ietf.org.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 June 15, 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

1. Introduction	3
1.1. Structure Of This Document	3
1.2. Why A New Mechanism Is Needed	3
1.3. Application to the WEBRTC MediaStream	4
2. The Msid Mechanism	4
3. The Msid-Semantic Attribute	5
4. Applying Msid to WebRTC MediaStreams	6
4.1. Handling of non-signalled tracks	7
5. IANA Considerations	8
6. Security Considerations	9
7. Acknowledgements	9
8. References	9
8.1. Normative References	9
8.2. Informative References	10
Appendix A. Design considerations, open questions and and alternatives	10
Appendix B. Change log	11
B.1. Changes from rtcweb-msid-00 to -01	11
B.2. Changes from rtcweb-msid-01 to -02	11
B.3. Changes from rtcweb-msid-02 to mmusic-msid-00	11
B.4. Changes from mmusic-msid-00 to -01	12
B.5. Changes from mmusic-msid-01 to -02	12
Author's Address	12

1. Introduction

1.1. Structure Of This Document

This document extends the SSRC grouping framework [RFC5888] by adding a new grouping relation that can cross RTP session boundaries if needed.

Section 1.2 gives the background on why a new mechanism is needed.

Section 2 gives the definition of the new mechanism.

Section 4 gives the application of the new mechanism for providing necessary semantic information for the association of MediaStreamTracks to MediaStreams in the WebRTC API .

1.2. Why A New Mechanism Is Needed

When media is carried by RTP [RFC3550], each RTP media stream is distinguished inside an RTP session by its SSRC; each RTP session is distinguished from all other RTP sessions by being on a different transport association (strictly speaking, 2 transport associations, one used for RTP and one used for RTCP, unless RTCP multiplexing [RFC5761] is used).

There exist cases where an application using RTP and SDP needs to signal some relationship between RTP media streams that may be carried in either the same RTP session or different RTP sessions. For instance, there may be a need to signal a relationship between a video track in one RTP session and an audio track in another RTP session. In traditional SDP, it is not possible to signal that these two tracks should be carried in one session, so they are carried in different RTP sessions.

The SSRC grouping mechanism ("a=ssrc-group") [RFC5576] can be used to associate RTP media streams when those RTP media streams are part of the same RTP session. The semantics of this mechanism prevent the association of RTP media streams that are spread across different RTP sessions.

The SDP grouping framework [RFC5888] can be used to group RTP sessions. When an RTP session carries one and only one RTP media stream, it is possible to associate RTP media streams across different RTP sessions. However, if an RTP session has multiple RTP media streams, using multiple SSRCS, the SDP grouping framework cannot be used for this purpose.

There are use cases (some of which are discussed in

[I-D.westerlund-avtcore-multiplex-architecture]) where neither of these approaches is appropriate; In those cases, a new mechanism is needed.

In addition, there is sometimes the need for an application to specify some application-level information about the association between the SSRC and the group. This is not possible using either of the frameworks above.

1.3. Application to the WEBRTC MediaStream

The W3C WebRTC API specification [W3C.WD-webrtc-20120209] specifies that communication between WebRTC entities is done via MediaStreams, which contain MediaStreamTracks. A MediaStreamTrack is generally carried using a single SSRC in an RTP session (forming an RTP media stream. The collision of terminology is unfortunate.) There might possibly be additional SSRCs, possibly within additional RTP sessions, in order to support functionality like forward error correction or simulcast. This complication is ignored below.

In the RTP specification, media streams are identified using the SSRC field. Streams are grouped into RTP Sessions, and also carry a CNAME. Neither CNAME nor RTP session correspond to a MediaStream. Therefore, the association of an RTP media stream to MediaStreams need to be explicitly signaled.

The marking needs to be on a per-SSRC basis, since one RTP session can carry media from multiple MediaStreams, and one MediaStream can have media in multiple RTP sessions. This means that the [RFC4574] "label" attribute, which is used to label RTP sessions, is not usable for this purpose.

The marking needs to also carry the unique identifier of the RTP media stream as a MediaStreamTrack within the media stream; this is done using a single letter to identify whether it belongs in the video or audio track list, and the MediaStreamTrack's position within that array.

This usage is described in Section 4.

2. The Msid Mechanism

This document extends the Source-Specific Media Attributes framework [RFC5576] by adding a new "msid" attribute that can be used with the "a=ssrc" SDP attribute. This new attribute allows endpoints to associate RTP media streams that are carried in the same or different RTP sessions, as well as allowing application-specific information to

the association.

The value of the "msid" attribute consists of an identifier and optional application-specific data, according to the following ABNF [RFC5234] grammar:

```
; "attribute" is defined in RFC 4566.  
; This attribute should be used with the ssrc-attr from RFC 5576.  
attribute =/ msid-attr  
msid-attr = "msid:" identifier [ " " appdata ]  
identifier = token  
appdata = token
```

An example MSID value for the SSRC 1234 might look like this:

```
a=ssrc:1234 msid:examplefoo v1
```

The identifier is a string of ASCII characters chosen from 0-9, a-z, A-Z and - (hyphen), consisting of between 1 and 64 characters. It MUST be unique among the identifier values used in the same SDP session. It is RECOMMENDED that it is generated using a random-number generator.

Application data is carried on the same line as the identifier, separated from the identifier by a space.

The identifier uniquely identifies a group within the scope of an SDP description.

There may be multiple msid attributes on a single SSRC. There may also be multiple SSRCs that have the same value for identifier and application data.

Endpoints can update the associations between SSRCs as expressed by msid attributes at any time; the semantics and restrictions of such grouping and ungrouping are application dependent.

3. The Msid-Semantic Attribute

In order to fully reproduce the semantics of the SDP and SSRC grouping frameworks, a session-level attribute is defined for signaling the semantics associated with an msid grouping.

This OPTIONAL attribute gives the group identifier and its group semantic; it carries the same meaning as the ssrc-group-attr of RFC

5576 section 4.2, but uses the identifier of the group rather than a list of SSRC values.

An empty list of identifiers is an indication that the sender understands the indicated semantic, but has no msid groupings of the given type in the present SDP.

The ABNF of msid-semantic is:

```
attribute =/ msid-semantic-attr
msid-semantic-attr = "msid-semantic:" " " token (" " identifier)*
token = <as defined in RFC 4566>
```

The semantic field may hold values from the IANA registries "Semantics for the "ssrc-group" SDP Attribute" and "Semantics for the "group" SDP Attribute".

An example msid-semantic might look like this:

```
a=msid-semantic: LS xyzzy forolow
```

This means that the SDP description has two lip sync groups, with the group identifiers xyzzy and forolow, respectively.

4. Applying Msid to WebRTC MediaStreams

This section creates a new semantic for use with the framework defined in Section 2, to be used for associating SSRCs representing MediaStreamTracks within MediaStreams as defined in [W3C.WD-webrtc-20120209].

The semantic token for this semantic is "WMS" (short for WebRTC Media Stream).

The value of the msid corresponds to the "id" attribute of a MediaStream.

In a WebRTC-compatible SDP description, all SSRCs intending to be sent from one peer will be identified in the SDP generated by that entity.

The appdata for a WebRTC MediaStreamTrack consists of the "id" attribute of a MediaStreamTrack.

If two different SSRCs have the same value for identifier and appdata, it means that these two SSRCs are both intended for the same MediaStreamTrack. This may occur if the sender wishes to use

simulcast or forward error correction, or if the sender intends to switch between multiple codecs on the same `MediaStreamTrack`.

When an SDP description is updated, a specific `msid` continues to refer to the same `MediaStream`. Once negotiation has completed on a session, there is no memory; an `msid` value that appears in a later negotiation will be taken to refer to a new `MediaStream`.

The following are the rules for handling updates of the list of SSRCs and their `msid` values.

- o When a new `msid` value occurs in the description, the recipient can signal to its application that a new `MediaStream` has been added.
- o When a description is updated to have more SSRCs with the same `msid` value, but different `appdata` values, the recipient can signal to its application that new media stream tracks have been added to the media stream.
- o When a description is updated to no longer list the `msid` value on a specific `ssrc`, the recipient can signal to its application that the corresponding media stream track has been closed.
- o When a description is updated to no longer list the `msid` value on any `ssrc`, the recipient can signal to its application that the media stream has been closed.

In addition to signaling that the track is closed when it disappears from the SDP, the track will also be signaled as being closed when the SSRC disappears by the rules of [RFC3550] section 6.3.4 (BYE packet received) and 6.3.5 (timeout).

4.1. Handling of non-signalled tracks

Pre-WebRTC entities will not send `msid`. This means that there will be some incoming RTP packets with SSRCs where the recipient does not know about a corresponding `MediaStream` id.

Handling will depend on whether or not any SSRCs are signaled in the relevant RTP session. There are two cases:

- o No SSRC is signaled with an `msid` attribute. The SDP session is assumed to be a backwards-compatible session. All incoming SSRCs, on all RTP sessions that are part of the SDP session, are assumed to belong to a single media stream. The identifier of this media stream is "default".

- o Some SSRCs are signaled with an msid attribute. In this case, the session is WebRTC compatible, and the newly arrived SSRCs are either caused by a bug or by timing skew between the arrival of the media packets and the SDP description. These packets MAY be discarded, or they MAY be buffered for a while in order to allow immediate startup of the media stream when the SDP description is updated. The arrival of media packets MUST NOT cause a new MediaStreamTrack to be signaled.

If a WebRTC entity sends a description, it MUST include the msid-semantic: WMS attribute, even if no media streams are sent. This allows us to distinguish between the case of no media streams at the moment and the case of legacy SDP generation.

It follows from the above that media stream tracks in the "default" media stream cannot be closed by signaling; the application must instead signal these as closed when the SSRC disappears according to the rules of RFC 3550 section 6.3.4 and 6.3.5.

5. IANA Considerations

This document requests IANA to register the "msid" attribute in the "att-field (source level)" registry within the SDP parameters registry, according to the procedures of [RFC5576]

The required information is:

- o Contact name, email: IETF, contacted via rtcweb@ietf.org, or a successor address designated by IESG
- o Attribute name: msid
- o Long-form attribute name: Media stream group Identifier
- o The attribute value contains only ASCII characters, and is therefore not subject to the charset attribute.
- o The attribute gives an association over a set of SSRCs, potentially in different RTP sessions. It can be used to signal the relationship between a WebRTC MediaStream and a set of SSRCs.
- o The details of appropriate values are given in RFC XXXX.

This document requests IANA to create a new registry called "Semantics for the msid-semantic SDP attribute", which should have exactly the same rules as for the "Semantics for the ssrc-group SDP attribute" registry, and to register the "WMS" semantic within this

new registry.

The required information is:

- o Description: WebRTC Media Stream, as given in RFC XXXX.
- o Token: WMS
- o Standards track reference: RFC XXXX

IANA is requested to replace "RFC XXXX" with the RFC number of this document upon publication.

6. Security Considerations

An adversary with the ability to modify SDP descriptions has the ability to switch around tracks between media streams. This is a special case of the general security consideration that modification of SDP descriptions needs to be confined to entities trusted by the application.

No attacks that are relevant to the browser's security have been identified that depend on this mechanism.

7. Acknowledgements

This note is based on sketches from, among others, Justin Uberti and Cullen Jennings.

Special thanks to Miguel Garcia and Paul Kyzivat for their work in reviewing this draft, with many specific language suggestions.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

[W3C.WD-webrtc-20120209]
Bergkvist, A., Burnett, D., Narayanan, A., and C. Jennings, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD WD-webrtc-20120209, February 2012,
<<http://www.w3.org/TR/2012/WD-webrtc-20120209>>.

8.2. Informative References

- [I-D.westerlund-avtcore-multiplex-architecture]
Westerlund, M., Burman, B., Perkins, C., and H. Alvestrand, "Guidelines for using the Multiplexing Features of RTP",
draft-westerlund-avtcore-multiplex-architecture-02 (work in progress), July 2012.
- [RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, April 2010.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

Appendix A. Design considerations, open questions and alternatives

This appendix should be deleted before publication as an RFC.

One suggested mechanism has been to use CNAME instead of a new attribute. This was abandoned because CNAME identifies a synchronization context; one can imagine both wanting to have tracks from the same synchronization context in multiple media streams and wanting to have tracks from multiple synchronization contexts within one media stream.

Another suggestion has been to put the msid value within an attribute of RTCP SR (sender report) packets. This doesn't offer the ability to know that you have seen all the tracks currently configured for a media stream.

There has been a suggestion that this mechanism could be used to mute tracks too. This is not done at the moment.

An alternative to the "default" media stream is to let each new media stream track without a msid attribute create its own media stream. Input on this question is sought.

Discarding of incoming data when the SDP description isn't updated yet (section 3) may cause clipping. However, the same issue exists when crypto keys aren't available. Input sought.

There's been a suggestion that acceptable SSRCs should be signaled in a response, giving a recipient the ability to say "no" to certain SSRCs. This is not supported in the current version of this document.

Appendix B. Change log

This appendix should be deleted before publication as an RFC.

B.1. Changes from rtcweb-msid-00 to -01

Added track identifier.

Added inclusion-by-reference of draft-lennox-mmusic-source-selection for track muting.

Some rewording.

B.2. Changes from rtcweb-msid-01 to -02

Split document into sections describing a generic grouping mechanism and sections describing the application of this grouping mechanism to the WebRTC MediaStream concept.

Removed the mechanism for muting tracks, since this is not central to the MSID mechanism.

B.3. Changes from rtcweb-msid-02 to mmusic-msid-00

Changed the draft name according to the wishes of the MMUSIC group chairs.

Added text indicting cases where it's appropriate to have the same appdata for multiple SSRCs.

Minor textual updates.

B.4. Changes from mmusic-msid-00 to -01

Increased the amount of explanatory text, much based on a review by Miguel Garcia.

Removed references to BUNDLE, since that spec is under active discussion.

Removed distinguished values of the MSID identifier.

B.5. Changes from mmusic-msid-01 to -02

Changed the order of the "msid-semantic: " attribute's value fields and allowed multiple identifiers. This makes the attribute useful as a marker for "I understand this semantic".

Changed the syntax for "identifier" and "appdata" to be "token".

Changed the registry for the "msid-semantic" attribute values to be a new registry, based on advice given in Atlanta.

Author's Address

Harald Alvestrand
Google
Kungsbron 2
Stockholm, 11122
Sweden

Email: harald@alvestrand.no

MMUSIC Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 11, 2013

C. Holmberg
Ericsson
H. Alvestrand
Google
J. Lennox
Vidyo
October 8, 2012

Multiplexed Media Types (MMT) Using Session Description Protocol (SDP)
Port Numbers
draft-holmberg-mmusic-sdp-mmt-negotiation-00.txt

Abstract

This specification defines a new SDP Grouping Framework SDP grouping framework extension, "MMT", and a new SDP media type, "anymedia". Together they can be used with the Session Description Protocol (SDP) Offer/Answer mechanism to negotiate the usage of multiplexed media types, which refers to the usage of a single 5-tuple for different media types.

This specification also defined a new SDP attribute, "mmtype", which can be used within a "anymedia" SDP Media Description to map PT (Payload Type) values to a specific media type.

Status of this Memo

This Internet-Draft is submitted to IETF 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 11, 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

1. Introduction	3
2. Terminology	4
3. Conventions	4
4. Applicability Statement	4
5. SDP Grouping Framework MMT Extension Semantics	4
5.1. General	4
5.2. Usage	5
5.2.1. General	5
5.2.2. SDP Offer/Answer Usage	5
6. anymedia SDP Media Type	6
6.1. General	6
6.2. SDP Extensions	6
6.3. SDP Attributes	6
6.4. Bandwidth	7
6.5. ICE Usage	7
6.6. RTP Sessions	7
7. mmtype SDP attribute	7
7.1. General	7
7.2. Syntax	7
7.3. Usage	7
7.3.1. General	7
7.3.2. SDP Offer/Answer Usage	8
8. Security Considerations	8
9. Example	8
10. IANA Considerations	10
11. Acknowledgements	11
12. Change Log	11
13. References	11
13.1. Normative References	11
13.2. Informative References	11
Authors' Addresses	12

1. Introduction

In the IETF RTCWEB WG, a need to use a single 5-tuple for sending and receiving media associated with multiple SDP Media Descriptions [RFC4566] has been identified. This would e.g. allow the usage of a single set of Interactive Connectivity Establishment (ICE) [RFC5245] candidates for multiple media descriptions. Normally different media types (audio, video etc) will be described using different SDP Media Descriptions.

As defined in RFC 4566 [RFC4566], the semantics of using the same port number for multiple SDP Media Descriptions is undefined. Therefore, in order to be able to use the same port value for multiple media types, it must be possible to describe multiple media types within a single SDP Media Description.

This specification defines a new SDP Grouping Framework SDP grouping framework [RFC5888] extension, "MMT", and a new SDP media type, "anymedia". Together they can be used with the Session Description Protocol (SDP) Offer/Answer mechanism [RFC3264] to negotiate the usage of multiplexed media types, which refers to the usage of a single 5-tuple for different media types.

This specification also defined a new SDP attribute, "mmtype", which can be used within a "anymedia" SDP Media Description to map PT (Payload Type) values to a specific media type.

When an endpoint generates an SDP Offer or SDP Answer, which includes one or more "MMT" groups, each group will contain one "anymedia" SDP Media Description and one or more SDP Media Descriptions for specific media types (audio, video, etc).

When media is transported using the Real-Time Protocol (RTP) [RFC3550], each SDP Media Description is assumed to form a separate RTP Session [RFC3550]. The same applies to media associated with a "anymedia" SDP Media Description, ie all media types associated with a "anymedia" SDP Media Description is by default assumed to form a single RTP Session.

The mechanism is backward compatible. Entities that do not support (or, for a given session, are not willing to use) the "MMT" grouping extension and the "anymedia" media type, are expected to generate an SDP Answer, which does not contain a "MMT" group, and where the "anymedia" SDP Media Description is rejected.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

5-tuple: A collection of the following values: source address, source port, destination address, destination port and protocol.

Multiplexed media types: Two or more RTP streams, possibly of different media types, using a single 5-tuple. The RTCP streams associated with the RTP streams also use a single 5-tuple, which might be the same, but can also be different, as the one used by the RTP streams.

3. Conventions

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 BCP 14, RFC 2119 [RFC2119].

4. Applicability Statement

The mechanism in this specification only applies to the Session Description Protocol (SDP) [RFC4566], when used together with the SDP Offer/Answer mechanism [RFC3264].

5. SDP Grouping Framework MMT Extension Semantics

5.1. General

This section defines a new SDP Grouping Framework extension, "MMT".

The "MMT" extension can be indicated using a "group" SDP session-level attribute. Each SDP Media Description ("m=" line) that is grouped together, using a "mid" SDP media-level attribute, is part of a specific "MMT" group.

A "MMT" group is not usable standalone. It MUST be used together with a "anymedia" SDP Media Description.

5.2. Usage

5.2.1. General

A "MMT" group MUST contain a "anymedia" SDP Media Description, and at least one SDP Media Description for a specific SDP Media Type (audio, video, etc).

An SDP Offerer [RFC3264] uses the "MMT" group to offer at least one SDP Media Description for specific SDP Media Types, and a "anymedia" SDP Media Description, which can contain multiple SDP Media Types sharing a single SDP Media Description. From a "MMT" group the SDP Answerer [RFC3264] will accept either the SDP Media Descriptions for the specific SDP Media Types, or the "anymedia" SDP Media Description.

It is RECOMMENDED that the capabilities of the "anymedia" SDP Media Description match the capabilities (codecs, RTCP multiplexing etc) of the SDP Media Descriptions for the specific SDP Media Types, in order to provide the same capabilities no matter whether SDP Media Descriptions for specific SDP Media Types, or the "anymedia" SDP Media Description, is used to establish the session.

NOTE: An SDP Message Body can contain multiple "MMT" groups.

5.2.2. SDP Offer/Answer Usage

5.2.2.1. SDP Offerer Procedures

When an SDP Offerer generates an SDP Offer, which contains one or more "MMT" groups, for each "MMT" group the SDP Offerer MUST:

- 1) Include a "anymedia" SDP Media Description; and
- 2) Include at least one SDP Media Description for a specific media type (audio, video, etc).

5.2.2.2. SDP Answerer Procedures

When an SDP Answerer generates an SDP Answer, for each "MMT" group in the associated SDP Offer it MUST either:

- 1) Accept the "anymedia" SDP Media Description, and reject all other SDP Media Descriptions associated with the "MMT" group; or
- 2) Reject the "anymedia" SDP Media Description, and accept some or all of the other SDP Media Descriptions associated with the MMT group.

NOTE: As described in [RFC3264], an SDP Media Description can be rejected by setting the port value of the associated m- line to zero in the SDP Answer.

NOTE: As described in [RFC3264] the SDP Answer must contain the same number of SDP Media Descriptions as the associated SDP Offer.

6. anymedia SDP Media Type

6.1. General

This section describes a new SDP media type [RFC4566], "anymedia", for SDP Media Descriptions [RFC4566]. "anymedia" does not refer to a specific media type, but allows multiple media types (audio, video etc) to be associated with a single SDP Media Description. All media associated with a "anymedia" SDP Media Description will share the same IP address+port, protocol (e.g. RTP/AVP) and other information (e.g. ICE candidates) associated with the SDP Media Description. It allows, if both endpoints support the mechanism, multiple media types to be multiplexed on a single 5-tuple. PT (Payload Type) values will be listed in a normal fashion in the format list of the SDP Media Description. The SDP rtpmap attribute [RFC4566] will be used in a normal fashion to map each PT to a codec, and the SDP mmtype attribute will be used to map each PT to a specific media type (e.g. audio, video, etc).

Within a "anymedia" SDP Media Description, each PT value SHOULD be described using an "rtpmap" SDP Attribute [RFC4566], even if the PT value is static. In addition, as it might not always be possible to retrieve the media type from the "rtpmap" SDP Attribute value, each PT value MUST be mapped to a specific media type, using the "mmtype" SDP Attribute.

6.2. SDP Extensions

OPEN ISSUE: Which, if any, SDP Extensions shall we require support of?

6.3. SDP Attributes

In a normal fashion, any media level SDP Attribute (e.g. the directionality attributes) associated with the "anymedia" SDP Media Description applies to all media associated with the SDP Media Description.

NOTE: Additional extensions are needed in order to specify SDP Attribute values for individual media types, or individual media

sources, associated with the "anymedia" SDP Media Description.

6.4. Bandwidth

The SDP bandwidth parameter, b=, is used in a normal fashion, as described in [RFC4566]

NOTE: Additional extensions are needed in order to specify SDP Bandwidth values for individual media types, and for a specific media direction.

6.5. ICE Usage

ICE [RFC5245], if supported, will be used in a normal fashion, and the ICE Candidate information will apply to all media types associated with the "anymedia" SDP Media Description.

6.6. RTP Sessions

By default, all media associated with a "anymedia" SDP Media Description is considered to be part of a single RTP Session [RFC3550].

7. mmttype SDP attribute

7.1. General

This section defines a new SDP media level attribute [RFC4566], "mmtype" (Multiplexed Media Type). The attribute is used within "anymedia" SDP Media Descriptions to indicate the media type associated with a specific PT value.

7.2. Syntax

a=mmtype: format media

format and media as defined in [RFC4566].

7.3. Usage

7.3.1. General

The attribute is used within "anymedia" SDP Media Descriptions to indicate the media type associated with a specific PT value. This specification does not define the usage of the attribute within other types of SDP Media Descriptions.

For each instance of the "mmtype" attribute, the associated PT value MUST also be listed in the format list of the associated SDP m- line. Within a given SDP Media Description, there MUST only be one 'mmtype' attribute associated with a given PT value. An entity MUST either reject or discard an SDP Media Description that contains 'mmtype' attributes with PT values not listed in the associated m- line. An entity MUST either reject or discard an SDP Media Description that contains multiple 'mmtype' attributes for the same PT value.

7.3.2. SDP Offer/Answer Usage

There are no SDP Offer/Answer specific procedures defined for the "mmtype" SDP attribute.

8. Security Considerations

TBA

9. Example

The example below shows an SDP Offer, where multiplexed media types is offered. The example also shows two SDP Answer alternatives: one where multiplexed media types is accepted, and one where multiplexed media types is rejected (or, not even supported) by the SDP Answerer.

SDP Offer (multiplexed media types offered)

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.com
s=
c=IN IP4 host.atlanta.com
t=0 0
a=group:MMT foo bar zoe
m=audio 10000 RTP/AVP 0 8 97
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 20000 RTP/AVP 31 32
a=mid:bar
b=AS:1000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
m=anymedia 30000 RTP/AVP 0 8 97 31 32
a=mid:zoe
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=mmtype: 0 audio
a=mmtype: 8 audio
a=mmtype: 97 audio
a=mmtype: 31 video
a=mmtype: 32 video
```

SDP Answer (multiplexed media types accepted)

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.com
s=
c=IN IP4 host.biloxi.com
t=0 0
a=group:MMT foo bar
m=audio 0 RTP/AVP 0
a=mid:foo
a=rtpmap:0 PCMU/8000
m=video 0 RTP/AVP 32
a=mid:bar
a=rtpmap:32 MPV/90000
m=anymedia 60000 RTP/AVP 0 32
a=mid:zoe
a=rtpmap:0 PCMU/8000
a=rtpmap:32 MPV/90000
a=mmtype: 0 audio
a=mmtype: 32 video
```

SDP Answer (multiplexed media types not accepted)

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.com
s=
c=IN IP4 host.biloxi.com
t=0 0
a=group:MMT foo bar
m=audio 40000 RTP/AVP 0
a=mid:foo
a=rtpmap:0 PCMU/8000
m=video 50000 RTP/AVP 32
a=mid:bar
a=rtpmap:32 MPV/90000
m=anymedia 0 RTP/AVP 0 32
```

a=mid:zoe

SDP Offer with ICE (multiplexed media types offered)

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.com
s=
c=IN IP4 host.atlanta.com
t=0 0
a=group:MMT foo bar zoe
m=audio 10000 RTP/AVP 0 8 97
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=candidate:1 1 UDP 1694498815 host.atlanta.com 10000 typ host
m=video 20000 RTP/AVP 31 32
a=mid:bar
b=AS:1000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=candidate:1 1 UDP 1694498815 host.atlanta.com 20000 typ host
m=anymedia 30000 RTP/AVP 0 8 97 31 32
a=mid:zoe
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/9000
a=mmtype: 0 audio
a=mmtype: 8 audio
a=mmtype: 97 audio
a=mmtype: 31 video
a=mmtype: 32 video
a=candidate:1 1 UDP 1694498815 host.atlanta.com 30000 typ host
```

10. IANA Considerations

This document requests IANA to register the new SDP Grouping semantic extension called MMT.

11. Acknowledgements

The usage of the SDP grouping mechanism is based on a similar alternative proposed by Harald Alvestrand. The SDP examples are also modified versions from the ones in the Alvestrand proposal.

The usage of a dedicated SDP media type to described multiplexed media types types is based on input from Jonathan Lennox.

12. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-holmberg-mmusic-sdp-mmt-negotiation-xx

- o Add change.

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.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

13.2. Informative References

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Harald Tveit Alvestrand
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: harald@alvestrand.no

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

Email: jonathan@vidyo.com

MMUSIC Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 21, 2013

C. Holmberg
Ericsson
H. Alvestrand
Google
August 20, 2012

Multiplexing Negotiation Using Session Description Protocol (SDP) Port
Numbers
draft-ietf-mmusic-sdp-bundle-negotiation-01.txt

Abstract

This specification defines a new SDP Grouping Framework SDP grouping framework extension, "BUNDLE", that can be used with the Session Description Protocol (SDP) Offer/Answer mechanism to negotiate the usage of bundled media, which refers to the usage of a single 5-tuple for media associated with multiple SDP media descriptions ("m=" lines).

Status of this Memo

This Internet-Draft is submitted to IETF 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 February 21, 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

1. Introduction	3
2. Terminology	3
3. Conventions	4
4. Applicability Statement	4
5. SDP Grouping Framework BUNDLE Extension Semantics	4
6. SDP Offer/Answer Procedures	4
6.1. General	4
6.2. SDP Offerer Procedures	5
6.3. SDP Answerer Procedures	6
6.4. Bundled SDP Information	6
6.4.1. General	6
6.4.2. Bandwidth (b=)	6
7. Single vs Multiple RTP Sessions	6
7.1. General	6
7.2. Single RTP Session	6
8. Usage With ICE	7
8.1. General	7
8.2. Candidates	7
9. Security Considerations	8
10. Example	8
11. IANA Considerations	10
12. Acknowledgements	10
13. Change Log	10
14. References	10
14.1. Normative References	10
14.2. Informative References	11
Authors' Addresses	11

1. Introduction

In the IETF RTCWEB WG, a need to use a single 5-tuple for sending and receiving media associated with multiple SDP media descriptions ("m=" lines) has been identified. This would e.g. allow the usage of a single set of Interactive Connectivity Establishment (ICE) [RFC5245] candidates for multiple media descriptions. Normally different media types (audio, video etc) will be described using different media descriptions.

This specification defines a new SDP Grouping Framework SDP grouping framework [RFC5888] extension, "BUNDLE", that can be used with the Session Description Protocol (SDP) Offer/Answer mechanism [RFC3264] to negotiate the usage of bundled media, which refers to the usage of a single 5-tuple for media associated with multiple SDP media descriptions ("m=" lines).

When an endpoint generates an SDP Offer or SDP Answer [RFC3264], which includes a "BUNDLE" group, each "m=" line associated with the group will share a single port number value.

As defined in RFC 4566 [RFC4566], the semantics of multiple "m=" lines using the same port number value are undefined, and there is no grouping defined by such means. Instead, an explicit grouping mechanism needs to be used to express the intended semantics. This specification provides such extension.

When media is transported using the Real-Time Protocol (RTP) [RFC3550], the default assumption of the mechanism is that all media associated with a "BUNDLE" group will form a single RTP Session [RFC3550]. However, future specifications can extend the mechanism, in order to negotiate RTP Session multiplexing, i.e. "BUNDLE" groups where media associated with a group form multiple RTP Sessions.

The mechanism is backward compatible. Entities that do not support the "BUNDLE" grouping extension, or do not want to enable the mechanism for a given session, are expected to generate a "normal" SDP Answer, using different port number values for each "m=" line, to the SDP Offer. The SDP Offerer [RFC3264] will still use a single port number value for each media, but as the SDP Answerer [RFC3264] will use separate ports a single 5-tuple will not be used for media associated with multiple "m=" lines between the endpoints.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in RFC 2119 [RFC2119].

5-tuple: A collection of the following values: source address, source port, destination address, destination port and protocol.

Bundled media: Two or more RTP streams using a single 5-tuple. The RTCP streams associated with the RTP streams also use a single 5-tuple, which might be the same, but can also be different, as the one used by the RTP streams.

3. Conventions

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 BCP 14, RFC 2119 [RFC2119].

4. Applicability Statement

The mechanism in this specification only applies to the Session Description Protocol (SDP) [RFC4566], when used together with the SDP Offer/Answer mechanism [RFC3264].

5. SDP Grouping Framework BUNDLE Extension Semantics

This section defines a new SDP Grouping Framework extension, "BUNDLE".

The "BUNDLE" extension can be indicated using an SDP session-level 'group' attribute. Each SDP media description ("m=" line) that is grouped together, using an SDP media-level 'mid' attribute, is part of a specific "BUNDLE" group.

6. SDP Offer/Answer Procedures

6.1. General

When an SDP Offerer or SDP Answerer generates an SDP Offer or SDP Answer, that describes bundled media, it MUST insert an SDP session-level 'group' attribute, with a "BUNDLE" value, and assign SDP media-level 'mid' attribute values to each "m=" line associated with the "BUNDLE" group.

In addition, the entity that generates the SDP Offer or SDP Answer

MUST, for each "m=" line that is part of the "BUNDLE" group:

- o 1. Use the same port number value.
- o 2. Use the same connection data ("c=" line) value.
- o 3. Use the same SDP 'rtcp' attribute value, when used.
- o 4. Use the same ICE candidate values, when used.
- o 5. Insert an SDP 'rtcp-mux' attribute.

NOTE: If an entity wants to disable specific media ("m=" line) associated with a "BUNDLE" group, it will use a zero port number value for the "m=" line associated with the media.

6.2. SDP Offerer Procedures

When an SDP Offerer creates an SDP Offer, that offers bundled media, it MUST create the SDP Offer according to the procedures in Section 6.1.

If the associated SDP Answer contains an SDP session-level 'group' attribute, with a "BUNDLE" value, and the SDP Answer is created according to the procedures in Section 6.1 (the same port number value is used for each "m=" line associated with the "BUNDLE" group, etc), the SDP Offerer can start using the same 5-tuple for sending and receiving media, associated with the group, between the entities.

If the SDP Answer does not include a session-level SDP 'group' attribute, with a "BUNDLE" value, the SDP Offerer cannot use the same 5-tuple for media associated with multiple "m=" lines.

If the SDP Answerer indicates that it will not use bundled media, the SDP Offerer will still use the single port number value for each "m=" line associated with the offered "BUNDLE" group, and it will normally be able to separate each individual media. The default mechanism for separating media received on a single IP address and port doing this is by using a 5-tuple based mapping for each individual media. If the SDP Offerer is aware of the Synchronization Source (SSRC) [RFC3550] values that the SDP Answerer will use in the media it sends, and the SSRC values will be unique for each media, the SDP Offerer can separate media based on the SSRC values.

NOTE: Assuming symmetric media is used, the SDP Offerer can use the port information from the SDP Answer in order to create the 5-tuple mapping for each media.

If the SDP Offerer is not able to separate multiple media received on a single port, it MUST send a new SDP Offer, without offering bundled media, where a separate port number value is provided for each "m=" line of the SDP Offer.

If an SDP Offer, offering a "BUNDLE" group, and the SDP Offerer has reasons to believe that the rejection is due to the usage of a single port number value for multiple "m=" lines, the SDP Offerer SHOULD send a new SDP Offer, without a "BUNDLE" group, where a separate port number value is provide for each "m=" line of the SDP offer.

6.3. SDP Answerer Procedures

When an SDP Answerer receives an SDP Offer, which offers bundled media, and the SDP Answerer accepts the offered bundle group, the SDP Answerer MUST create an SDP Answer according to the procedures in Section 6.1.

If the SDP Answerer does not accept the "BUNDLE" group in the SDP Offer, it MUST NOT include a session-level 'group' attribute, with a "BUNDLE" value, in the associated SDP Answer. In addition, the SDP Answerer MUST provide separate port number values for each "m=" line of the SDP Answer.

6.4. Bundled SDP Information

6.4.1. General

This section describes how SDP information, given for each media description, is calculated into a single value for a "BUNDLE" group.

6.4.2. Bandwidth (b=)

The total proposed bandwidth is the sum of the proposed bandwidth for each "m=" line associated with a negotiated BUNDLE group.

7. Single vs Multiple RTP Sessions

7.1. General

When entities negotiate the usage of bundled media, the default assumption is that all media associated with the bundled media will form a single RTP session.

The usage of multiple RTP Sessions within a "BUNDLE" group is outside the scope of this specification. Other specification needs to extend the mechanism in order to allow negotiation of such bundle groups.

7.2. Single RTP Session

When a single RTP Session is used, media associated with all "m=" lines part of a bundle group share a single SSRC [RFC3550] numbering

space.

In addition, the following rules and restrictions apply for a single RTP Session:

- o - The dynamic payload type values used in the "m=" lines MUST NOT overlap.
- o - The "proto" value in each "m=" line MUST be identical (e.g. RTP/AVPF).
- o - A given SSRC SHOULD NOT transmit RTP packets using payload types that originates from different "m=" lines.

NOTE: The last bullet above is to avoid sending multiple media types from the same SSRC. If transmission of multiple media types are done with time overlap RTP and RTCP fails to function. Even if done in proper sequence this causes RTP Timestamp rate switching issues [ref to draft-ietf-avtext-multiple-clock-rates].

8. Usage With ICE

8.1. General

This section describes how to use the "BUNDLE" grouping mechanism together with the Interactive Connectivity Establishment (ICE) mechanism [RFC5245].

8.2. Candidates

When an ICE-enabled SDP Offerer sends an SDP offer, it MUST include ICE candidates for each "m=" line associated with a "BUNDLE" group. The candidate values MUST be identical for each "m=" line associated with the group. This rule applies also to subsequent SDP Offers, when the usage of bundled media has already been negotiated.

When an ICE-enabled SDP Answerer receives an SDP Offer, offering a "BUNDLE" group and ICE, if the SDP Answerer enables ICE, MUST include ICE candidates for each "m=" line of the SDP Answer. This also applies for "m=" lines that are part of a "BUNDLE" group, in which case the candidate values MUST be identical for each "m=" line associated with the group. This rule applies also to subsequent SDP Answers, when the usage of bundled media has already been negotiated.

Once the usage of bundled media has been negotiated, ICE connectivity checks and keep-alives only needs to be performed for the whole "BUNDLE" group, instead of for each individual m= line associated with the group.

9. Security Considerations

TBA

10. Example

The example below shows an SDP Offer, where bundled media is offered. The example also shows two SDP Answer alternatives: one where bundled media is accepted, and one where bundled media is rejected (or, not even supported) by the SDP Answerer.

