MMUSIC WG J. Elwell
Internet-Draft A. Hutton
Intended status: Informational T. Stach
Expires: November 26, 2011 Siemens Enterprise
Communications
May 25, 2011

ICE Updated Offer Problematic
draft-elwell-mmusic-ice-updated-offer-01.txt

Abstract

Interactive Connectivity Establishment (ICE) requires an updated offer-answer cycle under some circumstances, to satisfy the needs of some devices on the signalling path. When used with SIP, this additional offer-answer cycle interacts with other things, such as fax and third party call control (3PCC). This document describes the problems and discusses possible remedies.

This work is being discussed on the mmusic@ietf.org mailing list.

Table of Contents

1. Introduction

Interactive Connectivity Establishment (ICE) [RFC5245] specifies a mechanism for NAT traversal for multimedia sessions established using the Session Description Protocol (SDP) [RFC4566] offer-answer model [RFC3264]. It allows a pair of endpoints to exchange candidate IP addresses and ports, perform checks to see which pairs of candidates work, and agree which pairs to use for a given component of a given medium (e.g., RTP stream, RTCP stream). The mechanism can also be used for IPv6 transition, for determining whether to use IPv4 or IPv6. A particular application of ICE is with the Session Initiation Protocol (SIP) [RFC3261].

Connectivity checks are performed on the media path between candidate pairs. Based on the results of connectivity checks and certain rules, the two endpoints each determine which pair of candidates to use for a given component and can then start exchanging data (e.g., RTP packets) on the agreed path. Further exchanges on the signalling path (i.e., the path on which the offer-answer exchange is performed) are not necessary for the endpoints to agree which candidates to use.

However, certain SIP/SDP-aware devices on the signalling path need to know which candidates have been selected (e.g., to prioritize that traffic or to remove the resources for non-selected candidates). For this reason ICE mandates a further offer-answer exchange in some circumstances, i.e., an updated SDP offer followed by an updated SDP answer. In some situations with SIP, this updated offer-answer exchange can be problematic. This document examines these problems.

2. Fax calls

2.1. Problem statement

Except where dedicated fax devices are involved, fax calls typically start as audio. Detection of CNG tone (calling tone) from the initiating fax machine and CED (called) tone from the receiving fax machine initiates a switch to T.38, i.e., a switch from audio to image. Where the audio call uses a compressed codec (e.g., G.729), if one tone is detected there may first be a switch to G.711, for more reliable tone detection or in case the call turns out to be a non-fax modem call. Thus there can be:

a switch from a compressed codec to G.711; or

a switch from audio to image; or

both in sequence.

Switching codec or switching from audio to image requires an SDP offer-answer cycle. ICE also requires an updated offer-answer cycle where the selected candidates are not those in the m/c-lines of the original offer-answer. If the UA that detects the need to switch because of fax is also the controlling agent from the ICE perspective, updated offer-answer for fax can follow the updated offer-answer for ICE and probably won't lead to problems.

However, if the UA that detects the need to switch because of fax is not the controlling agent from the ICE perspective, there is a significant danger of the two re-INVITE or UPDATE [RFC3311] requests colliding, resulting in a 491 response to each. According to [RFC3261] and [RFC3311], one UA (the one that owns the Call-ID) backs off for between 2.1 and 4 seconds, and the other UA backs off for between 0 and 2 seconds, before trying again. This can result in a delay of up to 4 seconds before the switch to fax, long enough in practice to cause fax calls to fail. It can also result in a delay of up to 4 seconds before the post-ICE updated offer-answer. SIP/SDP-aware devices that need the post-ICE updated offer-answer might not permit the flow of RTP packets throughout this period, which might also lead to failure of the fax call. An example flow is shown below:

```
  UA1 (Call-ID owner)                          UA2 (fax gateway)
    --------------INVITE / SDP offer audio------------------>
    <---------------183 / SDP answer audio-------------------
    <==================ICE negotiation=====================>
    <----------------200 / SDP answer audio -----------------
    ------------------------ACK------------------------------>
    <======================RTP============================>
 (ICE requires updated offer)
    ------UPDATE / SDP offer audio---
                                 \            (Fax detected)
    <-----UPDATE / SDP offer image----\----------------------
                                   \
                                    -------------------->
    -----------------491-----------
                               \
    <-------------------------------\-----------491----------
                                 \
                                  ---------------------->
                                            (back-off 0-2s)
    <-------------UPDATE / SDP offer image-------------------
    ----------------200 / SDP answer image------------------>
 (back-off 2.1-4s)
    ----UPDATE / SDP offer image (selected candidates)------>
```

```
<----200 / SDP answer image (selected candidates)--------
```

In this example UA1 is the ICE controlling agent and issues an updated offer on completion of ICE, and UA2 is a fax gateway that detects fax and attempts to change to image. UPDATE is supported by both and used for the updated offers. UA1 owns the Call-ID and has the longer back-off. In this example, the switch to image will probably be accomplished fast enough (back-off does not exceed 2 seconds), but the post-ICE updated offer can be delayed up to 4 seconds, perhaps leading to undesirable behaviour by SIP/SDP-aware devices on the signalling path, which might disrupt the flow of RTP and cause the fax call to fail.

Of course, collision of UPDATE or re-INVITE requests will not always occur - it is matter of timing. However, the probability of collision is not insignificant and, if that occurs, the probability of the fax call being adversely affected to the extent that it fails is not insignificant.

2.2. Possible remedies

2.2.1. Delay the ICE updated offer

UA1, as the ICE controlling agent, will be unaware that UA2 will detect fax. Therefore any delay in sending the ICE updated offer will need to apply to all calls and will need to be long enough to allow for differing amounts of time for UA2 to detect fax (perhaps several seconds). The question then is whether this would be long enough to introduce a risk of undesirable behaviour by SIP/SDP-aware devices on the signalling path, which could impact all calls, not just fax calls.

2.2.2. Delay the fax updated offer

UA2 will know that ICE has been used, and therefore can expect an updated offer from UA1, the ICE controlling agent. Normally this should arrive quite quickly (e.g., well under 100 ms), although it depends on the number of SIP intermediaries on the path and whether any retransmissions are needed because of packet loss. Therefore a delay of a 100 ms., say, would probably not impact the fax call and would help avoid collisions but would not be a guarantee.

2.2.3. Use reliable provisional responses and pre-conditions

By using a reliable 183 in accordance with [RFC3262] to send the SDP answer, when ICE completes the updated offer can be sent in an UPDATE request, rather than waiting for the 200 response to the INVITE request and then sending the updated offer. However, the fax machine might auto-answer and send the 200 response to the INVITE request as soon as ICE procedures complete, so the updated offer might collide with the 200 response, again leading to further signalling delays before things are resolved. This in turn could be avoided by using pre-conditions [RFC3312] to delay answering of the call until the updated offer has occurred.

This might work, although it is unclear how pre-conditions are intended to interact with ICE, i.e., whether ICE procedures can continue without waiting for pre-conditions to be satisfied. Perhaps an extension to pre-conditions would be required. Also this might introduce further adverse interactions with SIP/SDP-aware devices on the signalling path.

Even if it could be made to work, this approach would require the entities involved to support [RFC3262] and [RFC3312]. [RFC3262] is known to be rather complicated to implement (hence the reason the ICE mechanism was specifically designed to allow SDP answer to be sent in an unreliable provisional response (ICE provides acknowledgement on the media path, rather than requiring the use of PRACK). Pre-conditions are a further complication and not widely implemented. Therefore ICE implementations should not be expected to support [RFC3262] and [RFC3312].

3. Third party call control (3PCC)

3.1. Problem statement

3PCC [RFC3725] is a common technique used with SIP where calls are controlled from an application at a SIP B2BUA. In particular, calls can be established by 3PCC, whereby the application first makes a call to the first party (typically the device of a user requesting the call) and then makes a second call to the second party, the two calls being joined together such that media flow directly between the two devices. A typical implementation is in accordance with Flow I in [RFC3725]: the controlling B2BUA sends an offerless INVITE request to UA1, which alerts the first user. When the user answers, UA1 sends an offer in a 200 response to the INVITE request, and this offer is used by the B2BUA in a second INVITE request, this time to UA2.

The problem with using ICE with 3PCC is that 3PCC signalling does not permit the updated offer-answer for ICE to occur in a timely manner. UA2 will often take some time (seconds or tens of seconds) before sending the 200 response to its INVITE request. Yet if UA2 has already sent an SDP answer (e.g., in a 183 response), ICE can complete on the media paths and UA1, as the ICE controlling agent, can attempt an updated offer. This updated offer cannot be forwarded to UA2 until the INVITE transaction on that leg of the call has completed.

More specifically, consider the following example flow:

```
  UA1 (Call-ID owner)         B2BUA                        UA2
    <----INVITE (no SDP)------
    -----200 / SDP offer----->     ----INVITE / SDP offer--->
    <----ACK / SDP answer-----     <-----183 / SDP answer----
    <==================ICE negotiation=====================>
(ICE requires updated offer)
    ----UPDATE / SDP offer---> What next?
```

In this case, UA2 sends a 183 provisional response to its INVITE request. This contains an SDP answer, which is passed to UA1 through the ACK request. Thus UA1 and UA2 are able to conduct ICE negotiation on the media paths. UA2 will probably not alert its user until ICE negotiation is complete, but anyway, there will often be a significant delay before the user answers and UA2 sends a 200 response to its INVITE request. Meanwhile, UA1, as the ICE controlling agent, attempts to send an updated offer. In this case it chooses to use an UPDATE request, but similar considerations apply if it uses a re-INVITE request. The B2BUA cannot pass that request on until the INVITE transaction with UA2 has completed. Either the UPDATE request has to be delayed somehow or rejected, in either case leading to the possibility of undesirable behaviour by SIP/SDP-aware devices that require a timely updated offer. For example, UA2 might be transmitting early media, which might fail to be passed through correctly, or clipping might occur when the user answers.

It should be noted that the issue of sending an updated offer in a 3PCC situation before UA2 has answered is not solely an ICE issue. However, ICE substantially increases the need for such an updated offer.

3.2. Possible remedies

3.2.1. Avoid 3PCC

There are alternatives to this form of 3PCC. For example, UA1 could be instructed to issue a conventional INVITE request by sending a SIP REFER request to UA1, or by some non-SIP means. However, using a REFER request is not an option for some types of UA, for example PSTN gateways. If user 1 is a PSTN user, it is necessary to make a PSTN call to the user, and this can be achieved by sending an INVITE request to UA1, but not by sending a REFER request to UA1. Non-SIP means are either not standardized or little deployed.

A particular example of an application that uses 3PCC is one where the user uses a web page to make the call, having selected in advance the device he/she wishes to use to make the call. The application causes the B2BUA to send an INVITE request to that selected device, followed by an INVITE request to the called destination. If the selected device is, for example, a cellular device reachable via PSTN, that initial INVITE request will be sent to a PSTN gateway.

Because of the difficulties supporting such applications by other means, 3PCC is a commonly deployed technique. It is not possible to scrap 3PCC in order to introduce ICE.

3.2.2. Delay the updated offer

UA1 will typically not be aware of the state of the INVITE transaction to UA2, and will issue the updated offer in an UPDATE or re-INVITE request without knowing whether the B2BUA can pass it on. Therefore the onus is on the B2BUA to handle the situation when it receives the UPDATE or re-INVITE request. As a non-INVITE transaction, an UPDATE request has a relatively short timeout, but one possibility would be for the B2BUA to reject it with a 500 response and a Retry-After header field, relying on UA1 to try again later. In the case of re-INVITE, the B2BUA could delay forwarding the request to UA2 until the original transaction is complete. However, in either case, SIP/SDP-aware devices between the B2BUA and UA2 will not see the updated offer in a timely manner, and therefore might take action that prevents the correct handling of early media or clips media for a short time after the call is answered.

3.2.3. Delay ICE until UA2 answers

UA2 could delay ICE until UA2 answers, which means UA2 would not need to send SDP answer in a provisional response but could wait for the 200 response. This would mean the user would answer and experience a delay (clipping) before ICE completes and media start to flow. Since UA2 would not be aware of the 3PCC situation, this would impact all calls to UA2, not just those that use 3PCC.

3.2.4. Issue an early 200 response to the INVITE request to UA2

UA2 could issue a 200 response instead of a 183 response, even though the user has not yet been alerted and answered. This would be different from normal practice and might adversely impact behaviour at other SIP entities, e.g., charging, logging, forking, call forwarding. Again UA 2 would not be aware of the 3PCC situation, so this would impact all calls to UA2, not just those that use 3PCC.

3.2.5. Use reliable provisional responses

If UA2 and the B2BUA support reliable provisional responses [RFC3262], UA2 can send the 183 response with SDP answer reliably (resulting in a PRACK transaction), and then the B2BUA can send an UPDATE request with the updated offer without waiting for the INVITE transaction to complete. This would seem to work, except that it requires the entities involved to support [RFC3262], which, as explained in section Section 2.2.3, is undesirable. In particular, UA2, which is the "innocent party" in 3PCC, should not be expected to provide special functionality just to make 3PCC work. Furthermore, a B2BUA performing 3PCC would not be aware of ICE and hence the need to support [RFC3262].

4. Do we really need the updated offer?

4.1. Types of devices that rely on the updated offer

Devices on the signalling path that rely on the updated offer are SIP/SDP-aware devices (e.g., policy servers) that perform admission control or resource reservation based on SDP, without modifying the SDP as the signalling messages are forwarded. Devices that modify the SDP (e.g., Session Border Controllers) generally terminate ICE, so are not an issue.

One type of behaviour that relies on the updated offer is a device that is ICE-aware and reserves resources for all the ICE candidate-pairs. Such a device would need to know which candidates have been selected, so that unwanted resources can be freed.

A second type of behaviour that relies on the updated offer is a device that is not ICE-aware and admits traffic on ports identified in the m/c lines of the SDP offer. Such a device is assumed to let a moderate amount of traffic through on other ports, so would not prevent STUN connectivity checks, but would prevent a sustained transmission of RTP. The updated offer/answer would allow such a device to admit sustained traffic on the ports that have been negotiated using ICE.

4.2. Types of environment in which ICE is deployed

ICE has seen a certain amount of deployment. ICE is not solely for use in a SIP/SDP environment, and some of those deployments are in non-SIP/SDP environments (e.g., Jingle). ICE deployment with SIP is relatively sparse in some types of environment, the fundamental reason being that NAT traversal is frequently accomplished by Session Border Controllers. This is true, for example, for most enterprise deployments of SIP. Where such environments are migrated to IPv6, often SBCs are used at the border between IPv4 and IPv6 networks and therefore ICE is not needed for negotiating the IP version. Frequently the types of signalling path device that would require an ICE updated offer are deployed in this sort of environment, where ICE is currently not needed, not deployed and unlikely to be deployed.

On the other hand, the use of SIP across the public Internet without the use of SBCs does require ICE. In such deployments, however, it is unlikely there will be devices on the signalling path that would need an updated offer.

So basically there are environments where SIP is used and ICE is not deployed and not needed. There are also environments where SIP is used and ICE is needed, and to some extent deployed, but such environments generally do not require the ICE updated offer. Some such deployments may not be concerned with fax or with 3PCC, and therefore implementation of the updated offer might not be an issue, although it is believed that some implementations do not support the updated offer and can still operate in their target environments.

In the future, ICE is likely to be required in more environments. The present SBC-based approach in enterprise environments, for example, might not be the most appropriate for use in cloud-based deployments, where it is unrealistic to have media following the signalling path. ICE could be used in such environments, but the presence of signalling path devices that need the updated offer seems unlikely.

4.3. Relaxing the requirement

These considerations bring into question the mandatory requirement in [RFC5245] for an updated SDP offer under some circumstances. This could be relaxed such that it can be omitted in environments where it is not needed.

5. Conclusions

This document illustrates two common use cases where the introduction of ICE can lead to problems with the updated offer/answer cycle that ICE requires in certain circumstances. In the first case (fax), the problem arises at the two endpoints that are trying to accomplish ICE. In the second case (3PCC), the problem arises because of a particular B2BUA behaviour, yet the B2BUA is not involved in ICE, should not need to know anything about ICE, and should not need to implement any extensions to SIP or SDP in order for ICE to work between UAs. In both cases there are work-arounds, but these introduce dependencies that contrive to reduce the chances of successful interoperability.

The need, in some circumstances, to conduct an updated offer/answer cycle on conclusion of ICE is common to both problems. This need arises not from ICE itself, but from the certain types of SIP/SDP-aware devices on the signalling path whose normal functioning is impacted when endpoints use ICE, unless they have been upgraded to cope with the effects of ICE.

The two use cases illustrated might not be the only cases where the ICE updated offer is problematic. As more complex multimedia situations arise, involving mid-call (and in particular early-in-the-call) offer-answer cycles for changing media, changing security, etc., the more likely it is that the additional ICE update offer-answer cycle will intrude in an unhelpful way.

According to discussions in section Section 4, it seems to be the case that the updated offer is needed, in practice, in very few environments, and therefore consideration should be given to relaxing the requirement in [RFC5245].

6. IANA considerations

This document requires no IANA actions.

7. Security considerations

This document does not introduce any new security considerations.

8. 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, June 2002.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

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

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[RFC3312] Camarillo, G., Marshall, W., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312, October 2002.

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

[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.