SDP Offer (Bundled media offered)

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.com
s=
c=IN IP4 host.atlanta.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 10000 RTP/AVP 31 32
a=mid:bar
b=AS:1000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

SDP Answer (Bundled media accepted)

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.com
s=
c=IN IP4 host.biloxi.com
t=0 0
a=group:BUNDLE foo bar
m=audio 20000 RTP/AVP 0
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
m=video 20000 RTP/AVP 32
a=mid:bar
b=AS:1000
```

a=rtpmap:32 MPV/90000

SDP Answer (Bundled media not accepted)

v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.com
s=
c=IN IP4 host.biloxi.com
t=0 0
m=audio 20000 RTP/AVP 0
b=AS:200
a=rtpmap:0 PCMU/8000
m=video 30000 RTP/AVP 32
b=AS:1000
a=rtpmap:32 MPV/90000

SDP Offer with ICE (Bundled media offered)

v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.com
s=
c=IN IP4 host.atlanta.com
t=0 0
a=group:BUNDLE foo bar
m=audio 10000 RTP/AVP 0 8 97
a=mid:foo
b=AS:200
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=candidate:1 1 UDP 1694498815 host.atlanta.com 10000 typ host
m=video 10000 RTP/AVP 31 32
a=mid:bar
b=AS:1000
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=candidate:1 1 UDP 1694498815 host.atlanta.com 10000 typ host

11. IANA Considerations

This document requests IANA to register the new SDP Grouping semantic extension called BUNDLE.

12. Acknowledgements

The usage of the SDP grouping mechanism is based on a similar alternative proposed by Harald Alvestrand. The SDP examples are also modified versions from the ones in the Alvestrand proposal.

Thanks to the nice flight crew on AY 021 for providing good sparkling wine, and a nice working atmosphere, for working on this draft.

13. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-00

- o No changes. New version due to expiration.

Changes from draft-holmberg-mmusic-sdp-multiplex-negotiation-00

- o Draft name changed.
- o Harald Alvestrand added as co-author.
- o "Multiplex" terminology changed to "bundle".
- o Added text about single versus multiple RTP Sessions.
- o Added reference to RFC 3550.

14. References

14.1. 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.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

14.2. Informative References

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Harald Tveit Alvestrand
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: harald@alvestrand.no

MMUSIC Working Group
Internet-Draft
Updates: 3264,5888,7941 (if approved)
Intended status: Standards Track
Expires: June 18, 2019

C. Holmberg
Ericsson
H. Alvestrand
Google
C. Jennings
Cisco
December 15, 2018

Negotiating Media Multiplexing Using the Session Description Protocol
(SDP)
draft-ietf-mmusic-sdp-bundle-negotiation-54.txt

Abstract

This specification defines a new Session Description Protocol (SDP) Grouping Framework extension, 'BUNDLE'. The extension can be used with the SDP Offer/Answer mechanism to negotiate the usage of a single transport (5-tuple) for sending and receiving media described by multiple SDP media descriptions ("m=" sections). Such transport is referred to as a BUNDLE transport, and the media is referred to as bundled media. The "m=" sections that use the BUNDLE transport form a BUNDLE group.

This specification updates RFC 3264, to also allow assigning a zero port value to a "m=" section in cases where the media described by the "m=" section is not disabled or rejected.

This specification updates RFC 5888, to also allow an SDP 'group' attribute to contain an identification-tag that identifies a "m=" section with the port set to zero.

This specification defines a new RTP Control Protocol (RTCP) source description (SDS) item and a new RTP header extension that can be used to correlate bundled RTP/RTCP packets with their appropriate "m=" section.

This specification updates RFC 7941, by adding an exception, for the MID RTP header extension, to the requirement regarding protection of an SDS RTP header extension carrying an SDS item for the MID RTP header extension.

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 June 18, 2019.

Copyright Notice

Copyright (c) 2018 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
1.1. Background	4
1.2. BUNDLE Mechanism	4
1.3. Protocol Extensions	5
2. Terminology	6
3. Conventions	8
4. Applicability Statement	8

5.	SDP Grouping Framework BUNDLE Extension	8
6.	SDP 'bundle-only' Attribute	9
7.	SDP Offer/Answer Procedures	10
7.1.	Generic SDP Considerations	10
7.1.1.	Connection Data (c=)	10
7.1.2.	Bandwidth (b=)	11
7.1.3.	Attributes (a=)	11
7.2.	Generating the Initial SDP Offer	12
7.2.1.	Suggesting the Offerer tagged 'm=' section	13
7.2.2.	Example: Initial SDP Offer	13
7.3.	Generating the SDP Answer	14
7.3.1.	Answerer Selection of tagged 'm=' sections	16
7.3.2.	Moving A Media Description Out Of A BUNDLE Group	16
7.3.3.	Rejecting a Media Description in a BUNDLE Group	17
7.3.4.	Example: SDP Answer	18
7.4.	Offerer Processing of the SDP Answer	19
7.5.	Modifying the Session	19
7.5.1.	Adding a Media Description to a BUNDLE group	20
7.5.2.	Moving a Media Description Out of a BUNDLE Group	21
7.5.3.	Disabling a Media Description in a BUNDLE Group	21
8.	Protocol Identification	22
8.1.	STUN, DTLS, SRTP	22
9.	RTP Considerations	23
9.1.	Single RTP Session	23
9.1.1.	Payload Type (PT) Value Reuse	24
9.2.	Associating RTP/RTCP Streams with the Correct SDP Media Description	24
9.3.	RTP/RTCP Multiplexing	30
9.3.1.	SDP Offer/Answer Procedures	30
10.	ICE Considerations	32
11.	DTLS Considerations	33
12.	RTP Header Extensions Consideration	34
13.	Update to RFC 3264	34
13.1.	Original text of section 5.1 (2nd paragraph) of RFC 3264	34
13.2.	New text replacing section 5.1 (2nd paragraph) of RFC 3264	35
13.3.	Original text of section 8.4 (6th paragraph) of RFC 3264	35
13.4.	New text replacing section 8.4 (6th paragraph) of RFC 3264	35
14.	Update to RFC 5888	36
14.1.	Original text of section 9.2 (3rd paragraph) of RFC 5888	36
14.2.	New text replacing section 9.2 (3rd paragraph) of RFC 5888	36
15.	RTP/RTCP extensions for identification-tag transport	36
15.1.	RTCP MID SDES Item	37
15.2.	RTP SDES Header Extension For MID	38
16.	IANA Considerations	38
16.1.	New SDES item	38

16.2.	New RTP SDES Header Extension URI	39
16.3.	New SDP Attribute	39
16.4.	New SDP Group Semantics	40
17.	Security Considerations	40
18.	Examples	41
18.1.	Example: Tagged m= Section Selections	41
18.2.	Example: BUNDLE Group Rejected	43
18.3.	Example: Offerer Adds a Media Description to a BUNDLE Group	45
18.4.	Example: Offerer Moves a Media Description Out of a BUNDLE Group	46
18.5.	Example: Offerer Disables a Media Description Within a BUNDLE Group	48
19.	Acknowledgements	50
20.	Change Log	50
21.	References	61
21.1.	Normative References	61
21.2.	Informative References	64
Appendix A.	Design Considerations	65
A.1.	UA Interoperability	65
A.2.	Usage of Port Number Value Zero	67
A.3.	B2BUA And Proxy Interoperability	67
A.3.1.	Traffic Policing	68
A.3.2.	Bandwidth Allocation	68
A.4.	Candidate Gathering	68
Authors' Addresses	69

1. Introduction

1.1. Background

When the SDP offer/answer mechanism [RFC3264] is used to negotiate the establishment of multimedia communication sessions, if separate transports (5-tuples) are negotiated for each individual media stream, each transport consumes additional resources (especially when Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] is used). For this reason, it is attractive to use a single transport for multiple media streams.

1.2. BUNDLE Mechanism

This specification defines a way to use a single transport (BUNDLE transport) for sending and receiving media (bundled media) described by multiple SDP media descriptions ("m=" sections). The address:port combination used by an endpoint for sending and receiving bundled media is referred to as the BUNDLE address:port. The set of SDP attributes that are applied to each "m=" section within a BUNDLE group is referred to as BUNDLE attributes. The same BUNDLE transport

is used for sending and receiving bundled media, which means that the symmetric Real-time Transport Protocol (RTP) mechanism [RFC4961] is always used for RTP-based bundled media.

This specification defines a new SDP Grouping Framework [RFC5888] extension called 'BUNDLE'. The extension can be used with the Session Description Protocol (SDP) Offer/Answer mechanism [RFC3264] to negotiate which "m=" sections will become part of a BUNDLE group. In addition, the offerer and answerer [RFC3264] use the BUNDLE extension to negotiate the BUNDLE addresses:ports (offerer BUNDLE address:port and answerer BUNDLE address:port) and the set of BUNDLE attributes (offerer BUNDLE attributes and answerer BUNDLE attributes) that will be applied to each "m=" section within the BUNDLE group.

The use of a BUNDLE transport allows the usage of a single set of Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] candidates for the whole BUNDLE group.

A given BUNDLE address:port MUST only be associated with a single BUNDLE group. If an SDP offer or answer contains multiple BUNDLE groups, the procedures in this specification apply to each group independently. All RTP-based bundled media associated with a given BUNDLE group belong to a single RTP session [RFC3550].

The BUNDLE extension is backward compatible. Endpoints that do not support the extension are expected to generate offers and answers without an SDP 'group:BUNDLE' attribute, and are expected to assign a unique address:port to each "m=" section within an offer and answer, according to the procedures in [RFC4566] and [RFC3264].

1.3. Protocol Extensions

In addition to defining the new SDP Grouping Framework extension, this specification defines the following protocol extensions and RFC updates:

- o The specification defines a new SDP attribute, 'bundle-only', which can be used to request that a specific "m=" section (and the associated media) is used only if kept within a BUNDLE group.
- o The specification updates RFC 3264 [RFC3264], to also allow assigning a zero port value to a "m=" section in cases where the media described by the "m=" section is not disabled or rejected.
- o The specification defines a new RTP Control Protocol (RTCP) [RFC3550] source description (SDS) item, 'MID', and a new RTP SDS header extension that can be used to associate RTP streams with "m=" sections.

- o The specification updates [RFC7941], by adding an exception, for the MID RTP header extension, to the requirement regarding protection of an SDES RTP header extension carrying an SDES item for the MID RTP header extension.

2. Terminology

- o "m=" section: SDP bodies contain one or more media descriptions, referred to as "m=" sections. Each "m=" section is represented by an SDP "m=" line, and zero or more SDP attributes associated with the "m=" line. A local address:port combination is assigned to each "m=" section.
- o 5-tuple: A collection of the following values: source address, source port, destination address, destination port, and transport-layer protocol.
- o Unique address:port: An address:port combination that is assigned to only one "m=" section in an offer or answer.
- o Offerer BUNDLE-tag: The first identification-tag in a given SDP 'group:BUNDLE' attribute identification-tag list in an offer.
- o Answerer BUNDLE-tag: The first identification-tag in a given SDP 'group:BUNDLE' attribute identification-tag list in an answer.
- o Suggested offerer tagged "m=" section: The bundled "m=" section identified by the offerer BUNDLE-tag in an initial BUNDLE offer, before a BUNDLE group has been negotiated.
- o Offerer tagged "m=" section: The bundled "m=" section identified by the offerer BUNDLE-tag in a subsequent offer. The "m=" section contains characteristics (offerer BUNDLE address:port and offerer BUNDLE attributes) applied to each "m=" section within the BUNDLE group.
- o Answerer tagged "m=" section: The bundled "m=" section identified by the answerer BUNDLE-tag in an answer (initial BUNDLE answer or subsequent). The "m=" section contains characteristics (answerer BUNDLE address:port and answerer BUNDLE attributes) applied to each "m=" section within the BUNDLE group.
- o BUNDLE address:port: An address:port combination that an endpoint uses for sending and receiving bundled media.
- o Offerer BUNDLE address:port: the address:port combination used by the offerer for sending and receiving media.

- o Answerer BUNDLE address:port: the address:port combination used by the answerer for sending and receiving media.
- o BUNDLE attributes: IDENTICAL and TRANSPORT multiplexing category SDP attributes. Once a BUNDLE group has been created, the attribute values apply to each bundled "m=" section within the BUNDLE group.
- o Offerer BUNDLE attributes: IDENTICAL and TRANSPORT multiplexing category SDP attributes included in the offerer tagged "m=" section.
- o Answerer BUNDLE attributes: IDENTICAL and TRANSPORT multiplexing category SDP attributes included in the answerer tagged "m=" section.
- o BUNDLE transport: The transport (5-tuple) used by all media described by the "m=" sections within a BUNDLE group.
- o BUNDLE group: A set of bundled "m=" sections, created using an SDP Offer/Answer exchange, which uses a single BUNDLE transport, and a single set of BUNDLE attributes, for sending and receiving all media (bundled media) described by the set of "m=" sections. The same BUNDLE transport is used for sending and receiving bundled media.
- o Bundled "m=" section: An "m=" section, whose identification-tag is placed in an SDP 'group:BUNDLE' attribute identification-tag list in an offer or answer.
- o Bundle-only "m=" section: A bundled "m=" section that contains an SDP 'bundle-only' attribute.
- o Bundled media: All media associated with a given BUNDLE group.
- o Initial BUNDLE offer: The first offer, within an SDP session (e.g. a SIP dialog when the Session Initiation Protocol (SIP) [RFC3261] is used to carry SDP), in which the offerer indicates that it wants to negotiate a given BUNDLE group.
- o Initial BUNDLE answer: The answer to an initial BUNDLE offer in which the offerer indicates that it wants to negotiate a BUNDLE group, and where the answerer accepts the creation of the BUNDLE group. The BUNDLE group is created once the answerer sends the initial BUNDLE answer.
- o Subsequent offer: An offer which contains a BUNDLE group that has been created as part of a previous offer/answer exchange.

- o Subsequent answer: An answer to a subsequent offer.
- o Identification-tag: A unique token value that is used to identify an "m=" section. The SDP 'mid' attribute [RFC5888] in an "m=" section carries the unique identification-tag assigned to that "m=" section. The session-level SDP 'group' attribute [RFC5888] carries a list of identification-tags, identifying the "m=" sections associated with that particular 'group' attribute.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. Applicability Statement

The mechanism in this specification only applies to the Session Description Protocol (SDP) [RFC4566], when used together with the SDP offer/answer mechanism [RFC3264]. Declarative usage of SDP is out of scope of this document, and is thus undefined.

5. SDP Grouping Framework BUNDLE Extension

This section defines a new SDP Grouping Framework [RFC5888] extension, 'BUNDLE'. The BUNDLE extension can be used with the SDP Offer/Answer mechanism to negotiate a set of "m=" sections that will become part of a BUNDLE group. Within a BUNDLE group, each "m=" section uses a BUNDLE transport for sending and receiving bundled media. Each endpoint uses a single address:port combination for sending and receiving the bundled media.

The BUNDLE extension is indicated using an SDP 'group' attribute with a semantics value [RFC5888] of "BUNDLE". An identification-tag is assigned to each bundled "m=" section, and each identification-tag is listed in the SDP 'group:BUNDLE' attribute identification-tag list. Each "m=" section whose identification-tag is listed in the identification-tag list is associated with a given BUNDLE group.

SDP bodies can contain multiple BUNDLE groups. Any given bundled "m=" section MUST NOT be associated with more than one BUNDLE group at any given time.

NOTE: The order of the "m=" sections listed in the SDP 'group:BUNDLE' attribute identification-tag list does not have to be the same as the order in which the "m=" sections occur in the SDP.

The multiplexing category [I-D.ietf-mmusic-sdp-mux-attributes] for the 'group:BUNDLE' attribute is 'NORMAL'.

Section 7 defines the detailed SDP Offer/Answer procedures for the BUNDLE extension.

6. SDP 'bundle-only' Attribute

This section defines a new SDP media-level attribute [RFC4566], 'bundle-only'. 'bundle-only' is a property attribute [RFC4566], and hence has no value.

In order to ensure that an answerer that does not support the BUNDLE extension always rejects a bundled "m=" section in an offer, the offerer can assign a zero port value to the "m=" section. According to [RFC3264] an answerer will reject such an "m=" section. By including an SDP 'bundle-only' attribute in a bundled "m=" section, the offerer can request that the answerer accepts the "m=" section only if the answerer supports the BUNDLE extension, and if the answerer keeps the "m=" section within the associated BUNDLE group.

Name: bundle-only

Value: N/A

Usage Level: media

Charset Dependent: no

Example:

a=bundle-only

Once the offerer tagged "m=" section and the answerer tagged "m=" section have been selected, an offerer and answerer will include an SDP 'bundle-only' attribute in, and assign a zero port value to, every other bundled "m=" section.

The usage of the 'bundle-only' attribute is only defined for a bundled "m=" section with a zero port value. Other usage is unspecified.

Section 7 defines the detailed SDP Offer/Answer procedures for the 'bundle-only' attribute.

7. SDP Offer/Answer Procedures

This section describes the SDP Offer/Answer [RFC3264] procedures for:

- o Negotiating a BUNDLE group; and
- o Suggesting and selecting the tagged "m=" sections (offerer tagged "m=" section and answerer tagged "m=" section); and
- o Adding an "m=" section to a BUNDLE group; and
- o Moving an "m=" section out of a BUNDLE group; and
- o Disabling an "m=" section within a BUNDLE group.

The generic rules and procedures defined in [RFC3264] and [RFC5888] also apply to the BUNDLE extension. For example, if an offer is rejected by the answerer, the previously negotiated addresses:ports, SDP parameters and characteristics (including those associated with a BUNDLE group) apply. Hence, if an offerer generates an offer in order to negotiate a BUNDLE group, and the answerer rejects the offer, the BUNDLE group is not created.

The procedures in this section are independent of the media type or "m=" line proto value assigned to a bundled "m=" section. Section 9 defines additional considerations for RTP based media. Section 6 defines additional considerations for the usage of the SDP 'bundle-only' attribute. Section 10 defines additional considerations for the usage of Interactive Connectivity Establishment (ICE) [I-D.ietf-ice-rfc5245bis] mechanism.

Offers and answers can contain multiple BUNDLE groups. The procedures in this section apply independently to a given BUNDLE group.

7.1. Generic SDP Considerations

This section describes generic restrictions associated with the usage of SDP parameters within a BUNDLE group. It also describes how to calculate a value for the whole BUNDLE group, when parameter and attribute values have been assigned to each bundled "m=" section.

7.1.1. Connection Data (c=)

The "c=" line nettype value [RFC4566] associated with a bundled "m=" section MUST be 'IN'.

The "c=" line addrtype value [RFC4566] associated with a bundled "m=" section MUST be 'IP4' or 'IP6'. The same value MUST be associated with each "m=" section.

NOTE: Extensions to this specification can specify usage of the BUNDLE mechanism for other nettype and addrtype values than the ones listed above.

7.1.2. Bandwidth (b=)

An offerer and answerer MUST use the rules and restrictions defined in [I-D.ietf-mmusic-sdp-mux-attributes] for associating the SDP bandwidth (b=) line with bundled "m=" sections.

7.1.3. Attributes (a=)

An offerer and answerer MUST include SDP attributes in every bundled "m=" section where applicable, following the normal offer/answer procedures for each attribute, with the following exceptions:

- o In the initial BUNDLE offer, the offerer MUST NOT include IDENTICAL and TRANSPORT multiplexing category SDP attributes (BUNDLE attributes) in bundle-only "m=" sections. The offerer MUST include such attributes in all other bundled "m=" sections. In the initial BUNDLE offer each bundled "m=" line can contain a different set of BUNDLE attributes, and attribute values. Once the offerer tagged "m=" section has been selected, the BUNDLE attributes contained in the offerer tagged "m=" section will apply to each bundled "m=" section within the BUNDLE group.
- o In a subsequent offer, or in an answer (initial or subsequent), the offerer and answerer MUST include IDENTICAL and TRANSPORT multiplexing category SDP attributes (BUNDLE attributes) only in the tagged "m=" section (offerer tagged "m=" section or answerer tagged "m=" section). The offerer and answerer MUST NOT include such attributes in any other bundled "m=" section. The BUNDLE attributes contained in the tagged "m=" section will apply to each bundled "m=" section within the BUNDLE group.
- o In an offer (initial BUNDLE offer or subsequent), or in an answer (initial BUNDLE answer or subsequent), the offerer and answerer MUST include SDP attributes of other categories than IDENTICAL and TRANSPORT in each bundled "m=" section that a given attribute applies to. Each bundled "m=" line can contain a different set of such attributes, and attribute values, as such attributes only apply to the given bundled "m=" section in which they are included.

NOTE: A consequence of the rules above is that media-specific IDENTICAL and TRANSPORT multiplexing category SDP attributes which are applicable only to some of the bundled "m=" sections within the BUNDLE group might appear in the tagged "m=" section for which they are not applicable. For instance, the tagged "m=" section might contain an SDP 'rtcp-mux' attribute even if the tagged "m=" section does not describe RTP-based media (but another bundled "m=" section within the BUNDLE group does describe RTP-based media).

7.2. Generating the Initial SDP Offer

The procedures in this section apply to the first offer, within an SDP session (e.g. a SIP dialog when the Session Initiation Protocol (SIP) [RFC3261] is used to carry SDP), in which the offerer indicates that it wants to negotiate a given BUNDLE group. This could occur in the initial offer, or in a subsequent offer, of the SDP session.

When an offerer generates an initial BUNDLE offer, in order to negotiate a BUNDLE group, it MUST:

- o Assign a unique address:port to each bundled "m=" section, following the procedures in [RFC3264], excluding any bundle-only "m=" sections (see below); and
- o Pick a bundled "m=" section as the suggested offerer tagged "m=" section [Section 7.2.1]; and
- o Include SDP attributes in the bundled "m=" sections following the rules in [Section 7.1.3]; and
- o Include an SDP 'group:BUNDLE' attribute in the offer; and
- o Place the identification-tag of each bundled "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list. The offerer BUNDLE-tag indicates the suggested offerer tagged "m=" section.

NOTE: When the offerer assigns unique addresses:ports to multiple bundled "m=" sections, the offerer needs to be prepared to receive bundled media on each unique address:port, until it receives the associated answer and finds out which bundled "m=" section (and associated address:port combination) the answerer has selected as the offerer tagged "m=" section.

If the offerer wants to request that the answerer accepts a given bundled "m=" section only if the answerer keeps the "m=" section within the negotiated BUNDLE group, the offerer MUST:

- o Include an SDP 'bundle-only' attribute [Section 7.2.1] in the "m=" section; and
- o Assign a zero port value to the "m=" section.

NOTE: If the offerer assigns a zero port value to a bundled "m=" section, but does not include an SDP 'bundle-only' attribute in the "m=" section, it is an indication that the offerer wants to disable the "m=" section [Section 7.5.3].

[Section 7.2.2] and [Section 18.1] show an example of an initial BUNDLE offer.

7.2.1. Suggesting the Offerer tagged 'm=' section

In the initial BUNDLE offer, the bundled "m=" section indicated by the offerer BUNDLE-tag is the suggested offerer tagged "m=" section. The address:port combination associated with the "m=" section will be used by the offerer for sending and receiving bundled media if the answerer selects the "m=" section as the offerer tagged "m=" section [Section 7.3.1]. In addition, if the answerer selects the "m=" section as the offerer tagged "m=" section, the BUNDLE attributes included in the "m=" section will be applied to each "m=" section within the negotiated BUNDLE group.

The offerer MUST NOT suggest a bundle-only "m=" section as the offerer tagged "m=" section.

It is RECOMMENDED that the suggested offerer tagged "m=" section is a bundled "m=" section that the offerer believes it is unlikely that the answerer will reject, or move out of the BUNDLE group. How such assumption is made is outside the scope of this document.

7.2.2. Example: Initial SDP Offer

The example shows an initial BUNDLE offer. The offer includes two "m=" sections in the offer, and suggests that both "m=" sections are included in a BUNDLE group. The audio "m=" section is the suggested offerer tagged "m=" section, indicated by placing the identification-tag associated with the "m=" section (offerer BUNDLE-tag) first in the SDP group:BUNDLE attribute identification-id list.

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP6 2001:db8::3
s=
c=IN IP6 2001:db8::3
t=0 0
a=group:BUNDLE foo bar

m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

7.3. Generating the SDP Answer

When an answerer generates an answer (initial BUNDLE answer or subsequent) that contains a BUNDLE group the following general SDP grouping framework restrictions, defined in [RFC5888], also apply to the BUNDLE group:

- o The answerer is only allowed to include a BUNDLE group in an initial BUNDLE answer if the offerer requested the BUNDLE group to be created in the corresponding initial BUNDLE offer; and
- o The answerer is only allowed to include a BUNDLE group in a subsequent answer if the corresponding subsequent offer contains a previously negotiated BUNDLE group; and
- o The answerer is only allowed to include a bundled "m=" section in an answer if the "m=" section was indicated as bundled in the corresponding offer; and

- o The answerer is only allowed to include a bundled "m=" section in the same BUNDLE group as the bundled "m=" line in the corresponding offer.

In addition, when an answerer generates an answer (initial BUNDLE answer or subsequent) that contains a BUNDLE group, the answerer MUST:

- o In case of an initial BUNDLE answer, select the offerer tagged "m=" section using the procedures in Section 7.3.1. In case of a subsequent answer, the offerer tagged "m=" section is indicated in the corresponding subsequent offer, and MUST NOT be changed by the answerer; and
- o Select the answerer tagged "m=" section [Section 7.3.1]; and
- o Assign the answerer BUNDLE address:port to the answerer tagged "m=" section; and
- o Include an SDP 'bundle-only' attribute in, and assign a zero port value to, every other bundled "m=" section within the BUNDLE group; and
- o Include SDP attributes in the bundled "m=" sections following the rules in [Section 7.1.3]; and
- o Include an SDP 'group:BUNDLE' attribute in the answer; and
- o Place the identification-tag of each bundled "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list. The answerer BUNDLE-tag indicates the answerer tagged "m=" section [Section 7.3.1].

If the answerer does not want to keep an "m=" section within a BUNDLE group, it MUST:

- o Move the "m=" section out of the BUNDLE group [Section 7.3.2]; or
- o Reject the "m=" section [Section 7.3.3].

The answerer can modify the answerer BUNDLE address:port, add and remove SDP attributes, or modify SDP attribute values, in a subsequent answer. Changes to the answerer BUNDLE address:port and the answerer BUNDLE attributes will be applied to each bundled "m=" section within the BUNDLE group.

NOTE: If a bundled "m=" section in an offer contains a zero port value, but the "m=" section does not contain an SDP 'bundle-only'

attribute, it is an indication that the offerer wants to disable the "m=" section [Section 7.5.3].

7.3.1. Answerer Selection of tagged 'm=' sections

When the answerer selects the offerer tagged "m=" section, it first checks the suggested offerer tagged "m=" section [Section 7.2.1]. The answerer MUST check whether the "m=" section fulfils the following criteria:

- o The answerer will not move the "m=" section out of the BUNDLE group [Section 7.3.2]; and
- o The answerer will not reject the "m=" section [Section 7.3.3]; and
- o The "m=" section does not contain a zero port value.

If all of the criteria above are fulfilled, the answerer MUST select the "m=" section as the offerer tagged "m=" section, and MUST also mark the corresponding "m=" section in the answer as the answerer tagged "m=" section. In the answer the answerer BUNDLE-tag indicates the answerer tagged "m=" section.

If one or more of the criteria are not fulfilled, the answerer MUST pick the next identification-tag in the identification-tag list in the offer, and perform the same criteria check for the "m=" section indicated by that identification-tag. If there are no more identification-tags in the identification-tag list, the answerer MUST NOT create the BUNDLE group. Unless the answerer rejects the whole offer, the answerer MUST apply the answerer procedures for moving an "m=" section out of a BUNDLE group [Section 7.3.2] or rejecting an "m=" section within a BUNDLE group [Section 7.3.3] to every bundled "m=" section in the offer when creating the answer.

[Section 18.1] shows an example of an offerer BUNDLE address:port selection.

[Section 7.3.4] and [Section 18.1] show an example of an answerer tagged "m=" section selection.

7.3.2. Moving A Media Description Out Of A BUNDLE Group

When an answerer generates the answer, if the answerer wants to move a bundled "m=" section out of the negotiated BUNDLE group, the answerer MUST first check the following criteria:

- o In the corresponding offer, the "m=" section is within a previously negotiated BUNDLE group; and

- o In the corresponding offer, the "m=" section contains an SDP 'bundle-only' attribute.

If either criterium above is fulfilled the answerer can not move the "m=" section out of the BUNDLE group in the answer. The answerer can either reject the whole offer, reject each bundled "m=" section within the BUNDLE group [Section 7.3.3], or keep the "m=" section within the BUNDLE group in the answer and later create an offer where the "m=" section is moved out of the BUNDLE group [Section 7.5.2].

NOTE: One consequence of the rules above is that, once a BUNDLE group has been negotiated, a bundled "m=" section can not be moved out of the BUNDLE group in an answer. Instead an offer is needed.

When the answerer generates an answer, in which it moves a bundled "m=" section out of a BUNDLE group, the answerer:

- o MUST assign a unique address:port to the "m=" section; and
- o MUST include any applicable SDP attribute in the "m=" section, using the normal offer/answer procedures for the each Attributes; and
- o MUST NOT place the identification-tag associated with the "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group.
- o MUST NOT include an SDP 'bundle-only' attribute to the "m=" section; and

Because an answerer is not allowed to move an "m=" section from one BUNDLE group to another within an answer [Section 7.3], if the answerer wants to move an "m=" section from one BUNDLE group to another it MUST first move the "m=" section out of the current BUNDLE group, and then generate an offer where the "m=" section is added to another BUNDLE group [Section 7.5.1].

7.3.3. Rejecting a Media Description in a BUNDLE Group

When an answerer wants to reject a bundled "m=" section in an answer, it MUST first check the following criterion:

- o In the corresponding offer, the "m=" section is the offerer tagged "m=" section.

If the criterium above is fulfilled the answerer can not reject the "m=" section in the answer. The answerer can either reject the whole offer, reject each bundled "m=" section within the BUNDLE group, or

keep the "m=" section within the BUNDLE group in the answer and later create an offer where the "m=" section is disabled within the BUNDLE group [Section 7.5.3].

When an answerer generates an answer, in which it rejects a bundled "m=" section, the answerer:

- o MUST assign a zero port value to the "m=" section, according to the procedures in [RFC3264]; and
- o MUST NOT place the identification-tag associated with the "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group; and
- o MUST NOT include an SDP 'bundle-only' attribute in the "m=" section.

7.3.4. Example: SDP Answer

The example below shows an answer, based on the corresponding offer in [Section 7.2.2]. The answerer accepts both bundled "m=" sections within the created BUNDLE group. The audio "m=" section is the answerer tagged "m=" section, indicated by placing the identification-tag associated with the "m=" section (answerer BUNDLE-tag) first in the SDP group;BUNDLE attribute identification-id list. The answerer includes an SDP 'bundle-only' attribute in, and assigns a zero port value to, the video "m=" section.

SDP Answer

```
v=0
o=bob 2808844564 2808844564 IN IP6 2001:db8::1
s=
c=IN IP6 2001:db8::1
t=0 0
a=group:BUNDLE foo bar