Authors’ Addresses

John Elwell
Siemens Enterprise Communications

Phone: +44 1908 817801
Email: john.elwell@siemens-enterprise.com

Andy Hutton
Siemens Enterprise Communications

Phone: +44 1908 817920
Email: andrew.hutton@siemens-enterprise.com

Thomas Stach
Siemens Enterprise Communications

Phone: +43 5 7008 4377
Email: thomas.stach@siemens-enterprise.com

MMUSIC WG J. Elwell
Internet-Draft Unaffiliated
Intended status: Informational A. Hutton
Expires: June 14, 2012 T. Stach
Siemens Enterprise
Communications
December 12, 2011

ICE Updated Offer Problematic
draft-elwell-mmusic-ice-updated-offer-02.txt

Abstract

Interactive Connectivity Establishment (ICE) requires an updated offer-answer cycle under some circumstances, to satisfy the needs of some devices on the signalling path. When used with SIP, this additional offer-answer cycle interacts with other things, such as fax and third party call control (3PCC). This document describes the problems and discusses possible remedies.

This work is being discussed on the mmusic@ietf.org mailing list.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Interactive Connectivity Establishment (ICE) [RFC5245] specifies a mechanism for NAT traversal for multimedia sessions established using the Session Description Protocol (SDP) [RFC4566] offer-answer model [RFC3264]. It allows a pair of endpoints to exchange candidate IP addresses and ports, perform checks to see which pairs of candidates work, and agree which pairs to use for a given component of a given medium (e.g., RTP stream, RTCP stream). The mechanism can also be used for IPv6 transition, for determining whether to use IPv4 or IPv6. A particular application of ICE is with the Session Initiation Protocol (SIP) [RFC3261].

Connectivity checks are performed on the media path between candidate pairs. Based on the results of connectivity checks and certain rules, the two endpoints each determine which pair of candidates to use for a given component and can then start exchanging data (e.g., RTP packets) on the agreed path. Further exchanges on the signalling path (i.e., the path on which the offer-answer exchange is performed) are not necessary for the endpoints to agree which candidates to use.

However, certain SIP/SDP-aware devices on the signalling path need to know which candidates have been selected (e.g., to prioritize that traffic or to remove the resources for non-selected candidates). For this reason ICE mandates a further offer-answer exchange in some circumstances, i.e., an updated SDP offer followed by an updated SDP answer. In some situations with SIP, this updated offer-answer exchange can be problematic. This document examines these problems.

2. Fax calls

2.1. Problem statement

Except where dedicated fax devices are involved, fax calls typically start as audio. Detection of CNG tone (calling tone) from the initiating fax machine and CED (called) tone from the receiving fax machine initiates a switch to T.38, i.e., a switch from audio to image. Where the audio call uses a compressed codec (e.g., G.729), if one tone is detected there may first be a switch to G.711, for more reliable tone detection or in case the call turns out to be a non-fax modem call. Thus there can be:

a switch from a compressed codec to G.711; or

a switch from audio to image; or

both in sequence.

Switching codec or switching from audio to image requires an SDP offer-answer cycle. ICE also requires an updated offer-answer cycle where the selected candidates are not those in the m/c-lines of the original offer-answer. If the UA that detects the need to switch because of fax is also the controlling agent from the ICE perspective, updated offer-answer for fax can follow the updated offer-answer for ICE and probably won't lead to problems.

However, if the UA that detects the need to switch because of fax is not the controlling agent from the ICE perspective, there is a significant danger of the two re-INVITE or UPDATE [RFC3311] requests colliding, resulting in a 491 response to each. According to [RFC3261] and [RFC3311], one UA (the one that owns the Call-ID) backs off for between 2.1 and 4 seconds, and the other UA backs off for between 0 and 2 seconds, before trying again. This can result in a delay of up to 4 seconds before the switch to fax, long enough in practice to cause fax calls to fail. It can also result in a delay of up to 4 seconds before the post-ICE updated offer-answer. SIP/SDP-aware devices that need the post-ICE updated offer-answer might not permit the flow of RTP packets throughout this period, which might also lead to failure of the fax call. An example flow is shown below:

```
  UA1 (Call-ID owner)                         UA2 (fax gateway)
    --------------INVITE / SDP offer audio------------------>
    <---------------183 / SDP answer audio-------------------
    <==================ICE negotiation=====================>
    <----------------200 / SDP answer audio -----------------
    ------------------------ACK----------------------------->
    <======================RTP============================>
 (ICE requires updated offer)
    ------UPDATE / SDP offer audio---
                                     \            (Fax detected)
    <-----UPDATE / SDP offer image----\----------------------
                                       \
                                        -------------------->
    -----------------491-----------
                                   \
    <-------------------------------\-----------491----------
                                     \
                                      ---------------------->
                                                (back-off 0-2s)
    <-------------UPDATE / SDP offer image-------------------
    ----------------200 / SDP answer image------------------>
 (back-off 2.1-4s)
    ----UPDATE / SDP offer image (selected candidates)------>
```

```
<----200 / SDP answer image (selected candidates)--------
```

In this example UA1 is the ICE controlling agent and issues an updated offer on completion of ICE, and UA2 is a fax gateway that detects fax and attempts to change to image. UPDATE is supported by both and used for the updated offers. UA1 owns the Call-ID and has the longer back-off. In this example, the switch to image will probably be accomplished fast enough (back-off does not exceed 2 seconds), but the post-ICE updated offer can be delayed up to 4 seconds, perhaps leading to undesirable behaviour by SIP/SDP-aware devices on the signalling path, which might disrupt the flow of RTP and cause the fax call to fail.

Of course, collision of UPDATE or re-INVITE requests will not always occur - it is matter of timing. However, the probability of collision is not insignificant and, if that occurs, the probability of the fax call being adversely affected to the extent that it fails is not insignificant.

2.2. Possible remedies

2.2.1. Delay the ICE updated offer

UA1, as the ICE controlling agent, will be unaware that UA2 will detect fax. Therefore any delay in sending the ICE updated offer will need to apply to all calls and will need to be long enough to allow for differing amounts of time for UA2 to detect fax (perhaps several seconds). The question then is whether this would be long enough to introduce a risk of undesirable behaviour by SIP/SDP-aware devices on the signalling path, which could impact all calls, not just fax calls.

2.2.2. Delay the fax updated offer

UA2 will know that ICE has been used, and therefore can expect an updated offer from UA1, the ICE controlling agent. Normally this should arrive quite quickly (e.g., well under 100 ms), although it depends on the number of SIP intermediaries on the path and whether any retransmissions are needed because of packet loss. Therefore a delay of a 100 ms., say, would probably not impact the fax call and would help avoid collisions but would not be a guarantee.

2.2.3. Use reliable provisional responses and pre-conditions

By using a reliable 183 in accordance with [RFC3262] to send the SDP answer, when ICE completes the updated offer can be sent in an UPDATE request, rather than waiting for the 200 response to the INVITE request and then sending the updated offer. However, the fax machine might auto-answer and send the 200 response to the INVITE request as soon as ICE procedures complete, so the updated offer might collide with the 200 response, again leading to further signalling delays before things are resolved. This in turn could be avoided by using pre-conditions [RFC3312] to delay answering of the call until the updated offer has occurred.

This might work, although it is unclear how pre-conditions are intended to interact with ICE, i.e., whether ICE procedures can continue without waiting for pre-conditions to be satisfied. Perhaps an extension to pre-conditions would be required. Also this might introduce further adverse interactions with SIP/SDP-aware devices on the signalling path.

Even if it could be made to work, this approach would require the entities involved to support [RFC3262] and [RFC3312]. [RFC3262] is known to be rather complicated to implement (hence the reason the ICE mechanism was specifically designed to allow SDP answer to be sent in an unreliable provisional response (ICE provides acknowledgement on the media path, rather than requiring the use of PRACK). Pre-conditions are a further complication and not widely implemented. Therefore ICE implementations should not be expected to support [RFC3262] and [RFC3312].

3. Third party call control (3PCC)

3.1. Problem statement

3PCC [RFC3725] is a common technique used with SIP where calls are controlled from an application at a SIP B2BUA. In particular, calls can be established by 3PCC, whereby the application first makes a call to the first party (typically the device of a user requesting the call) and then makes a second call to the second party, the two calls being joined together such that media flow directly between the two devices. A typical implementation is in accordance with Flow I in [RFC3725]: the controlling B2BUA sends an offerless INVITE request to UA1, which alerts the first user. When the user answers, UA1 sends an offer in a 200 response to the INVITE request, and this offer is used by the B2BUA in a second INVITE request, this time to UA2.

The problem with using ICE with 3PCC is that 3PCC signalling does not permit the updated offer-answer for ICE to occur in a timely manner. UA2 will often take some time (seconds or tens of seconds) before sending the 200 response to its INVITE request. Yet if UA2 has already sent an SDP answer (e.g., in a 183 response), ICE can complete on the media paths and UA1, as the ICE controlling agent, can attempt an updated offer. This updated offer cannot be forwarded to UA2 until the INVITE transaction on that leg of the call has completed.

More specifically, consider the following example flow:

```
  UA1 (Call-ID owner)         B2BUA                         UA2
    <----INVITE (no SDP)------
    -----200 / SDP offer----->     ----INVITE / SDP offer--->
    <----ACK / SDP answer-----     <-----183 / SDP answer----
    <===================ICE negotiation=====================>
(ICE requires updated offer)
    ----UPDATE / SDP offer---> What next?
```

In this case, UA2 sends a 183 provisional response to its INVITE request. This contains an SDP answer, which is passed to UA1 through the ACK request. Thus UA1 and UA2 are able to conduct ICE negotiation on the media paths. UA2 will probably not alert its user until ICE negotiation is complete, but anyway, there will often be a significant delay before the user answers and UA2 sends a 200 response to its INVITE request. Meanwhile, UA1, as the ICE controlling agent, attempts to send an updated offer. In this case it chooses to use an UPDATE request, but similar considerations apply if it uses a re-INVITE request. The B2BUA cannot pass that request on until the INVITE transaction with UA2 has completed. Either the UPDATE request has to be delayed somehow or rejected, in either case leading to the possibility of undesirable behaviour by SIP/SDP-aware devices that require a timely updated offer. For example, UA2 might be transmitting early media, which might fail to be passed through correctly, or clipping might occur when the user answers.

It should be noted that the issue of sending an updated offer in a 3PCC situation before UA2 has answered is not solely an ICE issue. However, ICE substantially increases the need for such an updated offer.

3.2. Possible remedies

3.2.1. Avoid 3PCC

There are alternatives to this form of 3PCC. For example, UA1 could be instructed to issue a conventional INVITE request by sending a SIP REFER request to UA1, or by some non-SIP means. However, using a REFER request is not an option for some types of UA, for example PSTN gateways. If user 1 is a PSTN user, it is necessary to make a PSTN call to the user, and this can be achieved by sending an INVITE request to UA1, but not by sending a REFER request to UA1. Non-SIP means are either not standardized or little deployed.

A particular example of an application that uses 3PCC is one where the user uses a web page to make the call, having selected in advance the device he/she wishes to use to make the call. The application causes the B2BUA to send an INVITE request to that selected device, followed by an INVITE request to the called destination. If the selected device is, for example, a cellular device reachable via PSTN, that initial INVITE request will be sent to a PSTN gateway.

Because of the difficulties supporting such applications by other means, 3PCC is a commonly deployed technique. It is not possible to scrap 3PCC in order to introduce ICE.

3.2.2. Delay the updated offer

UA1 will typically not be aware of the state of the INVITE transaction to UA2, and will issue the updated offer in an UPDATE or re-INVITE request without knowing whether the B2BUA can pass it on. Therefore the onus is on the B2BUA to handle the situation when it receives the UPDATE or re-INVITE request. As a non-INVITE transaction, an UPDATE request has a relatively short timeout, but one possibility would be for the B2BUA to reject it with a 500 response and a Retry-After header field, relying on UA1 to try again later. In the case of re-INVITE, the B2BUA could delay forwarding the request to UA2 until the original transaction is complete. However, in either case, SIP/SDP-aware devices between the B2BUA and UA2 will not see the updated offer in a timely manner, and therefore might take action that prevents the correct handling of early media or clips media for a short time after the call is answered.

3.2.3. Delay ICE until UA2 answers

UA2 could delay ICE until UA2 answers, which means UA2 would not need to send SDP answer in a provisional response but could wait for the 200 response. This would mean the user would answer and experience a delay (clipping) before ICE completes and media start to flow. Since UA2 would not be aware of the 3PCC situation, this would impact all calls to UA2, not just those that use 3PCC.

3.2.4. Issue an early 200 response to the INVITE request to UA2

UA2 could issue a 200 response instead of a 183 response, even though the user has not yet been alerted and answered. This would be different from normal practice and might adversely impact behaviour at other SIP entities, e.g., charging, logging, forking, call forwarding. Again UA 2 would not be aware of the 3PCC situation, so this would impact all calls to UA2, not just those that use 3PCC.

3.2.5. Use reliable provisional responses

If UA2 and the B2BUA support reliable provisional responses [RFC3262], UA2 can send the 183 response with SDP answer reliably (resulting in a PRACK transaction), and then the B2BUA can send an UPDATE request with the updated offer without waiting for the INVITE transaction to complete. This would seem to work, except that it requires the entities involved to support [RFC3262], which, as explained in section Section 2.2.3, is undesirable. In particular, UA2, which is the "innocent party" in 3PCC, should not be expected to provide special functionality just to make 3PCC work. Furthermore, a B2BUA performing 3PCC would not be aware of ICE and hence the need to support [RFC3262].

4. Do we really need the updated offer?

4.1. Types of devices that rely on the updated offer

Devices on the signalling path that rely on the updated offer are SIP/SDP-aware devices (e.g., policy servers) that perform admission control or resource reservation based on SDP, without modifying the SDP as the signalling messages are forwarded. Devices that modify the SDP (e.g., Session Border Controllers) generally terminate ICE, so are not an issue.

One type of behaviour that relies on the updated offer is a device that is ICE-aware and reserves resources for all the ICE candidate-pairs. Such a device would need to know which candidates have been selected, so that unwanted resources can be freed.

A second type of behaviour that relies on the updated offer is a device that is not ICE-aware and admits traffic on ports identified in the m/c lines of the SDP offer. Such a device is assumed to let a moderate amount of traffic through on other ports, so would not prevent STUN connectivity checks, but would prevent a sustained transmission of RTP. The updated offer/answer would allow such a device to admit sustained traffic on the ports that have been negotiated using ICE.

4.2. Types of environment in which ICE is deployed

ICE has seen a certain amount of deployment. ICE is not solely for use in a SIP/SDP environment, and some of those deployments are in non-SIP/SDP environments (e.g., Jingle). ICE deployment with SIP is relatively sparse in some types of environment, the fundamental reason being that NAT traversal is frequently accomplished by Session Border Controllers. This is true, for example, for most enterprise deployments of SIP. Where such environments are migrated to IPv6, often SBCs are used at the border between IPv4 and IPv6 networks and therefore ICE is not needed for negotiating the IP version. Frequently the types of signalling path device that would require an ICE updated offer are deployed in this sort of environment, where ICE is currently not needed, not deployed and unlikely to be deployed.

On the other hand, the use of SIP across the public Internet without the use of SBCs does require ICE. In such deployments, however, it is unlikely there will be devices on the signalling path that would need an updated offer.

So basically there are environments where SIP is used and ICE is not deployed and not needed. There are also environments where SIP is used and ICE is needed, and to some extent deployed, but such environments generally do not require the ICE updated offer. Some such deployments may not be concerned with fax or with 3PCC, and therefore implementation of the updated offer might not be an issue, although it is believed that some implementations do not support the updated offer and can still operate in their target environments.

In the future, ICE is likely to be required in more environments. The present SBC-based approach in enterprise environments, for example, might not be the most appropriate for use in cloud-based deployments, where it is unrealistic to have media following the signalling path. ICE could be used in such environments, but the presence of signalling path devices that need the updated offer seems unlikely.

4.3. Relaxing the requirement

These considerations bring into question the mandatory requirement in [RFC5245] for an updated SDP offer under some circumstances. This could be relaxed such that it can be omitted in environments where it is not needed.

5. Conclusions

This document illustrates two common use cases where the introduction of ICE can lead to problems with the updated offer/answer cycle that ICE requires in certain circumstances. In the first case (fax), the problem arises at the two endpoints that are trying to accomplish ICE. In the second case (3PCC), the problem arises because of a particular B2BUA behaviour, yet the B2BUA is not involved in ICE, should not need to know anything about ICE, and should not need to implement any extensions to SIP or SDP in order for ICE to work between UAs. In both cases there are work-arounds, but these introduce dependencies that contrive to reduce the chances of successful interoperability.

The need, in some circumstances, to conduct an updated offer/answer cycle on conclusion of ICE is common to both problems. This need arises not from ICE itself, but from the certain types of SIP/SDP-aware devices on the signalling path whose normal functioning is impacted when endpoints use ICE, unless they have been upgraded to cope with the effects of ICE.

The two use cases illustrated might not be the only cases where the ICE updated offer is problematic. As more complex multimedia situations arise, involving mid-call (and in particular early-in-the-call) offer-answer cycles for changing media, changing security, etc., the more likely it is that the additional ICE update offer-answer cycle will intrude in an unhelpful way.

According to discussions in section Section 4, it seems to be the case that the updated offer is needed, in practice, in very few environments, and therefore consideration should be given to relaxing the requirement in [RFC5245].

6. IANA considerations

This document requires no IANA actions.

7. Security considerations

This document does not introduce any new security considerations.

8. 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, June 2002.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

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

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[RFC3312] Camarillo, G., Marshall, W., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312, October 2002.

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

[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.