m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid

m=video 0 RTP/AVP 32
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
```

7.4. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answer contains a BUNDLE group, the offerer MUST check that any bundled "m=" section in the answer was indicated as bundled in the corresponding offer. If there is no mismatch, the offerer MUST apply the properties (BUNDLE address:port, BUNDLE attributes etc) of the offerer tagged "m=" section (selected by the answerer [Section 7.3.1]) to each bundled "m=" section within the BUNDLE group.

NOTE: As the answerer might reject one or more bundled "m=" sections in an initial BUNDLE offer, or move a bundled "m=" section out of a BUNDLE group, a given bundled "m=" section in the offer might not be indicated as bundled in the corresponding answer.

If the answer does not contain a BUNDLE group, the offerer MUST process the answer as a normal answer.

7.5. Modifying the Session

When a BUNDLE group has previously been negotiated, and an offerer generates a subsequent offer, the offerer MUST:

- o Pick one bundled "m=" section as the offerer tagged "m=" section. The offerer can either pick the "m=" section that was previously selected by the answerer as the offerer tagged "m=" section, or pick another bundled "m=" section within the BUNDLE group; and
- o Assign a BUNDLE address:port (previously negotiated or newly suggest) to the offerer tagged "m=" section; and
- o Include an SDP 'bundle-only' attribute in, and assign a zero port value to, every other bundled "m=" section within the BUNDLE group; and
- o Include SDP attributes in the bundled "m=" sections following the rules in [Section 7.1.3]; and
- o Include an SDP 'group:BUNDLE' attribute in the offer; and
- o Place the identification-tag of each bundled "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list. The offerer BUNDLE-tag indicates the offerer tagged "m=" section.

The offerer MUST NOT pick a given bundled "m=" section as the offerer tagged "m=" section if:

- o The offerer wants to move the "m=" section out of the BUNDLE group [Section 7.5.2]; or
- o The offerer wants to disable the "m=" section [Section 7.5.3].

The offerer can modify the offerer BUNDLE address:port, add and remove SDP attributes, or modify SDP attribute values, in the subsequent offer. Changes to the offerer BUNDLE address:port and the offerer BUNDLE attributes will (if the offer is accepted by the answerer) be applied to each bundled "m=" section within the BUNDLE group.

7.5.1. Adding a Media Description to a BUNDLE group

When an offerer generates a subsequent offer, in which it wants to add a bundled "m=" section to a previously negotiated BUNDLE group, the offerer follows the procedures in Section 7.5. The offerer either picks the added "m=" section, or an "m=" section previously added to the BUNDLE group, as the offerer tagged "m=" section.

NOTE: As described in Section 7.3.2, the answerer can not move the added "m=" section out of the BUNDLE group in its answer. If the answer wants to move the "m=" section out of the BUNDLE group, it will have to first accept it into the BUNDLE group in the answer, and

then send a subsequent offer where the "m=" section is moved out of the BUNDLE group [Section 7.5.2].

7.5.2. Moving a Media Description Out of a BUNDLE Group

When an offerer generates a subsequent offer, in which it want to remove a bundled "m=" section from a BUNDLE group, the offerer:

- o MUST assign a unique address:port to the "m=" section; and
- o MUST include SDP attributes in the "m=" section following the normal offer/answer rules for each attribute; and
- o MUST NOT place the identification-tag associated with the "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group; and
- o MUST NOT assign an SDP 'bundle-only' attribute to the "m=" section.

For the other bundled "m=" sections within the BUNDLE group, the offerer follows the procedures in [Section 7.5].

An offerer MUST NOT move an "m=" section from one BUNDLE group to another within a single offer. If the offerer wants to move an "m=" section from one BUNDLE group to another it MUST first move the BUNDLE group out of the current BUNDLE group, and then generate a second offer where the "m=" section is added to another BUNDLE group [Section 7.5.1].

[Section 18.4] shows an example of an offer for moving an "m=" section out of a BUNDLE group.

7.5.3. Disabling a Media Description in a BUNDLE Group

When an offerer generates a subsequent offer, in which it want to disable a bundled "m=" section from a BUNDLE group, the offerer:

- o MUST assign a zero port value to the "m=" section, following the procedures in [RFC4566]; and
- o MUST NOT place the identification-tag associated with the "m=" section in the SDP 'group:BUNDLE' attribute identification-tag list associated with the BUNDLE group; and
- o MUST NOT assign an SDP 'bundle-only' attribute to the "m=" section.

For the other bundled "m=" sections within the BUNDLE group, the offerer follows the procedures in [Section 7.5].

[Section 18.5] shows an example of an offer and answer for disabling an "m=" section within a BUNDLE group.

8. Protocol Identification

Each "m=" section within a BUNDLE group MUST use the same transport-layer protocol. If bundled "m=" sections use different upper-layer protocols on top of the transport-layer protocol, there MUST exist a publicly available specification which describes a mechanism how to associate received data with the correct protocol for this particular protocol combination.

In addition, if received data can be associated with more than one bundled "m=" section, there MUST exist a publicly available specification which describes a mechanism for associating the received data with the correct "m=" section.

This document describes a mechanism to identify the protocol of received data among the STUN, DTLS and SRTP protocols (in any combination), when UDP is used as transport-layer protocol, but it does not describe how to identify different protocols transported on DTLS. While the mechanism is generally applicable to other protocols and transport-layer protocols, any such use requires further specification around how to multiplex multiple protocols on a given transport-layer protocol, and how to associate received data with the correct protocols.

8.1. STUN, DTLS, SRTP

Section 5.1.2 of [RFC5764] describes a mechanism to identify the protocol of a received packet among the STUN, DTLS and SRTP protocols (in any combination). If an offer or answer includes a bundled "m=" section that represents these protocols, the offerer or answerer MUST support the mechanism described in [RFC5764], and no explicit negotiation is required in order to indicate support and usage of the mechanism.

[RFC5764] does not describe how to identify different protocols transported on DTLS, only how to identify the DTLS protocol itself. If multiple protocols are transported on DTLS, there MUST exist a specification describing a mechanism for identifying each individual protocol. In addition, if a received DTLS packet can be associated with more than one "m=" section, there MUST exist a specification which describes a mechanism for associating the received DTLS packets with the correct "m=" section.

[Section 9.2] describes how to associate the packets in a received SRTP stream with the correct "m=" section.

9. RTP Considerations

9.1. Single RTP Session

All RTP-based media within a single BUNDLE group belong to a single RTP session [RFC3550].

Since a single BUNDLE transport is used for sending and receiving bundled media, the symmetric RTP mechanism [RFC4961] MUST be used for RTP-based bundled media.

Since a single RTP session is used for each BUNDLE group, all "m=" sections representing RTP-based media within a BUNDLE group will share a single SSRC numbering space [RFC3550].

The following rules and restrictions apply for a single RTP session:

- o A specific payload type value can be used in multiple bundled "m=" sections only if each codec associated with the payload type number shares an identical codec configuration [Section 9.1.1].
- o The proto value in each bundled RTP-based "m=" section MUST be identical (e.g., RTP/AVPF).
- o The RTP MID header extension MUST be enabled, by including an SDP 'extmap' attribute [RFC8285], with a 'urn:ietf:params:rtp-hdext:sdes:mid' URI value, in each bundled RTP-based "m=" section in every offer and answer.
- o A given SSRC MUST NOT transmit RTP packets using payload types that originate from different bundled "m=" sections.

NOTE: The last bullet above is to avoid sending multiple media types from the same SSRC. If transmission of multiple media types are done with time overlap, RTP and RTCP fail to function. Even if done in proper sequence this causes RTP Timestamp rate switching issues [RFC7160]. However, once an SSRC has left the RTP session (by sending an RTCP BYE packet), that SSRC can be reused by another source (possibly associated with a different bundled "m=" section) after a delay of 5 RTCP reporting intervals (the delay is to ensure the SSRC has timed out, in case the RTCP BYE packet was lost [RFC3550]).

[RFC7657] defines Differentiated Services (Diffserv) considerations for RTP-based bundled media sent using a mixture of Diffserv Codepoints.

9.1.1. Payload Type (PT) Value Reuse

Multiple bundled "m=" sections might describe RTP based media. As all RTP based media associated with a BUNDLE group belong to the same RTP session, in order for a given payload type value to be used inside more than one bundled "m=" section, all codecs associated with the payload type number MUST share an identical codec configuration. This means that the codecs MUST share the same media type, encoding name, clock rate and any parameter that can affect the codec configuration and packetization.

[I-D.ietf-mmusic-sdp-mux-attributes] lists SDP attributes, whose attribute values are required to be identical for all codecs that use the same payload type value.

9.2. Associating RTP/RTCP Streams with the Correct SDP Media Description

As described in [RFC3550], RTP packets are associated with RTP streams [RFC7656]. Each RTP stream is identified by an SSRC value, and each RTP packet includes an SSRC field that is used to associate the packet with the correct RTP stream. RTCP packets also use SSRCs to identify which RTP streams the packet relates to. However, a RTCP packet can contain multiple SSRC fields, in the course of providing feedback or reports on different RTP streams, and therefore can be associated with multiple such streams.

In order to be able to process received RTP/RTCP packets correctly, it MUST be possible to associate an RTP stream with the correct "m=" section, as the "m=" section and SDP attributes associated with the "m=" section contains information needed to process the packets.

As all RTP streams associated with a BUNDLE group use the same transport for sending and receiving RTP/RTCP packets, the local address:port combination part of the transport cannot be used to associate an RTP stream with the correct "m=" section. In addition, multiple RTP streams might be associated with the same "m=" section.

An offerer and answerer can inform each other which SSRC values they will use for an RTP stream by using the SDP 'ssrc' attribute [RFC5576]. However, an offerer will not know which SSRC values the answerer will use until the offerer has received the answer providing that information. Due to this, before the offerer has received the answer, the offerer will not be able to associate an RTP stream with the correct "m=" section using the SSRC value associated with the RTP

stream. In addition, the offerer and answerer may start using new SSRC values mid-session, without informing each other using the SDP 'ssrc' attribute.

In order for an offerer and answerer to always be able to associate an RTP stream with the correct "m=" section, the offerer and answerer using the BUNDLE extension MUST support the mechanism defined in Section 15, where the offerer and answerer insert the identification-tag associated with an "m=" section (provided by the remote peer) into RTP and RTCP packets associated with a BUNDLE group.

When using this mechanism, the mapping from an SSRC to an identification-tag is carried in RTP header extensions or RTCP SDES packets, as specified in Section 15. Since a compound RTCP packet can contain multiple RTCP SDES packets, and each RTCP SDES packet can contain multiple chunks, a single RTCP packet can contain several SSRC to identification-tag mappings. The offerer and answerer maintain tables used for routing that are updated each time an RTP/RTCP packet contains new information that affects how packets are to be routed.

However, some legacy implementations may not include this identification-tag in their RTP and RTCP traffic when using the BUNDLE mechanism, and instead use a payload type based mechanism to associate RTP streams with SDP "m=" sections. In this situation, each "m=" section needs to use unique payload type values, in order for the payload type to be a reliable indicator of the relevant "m=" section for the RTP stream. If an implementation fails to ensure unique payload type values it will be impossible to associate the RTP stream using that payload type value to a particular "m=" section. Note that when using the payload type to associate RTP streams with "m=" sections an RTP stream, identified by its SSRC, will be mapped to an "m=" section when the first packet of that RTP stream is received, and the mapping will not be changed even if the payload type used by that RTP stream changes. In other words, the SSRC cannot "move" to a different "m=" section simply by changing the payload type.

Applications can implement RTP stacks in many different ways. The algorithm below details one way that RTP streams can be associated with "m=" sections, but is not meant to be prescriptive about exactly how an RTP stack needs to be implemented. Applications MAY use any algorithm that achieves equivalent results to those described in the algorithm below.

To prepare to associate RTP streams with the correct "m=" section, the following steps MUST be followed for each BUNDLE group:

Construct a table mapping MID to "m=" section for each "m=" section in this BUNDLE group. Note that an "m=" section may only have one MID.

Construct a table mapping SSRCs of incoming RTP streams to "m=" section for each "m=" section in this BUNDLE group and for each SSRC configured for receiving in that "m=" section.

Construct a table mapping the SSRC of each outgoing RTP stream to "m=" section for each "m=" section in this BUNDLE group and for each SSRC configured for sending in that "m=" section.

Construct a table mapping payload type to "m=" section for each "m=" section in the BUNDLE group and for each payload type configured for receiving in that "m=" section. If any payload type is configured for receiving in more than one "m=" section in the BUNDLE group, do not include it in the table, as it cannot be used to uniquely identify an "m=" section.

Note that for each of these tables, there can only be one mapping for any given key (MID, SSRC, or PT). In other words, the tables are not multimaps.

As "m=" sections are added or removed from the BUNDLE groups, or their configurations are changed, the tables above MUST also be updated.

When an RTP packet is received, it MUST be delivered to the RTP stream corresponding to its SSRC. That RTP stream MUST then be associated with the correct "m=" section within a BUNDLE group, for additional processing, according to the following steps:

If the MID associated with the RTP stream is not in the table mapping MID to "m=" section, then the RTP stream is not decoded and the payload data is discarded.

If the packet has a MID, and the packet's extended sequence number is greater than that of the last MID update, as discussed in [RFC7941], Section 4.2.6, update the MID associated with the RTP stream to match the MID carried in the RTP packet, then update the mapping tables to include an entry that maps the SSRC of that RTP stream to the "m=" section for that MID.

If the SSRC of the RTP stream is in the incoming SSRC mapping table, check that the payload type used by the RTP stream matches a payload type included on the matching "m=" section. If so, associate the RTP stream with that "m=" section. Otherwise, the RTP stream is not decoded and the payload data is discarded.

If the payload type used by the RTP stream is in the payload type table, update the incoming SSRC mapping table to include an entry that maps the RTP stream's SSRC to the "m=" section for that payload type. Associate the RTP stream with the corresponding "m=" section.

Otherwise, mark the RTP stream as not for decoding and discard the payload.

If the RTP packet contains one or more contributing source (CSRC) identifiers, then each CSRC is looked up in the incoming SSRC table and a copy of the RTP packet is associated with the corresponding "m=" section for additional processing.

For each RTCP packet received (including each RTCP packet that is part of a compound RTCP packet), the packet is processed as usual by the RTP layer, then associated with the appropriate "m=" sections, and processed for the RTP streams represented by those "m=" sections. This routing is type-dependent, as each kind of RTCP packet has its own mechanism for associating it with the relevant RTP streams.

RTCP packets that cannot be associated with an appropriate "m=" section MUST still be processed as usual by the RTP layer, updating the metadata associated with the corresponding RTP streams. This situation can occur with certain multiparty RTP topologies, or when RTCP packets are sent containing a subset of the SDES information.

Additional rules for processing various types of RTCP packets are explained below.

If the RTCP packet is of type SDES, for each chunk in the packet whose SSRC is found in the incoming SSRC table, deliver a copy of the SDES packet to the "m=" section associated with that SSRC. In addition, for any SDES MID items contained in these chunks, if the MID is found in the table mapping MID to "m=" section, update the incoming SSRC table to include an entry that maps the RTP stream associated with the chunk's SSRC to the "m=" section associated with that MID, unless the packet is older than the packet that most recently updated the mapping for this SSRC, as discussed in [RFC7941], Section 4.2.6.

Note that if an SDES packet is received as part of a compound RTCP packet, the SSRC to "m=" section mapping might not exist until the SDES packet is handled (e.g., in the case where RTCP for a source is received before any RTP packets). Therefore, it can be beneficial for an implementation to delay RTCP packet routing, such that it either prioritizes processing of the SDES item to generate or update the mapping, or buffers the RTCP information

that needs to be routed until the SDES item(s) has been processed. If the implementation is unable to follow this recommendation, the consequence could be that some RTCP information from this particular RTCP compound packet is not provided to higher layers. The impact from this is likely minor, when this information relates to a future incoming RTP stream.

If the RTCP packet is of type BYE, it indicates that the RTP streams referenced in the packet are ending. Therefore, for each SSRC indicated in the packet that is found in the incoming SSRC table, first deliver a copy of the BYE packet to the "m=" section associated with that SSRC, then remove the entry for that SSRC from the incoming SSRC table after an appropriate delay to account for "straggler packets", as specified in [RFC3550], Section 6.2.1.

If the RTCP packet is of type SR or RR, for each report block in the report whose "SSRC of source" is found in the outgoing SSRC table, deliver a copy of the SR or RR packet to the "m=" section associated with that SSRC. In addition, if the packet is of type SR, and the sender SSRC for the packet is found in the incoming SSRC table, deliver a copy of the SR packet to the "m=" section associated with that SSRC.

If the implementation supports RTCP XR and the packet is of type XR, as defined in [RFC3611], for each report block in the report whose "SSRC of source" is found in the outgoing SSRC table, deliver a copy of the XR packet to the "m=" section associated with that SSRC. In addition, if the sender SSRC for the packet is found in the incoming SSRC table, deliver a copy of the XR packet to the "m=" section associated with that SSRC.

If the RTCP packet is a feedback message of type RTPFB or PSFB, as defined in [RFC4585], it will contain a media source SSRC, and this SSRC is used for routing certain subtypes of feedback messages. However, several subtypes of PSFB and RTPFB messages include target SSRC(s) in a section called Feedback Control Information (FCI). For these messages, the target SSRC(s) are used for routing.

If the RTCP packet is a feedback packet that does not include target SSRCs in its FCI section, and the media source SSRC is found in the outgoing SSRC table, deliver the feedback packet to the "m=" section associated with that SSRC. RTPFB and PSFB types that are handled in this way include:

Generic NACK: [RFC4585] (PT=RTPFB, FMT=1).

Picture Loss Indication (PLI): [RFC4585] (PT=PSFB, FMT=1).

Slice Loss Indication (SLI): [RFC4585] (PT=PSFB, FMT=2).

Reference Picture Selection Indication (RPSI): [RFC4585]
(PT=PSFB, FMT=3).

If the RTCP packet is a feedback message that does include target SSRC(s) in its FCI section, it can either be a request or a notification. Requests reference a RTP stream that is being sent by the message recipient, whereas notifications are responses to an earlier request, and therefore reference a RTP stream that is being received by the message recipient.

If the RTCP packet is a feedback request that includes target SSRC(s), for each target SSRC that is found in the outgoing SSRC table, deliver a copy of the RTCP packet to the "m=" section associated with that SSRC. PSFB and RTPFB types that are handled in this way include:

Full Intra Request (FIR): [RFC5104] (PT=PSFB, FMT=4).

Temporal-Spatial Trade-off Request (TSTR): [RFC5104] (PT=PSFB,
FMT=5).

H.271 Video Back Channel Message (VBCM): [RFC5104] (PT=PSFB,
FMT=7).

Temporary Maximum Media Bit Rate Request (TMMBR): [RFC5104]
(PT=RTPFB, FMT=3).

Layer Refresh Request (LRR): [I-D.ietf-avtext-lrr] (PT=PSFB,
FMT=10).

If the RTCP packet is a feedback notification that includes target SSRC(s), for each target SSRC that is found in the incoming SSRC table, deliver a copy of the RTCP packet to the "m=" section associated with the RTP stream with matching SSRC. PSFB and RTPFB types that are handled in this way include:

Temporal-Spatial Trade-off Notification (TSTN): [RFC5104]
(PT=PSFB, FMT=6). This message is a notification in response to a prior TSTR.

Temporary Maximum Media Bit Rate Notification (TMMBN): [RFC5104]
(PT=RTPFB, FMT=4). This message is a notification in response to a prior TMMBR, but can also be sent unsolicited.

If the RTCP packet is of type APP, then it is handled in an application specific manner. If the application does not recognise the APP packet, then it MUST be discarded.

9.3. RTP/RTCP Multiplexing

Within a BUNDLE group, the offerer and answerer MUST enable RTP/RTCP multiplexing [RFC5761] for the RTP-based bundled media (i.e., the same transport will be used for both RTP packets and RTCP packets). In addition, the offerer and answerer MUST support the SDP 'rtcp-mux-only' attribute [I-D.ietf-mmusic-mux-exclusive].

9.3.1. SDP Offer/Answer Procedures

This section describes how an offerer and answerer use the SDP 'rtcp-mux' attribute [RFC5761] and the SDP 'rtcp-mux-only' attribute [I-D.ietf-mmusic-mux-exclusive] to negotiate usage of RTP/RTCP multiplexing for RTP-based bundled media.

RTP/RTCP multiplexing only applies to RTP-based media. However, as described in Section 7.1.3, within an offer or answer the SDP 'rtcp-mux' and SDP 'rtcp-mux-only' attributes might be included in a bundled "m=" section for non-RTP-based media (if such "m=" section is the offerer tagged "m=" section or answerer tagged "m=" section).

9.3.1.1. Generating the Initial SDP BUNDLE Offer

When an offerer generates an initial BUNDLE offer, if the offer contains one or more bundled "m=" sections for RTP-based media (or, if there is a chance that "m=" sections for RTP-based media will later be added to the BUNDLE group), the offerer MUST include an SDP 'rtcp-mux' attribute [RFC5761] in each bundled "m=" section (excluding any bundle-only "m=" sections). In addition, the offerer MAY include an SDP 'rtcp-mux-only' attribute [I-D.ietf-mmusic-mux-exclusive] in one or more bundled "m=" sections for RTP-based media.

NOTE: Whether the offerer includes the SDP 'rtcp-mux-only' attribute depends on whether the offerer supports fallback to usage of a separate port for RTCP in case the answerer moves one or more "m=" sections for RTP-based media out of the BUNDLE group in the answer.

NOTE: If the offerer includes an SDP 'rtcp-mux' attribute in the bundled "m=" sections, but does not include an SDP 'rtcp-mux-only' attribute, the offerer can also include an SDP 'rtcp' attribute [RFC3605] in one or more RTP-based bundled "m=" sections in order to provide a fallback port for RTCP, as described in [RFC5761]. However, the fallback port will only be applied to "m=" sections for

RTP-based media that are moved out of the BUNDLE group by the answerer.

In the initial BUNDLE offer, the address:port combination for RTCP MUST be unique in each bundled "m=" section for RTP-based media (excluding a bundle-only "m=" section), similar to RTP.

9.3.1.2. Generating the SDP Answer

When an answerer generates an answer, if the answerer supports RTP-based media, and if a bundled "m=" section in the corresponding offer contained an SDP 'rtcp-mux' attribute, the answerer MUST enable usage of RTP/RTCP multiplexing, even if there currently are no bundled "m=" sections for RTP-based media within the BUNDLE group. The answerer MUST include an SDP 'rtcp-mux' attribute in the answerer tagged "m=" section, following the procedures for BUNDLE attributes [Section 7.1.3]. In addition, if the "m=" section that is selected as the offerer tagged "m=" section contained an SDP "rtcp-mux-only" attribute, the answerer MUST include an SDP "rtcp-mux-only" attribute in the answerer tagged "m=" section.

In an initial BUNDLE offer, if the suggested offerer tagged "m=" section contained an SDP 'rtcp-mux-only' attribute, the "m=" section was for RTP-based media, and the answerer does not accept the "m=" section in the created BUNDLE group, the answerer MUST either move the "m=" section out of the BUNDLE group [Section 7.3.2], include the attribute in the moved "m=" section and enable RTP/RTCP multiplexing for the media associated with the "m=" section, or reject the "m=" section [Section 7.3.3].

The answerer MUST NOT include an SDP 'rtcp' attribute in any bundled "m=" section in the answer. The answerer will use the port value of the tagged offerer "m=" section sending RTP and RTCP packets associated with RTP-based bundled media towards the offerer.

If the usage of RTP/RTCP multiplexing within a BUNDLE group has been negotiated in a previous offer/answer exchange, the answerer MUST include an SDP 'rtcp-mux' attribute in the answerer tagged "m=" section. It is not possible to disable RTP/RTCP multiplexing within a BUNDLE group.

9.3.1.3. Offerer Processing of the SDP Answer

When an offerer receives an answer, if the answerer has accepted the usage of RTP/RTCP multiplexing [Section 9.3.1.2], the answerer follows the procedures for RTP/RTCP multiplexing defined in [RFC5761]. The offerer will use the port value of the answerer

tagged "m=" section for sending RTP and RTCP packets associated with RTP-based bundled media towards the answerer.

NOTE: It is considered a protocol error if the answerer has not accepted the usage of RTP/RTCP multiplexing for RTP-based "m=" sections that the answerer included in the BUNDLE group.

9.3.1.4. Modifying the Session

When an offerer generates a subsequent offer, the offerer MUST include an SDP 'rtcp-mux' attribute in the offerer tagged "m=" section, following the procedures for IDENTICAL multiplexing category attributes in Section 7.1.3.

10. ICE Considerations

This section describes how to use the BUNDLE grouping extension together with the Interactive Connectivity Establishment (ICE) mechanism [I-D.ietf-ice-rfc5245bis].

The generic procedures for negotiating usage of ICE using SDP, defined in [I-D.ietf-mmusic-ice-sip-sdp], also apply to usage of ICE with BUNDLE, with the following exceptions:

- o When the BUNDLE transport has been established, ICE connectivity checks and keep-alives only need to be performed for the BUNDLE transport, instead of per individual bundled "m=" section within the BUNDLE group.
- o The generic SDP attribute offer/answer considerations [Section 7.1.3] also apply to ICE-related attributes. Therefore, when an offer sends an initial BUNDLE offer (in order to negotiate a BUNDLE group) the offerer include ICE-related media-level attributes in each bundled "m=" section (excluding any bundle-only "m=" section), and each "m=" section MUST contain unique ICE properties. When an answerer generates an answer (initial BUNDLE answer or subsequent) that contains a BUNDLE group, and when an offerer sends a subsequent offer that contains a BUNDLE group, ICE-related media-level attributes are only included in the tagged "m=" section (suggested offerer tagged "m=" section or answerer tagged "m=" section), and the ICE properties are applied to each bundled "m=" section within the BUNDLE group.

NOTE: Most ICE-related media-level SDP attributes belong to the TRANSPORT multiplexing category [I-D.ietf-mmusic-sdp-mux-attributes], and the generic SDP attribute offer/answer considerations for TRANSPORT multiplexing category apply to the attributes. However, in the case of ICE-related attributes, the same considerations also

apply to ICE-related media-level attributes that belong to other multiplexing categories.

NOTE: The following ICE-related media-level SDP attributes are defined in [I-D.ietf-mmusic-ice-sip-sdp]: 'candidate', 'remote-candidates', 'ice-mismatch', 'ice-ufrag', 'ice-pwd', and 'ice-pacing'.

Initially, before ICE has produced selected candidate pairs that will be used for media, there might be multiple transports established (if multiple candidate pairs are tested). Once ICE has selected candidate pairs, they form the BUNDLE transport.

Support and usage of ICE mechanism together with the BUNDLE extension is OPTIONAL, and the procedures in this section only apply when the ICE mechanism is used. Note that applications might mandate usage of the ICE mechanism even if the BUNDLE extension is not used.

NOTE: If the trickle ICE mechanism [I-D.ietf-mmusic-trickle-ice-sip] is used, an offerer and answerer might assign a port value of '9', and an IPv4 address of '0.0.0.0' (or, the IPv6 equivalent ':::') to multiple bundled "m=" sections in the initial BUNDLE offer. The offerer and answerer will follow the normal procedures for generating the offers and answers, including picking a bundled "m=" section as the suggested offerer tagged "m=" section, selecting the tagged "m=" sections etc. The only difference is that media can not be sent until one or more candidates have been provided. Once a BUNDLE group has been negotiated, trickled candidates associated with a bundled "m=" section will be applied to all bundled "m=" sections within the BUNDLE group.

11. DTLS Considerations

One or more media streams within a BUNDLE group might use the Datagram Transport Layer Security (DTLS) protocol [RFC6347] in order to encrypt the data, or to negotiate encryption keys if another encryption mechanism is used to encrypt media.

When DTLS is used within a BUNDLE group, the following rules apply:

- o There can only be one DTLS association [RFC6347] associated with the BUNDLE group; and
- o Each usage of the DTLS association within the BUNDLE group MUST use the same mechanism for determining which endpoints (the offerer or answerer) become DTLS client and DTLS server; and

- o Each usage of the DTLS association within the BUNDLE group MUST use the same mechanism for determining whether an offer or answer will trigger the establishment of a new DTLS association, or whether an existing DTLS association will be used; and
- o If the DTLS client supports DTLS-SRTP [RFC5764] it MUST include the 'use_srtp' extension [RFC5764] in the DTLS ClientHello message [RFC5764]. The client MUST include the extension even if the usage of DTLS-SRTP is not negotiated as part of the multimedia session (e.g., SIP session [RFC3261]).

NOTE: The inclusion of the 'use_srtp' extension during the initial DTLS handshake ensures that a DTLS renegotiation will not be required in order to include the extension, in case DTLS-SRTP encrypted media is added to the BUNDLE group later during the multimedia session.

12. RTP Header Extensions Consideration

When [RFC8285] RTP header extensions are used in the context of this specification, the identifier used for a given extension MUST identify the same extension across all the bundled media descriptions.

13. Update to RFC 3264

This section updates RFC 3264, in order to allow extensions to define the usage of a zero port value in offers and answers for other purposes than removing or disabling media streams. The following sections of RFC 3264 are updated:

- o Section 5.1 (Unicast Streams).
- o Section 8.4 (Putting a Unicast Media Stream on Hold).

13.1. Original text of section 5.1 (2nd paragraph) of RFC 3264

For recvonly and sendrecv streams, the port number and address in the offer indicate where the offerer would like to receive the media stream. For sendonly RTP streams, the address and port number indirectly indicate where the offerer wants to receive RTCP reports. Unless there is an explicit indication otherwise, reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer indicates that the stream is offered but MUST NOT be used. This has no useful semantics in an initial offer, but is allowed for reasons of completeness, since the answer can contain a zero port indicating a rejected stream

(Section 6). Furthermore, existing streams can be terminated by setting the port to zero (Section 8). In general, a port number of zero indicates that the media stream is not wanted.

13.2. New text replacing section 5.1 (2nd paragraph) of RFC 3264

For `recvonly` and `sendrecv` streams, the port number and address in the offer indicate where the offerer would like to receive the media stream. For `sendonly` RTP streams, the address and port number indirectly indicate where the offerer wants to receive RTCP reports. Unless there is an explicit indication otherwise, reports are sent to the port number one higher than the number indicated. The IP address and port present in the offer indicate nothing about the source IP address and source port of RTP and RTCP packets that will be sent by the offerer. A port number of zero in the offer by default indicates that the stream is offered but **MUST NOT** be used, but an extension mechanism might specify different semantics for the usage of a zero port value. Furthermore, existing streams can be terminated by setting the port to zero (Section 8). In general, a port number of zero by default indicates that the media stream is not wanted.

13.3. Original text of section 8.4 (6th paragraph) of RFC 3264

RFC 2543 [10] specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. Its usage for putting a call on hold is no longer recommended, since it doesn't allow for RTCP to be used with held streams, doesn't work with IPv6, and breaks with connection oriented media. However, it can be useful in an initial offer when the offerer knows it wants to use a particular set of media streams and formats, but doesn't know the addresses and ports at the time of the offer. Of course, when used, the port number **MUST NOT** be zero, which would specify that the stream has been disabled. An agent **MUST** be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP is to be sent to the peer.

13.4. New text replacing section 8.4 (6th paragraph) of RFC 3264

RFC 2543 [10] specified that placing a user on hold was accomplished by setting the connection address to 0.0.0.0. Its usage for putting a call on hold is no longer recommended, since it doesn't allow for RTCP to be used with held streams, doesn't work with IPv6, and breaks with connection oriented media. However, it can be useful in an initial offer when the offerer knows it wants to use a particular set of media streams and formats, but doesn't know the addresses and ports at the time of the offer. Of course, when used, the port number **MUST NOT** be zero, if it would specify that the stream has been disabled. However, an extension mechanism might specify different

semantics of the zero port number usage. An agent MUST be capable of receiving SDP with a connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP is to be sent to the peer.

14. Update to RFC 5888

This section updates RFC 5888 [RFC5888]), in order to allow extensions to allow an SDP 'group' attribute containing an identification-tag that identifies a "m=" section with the port set to zero Section 9.2 (Group Value in Answers) of RFC 5888 is updated.

14.1. Original text of section 9.2 (3rd paragraph) of RFC 5888

SIP entities refuse media streams by setting the port to zero in the corresponding "m" line. "a=group" lines MUST NOT contain identification-tags that correspond to "m" lines with the port set to zero.

14.2. New text replacing section 9.2 (3rd paragraph) of RFC 5888

SIP entities refuse media streams by setting the port to zero in the corresponding "m" line. "a=group" lines MUST NOT contain identification-tags that correspond to "m" lines with the port set to zero, but an extension mechanism might specify different semantics for including identification-tags that correspond to such "m=" lines.

15. RTP/RTCP extensions for identification-tag transport

SDP Offerers and Answerers [RFC3264] can associate identification-tags with "m=" sections within SDP Offers and Answers, using the procedures in [RFC5888]. Each identification-tag uniquely represents an "m=" section.

This section defines a new RTCP SDPS item [RFC3550], 'MID', which is used to carry identification-tags within RTCP SDPS packets. This section also defines a new RTP SDPS header extension [RFC7941], which is used to carry the 'MID' RTCP SDPS item in RTP packets.

The SDPS item and RTP SDPS header extension make it possible for a receiver to associate each RTP stream with a specific "m=" section, with which the receiver has associated an identification-tag, even if those "m=" sections are part of the same RTP session. The RTP SDPS header extension also ensures that the media recipient gets the identification-tag upon receipt of the first decodable media and is able to associate the media with the correct application.

A media recipient informs the media sender about the identification-tag associated with an "m=" section through the use of an 'mid'

attribute [RFC5888]. The media sender then inserts the identification-tag in RTCP and RTP packets sent to the media recipient.

NOTE: This text above defines how identification-tags are carried in SDP Offers and Answers. The usage of other signaling protocols for carrying identification-tags is not prevented, but the usage of such protocols is outside the scope of this document.

[RFC3550] defines general procedures regarding the RTCP transmission interval. The RTCP MID SDES item SHOULD be sent in the first few RTCP packets sent after joining the session, and SHOULD be sent regularly thereafter. The exact number of RTCP packets in which this SDES item is sent is intentionally not specified here, as it will depend on the expected packet loss rate, the RTCP reporting interval, and the allowable overhead.

The RTP SDES header extension for carrying the 'MID' RTCP SDES SHOULD be included in some RTP packets at the start of the session and whenever the SSRC changes. It might also be useful to include the header extension in RTP packets that comprise access points in the media (e.g., with video I-frames). The exact number of RTP packets in which this header extension is sent is intentionally not specified here, as it will depend on expected packet loss rate and loss patterns, the overhead the application can tolerate, and the importance of immediate receipt of the identification-tag.

For robustness, endpoints need to be prepared for situations where the reception of the identification-tag is delayed, and SHOULD NOT terminate sessions in such cases, as the identification-tag is likely to arrive soon.

15.1. RTCP MID SDES Item

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|          MID=TBD          |      length      | identification-tag  |...
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

The identification-tag payload is UTF-8 encoded [RFC3629], as in SDP.

The identification-tag is not zero terminated.

[RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.]

15.2. RTP SDES Header Extension For MID

The payload, containing the identification-tag, of the RTP SDES header extension element can be encoded using either the one-byte or two-byte header [RFC7941]. The identification-tag payload is UTF-8 encoded, as in SDP.

The identification-tag is not zero terminated. Note, that the set of header extensions included in the packet needs to be padded to the next 32-bit boundary using zero bytes [RFC8285].

As the identification-tag is included in either an RTCP SDES item or an RTP SDES header extension, or both, there needs to be some consideration about the packet expansion caused by the identification-tag. To avoid Maximum Transmission Unit (MTU) issues for the RTP packets, the header extension's size needs to be taken into account when encoding the media.

It is recommended that the identification-tag is kept short. Due to the properties of the RTP header extension mechanism, when using the one-byte header, a tag that is 1-3 bytes will result in a minimal number of 32-bit words used for the RTP SDES header extension, in case no other header extensions are included at the same time. Note, do take into account that some single characters when UTF-8 encoded will result in multiple octets. The identification-tag MUST NOT contain any user information, and applications SHALL avoid generating the identification-tag using a pattern that enables user- or application identification.

16. IANA Considerations

16.1. New SDES item

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

[RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.]

This document adds the MID SDES item to the IANA "RTP SDES item types" registry as follows:

Value:	TBD
Abbrev.:	MID
Name:	Media Identification
Reference:	RFCXXXX

16.2. New RTP SDES Header Extension URI

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document defines a new extension URI in the RTP SDES Compact Header Extensions sub-registry of the RTP Compact Header Extensions registry sub-registry, according to the following data:

Extension URI: urn:ietf:params:rtp-hdext:sdes:mid
Description: Media identification
Contact: IESG (iesg@ietf.org)
Reference: RFCXXXX

The SDES item does not reveal privacy information about the users. It is simply used to associate RTP-based media with the correct SDP media description ("m=" section) in the SDP used to negotiate the media.

The purpose of the extension is for the offerer to be able to associate received multiplexed RTP-based media before the offerer receives the associated SDP answer.

16.3. New SDP Attribute

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document defines a new SDP media-level attribute, 'bundle-only', according to the following data:

Attribute name: bundle-only
Type of attribute: media
Subject to charset: No
Purpose: Request a media description to be accepted in the answer only if kept within a BUNDLE group by the answerer.
Appropriate values: N/A
Contact name: IESG
Contact e-mail: iesg@ietf.org
Reference: RFCXXXX
Mux category: NORMAL

16.4. New SDP Group Semantics

[RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.]

This document registers the following semantics with IANA in the "Semantics for the "group" SDP Attribute" subregistry (under the "Session Description Protocol (SDP) Parameters" registry:

Semantics	Token	Reference
Media bundling	BUNDLE	[RFCXXXX]

Mux category: NORMAL

17. Security Considerations

The security considerations defined in [RFC3264] and [RFC5888] apply to the BUNDLE extension. Bundle does not change which information, e.g., RTP streams, flows over the network, with the exception of the usage of the MID SDES item as discussed below. Primarily it changes which addresses and ports, and thus in which (RTP) sessions the information is flowing. This affects the security contexts being used and can cause previously separated information flows to share the same security context. This has very little impact on the performance of the security mechanism of the RTP sessions. In cases where one would have applied different security policies on the different RTP streams being bundled, or where the parties having access to the security contexts would have differed between the RTP streams, additional analysis of the implications are needed before selecting to apply BUNDLE.

The identification-tag, independent of transport, RTCP SDES packet or RTP header extension, can expose the value to parties beyond the signaling chain. Therefore, the identification-tag values MUST be generated in a fashion that does not leak user information, e.g., randomly or using a per-bundle group counter, and SHOULD be 3 bytes or less, to allow them to efficiently fit into the MID RTP header extension. Note that if implementations use different methods for generating identification-tags this could enable fingerprinting of the implementation making it vulnerable to targeted attacks. The identification-tag is exposed on the RTP stream level when included in the RTP header extensions, however what it reveals of the RTP media stream structure of the endpoint and application was already possible to deduce from the RTP streams without the MID SDES header

extensions. As the identification-tag is also used to route the media stream to the right application functionality it is important that the value received is the one intended by the sender, thus integrity and the authenticity of the source are important to prevent denial of service on the application. Existing SRTP configurations and other security mechanisms protecting the whole RTP/RTCP packets will provide the necessary protection.

When the BUNDLE extension is used, the set of configurations of the security mechanism used in all the bundled media descriptions will need to be compatible so that they can be used simultaneously, at least per direction or endpoint. When using SRTP this will be the case, at least for the IETF defined key-management solutions due to their SDP attributes (a=crypto, a=fingerprint, a=mikey) and their classification in [I-D.ietf-mmusic-sdp-mux-attributes].

The security considerations of "RTP Header Extension for the RTP Control Protocol (RTCP) Source Description Items" [RFC7941] requires that when RTCP is confidentiality protected, then any SDES RTP header extension carrying an SDES item, such as the MID RTP header extension, is also protected using commensurate strength algorithms. However, assuming the above requirements and recommendations are followed, there are no known significant security risks with leaving the MID RTP header extension without confidentiality protection. Therefore, this specification updates RFC 7941 by adding the exception that this requirement MAY be ignored for the MID RTP header extension. Security mechanisms for RTP/RTCP are discussed in Options for Securing RTP Sessions [RFC7201], for example SRTP [RFC3711] can provide the necessary security functions of ensuring the integrity and source authenticity.

18. Examples

18.1. Example: Tagged m= Section Selections

The example below shows:

- o An initial BUNDLE offer, in which the offerer wants to negotiate a BUNDLE group, and indicates the audio m= section as the suggested offerer tagged "m=" section.
- o An initial BUNDLE answer, in which the answerer accepts the creation of the BUNDLE group, selects the audio m= section in the offer as the offerer tagged "m=" section, selects the audio "m=" section in the answer as the answerer tagged "m=" section and assigns the answerer BUNDLE address:port to that "m=" section.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP6 2001:db8::3
s=
c=IN IP6 2001:db8::3
t=0 0
a=group:BUNDLE foo bar

m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP6 2001:db8::1
s=
c=IN IP6 2001:db8::1
t=0 0
a=group:BUNDLE foo bar

m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 32
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:32 MPV/90000
```

```
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

18.2. Example: BUNDLE Group Rejected

The example below shows:

- o An initial BUNDLE offer, in which the offerer wants to negotiate a BUNDLE group, and indicates the audio m= section as the suggested offerer tagged "m=" section.
- o An initial BUNDLE answer, in which the answerer rejects the creation of the BUNDLE group, generates a normal answer and assigns a unique address:port to each "m=" section in the answer.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP6 2001:db8::3
s=
c=IN IP6 2001:db8::3
t=0 0
a=group:BUNDLE foo bar

m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 10002 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=rtcp-mux
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP6 2001:db8::1
s=
c=IN IP6 2001:db8::1
t=0 0

m=audio 20000 RTP/AVP 0
b=AS:200
a=rtcp-mux
a=rtpmap:0 PCMU/8000

m=video 30000 RTP/AVP 32
b=AS:1000
a=rtcp-mux
a=rtpmap:32 MPV/90000
```

18.3. Example: Offerer Adds a Media Description to a BUNDLE Group

The example below shows:

- o A subsequent offer, in which the offerer adds a new bundled "m=" section (video), indicated by the "zen" identification-tag, to a previously negotiated BUNDLE group, indicates the new "m=" section as the offerer tagged "m=" section and assigns the offerer BUNDLE address:port to that "m=" section.
- o A subsequent answer, in which the answerer indicates the new video "m=" section in the answer as the answerer tagged "m=" section and assigns the answerer BUNDLE address:port to that "m=" section.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP6 2001:db8::3
s=
c=IN IP6 2001:db8::3
t=0 0
a=group:BUNDLE zen foo bar

m=audio 0 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=bundle-only
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid

m=video 0 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid

m=video 10000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP6 2001:db8::1
s=
c=IN IP6 2001:db8::1
t=0 0
a=group:BUNDLE zen foo bar

m=audio 0 RTP/AVP 0
b=AS:200
a=mid:foo
a=bundle-only
a=rtpmap:0 PCMU/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 32
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 20000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

18.4. Example: Offerer Moves a Media Description Out of a BUNDLE Group

The example below shows:

- o A subsequent offer, in which the offerer removes a "m=" section (video), indicated by the "zen" identification-tag, from a previously negotiated BUNDLE group, indicates one of the bundled "m=" sections (audio) remaining in the BUNDLE group as the offerer tagged "m=" section and assigns the offerer BUNDLE address:port to that "m=" section.
- o A subsequent answer, in which the answerer removes the "m=" section from the BUNDLE group, indicates the audio "m=" section in the answer as the answerer tagged "m=" section and assigns the answerer BUNDLE address:port to that "m=" section.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP6 2001:db8::3
s=
c=IN IP6 2001:db8::3
t=0 0
a=group:BUNDLE foo bar

m=audio 10000 RTP/AVP 0 8 97
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 31 32
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 50000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP6 2001:db8::1
s=
c=IN IP6 2001:db8::1
t=0 0
a=group:BUNDLE foo bar

m=audio 20000 RTP/AVP 0
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

```
m=video 0 RTP/AVP 32
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid
```

```
m=video 60000 RTP/AVP 66
b=AS:1000
a=mid:zen
a=rtcp-mux
a=rtpmap:66 H261/90000
```

18.5. Example: Offerer Disables a Media Description Within a BUNDLE Group

The example below shows:

- o A subsequent offer, in which the offerer disables (by assigning a zero port value) a "m=" section (video), indicated by the "zen" identification-tag, from a previously negotiated BUNDLE group, indicates one of the bundled "m=" sections (audio) remaining active in the BUNDLE group as the offerer tagged "m=" section and assigns the offerer BUNDLE address:port to that "m=" section.
- o A subsequent answer, in which the answerer disables the "m=" section, indicates the audio "m=" section in the answer as the answerer tagged "m=" section and assigns the answerer BUNDLE address:port to that "m=" section.

SDP Offer (1)

```
v=0
o=alice 2890844526 2890844526 IN IP6 2001:db8::3
s=
t=0 0
a=group:BUNDLE foo bar

m=audio 10000 RTP/AVP 0 8 97
c=IN IP6 2001:db8::3
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
```

```
a=rtpmap:97 iLBC/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 31 32
c=IN IP6 2001:db8::3
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 66
a=mid:zen
a=rtpmap:66 H261/90000
```

SDP Answer (2)

```
v=0
o=bob 2808844564 2808844564 IN IP6 2001:db8::1
s=
t=0 0
a=group:BUNDLE foo bar

m=audio 20000 RTP/AVP 0
c=IN IP6 2001:db8::1
b=AS:200
a=mid:foo
a=rtcp-mux
a=rtpmap:0 PCMU/8000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 32
c=IN IP6 2001:db8::1
b=AS:1000
a=mid:bar
a=bundle-only
a=rtpmap:32 MPV/90000
a=extmap:1 urn:ietf:params:rtp-hdrext:sdes:mid

m=video 0 RTP/AVP 66
a=mid:zen
a=rtpmap:66 H261/90000
```

19. Acknowledgements

The usage of the SDP grouping extension for negotiating bundled media is based on similar alternatives proposed by Harald Alvestrand and Cullen Jennings. The BUNDLE extension described in this document is based on the different alternative proposals, and text (e.g., SDP examples) have been borrowed (and, in some cases, modified) from those alternative proposals.

The SDP examples are also modified versions from the ones in the Alvestrand proposal.

Thanks to Paul Kyzivat, Martin Thomson, Flemming Andreassen, Thomas Stach, Ari Keranen, Adam Roach, Christian Groves, Roman Shpount, Suhas Nandakumar, Nils Ohlmeier, Jens Guballa, Raju Makaraju, Justin Uberti, Taylor Brandstetter, Byron Campen and Eric Rescorla for reading the text, and providing useful feedback.

Thanks to Bernard Aboba, Peter Thatcher, Justin Uberti, and Magnus Westerlund for providing the text for the section on RTP/RTCP stream association.

Thanks to Magnus Westerlund, Colin Perkins and Jonathan Lennox for providing help and text on the RTP/RTCP procedures.

Thanks to Charlie Kaufman for performing the Sec-Dir review.

Thanks to Linda Dunbar for performing the Gen-ART review.

Thanks to Spotify for providing music for the countless hours of document editing.

20. Change Log

[RFC EDITOR NOTE: Please remove this section when publishing]

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-51

- o Changes based on IESG reviews.
- o - Clarification of 'initial offer' terminology.
- o - Merging of tagged m- section selection sections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-50

- o Changes based on IESG reviews.

- o - Adding of tagged m- section concept.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-49

- o Changes based on IESG reviews.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-48

- o Changes based on Sec-Dir review by Charlie Kaufman.

- o - s/unique address/unique address:port

- o Changes based on Gen-ART review by Linda Dunbar.

- o Mux category for group:BUNDLE attribute added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-47

- o Changes based on AD review by Ben Campbell.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-46

- o Pre-RFC5378 disclaimer removed put back.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-45

- o Mux category for SDP 'group:BUNDLE' attribute added.

- o <https://github.com/cdh4u/draft-sdp-bundle/pull/54>

- o Pre-RFC5378 disclaimer removed.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-44

- o Minor editorial nits based on pull request by Colin P.

- o <https://github.com/cdh4u/draft-sdp-bundle/pull/53>

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-43

- o Changes based on WG chairs review.

- o Text added in order to close GitHub issues by Taylor B.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-42

- o Changes based on final WG review.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-41

- o Update to section 6 o RFC 3264:
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/47>
- o Editorial clarification on BUNDLE address selection:
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/46>

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-40

- o Editorial changes and technical restrictions in order to make the specification more understandable:
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/45>
- o - BUNDLE address is only assigned to m- section indicated by BUNDLE-tag.
- o - bundle-only attribute also used in answers and subsequent offers.
- o - Answerer cannot reject, or remove, the bundled m- section that contains the BUNDLE address.
- o - ICE Offer/Answer sections removed, due to duplicated information.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-39

- o Editorial terminology changes.
- o RFC 5285 reference replaced by reference to RFC 8285.
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/44>
- o - Clarify that an m- section can not be moved between BUNDLE groups without first moving the m- section out of a BUNDLE group.
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/41>
- o - Addition of BUNDLE transport concept.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-38

- o Changes to RTP streaming mapping section based on text from Colin Perkins.

- o The following GitHub pull requests were merged:
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/34>
- o - Proposed updates to RTP processing
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/35>
- o - fixed reference to receiver-id section

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-37

- o The following GitHub pull request was merged:
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/33>

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-36

- o The following GitHub pull requests were merged:
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/32>
- o - extmap handling in BUNDLE.
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/31>
- o - Additional Acknowledgement text added.
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/30>
- o - MID SDES item security procedures updated
- o <https://github.com/cdh4u/draft-sdp-bundle/pull/29>
- o - Appendix B of JSEP moved into BUNDLE.
- o - Associating RTP/RTCP packets with SDP m- lines.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-35

- o Editorial changes on RTP streaming mapping section based on comments from Colin Perkins.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-34

- o RTP streams, instead of RTP packets, are associated with m- lines.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-33

- o Editorial changes based on comments from Eric Rescorla and Cullen Jennings:
- o - Changes regarding usage of RTP/RTCP multiplexing attributes.
- o - Additional text regarding associating RTP/RTCP packets with SDP m- lines.
- o - Reference correction.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-32

- o Editorial changes based on comments from Eric Rescorla and Cullen Jennings:
- o - Justification for mechanism added to Introduction.
- o - Clarify that the order of m- lines in the group:BUNDLE attribute does not have to be the same as the order in which the m- lines are listed in the SDP.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-31

- o Editorial changes based on GitHub Pull requests by Martin Thomson:
- o - <https://github.com/cdh4u/draft-sdp-bundle/pull/2>
- o - <https://github.com/cdh4u/draft-sdp-bundle/pull/1>
- o Editorial change based on comment from Diederick Huijbers (9th July 2016).
- o Changes based on comments from Flemming Andreassen (21st June 2016):
- o - Mux category for SDP bundle-only attribute added.
- o - Mux category considerations editorial clarification.
- o - Editorial changes.
- o RTP SDES extension according to draft-ietf-avtext-sdes-hdr-ext.
- o Note whether Design Considerations appendix is to be kept removed:
- o - Appendix is kept within document.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-30

- o Indicating in the Abstract and Introduction that the document updates RFC 3264.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-29

- o Change based on WGLC comment from Colin Perkins.
- o - Clarify that SSRC can be reused by another source after a delay of 5 RTCP reporting intervals.
- o Change based on WGLC comment from Alissa Cooper.
- o - IANA registry name fix.
- o - Additional IANA registration information added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-28

- o - Alignment with exclusive mux procedures.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-27

- o - Yet another terminology change.
- o - Mux category considerations added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-26

- o - ICE considerations modified: ICE-related SDP attributes only added to the bundled m- line representing the selected BUNDLE address.
- o - Reference to draft-ietf-mmusic-ice-sip-sdp added.
- o - Reference to RFC 5245 replaced with reference to draft-ietf-ice-rfc5245bis.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-25

- o - RTP/RTCP mux procedures updated with exclusive RTP/RTCP mux considerations.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-24

- o - Reference and procedures associated with exclusive RTP/RTCP mux added

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-23

- o - RTCP-MUX mandatory for bundled RTP m- lines
- o - Editorial fixes based on comments from Flemming Andreassen

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-22

- o - Correction of Ari's family name
- o - Editorial fixes based on comments from Thomas Stach
- o - RTP/RTCP correction based on comment from Magnus Westerlund
- o -- <http://www.ietf.org/mail-archive/web/mmusic/current/msg14861.html>

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-21

- o - Correct based on comment from Paul Kyzivat
- o -- 'received packets' replaced with 'received data'

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-20

- o - Clarification based on comment from James Guballa
- o - Clarification based on comment from Flemming Andreassen

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-19

- o - DTLS Considerations section added.
- o - BUNDLE semantics added to the IANA Considerations
- o - Changes based on WGLC comments from Adam Roach
- o -- <http://www.ietf.org/mail-archive/web/mmusic/current/msg14673.html>

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-18

- o - Changes based on agreements at IETF#92
- o -- BAS Offer removed, based on agreement at IETF#92.
- o -- Procedures regarding usage of SDP "b=" line is replaced with a reference to to draft-ietf-mmusic-sdp-mux-attributes.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-17

- o - Editorial changes based on comments from Magnus Westerlund.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-16

- o - Modification of RTP/RTCP multiplexing section, based on comments from Magnus Westerlund.
- o - Reference updates.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-15

- o - Editorial fix.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-14

- o - Editorial changes.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-13

- o Changes to allow a newly suggested offerer BUNDLE address to be assigned to each bundled m- line.
- o Changes based on WGLC comments from Paul Kyzivat
- o - Editorial fixes

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-12

- o Usage of SDP 'extmap' attribute added
- o SDP 'bundle-only' attribute scoped with "m=" lines with a zero port value
- o Changes based on WGLC comments from Thomas Stach
- o - ICE candidates not assigned to bundle-only m- lines with a zero port value
- o - Editorial changes
- o Changes based on WGLC comments from Colin Perkins
- o - Editorial changes:
 - o -- "RTP SDES item" -> "RTCP SDES item"
 - o -- "RTP MID SDES item" -> "RTCP MID SDES item"

- o - Changes in section 10.1.1:
- o -- "SHOULD NOT" -> "MUST NOT"
- o -- Additional text added to the Note
- o - Change to section 13.2:
- o -- Clarify that mid value is not zero terminated
- o - Change to section 13.3:
- o -- Clarify that mid value is not zero terminated
- o -- Clarify padding
- o Changes based on WGLC comments from Paul Kyzivat
- o - Editorial changes:
- o Changes based on WGLC comments from Jonathan Lennox
- o - Editorial changes:
- o - Definition of SDP bundle-only attribute aligned with structure in 4566bis draft

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-11

- o Editorial corrections based on comments from Harald Alvestrand.
- o Editorial corrections based on comments from Cullen Jennings.
- o Reference update (RFC 7160).
- o Clarification about RTCP packet sending when RTP/RTCP multiplexing is not used (<http://www.ietf.org/mail-archive/web/mmusic/current/msg13765.html>).
- o Additional text added to the Security Considerations.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-10

- o SDP bundle-only attribute added to IANA Considerations.
- o SDES item and RTP header extension added to Abstract and Introduction.

- o Modification to text updating section 8.2 of RFC 3264.
- o Reference corrections.
- o Editorial corrections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-09

- o Terminology change: "bundle-only attribute assigned to m= line" to "bundle-only attribute associated with m= line".
- o Editorial corrections.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-08

- o Editorial corrections.
- o - "of"->"if" (8.3.2.5).
- o - "optional"->"OPTIONAL" (9.1).
- o - Syntax/ABNF for 'bundle-only' attribute added.
- o - SDP Offer/Answer sections merged.
- o - 'Request new offerer BUNDLE address' section added

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-07

- o OPEN ISSUE regarding Receiver-ID closed.
- o - RTP MID SDES Item.
- o - RTP MID Header Extension.
- o OPEN ISSUE regarding insertion of SDP 'rtcp' attribute in answers closed.
- o - Indicating that, when rtcp-mux is used, the answerer MUST NOT include an 'rtcp' attribute in the answer, based on the procedures in section 5.1.3 of RFC 5761.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-06

- o Draft title changed.
- o Added "SDP" to section names containing "Offer" or "Answer".

- o Editorial fixes based on comments from Paul Kyzivat (<http://www.ietf.org/mail-archive/web/mmusic/current/msg13314.html>).
- o Editorial fixed based on comments from Colin Perkins (<http://www.ietf.org/mail-archive/web/mmusic/current/msg13318.html>).
- o - Removed text about extending BUNDLE to allow multiple RTP sessions within a BUNDLE group.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-05

- o Major re-structure of SDP Offer/Answer sections, to align with RFC 3264 structure.
- o Additional definitions added.
- o - Shared address.
- o - Bundled "m=" line.
- o - Bundle-only "m=" line.
- o - Offerer suggested BUNDLE mid.
- o - Answerer selected BUNDLE mid.
- o Q6 Closed (IETF#88): An Offerer MUST NOT assign a shared address to multiple "m=" lines until it has received an SDP Answer indicating support of the BUNDLE extension.
- o Q8 Closed (IETF#88): An Offerer can, before it knows whether the Answerer supports the BUNDLE extension, assign a zero port value to a 'bundle-only' "m=" line.
- o SDP 'bundle-only' attribute section added.
- o Connection data nettype/addrtype restrictions added.
- o RFC 3264 update section added.
- o Indicating that a specific payload type value can be used in multiple "m=" lines, if the value represents the same codec configuration in each "m=" line.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-04

- o Updated Offerer procedures (<http://www.ietf.org/mail-archive/web/mmusic/current/msg12293.html>).
- o Updated Answerer procedures (<http://www.ietf.org/mail-archive/web/mmusic/current/msg12333.html>).
- o Usage of SDP 'bundle-only' attribute added.
- o Reference to Trickle ICE document added.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-02

- o Mechanism modified, to be based on usage of SDP Offers with both different and identical port number values, depending on whether it is known if the remote endpoint supports the extension.
- o Cullen Jennings added as co-author.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-01

- o No changes. New version due to expiration.

Changes from draft-ietf-mmusic-sdp-bundle-negotiation-00

- o No changes. New version due to expiration.

Changes from draft-holmberg-mmusic-sdp-multiplex-negotiation-00

- o Draft name changed.
- o Harald Alvestrand added as co-author.
- o "Multiplex" terminology changed to "bundle".
- o Added text about single versus multiple RTP Sessions.
- o Added reference to RFC 3550.

21. References

21.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<https://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <<https://www.rfc-editor.org/info/rfc3605>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<https://www.rfc-editor.org/info/rfc3711>>.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <<https://www.rfc-editor.org/info/rfc4566>>.
- [RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, DOI 10.17487/RFC4961, July 2007, <<https://www.rfc-editor.org/info/rfc4961>>.
- [RFC5761] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", RFC 5761, DOI 10.17487/RFC5761, April 2010, <<https://www.rfc-editor.org/info/rfc5761>>.
- [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, DOI 10.17487/RFC5764, May 2010, <<https://www.rfc-editor.org/info/rfc5764>>.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <<https://www.rfc-editor.org/info/rfc5888>>.

- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC7941] Westerlund, M., Burman, B., Even, R., and M. Zanaty, "RTP Header Extension for the RTP Control Protocol (RTCP) Source Description Items", RFC 7941, DOI 10.17487/RFC7941, August 2016, <<https://www.rfc-editor.org/info/rfc7941>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <<https://www.rfc-editor.org/info/rfc8285>>.
- [I-D.ietf-ice-rfc5245bis]
Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-20 (work in progress), March 2018.
- [I-D.ietf-mmusic-sdp-mux-attributes]
Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-16 (work in progress), December 2016.
- [I-D.ietf-mmusic-mux-exclusive]
Holmberg, C., "Indicating Exclusive Support of RTP/RTCP Multiplexing using SDP", draft-ietf-mmusic-mux-exclusive-12 (work in progress), May 2017.
- [I-D.ietf-mmusic-ice-sip-sdp]
Petit-Huguenin, M., Nandakumar, S., and A. Keranen, "Session Description Protocol (SDP) Offer/Answer procedures for Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-sip-sdp-20 (work in progress), April 2018.
- [I-D.ietf-mmusic-trickle-ice-sip]
Ivov, E., Stach, T., Marocco, E., and C. Holmberg, "A Session Initiation Protocol (SIP) Usage for Trickle ICE", draft-ietf-mmusic-trickle-ice-sip-14 (work in progress), February 2018.

21.2. Informative References

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <<https://www.rfc-editor.org/info/rfc3611>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.
- [RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <<https://www.rfc-editor.org/info/rfc5576>>.
- [RFC7160] Petit-Huguenin, M. and G. Zorn, Ed., "Support for Multiple Clock Rates in an RTP Session", RFC 7160, DOI 10.17487/RFC7160, April 2014, <<https://www.rfc-editor.org/info/rfc7160>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<https://www.rfc-editor.org/info/rfc7201>>.
- [RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <<https://www.rfc-editor.org/info/rfc7656>>.
- [RFC7657] Black, D., Ed. and P. Jones, "Differentiated Services (Diffserv) and Real-Time Communication", RFC 7657, DOI 10.17487/RFC7657, November 2015, <<https://www.rfc-editor.org/info/rfc7657>>.

[I-D.ietf-ice-trickle]

Ivov, E., Rescorla, E., Uberti, J., and P. Saint-Andre,
"Trickle ICE: Incremental Provisioning of Candidates for
the Interactive Connectivity Establishment (ICE)
Protocol", draft-ietf-ice-trickle-21 (work in progress),
April 2018.

[I-D.ietf-avtext-lrr]

Lennox, J., Hong, D., Uberti, J., Holmer, S., and M.
Flodman, "The Layer Refresh Request (LRR) RTCP Feedback
Message", draft-ietf-avtext-lrr-07 (work in progress),
July 2017.

Appendix A. Design Considerations

One of the main issues regarding the BUNDLE grouping extensions has been whether, in SDP Offers and SDP Answers, the same port value can be inserted in "m=" lines associated with a BUNDLE group, as the purpose of the extension is to negotiate the usage of a single transport for media specified by the "m=" sections. Issues with both approaches, discussed in the Appendix have been raised. The outcome was to specify a mechanism which uses SDP Offers with both different and identical port values.

Below are the primary issues that have been considered when defining the "BUNDLE" grouping extension:

- o 1) Interoperability with existing UAs.
- o 2) Interoperability with intermediary Back to Back User Agent (B2BUA) and proxy entities.
- o 3) Time to gather, and the number of, ICE candidates.
- o 4) Different error scenarios, and when they occur.
- o 5) SDP Offer/Answer impacts, including usage of port number value zero.

A.1. UA Interoperability

Consider the following SDP Offer/Answer exchange, where Alice sends an SDP Offer to Bob:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0

m=audio 10000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 10002 RTP/AVP 97
a=rtpmap:97 H261/90000
```

SDP Answer

```
v=0
o=bob 2808844564 2808844564 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0

m=audio 20000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 20002 RTP/AVP 97
a=rtpmap:97 H261/90000
```

RFC 4961 specifies a way of doing symmetric RTP but that is a later extension to RTP and Bob can not assume that Alice supports RFC 4961. This means that Alice may be sending RTP from a different port than 10000 or 10002 – some implementations simply send the RTP from an ephemeral port. When Bob's endpoint receives an RTP packet, the only way that Bob knows if the packet is to be passed to the video or audio codec is by looking at the port it was received on. This led some SDP implementations to use the fact that each "m=" section had a different port number to use that port number as an index to find the correct m line in the SDP. As a result, some implementations that do support symmetric RTP and ICE still use an SDP data structure where SDP with "m=" sections with the same port such as:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0

m=audio 10000 RTP/AVP 97
a=rtpmap:97 iLBC/8000
m=video 10000 RTP/AVP 98
a=rtpmap:98 H261/90000
```

will result in the second "m=" section being considered an SDP error because it has the same port as the first line.

A.2. Usage of Port Number Value Zero

In an SDP Offer or SDP Answer, the media specified by an "m=" section can be disabled/rejected by setting the port number value to zero. This is different from e.g., using the SDP direction attributes, where RTCP traffic will continue even if the SDP "inactive" attribute is indicated for the associated "m=" section.

If each "m=" section associated with a BUNDLE group would contain different port values, and one of those port values would be used for a BUNDLE address:port associated with the BUNDLE group, problems would occur if an endpoint wants to disable/reject the "m=" section associated with that port, by setting the port value to zero. After that, no "m=" section would contain the port value which is used for the BUNDLE address:port. In addition, it is unclear what would happen to the ICE candidates associated with the "m=" section, as they are also used for the BUNDLE address:port.

A.3. B2BUA And Proxy Interoperability

Some back to back user agents may be configured in a mode where if the incoming call leg contains an SDP attribute the B2BUA does not understand, the B2BUA still generates that SDP attribute in the Offer for the outgoing call leg. Consider a B2BUA that did not understand the SDP "rtcp" attribute, defined in RFC 3605, yet acted this way. Further assume that the B2BUA was configured to tear down any call where it did not see any RTCP for 5 minutes. In this case, if the B2BUA received an Offer like:

SDP Offer

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0

m=audio 49170 RTP/AVP 0
a=rtcp:53020
```

It would be looking for RTCP on port 49171 but would not see any because the RTCP would be on port 53020 and after five minutes, it would tear down the call. Similarly, a B2BUA that did not understand BUNDLE yet put BUNDLE in its offer may be looking for media on the wrong port and tear down the call. It is worth noting that a B2BUA that generated an Offer with capabilities it does not understand is not compliant with the specifications.

A.3.1. Traffic Policing

Sometimes intermediaries do not act as B2BUAs, in the sense that they don't modify SDP bodies, nor do they terminate SIP dialogs. Still, however, they may use SDP information (e.g., IP address and port) in order to control traffic gating functions, and to set traffic policing rules. There might be rules which will trigger a session to be terminated in case media is not sent or received on the ports retrieved from the SDP. This typically occurs once the session is already established and ongoing.

A.3.2. Bandwidth Allocation

Sometimes intermediaries do not act as B2BUAs, in the sense that they don't modify SDP bodies, nor do they terminate SIP dialogs. Still, however, they may use SDP information (e.g., codecs and media types) in order to control bandwidth allocation functions. The bandwidth allocation is done per "m=" section, which means that it might not be enough if media specified by all "m=" sections try to use that bandwidth. That may either simply lead to bad user experience, or to termination of the call.

A.4. Candidate Gathering

When using ICE, a candidate needs to be gathered for each port. This takes approximately 20 ms extra for each extra "m=" section due to the NAT pacing requirements. All of this gathering can be overlapped

with other things while e.g., a web-page is loading to minimize the impact. If the client only wants to generate TURN or STUN ICE candidates for one of the "m=" lines and then use trickle ICE [I-D.ietf-ice-trickle] to get the non host ICE candidates for the rest of the "m=" sections, it MAY do that and will not need any additional gathering time.

Some people have suggested a TURN extension to get a bunch of TURN allocations at once. This would only provide a single STUN result so in cases where the other end did not support BUNDLE, it may cause more use of the TURN server but would be quick in the cases where both sides supported BUNDLE and would fall back to a successful call in the other cases.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Harald Tveit Alvestrand
Google
Kungsbron 2
Stockholm 11122
Sweden

Email: harald@alvestrand.no

Cullen Jennings
Cisco
400 3rd Avenue SW, Suite 350
Calgary, AB T2P 4H2
Canada

Email: fluffy@iii.ca

Network WG
Internet-Draft
Expires: January 16, 2013
Intended Status: Standards Track (PS)

James Polk
Subha Dhesikan
Paul Jones
Cisco Systems
July 16, 2012

The Session Description Protocol (SDP) 'trafficclass' Attribute
draft-ietf-mmusic-traffic-class-for-sdp-02

Abstract

This document proposes a new Session Description Protocol (SDP) attribute to identify the traffic class a session is requesting in its offer/answer exchange.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 16, 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

1. Introduction	2
2. Traffic Class Framework and String Definitions	5
3. Traffic Class Attribute Definition	11
4. Offer/Answer Behavior	14
4.1 Offer Behavior	14
4.2 Answer Behavior	15
5. Security considerations	16
6. IANA considerations	16
7. Acknowledgments	18
8. References	18
8.1. Normative References	18
8.2. Informative References	19
Authors' Addresses	19
Appendix	20

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1. Introduction

The Session Description Protocol (SDP) [RFC4566] provides a means for an offerer to describe the specifics of a session to an answerer, and for the answerer to respond back with its session specifics to the offerer. These session specifics include offering the codec or codecs to choose from, the specific IP address and port number the offerer wants to receive the RTP stream(s) on/at, the particulars about the codecs the offerer wants considered or mandated, and so on.

There are many facets within SDP to determine the Real-time Transport Protocol (RTP) [RFC3550] details for the session establishment between one or more endpoints, but identifying how the underlying network should process each stream still remains under-specified.

The ability to identify a traffic flow by port number gives an indication to underlying network elements to treat traffic with dissimilar ports in a different way, the same or in groups the same - but different from other ports or groups of ports.

Within the context of realtime communications, the labeling of an RTP session based on media descriptor lines as just a voice and/or video session is insufficient, and provides no guidelines to the underlying network on how to treat the traffic. A more granular labeling helps on several fronts to

- inform application layer elements in the signaling path the intent of this session.
- inform the network on how to treat the traffic if the network is configured to differentiate session treatments based on the type of session the RTP is, including the ability to provide call admission control based on the type of traffic in the network.
- allow network monitoring/management of traffic types realtime and after-the-fact analysis.

Some network operators want the ability to guarantee certain traffic gets a minimum amount of network bandwidth per link or through a series of links that make up a network such as a campus or WAN, or a backbone. For example, a call center voice application might get at least 20% of the available link bandwidth.

Some network operators want the ability to allow certain users or devices access to greater bandwidth during non-busy hours, than during busy hours of the day. For example, all desktop video might operate at 1080p during non-peak times, but a similar session might be curtailed between the same users or devices to 720p or 360p during peak hours. Another example would be to reduce the frames per second (fps) rate, say from 30fps to 15fps. This case is not as clear as accepting or denying similar sessions during different times of the day, but tuning the access to the bandwidth based on the type of session. In other words, tune down the bandwidth for desktop video during peak hours to allow a 3-screen Telepresence session that would otherwise look like the same type of traffic (RTP, and more granular, video).

RFC 4594 established a guideline for classifying the various flows in the network and the Differentiated Services Codepoints (DSCP) that apply to many traffic types (table 3 of [RFC4594]), including RTP based voice and video traffic sessions. The RFC also defines the per hop network behavior that is strongly encouraged for each of these application traffic types based on the traffic characteristics and tolerances to delay, loss and jitter within each traffic class.

Video was broken down into 4 categories in that RFC, and voice into another single category. We do not believe this satisfies the technical and business requirements to accomplish sufficiently unique labeling of RTP traffic.

If the application becomes aware of traffic labeling,

- this can be coded into layer 3 mechanisms.
- this can be coded into layer 4 protocols and/or mechanisms.
- this can be coded into a combination of mechanisms and protocols.

The layer 3 mechanism for differentiating traffic is either the port number or the Differentiated Services Codepoint (DSCP) value [RFC2474]. Within the public Internet, if the application is not part of a managed service, the DSCP likely will be best effort (BE), or reset to BE when ingressing a provider's network. Within the corporate LAN, this is usually completely configurable and a local IT department can take full advantage of this labeling to shape and manage their network as they see fit.

Within a network core, DiffServ typically does not apply. That said, DiffServ can be used to identify which traffic goes into which MPLS tunnel [RFC4124].

Labeling realtime traffic types using a layer 4 protocol would likely involve RSVP [RFC2205] or NSIS [RFC4080]. RSVP has an Application Identifier (app-ID) defined in [RFC2872] that provides a means for carrying a traffic class label along the media path. An advantage of this mechanism is that the label can inform each domain along the media path what type of traffic this traffic flow is, and allow each domain to adjust the appropriate DSCP value (set by each domain for use within that domain). Meaning, if a DSCP value is set by an endpoint or a router in the first domain and gets reset by a service provider, the far-end domain will be able to reset the DSCP value to the intended traffic class. There is a proposed extension to RSVP which creates individual profiles for what goes into each app-ID field to describe these traffic classes [ID-RSVP-PROF], which will take advantage of what is described in this document.

There are several proprietary mechanisms that can take advantage of this labeling, but none of those will be discussed here.

The idea of traffic - or service - identification is not new; it has been described in [RFC5897]. If that RFC is used as a guideline, identification that leads to stream differentiation can be quite useful. One of the points within RFC 5897 is that users cannot be allowed to assign any identification (fraud is one reason given). In addition, RFC 5897 recommends that service identification should be done in signaling, rather than guessing or deep packet inspection. Any network currently would have to currently guess or perform deep packet inspection to classify traffic and offer the service as per RFC 4594 since such service identification information is currently not available in SDP and therefore to the network elements. Since SDP understands how each stream is created (i.e., the particulars of the RTP stream), this is the right place to have this service differentiated. Such service differentiation can then be communicated to and leveraged by the network.

[Editor's Note: the words "traffic" and "service" are similar enough that the above paragraph talks about RFC 5897's "service identification", but this document only discuss and propose traffic indications in SDP.]

This document proposes a simple attribute line to identify the application a session is requesting in its offer/answer exchange. This document uses previously defined service class strings for consistency between IETF documents.

This document modifies the traffic classes originally created in RFC 4594 in Section 2, incrementing each class with application identifiers and optional adjective strings. Section 3 defines the new SDP attribute "trafficclass". Section 4 discusses the offerer and answerer behavior when generating or receiving this attribute.

2. Traffic Class Framework and String Definitions

The framework of the traffic class attribute will have at least two parts, allowing for several more to be included. The intention is to have a category class (e.g., Conversational) that merely serves as the anchor point for an application component that when paired together, form the highest level traffic class. An adjective component provides further granularity for the application. There can be more than one adjective within a traffic class label to further refine the uniqueness of a traffic class being described.

The traffic class label will have the following structure,

category.application(.adjective)(.adjective)

[Editor's Note: the above is not exactly the ABNF to be used. The order is right. The category and application MUST appear first (each only once) and zero or more adjectives can appear following the application component.]

Where

- 1) the 1st component is the human understandable category;
- 2) the 2nd component is the application;
- 3) an optional 3rd component or series of components are adjective(s) used to further refine the application component;

The construction of the traffic class label for Telepresence video would follow the minimum form of:

Conversational.video.immersive

where there might be one or more adjective after '.immersive'.

There is no traffic class or DSCP value associated with just "Conversational". There is a traffic class associated with "Conversational.video", creating a differentiation between it and a "Conversational.video.immersive" traffic class, which would have DSCP associated with the latter traffic class, depending on local

policy. Each category component is defined below, as are several of application and adjective strings.

[Editor's Note: We're not yet sure how much of what's below will be proposed for IANA registration, but the 5 category components will be, as well as at least some application components per category component. Some adjective components will also likely be proposed for IANA registration.

The 5 category components of the traffic class attribute are as follows:

- o Conversational
- o Multimedia-Conferencing
- o Realtime-Interactive
- o Multimedia-Streaming
- o Broadcast

The following application components of the traffic class attribute are as follows:

- o Audio
- o Video
- o Text
- o application-sharing
- o Presentation-data
- o Whiteboarding
- o Webchat/IM
- o Gaming
- o Virtual-desktop (interactive)
- o Remote-desktop
- o Telemetry (e.g., NORAD missile control)
- o Multiplex (i.e., combined streams)
- o Webcast
- o IPTV
- o Live-events (one-way, in realtime)
- o surveillance

The following adjective components of the traffic class attribute are as follows:

- o Immersive
- o avconf
- o Realtime-Text
- o web

Each of the above 3 lists will be defined in the following subsections.

2.1 Conversational Category Traffic Class

The Conversational traffic class is best suited for applications that require very low delay variation and generally intended to enable realtime, bi-directional person-to-person or multi-directional via an MCU communication. The following application components are appropriate for use with the Conversational category:

- o Audio (voice)**
- o Video**
- o Text (i.e., real-time text required by deaf users)

**The above applications will also be used within Multimedia Streaming and Broadcast

With adjective substrings to the above

Immersive (TP) - An interactive audio-visual communications experience between remote locations, where the users enjoy a strong sense of realism and presence between all participants by optimizing a variety of attributes such as audio and video quality, eye contact, body language, spatial audio, coordinated environments and natural image size.

Avconf - An interactive audio-visual communication experience that is not immersive in nature, though can have a high resolution video component.

Realtime-Text (RTT) - a term for real-time transmission of text in a character-by-character fashion for use in conversational services, often as a text equivalent to voice-based conversational services. Conversational text is defined in the ITU-T Framework for multimedia services, Recommendation F.700 [RFC5194].

Web - for realtime aspects of web conferencing; mutually exclusive of both Immersive and Desktop video experiences

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Conversational	High priority, typically small packets (large video frames produce large packets), generally sustained high packet rate, low inter-packet transmission interval,	Very Low	Very Low	Very Low

	usually UDP framed in (S)RTP			
+-----+	+-----+	+-----+	+-----+	+-----+

Figure 1. Conversational Traffic Characteristics

2.2 Multimedia-Conferencing Category Traffic Class

Multimedia-Conferencing traffic class is best suited for applications that are generally intended for communication between human users, but are less demanding in terms of delay, packet loss, and jitter than what Conversational applications require. These applications require low to medium delay and may have the ability to change encoding rate (rate adaptive) or transmit data at varying rates. The following application components are appropriate for use with the Multimedia-Conferencing category:

- o application-sharing (that webex does or protocols like T.128) - An application that shares the output of one or more running applications or the desktop on a host. This can utilize vector graphics, raster graphics or video.
- o Presentation-data - can be a series of still images or motion video.
- o Whiteboarding - an application enabling the exchange of graphical information including images, pointers and filled and unfilled parametric drawing elements (points, lines, polygons and ellipses).
- o (RTP-based) file transfer
- o Web (conference) chat/instant messaging

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Multimedia Conferencing	Variable size packets, Variable transmit interval, rate adaptive, reacts to loss, usually TCP-based	Low	Low	Low
		-	-	-
		Medium	Medium	Medium

Figure 2. Multimedia Conferencing Traffic Characteristics

2.3 Realtime-Interactive Category Traffic Class

Realtime-Interactive traffic class is intended for interactive variable rate inelastic applications that require low jitter and loss and very low delay. The following application components are

appropriate for use with the Realtime-Interactive category:

- o Gaming - interactive player video games with other users on other hosts (e.g., Doom)
- o Virtualized desktop (interactive) - similar to an X-windows station, has no local hard drive, or is operating an application with no local storage
- o Remote Desktop - controlling a remote node with local peripherals (i.e., monitor, keyboard and mouse)
- o Telemetry - a communication that allows remote measurement and reporting of information (e.g., post launch missile status or energy monitoring)

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Realtime Interactive	Inelastic, mostly variable rate, rate increases with user activity	Low	Very Low	Low

Figure 3. Realtime Interactive Traffic Characteristics

2.4 Multimedia-Streaming Category Traffic Class

Multimedia-Streaming traffic class is best suited for variable rate elastic streaming media applications where a human is waiting for output and where the application has the capability to react to packet loss by reducing its transmission rate. The following application components are appropriate for use with the Multimedia-Streaming category:

- o Audio (see Section 2.1)
- o Video (see Section 2.1)
- o Multiplex (i.e., combined a/v streams)

With adjective substrings to the above (which may or may not get IANA registered)

Webcast

The primary difference from the Multimedia-streaming category class and the Broadcast category class is about the length of time for buffering. Buffered streaming audio and/or video which are initiated by SDP, and not HTTP. Buffering here can be from many seconds to

hours, and is typically at the destination end (as opposed to Broadcast buffering which is minimal at the destination). The buffering aspect is what differentiates this category class from the Broadcast class (which has minimal or no buffering).

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Multimedia Streaming	Variable size packets, elastic with variable rate	Low - Medium	Medium - High	High

Figure 4. Multimedia Streaming Traffic Characteristics

2.5 Broadcast Category Traffic Class

Broadcast traffic class is best suited for inelastic streaming media Applications, which might have a 'wardrobe malfunction' delay at or near the source but not typically at the destination, that may be of constant or variable rate, requiring low jitter and very low packet loss. The following application components are appropriate for use with the Broadcast category:

- o Audio (see Section 2.1)
- o Video (see Section 2.1)
- o Multiplex (i.e., combined a/v streams)

With adjective substrings to the above:

- o IPTV
- o Live events (non-buffered)
- o surveillance - one way audio from a microphone or video from a camera (e.g., observing a parking lot or building exit), typically enabled for long periods of time, usually stored at the destination.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
Broadcast	Constant and variable rate, inelastic, generally non-bursty flows, generally sustained high packet rate, low inter-packet transmission	Very Low	Low - Medium	Low - Medium

	interval, usually UDP framed				
	in (S)RTP				
+-----+-----+-----+-----+-----+					

Figure 5. Broadcast Traffic Characteristics

3. SDP Attribute Definition

This document proposes the 'trafficclass' session and media-level SDP [RFC4566] attribute. The following is the Augmented Backus-Naur Form (ABNF) [RFC5234] syntax for this attribute, which is based on the SDP [RFC4566] grammar:

```

attribute                =/ traffic-class-label

traffic-class-label      = "trafficclass" ":" [SP] category
                          "." application *( "." adjective )

category                 = "Broadcast" /
                          "Realtime-Interactive" /
                          "Multimedia-Conferencing" /
                          "Multimedia-Streaming" /
                          "Conversational" / tcl-token

application              = tcl-token

adjective                = classified-adjective /
                          unclassified-adjective

classified-adjective     = tcl-token ":" tcl-token

unclassified-adjective   = tcl-token

tcl-token                = %2D / %30-%39 / %41-%5A / %61-7A

```

The attribute is named "trafficclass", for traffic classification, identifying which one of the five categories applies to the media stream associated with this m-line. There MUST NOT be more than one category component per media line.

The category in this document are an augmented version of the application labels introduced by table 3 of RFC 4595 (which will be rewritten based on the updated labels and treatments expected for each traffic class defined in this document).

+-----+-----+-----+-----+-----+	
Application Labels	Category Classes Defined
Defined in RFC 4594	in this document
+=====+=====+=====+=====+=====+	
Broadcast-video	Broadcast

Realtime-Interactive	Realtime-Interactive	
Multimedia-Conferencing	Multimedia-Conferencing	
Multimedia-Streaming	Multimedia-Streaming	
Telephony	Conversational	

Figure 6. Label Change Differences from RFC 4594

As is evident from the changes above, from left to right, two labels are different and each of the meanings are different in this document relative to how RFC 4594 defined them. These differences are articulated in Section 2 of this document.

A category is a human understandable categorization, and MUST NOT be the only component of the traffic class label present in the attribute. The category string MUST always be paired with an application component, with a "." as the component separator.

The application types define the application of a particular traffic flow, for example, audio or video. The application types are listed and defined in Section 2 of this document. Not every category is paired with application each listed, at least as defined in this document. Section 2.1 through 2.5 list many of the expected combinations.

For additional application type granularity, adjective components can be added (also listed in Section 2). One or more adjectives can be within the same traffic class attribute. It is also permitted to include one or more non-IANA registered adjective component, but these MUST be prefaced by the additional delimiter "_", creating a possibility such as

```
category.application-type.adjective._non-standard-adjective
                        ^^^^
```

See the underscore

For example, this is valid:

```
m=video 50000 RTP/AVP 112
a=trafficclass Conversational.video.immersive._foo._bar
```

where both "foo" and "bar" are not IANA registered adjectives, but "immersive" is IANA registered. However, including non-registered adjectives without the "_" delimiter MUST NOT occur, such as the following:

```
m=video 50000 RTP/AVP 112
a=trafficclass Conversational.video.immersive.foo.bar
```

There is no limit to the number of adjectives allowed, without regard for whether they are registered or not. These non-registered adjectives can be vendor generated, or merely considered to be proprietary in nature.

It is important to note that the order of component types matter, but not the order of the adjective components. There might be local significance to the ordering of adjectives though. In other words, the category class component **MUST** be before the application component, which **MUST** be before any and all adjective component(s).

Some algorithm such as alphabetizing the list and matching the understood strings **SHOULD** be used.

Adjectives can be either unqualified or qualified. Qualified adjectives have a delimiter ":" after the previous "." separating the string component into two parts.

The tcl-token "aq" is the first part of an adjective if it is qualified, and either the "admitted", "non-admitted" or "none" tcl-token is the second part of the qualified adjective allowable according to this specification. In the form

aq:admitted|non-admitted|none

The only valid use of the tcl-token "aq" is to pair with either the "admitted", "non-admitted" or "none" tcl-token. At the same time, the tcl-tokens "admitted", "non-admitted" or "none" **MUST NOT** appear without a preceding "aq:".

Like all adjectives, it is **OPTIONAL** to include this adjective in any trafficclass attribute, and has the following meanings:

- aq - for 'admission qualifier' to indicate the purpose of the following adjective parts with respect to the capacity admission status of this traffic flow described by this m-line.
- admitted - capacity admission mechanisms or protocols are to be or were used for the full amount of bandwidth in relation to this m= line.
- non-admitted - capacity admission mechanisms or protocols were attempted but failed in relation to this m= line. This does not mean the flow described by this m= line failed. It just failed to attain the capacity admission mechanism or protocol necessary for a predictable quality of service, and is likely to continue with only a class of service marking or best effort.
- none - no capacity admission mechanisms or protocols are or

were attempted in relation to this m= line.

The default for any flow generated from an m-line not having a trafficclass adjective of 'aq:admitted' or 'aq:non-admitted' MUST be the equivalent of 'aq:none', whether or not it is present.

Any category class, application, or adjective string component within this attribute that is not understood MUST be ignored, leaving all that is understood to be processed. Ignored string components SHOULD NOT be deleted, as a downstream entity could understand the component(s) and use it/them during processing.

Not understanding the category class string SHOULD mean that this attribute is ignored.

The following is an example of media level description with a 'trafficclass' attribute:

```
m=video 50000 RTP/AVP 112
a=trafficclass conversational.video.immersive.aq:admitted
```

The above indicates a Telepresence session that has had capacity admission process applied to its media flow.

4. Offer/Answer Behavior

Through the inclusion of the 'trafficclass' attribute, an offer/answer exchange identifies the application type for use by endpoints within a session. Policy elements can use this attribute to determine the acceptability and/or treatment of that session through lower layers. One specific use-case is for setting of the DSCP specific for that application type (say a Broadcast instead of a Conversational video), decided on a per domain basis - instead of exclusively by the offering domain.

4.1 Offer Behavior

Offerers include the 'trafficclass' attribute with a single string comprised of two or more components (from the list in Section 2) to obtain configurable and predictable classification between the answerer and the offerer. The offerer can also include a private set of components, or a combination of IANA registered and private components within a single domain (e.g., enterprise networks).

Offerers of this 'trafficclass' attribute MUST NOT change the label in transit (e.g., wrt to B2BUAs). Session Border Controllers (SBC) at domain boundaries can change this attribute through local policy.

Offers containing a 'trafficclass' label not understood are ignored by default (i.e., as if there was no 'trafficclass' attribute in the

offer).

4.2 Answer Behavior

Upon receiving an offer containing a 'trafficclass' attribute, if the offer is accepted, the answerer will use this attribute to classify the session or media (level) traffic accordingly towards the offerer. This answer does not need to match the traffic class in the offer, though this will likely be the case most of the time.

In order to understand the traffic class attribute, the answerer MUST check several components within the attribute, such as

- 1 - does the answerer understand the category component?

If not, the attribute SHOULD be ignored.

If yes, it checks the application component.

- 2 - does the answerer understand the application component?

If not, the answerer needs to check if it has a local policy to proceed without an application component. The default for this situation is as if the category component was not understood, the attribute SHOULD be ignored.

If yes, it checks to see if there are any adjective components present in this attribute to start its classification.

- 3 - does the answerer understand the adjective component or components if any are present?

If not present, process and match the trafficclass label value as is.

If yes, determine if there is more than one. Search for each that is understood. Any adjectives not understood are to be ignored, as if they are not present. Match all remaining understood components according to local policy and process attribute.

The answerer will answer the offer with its own 'trafficclass' attribute, which will likely be the same value, although this is not mandatory (at this time). The Offerer will process the received answer just as the answerer processed the offer. In other words, the processing steps and rules are identical for each end.

The answerer should expect to receive RTP packets marked as indicated by its 'trafficclass' attribute in the answer itself.

An Answer MAY have a 'trafficclass' attribute when one was not in

the offer. This will at least aid the local domain, and perhaps each domain the session transits, to categorize the application type of this RTP session.

Answerers that are middleboxes can use the 'trafficclass' attribute to classify the RTP traffic within this session however local policy determines. In other words, this attribute can help in deciding which DSCP an RTP stream is assigned within a domain, if the answerer were an inbound SBC to a domain.

5. Security considerations

RFC 5897 [RFC5897] discusses many of the pitfalls of service classification, which is similar enough to this idea of traffic classification to apply here as well. That document highly recommends the user not being able to set any classification. Barring a hack within an endpoint (i.e., to intentionally misclassifying (i.e., lying) about which classification an RTP stream is), this document's solution makes the classification part of the signaling between endpoints, which is recommended by RFC 5897.

6. IANA considerations

6.1 Registration of the SDP 'trafficclass' Attribute

This document requests IANA to register the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: jmpolk@cisco.com

Attribute name: trafficclass

Long-form attribute name: Traffic Classification

Type of attribute: Session and Media levels

Subject to charset: No

Purpose of attribute: To indicate the Traffic Classification application for this session

Allowed attribute values: IANA Registered Tokens

Registration Procedures: Specification Required

Type	SDP Name	Reference
----	-----	-----
att-field (both session and media level)		

trafficclass

[this document]

6.2 The Traffic Classification Category Registration

This document requests IANA to create a new registry for the traffic Category classes similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Category Attribute Values

Reference: [this document]

Registration Procedures: Standards-Track document Required

Category Values	Reference
-----	-----
Broadcast	[this document]
Realtime-Interactive	[this document]
Multimedia-Conferencing	[this document]
Multimedia-Streaming	[this document]
Conversational	[this document]

6.3 The Traffic Classification Application Type Registration

This document requests IANA to create a new registry for the traffic application classes similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" Attribute Application Type Values

Reference: [this document]

Registration Procedures: Specification Required

Application Values	Reference
-----	-----
Audio	[this document]
Video	[this document]
Text	[this document]
Application-sharing	[this document]
Presentation-data	[this document]
Whiteboarding	[this document]
Webchat/IM	[this document]
Gaming	[this document]
Virtualized-desktop	[this document]
Remote-desktop	[this document]
Telemetry	[this document]
Multiplex	[this document]
Webcast	[this document]
IPTV	[this document]
Live-event	[this document]
surveillance	[this document]

6.4 The Traffic Classification Unqualified Adjective Registration

This document requests IANA to create a new registry for the traffic adjective values similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" Attribute Adjective Values

Reference: [this document]

Registration Procedures: Specification Required

Adjective Values	Reference
-----	-----
Immersive	[this document]
Desktop-video	[this document]
Realtime-Text	[this document]
web	[this document]
aq	[this document]
admitted	[this document]
non-admitted	[this document]
none	[this document]

7. Acknowledgments

To Dave Oran, Toerless Eckert, Henry Chen, David Benham, David Benham, Mo Zanty, Michael Ramalho, Glen Lavers, Charles Ganzhorn, Paul Kyzivat and Greg Edwards for their comments and suggestions.

8. References

8.1. Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997
- [RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997
- [RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474, December 1998
- [RFC2872] Y. Bernet, R. Pabbati, "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

- [RFC4080] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005
- [RFC4124] F. Le Faucheur, Ed., " Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering ", RFC 4124, June 2005
- [RFC4566] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006
- [RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010
- [RFC5897] J. Rosenberg, "Identification of Communications Services in the Session Initiation Protocol (SIP)", RFC 5897, June 2010

8.2. Informative References

- [RFC4594] J. Babiarez, K. Chan, F Baker, "Configuration Guidelines for Diffserv Service Classes", RFC 4594, August 2006
- [ID-RSVP-PROF] J. Polk, S. Dhesikan, "Resource Reservation Protocol (RSVP) Application-ID Profiles for Voice and Video Streams", work in progress, Mar 2011

Author's Addresses

James Polk
3913 Treemont Circle
Colleyville, Texas, USA
+1.818.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan
170 W Tasman St
San Jose, CA, USA
+1.408-902-3351

mailto: sdhesika@cisco.com

Paul E. Jones
7025 Kit Creek Rd.
Research Triangle Park, NC, USA
+1 919 476 2048

mailto: paulej@packetizer.com

Appendix - Changes from Previous Versions

A.1 From -01 to -02

These are the following changes made between the WG -01 version and the -02 version:

- converged the use of terms 'parent' and 'category' to just 'category' for consistency.
- changed ABNF to reflect extensibility by not having applications and adjectives named in the ABNF, rather have them merely IANA registered.
- merged the qualified and unqualified adjective sections into a single section on adjectives, but allowing some to have a preceding qualifier.
- text clean-up

A.2 From -00 to -01

These are the following changes made between the WG -00 version and the -01 version:

- removed the non-SDP applications Netflix and VOD
- switched the adjective 'desktop' to 'avconf'
- Labeled each of the figures.
- clarified the differences between Multimedia-Streaming and Broadcast category categories.
- defined Video surveillance
- added the concept of a 'qualified' adjective, and modified the ABNF.
- deleted the idea of a 'cac-class' as a separate component, and made the equivalent a qualified adjective.
- modified the answerer behavior because of the removal of the

'cac-class' component.

- created an IANA registry for qualified adjectives
- general clean-up of the doc.

Did **not** do the following in this version:

- add the ability to have more than one trafficclass attribute based on the codec chosen, as feedback indicated this was a bad idea.
- no swap of the Multimedia-Conferencing category with the offered Collaboration category, as doing this did not solve any perceived problems.
- add more to the 'how does this get processed' portion of Section 3. That will come in the next revision.

Network WG
Internet-Draft
Expires: January 3, 2015
Intended Status: Standards Track (PS)

James Polk
Subha Dhesikan
Paul Jones
Cisco Systems
July 3, 2014

The Session Description Protocol (SDP) 'trafficclass' Attribute
draft-ietf-mmusic-traffic-class-for-sdp-05

Abstract

This document proposes a new Session Description Protocol (SDP) attribute to identify the traffic class a session is requesting in its offer/answer exchange.

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 July 4, 2015.

Copyright Notice

Copyright (c) 2014 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

1. Introduction	2
2. Traffic Class Framework and Component Definitions	5
3. Traffic Class Attribute Definition	6
3.1 Categories within the SDP Traffic Class Label	8
3.2 Applications within the SDP Traffic Class Label	9
3.3 Adjectives within the SDP Traffic Class Label	9
3.3.1 Qualified Adjectives	9
4. Matching Categories with Applications and Adjectives	11
4.1 Conversational Category Traffic Class	11
4.2 Multimedia-Conferencing Category Traffic Class	12
4.3 Realtime-Interactive Category Traffic Class	14
4.4 Multimedia-Streaming Category Traffic Class	15
4.5 Broadcast Category Traffic Class	17
4.6 Intermittent Category Traffic Class	18
5. Offer/Answer Behavior	19
5.1 Offer Behavior	20
5.2 Answer Behavior	20
6. Security considerations	21
7. IANA considerations	21
8. Acknowledgments	25
9. References	25
9.1. Normative References	25
9.2. Informative References	26
Authors' Addresses	26
Appendix	27

1. Introduction

The Session Description Protocol (SDP) [RFC4566] provides a means for an offerer to describe the specifics of a session to an answerer, and for the answerer to respond back with its session specifics to the offerer. These session specifics include offering the codec or codecs to choose from, the specific IP address and port number the offerer wants to receive the RTP stream(s) on/at, the particulars about the codecs the offerer wants considered or mandated, and so on.

There are many facets within SDP to determine the Real-time Transport Protocol (RTP) [RFC3550] details for the session establishment between one or more endpoints, but identifying how the underlying network should process each stream still remains under-specified.

The ability to identify a traffic flow by port number gives an

indication to underlying network elements to treat traffic with dissimilar ports in a different way, the same or in groups the same - but different from other ports or groups of ports.

Within the context of realtime communications, the labeling of an RTP session based on media descriptor lines as just a voice and/or video session is insufficient, and provides no guidelines to the underlying network on how to treat the traffic. A more granular labeling helps on several fronts to

- inform application layer elements in the signaling path the intent of this session.
- inform the network on how to treat the traffic if the network is configured to differentiate session treatments based on the type of session the RTP is, including the ability to provide call admission control based on the type of traffic in the network.
- allow network monitoring/management of traffic types realtime and after-the-fact analysis.

Some network operators want the ability to guarantee certain traffic gets a minimum amount of network bandwidth per link or through a series of links that make up a network such as a campus or WAN, or a backbone. For example, a call center voice application might get at least 20% of the available link bandwidth.

Some network operators want the ability to allow certain users or devices access to greater bandwidth during non-busy hours than during busy hours of the day. For example, all desktop video might operate at 1080p during non-peak times, but a similar session might be curtailed between the same users or devices to 720p or 360p during peak hours. Another example would be to reduce the frames per second (fps) rate, say from 30fps to 15fps. This case is not as clear as accepting or denying similar sessions during different times of the day, but tuning the access to the bandwidth based on the type of session. In other words, tune down the bandwidth for desktop video during peak hours to allow a 3-screen Telepresence session that would otherwise look like the same type of traffic (RTP, and more granular, video).

RFC 4594 established a guideline for classifying the various flows in the network and the Differentiated Services Codepoint (DSCP) values that apply to many traffic types (table 3 of [RFC4594]), including RTP based voice and video traffic sessions. The RFC also defined the per hop network behavior that is strongly encouraged for each of these application traffic types based on the traffic characteristics and tolerances to delay, loss and jitter within each traffic class.

Video was broken down into four categories in that RFC, and voice in another single category. We do not believe this satisfies the

technical and business requirements to accomplish sufficiently unique labeling of RTP traffic.

If the application becomes aware of traffic labeling,

- this can be coded into layer 3 mechanisms.
- this can be coded into layer 4 protocols and/or mechanisms.
- this can be coded into a combination of mechanisms and protocols.

A lower layer mechanism for differentiating traffic is either the port number or the Differentiated Services Codepoint (DSCP) value [RFC2474]. Within the public Internet, if the application is not part of a managed service, the DSCP value likely will be best effort (BE), or reset to BE, at ingress to a provider's network. Within the corporate LAN, this is usually completely configurable and a local IT department can take full advantage of this labeling to shape and manage their network as they see fit.

Within a network core, DiffServ typically does not apply. That said, DiffServ can be used to identify which traffic goes into which MPLS tunnel [RFC4124].

Labeling realtime traffic types using a layer 4 protocol would likely involve RSVP [RFC2205] or NSIS [RFC4080]. RSVP has an Application Identifier (app-ID) defined in [RFC2872] that provides a means for carrying a traffic class label along the media path. An advantage of this mechanism is that the label can inform each domain along the media path what type of traffic this traffic flow is, and allow each domain to adjust the appropriate DSCP value (set by each domain for use within that domain). Meaning, if a DSCP value is set by an endpoint or a router in the first domain and gets reset by a service provider, the far-end domain will be able to reset the DSCP value appropriate for the intended traffic class. There is a proposed extension to RSVP which creates individual profiles for what goes into each app-ID field to describe these traffic classes [ID-RSVP-PROF], which will take advantage of what is described in this document.

There are several proprietary mechanisms that can take advantage of this labeling, but none of those will be discussed here.

The idea of traffic - or service - identification is not new; it has been described in [RFC5897]. If that RFC is used as a guideline, identification that leads to stream differentiation can be quite useful. One of the points within RFC 5897 is that users cannot be allowed to assign any identification (fraud is one reason given). In addition, RFC 5897 recommends that service identification should be done in signaling, rather than guessing or deep packet inspection. Currently, any network would have to guess or perform deep packet inspection to classify traffic and offer the service as per

RFC 4594 as such service identification information is currently not available in SDP and therefore to the network elements. Since SDP understands how each stream is created (i.e., the particulars of the RTP stream), this is the right place to have this service differentiated. Such service differentiation can then be communicated to and leveraged by the network.

[Editor's Note: the words "traffic" and "service" are similar enough that the above paragraph talks about RFC 5897's "service identification", but this document only discusses and proposes traffic indications in SDP.]

This document proposes a simple attribute line to identify the application a session is requesting in its offer/answer exchange. This document uses previously defined service class strings for consistency between IETF documents.

This document utilizes the traffic classes originally created in RFC 4594 in Section 2, enhancing each class with application identifiers and optional adjective strings. Section 3 defines the new SDP attribute "trafficclass". Section 4 discusses the offerer and answerer behavior when generating or receiving this attribute.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Traffic Class Framework and Component Definitions

The framework of the traffic class attribute will have at least two parts, called components, allowing for several more to be included further distinguishing a particular session's traffic classification from another session's traffic classification. The amount of indicated differentiation between sessions is not a goal, and should only have additional components for differentiation if there is a need to uniquely identify traffic in different sessions.

The intention is to have a category component (e.g., conversational) that identifies the traffic pattern for a session. Is the traffic within a session one-way or two-way? Can the traffic be buffered before reaching the destination or not? What is this session's tolerance to packet loss and can there be retransmissions?

The application component (e.g., video) identifies the basic type of traffic within a category. Is it media or data packets? If media, which type of media? If data packets, which application of data packets are in this session?

The optional adjective component(s) (e.g., immersive) help to

further refine the traffic within a session by providing more description. For instance, if a session is two-way voice, what additional information can be given about this particular session to refine its description? Is it part of a conference or telepresence session? Is it just standalone voice call? Has a capacity admission protocol or mechanism been applied to this session?

The 'traffic class label' (TCL) will have the following structure,

```
category.application(.adjective)(.adjective)...
```

[Editor's Note: the above is not the exact ABNF to be used.
The order is right. The category and application
MUST appear first (each only once) and zero or more
adjectives can appear following the application
component.]

Where

- 1) the 1st component is the category, and is mandatory;
- 2) the 2nd component is the application, and is mandatory;
- 3) an optional 3rd component or series of components are adjective(s) used to further refine the application component;

The construction of the traffic class label for Telepresence video would follow the minimum form of:

```
conversational.video.immersive
```

where there might be one or more adjective after '.immersive'.

There is no traffic class or DSCP value associated with just "conversational". There is a traffic class associated with "conversational.video", creating a differentiation between it and a "conversational.video.immersive" traffic class, which would have DSCP associated with the latter traffic class, depending on local policy. Each category component is defined below, as are several of application and adjective strings. This is shown in [ID-RSVP-PROF] for the RSVP mapping of distinguishable traffic types.

Mapping a specific Traffic Class Label to a DSCP value might be accomplished in any of the following ways:

- o statically within the offerer and/or answerer; or
- o taken from a local mapping table/file, which might be downloaded once, periodically or as changes in the network are observed; or
- o from feedback from the network.

3. Traffic Class Attribute Definition

This document defines the 'trafficclass' media-level SDP attribute. The following is the Augmented Backus-Naur Form (ABNF) [RFC5234] syntax for this attribute, which is based on the SDP [RFC4566] grammar:

```

attribute                =/ traffic-class-label

traffic-class-label      = "trafficclass" ":" [SP] category
                          "." application *( "." adjective )

category                 = "broadcast" /
                          "realtime-interactive" /
                          "multimedia-conferencing" /
                          "multimedia-streaming" /
                          "conversational" /
                          "intermittent" / tcl-token

application              = tcl-token

adjective                = classified-adjective /
                          unclassified-adjective

classified-adjective     = tcl-token ":" tcl-token

unclassified-adjective   = tcl-token

tcl-token                = ALPHA *( [ "-" ] ALPHA / DIGIT )

```

A TCL "component" is any of the following:

- category,
- application, or
- adjective (which is the only optional component, and can have zero or more of these type of components)

The attribute is named "trafficclass", for traffic classification, identifying which one of the six categories applies to the media stream associated with this m-line. There MUST NOT be more than one category component per SDP media line.

The categories in this document are an augmented version of the application labels introduced by table 3 of RFC 4594 (which will be rewritten based on the updated labels and treatments expected for each traffic class defined in this document).

Application Labels Defined in RFC 4594	Category Classes Defined in this document
broadcast-video	broadcast

realtime-interactive	realtime-interactive	
multimedia-conferencing	multimedia-conferencing	
multimedia-streaming	multimedia-streaming	
telephony	conversational	

Figure 1. Label Differences from RFC 4594

As is evident from the changes above, from left to right, two labels are different and each of the meanings are different in this document relative to how RFC 4594 defined them. These differences are articulated in Section 4 of this document.

Applications and adjectives are defined using the syntax of "tcl-token" defined above.

RFC 4566 defined SDP as case sensitive. Everything is here as well.

An algorithm such as alphabetizing the list of components and matching the understood strings SHOULD be used for determining the traffic within a session. Strings not understood by an entity MUST be ignored during processing, but MUST NOT be deleted.

Any category, application, or adjective string component within this attribute that is not understood MUST be ignored, leaving all that is understood to be processed. Ignored components SHOULD NOT be deleted, as a downstream entity could understand the component(s) and use it/them during processing.

The following is an example of media level description with a 'trafficclass' attribute:

```
m=video 50000 RTP/AVP 112
a=trafficclass:conversational.video.immersive.aq:admitted
```

The above indicates the video part of a Telepresence session that has had capacity admission process applied to its media flow.

3.1 Categories within the SDP Traffic Class Label

The category component within the traffic class attribute describes the type of communication that will occur within that session. It answers these questions, is the traffic

- one-way or two-or-more-way interactive?
- buffered or (virtually) non-buffered?

- media or non-media (data)?

The six category components of the traffic class attribute defined within this specification are as follows:

- o conversational
- o multimedia-conferencing
- o realtime-interactive
- o multimedia-streaming
- o broadcast
- o intermittent

Sections 4.1 through 4.6 define each of the above.

The category component **MUST NOT** be the only component present in a traffic class attribute. The category component **MUST BE** paired with an 'application' component to give enough meaning to the traffic class labeling goal.

Not understanding the category component **SHOULD** mean that this attribute is ignored, because of the information about the expected behavior of this communication flow is identified by or within that component.

3.2 Applications within the SDP Traffic Class Label

The application component identifies the application of a particular traffic flow, for example, audio or video. The application types are listed and defined in Section 4 of this document. Not every category is paired with every application listed, at least as defined in this document. One or more applications are inappropriate in one or more categories.

Section 4.1 through 4.6 list many of the expected combinations.

3.3 Adjectives within the SDP Traffic Class Label

For additional application type granularity, adjective components can be added. One or more adjectives can be within the same traffic class attribute to provide more differentiation.

It is important to note that while the order of component types matter, the order of the adjective components do not. In other words, the category class component **MUST** be before the application component, which **MUST** be before any and all adjective component(s).

There is no limit to the number of adjectives allowed.

Adjective components come in two versions, unqualified and

qualified. One has a prefix (qualified), the other (unqualified) does not. A defined qualified adjective MUST NOT appear without its qualifier name, even in future extensions to this specification. Some implementations will likely perform a search within this attribute for the presence of qualifiers, which might be as simple as searching for the ":" COLON character. Implementations will be confused with inconsistent coding, therefore strict adherence is necessary.

3.3.1 Qualified Adjectives

Adjectives can be either unqualified or qualified. Qualified adjectives have a delimiter ":" character between the "qualifier name" and the "qualifier value". As one example, we introduce in this specification the "admission qualifier" and it has a qualifier name of "aq". We also define several possible qualifier values for the admission qualifier, namely "admitted", "non-admitted", "partial", and "none". When present in a TCL component, the qualified adjectives look like these admission qualifier adjectives:

```
aq:admitted
aq:non-admitted
aq:partial
aq:none
```

Defining some adjectives as qualified adjectives allows entities processing the traffic class label to potentially recognize a particular qualifier name and act on it, even if it does not understand the qualifier value. In the future, a new admission qualifier value might be defined, e.g. "foo", and entities could at least recognize the admission qualifier adjective, even if it did not understand the qualifier value "foo".

Like all adjectives, it is OPTIONAL to include the admission qualifier adjective in any trafficclass attribute.

The admission qualifier and its qualifier values are defined as:

- aq - 'admission qualifier' - this is the qualifier name for the admission qualifier adjectives, wherein the following qualifier values indicate the admission status for the traffic flow described by this m-line.
- admitted - capacity admission mechanisms or protocols are to be or were used for the full amount of bandwidth in relation to this m= line.
- non-admitted - capacity admission mechanisms or protocols were attempted but failed in relation to this m= line. This does not mean the flow described by this m= line failed. It just failed to attain the capacity admission

mechanism or protocol necessary for a predictable quality of service, and is likely to continue with only a class of service marking or best effort.

- partial - capacity admission mechanisms or protocols are to be or were used for the part of the amount of bandwidth in relation to this m= line. All traffic above a certain amount will have no capacity admission mechanisms applied. In other words, there is more traffic sent than was agreed to. The burden is on the sender and receiver to deal with any sent and lost information.
- none - no capacity admission mechanisms or protocols are or were attempted in relation to this m= line.

The default for any flow generated from an m-line not having a trafficclass adjective of 'aq:admitted' or 'aq:non-admitted' MUST be the equivalent of 'aq:none', whether or not it is present.

4. Matching Categories with Applications and Adjectives

This section describes each component within this document, as well as provides the combinations of categories and applications and adjectives. Given that not every combination makes sense, we express the limits here - which will be IANA registered. The majority of these TCLs in this document are found in [ID-RSVP-PROF], where RSVP is appropriate. Look at that other document for example usage of a specified TCL here.

4.1 Conversational Category Traffic Class

The "conversational" traffic class is best suited for applications that require very low delay variation and generally intended to enable realtime, bi-directional person-to-person or multi-directional via an MCU communication. Conversational flows are inelastic, and with few exceptions, use a UDP transport.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
conversational	High priority, typically consistent sized packets (small audio samples produce small packets and large video samples produce large packets), generally sustained at a high packet rate, low inter-packet transmission interval	Very Low	Very Low	Very Low

Figure 2. Conversational Traffic Characteristics

The following application components are appropriate for use with the Conversational category:

- o audio (voice)
- o video
- o multiplex (i.e., combined a/v streams) an application wherein media of different forms (e.g., audio and video) is multiplexed within the same media flow.

With adjective substrings to the above

immersive (TP) - An interactive audio-visual communications experience between remote locations, where the users enjoy a strong sense of realism and presence between all participants by optimizing a variety of attributes such as audio and video quality, eye contact, body language, spatial audio, coordinated environments and natural image size.

avconf - An interactive audio-visual communication experience that is not immersive in nature, though can have a high resolution video component.

Category	Application	Adjective
conversational	audio	immersive
		avconf
		aq:admitted
		aq:non-admitted
		aq:partial
		aq:none
	video	immersive
		avconf
		aq:admitted
		aq:non-admitted
		aq:partial
		aq:none
	multiplex	immersive
		avconf
		aq:admitted
		aq:non-admitted
		aq:partial
		aq:none

+-----+-----+-----+-----+

Figure 3. Conversational Applications and Adjective Combinations

4.2 Multimedia-Conferencing Category Traffic Class

The "multimedia-conferencing" traffic class is best suited for applications that are generally intended for communication between human users, but are less demanding in terms of delay, packet loss, and jitter than what conversational applications require. These applications require low to medium delay and may have the ability to change encoding rate (rate adaptive) or transmit data at varying rates.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
multimedia- conferencing	Variable size packets, Variable transmit interval, rate adaptive, reacts to loss, often one-way or unidirectional	Low	Low	Low
		- Medium	- Medium	- Medium

Figure 4. Multimedia Conferencing Traffic Characteristics

Multimedia-conferencing flows are not media flows which are conversational in nature. Multimedia-conferencing flows are those data flows that are typically transmitted in parallel to currently active conversational media flows. For example, a two-way conference session in which the users share a presentation. The presentation part of that conference call uses the Multimedia-conferencing category, whereas the audio and any video uses the conversational category indication.

The following application components are appropriate for use with the Multimedia-Conferencing category:

- o application-sharing (that webex does or protocols like T.128) - An application that shares the output of one or more running applications or the desktop on a host. This can utilize vector graphics, raster graphics or video.
- o presentation-data - can be a series of still images; could be at a rapid or busty rate, just not a continuous 24 fps or greater.
- o presentation-video - motion video that is transmitted and rendered as part of a presentation.
- o presentation-audio - the audio that is transmitted and rendered as

part of a presentation.

- o whiteboarding - an application enabling the exchange of graphical information including images, pointers and filled and unfilled parametric drawing elements (points, lines, polygons and ellipses).
- o (RTP-based) file-transfer as defined in RFC 5547
- o instant messaging

Category	Application	Adjective
multimedia-conferencing	application-sharing	aq:admitted aq:non-admitted aq:partial aq:none
	whiteboarding	aq:admitted aq:non-admitted aq:partial aq:none
	presentation-data	aq:admitted aq:non-admitted aq:partial aq:none
	presentation-video	aq:admitted aq:non-admitted aq:partial aq:none
	presentation-audio	aq:admitted aq:non-admitted aq:partial aq:none
	instant-messaging	aq:admitted aq:non-admitted aq:partial aq:none
	file-transfer	aq:admitted aq:non-admitted aq:partial aq:none

Figure 5. Multimedia Conferencing Applications and Adjective Combinations

4.3 Realtime-Interactive Category Traffic Class

The "Realtime-Interactive" traffic class is intended for interactive variable rate inelastic applications that require low jitter and loss and very low delay. Many of the applications that use the Realtime-Interactive category use TCP or SCTP, even gaming, because lost packets is information that is still required - therefore it is retransmitted.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
realtime- interactive	Inelastic, mostly variable rate, rate increases with user activity	Low	Very Low	Low

Figure 6. Realtime Interactive Traffic Characteristics

The following application components are appropriate for use with the Realtime-Interactive category:

- o gaming - interactive player video games with other users on other hosts (e.g., Doom)
- o remote-desktop - controlling a remote node with local peripherals (i.e., monitor, keyboard and mouse)
- o telemetry - a communication that allows remote measurement and reporting of information (e.g., post launch missile status or energy monitoring)

With adjective substrings to the above

- o virtual - To be used with the remote-desktop application component specifically when the traffic is a virtual desktop similar to an X-windows station, has no local hard drive, or is operating a computer application with no local storage.

Category	Application	Adjective
realtime-interactive	gaming	aq:admitted aq:non-admitted aq:partial aq:none
	remote-desktop	virtual aq:admitted aq:non-admitted

		aq:partial
		aq:none
	telemetry	aq:admitted
		aq:non-admitted
		aq:partial
		aq:none

Figure 7. Realtime-Interactive Applications and Adjective Combinations

4.4 Multimedia-Streaming Category Traffic Class

The "multimedia-streaming" traffic class is best suited for variable rate elastic streaming media applications where a human is waiting for output and where the application has the capability to react to packet loss by reducing its transmission rate.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
multimedia-streaming	Variable size packets, elastic with variable rate	Low - Medium	Medium - High	High

Figure 8. Multimedia Streaming Traffic Characteristics

The following application components are appropriate for use with the Multimedia-Streaming category:

- o audio (see Section 4.1)
- o video (see Section 4.1)
- o webcast - is a media file distributed over the Internet or enterprise network using streaming media technology.
- o multiplex (see Section 4.1)

The primary difference between the multimedia-streaming category and the broadcast category is the length of time for buffering. Buffered streaming of audio and/or video which is often initiated by HTTP, and not SDP. Buffering here can be from many seconds to hours, and is typically at the destination end (as opposed to Broadcast buffering which is minimal at the destination). The buffering aspect is what differentiates this category class from the broadcast category (which has minimal or no buffering).

Category	Application	Adjective
multimedia-streaming	audio	aq:admitted aq:non-admitted aq:partial aq:none
	video	aq:admitted aq:non-admitted aq:partial aq:none
	webcast	aq:admitted aq:non-admitted aq:partial aq:none
	multiplex	aq:admitted aq:non-admitted aq:partial aq:none

Figure 9. Multimedia Streaming Applications and Adjective Combinations

4.5 Broadcast Category Traffic Class

The "broadcast" traffic class is best suited for inelastic streaming media Applications, which might have a 'wardrobe malfunction' delay at or near the source but not typically at the destination, that may be of constant or variable rate, requiring low jitter and very low packet loss.

See Section 4.4 for the difference between Multimedia-Streaming and Broadcast; it all has to do with buffering.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
broadcast	Constant and variable rate, inelastic, generally non-bursty flows, generally sustained high packet rate, low inter-packet transmission interval	Very Low	Low - Medium	Low - Medium

Figure 10. Broadcast Traffic Characteristics

The following application components are appropriate for use with the Broadcast category:

- o audio (see Section 4.1)
- o video (see Section 4.1)
- o multiplex (see Section 4.1)

With adjective substrings to the above:

- o live (non-buffered) - refers to various types of media broadcast without a significant delay, typically measured in milliseconds to a few seconds only.
- o surveillance - one way audio from a microphone or video from a camera (e.g., observing a parking lot or building exit), typically enabled for long periods of time, usually stored at the destination.

Category	Application	Adjective
broadcast	audio	surveillance
		live
		aq:admitted
		aq:non-admitted
		aq:partial
		aq:none
	video	surveillance
		live
		aq:admitted
		aq:non-admitted
		aq:partial
		aq:none
	multiplex	surveillance
		live
		aq:admitted
		aq:non-admitted
		aq:partial
		aq:none

Figure 11. Broadcast Applications and Adjective Combinations

4.6 Intermittent Category Traffic Class

The "intermittent" traffic class is best suited for inconstant rate applications such as those from a sensor device, where tolerance to loss, delay and jitter is often medium to high in nature. This category is not to be used for bulk file transfers, rather it can be sometimes bursty for brief periods of time, but then not produce traffic for short or long (i.e., hours or days) durations. Nor is this category to be used for any kind of regular paced rate of transmission, no matter how long the interval.

Traffic Class Name	Traffic Characteristics	Tolerance to		
		Loss	Delay	Jitter
intermittent	Inconstant rate, infrequent but sometimes bursty flows, generally non-regular, variable inter-packet transmission interval	Medium - High	Medium - High	High

Figure 12. Intermittent Traffic Characteristics

The following application components are appropriate for use with the Broadcast category:

- o text (i.e., text required by deaf users) a term for seemingly real-time transmission of text in a character-by-character fashion, often as a text equivalent to voice-based conversational services, without the timing constraints of conversational text is defined in the ITU-T Framework for multimedia services, Recommendation F.700 [RFC5194].
- o sensor - a flow containing information obtained from a sensor, such as a temperature or motion sensor.

With adjective substrings to the above:

- o there are no defined adjectives for the 'sensor' application at this time. There are many examples one could think would be viable adjectives, such as light, motion, temperature, magnetic fields, gravity, humidity, moisture, vibration, pressure, electrical fields, and other physical aspects of the external environment measured by the sensor.

Category	Application	Adjective
intermittent	sensor	(undefined at this time)

	text	aq:admitted aq:non-admitted aq:partial aq:none
--	------	---

Figure 13. Intermittent Applications and Adjective Combinations

5. Offer/Answer Behavior

Through the inclusion of the 'trafficclass' attribute, an offer/answer exchange identifies the application type(s) for use by the endpoints within the media streams of a session. Signaling elements can use this attribute to determine the acceptability and/or treatment of that session through lower layers, communicating a desired treatment for a particular flow to endpoints using SDP, interacting with network elements using some unspecified mechanism, or having endpoints communicate with network elements using some unspecified mechanism.

In order to understand the traffic class attribute, the SDP entity MUST check several components within the Traffic Class Label. By understand, we mean that the value of each component of the TCL is recognized, i.e., both the category and application components MUST be a recognized set for a TCL to be understood. Adjectives that are not recognized are simply ignored and MAY be discarded, however many there are. Adjectives which are not understood SHOULD NOT be discarded, as each/any adjective might be understood by some or all other downstream nodes in the signaling path.

The following pertains to both the receiver of an offer and receiver of an answer when either or both contain a Traffic Class Label attribute.

- 1 - can the receiver of the SDP containing a trafficclass attribute successfully process the category component?

If not, the attribute SHOULD be ignored.

If yes, it checks the application component.

- 2 - can the receiver of the SDP containing a trafficclass attribute successfully process the application component?

If not, the answerer needs to check if it has a local policy to proceed without an application component. The default for this situation is as if the category component was not understood, meaning the attribute SHOULD be ignored.

If yes, it checks to see if there are any adjective components

present in this attribute to start its classification.

- 3 - can the receiver of the SDP containing a trafficclass attribute successfully process the adjective component or components if any are present?

If not present, process and match the trafficclass label value as is.

If yes, determine if there is more than one. Search for each that is understood. Any adjectives not understood are to be ignored, as if they are not present. Match all remaining understood components according to local policy and process attribute.

5.1 Offer Behavior

Offerers include the 'trafficclass' attribute within a single string per m= line comprised of at least a category and application component (see Section 4) to establish the non-generic classification of the media stream between the answerer and the offerer. The offerer can also include one or more adjective components, which might be a combination of registered and private adjectives to further refine the identification of what this particular media stream is.

Session Border Controllers (SBC) at domain boundaries can change this attribute through local policy.

5.2 Answer Behavior

Upon receiving an offer containing a 'trafficclass' attribute, if the offer is accepted - including the ability to process the 3 bulleted rules in Section 5.0, the answerer will use this attribute to classify the media level traffic accordingly towards the offerer.

The answerer will answer the offer with its own 'trafficclass' attribute, which will likely be the same value, although this is not mandatory (at this time). The Offerer will process the received answer just as the answerer processed the offer. In other words, the processing steps and rules are identical for each end (see Section 5).

An Answer MAY have a 'trafficclass' attribute when one was not in the offer. This will at least aid the local domain, and perhaps each domain the session transits, to categorize and in some cases group the media-types of this session.

6. Security considerations

The security considerations from RFC 4566 are also applicable, particularly since intermediary devices might be able to look at an m= line and determine, not only is it audio, but that it is presentation-audio (i.e., 'multimedia-conferencing.presentation-audio') versus conversational audio.

RFC 5897 [RFC5897] discusses many of the pitfalls of service classification, which is similar enough to this idea of traffic classification to apply here as well. That document highly recommends the user not being able to set any classification. Barring a hack within an endpoint (i.e., to intentionally misclassifying (i.e., lying) about which classification an RTP stream is), this document's solution makes the classification part of the signaling between endpoints, which is recommended by RFC 5897.

7. IANA considerations

7.1 Registration of the SDP 'trafficclass' Attribute

This document requests IANA to register the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: jmpolk@cisco.com

Attribute name: trafficclass

Long-form attribute name: Traffic Classification

Type of attribute: Media levels

Subject to charset: No

Purpose of attribute: To indicate the Traffic Classification application for this session

Allowed attribute values: IANA Registered Tokens

Registration Procedures: (there are multiple RFC5226 registration procedures; see below within each sub-section)

Designated Experts: James Polk (jmpolk@cisco.com)
Paul Jones (paulej@packetizer.com)

Type	SDP Name	Reference
----	-----	-----
att-field	(media level)	

trafficclass

[this document]

7.2 The Traffic Classification Category Registration

This document requests IANA to create a new registry for the traffic category classes similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Category Attribute Values
Reference: [this document]
Registration Procedures: Standards Action Required [RFC5226]

Category Values	Reference
-----	-----
broadcast	[this document]
realtime-interactive	[this document]
multimedia-conferencing	[this document]
multimedia-streaming	[this document]
conversational	[this document]
intermittent	[this document]

7.3 The Traffic Classification Application Type Registration

This document requests IANA to create a new registry for the traffic application classes similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Application Attribute Values
Reference: [this document]
Registration Procedures: Specification Required [RFC5226]

Application Values	Reference
-----	-----
audio	[this document]
video	[this document]
text	[this document]
application-sharing	[this document]
presentation-data	[this document]
presentation-video	[this document]
presentation-audio	[this document]
whiteboarding	[this document]
instant-messaging	[this document]
gaming	[this document]
remote-desktop	[this document]
telemetry	[this document]
multiplex	[this document]
webcast	[this document]
sensor	[this document]

7.4 The Traffic Classification Adjective Registration

This document requests IANA to create a new registry for the traffic adjective values similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Adjective Attribute Values
Reference: [this document]
Registration Procedures: Specification Required [RFC5226]

Adjective Values	Reference
-----	-----
immersive	[this document]
avconf	[this document]
realtime	[this document]
web	[this document]
virtual	[this document]
live	[this document]
surveillance	[this document]
aq:admitted	[this document]
aq:non-admitted	[this document]
aq:partial	[this document]
aq:none	[this document]

7.5 The Traffic Classification Component Mapping

7.5.1 Broadcast Applications and Adjective Combinations

This document requests IANA to create a new registry for the Broadcast category mapping similar to Figure 11 in Section 4.5 of this document within the Session Description Protocol (SDP) Parameters registry:

Registry Name: Broadcast Applications and Adjective Combinations
Table
Reference: [this document]
Registration Procedures: Specification Required [RFC5226]

7.5.2 Realtime Interactive Applications and Adjective Combinations

This document requests IANA to create a new registry for the Realtime Interactive category mapping similar to Figure 7 in Section 4.3 of this document within the Session Description Protocol (SDP) Parameters registry:

Registry Name: Realtime Interactive Applications and Adjective
Combinations Table
Reference: [this document]
Registration Procedures: Specification Required [RFC5226]

7.5.3 Multimedia Conferencing Applications and Adjective Combinations

This document requests IANA to create a new registry for the Multimedia Conferencing category mapping similar to Figure 5 in Section 4.2 of this document within the Session Description Protocol (SDP) Parameters registry:

Registry Name: Multimedia Conferencing Applications and Adjective Combinations Table

Reference: [this document]

Registration Procedures: Specification Required [RFC5226]

7.5.4 Multimedia-Streaming Applications and Adjective Combinations

This document requests IANA to create a new registry for the Multimedia-Streaming category mapping similar to Figure 9 in Section 4.4 of this document within the Session Description Protocol (SDP) Parameters registry:

Registry Name: Multimedia-Streaming Applications and Adjective Combinations Table

Reference: [this document]

Registration Procedures: Specification Required [RFC5226]

7.5.5 Conversational Applications and Adjective Combinations

This document requests IANA to create a new registry for the conversational category mapping similar to Figure 3 in Section 4.1 of this document within the Session Description Protocol (SDP) Parameters registry:

Registry Name: Conversational Applications and Adjective Combinations Table

Reference: [this document]

Registration Procedures: Specification Required [RFC5226]

7.5.6 Intermittent Applications and Adjective Combinations

This document requests IANA to create a new registry for the intermittent category mapping similar to Table 13 in Section 4.6 of this document within the Session Description Protocol (SDP) Parameters registry:

Registry Name: Intermittent Applications and Adjective Combinations Table

Reference: [this document]

Registration Procedures: Specification Required [RFC5226]

7.6 Designated Expert Reviewers

The following will be the designated expert reviewers of new 'trafficclass' registry requests:

- James Polk <jmpolk@cisco.com>
- Paul E. Jones <paulej@packetizer.com>

There SHALL remain two designated Expert reviewers at all times. The MMUSIC WG chairs should be consulted for replacements, if one or both are needed.

8. Acknowledgments

To Dave Oran, Toerless Eckert, Henry Chen, David Benham, David Benham, Mo Zanty, Michael Ramalho, Glen Lavers, Charles Ganzhorn, Paul Kyzivat, Greg Edwards, Charles Eckel, Dan Wing, Cullen Jennings and Peter Saint-Andre for their comments and suggestions.

9. References

9.1. Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997
- [RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997
- [RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474, December 1998
- [RFC2872] Y. Bernet, R. Pabbati, "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC4080] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005
- [RFC4124] F. Le Faucheur, Ed., " Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering ", RFC 4124,

June 2005

- [RFC4566] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006
- [RFC5226] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008
- [RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5547] M. Garcia-Martin, M. Isomaki, G. Camarillo, S. Loreto, P. , Kyzivat, "A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer ", RFC 5547, May 2009
- [RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010
- [RFC5897] J. Rosenberg, "Identification of Communications Services in the Session Initiation Protocol (SIP)", RFC 5897, June 2010

9.2. Informative References

- [RFC4594] J. Babiarez, K. Chan, F Baker, "Configuration Guidelines for Diffserv Service Classes", RFC 4594, August 2006
- [ID-RSVP-PROF] J. Polk, S. Dhesikan, "Resource Reservation Protocol (RSVP) Application-ID Profiles for Voice and Video Streams", work in progress, Feb 2013

Author's Addresses

James Polk
3913 Treemont Circle
Colleyville, Texas, USA
+1.818.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan
170 W Tasman St
San Jose, CA, USA
+1.408-902-3351

mailto: sdhesika@cisco.com

Paul E. Jones
7025 Kit Creek Rd.
Research Triangle Park, NC, USA
+1 919 476 2048

mailto: paulej@packetizer.com

Appendix - Changes from Previous Versions

A.1 From -04 to -05

These are the following changes made between the WG -03 version and the -04 version:

- general clean-up of text.
- added presentation-video and presentation-audio to the multimedia-conferencing section.
- brought forward the text describing how a SDP entity handles receiving a session description containing the trafficclass attribute to Section 5, from 5.2.
- added RFC 5547 as a normative reference.
- expended the security considerations section.

A.2 From -03 to -04

These are the following changes made between the WG -03 version and the -04 version:

- minimal text changes.
- introduced the "intermittent" category based on IETF86 feedback in the WG.

A.3 From -02 to -03

These are the following changes made between the WG -02 version and the -03 version:

- Rearranged a fair amount of text
- Separated and defined the components into separate subsections.
- built 5 different tables, one per category, that lists within each category - what applications are appropriate as well as what adjectives are appropriate for each application within that

category.

- added the 'partial' admission qualifier for those flows that have only part of their respective flow admitted (i.e., CAC'd).

A.4 From -01 to -02

These are the following changes made between the WG -01 version and the -02 version:

- converged the use of terms 'parent' and 'category' to just 'category' for consistency.
- changed ABNF to reflect extensibility by not having applications and adjectives named in the ABNF, rather have them merely IANA registered.
- merged the qualified and unqualified adjective sections into a single section on adjectives, but allowing some to have a preceding qualifier.
- text clean-up

A.5 From -00 to -01

These are the following changes made between the WG -00 version and the -01 version:

- removed the non-SDP applications Netflix and VOD
- switched the adjective 'desktop' to 'avconf'
- Labeled each of the figures.
- clarified the differences between Multimedia-Streaming and Broadcast category categories.
- defined Video surveillance
- added the concept of a 'qualified' adjective, and modified the ABNF.
- deleted the idea of a 'cac-class' as a separate component, and made the equivalent a qualified adjective.
- modified the answerer behavior because of the removal of the 'cac-class' component.
- created an IANA registry for qualified adjectives
- general clean-up of the doc.

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: April 7, 2013

T. Reddy
P. Patil
D. Wing
Cisco
October 4, 2012

Happy Eyeballs Extension for ICE
draft-reddy-mmusic-ice-happy-eyeballs-00

Abstract

This document specifies requirements for algorithms that make ICE connectivity checks more aggressive to reduce delays in dual stack host connectivity checks when there is a path failure for the address family preferred by the application or by the operating system. As IPv6 is usually preferred, the procedures in this document helps avoid user-noticable delays when the IPv6 path is broken or excessively slow.

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 7, 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

1. Introduction	3
2. Notational Conventions	3
3. Candidates Priority	3
4. Algorithm overview	4
4.1. Processing the Results	5
5. Relayed Candidates	7
6. Setting Te, Tr and MAX_PAIRS_HAPPYEYE_STAGE	8
7. IANA Considerations	8
8. Security Considerations	8
9. References	8
9.1. Normative References	8
9.2. Informative References	9
Authors' Addresses	9

1. Introduction

In situations where there are many IPv6 addresses, ICE [RFC5245] will prefer IPv6 [RFC6724] and will attempt connectivity checks on all the IPv6 candidates before trying an IPv4 candidate. If the IPv6 path is broken, this fallback to IPv4 can consume a lot of time, harming user satisfaction of dual stack devices.

This document describes an algorithm that makes ICE connectivity checks more responsive to failures of an address family by performing connectivity checks with both IPv6 and IPv4 candidates in parallel if IPv6 connectivity checks have not yet succeeded. This document specifies requirements for any such algorithm, with the goals that the ICE agent need not be inordinately harmed with a simple parallelisation of IPv6 and IPv4 connectivity checks and ensuring that the priority of precedence defined in [RFC6724] be honored.

For either of the address families, there is also a very realistic chance that connectivity checks for relayed candidates will always work. There are scenarios where firewalls block connectivity checks for Host/Server Reflexive candidates or for IPv4 or for IPv6. This document also proposes an optimization where connectivity checks with relayed checks are performed earlier than usual if connectivity checks using other candidates do not succeed.

2. Notational Conventions

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].

This note uses terminology defined in [RFC5245].

3. Candidates Priority

A prioritization formula is used by ICE [RFC5245] so that most preferred address pairs are tested first, and if a sufficiently good pair is discovered, the tests can be stopped. With IPv6, addresses obtained from local network interfaces, called host candidates, are recommended as high-priority ones to be tested first since if they work, they provide usually the best path between the two hosts. The ICE specification recommends to use the rules defined in [RFC6724] as part of the prioritization formula for IPv6 host candidates and [I-D.keranen-mmusic-ice-address-selection] updates the ICE rules on how IPv6 host candidates are selected.

For dual stack hosts the preference for IPv6 host candidates is higher than IPv4 host candidates based on precedence value of IP addresses described in [RFC6724]. IPv6 server reflexive candidates have higher precedence than IPv4 server reflexive candidate since NPTv6 is stateless and transport-agnostic.

(highest)	IPv6 Host Candidate
	IPv4 Host Candidate
	IPv6 Server Reflexive Candidate
	IPv4 Server Reflexive Candidate
	IPv6 Relayed Transport Candidate
(lowest)	IPv4 Relayed Transport Candidate

Figure 1: Candidate Preferences in decreasing order

By using the technique in Section 4 IPv6 candidate pairs will be tested first as usual, but if connectivity checks are not successful after a certain period of time, the algorithm will become more aggressive and connectivity checks using IPv6/IPv4 host/server-reflexive candidates will be performed simultaneously. If connectivity checks with IPv6 candidate pairs do not yield any successful result then ICE endpoints can immediately start sending media using IPv4 host/server-reflexive candidates.

Note: [RFC6724] permits administrator to change the policy table to prefer IPv4 addresses over IPv6 addresses in which case the algorithm described in the next section is reversed.

4. Algorithm overview

The Happy Eyeballs Extension for ICE is governed by a timer (T_e) that is started just before carrying out the ICE connectivity checks for each check list under the following conditions:

1. when the candidates pairs include IPv6 and IPv4 addresses
2. list of IPv6 candidate pairs is higher than a configured threshold (`MAX_PAIRS_HAPPY_EYE_STAGE_I`). [RFC5245] recommends a limit of 100 for the candidate pairs.

When the timer (T_e) fires, if the connectivity check using IPv6 candidate pairs are not yet successful and if the number of IPv6 candidate pairs with remote candidates of type host in the check list that are in Waiting and Frozen state are non-zero, the ICE agent performs the following Happy Eyeball steps in parallel with the regular ICE Ordinary checks:

- o Find the highest priority pair in the checklist that is in the Waiting state with candidate address family being IPv4 and remote candidate of type host. If there are no remote IPv6 candidates of type server-reflexive then IPv4 remote candidates of type server-reflexive will be added to the search.
 - 1. If there is such a pair then perform ICE connectivity check on this pair and set the state of the candidate pair to In-Progress.
 - 2. If there is no such pair find the highest priority pair in the checklist that is in the Frozen state with candidate address family being IPv4 and remote candidate of type host candidate. If there are no remote IPv6 candidates of type server-reflexive then IPv4 remote candidates of type server-reflexive will be added to the search. If there is such pair in Frozen state then unfreeze the pair, perform connectivity check on this pair and set the state of the candidate pair to In-Progress.
- o The above mentioned steps will be followed every T_a milliseconds and stopped when any of the below conditions are met:
 - 1. All IPv6 candidate pairs with remote candidates of type host in the check list are in any of the following states Succeeded, In-Progress or Failed states. The parallel activity is not required beyond this point because the regular ICE algorithm will itself pick up IPv4 candidate pairs not yet tested.
 - 2. All IPv4 candidate pairs with remote candidates of type host/server reflexive are in any of the following states Succeeded, In-Progress or Failed states.

4.1. Processing the Results

If ICE connectivity checks using an IPv4 candidate is successful then ICE Agent will performs as usual "Discovering Peer Reflexive Candidates" (Section 7.1.3.2.1 of [RFC5245]), "Constructing a Valid Pair" (Section 7.1.3.2.2 of [RFC5245]), "Updating Pair States" (Section 7.1.3.2.3 of [RFC5245]), "Updating the Nominated Flag" (Section 7.1.3.2.4 of [RFC5245]).

If ICE connectivity checks using an IPv4 candidate is successful for each component of the media stream and connectivity checks using IPv6 candidates is not yet successful, the ICE endpoint will declare victory, conclude ICE for the media stream and start sending media using IPv4. However, it is also possible that ICE endpoint continues

to perform ICE connectivity checks with IPv6 candidate pairs and if checks using higher-priority IPv6 candidate pair is successful then media stream can be moved to the IPv6 candidate pair. Continuing to perform connectivity checks can be useful for subsequent connections, to optimize which connectivity checks are tried first. Such optimization is out of scope of this document.

The following diagram shows the behaviour during the connectivity check when Alice calls Bob and Agent Alice is the controlling agent and uses the aggressive nomination algorithm. "USE-CAND" implies the presence of the USE-CANDIDATE attribute.

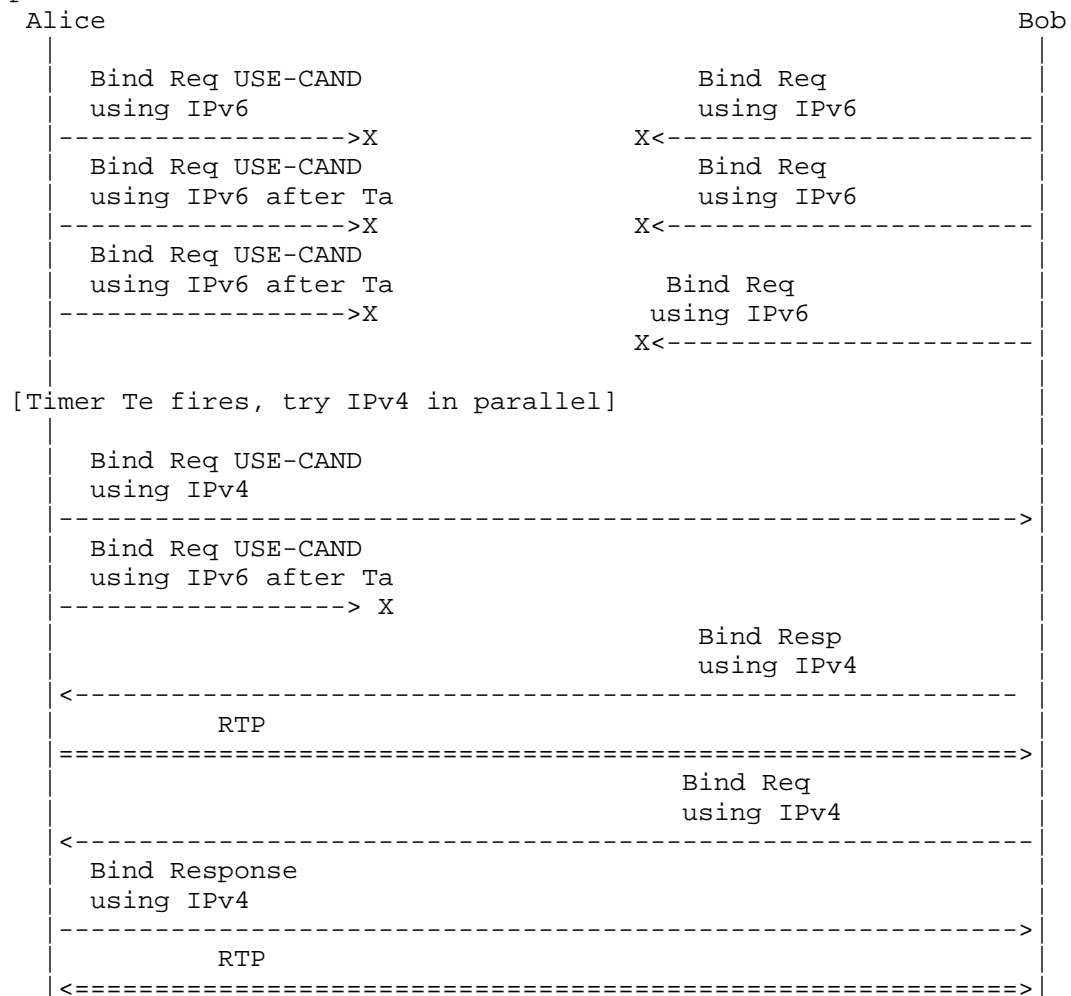


Figure 2: Happy Eyeballs Extension for ICE

5. Relayed Candidates

The optimization proposes doing connectivity checks with relayed candidates in parallel with other candidates. The algorithm does not make a distinction between IPv6/IPv4 relayed candidates and will choose the existing order among relayed candidate pair defined by ICE. If ICE connectivity check is successful using a relayed candidate from either of the IP address families, the ICE agent can stop connectivity checks for other relayed candidates.

This part of the Happy Eyeballs Extension for ICE is governed by a timer (Tr) that is started just before carrying out the ICE connectivity checks for each check list under the following conditions:

1. when the candidates pairs include IPv6 and IPv4 relayed addresses
2. list of candidate pairs is higher than a configured threshold (MAX_PAIRS_HAPPYEYE_STAGE_I).

When the timer (Tr) fires, If no ICE connectivity checks are successful as yet and if ICE Connectivity checks using IPv6 and IPv4 local relayed candidates have not yet been attempted then the following steps will be started by the ICE agent in parallel with other connectivity checks:

- o Find the highest priority pair in the checklist that is in the Waiting state with local candidate of type relayed.
1. If there is such a pair then perform ICE connectivity check on this pair and set the state of the candidate pair to In-Progress.
 2. If there is no such pair find the highest priority pair in the checklist that is in the Frozen state with local candidate of type relayed. If there is such pair in Frozen state then unfreeze the pair, perform connectivity check on this pair and set the state of the candidate pair to In-Progress.

If ICE connectivity checks using relayed candidate is successful then ICE Agent will performs as usual "Constructing a Valid Pair" (Section 7.1.3.2.2 of [RFC5245]), "Updating Pair States" (Section 7.1.3.2.3 of [RFC5245]), "Updating the Nominated Flag" (Section 7.1.3.2.4 of [RFC5245]). If ICE connectivity checks using local relayed candidates is successful for each component of the media stream and connectivity checks using higher priority candidate pairs has not yet succeeded then conclude ICE for the media stream and proceed to send media using local relayed candidate.

However ICE connectivity checks MUST be continued and if the check succeeds for a pair whose priority is higher than the previously selected candidate pair then media session will be moved to this pair. Hence media will only be sent briefly on TURN relays. Additional TURN server load is created due to this recommendations, especially when connectivity check using IPv6/IPv4 host/server-reflexive candidates are not completing quickly and the side affect could be that RTP receivers will receive packets out of order during switchover.

6. Setting Te, Tr and MAX_PAIRS_HAPPYEYE_STAGE

The value of Ta, Tr, MAX_PAIRS_HAPPYEYE_STAGE_I, MAX_PAIRS_HAPPYEYE_STAGE_II and SHOULD be configurable, and SHOULD have a default of:

```
Te : 150ms
Tr : 500ms
MAX_PAIRS_HAPPYEYE_STAGE_I : 12
MAX_PAIRS_HAPPYEYE_STAGE_II : 6
```

Figure 3: Default Values

7. IANA Considerations

None.

8. Security Considerations

STUN connectivity check using MAC computed during key exchanged in the signaling channel provides message integrity and data origin authentication as described in section 2.5 of [RFC5245] apply to this use.

9. References

9.1. Normative References

[I-D.keranen-mmusic-ice-address-selection]
Keranen, A. and J. Arkko, "Update on Candidate Address Selection for Interactive Connectivity Establishment (ICE)", draft-keranen-mmusic-ice-address-selection-01 (work in progress), July 2012.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

9.2. Informative References

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marthalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: praspatti@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: January 1, 2015

T. Reddy
P. Patil
P. Martinsen
Cisco
June 30, 2014

Happy Eyeballs Extension for ICE
draft-reddy-mmusic-ice-happy-eyeballs-07

Abstract

This document provides guidelines on how to make Interactive Connectivity Establishment (ICE) conclude faster in IPv4/IPv6 dual-stack scenarios where broken paths exist. The provided guidelines are backwards compatible with the original ICE specification.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 1, 2015.

Copyright Notice

Copyright (c) 2014 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

1. Introduction	2
2. Notational Conventions	3
3. Improving ICE Dual-stack Fairness	3
4. Compatibility	3
5. IANA Considerations	5
6. Security Considerations	5
7. Acknowledgements	5
8. Normative References	6
Appendix A. Example Algorithm for Choosing the Local Preference	6
Authors' Addresses	7

1. Introduction

There is a need to introduce more fairness in the handling of connectivity checks for different IP address families in dual-stack IPv4/IPv6 ICE scenarios. Section 4.1.2.1 of ICE [RFC5245] points to [RFC3484] for prioritizing among the different IP families. [RFC3484] is obsoleted by [RFC6724] but following the recommendations from the updated RFC will lead to prioritization of IPv6 over IPv4 for the same candidate type. Due to this, connectivity checks for candidates of the same type (host, reflexive or relay) are sent such that an IP address family is completely depleted before checks from the other address family are started. This results in user noticeable setup delays if the path for the prioritized address family is broken.

To avoid such user noticeable delays when either IPv6 or IPv4 path is broken or excessive slow, this specification encourages intermingling the different address families when connectivity checks are performed. Introducing IP address family fairness into ICE connectivity checks will lead to more sustained dual-stack IPv4/IPv6 deployment as users will no longer have an incentive to disable IPv6. The cost is a small penalty to the address type that otherwise would have been prioritized.

The guidelines outlined in this specification are backward compatible with a standard ICE implementation. This specification only alters the values used to create the resulting checklists in such a way that the core mechanisms from ICE [RFC5245] are still in effect. The introduced fairness might be better, but not worse than what exists today.

2. Notational Conventions

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].

This document uses terminology defined in [RFC5245].

3. Improving ICE Dual-stack Fairness

Candidates SHOULD be prioritized such that a long sequence of candidates belonging to the same address family will be intermingled with candidates from an alternate IP family. For example, promoting IPv4 candidates in the presence of many IPv6 candidates such that an IPv4 address candidate is always present after a small sequence of IPv6 candidates, i.e., reordering candidates such that both IPv6 and IPv4 candidates get a fair chance during the connectivity check phase. This makes ICE connectivity checks more responsive to broken path failures of an address family.

An ICE agent can choose an algorithm or a technique of its choice to ensure that the resulting check lists have a fair intermingled mix of IPv4 and IPv6 address families. Modifying the check list directly can lead to uncoordinated local and remote check lists that result in ICE taking longer to complete or in the worst case scenario fail. The best approach is to modify the formula for calculating the candidate priority value described in ICE [RFC5245] section 4.1.2.1.

4. Compatibility

ICE [RFC5245] section 4.1.2 states that the formula in section 4.1.2.1 SHOULD be used to calculate the candidate priority. The formula is as follows:

$$\begin{aligned} \text{priority} = & (2^{24}) * (\text{type preference}) + \\ & (2^8) * (\text{local preference}) + \\ & (2^0) * (256 - \text{component ID}) \end{aligned}$$

ICE [RFC5245] section 4.1.2.2 has guidelines for how the type preference and local preference value should be chosen. Instead of having a static local preference value for IPv4 and IPv6 addresses, it is possible to choose this value dynamically in such a way that IPv4 and IPv6 address candidate priorities ends up intermingled within the same candidate type.

It is also possible to dynamically change the type preference in such a way that IPv4 and IPv6 address candidates end up intermingled regardless of candidate type. This is useful if there are a lot of

IPv6 host candidates effectively blocking connectivity checks for IPv4 server reflexive candidates.

The list below shows a sorted local candidate list where the priority is calculated in such a way that the IPv4 and IPv6 candidates are intermingled. To allow for earlier connectivity checks for the IPv4 server reflexive candidates, some of the IPv6 host candidates was demoted. This is just an example of how a candidate priorities can be calculated to provide better fairness between IPv4 and IPv6 candidates without breaking any of the ICE connectivity checks.

	Candidate Type	Address Type	Component ID	Priority
(1)	HOST	IPv6	(1)	212928947
(2)	HOST	IPv6	(2)	2129289470
(3)	HOST	IPv4	(1)	2129033471
(4)	HOST	IPv4	(2)	2129033470
(5)	HOST	IPv6	(1)	2128777471
(6)	HOST	IPv6	(2)	2128777470
(7)	HOST	IPv4	(1)	2128521471
(8)	HOST	IPv4	(2)	2128521470
(9)	HOST	IPv6	(1)	2127753471
(10)	HOST	IPv6	(2)	2127753470
(11)	SRFLX	IPv6	(1)	1693081855
(12)	SRFLX	IPv6	(2)	1693081854
(13)	SRFLX	IPv4	(1)	1692825855
(14)	SRFLX	IPv4	(2)	1692825854
(15)	HOST	IPv6	(1)	1692057855
(16)	HOST	IPv6	(2)	1692057854
(17)	RELAY	IPv6	(1)	15360255
(18)	RELAY	IPv6	(2)	15360254
(19)	RELAY	IPv4	(1)	15104255
(20)	RELAY	IPv4	(2)	15104254

SRFLX = server reflexive

Note that the list does not alter the component ID part of the formula. This keeps the different components (RTP and RTCP) close in the list. What matters is the ordering of the candidates with component ID 1. Once the checklist is formed for a media stream the candidate pair with component ID 1 will be tested first. If ICE connectivity check is successful then other candidate pairs with the same foundation will be unfrozen ([RFC5245] section 5.7.4. Computing States).

The local and remote agent can have different algorithms for choosing the local preference and type preference values without impacting the synchronization between the local and remote check lists.

The check list is made up by candidate pairs. A candidate pair is two candidates paired up and given a candidate pair priority as described in [RFC5245] section 5.7.2. Using the pair priority formula:

$$\text{pair priority} = 2^{32} * \text{MIN}(G, D) + 2 * \text{MAX}(G, D) + (G > D ? 1 : 0)$$

Where G is the candidate priority provided by the controlling agent and D the candidate priority provided by the controlled agent. This ensures that the local and remote check lists are coordinated.

Even if the two agents have different algorithms for choosing the candidate priority value to get an intermingled set of IPv4 and IPv6 candidates, the resulting checklist, that is a list sorted by the pair priority value, will be identical on the two agents.

The agent that has promoted IPv4 cautiously i.e. lower IPv4 candidate priority values compared to the other agent, will influence the check list the most due to $(2^{32} * \text{MIN}(G, D))$ in the formula.

These recommendations are backward compatible with a standard ICE implementation. The resulting local and remote checklist will still be synchronized. The introduced fairness might be better, but not worse than what exists today

5. IANA Considerations

None.

6. Security Considerations

STUN connectivity check using MAC computed during key exchanged in the signaling channel provides message integrity and data origin authentication as described in section 2.5 of [RFC5245] apply to this use.

7. Acknowledgements

Authors would like to thank Dan Wing, Ari Keranen, Bernard Aboba, Martin Thomson, Jonathan Lennox and Balint Menyhart for their comments and review.

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

Appendix A. Example Algorithm for Choosing the Local Preference

The value space for the local preference is from 0 to 65535 inclusive. This value space can be divided up in chunks for each IP address family.

An IPv6 and IPv4 start priority must be given. In this example IPv6 starts at 60000 and IPv4 at 59000. This leaves enough address space to further play with the values if different interface priorities needs to be added. The highest value should be given to the address family that should be prioritized.

	IPv6	IPv4						
	Start	Start						
65535	60k	59k	58k	57k	56k	55k		0
+-----+-----+-----+-----+-----+-----+-----+-----+								
	IPv6	IPv4	IPv6	IPv4	IPv6			
	(1)	(1)	(2)	(2)	(3)			
+-----+-----+-----+-----+-----+-----+-----+-----+								
	<- N->							

The local preference can be calculated by the given formula:

$$\text{local_preference} = S - N * 2 * (C_n / C_{\text{max}})$$

S: Address Type specific start value (IPv4 or IPv6 Start)

N: Absolute value of IPv6_start-IPv4_start. This ensures a positive number even if IPv4 is the highest priority.

Cn: Number of current candidates of a specific IP address type and candidate type (host, server reflexive or relay).

Cmax: Number of allowed consecutive candidates of the same IP address type.

Using the values $N = \text{abs}(60000 - 59000)$ and $C_{\text{max}} = 2$ yields the following sorted local candidate list:

```
(1)  HOST  IPv6 (1) Priority: 2129289471
(2)  HOST  IPv6 (2) Priority: 2129289470
(3)  HOST  IPv4 (1) Priority: 2129033471
(4)  HOST  IPv4 (2) Priority: 2129033470
(5)  HOST  IPv6 (1) Priority: 2128777471
(6)  HOST  IPv6 (2) Priority: 2128777470
(7)  HOST  IPv4 (1) Priority: 2128521471
(8)  HOST  IPv4 (2) Priority: 2128521470
(9)  HOST  IPv6 (1) Priority: 2128265471
(10) HOST  IPv6 (2) Priority: 2128265470
(11) SRFLX IPv6 (1) Priority: 1693081855
(12) SRFLX IPv6 (2) Priority: 1693081854
(13) SRFLX IPv4 (1) Priority: 1692825855
(14) SRFLX IPv4 (2) Priority: 1692825854
(15) RELAY IPv6 (1) Priority: 15360255
(16) RELAY IPv6 (2) Priority: 15360254
(17) RELAY IPv4 (1) Priority: 15104255
(18) RELAY IPv4 (2) Priority: 15104254
```

The result is an even spread of IPv6 and IPv4 candidates among the different candidate types (host, server reflexive, relay). The local preference value is calculated separately for each candidate type.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tiredy@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

Email: praspati@cisco.com

Paal-Erik Martinsen
Cisco Systems, Inc.
Philip Pedersens Vei 22
Lysaker, Akershus 1325
Norway

Email: palmarti@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2013

E. Rescorla
RTFM, Inc.
J. Uberti
Google
E. Iovov
Jitsi
October 23, 2012

Trickle ICE: Incremental Provisioning of Candidates for the Interactive
Connectivity Establishment (ICE) Protocol
draft-rescorla-mmusic-ice-trickle-01

Abstract

This document describes an extension to the Interactive Connectivity Establishment (ICE) protocol that allows ICE agents to send and receive candidates incrementally rather than exchanging complete lists. With such incremental provisioning, ICE agents can begin connectivity checks while they are still gathering candidates and considerably shorten the time necessary for ICE processing to complete.

The above mechanism is also referred to as "trickle ICE".

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

1. Introduction	3
2. Terminology	4
3. Incompatibility with Standard ICE	4
4. Detecting Support for Trickle ICE	5
5. Relationship to the Offer/Answer Model	6
6. Sending the Initial Description	7
7. Receiving the Initial Description	8
7.1. Responding to an Initial ICE Description	8
7.2. Forming check lists and beginning connectivity checks	9
8. Receiving a Response to the Initial ICE Description	9
9. Performing Connectivity Checks	9
9.1. Check List and Timer State Updates	9
10. Learning and Sending Additional Local Candidates	10
10.1. Announcing End of Candidates	12
11. Receiving Additional Remote Candidates	12
12. Concluding ICE Processing with Trickle ICE	12
13. Interaction with non-Trickle ICE implementations	13
14. Example Flow	13
15. Security Considerations	13
16. Acknowledgements	13
17. References	14
17.1. Normative References	14
17.2. Informative References	14
Appendix A. Open issues	15
Appendix B. Changes From Earlier Versions	15
B.1. Changes From Individual Submission Draft -00	15
Authors' Addresses	15

1. Introduction

The Interactive Connectivity Establishment (ICE) protocol [RFC5245] describes mechanisms for gathering, candidates, prioritizing them, choosing default ones, exchanging them with the remote party, pairing them and ordering them into check lists. Once all of the above have been completed, and only then, the participating agents can begin a phase of connectivity checks and eventually select the pair of candidates that will be used in the following session.

While the above sequence has the advantage of being relatively straightforward to implement and debug once deployed, it may also prove to be rather lengthy. Gathering candidates or candidate harvesting would often involve things like querying STUN [RFC5389] servers, discovering UPnP devices, and allocating relayed candidates at TURN [RFC5766] servers. All of these can be delayed for a noticeable amount of time and while they can be run in parallel, they still need to respect the pacing requirements from [RFC5245], which is likely to delay them even further. Some or all of the above would also have to be completed by the remote agent. Both agents would next perform connectivity checks and only then would they be ready to begin streaming media.

All of the above could lead to relatively lengthy session establishment times and degraded user experience.

The purpose of this document is to define an alternative mode of operation for ICE implementations, also known as "trickle ICE", where candidates can be exchanged incrementally. This would allow ICE agents to exchange host candidates as soon as a session has been initiated. Connectivity checks for a media stream would also start as soon as the first candidates for that stream have become available.

Trickle ICE allows reducing session establishment times in cases where connectivity is confirmed for the first exchanged candidates (e.g. where the host candidates for one of the agents are directly reachable from the second agent). Even when this is not the case, running candidate harvesting for both agents and connectivity checks all in parallel allows to considerably reduce ICE processing times.

It is worth pointing out that before being introduced to the IETF, trickle ICE had already been included in specifications such as XMPP Jingle [XEP-0176] and it has been in use in various implementations and deployments.

In addition to the basics of trickle ICE, this document also describes how support for trickle ICE needs to be discovered, how

regular ICE processing needs to be modified when building and updating check lists, and how trickle ICE implementations should interoperate with agents that only implement [RFC5245] processing.

This specification does not define usage of trickle ICE with any specific signalling or media description protocol, contrary to [RFC5245] which defined a usage for ICE with SIP and SDP. Such usages would have to be specified in separate documents.

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].

This specification makes use of all terminology defined by the protocol for Interactive Connectivity Establishment in [RFC5245].

Vanilla ICE: The Interactive Connectivity Establishment protocol as defined in [RFC5245].

Candidate Harvester: A module used by an ICE agent to obtain local candidates. Candidate harvesters use different mechanisms for discovering local candidates. Some of them would typically make use of protocols such as STUN or TURN. Others may also employ techniques that are not referenced within [RFC5245]. UPnP based port allocation and XMPP Jingle Relay Nodes [XEP-0278] are among the possible examples.

ICE Description: A set of parameters necessary for ICE initiation. Such parameters include ice-ufrag, ice-pwd and other attributes that agents need to exchange in the beginning of an ICE session. ICE descriptions are transmitted by signalling. In the case of protocols using Offer/Answer semantics, they are always part of the offer and often also part of the answer.

3. Incompatibility with Standard ICE

The ICE protocol was designed to be fairly flexible so that it would work in and adapt to as many network environments as possible. It is hence important to point out at least some of the reasons why, despite its flexibility, the specification in [RFC5245] would not support trickle-ICE.

[RFC5245] describes the conditions required to update check lists and timer states while an ICE agent is in the Running state. These conditions are verified upon transaction completion and one of them stipulates that:

If there is not a pair in the valid list for each component of the media stream, the state of the check list is set to Failed.

This could be a problem and cause ICE processing to fail prematurely in a number of scenarios. Consider the following case:

- o Alice and Bob are both located in different networks with Network Address Translation (NAT). Alice and Bob themselves have different address but both networks use the same [RFC1918] block.
- o Alice sends Bob the candidate 10.0.0.10 which also happens to correspond to an existing host on Bob's network.
- o Bob creates a check list consisting solely of 10.0.0.10 and starts checks.
- o These checks reach the host at 10.0.0.10 in Bob's network, which responds with an ICMP "port unreachable" error and per [RFC5245] Bob marks the transaction as Failed.

At this point the check list only contains Failed candidates and the valid list is empty. This causes the media stream and potentially all ICE processing to Fail.

A similar race condition would occur if the initial offer from Alice only contains candidates that can be determined as unreachable (per [I-D.keranen-mmusic-ice-address-selection]) from any of the candidates that Bob has gathered. This would be the case if Bob's candidates only contain IPv4 addresses and the first candidate that he receives from Alice is an IPv6 one.

Another potential problem could arise when a non-trickle ICE implementation sends an offer to a trickle one. Consider the following case:

- o Alice's client has a non-trickle ICE implementation
- o Bob's client has support for trickle ICE.
- o Alice and Bob are behind NATs with address-dependent filtering [RFC4787].
- o Bob has two STUN servers but one of them is currently unreachable. After Bob's agent receives Alice's offer it would immediately start connectivity checks. It would also start gathering candidates, which would take long because of the unreachable STUN server. By the time Bob's answer is ready and sent to Alice, Bob's connectivity checks may well have failed: until Alice gets Bob's answer, she won't be able to start connectivity checks and punch holes in her NAT. The NAT would hence be filtering Bob's checks as originating from an unknown endpoint.

4. Detecting Support for Trickle ICE

In order to avoid interoperability problems such as those described in Section 3, it is important that trickle ICE sessions are only attempted in cases where both parties support this specification.

This means that usages of trickle for specific protocols MUST provide one of the following:

- o A way for agents to verify support of trickle ICE prior to initiation a session.
- o A procedure for attempting a trickle ICE session in a way that, when received by non-supporting agents, would either allow for a smooth fallback to vanilla ICE, or fail in a predictable way so that a vanilla ICE session can be retried subsequently.

The exact mechanisms that would allow for the verifications above are outside the scope of this document and should be handled by the signalling protocol that is employing ICE.

Examples of how some signalling protocols already handle service and capabilities discovery include:

- o Service discovery [XEP-0030] and Entity capabilities [XEP-0115] for XMPP
- o Indicating User Agent Capabilities [RFC3840] for SIP

Usages of trickle ICE SHOULD make use of these mechanisms where they exist and can provide reliable indication.

In some cases, agents may choose to just send an offer that the remote party would reject as invalid unless it supports trickling. One such example would be an offer with no ICE candidates and an invalid default address (e.g. 0.0.0.0).

Usages of trickle ICE MUST define a way for ICE descriptions to indicate support for trickling as well as a clear procedure for falling back to vanilla ICE in the absence of such support.

5. Relationship to the Offer/Answer Model

The vanilla ICE specification uses the Offer/Answer model for exchanging all ICE parameters. Using just a couple of signalling messages is obviously no longer possible with continuous candidate provisioning and trying to fit candidate exchanges into consecutive offer/answer pairs is clearly not practical. This specification therefore loosens the relationship with the Offer/Answer model by splitting trickle ICE signalling into two phases: initial ICE Descriptions and subsequent exchange of additional candidates.

ICE descriptions contain session or media-level parameters that are necessary for ICE processing to begin. Those include attributes such as ice-ufrag and ice-pwd. Due to their nature ICE descriptions are exchanged in the beginning of a session and trickle ICE agents MUST

NOT send any candidates prior to a description. It is however possible for ICE descriptions to be accompanied by a first set of candidates.

When using trickle ICE with Offer/Answer protocols agents MUST include an initial ICE description in their Offers. Answerers in this situation MAY send their ICE description at any point after receiving that of the offerer but no later than sending their answer, which MUST contain an ICE description if the agent did not provide one before.

After sending an ICE description each agent can continue trickling candidates regardless of what the state of the Offer/Answer negotiation is.

6. Sending the Initial Description

An agent starts gathering candidates as soon as it has an indication that communication is imminent (e.g. a user interface cue or an explicit request to initiate a session). Contrary to vanilla ICE, implementations of trickle ICE do not need to gather candidates in a blocking manner, and SHOULD generate and transmit their initial ICE description as early as possible.

In the case of protocols using the Offer/Answer model, agents MUST include the initial ICE description in the corresponding offer.

Trickle ICE agents MAY include any set of candidates in an ICE description. This includes the possibility of generating a description with no candidates, or one that contains all the candidates that the agent is planning on using in the following session.

For optimal performance, it is RECOMMENDED that an ICE description contains host candidates only. This would allow both agents to start gathering server reflexive, relayed and other non-host candidates simultaneously, and it would also enable them to begin connectivity checks.

If the privacy implications of revealing host addresses are a concern, agents MAY generate an initial ICE description that contains no candidates and then only trickle candidates that do not reveal host addresses (e.g. relayed candidates).

Prior to actually sending an initial ICE description, agents MAY verify if the remote party supports trickle ICE. If absence of such support is confirmed agents SHOULD fall back to using vanilla ICE or

abandon the entire session.

All trickle ICE descriptions MUST indicate support of this specification. The exact syntax of providing this indication is left to the usages that define how signalling protocols employ trickle ICE.

Calculating priorities and foundations, as well as determining redundancy of candidates work the same way they do with vanilla ICE.

7. Receiving the Initial Description

When an agent receives an initial ICE description, in the case of protocols using Offer/Answer this description will be part of the offer, it will check if it indicates support for trickle ICE as explained in Section 4. If this is not the case, the agent MUST process the description according to the [RFC5245] procedures or standard [RFC3264] processing in case no ICE support is detected at all.

If, the description does indicate support for trickle ICE, the agent will determine its role, start gathering and prioritizing candidates and, while doing so it will also respond by sending its own ICE description, so that both agents can start forming check lists and begin connectivity checks.

Otherwise the agent would simply fallback to vanilla ICE processing.

7.1. Responding to an Initial ICE Description

An agent can respond to an initial ICE description at any point while gathering candidates. Just as with initial ICE descriptions (Section 6), the agent does send the description without any candidates or with all those it is planning on using. Again, as with initial descriptions it is RECOMMENDED that responses to initial ICE descriptions contain host candidates so that the remote party can also start forming checklists and performing connectivity checks.

The answer MUST indicate support for trickle ICE as described by usage specifications.

For protocols using Offer/Answer semantics the response to the initial ICE description would either be transmitted prior to the [RFC3264] answer or as a part of it.

7.2. Forming check lists and beginning connectivity checks

After exchanging descriptions, and as soon as they have gathered any candidates, agents will begin forming candidate pairs, computing their priorities and creating check lists according to the vanilla ICE procedures described in [RFC5245]. Obviously in order for candidate pairing to be possible, it would be necessary that both descriptions contained candidates. If this was not the case agents will still create the check lists (so that their Active/Frozen state could be monitored and updated) but they will only populate them once they have learned any local and remote candidates.

Initially, all check lists will have their Active/Frozen state set to Frozen.

Trickle ICE agents will then also attempt to unfreeze the check list for the first media stream (i.e. the first media stream that was reported to the ICE implementation from the using application). If this checklist is still empty however, agents will continue examining media streams in the order they were reported and will unfreeze the first non-empty checklist.

Respecting the order in which lists have been reported to an ICE implementation, or in other words, the order in which streams had been described by the signalling protocol (e.g. SDP), is helpful so that checks for the same media stream is more likely to be performed simultaneously by both agents.

8. Receiving a Response to the Initial ICE Description

When receiving an answer, agents will follow vanilla ICE procedures to determine their role and they would then form check lists and begin connectivity checks as described in Section 7.2.

9. Performing Connectivity Checks

For the most part, trickle ICE agents perform connectivity checks following vanilla ICE procedures. Of course, the asynchronous nature of candidate harvesting in trickle ICE would impose a number of changes:

9.1. Check List and Timer State Updates

The vanilla ICE specification requires that agents update check lists and timer states upon completing a connectivity check transaction. During such an update vanilla ICE agents would set the state of a

check list to Failed if the following two conditions are satisfied:

- o all of the pairs in the check list are either in the Failed or Succeeded state;
- o if at least one of the components of the media stream has no pairs in its valid list.

With trickle ICE, the above situation would often occur when candidate harvesting and trickling are still in progress and it is perfectly possible that future checks will succeed. For this reason trickle ICE agents add the following conditions to the above list:

- o all candidate harvesters have completed and the agent is not expecting to learn any new candidates;
- o the remote agent has sent an end-of-candidates message for that check list as described in Section 10.1.

Vanilla ICE requires that agents then update all other check lists, placing one pair in each of them into the Waiting state, effectively unfreezing the check list. Given that with trickle ICE, other check lists may still be empty at that point, a trickle ICE agent SHOULD also maintain an explicit Active/Frozen state for every check list, rather than deducing it from the state of the pairs it contains. This state should be set to Active when unfreezing the first pair in a list or when that couldn't happen because a list was empty.

10. Learning and Sending Additional Local Candidates

After an ICE description has been sent or received, agents will most likely continue discovering new local candidates as STUN, TURN and other non-host candidate harvesting mechanisms begin to yield results. Whenever such a new candidate is learned agents will compute its priority, type, foundation and component id according to normal vanilla ICE procedures.

The new candidate is then checked for redundancy against the existing list of local candidates. If its transport address and base match those of an existing candidate, it will be considered redundant and will be ignored. This would often happen for server reflexive candidates that match the host addresses they were obtained from (e.g. when the latter are public IPv4 addresses). Contrary to vanilla ICE, trickle ICE agents will consider the new candidate redundant regardless of its priority. [TODO: is this OK? if not we need to check if the existing candidate was already used in conn checks, cancel them, and then restart them with the new candidate ... and in this specific case there's probably no point to do that].

Then, if no remote candidates are currently known for this same

stream, the new candidate will simply be added to the list of local candidates.

Otherwise, if the agent has already learned of one or more remote candidates for this stream and component, it will begin pairing the new local candidates with them and adding the pairs to the existing check lists according to their priority. Forming candidate pairs will work the way it is described by the vanilla ICE specification. Actually adding the new pair to a check list however, will happen according to the rules described below.

If the new pair's local candidate is server reflexive, the server reflexive candidate **MUST** be replaced by its base before adding the pair to the list. Once this is done, the agent examines the check list looking for another pair that would be redundant with the new one. If such a pair exists and its state is:

Succeeded: the newly formed pair is ignored.

Frozen or Waiting: the agent chooses the pair with the higher priority local candidate, places it in the state that the old pair was in (i.e. Frozen or Waiting) and removes the other one as redundant.

Failed: the agent chooses the pair with the higher priority local candidate, places it in the Waiting state and removes the other one as redundant.

In-Progress: The agent cancels the in-progress transaction (where cancellation happens as explained in Section 7.2.1.4 of [RFC5245]), then it chooses the pair with the higher priority local candidate, places it in the Waiting state and removes the other one as redundant.

For all other pairs, including those with a server reflexive local candidate that were not found to be redundant:

- o if this check list is Frozen then the new pair will also be assigned a Frozen state.
- o else if the check list is Active and it is either empty or contains only candidates in the Succeeded and Failed states, then the new pair's state is set to Waiting.
- o else if the check list is non-empty and Active, then the new pair state will be set to

Frozen: if there is at least one pair in the list whose foundation matches the one in the new pair and whose state is neither Succeeded nor Failed (eventually the new pair will get unfrozen after the the on-going check for the existing pair concludes);

Waiting: if the list contains no pairs with the same foundation as the new one, or, in case such pairs exist, they are all in either the Succeeded or Failed states.

10.1. Announcing End of Candidates

Once all candidate harvesters for a specific media stream complete, or expire, the agent MUST generate an "end-of-candidates" event for that stream and send it to the remote agent via the signalling channel. This would allow the remote agent to begin updating check list states and, in case valid pairs do not exist for every component in every media stream, determine that ICE processing has failed.

An agent MAY also choose to generate an "end-of-candidates" event before candidate harvesting has actually completed, if the agent determines that harvesting has continued for more than an acceptable period of time.

Once the agent sends the end-of-candidates event, it SHOULD update the state of the corresponding check list as explained in section Section 9.1

[TODO: should we also have an end-of-candidates for the entire harvesting process (as opposed to that of a single stream)]

11. Receiving Additional Remote Candidates

At any point of ICE processing, a trickle ICE agent may receive new candidates from the remote agent. When this happens and no local candidates are currently known for this same stream, the new remote candidates are simply added to the list of remote candidates.

Otherwise, the new candidates are used for forming candidate pairs with the pool of local candidates.

Once the remote agent has completed candidate harvesting, it will send an "end-of-candidates" event. Upon receiving such an event, the local agent MUST update check list states as per Section 9.1. This may lead to some check lists being marked as Failed.

12. Concluding ICE Processing with Trickle ICE

Trickle ICE processing SHOULD be concluded as explained in Section 8 of [RFC5245].

13. Interaction with non-Trickle ICE implementations

Trickle ICE implementations MUST behave as non-trickle and follow [RFC5245] unless they can confirm that the remote party supports this specification. [TODO: anything else?]

14. Example Flow

A typical successful trickle ICE exchange with an Offer/Answer protocol would look this way:

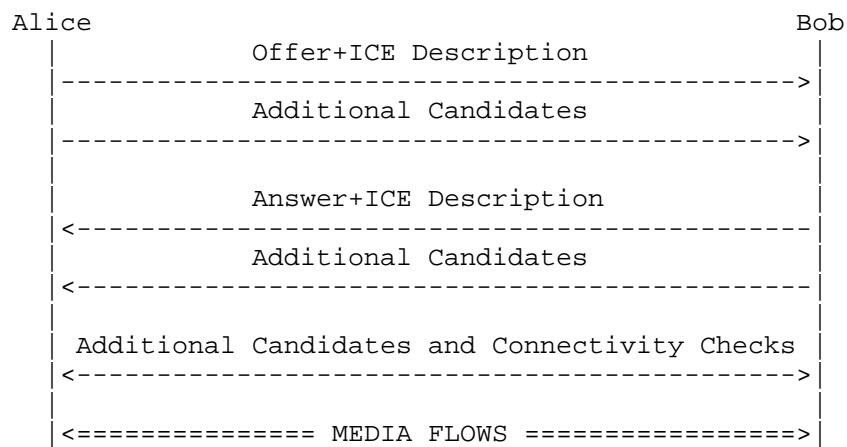


Figure 1: Example

15. Security Considerations

[TODO]

16. Acknowledgements

The authors would like to thank Christer Holmberg and Martin Thomson for their reviews and suggestions on improving this document.

17. References

17.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

17.2. Informative References

- [I-D.keranen-mmusic-ice-address-selection]
Keranen, A. and J. Arkko, "Update on Candidate Address Selection for Interactive Connectivity Establishment (ICE)", draft-keranen-mmusic-ice-address-selection-01 (work in progress), July 2012.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.
- [XEP-0030]
Hildebrand, J., Millard, P., Eatmon, R., and P. Saint-Andre, "XEP-0030: Service Discovery", XEP XEP-0030, June 2008.
- [XEP-0115]

Hildebrand, J., Saint-Andre, P., Troncon, R., and J. Konieczny, "XEP-0115: Entity Capabilities", XEP XEP-0115, February 2008.

[XEP-0176]

Beda, J., Ludwig, S., Saint-Andre, P., Hildebrand, J., Egan, S., and R. McQueen, "XEP-0176: Jingle ICE-UDP Transport Method", XEP XEP-0176, June 2009.

[XEP-0278]

Camargo, T., "XEP-0278: Jingle Relay Nodes", XEP XEP-0278, June 2011.

Appendix A. Open issues

At the time of writing of this document the authors have no clear view on how and if the following list of issues should be address here:

1. Do we need a "stop sending me candidates" message. What would be the use case for that.
2. Is there anything specific we need to say about ICE lite?

Appendix B. Changes From Earlier Versions

Note to the RFC-Editor: please remove this section prior to publication as an RFC.

B.1. Changes From Individual Submission Draft -00

- o Relaxed requirements about verifying support following a discussion on MMUSIC.
- o Introduced ICE descriptions in order to remove ambiguous use of 3264 language and inappropriate references to offers and answers.
- o Removed inappropriate assumption of adoption by RTCWEB pointed out by Martin Thomson.

Authors' Addresses

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Phone: +1 650 678 2350
Email: ekr@rtfm.com

Justin Uberti
Google
747 6th St S
Kirkland, WA 98033
USA

Phone: +1 857 288 8888
Email: justin@uberti.name

Emil Ivov
Jitsi
Strasbourg 67000
France

Phone: +33 6 72 81 15 55
Email: emcho@jitsi.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 30, 2013

C. Holmberg
M. Westerlund
B. Burman
F. Jansson
Ericsson
September 26, 2012

Multiple Synchronization Sources (SSRC) in SDP Media Descriptions
draft-westerlund-mmusic-max-ssrc-00

Abstract

This document describes use-cases where endpoints, for a given media type, use multiple media sources. It also describes how endpoints normally support media sources, and limitations associated with the support of multiple sources.

This document also defines two new SDP attributes, "max-send-ssrc" and "max-recv-ssrc". The attributes allows an entity to, for a given media description, indicate sending and receiving capabilities of multiple media sources, based on codec usage.

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 March 30, 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

1. Introduction	3
2. Definitions	3
2.1. Requirements Language	3
2.2. Terminology	3
3. Multiple Media Stream Support	4
3.1. General	4
3.2. Multiple Source Support Limitations	5
4. SDP max-send-ssrc And max-recv-ssrc Attributes	5
4.1. Introduction	5
4.2. Usage	5
4.3. Syntax	6
4.4. Use in Offer/Answer	6
5. Examples	7
6. IANA Considerations	7
7. Security Considerations	8
8. References	8
8.1. Normative References	8
8.2. Informative References	8
Authors' Addresses	9

1. Introduction

An RTP session [RFC3550] contains a Synchronization Sources (SSRC) space. This space can encompass a number of endpoints and network entities, and associated media streams, within the RTP session. As defined in RFC 3550, within an RTP session, each entity may use zero, one or multiple SSRCs to indicate a physical media source (e.g. a camera or a microphone), a conceptual source (e.g. the most active speaker, selected by an RTP mixer, within a conference), or to identify a receiver that provides feedback and reports on reception. A given SSRC value is associated with a physical media source. Multiple SSRC values can be associated with the same source.

The Session Description Protocol (SDP) [RFC4566] describes media streams using media descriptions (m- lines). Each m- line is normally associated with a given media type (e.g. audio or video).

Multiple media streams and media sources might be associated with a single SDP media description. Each media stream will share the parameters and characteristics (e.g. payload type values and codecs) that have been indicated in the SDP media description.

This document describes use-cases where endpoints, for a given media type, use multiple media sources. It also describes how endpoints normally support media sources, and limitations associated with the support of multiple sources.

This document also defines two new SDP attributes, "max-send-ssrc" and "max-recv-ssrc". The attributes allows an entity to, for a given media description, indicate sending and receiving capabilities of multiple media sources, based on codec usage .

2. Definitions

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Terminology

The following terms and abbreviations are used in this document:

Encoding: A particular encoding is the choice of the media encoder (codec) that has been used to compress the media, the fidelity of that encoding through the choice of sampling, bit-rate and other configuration parameters.

Different encodings: An encoding is different when some parameter that characterize the encoding of a particular media source has been changed. Such changes can be one or more of the following parameters; codec, codec configuration, bit-rate, sampling.

Media source: The source of a stream of data of one Media Type, It can either be a single media capturing device such as a video camera, a microphone, or a specific output of a media production function, such as an audio mixer, or some video editing function.

Media stream: Within the scope of this document, a media stream represents a media flow, associated with a media source (identified by a SSRC value) and an SDP media description (m-line).

3. Multiple Media Stream Support

3.1. General

Many applications and systems have been designed to ensure that any given endpoint only needs to, for a given SDP media description, send or receive a single media stream, associated with a single source. Some applications might be able to switch between different SSRC values, but will still only be able to process media associated with a single SSRC value at any given time.

Then there is some applications that are designed to use multiple SSRCs simultaneously. Some send media from multiple media sources, for example an application which sends video from multiple cameras. Some RTP extension mechanisms require endpoints to be able to handle multiple SSRCs. An example is the mechanism for SSRC multiplexed RTP retransmissions [RFC4588]. Multicast applications typically support multiple media streams, as they might receive media from multiple remote entities. Some unicast multi-party applications also receive multiple media sources from a central entity relaying sources from multiple origins.

NOTE: Even if an endpoint is not used in scenarios where multiple media streams (SSRCs) are sent and received, according to RFC 3550, the endpoint need to be able to support cases where the SSRC value used for a media stream is changed, e.g. due to an SSRC value collision [RFC3550].

3.2. Multiple Source Support Limitations

The theoretical maximum number of sources, for a given RTP session, is 2^{32} . However, even if applications, for a given RTP session, are able to handle multiple media streams simultaneously, entities only have resources to handle a given number (typically far smaller the theoretical maximum number) of media streams. The number of supported media streams might depend on the type of codecs, or codec configurations, that are used for the media streams. Networks might also put constraints on the number of media streams that can be supported.

In environments where endpoints, for a given SDP media description, have different amount of resources to handle multiple media stream of handling multiple media streams, network entities (e.g. RTP mixers) might be used, in order to select, or combine, media streams into a number of media streams that is supported by the endpoints to which the media is sent. The policies and algorithms to select and combine media streams are outside the scope of this document.

4. SDP max-send-ssrc And max-recv-ssrc Attributes

4.1. Introduction

As different applications and entities typically are able to simultaneously handle a different number of media streams associated with a given SDP media description, it is necessary for an entity to be able to declare how many media streams it is able to simultaneously send and receive, and whether the used codecs and codec configurations have impact on the number of media streams.

This section defines two new media level SDP attributes, "max-send-ssrc" and "max-recv-ssrc". The attributes are used to, for a given SDP media description, indicate the multiple stream capabilities of an entity. The "max-send-ssrc" attribute is used to indicate simultaneous sending capabilities, and the "max-recv-ssrc" attribute is used to indicate simultaneous receiving capabilities.

4.2. Usage

The SDP attributes are used to describe multiple stream capabilities based on which codecs or codec configurations are used for each stream. The attributes allow to describe multiple alternatives.

Each alternative contains one or more codecs or codec configurations (indexed using the payload type value which describes the codec in the SDP), and for each codec the number of simultaneous streams

the endpoint is able to handle.

For a given alternative, payload type values can be explicitly listed. It is also possible to use a payload type range, which includes all payload type values within the range. Alternatively it is possible to use a wildcard value, which indicates that the indicated number of SSRCs is independent of which codec is used.

The number of streams that an entity can simultaneously send can be different from the number it can receive.

4.3. Syntax

The syntax for the attributes is in ABNF [RFC5234]:

```
max-ssrc      = "a="( "max-send-ssrc:" / "max-recv-ssrc:" ) alt-list
alt-list      = alt-set *(WSP alt-set)
alt-set       = "{ " alt *("&" alt) "}"
alt           = pt ":" limit
pt            = ( pt-list / pt-wildcard )
pt-list       = ( pt-value / pt-range ) *("," ( pt-value / pt-range ))
pt-value      = 1*3DIGIT
pt-range      = pt-value "-" pt-value
pt-wildcard   = "*"
limit         = 1*8DIGIT
; WSP and DIGIT defined in [RFC5234]
```

4.4. Use in Offer/Answer

An SDP Offerer that supports and uses the mechanism in this document MUST include the SDP attributes in the initial SDP offer of a session. If the SDP Answerer also supports and uses the mechanism, the attributes MUST be included in each subsequent SDP Offer and Answer during the session.

An SDP Answerer MUST NOT include the SDP attributes in the SDP Answer unless the associated SDP Offer also contains them.

For sendrecv SDP media descriptions (m- lines), an endpoint that uses the mechanism described in this document MUST include both the "max-send-ssrc" and "max-recv-ssrc" attributes in an SDP Offer and Answer [RFC3264], also for directions in which the endpoint only supports a single SSRC.

For sendonly or recvonly SDP media descriptions, an endpoint MUST include that attribute which corresponds to the direction attribute. For information purpose, the endpoint MAY include also the other

attribute.

TODO: Guidance text is needed, which describes how the SDP Answerer indicates its capabilities in a way so that they match the capabilities of the SDP Offerer as far as possible.

5. Examples

The SDP examples below are not complete. Only the relevant parts are shown.

```
m=video 49200 RTP/AVP 99
a=rtpmap:99 H264/90000
a=max-send-ssrc:{*:2}
a=max-recv-ssrc:{*:4}
```

The SDP indicates that the endpoint is able to send 2 simultaneous SSRCs, and is able to receive 4 simultaneous SSRCs. The wildcarded payload type value indicates that the indicated capabilities apply for any of the indicated codecs (only a single one in this example).

```
m=video 50324 RTP/AVP 96 97
a=rtpmap:96 H264/90000
a=rtpmap:97 H263-2000/90000
a=max-recv-ssrc:{96:2&97:3} {96:1&97:4} {97:5}
a=max-send-ssrc:{* 1}
```

The SDP indicates 3 different receiving capability alternatives. The first alternative indicates that the endpoint is able to receive at most 2 SSRCs using the H.264 codec (payload type value 96) and 3 SSRCs using the H.263 codec (payload type value 97). The second alternative indicates that the endpoint is able to receive 1 SSRC using the H.264 codec and 4 SSRCs using the H.263 codec. The third alternative indicates that the endpoint is able to receive 5 SSRCs using the H.263 codec. The SDP indicates that the endpoint is only able to send one SSRC, no matter which of the indicated codecs are used.

6. IANA Considerations

This document registers two media level SDP attributes.

TODO: IANA registration template

7. Security Considerations

The "max-recv-ssrc" and "max-send-ssrc" SDP attributes describe capabilities of the endpoint that sends the attributes. Knowledge of the capabilities might be used to trigger different kind of attacks. The primary security concern is when a malicious man-in-the-middle (MiTM) modifies the attribute values so that endpoints have wrong information about the capabilities of the other endpoints. Such modification of the capabilities might cause bad user experience, or prevent services all together. For example, if an endpoint has indicated that it is only able to receive a single media stream, and a MiTM increases that number, the endpoint might end up receiving more media streams than it is able to handle.

In order to prevent a MiTM to modify the SDP attributes, it is RECOMMENDED to use a mechanism that provides authentication and integrity protection of the SDP.

8. References

8.1. 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.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

8.2. Informative References

- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Bo Burman
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 13 11
Email: bo.burman@ericsson.com

Fredrik Jansson
Ericsson
Farogatan 6
Kista, SE-164 80
Sweden

Phone: +46 10 719 00 00
Fax:
Email: fredrik.k.jansson@ericsson.com
URI:

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 23, 2014

C. Holmberg
M. Westerlund
B. Burman
F. Jansson
Ericsson
September 19, 2013

Multiple Synchronization Sources (SSRC) in SDP Media Descriptions
draft-westerlund-mmusic-max-ssrc-02.txt

Abstract

This document describes use-cases where endpoints, for a given media type, use multiple media sources. It also describes how endpoints normally support media sources, and limitations associated with the support of multiple sources.

This document also defines two new SDP attributes, "max-send-ssrc" and "max-recv-ssrc". The attributes allows an entity to, for a given media description, indicate sending and receiving capabilities of multiple media sources, based on codec usage .

Status of This Memo

This Internet-Draft is submitted to IETF 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 March 23, 2014.

Copyright Notice

Copyright (c) 2013 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.

Table of Contents

1. Introduction	2
2. Definitions	3
2.1. Requirements Language	3
2.2. Terminology	3
3. Multiple Media Stream Support	3
3.1. General	3
3.2. Multiple Source Support Limitations	4
4. SDP max-send-ssrc And max-recv-ssrc Attributes	4
4.1. Introduction	4
4.2. Usage	5
4.3. Syntax	5
4.4. Use in Offer/Answer	6
5. Examples	6
6. IANA Considerations	7
7. Security Considerations	7
8. References	8
8.1. Normative References	8
8.2. Informative References	8
Authors' Addresses	8

1. Introduction

An RTP session [RFC3550] contains a Synchronization Sources (SSRC) space. This space can encompass a number of endpoints and network entities, and associated media streams, within the RTP session. As defined in RFC 3550, within an RTP session, each entity may use zero, one or multiple SSRCs to indicate a physical media source (e.g. a camera or a microphone), a conceptual source (e.g. the most active speaker, selected by an RTP mixer, within a conference), or to identify a receiver that provides feedback and reports on reception. A given SSRC value is associated with a physical media source. Multiple SSRC values can be associated with the same source.

The Session Description Protocol (SDP) [RFC4566] describes media streams using media descriptions (m- lines). Each m- line is normally associated with a given media type (e.g. audio or video).

Multiple media streams and media sources might be associated with a single SDP media description. Each media stream will share the parameters and characteristics (e.g. payload type values and codecs) that have been indicated in the SDP media description.

This document describes use-cases where endpoints, for a given media type, use multiple media sources. It also describes how endpoints normally support media sources, and limitations associated with the support of multiple sources.

This document also defines two new SDP attributes, "max-send-ssrc" and "max-recv-ssrc". The attributes allows an entity to, for a given media description, indicate sending and receiving capabilities of multiple media sources, based on codec usage .

2. Definitions

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Terminology

The following terms and abbreviations are used in this document:

Encoding: A particular encoding is the choice of the media encoder (codec) that has been used to compress the media, the fidelity of that encoding through the choice of sampling, bit-rate and other configuration parameters.

Different encodings: An encoding is different when some parameter that characterize the encoding of a particular media source has been changed. Such changes can be one or more of the following parameters; codec, codec configuration, bit-rate, sampling.

Media source: The source of a stream of data of one Media Type, It can either be a single media capturing device such as a video camera, a microphone, or a specific output of a media production function, such as an audio mixer, or some video editing function.

Media stream: Within the scope of this document, a media stream represents a media flow, associated with a media source (identified by a SSRC value) and an SDP media description (m-line).

3. Multiple Media Stream Support

3.1. General

Many applications and systems have been designed to ensure that any given endpoint only needs to, for a given SDP media description, send

or receive a single media stream, associated with a single source. Some applications might be able to switch between different SSRC values, but will still only be able to process media associated with a single SSRC value at any given time.

Then there is some applications that are designed to use multiple SSRCs simultaneously. Some send media from multiple media sources, for example an application which sends video from multiple cameras. Some RTP extension mechanisms require endpoints to be able to handle multiple SSRCs. An example is the mechanism for SSRC multiplexed RTP retransmissions [RFC4588]. Multicast applications typically support multiple media streams, as they might receive media from multiple remote entities. Some unicast multi-party applications also receive multiple media sources from a central entity relaying sources from multiple origins.

NOTE: Even if an endpoint is not used in scenarios where multiple media streams (SSRCs) are sent and received, according to RFC 3550, the endpoint need to be able to support cases where the SSRC value used for a media stream is changed, e.g. due to an SSRC value collision [RFC3550].

3.2. Multiple Source Support Limitations

The theoretical maximum number of sources, for a given RTP session, is 2^{32} . However, even if applications, for a given RTP session, are able to handle multiple media streams simultaneously, entities only have resources to handle a given number (typically far smaller the theoretical maximum number) of media streams. The number of supported media streams might depend on the type of codecs, or codec configurations, that are used for the media streams. Networks might also put constraints on the number of media streams that can be supported.

In environments where endpoints, for a given SDP media description, have different amount of resources to handle multiple media stream of handling multiple media streams, network entities (e.g. RTP mixers) might be used, in order to select, or combine, media streams into a number of media streams that is supported by the endpoints to which the media is sent. The policies and algorithms to select and combine media streams are outside the scope of this document.

4. SDP max-send-ssrc And max-recv-ssrc Attributes

4.1. Introduction

As different applications end entities typically are able to simultaneously handle a different number of media streams associated

with a given SDP media description, it is necessary for an entity to be able to declare how many media streams it is able to simultaneously send and receive, and whether the used codecs and codec configurations have impact on the number of media streams.

This section defines two new media level SDP attributes, "max-send-ssrc" and "max-recv-ssrc". The attributes are used to, for a given SDP media description, indicate the multiple stream capabilities of an entity. The "max-send-ssrc" attribute is used to indicate simultaneous sending capabilities, and the "max-recv-ssrc" attribute is used to indicate simultaneous receiving capabilities.

4.2. Usage

The SDP attributes are used to describe multiple stream capabilities based on which codecs or codec configurations are used for each stream. The attributes allow to describe multiple alternatives.

Each alternative contains one or more codecs or codec configurations (indexed using the payload type value which describes the codec in the SDP), and for each codec the number of simultaneous streams the endpoint is able to handle.

For a given alternative, payload type values can be explicitly listed. It is also possible to use a payload type range, which includes all payload type values within the range. Alternatively it is possible to use a wildcard value, which indicates that the indicated number of SSRCs is independent of which codec is used.

The number of streams that an entity can simultaneously send can be different from the number it can receive.

4.3. Syntax

The syntax for the attributes is in ABNF [RFC5234]:

```
max-ssrc      = "a="( "max-send-ssrc:" / "max-recv-ssrc:" ) alt-list
alt-list      = alt-set *(WSP alt-set)
alt-set       = "{" alt *("&" alt)) "}"
alt           = pt ":" limit
pt            = ( pt-list / pt-wildcard )
pt-list       = ( pt-value / pt-range ) *(","( pt-value / pt-range ))
pt-value      = 1*3DIGIT
pt-range      = pt-value "-" pt-value
pt-wildcard   = "*"
limit         = 1*8DIGIT
; WSP and DIGIT defined in [RFC5234]
```

4.4. Use in Offer/Answer

An SDP Offerer that supports and uses the mechanism in this document MUST include the SDP attributes in the initial SDP offer of a session. If the SDP Answerer also supports and uses the mechanism, the attributes MUST be included in each subsequent SDP Offer and Answer during the session.

An SDP Answerer MUST NOT include the SDP attributes in the SDP Answer unless the associated SDP Offer also contains them.

For sendrecv SDP media descriptions (m- lines), an endpoint that uses the mechanism described in this document MUST include both the "max-send-ssrc" and "max-recv-ssrc" attributes in an SDP Offer and Answer [RFC3264], also for directions in which the endpoint only supports a single SSRC.

For sendonly or recvonly SDP media descriptions, an endpoint MUST include that attribute which corresponds to the direction attribute. For information purpose, the endpoint MAY include also the other attribute.

TODO: Guidance text is needed, which describes how the SDP Answerer indicates its capabilities in a way so that they match the capabilities of the SDP Offerer as far as possible.

5. Examples

The SDP examples below are not complete. Only the relevant parts are shown.

```
m=video 49200 RTP/AVP 99
a=rtpmap:99 H264/90000
a=max-send-ssrc:{*:2}
```

```
a=max-recv-ssrc:{*:4}
```

The SDP indicates that the endpoint is able to send 2 simultaneous SSRCS, and is able to receive 4 simultaneous SSRCS. The wildcarded payload type value indicates that the indicated capabilities apply for any of the indicated codecs (only a single one in this example).