Authors' Addresses

John Elwell
Unaffiliated

Email: john.r.elwell@gmail.com

Andrew Hutton
Siemens Enterprise Communications

Phone: +44 1908 817920
Email: andrew.hutton@siemens-enterprise.com

Thomas Stach
Siemens Enterprise Communications

Phone: +43 5 7008 4377
Email: thomas.stach@siemens-enterprise.com

mmusic
Internet-Draft
Intended status: Standards Track
Expires: November 20, 2011

B. Greevenbosch
Y. Hui
Huawei Technologies

May 19, 2011

SDP attribute to signal parallax
draft-greevenbosch-mmusic-parallax-attribute-01

Abstract

This document introduces a "ParallaxInfo" attribute in SDP. This attribute can be used in stereoscopic applications, to signal the depth positioning of 2D media data within the 3D space.

Note

Discussion and suggestions for improvement are requested, and should be sent to mmusic@ietf.org.

Status of this Memo

This Internet-Draft will expire on November 20, 2011.

Copyright Notice

Table of Contents

1. Introduction

To see a 3D scene, the human brain interprets two different views as perceived by the left and right eyes, and fuses these views into a single 3D perception. The depth of the object is perceived by interpreting the horizontal shift of that object between the views. This shift is called "parallax".

In stereoscopic 3D multimedia applications, there are various ways to transmit media streams in 3D. One way is to transmit two different streams, one for the left eye and one for the right eye. These streams are then projected to the relevant eyes using the appropriate technology.

When sending text streams in 3D, the solution mentioned above would imply sending the same text stream twice. Since the two streams would only differ in horizontal positioning, this introduces a lot of unnecessary overhead.

This document specifies a "ParallaxInfo" attribute in SDP [RFC4566], which is used to transfer the parallax information. It eliminates the need to send two different streams separately, as they can be calculated from a single stream and the "ParallaxInfo" attribute value.

The attribute transfers this information as two parameters: one indicating which view (left/right/centre) is carried by the stream, and another to signal the parallax between the objects.

The attribute is not restricted for signalling the parallax of text streams, but it can also be used to place other 2D objects in the 3D space. Examples include a channel logo, an electronic programme guide and on-screen display of an audio volume indicator.

2. Requirements notation

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].

3. Definitions

L-view
: A visual entity that is to be projected to the left eye. In the case of video, the L-view is a video frame designated for the left eye. In the case of text, the L-view is the text positioned for viewing by the left eye.

L-stream
: A sequence of L-views, which is streamed to the device.

R-view
: A visual entity that is to be projected to the right eye. In the case of video, the R-view is a video frame designated for the right eye. In the case of text, the R-view is the text positioned for viewing by the right eye.

R-stream
: A sequence of R-views, which is streamed to the device.

C-view
: The centre view: a visual entity as seen from a viewpoint between the left and right eyes. The C-view can be used to calculate the L- and R-views.

C-stream
: A sequence of C-views, which is streamed to the device.

stereoscopic device
: A device that is able to produce and display different images for the left and right eyes, such that the viewer can experience a 3D view.

2D device
: A device that is not able to produce and display different images for the left and right eyes.

2D media stream
: A sequence of two dimensional visual entities (such as text or 2D graphics), which is streamed to the device.

4. The ParallaxInfo attribute

The SDP attribute "ParallaxInfo" is used to transmit the depth positioning of 2D media data, such as a text stream, in the 3D space.

The attribute has the following syntax:

```
a=ParallaxInfo:<transmitted position> <parallax>
```

The <transmitted position> indicates whether the L-, C- or R-stream is transmitted, whereas <parallax> indicates the parallax (i.e. shift) between corresponding L- and R-views in pixels, normalised to a screen width of 11520 pixels. To convert the value of <parallax> to match the display video screen width W, it has to be divided by a factor F=11520/W.

The <transmitted position> can have the following values:

"L" indicates that the transmitted stream is the L-stream. A stereoscopic device MUST calculate the corresponding R-views by shifting the L-views <parallax>/F pixels towards the right.

"C" indicates that the transmitted stream is the C-stream. A stereoscopic device MUST calculate the corresponding L-views by shifting the C-views <parallax>/(2*F) pixels towards the left, and the R-views by shifting the C-views <parallax>/(2*F) pixels towards the right. <parallax> SHOULD be even. If it is odd, the divided value MUST be rounded off towards zero.

"R" indicates that the transmitted stream is the R-stream. A stereoscopic device MUST calculate the corresponding L-views by shifting the R-views <parallax>/F pixels towards the left.

<parallax> MAY be negative. In this case, the shift direction is reversed.

The "ParallaxInfo" attribute can be a session-level attribute or a media-level attribute. As a session-level attribute, it specifies the default parallax value which can be applied to all the 2D media streams in the session being described. As a media-level attribute, it specifies the parallax value which can be applied to the associated 2D media stream, overriding any session-level parallax value specified.

The stereoscopic device MAY use the session-level attribute value for on-screen display, for example an audio volume indicaton, channel indication or electronic programme guide.

Notice that a 2D device that does not support the "ParallaxInfo" attribute will ignore it, and therefore display the 2D media data on the position as transmitted.

5. Example

The following is an example of an SDP description of a session with an audio stream, a video stream and a 3GPP timed text stream (see [3gpp-tt]), streamed using RTP as per [RFC4396]. If the display resolution is 1280x720, the parallax is scaled down with a factor F=11952/1280=9. The transmitted text stream is the L-stream, and with the example display resolution each R-view is 144/9=16 pixels on the left of the L-view. This corresponds to the virtual positioning of the text in front of the screen.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=ParallaxInfo:L -180
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
m=video 52888 RTP/AVP 97
a=rtpmap:97 3gpp-tt/1000
a=ParallaxInfo:L -144
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

Notice that the default value "-180" is overridden by the value "-144" for the text stream. However the "-180" value is still signalled for on-screen display of e.g. volume control and other 2D graphics.

In case each R-view is 24 pixels (216 pixels in the reference resolution) on the right of the associated L-view, i.e. the virtual positioning of the text is behind the screen, then the three lines defined for 3gpp-tt can be replaced as follows:

```
m=video 52888 RTP/AVP 97
a=rtpmap:97 3gpp-tt/1000
a=ParallaxInfo:L 216
```

6. Remarks

A positive parallax value indicates virtual positioning of the 2D media data behind the screen. This is the case when the objects in the L-view are on the left of the same objects in the R-view. Similarly, a negative parallax value indicates that the objects in the R-view are on the left of the same objects in the L-view, and corresponds to virtual positioning in front of the screen.

Since the "ParallaxInfo" attribute indicates a shift of the transmitted stream, it might happen that the L- or R-view trespasses the bounderies of the display. In this case, clipping is necessary, as illustrated by Figure 1.

Similarly, there are areas which are covered by the L-view but not by the R-view and vice versa. These areas need to be filled in a sensible way. Since the "ParallaxInfo" attribute is designed for objects that overlay other video data (e.g. subtitles), it is trivial to fill in uncovered areas by using the underlying video data. However, if there is no underlying video data, other mechanisms to fill in the uncovered areas need to be defined. Definition of these mechanisms are out of scope of this document.

```
      +-----------------------+          +-----------------------+
      |SCREEN       .         |          |SCREEN       .         |
      |             .         |          |             .         |
      |    +-------------------+         |  +-------------------+|
      |    |TEXT     .         |         |  |TEXT     .         ||
      |    +-------------------+         |  +-------------------+|
      |             .         |          |             .         |
      |             .         |          |             .         |
      +-----------------------+          +-----------------------+

                           |
                           | CLIP
                           V

      +-----------------------+          +-----------------------+
      |SCREEN       .         |          |SCREEN       .         |
      |             .         |          |             .         |
      |    +------------------+          |  +-------------------+|
      |    |TEXT     .        |          |  |TEXT     .         ||
      |    +------------------+          |  +-------------------+|
      |             .         |          |             .         |
      |             .         |          |             .         |
      +-----------------------+          +-----------------------+
```

Figure 1

The normalisation of the parallax towards a reference screen width of 11520 pixels was chosen to ensure correct display of the same stream on different video display resolutions. This is especially useful when the underlying video is coded with a scalable video codec, which does not have a fixed video resolution.

7. Security Considerations

The authors foresee no security issues in addition to those already listed in [RFC4566].

8. IANA Considerations

Following the guidelines in [RFC4566], the SDP attribute has to be registered at IANA:

- o Contact name/email: authors of this RFC

- o Attribute name: ParallaxInfo

- o Long-form attribute name: Parallax info for the depth positioning of 2D media data in the 3D space

- o Type of attribute: session level and media level

- o Subject to charset: no

This attribute is used to signal how 2D media data is to be displayed in the 3D space. It indicates the shift of the respective left and right views.

The attribute has the following ABNF (see [RFC4234]) description:

```
ParalaxInfo          = "a=ParallaxInfo:" TransmittedPosition Parallax
TransmittedPosition  = "C"/"L"/"R"
Parallax             = num-val
```

The attribute transfers this information as two parameters: "TransmittedPosition" indicates which view of the 2D media data (left "L", right "R" or centre "C") is carried by the stream, and "Parallax" signals the parallax (in pixels) of objects in the 2D media stream.

9. Normative References

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

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4396] Rey, J. and Y. Matsui, "RTP Payload Format for 3rd Generation Partnership Project (3GPP) Timed Text", RFC 4396, February 2006.

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

[3gpp-tt] 3GPP, "Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs (Release 5)", TS 26.234 v5.3.0, December 2002.

Authors' Addresses

Bert Greevenbosch
Huawei Technologies Co., Ltd.
Huawei Industrial Base
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone: +86-755-28978088
Email: bgreeven@huawei.com

Hui Yu
Huawei Technologies Co., Ltd.
Huawei Nanjing R&D Center
101 Software Avenue
Yuhuatai District
Nanjing 210012
P.R. China

Phone: +86-25-56620323
Email: huiyu@huawei.com

mmusic B. Greevenbosch
Internet-Draft Y. Hui
Intended status: Standards Track Huawei
Expires: November 20, 2011 May 19, 2011

Signal 3D format
draft-greevenbosch-mmusic-signal-3d-format-01

Abstract

This document introduces the SDP attribute "3dFormat", which provides format description of stereoscopic 3D video. In addition, the grouping mechanism for SDP is extended to cater for stereoscopic 3D video.

Note

Discussion and suggestions for improvement are requested, and should be sent to mmusic@ietf.org.

Status of this Memo

This Internet-Draft will expire on November 20, 2011.

Copyright Notice

Table of Contents

1. Introduction

In stereoscopic 3D multimedia applications, two views are displayed, one for the left eye and one for the right eye.

There are various ways of formatting the views of Stereoscopic 3D video. Examples of 3D formats are frame packing (see [HDMIv1.4a]) and the combination of 2D video and auxiliary data such as depth maps (see [ISO/IEC 23002-3]). Stereoscopic 3D video may be carried over a single stream or over several streams, depending on its 3D format.

In multimedia streaming applications, the Session Description Protocol (SDP) [RFC4566] can be used to provide to the receiver sufficient information about the media streams, and to enable the receiver to join and participate in the session.

This document defines an extension to SDP that provides sufficient information about the format of stereoscopic 3D video carried in the media stream(s). Before accessing the stream(s), the receiver can use the 3D format description from SDP to determine whether it has the capability to receive and render the stereoscopic 3D video content, and whether it can participate in the session.

The mentioned SDP extension is a new SDP attribute "3dFormat", which provides the format description of stereoscopic 3D video. The design of the attribute is based on the following requirements, which are listed only for informational purposes:

o It MUST be possible to signal that the left and right views are carried in a single stream, by the use of frame packing.

o It MUST be possible to signal that 2D video and auxiliary video (such as depth maps) are carried in a single stream.

o It MUST be possible to signal that the left and right views are carried in two separate streams.

o It MUST be possible to signal that 2D video and auxiliary video (such as depth maps) are carried in separate streams.

To bind multiple video streams that carry a single stereoscopic 3D video, this document also extends the SDP grouping mechanism from [RFC5888].

2. Requirements notation

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

3. Definitions

2D video
Video that does not in itself contain depth or parallax information.

auxiliary video
A sequence of depth or parallax maps, which are used to add depth to 2D video.

C-view
The centre view: a visual entity as seen from a viewpoint between the left and right eyes. The C-view can be used to calculate the L- and R-views.

C-stream
A 2D video stream consisting of a sequence of C-views.

depth map
A two dimensional map, each pixel of which defines the depth of one or more pixels in an associated 2D video frame.

depth map stream
An auxiliary stream, which contains a sequence of depth maps. The depth map stream is synchronised with the associated 2D video stream.

frame packing
A format that packs the L- and R-views into a single 2D video stream. The packing may be done spatially, where each video frame is divided into sub-frames, one containing the L-view and one containing the R-view. The packing can also be done sequentially, where alternating video frames represent L- and R-views.

legacy answerer
An answerer (in the SDP offer/answer model [RFC3264]) that does not support the "3dFormat" attribute. The legacy answerer can be the streaming server or the streaming client, but is not compliant to this document.

L-view
A visual entity that is to be projected to the left eye.

L-stream
A 2D video stream consisting of a sequence of L-views.

parallax map
A two dimensional map, each pixel of which defines the parallax of one or more pixels in an associated 2D video frame.

parallax map stream
An auxiliary stream, which contains a sequence of parallax maps. The parallax map stream is synchronised with the associated 2D video stream.

R-view
A visual entity that is to be projected to the right eye.

R-stream
A 2D video stream consisting of a sequence of R-views.

stereoscopic 3D video
The L- and R-streams, ready to be projected to the viewer’s left and right eyes.

sub-frame
A part of a video frame.

4. The "3dFormat" attribute

The media-level SDP attribute "3dFormat" signals the format of stereoscopic 3D video. The attribute transfers this information through two parameters: one indicating the format type of the stereoscopic 3D video carried in the media stream(s), and the other indicating the type of the video component, which is a constituent element of the stereoscopic 3D video. The video component type depends on the format type of the stereoscopic 3D video. The syntax of the attribute is defined as follows:

```
a=3dFormat:<Format Type> <Component Type>
```

The <Format Type> can have the following values (as indicated between the quotes):

"FP" Frame Packing
: The L- and R-views are packed into a single stream. The packing may use a side-by-side, top-and-bottom, interleaved, checkerboard or frame sequential format.

"SC" Simulcast
: The L- and R-streams are transmitted separately.

"2DA" 2D + auxiliary
: 2D video and auxiliary data (such as depth maps or parallax maps) are transmitted. These can be transmitted in a single stream, as well as in two separate streams.

The <Component Type> can have the following values (as indicated between the quotes):

"C" Centre view
: The associated stream is a C-stream.

"CD" centre view and depth map
: The associated stream contains both the C-view and depth map sequences.

"ChB" Checkerboard
: The video frame consists of alternating pixels from the corresponding L- and R-views, as illustrated by Figure 1.

"CP" Centre view and parallax map
: The associated stream contains both the C-view and parallax map sequences.

"D" Depth map
The associated stream is a sequence of depth maps.

"L" Left view
The associated stream is the L-stream.

"LD" Left view and depth map
The associated stream contains both the L-view and depth map sequences.

"LIL" Line Interleaved
Each video frame consists of alternating scan lines from the L- and R-views.

"LP" Left view and parallax map
The associated stream contains both the L-view and parallax map sequences.

"P" Parallax map
The associated stream is a sequence of parallax maps.

"R" Right view
The associated stream is the R-stream.

"SbS" Side by Side
Each video frame is divided in two equally sized sub-frames, spatially positioned side by side of each other. One sub-frame contains the L-view, whereas the other contains the R-view.

"Seq" Frame sequential
The single video stream consists of alternating frames from the L- and R-streams.

"TaB" Top and Bottom
Each video frame is divided in two equally sized sub-frames, spatially positioned above each other. One sub-frame contains the L-view, whereas the other contains the R-view.

```
+-+-+-+-+-+-+
|L|R|L|R|L|R|
+-+-+-+-+-+-+
|R|L|R|L|R|L|
+-+-+-+-+-+-+
|L|R|L|R|L|R|
+-+-+-+-+-+-+
```

The checkerboard pattern. The transmitted video frame is composed of pixels from the L- and R-views. Samples from the L-view are indicated with "L", whereas samples from the R-view are indicated with "R".

Figure 1

5. Grouping

When multiple streams carry a single stereoscopic 3D video, (e.g. C-stream and parallax map, or separately transmitted L- and R-streams), the grouping mechanism from [RFC5888] MUST be used.

However, to cater for the special requirements of 3D signalling, the semantics are expanded:

```
group-attribute     = "a=group:" semantics *(SP identification-tag)
semantics           = "LS" / "FID" / "3DS" / semantics-extension
semantics-extension = token
```

The grouping is needed when multiple streams carry a single stereoscopic 3D video. This is the case when the <format type> is "SC", or the <format type> is "2DA" and the 2D video and auxiliary data are transmitted as multiple streams. A group with the "3Ds" semantics is called a "3DS group".

A 3DS group MUST NOT contain data that is (potentially) inconsistent with other data in the 3DS group:

- o A 3DS group MUST NOT contain both a parallax map stream and a depth map stream.
- o A 3DS group MUST NOT contain more than one parallax map stream.
- o A 3DS group MUST NOT contain more than one depth map stream.
- o A 3DS group MUST contain at least one 2D video stream.
- o If a 3GS group contains an L- and an R-stream, it MUST NOT contain a depth map or a parallax map.
- o If a 3DS group contains only one 2D video stream, it MUST also contain a parallax map stream or a depth map stream.
- o If a 3DS group contains a parallax map stream or a depth map stream, it MUST also contain a 2D video stream.

6. Combinations of attribute values and group usage

The following table summarises the possible combinations of attribute values and grouping:

	FP	SC	2DA
C			D/P,3DS
CD			T
ChB	T		
CP			T
D			C/L,3DS
L		R,3DS	D/P,3DS
LD			T
LIL	T		
LP			T
P			C/L,3DS
R		L,3DS	
SbS	T		
Seq	T		
TaB	T		

The table is to be read as follows:

- o The columns indicate <Format Type> values, whereas the rows indicate <Component Type> values.
- o For one particular column, we denote the <Format Type> value by "FT" and the <Component Type> value by "CT".
- o When an entry in the table is empty, it means that the corresponding combination of FT and CT is not allowed.

- When an entry in the table contains a single <Component Type> value CTsec, it means that another stream with the <Component Type> value CTsec and the same <Format Type> value FT is needed.

- When multiple <Component Type> values are listed, separated by a "/" symbol, only one secondary stream is needed, which must have one of the listed <Component Type> values, and the same <Format Type> value FT.

- When an entry contains "3DS", it means that a 3DS group is needed.

- When an entry in the table contains the letter "T" (true), it means that the corresponding combination FT and CT is allowed, that there is no required secondary stream, and that a 3DS group is not needed.

7. SDP offer/answer with 3D support

This section describes how the SDP offer/answer model (see [RFC3264]) can be used to negotiate the 3D format. It is assumed that both offerer and answerer are compliant to this document. The case where the answerer is a legacy answerer is described in Section 8.

An example where the SDP offer/answer model can be used to negotiate the 3D format, is the case where the offerer offers two representations of the same stereoscopic 3D video: one frame packed and one as L/R simulcast. In this case, the answerer can select the format of its preference, according to its capabilities or as a trade-off between bandwidth and video quality.

There may also be cases where the answerer prefers to receive a 2D version, even when it supports stereoscopic 3D video and the "3dFormat" attribute. For example, this might happen when the user prefers to watch without glasses this time.

The following statements apply for the answerer:

- o The answerer MUST NOT omit the "3dFormat" attribute for the accepted streams. The answerer MAY omit the "3dFormat" attribute for the rejected streams.

- o The answerer MUST NOT change the value of the "3dFormat" attribute. This means, that the answerer can only choose between the 3D formats advertised in the offer.

- o In case the offer contains simulcast of the L- and R-view, the answerer MAY choose just one view. In this case, it MUST select only that view. This means that the port number of unselected view MUST be set to zero in the answer.

- o In case the offer contains a 2D stream and an auxiliary stream as separate streams, the answerer MAY choose only the 2D stream. In this case, it MUST select the 2D stream, and MUST NOT select the auxiliary stream. This means that the port number of the auxiliary stream MUST be set to zero in the answer.

- o In case the offer contains a 2D stream and an auxiliary stream as a single stream, the answerer MAY choose to reject the stream by setting the port number in the answer to zero.

- o In case of frame packing, if the answerer prefers not to have frame packing, it MUST reject the stream by setting the port number in the answer to zero.

o The answerer SHOULD select only one 3D format. If the answerer has no preference on the offered 3D formats, it is RECOMMENDED that it selects the one that is first listed in the offer.

o If the answerer selects multiple 3D formats, it MUST be prepared to send/receive (depending on whether it is a streaming server or client or both) associated streams simultaneoulsy.

The following statements apply for the offerer:

o The offerer MUST check if the "3dFormat" attribute is included in the answer. If it is not, it SHOULD handle the answer as described in Section 8.

o The offerer SHOULD list the 3D formats in order of preference.

o When multiple 3D formats are selected, the offerer MAY initiate all associated streams. Alternatively, it MAY update its offer with a reduced number of 3D formats.

o If all 3D formats have been rejected, the offerer MAY issue a new offer with 2D video instead.

o If only an auxiliary stream is selected in the answer, the offerer SHOULD update its offer with only the associated 2D video stream. Alternatively, it MAY update its offer advertising another 3D format.

8. SDP offer/answer without 3D support

The legacy answerer may reject the offer. For example, with SIP (see [RFC3261]), it could send an "SIP 488 Not acceptable here" error with a "306 Attribute not understood" warning code. In this case the offerer MAY send a new offer with only a 2D video stream.

On the other hand, it is also possible that the legacy answerer accepts the offer and omits the "3dFormat" attribute in the answer. In this case the "3dFormat" attribute is omitted in the answer but the associated stream was accepted (see [RFC3264]), the offerer is able to deduct that the answerer is a legacy answerer without 3D support. In the following subsections, we describe what the offerer still can do to provide a good user experience with a legacy answerer, for each of the 3D format styles. We assume that the offer was accepted, but a legacy answerer was detected.

8.1. Frame packing

In case the original offer contains frame packing, and the answer does not contain the "3dFormat" attribute, the offerer MAY update the offer with a 2D stream. If the offerer is the streaming server, it MAY choose to stream the frame packed video as it is.

8.2. 2D and auxiliary as a single stream

If the original offer contains a 2D video and an auxiliary video in a single stream, and the answer does not contain the "3dFormat" attribute, the offerer MAY update its offer by offering only a 2D video stream.

8.3. 2D and auxiliary as two separate streams

When the offerer sends an offer to a legacy answerer, and the offer contains a 2D video and an auxiliary video in two separate streams, there are the following possibilities:

- o If the answerer selects only the 2D video steam then 2D video streaming can be done as agreed.
- o If the answerer selects only the auxiliary video, the offerer SHOULD update its offer without the auxiliary video.
- o If the answerer selects both video streams, but omits the "3dFormat" attribute, the offerer MAY update its offer without the auxiliary video.

In case the offerer updates its offer by setting the port for auxiliary video to zero, it MUST NOT include the "3dFormat" attribute or use "3DS" grouping for the 2D stream.

8.4. Simulcast of L- and R-views

When the offerer sends an offer to simulcast the L- and R-view to the legacy answerer, we have the following possibilities:

- o If the answerer selects only one video stream, the offerer MAY stream the 2D video as agreed.

- o If the answerer selects both video streams, but omits the "3dFormat" attribute, the offerer MAY update its offer with only the L- or the R-stream.

In case the offerer updates its offer with only the L- or R-stream by setting one of the ports to zero, it MUST NOT include the "3dFormat" attribute or use "3DS" grouping for the offered stream.

9. Examples

9.1. One single frame compatible stream

The following is an example of an SDP description of a session which contains a single stream, in which the L- and R-streams are packed, in side by side fashion.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:FP SbS
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

9.2. Two separate streams

The following is an example of an SDP description of a session with an audio stream, an L-stream and an R-stream.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:SC L
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:SC R
a=mid:2
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

9.3. C-stream and depth map stream

The following is an example of an SDP description of a session with an audio stream, a C-stream and a depth map stream.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:2DA C
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:2DA D
a=mid:2
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

9.4. Stereoscopic 3D video with two different formats

In the following example, there are two different formats for stereoscopic 3D video. One consists of stream 1 (C-stream) and stream 2 (parallax map stream), whereas the other consists of stream 3 (L-stream) and stream 4 (R-stream). There also is an audio stream, which can be used with both formats.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
a=group:3DS 3 4
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:2DA C
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:2DA P
a=mid:2
m=video 49174 RTP/AVP 103
a=rtpmap:103 H264/90000
a=3dFormat:SC L
a=mid:3
m=video 49176 RTP/AVP 105
a=rtpmap:105 H264/90000
a=3dFormat:SC R
a=mid:4
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

10. Formal ABNF grammar of the "3dFormat" attribute

This section contains the formal ABNF grammar of the "3dFormat" attribute.

```
3dFormat-attribute      = "a=3dFormat:" formatType componentType
formatType              = "FP"/"SC"/"2DA"/formatType-extension
formatType-extension    = token
componentType           = "C"/"CD"/"ChB"/"CP"/"D"/"L"/"LD"/
                          "LIL"/"LP"/"P"/"R"/"SbS"/"Seq"/"TaB"/
                          componentType-extension
componentType-extension = token
```

11. Security Considerations

The authors foresee no security issues in addition to those already listed in [RFC4566].

12. IANA Considerations

12.1. "3dFormat" attribute

Following the guidelines in [RFC4566], the SDP attribute has to be registered at IANA:

- o Contact name/email: authors of this RFC
- o Attribute name: 3dFormat
- o Long-form attribute name: Attribute for signalling the format of a stereoscopic 3D video carried in the media stream(s).
- o Type of attribute: media level
- o Subject to charset: no

The "3dFormat" SDP media-level attribute is used to signal the format of stereoscopic 3D video, carried in one or more media stream(s).

The attribute has the following syntax:

```
a=3dFormat:<Format Type> <Component Type>
```

The <Format Type> indicates the format type of the stereoscopic 3D video carried in the media stream(s). It indicates whether the stereoscopic 3D video is frame packed, simulcast or consists of a 2D video stream and an auxiliary stream. The <Format Type> can have the following values (as indicated between the quotes):

```
"FP"     frame packed
"SC"     simulcast
"2DA"    2D + auxiliary
```

The <Component Type> indicates the type of the video component, which is a constituent element of the stereoscopic 3D video. It can have the following values:

"C"	centre view
"CD"	centre view and depth map
"ChB"	checkerboard
"CP"	centre view and parallax map
"D"	depth map
"L"	left view
"LD"	left view and depth map
"LIL"	line interleaved
"LP"	left view and parallax map
"P"	parallax map
"R"	right view
"SbS"	side by side
"Seq"	frame sequential
"TaB"	top and bottom

12.2. "3DS" value for "group" semantics

Following the standards action policy from [RFC5226], the following semantics have to be registered with IANA in the "Semantics for the "group" SDP Attribute" registry under "SDP Parameters":

Semantics	Token	Reference
3D synchronised	3DS	this RFC

13. Normative References

[HDMIv1.4a]
HDMI, "HDMI Specification Version 1.4a", March 2010.

[ISO/IEC 23002-3]
MPEG, "MPEG video technologies part 3: Representation of auxiliary video and supplemental information", ISO/IEC FDIS 23002-3:2007(E), December 2002.

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

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[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.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

Authors' Addresses

Bert Greevenbosch
Huawei Technologies Co., Ltd.
Huawei Industrial Base
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone: +86-755-28978088
Email: bgreeven@huawei.com

Hui Yu
Huawei Technologies Co., Ltd.
Huawei Nanjing R&D Center
101 Software Avenue
Yuhuatai District
Nanjing 210012
P.R. China

Phone: +86-25-56620323
Email: huiyu@huawei.com

mmusic
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012
B. Greevenbosch
Y. Hui
Huawei
October 31, 2011

Signal 3D format
draft-greevenbosch-mmusic-signal-3d-format-02

Abstract

This document introduces the SDP attribute "3dFormat", which provides format description of stereoscopic 3D video. In addition, the grouping mechanism for SDP is extended to cater for stereoscopic 3D video.

Note

Discussion and suggestions for improvement are requested, and should be sent to mmusic@ietf.org.

Status of this Memo

This Internet-Draft will expire on May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

In stereoscopic 3D multimedia applications, two views are displayed, one for the left eye and one for the right eye.

There are various ways of formatting the views of Stereoscopic 3D video. Examples of 3D formats are frame packing (see [HDMIv1.4a] and [ISO/IEC 14496-10]) and the combination of 2D video and auxiliary data such as depth maps or parallax maps (for both, see [ISO/IEC 23002-3]). Stereoscopic 3D video may be carried over a single stream or over several streams, depending on its 3D format.

In multimedia streaming applications, the Session Description Protocol (SDP) [RFC4566] can be used to provide to the receiver sufficient information about the media streams, and to enable the receiver to join and participate in the session.

This document defines an extension to SDP that provides sufficient information about the format of stereoscopic 3D video carried in the media stream(s). Before accessing the stream(s), the receiver can use the 3D format description from SDP to determine whether it has the capability to receive and render the stereoscopic 3D video content, and whether it can participate in the session.

The mentioned SDP extension is a new SDP attribute "3dFormat", which provides the format description of stereoscopic 3D video. The design of the attribute is based on the following requirements, which are listed only for informational purposes:

o It MUST be possible to signal that the left and right views are carried in a single stream, by the use of frame packing.

o It MUST be possible to signal that 2D video and auxiliary video (such as depth maps) are carried in a single stream.

o It MUST be possible to signal that the left and right views are carried in two separate streams.

o It MUST be possible to signal that 2D video and auxiliary video (such as depth maps) are carried in separate streams.

To bind multiple video streams that carry a single stereoscopic 3D video, this document also extends the SDP grouping mechanism from [RFC5888].

2. Requirements notation

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].

3. Definitions

2D video
Video that does not in itself contain depth or parallax information.

auxiliary video
A sequence of depth or parallax maps, which are used to add depth to 2D video.

C-view
The centre view: a visual entity as seen from a viewpoint between the left and right eyes. The C-view can be used to calculate the L- and R-views.

C-stream
A 2D video stream consisting of a sequence of C-views.

depth map
A two dimensional map, each pixel of which defines the depth of one or more pixels in an associated 2D video frame.

depth map stream
An auxiliary stream, which contains a sequence of depth maps. The depth map stream is synchronised with the associated 2D video stream.

frame packing
A format that packs the L- and R-views into a single 2D video stream. The packing may be done spatially, where each video frame is divided into sub-frames, one containing the L-view and one containing the R-view. The packing can also be done sequentially, where alternating video frames represent L- and R-views.

legacy answerer
An answerer (in the SDP offer/answer model [RFC3264]) that does not support the "3dFormat" attribute. The legacy answerer can be the streaming server or the streaming client, but is not compliant to this document.

L-view
A visual entity that is to be projected to the left eye.

L-stream
A 2D video stream consisting of a sequence of L-views.

parallax map
A two dimensional map, each pixel of which defines the parallax of one or more pixels in an associated 2D video frame.

parallax map stream
An auxiliary stream, which contains a sequence of parallax maps. The parallax map stream is synchronised with the associated 2D video stream.

R-view
A visual entity that is to be projected to the right eye.

R-stream
A 2D video stream consisting of a sequence of R-views.

stereoscopic 3D video
The L- and R-streams, ready to be projected to the viewer's left and right eyes.

sub-frame
A part of a video frame.

4. The "3dFormat" attribute

The media-level SDP attribute "3dFormat" signals the format of stereoscopic 3D video. The attribute transfers this information through two parameters: one indicating the format type of the stereoscopic 3D video carried in the media stream(s), and the other indicating the type of the video component, which is a constituent element of the stereoscopic 3D video. The video component type depends on the format type of the stereoscopic 3D video. The syntax of the attribute is defined as follows:

```
a=3dFormat:<Format Type> <Component Type>
```

The <Format Type> can have the following values (as indicated between the quotes):

"FP" Frame Packing
The L- and R-views are packed into a single stream. The packing may use a side-by-side, top-and-bottom, interleaved, checkerboard or frame sequential format.

"SC" Simulcast
The L- and R-streams are transmitted separately.

"2DA" 2D + auxiliary
2D video and auxiliary data (such as depth maps or parallax maps) are transmitted. These can be transmitted in a single stream, as well as in two separate streams.

The <Component Type> can have the following values (as indicated between the quotes):

"C" Centre view
The associated stream is a C-stream.

"CD" centre view and depth map
The associated stream contains both the C-view and depth map sequences.

"ChB" Checkerboard
The video frame consists of alternating pixels from the corresponding L- and R-views, as illustrated by Figure 1.

"CP" Centre view and parallax map
The associated stream contains both the C-view and parallax map sequences.

"D" Depth map
The associated stream is a sequence of depth maps.

"L" Left view
The associated stream is the L-stream.

"LD" Left view and depth map
The associated stream contains both the L-view and depth map sequences.

"LIL" Line Interleaved
Each video frame consists of alternating scan lines from the L- and R-views.

"LP" Left view and parallax map
The associated stream contains both the L-view and parallax map sequences.

"P" Parallax map
The associated stream is a sequence of parallax maps.

"R" Right view
The associated stream is the R-stream.

"SbS" Side by Side
Each video frame is divided in two equally sized sub-frames, spatially positioned side by side of each other. One sub-frame contains the L-view, whereas the other contains the R-view.

"Seq" Frame Sequential
The single video stream consists of alternating frames from the L- and R-streams. Additional signalling, e.g. AVC SEI messages [ISO/IEC 14496-10], is needed to signal which frames contain L- and which contain R-views.

"TaB" Top and Bottom
Each video frame is divided in two equally sized sub-frames, spatially positioned above each other. One sub-frame contains the L-view, whereas the other contains the R-view.

```
+-+-+-+-+-+-+
|L|R|L|R|L|R|
+-+-+-+-+-+-+
|R|L|R|L|R|L|
+-+-+-+-+-+-+
|L|R|L|R|L|R|
+-+-+-+-+-+-+
```

The checkerboard pattern. The transmitted video frame is composed of pixels from the L- and R-views. Samples from the L-view are indicated with "L", whereas samples from the R-view are indicated with "R".

Figure 1

5. Grouping

When multiple streams carry a single stereoscopic 3D video, (e.g. C-stream and parallax map, or separately transmitted L- and R-streams), the grouping mechanism from [RFC5888] MUST be used.

However, to cater for the special requirements of 3D signalling, the semantics are expanded:

```
group-attribute     = "a=group:" semantics *(SP identification-tag)
semantics           = "LS" / "FID" / "3DS" / semantics-extension
semantics-extension = token
```

The grouping is needed when multiple streams carry a single stereoscopic 3D video. This is the case when the <format type> is "SC", or the <format type> is "2DA" and the 2D video and auxiliary data are transmitted as multiple streams. A group with the "3Ds" semantics is called a "3DS group".