```
m=video 50324 RTP/AVP 96 97
a=rtpmap:96 H264/90000
a=rtpmap:97 H263-2000/90000
a=max-recv-ssrc:{96:2&97:3} {96:1&97:4} {97:5}
a=max-send-ssrc:{* 1}
```

The SDP indicates 3 different receiving capability alternatives. The first alternative indicates that the endpoint is able to receive at most 2 SSRCS using the H.264 codec (payload type value 96) and 3 SSRCS using the H.263 codec (payload type value 97). The second alternative indicates that the endpoint is able to receive 1 SSRCS using the H.264 codec and 4 SSRCS using the H.263 codec. The third alternative indicates that the endpoint is able to receive 5 SSRCS using the H.263 codec. The SDP indicates that the endpoint is only able to send one SSRCS, no matter which of the indicated codecs are used.

6. IANA Considerations

This document registers two media level SDP attributes.

TODO: IANA registration template

7. Security Considerations

The "max-recv-ssrc" and "max-send-ssrc" SDP attributes describe capabilities of the endpoint that sends the attributes. Knowledge of the capabilities might be used to trigger different kind of attacks. The primary security concern is when a malicious man-in-the-middle (MiTM) modifies the attribute values so that endpoints have wrong information about the capabilities of the other endpoints. Such modification of the capabilities might cause bad user experience, or prevent services all together. For example, if an endpoint has indicated that it is only able to receive a single media stream, and a MiTM increases that number, the endpoint might end up receiving more media streams than it is able to handle.

In order to prevent a MiTM to modify the SDP attributes, it is RECOMMENDED to use a mechanism that provides authentication and integrity protection of the SDP.

8. References

8.1. 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.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

8.2. Informative References

- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Bo Burman
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 13 11
Email: bo.burman@ericsson.com

Fredrik Jansson
Ericsson
Farogatan 6
Kista SE-164 80
Sweden

Phone: +46 10 719 00 00
Email: fredrik.k.jansson@ericsson.com

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: April 9, 2013

D. Wing
P. Patil
T. Reddy
P. Martinsen
Cisco
October 6, 2012

Mobility with ICE (MICE)
draft-wing-mmusic-ice-mobility-02

Abstract

This specification describes how endpoint mobility can be achieved using ICE. Two mechanisms are shown, one where both endpoints support ICE and another where only one endpoint supports ICE.

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 9, 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

1. Introduction	3
2. Notational Conventions	4
3. Break Before Make	4
3.1. Absence of other interfaces in Valid list	5
3.1.1. Receiving ICE Mobility event	6
3.2. Presence of other interfaces in Valid list	7
3.2.1. Receiving ICE Mobility event	8
3.3. Losing an Interface	8
3.3.1. Keeping unused candidates in the valid list active	9
3.3.2. Keeping unused relayed candidates active	9
3.4. New STUN Attributes	10
4. Make Before Break	10
5. Mobility using TURN	11
5.1. Creating an Allocation	12
5.1.1. Sending an Allocate Request	12
5.1.2. Receiving an Allocate Request	12
5.1.3. Receiving an Allocate Success Response	13
5.1.4. Receiving an Allocate Error Response	13
5.2. Refreshing an Allocation	13
5.2.1. Sending a Refresh Request	13
5.2.2. Receiving a Refresh Request	13
5.2.3. Receiving a Refresh Response	14
5.3. New STUN Attribute MOBILITY-TICKET	14
5.4. New STUN Error Response Code	14
6. IANA Considerations	15
7. Security Considerations	15
7.1. Considerations for ICE mechanism	15
7.2. Considerations for TURN mechanism	15
8. Acknowledgements	15
9. Change History	16
9.1. Changes from draft-wing-mmusic-ice-mobility-00 to -01	16
9.2. Changes from draft-wing-mmusic-ice-mobility-01 to -02	16
10. References	16
10.1. Normative References	16
10.2. Informative References	16
Authors' Addresses	17

1. Introduction

When moving between networks, an endpoint has to change its IP address. This change breaks upper layer protocols such as TCP and RTP. Various techniques exist to prevent this breakage, all tied to making the endpoint's IP address static (e.g., Mobile IP, Proxy Mobile IP, LISP). Other techniques exist, which make the upper layer protocol ambivalent to IP address changes (e.g., SCTP). The mechanisms described in this document are in that last category.

ICE [RFC5245] ensures two endpoints have a working media path between them, and is typically used by Internet-connected interactive media systems (e.g., SIP endpoints). ICE does not expect either the local host or the remote host to change their IP addresses. Although ICE does allow an "ICE restart", this is done by sending a re-INVITE which goes over the SIP signaling path. The SIP signaling path is often slower than the media path (which needs to be recovered as quickly as possible), consumes an extra half round trip, and incurs an additional delay if the mobility event forces the endpoint to re-connect with its SIP proxy. When a device changes its IP address, it is necessary for it to re-establish connectivity with its SIP proxy, which can be performed in parallel with the steps described in this document. This document describes how mobility is performed entirely in the media path, without the additional delay of re-establishing SIP connectivity, issuing a new offer/answer, or the complications of multiple SIP offers. This document considers re-establishing bi-directional media the most critical aspect of a successful mobility event, and its efforts are towards meeting that goal.

A TURN [RFC5766] server relays media packets and is used for a variety of purposes, including overcoming NAT and firewall traversal issues and IP address privacy. The existing TURN specification does not allow the client address to change, especially if multiple clients share the same TURN username (e.g., the same credentials are used on multiple devices).

This document proposes two mechanisms to achieve RTP mobility: a mechanism where both endpoints support ICE, and a mechanism where only one endpoint supports ICE. When both endpoints support ICE, ICE itself can be used to provide mobility. When only one endpoint supports ICE, a TURN server provides mobility. Both mobility techniques work across and between network types (e.g., between 3G and wired Internet access), so long as the client can still access the remote ICE peer or TURN server.

Readers are assumed to be familiar with ICE [RFC5245].

2. Notational Conventions

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].

This note uses terminology defined in [RFC5245], and the following additional terminology:

Break Before Make: The initially selected interface for communication may become unavailable (e.g due to loss of coverage when moving out of a WiFi hotspot) and new interfaces may become available due to administrative action (e.g manual activation of a specific connectivity technology) or due to dynamic conditions (e.g. Entering coverage area of a wireless network).

Make Before Break: The initially selected interface for communication may become deprioritized (e.g new interface becoming available and it's per bit cost is cheaper and the connection speed is faster than existing interface used for communication).

Simultaneous Mobility: If both the endpoints are mobile and roam at the same time between networks.

3. Break Before Make

When both endpoints support ICE, ICE itself can provide mobility functions. One of the primary aspects of ICE is its address gathering, wherein ICE has each endpoint determine all of the IP addresses and ports that might be usable for that endpoint and communicate that list of addresses and ports to its peer, usually over SDP. That enables the next primary aspect of ICE, which is its connectivity checks: each ICE endpoint sends a connectivity check to that list of addresses and ports. A connectivity check may unknowingly traverse a NAT, which means the ICE endpoint receiving the connectivity check cannot validate the source IP address or port of the connectivity against the list of IP addresses and ports provided by the ICE peer. In fact, if the source IP address and port is not known to the ICE endpoint, it is added to the list of candidates (Section 7.2.1.3 of [RFC5245]). ICE Mobility takes advantage of that existent ICE functionality.

Endpoints that support ICE Mobility perform ICE normally, and MUST also include the MOBILITY-SUPPORT attribute in all of their STUN requests and their STUN responses. The inclusion of this attribute allows the ICE peer to determine if it can achieve mobility using ICE or needs to use TURN. To force the use of TURN to achieve ICE

mobility, the ICE endpoint SHOULD NOT respond to ICE connectivity checks that have an IP address and port different from the TURN server, unless those connectivity checks contain the MOBILITY-SUPPORT attribute. In this way, the remote peer will think those other candidates are invalid (because its connectivity checks did not succeed).

After concluding ICE and moving to the ICE completed state (see Section 8 of [RFC5245] either endpoint or both endpoints can initiate ICE Mobility, no matter if it was the Controlling Agent or the Controlled Agent during normal ICE processing.

3.1. Absence of other interfaces in Valid list

When the interface currently being used for communication becomes unavailable then ICE agent acquires a list of interfaces that are available and based on the locally configured host policy preferences, the ICE endpoint performs ICE Mobility using one of the available interfaces. In this case local candidates from the selected interface are not present in the valid list. ICE Mobility is performed by :

1. The ICE agent remembers the remote host/server-reflexive candidates for each component of the media streams previously used from the valid list before clearing its ICE check list and ICE Valid List.
2. The ICE endpoint gathers host candidates on the new interface, forms a check list by creating candidate pairs with local host candidates and remote host/server-reflexive candidates collected in step 1, performs "Computing Pair Priority and Ordering Pairs" (Section 5.7.2 of [RFC5245]), "Pruning the Pairs" (Section 5.7.3 of [RFC5245]), "Computing states" (Section 5.7.4 of [RFC5245]).
3. The ICE endpoint initiates ICE connectivity checks on those candidates from the check list in the previous step, and includes the MOBILITY-EVENT attribute in those connectivity checks.
4. The ICE endpoint acts as controlling agent and the ICE connectivity check from the previous step SHOULD also include the USE-CANDIDATE attribute to signal an aggressive nomination (see Section 2.6 of [RFC5245]). An aggressive nomination allows sending media immediately after the connectivity check completes, without waiting for other connectivity checks to complete.
5. The ICE endpoint performs "Discovering Peer Reflexive Candidates" (Section 7.1.3.2.1 of [RFC5245]), "Constructing a Valid Pair" (Section 7.1.3.2.2 of [RFC5245]), "Updating Pair States" (Section

7.1.3.2.3 of [RFC5245]), and "Updating the Nominated Flag" (Section 7.1.3.2.4 of [RFC5245]). When the valid list contains a candidate pair for each component then ICE processing is considered complete for the media stream and ICE agent can start sending media using highest-priority nominated candidate pair.

6. Once ICE connectivity checks for all of the media streams are completed, the controlling ICE endpoint follows the procedures in Section 11.1 of [RFC5245], specifically to send updated offer if the candidates in the m and c lines for the media stream (called the DEFAULT CANDIDATES) do not match ICE's SELECTED CANDIDATES (also see Appendix B.9 of [RFC5245]).

The ICE endpoint even after Mobility using ICE is successful can issue an updated offer indicating ICE restart if connectivity checks using higher priority candidate pairs are not successful.

Mobility using ICE could fail in case of Simultaneous Mobility or if the ICE peer is behind NAT that performs Address-Dependent Filtering (see Section 5 of [RFC5245]). Hence the ICE endpoint in parallel will re-establish connection with the SIP proxy. It will then determine whether to initiate ICE restart under the following conditions :

1. After re-establishing connection with the SIP proxy and before sending new offer to initiate ICE restart if Mobility using ICE is successful then stop sending the new offer.
2. After successful negotiation of updated offer/answer to initiate ICE restart, proceed with ICE restart and stop Mobility using ICE if ICE checks are in the Running/Failed states or ICE is partially successful and not yet reached ICE complete state. It's not implementation friendly to have to two checks running in parallel. ICE restart can re-use partial successful ICE connectivity check results from Mobility using ICE if required as optimization.

3.1.1.1. Receiving ICE Mobility event

A STUN Binding Request containing the MOBILITY-EVENT attribute MAY be received by an ICE endpoint. The agent MUST use short-term credential to authenticate the STUN request containing the MOBILITY-EVENT attribute and perform a message integrity check. The ICE endpoint will generate STUN Binding Response containing the MOBILE-SUPPORT attribute and the ICE agent takes role of controlled agent. If STUN Request containing the MOBILITY-EVENT attribute is received before the endpoint is in the ICE Completed state, it should be silently discarded.

The agent remembers the highest-priority nominated pairs in the Valid list for each component of the media stream, called the previous selected pairs before removing all the selected candidate pairs from the Valid List . It continues sending media to that address until it finishes with the steps described below. Because those packets might not be received due to the mobility event, it MAY cache a copy of those packets.

1. The ICE endpoint constructs a pair whose local candidate is equal to the transport address on which the STUN request was received with MOBILITY-EVENT, USE-CANDIDATE attributes and a remote candidate equal to the source transport address where the STUN request came from.
2. The ICE endpoint will add this pair to the valid list if not already present.
3. The agent sets the nominated flag for that pair in the valid pair to true. ICE processing is considered complete for a media stream if the valid list contains a selected candidate pair for each component and ICE agent can start sending media.

The ICE endpoint will follow Steps 1 to 3 when subsequent STUN Binding Requests are received with MOBILITY-EVENT and USE-CANDIDATE attributes.

3.2. Presence of other interfaces in Valid list

Note : This technique is optional and only relevant if there is a host policy to maintain unused candidates on other interfaces using the steps in Section 3.3.1. When the interface currently being used for media communication becomes unavailable. If other interfaces are available and local candidates from these interfaces are already present in the valid list then ICE endpoint will perform the following steps :

1. The ICE endpoint based on the locally configured host policy preferences, will select a interface whose candidates are already present in the valid list.
2. The ICE endpoint clears all the pairs in the valid list containing the IP addresses from the interface that become unavailable.
3. The ICE endpoint initiates ICE connectivity checks on the selected interface. The ICE endpoint acts as controlling agent and MUST include MOBILITY-EVENT attribute to signal mobility event and SHOULD also include the USE-CANDIDATE attribute to

signal an aggressive nomination (see Section 2.6 of [RFC5245]). When all components have a nominated pair in the valid list, media can begin to flow using the highest priority nominated pair.

4. The ICE endpoint will re-establish connection with the SIP proxy. Once ICE connectivity checks for all of the media streams are completed, the controlling ICE endpoint follows the procedures in Section 11.1 of [RFC5245], specifically to send updated offer if the candidates in the m and c lines for the media stream (called the DEFAULT CANDIDATES) do not match ICE's SELECTED CANDIDATES (also see Appendix B.9 of [RFC5245]).

The ICE endpoint after Mobility using ICE is successful can issue an updated offer indicating ICE restart if higher priority interface becomes available.

3.2.1. Receiving ICE Mobility event

The ICE endpoint that receives ICE Mobility Event will perform the steps in Section 3.1.1.

3.3. Losing an Interface

When an interface is lost, the SDP MAY be updated, so that the remote ICE host does not waste its efforts with connectivity checks to that address, as those checks will fail. Because it can be argued that this is merely an optimization, and that the interface loss might be temporary (and soon regained), and that ICE has reasonable accommodation for candidates where connectivity checks timeout, this specification does not strongly encourage updating the SDP to remove a lost interface.

Likewise, this specification recommends that ICE candidate addresses in valid list be maintained actively, subject to the host's policy. For example, battery operated hosts have a strong incentive to not maintain NAT binding for server reflexive candidates learnt through STUN Binding Request, as the maintenance requires sending periodic STUN Binding Indication. As another example, a host that is receiving media over IPv6 may not want to persist with keeping a NATted IPv4 mapping alive (because that consumes a NAT mapping that could be more useful to a host actively utilizing the mapping for real traffic).

Note: this differs from Section 8.3 of [RFC5245], which encourages abandoning unused candidates.

3.3.1. Keeping unused candidates in the valid list active

ICE endpoint subject to host policy can continue performing ICE connectivity checks using candidates from other interfaces on the host even after ICE is complete. If valid list contains unused candidate pairs from other interfaces and one of these interfaces can be selected to send to media in case the existing interface used for media is unavailable then ICE endpoint can keep the unused candidate pairs from other interface{s} alive by sending keepalives every NN seconds. It is recommended to only keep host/server-reflexive candidates active in the valid list and not the relayed candidates.

3.3.1.1. Sending keep alive requests

Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows [RFC6263] describes various reasons for doing keepalives on inactive streams and how to keep NAT mapping alive. However this specification requires some additional functionality associated with the keepalives.

STUN binding requests MUST be used as the keepalive message instead of the STUN Binding indication as specified in [RFC5245]. This is to ensure positive peer consent from the remote side that the candidate pair is still active and in future mobility can be achieved using the steps in Section 3.2 . The request must include the MOBILITY-SUPPORT attribute. If the STUN binding response matches a pair in the checklist then that candidate pair should be kept in the list. If the STUN transaction fails then the candidate pair will be removed from valid list.

3.3.1.2. Receiving keep alive requests

Upon receiving a STUN binding request containing a MOBILITY-SUPPORT attribute even when ICE processing is in the Completed state, the ICE endpoint will add this pair to the valid list if not already present and generate STUN Binding Response containing the MOBILE-SUPPORT attribute.

3.3.2. Keeping unused relayed candidates active

Discussion : The ICE endpoints can maintain the relayed candidates active even when not actively used, so that relayed candidates can be tried if ICE connectivity checks using other candidate types fails. The ICE agent will have to create permissions in the TURN server for the remote relayed candidate IP addresses and perform the following steps :

1. The ICE agent will keep the relayed candidates alive using Refresh transaction, as described in [RFC5766].
2. When the endpoint IP address changes due to mobility, the ICE agent will refresh it's allocation with TURN server using Section 5.2.
3. The ICE agent will pair local and remote relayed candidates for connectivity checks when performing the steps in Section 3.1.
4. If the ICE connectivity check succeeds only with local and remote relayed candidates, it suggests that either other peer is roaming at the same time or is behind Address-Dependent Filtering NAT. The ICE agent adds the relayed candidate pair to the valid list and marks it as selected. The ICE agent can now send media using the newly selected relayed candidate pair. The Mobile device must re-establish connection with SIP proxy, issue an updated offer indicating ICE restart so that media can switched to higher-priority candidate pairs.

This approach assists Mobility using ICE to succeed but brings in additional overhead of maintaining relayed candidates.

3.4. New STUN Attributes

Three new attributes are defined by this section: MOBILITY-EVENT, MOBILITY-SUPPORT.

The MOBILITY-EVENT attribute indicate the sender experienced a mobility event. This attribute has no value, thus the attribute length field MUST always be 0. Rules for sending and interpretation of receiving are described above.

The MOBILITY-SUPPORT attribute indicates the sender supports ICE Mobility, as defined in this document. This attribute has no value, thus the attribute length field MUST always be 0. Rules for sending and interpretation of receiving are described above.

4. Make Before Break

When a new interface comes up and initially selected interface becomes deprioritized (e.g due to a low cost interface becoming available). The ICE endpoint re-connects to the SIP proxy using the new interface, gather candidates, exchange updated offer/exchange to restart ICE. Once ICE processing has reached the Completed state then the ICE endpoint can successfully switch the media over to the new interface. The interface initially used for communication can

now be turned off without disrupting communications.

5. Mobility using TURN

To achieve mobility, a TURN client should be able to retain an allocation on the TURN server across changes in the client IP address as a consequence of movement to other networks.

When the client sends the initial Allocate request to the TURN server, it will also include the new STUN attribute MOBILITY-TICKET (with zero length value), which indicates that the client is capable of mobility and desires a ticket. The TURN server provisions a ticket that is sent inside the new STUN attribute MOBILITY-TICKET in the Allocate Success response to the client. The ticket will be used by the client when it wants to refresh the allocation but with a new client IP address and port. It also ensures that the allocation can only be refreshed this way by the same client. When a client's IP address changes due to mobility, it presents the previously obtained ticket in a Refresh Request to the TURN server. If the ticket is found to be valid, the TURN server will retain the same relayed address/port for the new IP address/port allowing the client to continue using previous channel bindings -- thus, the TURN client does not need to obtain new channel bindings. Any data from external peer will be delivered by the TURN server to this new IP address/port of the client. The TURN client will continue to send application data to its peers using the previously allocated channelBind Requests.

TURN client	TURN server	Peer A
-- Allocate request ----->		
+ MOBILITY-TICKET (length=0)		
<----- Allocate failure --		
(401 Unauthorized)		
-- Allocate request ----->		
+ MOBILITY-TICKET (length=0)		
<----- Allocate success resp --		
+ MOBILITY-TICKET		
...
(changes IP address)		
-- Refresh request ----->		
+ MOBILITY-TICKET		
<----- Refresh success resp --		
+ MOBILITY-TICKET		

5.1. Creating an Allocation

5.1.1. Sending an Allocate Request

In addition to the process described in Section 6.1 of [RFC5766], the client includes the MOBILITY-TICKET attribute with length 0. This indicates the client is a mobile node and wants a ticket.

5.1.2. Receiving an Allocate Request

In addition to the process described in Section 6.2 of [RFC5766], the server does the following:

If the MOBILITY-TICKET attribute is included, and has length zero, and the TURN session mobility is forbidden by local policy, the server MUST reject the request with the new Mobility Forbidden error code. Following the rules specified in [RFC5389], if the server does not understand the MOBILITY-TICKET attribute, it ignores the attribute.

If the server can successfully process the request create an allocation, the server replies with a success response that includes a STUN MOBILITY-TICKET attribute. TURN server stores it's session state, such as 5-tuple and NONCE, into a ticket that is encrypted by a key known only to the TURN server and sends the ticket in the STUN

MOBILITY-TICKET attribute as part of Allocate success response.

The ticket is opaque to the client, so the structure is not subject to interoperability concerns, and implementations may diverge from this format. TURN Allocation state information is encrypted using 128-bit key for Advance Encryption Standard (AES) and 256-bit key for HMAC-SHA-256 for integrity protection.

5.1.3. Receiving an Allocate Success Response

In addition to the process described in Section 6.3 of [RFC5766], the client will store the MOBILITY-TICKET attribute, if present, from the response. This attribute will be presented by the client to the server during a subsequent Refresh request to aid mobility.

5.1.4. Receiving an Allocate Error Response

If the client receives an Allocate error response with error code TBD (Mobility Forbidden), the error is processed as follows:

- o TBD (Mobility Forbidden): The request is valid, but the server is refusing to perform it, likely due to administrative restrictions. The client considers the current transaction as having failed. The client MAY notify the user or operator and SHOULD NOT retry the same request with this server until it believes the problem has been fixed.

All other error responses must be handled as described in [RFC5766].

5.2. Refreshing an Allocation

5.2.1. Sending a Refresh Request

If a client wants to refresh an existing allocation and update its time-to-expiry or delete an existing allocation, it will send a Refresh Request as described in Section 7.1 of [RFC5766]. If the client wants to retain the existing allocation in case of IP change, it will include the MOBILITY-TICKET attribute received in the Allocate Success response. If a Refresh transaction was previously made, the MOBILITY-TICKET attribute received in the Refresh Success response of the transaction must be used.

5.2.2. Receiving a Refresh Request

In addition to the process described in Section 7.2 of [RFC5766], the client does the following:

If the STUN MOBILITY-TICKET attribute is included in the Refresh

Request then the server will not retrieve the 5-tuple from the packet to identify an associated allocation. Instead TURN server will decrypt the received ticket, verify the ticket's validity and retrieve the 5-tuple allocation from the contents of the ticket. If this 5-tuple obtained from the ticket does not identify an existing allocation then the server MUST reject the request with an error.

If the source IP address and port of the Refresh Request is different from the stored 5-tuple allocation, the TURN server proceeds with checks to see if NONCE in the Refresh request is the same as the one provided in the ticket. The TURN server also uses MESSAGE-INTEGRITY validation to identify that it is the same user which had previously created the TURN allocation. If the above checks are not successful then server MUST reject the request with a 441 (Wrong Credentials) error.

If all of the above checks pass, the TURN server understands that the client has moved to a new network and acquired a new IP address. The source IP address of the request could either be the host transport address or server-reflexive transport address. The server then updates its 5-tuple with the new client IP address and port. TURN server calculates the ticket with the new 5-tuple and sends the new ticket in the STUN MOBILITY-TICKET attribute as part of Refresh Success response.

5.2.3. Receiving a Refresh Response

In addition to the process described in Section 7.3 of [RFC5766], the client will store the MOBILITY-TICKET attribute, if present, from the response. This attribute will be presented by the client to the server during a subsequent Refresh Request to aid mobility.

5.3. New STUN Attribute MOBILITY-TICKET

This attribute is used to retain an Allocation on the TURN server. It is exchanged between the client and server to aid mobility. The value is encrypted and identifies session state such as 5-tuple and NONCE. The value of MOBILITY-TICKET is a variable-length value.

5.4. New STUN Error Response Code

This document defines the following new error response code:

Mobility Forbidden: Mobility request was valid but cannot be performed due to administrative or similar restrictions.

6. IANA Considerations

IANA is requested to add the following attributes to the STUN attribute registry [iana-stun],

- o MOBILITY-TICKET (0x802E, in the comprehension-optional range)
- o MOBILITY-EVENT (0x802, in the comprehension-required range)
- o MOBILITY-SUPPORT (0x8000, in the comprehension-optional range)

and to add a new STUN error code "Mobility Forbidden" with the value 501 to the STUN Error Codes registry [iana-stun].

7. Security Considerations

7.1. Considerations for ICE mechanism

A mobility event only occurs after both ICE endpoints have exchanged their ICE information. Thus, both username fragments are already known to both endpoints. Each endpoint contributes at least 24 bits of randomness to the ice-ufrag (Section 15.4 of [RFC5245]), which provides 48 bits of randomness. An off-path attacker would have to guess those 48 bits to cause the endpoints to perform HMAC-SHA1 validation of the MESSAGE-INTEGRITY attribute.

An attacker on the path between the ICE endpoints will see both ice-ufrags, and can cause the endpoints to perform HMAC-SHA1 validation by sending messages from any IP address.

7.2. Considerations for TURN mechanism

TURN server MUST use strong encryption and integrity protection for the ticket to prevent an attacker from using a brute force mechanism to obtain the ticket's contents or refreshing allocations.

Security considerations described in [RFC5766] are also applicable to this mechanism.

8. Acknowledgements

Thanks to Alfred Heggstad, Lishitao, Sujing Zhou, Martin Thomson, Emil Ivov for review and comments.

9. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

9.1. Changes from draft-wing-mmusic-ice-mobility-00 to -01

- o Updated section 3

9.2. Changes from draft-wing-mmusic-ice-mobility-01 to -02

- o Updated Introduction, Notational Conventions, sections 3.1, 3.2.
- o Updated section 3.5

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

10.2. Informative References

- [RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, June 2011.
- [iana-stun] IANA, "IANA: STUN Attributes", April 2011, <<http://www.iana.org/assignments/stun-parameters/stun-parameters.xml>>.

Authors' Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marthalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: praspati@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tiredy@cisco.com

Paal-Erik Martinsen
Cisco Systems, Inc.
Philip Pedersens vei 22
Lysaker, Akershus 1325
Norway

Email: palmarti@cisco.com

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: December 19, 2014

D. Wing
T. Reddy
P. Patil
P. Martinsen
Cisco
June 17, 2014

Mobility with ICE (MICE)
draft-wing-mmusic-ice-mobility-07

Abstract

This specification describes how endpoint mobility can be achieved using ICE.

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 December 19, 2014.

Copyright Notice

Copyright (c) 2014 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

1. Introduction	3
2. Notational Conventions	3
3. Break Before Make	4
3.1. Absence of other interfaces in Valid list	5
3.1.1. Receiving ICE Mobility event	6
3.2. Keeping unused relayed candidates active	7
3.3. New STUN Attributes	8
4. Make Before Break	8
5. Comparison to ICE Restart and Trickle ICE	8
5.1. Break Before Make - ICE Restart	9
5.2. Break Before Make - Trickle ICE	10
6. IANA Considerations	10
7. Security Considerations	11
8. Acknowledgements	11
9. Change History	11
9.1. Changes from draft-wing-mmusic-ice-mobility-00 to -01	11
9.2. Changes from draft-wing-mmusic-ice-mobility-01 to -02	11
9.3. Changes from draft-wing-mmusic-ice-mobility-02 to -03	11
9.4. Changes from draft-wing-mmusic-ice-mobility-03 to -04	12
9.5. Changes from draft-wing-mmusic-ice-mobility-04 to -05	12
9.6. Changes from draft-wing-mmusic-ice-mobility-05 to -06	12
9.7. Changes from draft-wing-mmusic-ice-mobility-06 to -07	12
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Appendix A.	13
A.1. Presence of other interfaces in Valid list	13
A.1.1. Receiving ICE Mobility event	14
A.2. Losing an Interface	14
A.2.1. Keeping unused candidates in the valid list active	15
Authors' Addresses	16

1. Introduction

When moving between networks, an endpoint has to change its IP address. This change breaks upper layer protocols such as TCP and RTP. Various techniques exist to prevent this breakage, all tied to making the endpoint's IP address static (e.g., Mobile IP, Proxy Mobile IP, LISP). Other techniques exist, which make the upper layer protocol ambivalent to IP address changes (e.g., SCTP). The mechanisms described in this document are in that last category.

ICE [RFC5245] ensures two endpoints have a working media path between them, and is typically used by Internet-connected interactive media systems (e.g., SIP endpoints). ICE does not expect either the local host or the remote host to change their IP addresses. Although ICE does allow an "ICE restart", this is done by sending a re-INVITE which goes over the SIP signaling path. The SIP signaling path is often slower than the media path (which needs to be recovered as quickly as possible), consumes an extra half round trip, and incurs an additional delay if the mobility event forces the endpoint to re-connect with its SIP proxy. When a device changes its IP address, it is necessary for it to re-establish connectivity with its SIP proxy, which can be performed in parallel with the steps described in this document. This document describes how mobility is performed entirely in the media path, without the additional delay of re-establishing SIP connectivity, issuing a new offer/answer, or the complications of multiple SIP offers. This document considers re-establishing bi-directional media the most critical aspect of a successful mobility event, and its efforts are towards meeting that goal.

This document proposes a mechanism to achieve RTP mobility when both endpoints support MICE. When both endpoints support MICE, ICE itself can be used to provide mobility. When only one endpoint supports MICE, a TURN server provides mobility as described in [I-D.wing-tram-turn-mobility]. Both mobility techniques work across and between network types (e.g., between 3G and wired Internet access), so long as the client can still access the remote ICE peer or TURN server.

Readers are assumed to be familiar with ICE [RFC5245].

2. Notational Conventions

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].

This note uses terminology defined in [RFC5245], and the following

additional terminology:

Break Before Make: The initially selected interface for communication may become unavailable (e.g due to loss of coverage when moving out of a WiFi hotspot) and new interfaces may become available due to administrative action (e.g manual activation of a specific connectivity technology) or due to dynamic conditions (e.g. Entering coverage area of a wireless network).

Make Before Break: The initially selected interface for communication may become deprioritized (e.g new interface becoming available and it's per bit cost is cheaper and the connection speed is faster than existing interface used for communication).

Simultaneous Mobility: If both the endpoints are mobile and roam at the same time between networks.

3. Break Before Make

When both endpoints support ICE, ICE itself can provide mobility functions. One of the primary aspects of ICE is its address gathering, wherein ICE has each endpoint determine all of the IP addresses and ports that might be usable for that endpoint and communicate that list of addresses and ports to its peer, usually over SDP. That enables the next primary aspect of ICE, which is its connectivity checks: each ICE endpoint sends a connectivity check from a checklist created by the local and remote candidates exchanged in the initial offer/answer exchange. When the ICE endpoint checks the mapped address from the STUN response during ICE connectivity checks and finds that the transport address does not match any of the local candidates that the ICE agent knows about, the mapped address represents a new candidate -- a peer reflexive candidate. This will cause the endpoint to construct a new pair and insert it into the local checklist (Section 7.2.1.3 of [RFC5245]). ICE Mobility (MICE) takes advantage of that existing ICE functionality to provide faster mobility.

Endpoints that support ICE Mobility perform ICE normally, and MUST also include the MOBILITY-SUPPORT attribute in all of their STUN requests and their STUN responses. The inclusion of this attribute allows the ICE peer to determine if it can achieve mobility using ICE or needs to use TURN. To force the use of TURN to achieve ICE mobility, the ICE endpoint SHOULD NOT respond to ICE connectivity checks that have an IP address and port different from the TURN server, unless those connectivity checks contain the MOBILITY-SUPPORT attribute. In this way, the remote peer will think those other candidates are invalid (because its connectivity checks did not

succeed).

After concluding ICE and moving to the ICE completed state (see Section 8 of [RFC5245]) either endpoint or both endpoints can initiate ICE Mobility, no matter if it was the Controlling Agent or the Controlled Agent during normal ICE processing.

3.1. Absence of other interfaces in Valid list

When the interface currently being used for communication becomes unavailable then ICE agent acquires a list of interfaces that are available and based on the locally configured host policy preferences, the ICE endpoint performs ICE Mobility using one of the available interfaces. In this case local candidates from the selected interface are not present in the valid list. ICE Mobility is performed by:

1. The ICE agent remembers the remote host/server reflexive/peer reflexive candidates for each component of the media streams previously used from the valid list before clearing its ICE check list and ICE Valid List.
2. The ICE endpoint gathers host candidates of the same address family as the remote peer on the new interface, forms a check list by creating candidate pairs with local host candidates and remote host/server-reflexive candidates collected in step 1, performs "Computing Pair Priority and Ordering Pairs" (Section 5.7.2 of [RFC5245]), "Pruning the Pairs" (Section 5.7.3 of [RFC5245]), "Computing states" (Section 5.7.4 of [RFC5245]).
3. The ICE endpoint initiates ICE connectivity checks on those candidates from the check list in the previous step, and includes the MOBILITY-EVENT attribute in those connectivity checks.
4. The ICE endpoint acts as controlling agent and the ICE connectivity check from the previous step SHOULD also include the USE-CANDIDATE attribute to signal an aggressive nomination (see Section 2.6 of [RFC5245]).
5. The ICE endpoint performs "Discovering Peer Reflexive Candidates" (Section 7.1.3.2.1 of [RFC5245]), "Constructing a Valid Pair" (Section 7.1.3.2.2 of [RFC5245]), "Updating Pair States" (Section 7.1.3.2.3 of [RFC5245]), and "Updating the Nominated Flag" (Section 7.1.3.2.4 of [RFC5245]). When the valid list contains a candidate pair for each component then ICE processing is considered complete for the media stream and ICE agent can start sending media using the nominated candidate pair.

6. Once ICE connectivity checks for all of the media streams are completed, the controlling ICE endpoint follows the procedures in Section 11.1 of [RFC5245], specifically to send updated offer if the candidates in the m and c lines for the media stream (called the DEFAULT CANDIDATES) do not match ICE's SELECTED CANDIDATES (also see Appendix B.9 of [RFC5245]).

The ICE endpoint even after Mobility using ICE is successful can issue an updated offer indicating ICE restart if connectivity checks using higher priority candidate pairs are not successful.

Mobility using ICE could fail in case of Simultaneous Mobility or if the ICE peer is behind NAT that performs Address-Dependent Filtering (see Section 5 of [RFC5245]). Hence the ICE endpoint in parallel will re-establish connection with the SIP proxy. It will then determine whether to initiate ICE restart under the following conditions:

- a. After re-establishing connection with the SIP proxy and before sending new offer to initiate ICE restart if Mobility using ICE is successful then stop sending the new offer.
- b. After successful negotiation of updated offer/answer to initiate ICE restart, proceed with ICE restart and stop Mobility using ICE if ICE checks are in the Running/Failed states or ICE is partially successful and not yet reached ICE complete state. It's not implementation friendly to have to two checks running in parallel. ICE restart can re-use partial successful ICE connectivity check results from Mobility using ICE if required as optimization.

3.1.1. Receiving ICE Mobility event

A STUN Binding Request containing the MOBILITY-EVENT attribute MAY be received by an ICE endpoint. The agent MUST use short-term credential to authenticate the STUN request containing the MOBILITY-EVENT attribute and perform a message integrity check. The ICE endpoint will generate STUN Binding Response containing the MOBILE-SUPPORT attribute and the ICE agent takes role of controlled agent. If STUN Request containing the MOBILITY-EVENT attribute is received before the endpoint is in the ICE Completed state, it should be silently discarded.

The agent remembers the highest-priority nominated pairs in the Valid list for each component of the media stream, called the previous selected pairs before removing all the selected candidate pairs from the Valid List. It continues sending media to that address until it finishes with the steps described below. Because those packets might

not be received due to the mobility event, it MAY cache a copy of those packets.

1. The ICE endpoint constructs a pair whose local candidate is equal to the transport address on which the STUN request was received with MOBILITY-EVENT, USE-CANDIDATE attributes and a remote candidate equal to the source transport address where the STUN request came from.
2. The ICE endpoint will add this pair to the valid list if not already present.
3. The agent sets the nominated flag for that pair in the valid pair to true. ICE processing is considered complete for a media stream if the valid list contains a selected candidate pair for each component and ICE agent can start sending media.

The ICE endpoint will follow Steps 1 to 3 when subsequent STUN Binding Requests are received with MOBILITY-EVENT and USE-CANDIDATE attributes.

3.2. Keeping unused relayed candidates active

The ICE endpoints can maintain the relayed candidates active even when not actively used, so that relayed candidates can be tried if ICE connectivity checks using other candidate types fails. The ICE agent will have to create permissions in the TURN server for the remote relayed candidate IP addresses and perform the following steps:

1. The ICE agent will keep the relayed candidates alive using Refresh transaction, as described in [RFC5766].
2. When the endpoint IP address changes due to mobility, the ICE agent will refresh it's allocation with TURN server using [I-D.wing-tram-turn-mobility].
3. The ICE agent will pair local and remote relayed candidates for connectivity checks when performing the steps in Section 3.1.
4. If the ICE connectivity check succeeds only with local and remote relayed candidates, it suggests that either other peer is roaming at the same time or is behind Address-Dependent Filtering NAT. The ICE agent adds the relayed candidate pair to the valid list and marks it as selected. The ICE agent can now send media using the newly selected relayed candidate pair. The Mobile device must re-establish connection with SIP proxy, issue an updated offer indicating ICE restart so that media can switched to

higher-priority candidate pairs.

This approach assists Mobility using ICE to succeed but brings in additional overhead of maintaining relayed candidates. In case of Simultaneous Mobility, host candidates can change for both the endpoints by maintaining relayed candidates and using [I-D.wing-tram-turn-mobility], media session can be established using the relayed candidate pair.

3.3. New STUN Attributes

Three new attributes are defined by this section: MOBILITY-EVENT, MOBILITY-SUPPORT.

The MOBILITY-EVENT attribute indicates the sender experienced a mobility event. This attribute has no value, thus the attribute length field MUST always be 0. Rules for sending and interpretation of receiving are described above.

The MOBILITY-SUPPORT attribute indicates the sender supports ICE Mobility, as defined in this document. This attribute has no value, thus the attribute length field MUST always be 0. Rules for sending and interpretation of receiving are described above.

4. Make Before Break

When a new interface comes up and initially selected interface becomes deprioritized (e.g. due to a low cost interface becoming available). The ICE endpoint re-connects to the SIP proxy using the new interface, gathers candidates, exchanges updated offer/exchange to restart ICE. Once ICE processing has reached the Completed state then the ICE endpoint can successfully switch the media over to the new interface. The interface initially used for communication can now be turned off without disrupting communications.

5. Comparison to ICE Restart and Trickle ICE

There has been some concern that ICE Mobility is unnecessary, and that an ICE restart (section 9.1.1.1 of [RFC5245]) would provide exactly the same functionality as ICE Mobility. These sections examine how ICE restart and Trickle ICE [I-D.rescorla-mmusic-ice-trickle] compare with ICE Mobility.

5.1. Break Before Make - ICE Restart

- o If ICE Restart is used for RTP Mobility then in case of Break before Make,
 - 1. Before the endpoint can send an ICE restart message, it has to first re-establish communication with its SIP proxy. This consumes one round-trip for both TCP and UDP. If the connection is protected with TLS (TCP) or DTLS (UDP), we can assume TLS session resumption [RFC5077] will be used to reduce the number of TLS messages. With TLS session resumption, this consumes 1 round trip. If TLS session resumption is not available, a full TLS handshake consumes 2 round trips. This is a total of 2 round trips (with session resumption) to 3 round trips (without session resumption), which is multiplied by the round trip time to the SIP proxy. The round trip time is dependent on a particular network or deployment, for example in second (2.5G), third (3G) generation wireless networks and satellite communication round trip time could be higher than 250ms. These calculations are only considering the network round-trip time and do not consider the wall-clock time to validate the TLS certificates or generate the TLS keys on the TLS client or the TLS server, which would make this longer.
 - 2. While performing the above steps to re-establish SIP connectivity with its SIP proxy, the endpoint will gather host candidates which incur no network traffic, server-reflexive candidates which incur a round-trip to a STUN server, and relayed candidates which incurs three round trips (two for re-authentication and one for creating the TURN permission). The STUN and TURN communications can be performed in parallel with the SIP connectivity check from step (1), above.
 - 3. The endpoints through the SIP server will exchange offer/answer. The SIP server could also be located halfway around the world from the endpoints and the delay could be significant. For SIP over UDP the endpoint will have send a SIP request and wait for the response to arrive.
 - 4. ICE restart requires sending a new INVITE. A new INVITE cannot be sent if there is an open SIP dialog, such as a previous INVITE. This means rapid mobility events will not work well, and there is also an increased likelihood for glare (both endpoints sending INVITES at the same time).

5.2. Break Before Make - Trickle ICE

- o If Trickle ICE [I-D.rescorla-mmusic-ice-trickle] is used for RTP Mobility then in case of Break before Make,
 - 1. Trickle ICE can begin connectivity checks while the endpoint is still gathering candidates and can considerably shorten the time necessary for ICE processing to complete. It still involves the overhead of step 1 explained in section Section 5.1.
 - 2. The endpoint would learn host candidates and inform them to the remote peer in offer, the remote peer will provide its candidates in answer. The host, server reflexive, peer reflexive and relayed candidates of the remote peer may not change and the remote peer does not have to gather the candidates again. Trickle ICE will test local host candidates with all types of remote candidates provided by the remote peer in the answer.
 - a. If the endpoint is not behind NAT and the ICE peer is behind NAT performing endpoint dependent filtering (or firewall blocking unsolicited incoming traffic) then ICE connectivity checks initiated by the endpoint to the remote peer will succeed as a consequence of suicide ICE connectivitivy check packets.
 - b. If the endpoint is behind NAT and ICE peer is behind endpoint-dependent filtering NAT then ICE connectivity checks using the first offer/answer will fail but will later succeed in subsequent offer/answer where the endpoint provides server-reflexive candidates.
 - 3. Trickle ICE must be supported by both endpoints for it be used.
- o If both endpoints support TRICKLE ICE then it is RECOMMENDED that TRICKLE ICE be tried instead of ICE restart in steps (a) and (b) of Section 3.1.

6. IANA Considerations

IANA is requested to add the following attributes to the STUN attribute registry [iana-stun],

- o MOBILITY-EVENT (0x802, in the comprehension-required range)

- o MOBILITY-SUPPORT (0x8000, in the comprehension-optional range)

7. Security Considerations

A mobility event only occurs after both ICE endpoints have exchanged their ICE information. Thus, both username fragments are already known to both endpoints. Each endpoint contributes at least 24 bits of randomness to the ice-ufrag (Section 15.4 of [RFC5245]), which provides 48 bits of randomness. An off-path attacker would have to guess those 48 bits to cause the endpoints to perform HMAC-SHA1 validation of the MESSAGE-INTEGRITY attribute.

An attacker on the path between the ICE endpoints will see both ice-ufrags, and can cause the endpoints to perform HMAC-SHA1 validation by sending messages from any IP address.

8. Acknowledgements

Thanks to Alfred Heggstad, Lishitao, Sujing Zhou, Martin Thomson, Emil Ivov for review and comments.

9. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

9.1. Changes from draft-wing-mmusic-ice-mobility-00 to -01

- o Updated section 3

9.2. Changes from draft-wing-mmusic-ice-mobility-01 to -02

- o Updated Introduction, Notational Conventions, sections 3.1, 3.2.
- o Updated section 3.5

9.3. Changes from draft-wing-mmusic-ice-mobility-02 to -03

- o Moved sections Presence of other interfaces in Valid list, Losing an Interface to Appendix.

9.4. Changes from draft-wing-mmusic-ice-mobility-03 to -04

- o Added Section 6.

9.5. Changes from draft-wing-mmusic-ice-mobility-04 to -05

- o Updated Section 6.

9.6. Changes from draft-wing-mmusic-ice-mobility-05 to -06

- o Updated Section 5.
- o Added Implementation Status section.

9.7. Changes from draft-wing-mmusic-ice-mobility-06 to -07

- o Removed Turn Mobility

10. References

10.1. Normative References

- [I-D.wing-tram-turn-mobility]
Wing, D., Patil, P., Reddy, T., and P. Martinsen,
"Mobility with TURN", draft-wing-tram-turn-mobility-00
(work in progress), June 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment
(ICE): A Protocol for Network Address Translator (NAT)
Traversal for Offer/Answer Protocols", RFC 5245,
April 2010.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing,
"Session Traversal Utilities for NAT (STUN)", RFC 5389,
October 2008.
- [RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using
Relays around NAT (TURN): Relay Extensions to Session
Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

10.2. Informative References

- [I-D.rescorla-mmusic-ice-trickle]
Rescorla, E., Uberti, J., and E. Ivov, "Trickle ICE:

Incremental Provisioning of Candidates for the Interactive Connectivity Establishment (ICE) Protocol", draft-rescorla-mmusic-ice-trickle-01 (work in progress), October 2012.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, May 2010.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, May 2010.

[RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, June 2011.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.

[iana-stun]
IANA, "IANA: STUN Attributes", April 2011,
<<http://www.iana.org/assignments/stun-parameters/stun-parameters.xml>>.

Appendix A.

A.1. Presence of other interfaces in Valid list

This technique is optional and only relevant if there is a host policy to maintain unused candidates on other interfaces using the steps in Appendix A.2.1. ICE Agent can maintain unused candidates on other interfaces if it detects that it is behind Address-Dependent Filtering NAT or Firewall. ICE Agent can detect NAT, Firewall behaviour using the procedure explained in [RFC5780]. When the interface currently being used for media communication becomes unavailable. If other interfaces are available and local candidates from these interfaces are already present in the valid list then ICE endpoint will perform the following steps:

1. The ICE endpoint based on the locally configured host policy preferences, will select a interface whose candidates are already present in the valid list.
2. The ICE endpoint clears all the pairs in the valid list containing the IP addresses from the interface that become unavailable.
3. The ICE endpoint initiates ICE connectivity checks on the selected interface. The ICE endpoint acts as controlling agent and MUST include MOBILITY-EVENT attribute to signal mobility event and SHOULD also include the USE-CANDIDATE attribute to signal an aggressive nomination (see Section 2.6 of [RFC5245]). When all components have a nominated pair in the valid list, media can begin to flow using the highest priority nominated pair.
4. The ICE endpoint will re-establish connection with the SIP proxy. Once ICE connectivity checks for all of the media streams are completed, the controlling ICE endpoint follows the procedures in Section 11.1 of [RFC5245], specifically to send updated offer if the candidates in the m and c lines for the media stream (called the DEFAULT CANDIDATES) do not match ICE's SELECTED CANDIDATES (also see Appendix B.9 of [RFC5245]).

The ICE endpoint after Mobility using ICE is successful can issue an updated offer indicating ICE restart if higher priority interface becomes available.

A.1.1. Receiving ICE Mobility event

The ICE endpoint that receives ICE Mobility Event will perform the steps in Section 3.1.1.

A.2. Losing an Interface

When an interface is lost, the SDP MAY be updated, so that the remote ICE host does not waste its efforts with connectivity checks to that address, as those checks will fail. Because it can be argued that this is merely an optimization, and that the interface loss might be temporary (and soon regained), and that ICE has reasonable accommodation for candidates where connectivity checks timeout, this specification does not strongly encourage updating the SDP to remove a lost interface.

Likewise, this specification recommends that ICE candidate addresses in valid list be maintained actively, subject to the host's policy. For example, battery operated hosts have a strong incentive to not

maintain NAT binding for server reflexive candidates learnt through STUN Binding Request, as the maintenance requires sending periodic STUN Binding Indication. As another example, a host that is receiving media over IPv6 may not want to persist with keeping a NATted IPv4 mapping alive (because that consumes a NAT mapping that could be more useful to a host actively utilizing the mapping for real traffic).

Note: this differs from Section 8.3 of [RFC5245], which encourages abandoning unused candidates.

A.2.1. Keeping unused candidates in the valid list active

ICE endpoint subject to host policy can continue performing ICE connectivity checks using candidates from other interfaces on the host even after ICE is complete. If valid list contains unused candidate pairs from other interfaces and one of these interfaces can be selected to send to media in case the existing interface used for media is unavailable then ICE endpoint can keep the unused candidate pairs from other interface{s} alive by sending keepalives every NN seconds. It is recommended to only keep host/server-reflexive candidates active in the valid list and not the relayed candidates.

A.2.1.1. Sending keep alive requests

Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows [RFC6263] describes various reasons for doing keepalives on inactive streams and how to keep NAT mapping alive. However this specification requires some additional functionality associated with the keepalives.

STUN binding requests MUST be used as the keepalive message instead of the STUN Binding indication as specified in [RFC5245]. This is to ensure positive peer consent from the remote side that the candidate pair is still active and in future mobility can be achieved using the steps in Appendix A.1 . The request must include the MOBILITY-SUPPORT attribute. If the STUN binding response matches a pair in the checklist then that candidate pair should be kept in the list. If the STUN transaction fails then the candidate pair will be removed from valid list.

A.2.1.2. Receiving keep alive requests

Upon receiving a STUN binding request containing a MOBILITY-SUPPORT attribute even when ICE processing is in the Completed state, the ICE endpoint will add this pair to the valid list if not already present and generate STUN Binding Response containing the MOBILE-SUPPORT attribute.

Authors' Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tiredy@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marthalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: praspati@cisco.com

Paal-Erik Martinsen
Cisco Systems, Inc.
Philip Pedersens vei 22
Lysaker, Akershus 1325
Norway

Email: palmarti@cisco.com