A 3DS group MUST NOT contain data that is (potentially) inconsistent with other data in the 3DS group:

- o A 3DS group MUST NOT contain both a parallax map stream and a depth map stream.
- o A 3DS group MUST NOT contain more than one parallax map stream.
- o A 3DS group MUST NOT contain more than one depth map stream.
- o A 3DS group MUST contain at least one 2D video stream.
- o If a 3GS group contains an L- and an R-stream, it MUST NOT contain a depth map or a parallax map.
- o If a 3DS group contains only one 2D video stream, it MUST also contain a parallax map stream or a depth map stream.
- o If a 3DS group contains a parallax map stream or a depth map stream, it MUST also contain a 2D video stream.

6. Combinations of attribute values and group usage

The following table summarises the possible combinations of attribute values and grouping:

	FP	SC	2DA
C			D/P,3DS
CD			T
ChB	T		
CP			T
D			C/L,3DS
L		R,3DS	D/P,3DS
LD			T
LIL	T		
LP			T
P			C/L,3DS
R		L,3DS	
SbS	T		
Seq	T		
TaB	T		

The table is to be read as follows:

- o The columns indicate <Format Type> values, whereas the rows indicate <Component Type> values.
- o For one particular column, we denote the <Format Type> value by "FT" and the <Component Type> value by "CT".
- o When an entry in the table is empty, it means that the corresponding combination of FT and CT is not allowed.

- o When an entry in the table contains a single <Component Type> value CTsec, it means that another stream with the <Component Type> value CTsec and the same <Format Type> value FT is needed.

- o When multiple <Component Type> values are listed, separated by a "/" symbol, only one secondary stream is needed, which must have one of the listed <Component Type> values, and the same <Format Type> value FT.

- o When an entry contains "3DS", it means that a 3DS group is needed.

- o When an entry in the table contains the letter "T" (true), it means that the corresponding combination FT and CT is allowed, that there is no required secondary stream, and that a 3DS group is not needed.

7. SDP offer/answer with 3D support

This section describes how the SDP offer/answer model (see [RFC3264]) can be used to negotiate the 3D format. It is assumed that both offerer and answerer are compliant to this document. The case where the answerer is a legacy answerer is described in Section 8.

An example where the SDP offer/answer model can be used to negotiate the 3D format, is the case where the offerer offers two representations of the same stereoscopic 3D video: one frame packed and one as L/R simulcast. In this case, the answerer can select the format of its preference, according to its capabilities or as a trade-off between bandwidth and video quality.

There may also be cases where the answerer prefers to receive a 2D version, even when it supports stereoscopic 3D video and the "3dFormat" attribute. For example, this might happen when the user prefers to watch without glasses this time.

The following statements apply for the answerer:

- o The answerer MUST NOT omit the "3dFormat" attribute for the accepted streams. The answerer MAY omit the "3dFormat" attribute for the rejected streams.

- o The answerer MUST NOT change the value of the "3dFormat" attribute. This means, that the answerer can only choose between the 3D formats advertised in the offer.

- o In case the offer contains simulcast of the L- and R-view, the answerer MAY choose just one view. In this case, it MUST select only that view. This means that the port number of the other view MUST be set to zero in the answer.

- o In case the offer contains a 2D stream and an auxiliary stream as separate streams, the answerer MAY choose only the 2D stream. In this case, it MUST select the 2D stream, and MUST NOT select the auxiliary stream. This means that the port number of the auxiliary stream MUST be set to zero in the answer.

- o In case the offer contains a 2D stream and an auxiliary stream as a single stream, the answerer MAY choose to reject the stream by setting the port number in the answer to zero.

- o In case of frame packing, if the answerer prefers not to have frame packing, it MUST reject the stream by setting the port number in the answer to zero.

o If the answerer selects multiple 3D formats, it MUST be prepared to send/receive (depending on whether it is a streaming server or client or both) associated streams simultaneoulsy.

The following statements apply for the offerer:

o The offerer MUST check if the "3dFormat" attribute is included in the answer. If it is not, it SHOULD handle the answer as described in Section 8.

o The offerer SHOULD list the 3D formats in order of preference.

o When multiple 3D formats are selected, the offerer MAY initiate all associated streams. Alternatively, it MAY update its offer with a reduced number of 3D formats.

o If all 3D formats have been rejected, the offerer MAY issue a new offer with 2D video instead.

o If only an auxiliary stream is selected in the answer, the offerer SHOULD update its offer with only the associated 2D video stream. Alternatively, it MAY update its offer advertising another 3D format.

8. SDP offer/answer without 3D support

Since a legacy answerer does not support the "3dFormat" attribute, it might reject the offer. In this case the offerer MAY send a new offer with only a 2D video stream.

On the other hand, it is also possible that the legacy answerer accepts the offer but omits the "3dFormat" attribute in the answer. In this case the offerer is able to deduct that the answerer is a legacy answerer without 3D support. In the following subsections, we describe what the offerer still can do to provide a good user experience with a legacy answerer, for each of the 3D format styles. We assume that the offer was accepted, but a legacy answerer was detected.

8.1. Frame packing

In case the original offer contains frame packing, and the answer does not contain the "3dFormat" attribute, the offerer SHOULD treat that media stream as a 2D stream.

Note: in some cases, the answerer may be a legacy device that is capable of rendering a frame packed 3D stream, but does not understand the "3dFormat" attribute. For example, the user may be able to switch manually to 3D. Therefore, the server MAY stream the frame packed video as it is.

8.2. 2D and auxiliary as a single stream

If the original offer contains a 2D video and an auxiliary video in a single stream, and the answer does not contain the "3dFormat" attribute, the offerer SHOULD treat that media stream as a 2D stream.

8.3. 2D and auxiliary as two separate streams

When the offerer sends an offer to a legacy answerer, and the offer contains a 2D video and an auxiliary video in two separate streams, there are the following possibilities:

- o If the answerer selects only the 2D video stream then 2D video streaming can be done as agreed.

- o If the answerer selects only the auxiliary video, the offerer MAY treat that stream as a 2D video stream. If it does not, the offerer SHOULD update its offer without the auxiliary video.

- o If the answerer selects both video streams, but omits the "3dFormat" attribute, the offerer MAY update its offer without the auxiliary video.

In case the offerer updates its offer by setting the port for auxiliary video to zero, it MUST NOT include the "3dFormat" attribute or use "3DS" grouping for the 2D stream.

8.4. Simulcast of L- and R-views

When the offerer sends an offer to simulcast the L- and R-view to the legacy answerer, we have the following possibilities:

- o If the answerer selects only one video stream, the offerer MAY stream the 2D video as agreed.

- o If the answerer selects both video streams, but omits the "3dFormat" attribute, the offerer MAY update its offer with only the L- or the R-stream.

In case the offerer updates its offer with only the L- or R-stream by setting one of the ports to zero, it MUST NOT include the "3dFormat" attribute or use "3DS" grouping for the offered stream.

9. Examples

9.1. One single frame compatible stream

The following is an example of an SDP description of a session which contains a single stream, in which the L- and R-streams are packed, in side by side fashion.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:FP SbS
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

9.2. Two separate streams

The following is an example of an SDP description of a session with an audio stream, an L-stream and an R-stream.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:SC L
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:SC R
a=mid:2
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

9.3. C-stream and depth map stream

The following is an example of an SDP description of a session with an audio stream, a C-stream and a depth map stream.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:2DA C
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:2DA D
a=mid:2
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

9.4. Stereoscopic 3D video with two different formats

In the following example, there are two different formats for stereoscopic 3D video. One consists of stream 1 (C-stream) and stream 2 (parallax map stream), whereas the other consists of stream 3 (L-stream) and stream 4 (R-stream). There also is an audio stream, which can be used with both formats.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
a=group:3DS 3 4
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:2DA C
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:2DA P
a=mid:2
m=video 49174 RTP/AVP 103
a=rtpmap:103 H264/90000
a=3dFormat:SC L
a=mid:3
m=video 49176 RTP/AVP 105
a=rtpmap:105 H264/90000
a=3dFormat:SC R
a=mid:4
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

10. Formal ABNF grammar of the "3dFormat" attribute

This section contains the formal ABNF grammar of the "3dFormat" attribute.

```
3dFormat-attribute      = "a=3dFormat:" formatType componentType
formatType              = "FP"/"SC"/"2DA"/formatType-extension
formatType-extension    = token
componentType           = "C"/"CD"/"ChB"/"CP"/"D"/"L"/"LD"/
                          "LIL"/"LP"/"P"/"R"/"SbS"/"Seq"/"TaB"/
                          componentType-extension
componentType-extension = token
```

11. Security Considerations

The authors foresee no security issues in addition to those already listed in [RFC4566].

12. IANA Considerations

12.1. "3dFormat" attribute

Following the guidelines in [RFC4566], the SDP attribute has to be registered at IANA:

- o Contact name/email: authors of this RFC

- o Attribute name: 3dFormat

- o Long-form attribute name: Attribute for signalling the format of a stereoscopic 3D video carried in the media stream(s).

- o Type of attribute: media level

- o Subject to charset: no

The "3dFormat" SDP media-level attribute is used to signal the format of stereoscopic 3D video, carried in one or more media stream(s).

The attribute has the following syntax:

```
a=3dFormat:<Format Type> <Component Type>
```

The <Format Type> indicates the format type of the stereoscopic 3D video carried in the media stream(s). It indicates whether the stereoscopic 3D video is frame packed, simulcast or consists of a 2D video stream and an auxiliary stream. The <Format Type> can have the following values (as indicated between the quotes):

```
"FP"     frame packed
"SC"     simulcast
"2DA"    2D + auxiliary
```

The <Component Type> indicates the type of the video component, which is a constituent element of the stereoscopic 3D video. It can have the following values:

"C"	centre view
"CD"	centre view and depth map
"ChB"	checkerboard
"CP"	centre view and parallax map
"D"	depth map
"L"	left view
"LD"	left view and depth map
"LIL"	line interleaved
"LP"	left view and parallax map
"P"	parallax map
"R"	right view
"SbS"	side by side
"Seq"	frame sequential
"TaB"	top and bottom

12.2. "3DS" value for "group" semantics

Following the standards action policy from [RFC5226], the following semantics have to be registered with IANA in the "Semantics for the "group" SDP Attribute" registry under "SDP Parameters":

Semantics	Token	Reference
3D synchronised	3DS	this RFC

13. Acknowledgements

The authors would like to thank Stephen Botzko, Imed Bouazizi, Pedro Capelastegui, Roni Even, Miguel Garcia, Ted Hardie, Jonathan Lennox, Yue Peiyu and Tian Linyi for their review comments.

14. Normative References

[HDMIv1.4a]
HDMI, "HDMI Specification Version 1.4a", March 2010.

[ISO/IEC 23002-3]
MPEG, "MPEG video technologies part 3: Representation of auxiliary video and supplemental information", ISO/IEC FDIS 23002-3:2007(E), December 2002.

[ISO/IEC 14496-10]
MPEG, "H.264/MPEG-4 Part 10: Advanced video coding for generic audiovisual services", ISO/IEC FDIS 14496-10:2010, March 2010.

[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.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

Authors' Addresses

Bert Greevenbosch
Huawei Technologies Co., Ltd.
Huawei Industrial Base
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone: +86-755-28978088
Email: bert.greevenbosch@huawei.com

Hui Yu
Huawei Technologies Co., Ltd.
Huawei Nanjing R&D Center
101 Software Avenue
Yuhuatai District
Nanjing 210012
P.R. China

Phone: +86-25-56620323
Email: huiyu@huawei.com

MMUSIC S. Loreto
Internet-Draft G. Camarillo
Intended status: Standards Track Ericsson
Expires: January 5, 2012 July 4, 2011

Stream Control Transmission Protocol (SCTP)-Based Media Transport in the Session Description Protocol (SDP)
draft-ietf-mmusic-sctp-sdp-00

Abstract

SCTP (Stream Control Transmission Protocol) is a transport protocol used to establish associations between two endpoints. This document describes how to express media transport over SCTP in SDP (Session Description Protocol). This document defines the 'SCTP' and 'SCTP/DTLS' protocol identifiers for SDP.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

SDP (Session Description Protocol) [RFC4566] provides a general-purpose format for describing multimedia sessions in announcements or invitations. RFC4145 [RFC4145] specifies a general mechanism for describing and establishing TCP (Transmission Control Protocol) streams. RFC 4572 [RFC4572] extends RFC4145 [RFC4145] for describing TCP-based media streams that are protected using TLS (Transport Layer Security) [RFC5246].

This document defines a new protocol identifier, 'SCTP', to describe SCTP-based [RFC4960] media streams. Additionally, this document specifies the use of the 'setup' and 'connection' SDP attributes to establish SCTP associations. These attributes were defined in RFC4145 [RFC4145] for TCP. This document discusses their use with SCTP.

Additionally this document defines a new protocol identifier, 'SCTP/DTLS', to establish secure SCTP-based media streams over DTLS (Datagram Transport Layer Security) [RFC4347], as specified in [RFC6083], using SDP. The authentication certificates are interpreted and validated as defined in RFC4572 [RFC4572]. Self-signed certificates can be used securely, provided that the integrity of the SDP description is assured as defined in RFC4572 [RFC4572].

TLS is designed to run on top of a byte-stream oriented transport protocol providing a realible, in-sequence delivery like TCP. Since no-one so far has implemented SCTP over TLS, due to some serious limitations described in [RFC6083], this document does not make use of TLS over SCTP as described in RFC3436 [RFC3436].

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

3. Protocol Identifier

The following is the format for an 'm' line, as specified in RFC4566 [RFC4566]:

```
m=<media> <port> <proto> <fmt> ...
```

This document defines two new values for the 'proto' field: 'SCTP' and 'SCTP/DTLS'.

The 'SCTP' protocol identifier is similar to both the 'UDP' and 'TCP' protocol identifiers in that it only describes the transport protocol and not the upper-layer protocol. Media described using an 'm' line containing the 'SCTP' protocol identifier are carried using SCTP [RFC4960].

The 'SCTP/DTLS' protocol identifier indicates that the media described will use the Datagram Transport Layer Security (DTLS) [RFC4347] over SCTP as specified in [RFC6083].

An 'm' line that specifies 'SCTP' or 'SCTP/DTLS' MUST further qualify the application-layer protocol using an fmt identifier.

An 'm' line that specifies 'SCTP/DTLS' MUST further provide a certificate fingerprint. An SDP attribute (an 'a' line) is used to transport and exchange end point certificate. The authentication certificates are interpreted and validated as defined in [RFC4572].

4. The Setup and Connection Attributes and Association Management

The use of the 'setup' and 'connection' attributes in the context of an SCTP association is identical to the use of these attributes in the context of a TCP connection. That is, SCTP endpoints MUST follow the rules in Sections 4 and 5 of RFC 4145 [RFC4145] when it comes to the use of the 'setup' and 'connection' attributes in offer/answer [RFC3264] exchanges.

The management of an SCTP association is identical to the management of a TCP connection. That is, SCTP endpoints MUST follow the rules in Section 6 of RFC 4145 [RFC4145] to manage SCTP associations. Whether to use the SCTP ordered or unordered delivery service is up to the applications using the SCTP association.

5. Multihoming

An SCTP endpoint, unlike a TCP endpoint, can be multihomed. An SCTP endpoint is considered to be multihomed if it has more than one IP address. A multihomed SCTP endpoint informs a remote SCTP endpoint about all its IP addresses using the address parameters of the INIT or the INIT-ACK chunk (depending on whether the multihomed endpoint is the one initiating the establishment of the association). Therefore, once the address provided in the 'c' line has been used to establish the SCTP association (i.e., to send the INIT chunk), address management is performed using SCTP. This means that two SCTP endpoints can use addresses that were not listed in the 'c' line but that were negotiated using SCTP mechanisms.

During the lifetime of an SCTP association, the endpoints can add and remove new addresses from the association at any point [RFC5061]. If an endpoint removes the IP address listed in its 'c' line from the SCTP association, the endpoint MUST update the 'c' line (e.g., by sending a re-INVITE with a new offer) so that it contains an IP address that is valid within the SCTP association.

In some environments, intermediaries performing firewall control use the addresses in offer/answer exchanges to perform media authorization. That is, policy-enforcement network elements do not let media through unless it is sent to the address in the 'c' line.

In such network environments, the SCTP endpoints can only exchange media using the IP addresses listed in their 'c' lines. In these environments, an endpoint wishing to use a different address needs to update its 'c' line (e.g., by sending a re-INVITE with a new offer) so that it contains the new IP address.

6. Network Address Translation (NAT) Considerations

SCTP specific features (not present in UDP/TCP), such as the checksum (CRC32c) value calculated on the whole packet (not just the header) or its multihoming capabilities, present new challenges for NAT traversal. [I-D.ietf-behave-sctpnat] describes an SCTP specific variant of NAT, which provides similar features of Network Address and Port Translation (NAPT).

Current NATs do not typically support SCTP. As an alternative to design SCTP specific NATs, Encapsulating SCTP into UDP [I-D.tuexen-sctp-udp-encaps] makes it possible to use SCTP in networks with legacy NAT and firewalls not supporting SCTP.

At the time of writing, the work on NAT traversal for SCTP is still work in progress. Additionally, no extension has been defined to integrate ICE (Interactive Connectivity Establishment) [RFC5768] with SCTP and its multihoming capabilities either. Therefore, this specification does not define how to describe SCTP-over-UDP streams in SDP or how to establish and maintain SCTP associations using ICE. Should these features be specified for SCTP in the future, there will be a need to specify how to use them in an SDP environment as well.

7. Examples

The following examples show the use of the 'setup' and 'connection' SDP attributes. As discussed in Section 4, the use of these attributes with an SCTP association is identical to their use with a TCP connection. For the purpose of brevity, the main portion of the session description is omitted in the examples, which only show 'm' lines and their attributes (including 'c' lines).

7.1. Actpass/Passive

An offerer at 192.0.2.2 signals its availability for an SCTP association at SCTP port 54111. Additionally, this offerer is also willing to initiate the SCTP association:

```
m=image 54111 SCTP *
c=IN IP4 192.0.2.2
a=setup:actpass
a=connection:new
```

Figure 1

The endpoint at 192.0.2.1 responds with the following description:

```
m=image 54321 SCTP *
c=IN IP4 192.0.2.1
a=setup:passive
a=connection:new
```

Figure 2

This will cause the offerer (at 192.0.2.2) to initiate an SCTP association to port 54321 at 192.0.2.1.

7.2. Existing Connection Reuse

Subsequent to the exchange in Section 7.1, another offer/answer exchange is initiated in the opposite direction. The endpoint at 192.0.2.1, which now acts as the offerer, wishes to continue using the existing association:

```
m=application 54321 SCTP *
c=IN IP4 192.0.2.1
a=setup:passive
a=connection:new
```

Figure 3

The endpoint at 192.0.2.2 also wishes to use the existing SCTP association and responds with the following description:

```
m=application 9 SCTP *
c=IN IP4 192.0.2.2
a=setup:active
a=connection:new
```

Figure 4

The existing SCTP association between 192.0.2.2 and 192.0.2.1 will be reused.

7.3. SDP description for DTLS Connection

An offerer at 192.0.2.2 signals the availability of a T.38 fax session over SCTP/DTLS.

```
m=image 54111 SCTP/DTLS t38
c=IN IP4 192.0.2.2
a=setup:actpass
a=connection:new
a=fingerprint:SHA-1 \
  4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

Figure 5

8. Security Considerations

See RFC 4566 [RFC4566] for security considerations on the use of SDP in general. See RFC 3264 [RFC3264], RFC 4145 [RFC4145] and RFC 4572 [RFC4572] for security considerations on establishing media streams using offer/answer exchanges. See RFC 4960 [RFC4960] for security considerations on SCTP in general and [RFC6083] for security consideration using DTLS on top of SCTP. This specification does not introduce any new security consideration in addition to the ones discussed in those specifications.

9. IANA Considerations

This document defines two new proto values: 'SCTP' and 'SCTP/DTLS'. Their formats are defined in Section 3. These proto values should be registered by the IANA under "Session Description Protocol (SDP) Parameters" under "proto".

The SDP specification, [RFC4566], states that specifications defining new proto values, like the SCTP and SCTP/DTLS proto values defined in this RFC, must define the rules by which their media format (fmt) namespace is managed. For the SCTP protocol, new formats SHOULD have an associated MIME registration. Use of an existing MIME subtype for the format is encouraged. If no MIME subtype exists, it is RECOMMENDED that a suitable one is registered through the IETF process [RFC4288] [RFC4289] by production of, or reference to, a standards-track RFC that defines the transport protocol for the format.

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.

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

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4289] Freed, N. and J. Klensin, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 4289, December 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

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

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, July 2006.

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

[RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP)

Dynamic Address Reconfiguration", RFC 5061, September 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

10.2. Informative References

[RFC3436] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol", RFC 3436, December 2002.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, January 2011.

[RFC5768] Rosenberg, J., "Indicating Support for Interactive Connectivity Establishment (ICE) in the Session Initiation Protocol (SIP)", RFC 5768, April 2010.

[I-D.ietf-behave-sctpnat] Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation", draft-ietf-behave-sctpnat-04 (work in progress), December 2010.

[I-D.tuexen-sctp-udp-encaps] Tuexen, M. and R. Stewart, "UDP Encapsulation of SCTP Packets", draft-tuexen-sctp-udp-encaps-06 (work in progress), January 2011.

Authors' Addresses

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Salvatore.Loreto@ericsson.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: October 22, 2017

C. Holmberg
Ericsson
R. Shpount
TurboBridge
S. Loreto
G. Camarillo
Ericsson
April 20, 2017

Session Description Protocol (SDP) Offer/Answer Procedures For Stream Control Transmission Protocol (SCTP) over Datagram Transport Layer Security (DTLS) Transport.

draft-ietf-mmusic-sctp-sdp-26

Abstract

The Stream Control Transmission Protocol (SCTP) is a transport protocol used to establish associations between two endpoints. draft-ietf-tsvwg-sctp-dtls-encaps-09 specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol, referred to as SCTP-over-DTLS.

This specification defines the following new Session Description Protocol (SDP) protocol identifiers (proto values):'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP'. This specification also specifies how to use the new proto values with the SDP Offer/Answer mechanism for negotiating SCTP-over-DTLS associations.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

SDP (Session Description Protocol) [RFC4566] provides a general-purpose format for describing multimedia sessions in announcements or invitations. TCP-Based Media Transport in the Session Description Protocol (SDP) [RFC4145] specifies a general mechanism for describing and establishing TCP [RFC0793] streams. Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in SDP [RFC8122] extends RFC4145 [RFC4145] for describing TCP-based media streams that are protected using TLS.

The Stream Control Transmission Protocol (SCTP) [RFC4960] is a reliable transport protocol used to transport data between two endpoints using SCTP associations.

[I-D.ietf-tsvwg-sctp-dtls-encaps] specifies how SCTP can be used on top of the Datagram Transport Layer Security (DTLS) protocol, referred to as SCTP-over-DTLS.

This specification defines the following new Session Description Protocol (SDP) [RFC4566] protocol identifiers (proto values):'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP'. This specification also specifies how to use the new proto values with the SDP Offer/Answer mechanism [RFC3264] for negotiating SCTP-over-DTLS associations.

NOTE: Due to the characteristics of TCP, while multiple SCTP streams can still be used, usage of 'TCP/DTLS/SCTP' will always force ordered and reliable delivery of the SCTP packets, which limits the usage of the SCTP options. Therefore, it is RECOMMENDED that TCP is only used in situations where UDP traffic is blocked.

2. 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].

3. SCTP Terminology

SCTP Association: A protocol relationship between SCTP endpoints, composed of the two SCTP endpoints and protocol state information including Verification Tags and the currently active set of Transmission Sequence Numbers (TSNs), etc. An association can be uniquely identified by the transport addresses used by the endpoints in the association.

SCTP Stream: A unidirectional logical channel established from one to another associated SCTP endpoint, within which all user messages are delivered in sequence except for those submitted to the unordered delivery service.

SCTP-over-DTLS: SCTP used on top of DTLS, as specified in [I-D.ietf-tsvwg-sctp-dtls-encaps].

4. SDP Media Descriptions

4.1. General

This section defines the following new SDP Media Description (m-line) protocol identifiers (proto values) for describing an SCTP association: 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP'. The section also describes how an m- line, associated with the proto values, is created.

The following is the format for an m- line, as specified in RFC4566 [RFC4566]:

```
m=<media> <port> <proto> <fmt> ...
```

The 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' proto values are similar to both the 'UDP' and 'TCP' proto values in that they only describe the transport-layer protocol and not the upper-layer protocol.

NOTE: When the 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' proto values are used, the underlying transport protocol is respectively UDP and TCP; SCTP is carried on top of DTLS which is on top of those transport-layer protocols.

4.2. Protocol Identifiers

The new proto values are defined as below:

- The 'UDP/DTLS/SCTP' proto value describes an SCTP association on top of a DTLS association on top of UDP, as defined in Section 7.

- The 'TCP/DTLS/SCTP' proto value describes an SCTP association on top of a DTLS association on top of TCP, as defined in Section 8.

4.3. Media Format Management

[RFC4566] defines that specifications defining new proto values must define the rules by which their media format (fmt) namespace is managed.

An m- line with a proto value of 'UDP/DTLS/SCTP' or 'TCP/DTLS/SCTP' always describes a single SCTP association.

In addition, such m- line MUST further indicate the application-layer protocol using an 'fmt' identifier. There MUST be exactly one fmt value per m- line associated with the proto values defined in this specification. The 'fmt' namespace associated with those proto values describes the generic application usage of the entire SCTP association, including the associated SCTP streams.

When the 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' proto values, the m-line fmt value, identifying the application-layer protocol, MUST be registered by IANA. Section 15.3 defines the IANA registry for the media format namespace.

NOTE: A mechanism on how to describe, and manage, individual SCTP streams within an SCTP association, is outside the scope of this specification. [I-D.ietf-mmusic-data-channel-sdpneg] defines a mechanism for negotiating individual SCTP streams used to realize WebRTC data channels [I-D.ietf-rtcweb-data-channel].

4.4. Syntax

4.4.1. General

This section defines the values that can be used within an SDP media description ("m=" line) associated with an SCTP-over-DTLS association.

This specification creates an IANA registry for 'association-usage' values.

4.4.2. SDP Media Description values

```
m= line parameter         parameter value(s)
------------------------------------------------------------------
<media>:                  'application'
<proto>:                  'UDP/DTLS/SCTP' or 'TCP/DTLS/SCTP'
<port>:                   UDP port number (for 'UDP/DTLS/SCTP')
                          TCP port number (for 'TCP/DTLS/SCTP')
<fmt>:                    a string denoting the association-usage,
                          limited to the syntax of a 'token' as
                          defined in RFC4566.
```

4.5. Example

```
m=application 12345 UDP/DTLS/SCTP webrtc-datachannel
a=sctp-port:5000
a=max-message-size:100000
```

NOTE: 'webrtc-datachannel' indicates the WebRTC Data Channel Establishment Protocol defined in [I-D.ietf-rtcweb-data-protocol].

5. SDP 'sctp-port' Attribute

5.1. General

This section defines a new SDP media-level attribute, 'sctp-port'. The attribute can be associated with an SDP media description (m-line) with a 'UDP/DTLS/SCTP' or a 'TCP/DTLS/SCTP' proto value. In that case the m- line port value indicates the port of the underlying transport layer protocol (UDP or TCP), and the 'sctp-port' value indicates the SCTP port.

No default value is defined for the SDP sctp-port attribute. Therefore, if the attribute is not present, the associated m- line MUST be considered invalid.

NOTE: This specification only defines the usage of the SDP 'sctp-port' attribute when associated with an m- line containing one of the following proto values: 'UDP/DTLS/SCTP' or 'TCP/DTLS/SCTP'. Usage of the attribute with other proto values needs to be defined in a separate specification.

5.2. Syntax

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

The definition of the SDP 'sctp-port' attribute is:

```
Attribute name:     sctp-port
Type of attribute:  media
Mux category:       CAUTION
Subject to charset: No
Purpose:            Indicate the SCTP port value associated with
                    the SDP Media Description.
Appropriate values: Integer
Contact name:       Christer Holmberg
Contact e-mail:     christer.holmberg@ericsson.com
Reference:          RFCXXXX
```

Syntax:

```
      sctp-port-value = 1*5<DIGIT defined in RFC4566>
```

The SCTP port range is between 0 and 65535 (both included). Leading zeroes MUST NOT be used.

Example:

```
     a=sctp-port:5000
```

5.3. Mux Category

The mux category [I-D.ietf-mmusic-sdp-mux-attributes] for the SDP 'sctp-port' attribute is CAUTION.

As the usage of multiple SCTP associations on top of a single DTLS association is outside the scope of this specification, no mux rules are specified for the 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' proto values. Future extensions, that define how to negotiate multiplexing of multiple SCTP associations of top of a single DTLS association, need to also define the mux rules for the attribute.

6. SDP 'max-message-size' Attribute

6.1. General

This section defines a new SDP media-level attribute, 'max-message-size'. The attribute can be associated with an m- line to indicate the maximum SCTP user message size (indicated in bytes) that an SCTP endpoint is willing to receive on the SCTP association associated with the m- line. Different attribute values can be used in each direction.

An SCTP endpoint MUST NOT send a SCTP user message with a message size that is larger than the maximum size indicated by the peer, as it cannot be assumed that the peer would accept such message.

If the SDP 'max-message-size' attribute contains a maximum message size value of zero, it indicates the SCTP endpoint will handle messages of any size, subject to memory capacity etc.

If the SDP 'max-message-size' attribute is not present, the default value is 64K.

NOTE: This specification only defines the usage of the SDP 'max-message-size' attribute when associated with an m- line containing one of the following proto values: 'UDP/DTLS/SCTP' or 'TCP/DTLS/SCTP'. Usage of the attribute with other proto values needs to be defined in a separate specification.

6.2. Syntax

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

The definition of the SDP 'max-message-size' attribute is:

```
Attribute name:     max-message-size
Type of attribute:  media
Mux category:       CAUTION
Subject to charset: No
Purpose:            Indicate the maximum message size
                    (indicated in bytes) that an SCTP
                    endpoint is willing to receive on the
                    SCTP association associated with the SDP
                    Media Description.
Appropriate values: Integer
Contact name:       Christer Holmberg
Contact e-mail:     christer.holmberg@ericsson.com
Reference:          RFCXXXX
```

Syntax:

```
max-message-size-value = 1*<DIGIT defined in RFC4566>
```

Leading zeroes MUST NOT be used.

Example:

```
a=max-message-size:100000
```

6.3. Mux Category

The mux category for the SDP 'max-message-size' attribute is CAUTION.

As the usage of multiple SCTP associations on top of a single DTLS association is outside the scope of this specification, no mux rules are specified for the 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' proto values.

7. UDP/DTLS/SCTP Transport Realization

The UDP/DTLS/SCTP transport is realized as described below:

- o SCTP on top of DTLS is realized according to the procedures defined in [I-D.ietf-tsvwg-sctp-dtls-encaps]; and

- o DTLS on top of UDP is realized according to the procedures in defined in [RFC6347].

NOTE: While [I-D.ietf-tsvwg-sctp-dtls-encaps] allows multiple SCTP associations on top of a single DTLS association, the procedures in this specification only support the negotiation of a single SCTP association on top of any given DTLS association.

8. TCP/DTLS/SCTP Transport Realization

The TCP/DTLS/SCTP transport is realized as described below:

- o SCTP on top of DTLS is realized according to the procedures defined in [I-D.ietf-tsvwg-sctp-dtls-encaps]; and

- o DTLS on top of TCP is realized using the framing method defined in [RFC4571], with DTLS packets being sent and received instead of RTP/RTCP packets using the shim defined in [RFC4571], so that length field defined in [RFC4571] precedes each DTLS message, and SDP signaling according to the procedures defined in this specification.

NOTE: TLS on top of TCP, without using the framing method defined in [RFC4571] is outside the scope of this specification. A separate proto value would need to be registered for such transport realization.

9. Association And Connection Management

9.1. General

This section describes how to manage an SCTP association, DTLS association and TCP connection using SDP attributes.

The SCTP association, the DTLS association and the TCP connection are managed independently from each other. Each can be established and closed without impacting others.

The detailed SDP Offer/Answer [RFC3264] procedures for the SDP attributes are described in Section 10.

9.2. SDP sendrecv/sendonly/recvonly/inactive Attribute

This specification does not define semantics for the SDP direction attributes [RFC4566]. Unless semantics of these attributes for an SCTP association usage have been defined, SDP direction attributes MUST be ignored if present.

9.3. SCTP Association

When an SCTP association is established, both SCTP endpoints MUST initiate the SCTP association (i.e. both SCTP endpoints take the ’active’ role), and MUST use the same SCTP port as client port and server port (in order to prevent two separate SCTP associations from being established).

As both SCTP endpoints take the 'active' role, the SDP 'setup' attribute [RFC4145] does not apply to SCTP association establishment. However the 'setup' attribute does apply to establishment of the underlying DTLS association and TCP connection.

NOTE: The procedure above is different from TCP, where one endpoint takes the 'active' role, the other endpoint takes the 'passive' role, and only the 'active' endpoint initiates the TCP connection [RFC4145].

NOTE: When the SCTP association is established it is assumed that any NAT traversal procedures for the underlying transport protocol (UDP or TCP) have successfully been performed.

The SDP 'connection' attribute [RFC4145] does not apply to the SCTP association. In order to trigger the closure of an existing SCTP association, and establishment of a new SCTP association, the SDP 'sctp-port' attribute [Section 5] is used to indicate a new (different than the ones currently used) SCTP port. The existing SCTP association is closed, and the new SCTP association is established, if one or both endpoints signal a new SCTP port. The 'connection' attribute does apply to establishment of underlying TCP connections.

Alternatively, an SCTP association can be closed using the SDP 'sctp-port' attribute with a zero attribute value. Later, a new SCTP association can be established using the procedures in this section for establishing an SCTP association.

SCTP associations might be closed without SDP signalling, e.g, in case of a failure. The procedures in this section MUST be followed to establish a new SCTP association. This requires a new SDP Offer/Answer exchange. New (different than the ones currently used) SCTP ports MUST be used by both endpoints.

NOTE: Closing and establishing a new SCTP association using the SDP 'sctp-port' attribute will not affect the state of the underlying DTLS association.

9.4. DTLS Association (UDP/DTLS/SCTP And TCP/DTLS/SCTP)

A DTLS association is managed according to the procedures in [I-D.ietf-mmusic-dtls-sdp]. Hence, the SDP 'setup' attribute is used to negotiate the (D)TLS roles ('client' and 'server') [RFC8122].

NOTE: The SDP 'setup' attribute is used to negotiate both the DTLS roles and the TCP roles (Section 9.5).

NOTE: As described in [RFC5245], if the Interactive Connectivity Establishment (ICE) mechanism [RFC5245] is used, all ICE candidates associated with a DTLS association are considered part of the same DTLS association. Thus, a switch from one candidate pair to another candidate pair will not trigger the establishment of a new DTLS association.

9.5. TCP Connection (TCP/DTLS/SCTP)

The TCP connection is managed according to the procedures in [RFC4145]. Hence, the SDP 'setup' attribute is used to negotiate the TCP roles ('active' and 'passive'), and the SDP 'connection' attribute is used to indicate whether to use an existing TCP connection, or create a new one. The SDP 'setup' attribute 'holdconn' value MUST NOT be used.

NOTE: A change of the TCP roles will also trigger a closure of the DTLS association, and establishment of a new DTLS association, according to the procedures in [I-D.ietf-mmusic-dtls-sdp].

NOTE: As specified in [I-D.ietf-mmusic-dtls-sdp], usage of the SDP 'setup' attribute 'holdconn' value is not allowed. Therefore this specification also forbids usage of the attribute value for TCP, as DTLS is transported on top of TCP.

10. SDP Offer/Answer Procedures

10.1. General

This section defines the SDP Offer/Answer [RFC3264] procedures for negotiating and establishing an SCTP-over-DTLS association. Unless explicitly stated, the procedures apply to both the 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' m- line proto values.

Each endpoint MUST associate one or more certificate fingerprints, using the SDP 'fingerprint' attribute with the m- line, following the procedures in [RFC8122].

The authentication certificates are interpreted and validated as defined in [RFC8122]. Self-signed certificates can be used securely, provided that the integrity of the SDP description is assured as defined in [RFC8122].

Each endpoint MUST associate an SDP 'tls-id' attribute with the m-line, following the procedures in [I-D.ietf-mmusic-dtls-sdp].

10.2. Generating the Initial SDP Offer

When the offerer creates an initial offer, the offerer:

- o MUST associate an SDP setup attribute with the m- line;
- o MUST associate an SDP 'sctp-port' attribute with the m- line;
- o MUST, in the case of TCP/DTLS/SCTP, associate an SDP 'connection' attribute, with a 'new' attribute value, with the m- line; and
- o MAY associate an SDP 'max-message-size' attribute [Section 6] with the m- line.

10.3. Generating the SDP Answer

When the answerer receives an offer, which contains an m- line describing an SCTP-over-DTLS association, if the answerer accepts the association, the answerer:

- o MUST insert a corresponding m- line in the answer, with an m- line proto value [RFC3264] identical to the value in the offer;
- o MUST associate an SDP 'setup' attribute with the m- line;
- o MUST associate an SDP 'sctp-port' attribute with the m- line. If the offer contained a new (different than the one currently used) SCTP port value the answerer MUST also associate a new SCTP port value. If the offer contained a zero SCTP port value, or if the answerer does not accept the SCTP association, the answerer MUST also associate a zero SCTP port value; and
- o MAY associate an SDP 'max-message-size' attribute [Section 6] with the m- line. The attribute value in the answer is independent from the value (if present) in the corresponding m- line of the offer.

Once the answerer has sent the answer the answerer:

- o MUST, in the case of TCP/DTLS/SCTP, if a TCP connection has not yet been established, or if an existing TCP connection is to be closed and replaced by a new TCP connection, follow the procedures in [RFC4145] for closing and establishing a TCP connection;
- o MUST, if a DTLS association has not yet been established, or if an existing DTLS association is to be closed and replaced by a new DTLS association, follow the procedures in

[I-D.ietf-mmusic-dtls-sdp] for closing the currently used, and establishing a new, DTLS association; and

- MUST, if an SCTP association has not yet been established, or if an existing SCTP association is to be closed and replaced by a new SCTP association, initiate the closing of the existing SCTP association (if applicable) and establishment of the SCTP association.

If the SDP 'sctp-port' attribute in the answer contains a zero attribute value, the answerer MUST NOT establish an SCTP association. If an SCTP association exists, the offerer MUST close it.

If the answerer does not accept the m- line in the offer, it MUST assign a zero port value to the corresponding m- line in the answer, following the procedures in [RFC3264]. In addition, the answerer MUST NOT initiate the establishment of a TCP connection, a DTLS association or a DTLS association associated with the m- line.

10.4. Offerer Processing of the SDP Answer

Once the offerer has received the answer the offerer:

- MUST, in the case of TCP/DTLS/SCTP, if a TCP connection has not yet been established, or if an existing TCP connection is to be closed and replaced by a new TCP connection, follow the procedures in [RFC4145] for closing and establishing a TCP connection;

- MUST, if a DTLS association has not yet been established, or if an existing DTLS association is to be closed and replaced by a new DTLS association, follow the procedures in [I-D.ietf-mmusic-dtls-sdp] for closing and establishing a DTLS association; and

- MUST, if an SCTP association has not yet been established, or if an existing SCTP association is to be closed and replaced by a new SCTP association, initiate the closing of the existing SCTP association (if applicable) and establishment of the SCTP association.

If the SDP 'sctp-port' attribute in the answer contains a zero attribute value, the offerer MUST NOT establish an SCTP association. If an SCTP association exists in that case, the offerer MUST close it.

If the m- line in the answer contains a zero port value, the offerer MUST NOT initiate the establishment a TCP connection, a DTLS association or an SCTP association associated with the m- line. If a

TCP connection, or a DTLS association or an SCTP association exists in that case, the offerer MUST close it.

10.5. Modifying the Session

When an offerer sends an updated offer, in order to modify a previously established SCTP association, it follows the procedures in Section 10.2, with the following exceptions:

- o If the offerer wants to close an SCTP association, and immediately establish a new SCTP association, the offerer MUST associate an SDP 'sctp-port' attribute with a new (different than the one currently used) attribute value. This will not impact the underlying DTLS association (and TCP connection in case of TCP/DTLS/SCTP).

- o If the offerer wants to close an SCTP association, without immediately establishing a new SCTP association, the offerer MUST associate an SDP 'sctp-port' attribute with a zero attribute value. This will not impact the underlying DTLS association (and TCP connection in case of TCP/DTLS/SCTP).

- o If the offerer wants to establish an SCTP association, and another SCTP association was previously closed, the offerer MUST associate an SDP 'sctp-port' attribute with a new attribute value (different than the value associated with the previous SCTP association). If the previous SCTP association was closed successfully following use of an SDP 'sctp-port' attribute with a zero attribute value, the offerer MAY use the same attribute value for the new SCTP association that was used with the previous SCTP association before it was closed. This will not impact the underlying DTLS association (and TCP connection in case of TCP/DTLS/SCTP).

- o If the offerer wants to close an existing SCTP association, and the underlying DTLS association (and the underlying TCP connection in case of TCP/DTLS/SCTP) it MUST assign a zero port value to the m- line associated with the SCTP and DTLS associations (and TCP connection in case of TCP/DTLS/SCTP), following the procedures in [RFC3264].

- o NOTE: This specification does not define a mechanism for explicitly closing a DTLS association while maintaining the overlying SCTP association. However, if a DTLS association is closed and replaced with a new DTLS association, as a result of some other action [I-D.ietf-mmusic-dtls-sdp], the state of the SCTP association is not affected.

The offer follows the procedures in [I-D.ietf-mmusic-dtls-sdp] regarding the DTLS association impacts when modifying a session.

In the case of TCP/DTLS/SCTP, the offerer follows the procedures in [RFC4145] regarding the TCP connection impacts when modifying a session.

11. Multihoming Considerations

Multihoming is not supported when sending SCTP on top of DTLS, as DTLS does not expose address management of the underlying transport protocols (UDP or TCP) to its upper layer.

12. NAT Considerations

12.1. General

When SCTP-over-DTLS is used in NAT environment, it relies on the NAT traversal procedures for the underlying transport protocol (UDP or TCP).

12.2. ICE Considerations

When SCTP-over-DTLS is used with UDP based ICE candidates [RFC5245] then the procedures for UDP/DTLS/SCTP [Section 7] are used.

When SCTP-over-DTLS is used with TCP based ICE candidates [RFC6544] then the procedures for TCP/DTLS/SCTP [Section 8] are used.

In ICE environments, during the nomination process, endpoints go through multiple ICE candidate pairs, until the most preferred candidate pair is found. During the nomination process, data can be sent as soon as the first working candidate pair is found, but the nomination process still continues and selected candidate pairs can still change while data is sent. Furthermore, if endpoints roam between networks, for instance when mobile endpoint switches from mobile connection to WiFi, endpoints will initiate an ICE restart, which will trigger a new nomination process between the new set of candidates and likely result in the new nominated candidate pair.

Implementations MUST treat all ICE candidate pairs associated with an SCTP association on top of a DTLS association as part of the same DTLS association. Thus, there will only be one SCTP handshake and one DTLS handshake even if there are multiple valid candidate pairs, and shifting from one candidate pair to another, including switching between UDP to TCP candidate pairs, will not impact the SCTP or DTLS associations. If new candidates are added, they will also be part of the same SCTP and DTLS associations. When transitioning between candidate pairs, different candidate pairs can be currently active in different directions and implementations MUST be ready to receive data on any of the candidates, even if this means sending and receiving data using UDP/DTLS/SCTP and TCP/DTLS/SCTP at the same time in different directions.

In order to maximize the likelihood of interoperability between the endpoints, all ICE enabled SCTP-over-DTLS endpoints SHOULD implement support for UDP/DTLS/SCTP.

When an SDP offer or answer is sent with multiple ICE candidates during initial connection negotiation or after ICE restart, UDP based candidates SHOULD be included and default candidate SHOULD be chosen from one of those UDP candidates. The proto value MUST match the transport protocol associated with the default candidate. If UDP transport is used for the default candidate, then 'UDP/DTLS/SCTP' proto value MUST be used. If TCP transport is used for the default candidate, then 'TCP/DTLS/SCTP' proto value MUST be used. Note that under normal circumstances the proto value for offers and answers sent during ICE nomination SHOULD be 'UDP/DTLS/SCTP'.

When a subsequent SDP offer or answer is sent after ICE nomination is complete, and does not initiate ICE restart, it will contain only the nominated ICE candidate pair. In this case, the proto value MUST match the transport protocol associated with the nominated ICE candidate pair. If UDP transport is used for the nominated pair, then 'UDP/DTLS/SCTP' proto value MUST be used. If TCP transport is used for the nominated pair, then 'TCP/DTLS/SCTP' proto value MUST be used. Please note that if an endpoint switches between TCP-based and UDP-based candidates during the nomination process the endpoint is not required to send an SDP offer for the sole purpose of keeping the proto value of the associated m- line in sync.

NOTE: The text in the paragraph above only applies when the usage of ICE has been negotiated. If ICE is not used, the proto value MUST always reflect the transport protocol used at any given time.

13. Examples

13.1. Establishment of UDP/DTLS/SCTP association

SDP Offer:

```
m=application 54111 UDP/DTLS/SCTP webrtc-datachannel
c=IN IP6 2001:DB8::A8FD
a=tls-id:abc3de65cddef001be82
a=setup:actpass
a=sctp-port:5000
a=max-message-size:100000
```

- The offerer indicates that the usage of the UDP/DTLS/SCTP association will be as defined for the 'webrtc-datachannel' format value.
- The offerer UDP port value is 54111.
- The offerer SCTP port value is 5000.
- The offerer indicates that it can take either the client or the server DTLS role.

SDP Answer:

```
m=application 64300 UDP/DTLS/SCTP webrtc-datachannel
c=IN IP6 2001:DB8::001D
a=tls-id:dbc8de77cddef001be90
a=setup:passive
a=sctp-port:6000
a=max-message-size:100000
```

- The answerer UDP port value is 64300.
- The answerer SCTP port value is 6000.
- The answerer takes the server DTLS role.

14. Security Considerations

[RFC4566] defines general SDP security considerations, while [RFC3264], [RFC4145] and [RFC8122] define security considerations when using the SDP offer/answer mechanism to negotiate media streams.

[RFC4960] defines general SCTP security considerations and [I-D.ietf-tsvwg-sctp-dtls-encaps] defines security considerations when using SCTP on top of DTLS.

This specification does not introduce new security considerations in addition to those defined in the specifications listed above.

15. IANA Considerations

15.1. New SDP proto values

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

This document updates the "Session Description Protocol (SDP) Parameters" registry, following the procedures in [RFC4566], by adding the following values to the table in the SDP "proto" field registry:

Type	SDP Name	Reference
proto	UDP/DTLS/SCTP	[RFCXXXX]
proto	TCP/DTLS/SCTP	[RFCXXXX]

Table 1: SDP "proto" field values

15.2. New SDP Attributes

15.2.1. sctp-port

This document defines a new SDP media-level attribute,'sctp-port'. The details of the attribute are defined in Section 5.2.

15.2.2. max-message-size

This document defines a new SDP media-level attribute,'max-message-size'. The details of the attribute are defined in Section 6.2.

15.3. association-usage Name Registry

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

This specification creates a new IANA registry, following the procedures in [RFC5226], for the namespace associated with the 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP' protocol identifiers. Each fmt value describes the usage of an entire SCTP association, including all SCTP streams associated with the SCTP association.

NOTE: Usage indication of individual SCTP streams is outside the scope of this specification.

The fmt value, "association-usage", used with these "proto" values is required. It is defined in Section 4.

As part of this registry, IANA maintains the following information:

association-usage name: The identifier of the subprotocol, as will be used as the fmt value.

association-usage reference: A reference to the document in which the association-usage is defined.

association-usage names are to be subject to the "First Come First Served" IANA registration policy [RFC5226].

IANA is asked to add initial values to the registry.

name	Reference
webrtc-datachannel	draft-ietf-rtcweb-data-protocol-xx, RFCXXX

[RFC EDITOR NOTE: Please hold the publication of this draft until draft-ietf-rtcweb-data-protocol has been published as an RFC. Then, replace the reference to draft-ietf-rtcweb-data-protocol with the RFC number.]

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

Figure 1

16. Acknowledgments

The authors wish to thank Harald Alvestrand, Randell Jesup, Paul Kyzivat, Michael Tuexen, Juergen Stoetzer-Bradler, Flemming Andreasen and Ari Keranen for their comments and useful feedback. Ben Campbell provided comments as part of his AD review. Brian Carpenter performed the Gen-ART review.

17.

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

Changes from draft-ietf-mmusic-sctp-sdp-25 o SDP ’dtls-id’ attribute re-named to ’tls-id’.

Changes from draft-ietf-mmusic-sctp-sdp-24

o Minor editorial fix by Roman.

Changes from draft-ietf-mmusic-sctp-sdp-23

o Changes based on IESG review.

o - Proto value clarifications.

Changes from draft-ietf-mmusic-sctp-sdp-22

o Changes based on Gen-ART review by Brian Carpenter.

Changes from draft-ietf-mmusic-sctp-sdp-21

o Changes based on AD review by Ben Campbell.

Changes from draft-ietf-mmusic-sctp-sdp-20

o Informative reference to draft-ietf-rtcweb-data-protocol added.

Changes from draft-ietf-mmusic-sctp-sdp-19

o Changes based on WG chair comments from Flemming Andreasen.

Changes from draft-ietf-mmusic-sctp-sdp-18

o Changes based on WGLC comments from Paul Kyzivat.

Changes from draft-ietf-mmusic-sctp-sdp-17

o Removal of ’SCTP’.

o Document title changed.

o Disallow usage of SDP ’setup’ attribute ’holdconn’ value.

o Roman Shpount added as co-editor.

Changes from draft-ietf-mmusic-sctp-sdp-15

o Chapter about SCTP, DTLS and TCP association/connection management modified.

o Removal of SCTP/DTLS.

Changes from draft-ietf-mmusic-sctp-sdp-14

- o Changes based on WGLC comments from Magnus Westerlund.
- o - ABNF clarification that token and port are defined in RFC4566.
- o - Specify 40 as maximum digit character length for the SDP max-message-size value.
- o - Editorial clarification.
- o Changes based on discussions at IETF#92.
- o - Specify that all ICE candidate pairs belong to the same DTLS association.

Changes from draft-ietf-mmusic-sctp-sdp-13

- o Changes based on comments from Paul Kyzivat.
- o - Text preventing usage of well-known ports removed.
- o - Editorial clarification.

Changes from draft-ietf-mmusic-sctp-sdp-12

- o Mux category rules added for new SDP attributes.
- o Reference to draft-ietf-mmusic-sdp-mux-attributes added.
- o Changes based on comments from Roman Shpount:
- o - Specify that fingerprint or setup roles must not be modified, unless underlying transport protocol is also modified.
- o Changes based on comments from Ari Keranen:
- o - Editorial corrections.
- o Changes based on comments from Flemming Andreasen:
- o - Clarify that, if UDP/DTLS/SCTP or TCP/DTLS/SCTP is used, the DTLS association is established before the SCTP association.
- o - Clarify that max-message-size value is given in bytes, and that different values can be used per direction.
- o - Section on fmtp attribute removed.

- - Editorial corrections.

Changes from draft-ietf-mmusic-sctp-sdp-11

- Example added.

Changes from draft-ietf-mmusic-sctp-sdp-10

- SDP max-message-size attribute added to IANA considerations.
- Changes based on comments from Paul Kyzivat:
- - Text about max message size removed from fmtp attribute section.

Changes from draft-ietf-mmusic-sctp-sdp-09

- 'DTLS/SCTP' split into 'UDP/DTLS/SCTP' and 'TCP/DTLS/SCTP'
- Procedures for realizing UDP/DTLS/SCTP- and TCP/DTLS/SCTP transports added.

Changes from draft-ietf-mmusic-sctp-sdp-08

- Default SCTP port removed:
- - Usage of SDP sctp-port attribute mandatory.
- SDP max-message-size attribute defined:
- - Attribute definition.
- - SDP Offer/Answer procedures.
- Text about SDP direction attributes added.
- Text about TLS role determination added.

18. References

18.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <http://www.rfc-editor.org/info/rfc793>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://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, <http://www.rfc-editor.org/info/rfc3264>.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <http://www.rfc-editor.org/info/rfc4145>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, DOI 10.17487/RFC4571, July 2006, <http://www.rfc-editor.org/info/rfc4571>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://www.rfc-editor.org/info/rfc4960>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC6544] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, DOI 10.17487/RFC6544, March 2012, <http://www.rfc-editor.org/info/rfc6544>.

[RFC8122] Lennox, J. and C. Holmberg, "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 8122, DOI 10.17487/RFC8122, March 2017, <http://www.rfc-editor.org/info/rfc8122>.

[I-D.ietf-mmusic-dtls-sdp]
Holmberg, C. and R. Shpount, "Using the SDP Offer/Answer Mechanism for DTLS", draft-ietf-mmusic-dtls-sdp-24 (work in progress), April 2017.

[I-D.ietf-tsvwg-sctp-dtls-encaps]
Tuexen, M., Stewart, R., Jesup, R., and S. Loreto, "DTLS Encapsulation of SCTP Packets", draft-ietf-tsvwg-sctp-dtls-encaps-09 (work in progress), January 2015.

[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.

18.2. Informative References

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, DOI 10.17487/RFC5245, April 2010, <http://www.rfc-editor.org/info/rfc5245>.

[I-D.ietf-rtcweb-data-channel]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-13 (work in progress), January 2015.

[I-D.ietf-mmusic-data-channel-sdpneg]
Drage, K., Makaraju, M., Stoetzer-Bradler, J., Ejzak, R., and J. Marcon, "SDP-based Data Channel Negotiation", draft-ietf-mmusic-data-channel-sdpneg-12 (work in progress), March 2017.

[I-D.ietf-rtcweb-data-protocol]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel Establishment Protocol", draft-ietf-rtcweb-data-protocol-09 (work in progress), January 2015.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Roman Shpount
TurboBridge
4905 Del Ray Avenue, Suite 300
Bethesda, MD 20814
USA

Phone: +1 (240) 292-6632
Email: rshpount@turbobridge.com

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Salvatore.Loreto@ericsson.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

Network Working Group M. Petit-Huguenin
Internet-Draft Stonyfish, Inc.
Updates: 5245 (if approved) June 30, 2011
Intended status: Standards Track
Expires: January 1, 2012

Media level ice-options SDP attribute
draft-petithuguenin-mmusic-ice-attributes-level-01

Abstract

This document redefines the ice-options SDP attribute as a session-level and media-level attribute.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

ICE [RFC5245] defines the ice-options SDP attribute as session-level only attribute, but when ICE is used with disaggregated media (see section 3 of [I-D.loreto-splices-disaggregated-media]), there is a possibility that different media use different ICE implementations and/or different networks, and so these different media will require different values for this attribute.

As an example, the ice-options attribute value "rtp+ecn" (defined in [I-D.ietf-avtcore-ecn-for-rtp]) signals ECN capability. Two aggregated media using two different RTP implementations may want to use different values for this attribute.

Note that there is a similar problem for the ice-lite attribute but unfortunately it does not seem possible to design a way to use the ice-lite attribute at the media level that is compatible with legacy implementations that recognize only the session-level attribute.

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].

3. The ice-options Attribute

The ice-options attribute is redefined by this document as a session-level and media-level attribute.

All future new ICE options MUST also define how media-level ICE options using this new value are aggregated to eventually generate the value of the session-level ICE option, so legacy implementations that only recognize session-level ICE options can interoperate with implementations that recognize ICE options at both levels.

Before applying this specific aggregation rule, the session-level ice-options attribute MUST be copied as media-level attribute in each media.

4. Specific Aggregation Rule for the rtp+ecn ICE Option

If all aggregated media using ICE contain a media-level "rtp+ecn" ICE option, as defined by [I-D.ietf-avtcore-ecn-for-rtp], then an "rtp+ecn" ICE option MUST be inserted at the session-level.

5. Security Considerations

This document does not add any security considerations beyond what is discussed in [RFC5245].

6. IANA Considerations

No IANA considerations.

7. Acknowledgements

This document was written with the xml2rfc tool described in [RFC2629].

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.

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

[I-D.ietf-avtcore-ecn-for-rtp] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", draft-ietf-avtcore-ecn-for-rtp-02 (work in progress), May 2011.

8.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[I-D.loreto-splices-disaggregated-media] Camarillo, G., Loreto, S., and R. Shekh-Yusef, "Disaggregated Media in the Session Initiation Protocol (SIP)", draft-loreto-splices-disaggregated-media-02 (work in progress), June 2011.

Appendix A. Examples

A.1. Aggregating media all supporting ICE

In this example, we have two SDP to aggregate. The first SDP contains an ice-options attribute at the media level:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

The second SDP also have an ice-options attribute at the media level:

```
v=0
o=jdoe 1 1 IN IP4 10.0.1.2
s=
c=IN IP4 192.0.2.4
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:f7sD7f7dF87s87d7da5564
a=ice-ufrag:776G
m=video 10000 RTP/AVP
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.2 10000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.4 45000 typ srflx raddr
  10.0.1.1 rport 10000
```

The first step is to copy the session-level ice-options attribute as media-level attribute. The first SDP is modified like this:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

The second SDP is modified like this:

```
v=0
o=jdoe 1 1 IN IP4 10.0.1.2
s=
c=IN IP4 192.0.2.4
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:f7sD7f7dF87s87d7da5564
a=ice-ufrag:776G
m=video 10000 RTP/AVP
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.2 10000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.4 45000 typ srflx raddr
  10.0.1.1 rport 10000
```

After aggregation, all the individual media keep their media-level ice-options attribute, and a session-level ice-options attribute is added as per the rule in Section 3:

```
v=0
o=- 1309452627 1309452627 IN IP4 10.0.1.1
s=
t=0 0
a=ice-options:rtp+ecn
m=audio 45664 RTP/AVP 0
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
m=video 10000 RTP/AVP
c=IN IP4 192.168.2.4
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.2 10000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.4 45000 typ srflx raddr
  10.0.1.1 rport 10000
```

A.2. Aggregating media partially supporting ICE

In this example, we have two SDP to aggregate, but the second one does not use ICE. The first SDP contains an ice-options attribute at the media level:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

The second SDP does not contain any ice-options attribute:

```
v=0
o=jdoe 1 1 IN IP4 10.0.1.2
s=
c=IN IP4 192.0.2.4
t=0 0
m=video 10000 RTP/AVP
a=rtpmap:0 PCMU/8000
```

The first step is to copy the session-level ice-options attribute as media-level attribute. Only the first SDP is modified in this example:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

After aggregation, all the individual media keep their media-level ice-options attribute, and a session-level ice-options attribute is added as per the rule in Section 3:

```
v=0
o=- 1309452627 1309452627 IN IP4 10.0.1.1
s=
t=0 0
a=ice-options:rtp+ecn
m=audio 45664 RTP/AVP 0
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
m=video 10000 RTP/AVP
c=IN IP4 192.168.2.4
a=rtpmap:0 PCMU/8000
```

Appendix B. Release notes

This section must be removed before publication as an RFC.

B.1. Modifications between -01 and -00

- Changed the rtp+ecn aggregation rule so that non-ICE media are not used when aggregating.
- Filled Security and IANA sections.
- Added examples of aggregation.
- Added a design note about using different attribute name at media level.

B.2. Design Notes

- o It has been proposed multiple times to use a different attribute name for the ice-options attribute when used at the media-level. Using a different name does not solve the aggregation problem and, in the opinion of this author, could create confusion.

Author's Address

Marc Petit-Huguenin
Stonyfish, Inc.

Email: petithug@acm.org

MMUSIC M. Petit-Huguenin
Internet-Draft Impedance Mismatch
Updates: 5245 (if approved) October 8, 2012
Intended status: Standards Track
Expires: April 11, 2013

Media level ice-options SDP attribute
draft-petithuguenin-mmusic-ice-attributes-level-04

Abstract

This document normatively updates RFC 5245 by redefining the ice-options SDP attribute as a session-level and media-level attribute.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

ICE [RFC5245] defines the ice-options SDP attribute as session-level only attribute, but when ICE is used with disaggregated media (see section 3 of [I-D.loreto-splices-disaggregated-media]), there is a possibility that different media use different ICE implementations and/or different networks, and so these different media will require different values for this attribute.

As an example, the ice-options attribute value "rtp+ecn" (defined in [RFC6679]) signals ECN capability. Two aggregated media using two different RTP implementations may want to use different values for this attribute.

Note that there is a similar problem for the ice-lite attribute but unfortunately it does not seem possible to design a way to use the ice-lite attribute at the media level that is compatible with legacy implementations that recognize only the session-level attribute.

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] .

3. The ice-options Attribute

The ice-options attribute is redefined by this document as a session-level and media-level attribute.

All future new ICE options MUST also define how media-level ICE options using this new value are aggregated to eventually generate the value of the session-level ICE option, so legacy implementations that only recognize session-level ICE options can interoperate with implementations that recognize ICE options at both levels.

Before applying this specific aggregation rule, the session-level ice-options attribute MUST be copied as media-level attribute in each media.

4. The ice-lite Attribute

The ice-lite attribute is not redefined by this specification.

5. The ice-mismatch Attribute

[RFC5245] section 15.3 defines this attribute as been media level, which seems correct, but section 21.1.4 erroneously registered this attribute in IANA as session level. An errata [1] has been filled and the IANA registry has been accordingly fixed.

6. Specific Aggregation Rule for the rtp+ecn ICE Option

If all aggregated media using ICE contain a media-level "rtp+ecn" ICE option, as defined by [RFC6064] , then an "rtp+ecn" ICE option MUST be inserted at the session-level.

7. Security Considerations

This document does not add any security considerations beyond what is discussed in [RFC5245] .

8. IANA Considerations

No IANA considerations.

9. Acknowledgements

This document was written with the xml2rfc tool described in [RFC2629] .

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.

[RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.

10.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

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

[RFC5159] Dondeti, L. and A. Jerichow, "Session Description Protocol (SDP) Attributes for Open Mobile Alliance (OMA) Broadcast (BCAST) Service and Content Protection", RFC 5159, March 2008.

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

[RFC6064] Westerlund, M. and P. Frojdh, "SDP and RTSP Extensions Defined for 3GPP Packet-Switched Streaming Service and Multimedia Broadcast/Multicast Service", RFC 6064, January 2011.

[I-D.loreto-splices-disaggregated-media] Camarillo, G., Loreto, S., and R. Shekh-Yusef, "Disaggregated Media in the Session Initiation Protocol (SIP)", draft-loreto-splices-disaggregated-media-02 (work in progress), June 2011.

URIs

[1] <http://www.rfc-editor.org/errata_search.php?eid=3149>

Appendix A. Examples

A.1. Aggregating media all supporting ICE

In this example, we have two SDP to aggregate. The first SDP contains an ice-options attribute at the media level:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

The second SDP also have an ice-options attribute at the media level:

```
v=0
o=jdoe 1 1 IN IP4 10.0.1.2
s=
c=IN IP4 192.0.2.4
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:f7sD7f7dF87s87d7da5564
a=ice-ufrag:776G
m=video 10000 RTP/AVP
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.2 10000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.4 45000 typ srflx raddr
  10.0.1.1 rport 10000
```

The first step is to copy the session-level ice-options attribute as media-level attribute. The first SDP is modified like this:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

The second SDP is modified like this:

```
v=0
o=jdoe 1 1 IN IP4 10.0.1.2
s=
c=IN IP4 192.0.2.4
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:f7sD7f7dF87s87d7da5564
a=ice-ufrag:776G
m=video 10000 RTP/AVP
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.2 10000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.4 45000 typ srflx raddr
  10.0.1.1 rport 10000
```

After aggregation, all the individual media keep their media-level ice-options attribute, and a session-level ice-options attribute is added as per the rule in Section 3 :

```
v=0
o=- 1309452627 1309452627 IN IP4 10.0.1.1
s=
t=0 0
a=ice-options:rtp+ecn
m=audio 45664 RTP/AVP 0
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
m=video 10000 RTP/AVP
c=IN IP4 192.168.2.4
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.2 10000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.4 45000 typ srflx raddr
  10.0.1.1 rport 10000
```

A.2. Aggregating media partially supporting ICE

In this example, we have two SDP to aggregate, but the second one does not use ICE. The first SDP contains an ice-options attribute at the media level:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

The second SDP does not contain any ice-options attribute:

```
v=0
o=jdoe 1 1 IN IP4 10.0.1.2
s=
c=IN IP4 192.0.2.4
t=0 0
m=video 10000 RTP/AVP
a=rtpmap:0 PCMU/8000
```

The first step is to copy the session-level ice-options attribute as media-level attribute. Only the first SDP is modified in this example:

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
```

After aggregation, all the individual media keep their media-level ice-options attribute, and a session-level ice-options attribute is added as per the rule in Section 3 :

```
v=0
o=- 1309452627 1309452627 IN IP4 10.0.1.1
s=
t=0 0
a=ice-options:rtp+ecn
m=audio 45664 RTP/AVP 0
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
m=text 45666 RTP/AVP 98
c=IN IP4 192.168.2.3
b=RS:0
b=RR:0
a=rtpmap:98 t140/1000
a=ice-options:rtp+ecn
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=candidate:1 1 UDP 2130706431 10.0.1.1 9000 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45666 typ srflx raddr
  10.0.1.1 rport 9000
m=video 10000 RTP/AVP
c=IN IP4 192.168.2.4
a=rtpmap:0 PCMU/8000
```

Appendix B. Analysis of similar issues in other attributes

In the MMUSIC WG session in Quebec City, it was suggested that the problem was perhaps larger than just ICE attributes. This section is the result of a systematic look at all the session level only SDP attributes registered by IANA at the date of this document. The conclusion is that only the ICE attributes are of concern but that steps should be taken to ensure that these problems cannot happen for future new attributes.

Attribute	Reference	Comments
cat	[RFC4566]	OK
keywds	[RFC4566]	OK
type	[RFC4566]	OK
type:broadcast	[RFC4566]	Appendix B.1
type:H332	[ITU Recommendation H.332]	OK
type:meeting	[RFC4566]	OK
type:moderated	[RFC4566]	OK
type:test	[RFC4566]	OK
charset	[RFC4566]	Appendix B.2
charset:iso8895-1	[RFC4566]	Appendix B.2
tool	[RFC4566]	Appendix B.3
ipbcp	[ITU-T Q.1970]	Appendix B.4
group	[RFC5888]	OK
ice-lite	[RFC5245]	Section 4
ice-mismatch	[RFC5245]	Section 5
ice-options	[RFC5245]	Section 3
bcastversion	[RFC5159]	Appendix B.4
3GPP-Integrity-Key	[RFC6064]	Appendix B.4
3GPP-SDP-Auth	[RFC6064]	Appendix B.4
alt-group	[RFC6064]	Appendix B.4
PSCid	[TS 183 063]	Appendix B.4
bc_service	[TS 183 063]	Appendix B.4
bc_program	[TS 183 063]	Appendix B.4
bc_service_package	[TS 183 063]	Appendix B.4

B.1. The type:broadcast Attribute

The "type:broadcast" does not have any issue by itself, but it should be noted that it implies a default attribute of recvonly, so the disaggregation process must take this in account.

B.2. The charset Attribute

Because the main reason to use a different charset for a session description is to generate a more compact representation, it is probably OK that this attribute exists only at the session level. But the aggregation/disaggregation rules must explicitly convert the individual media descriptions to and from the common charset, ISO-10646.

B.3. The tool Attribute

The definition of this attribute make it clear that this attribute contains the name and version number of the tool that created the session description as a whole. But it probably would be useful to define this attribute at the media level too, so we can know the tools used for create the individual media descriptions.

B.4. The ipbcp, bcastversion, 3GPP-Integrity-Key, 3GPP-SDP-Auth, alt-group, PSCid, bc_service, bc_program and bc_service_package Attributes

These attributes were not defined in IETF Standard Track documents, so the analysis is left to the SDOs that produced this specifications.

Appendix C. Release notes

This section must be removed before publication as an RFC.

C.1. Modifications between -04 and -03

- o Updated rtp+ecn reference.
- o IANA registry for ice-mismatch fixed.

C.2. Design Notes

- o It has been proposed multiple times to use a different attribute name for the ice-options attribute when used at the media-level. Using a different name does not solve the aggregation problem and, in the opinion of this author, could create confusion.

Author's Address

Marc Petit-Huguenin
Impedance Mismatch

Email: petithug@acm.org

Network WG James Polk
Internet-Draft Subha Dhesikan
Expires: January 11, 2012 Paul Jones
Intended Status: Standards Track (PS) Cisco Systems
July 11, 2011

The Session Description Protocol (SDP) 'trafficclass' Attribute
draft-polk-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.

Legal

Status of this Memo

Table of Contents

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 perhaps makes up a network such as a campus or WAN, or a backbone. For example, a call center voice application gets at least 20% of a link as a minimum 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 to operate at 1080p during non-peak times, but curtail a similar session between the same users or devices to 720p or 360p during peak hours. 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 telepresense 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.

A question arises about once we properly label the traffic, what does that get us? This is a fair question, but out of scope for this document because that answer lies within other RFCs and IDs in other WGs and/or Areas (specifically the Transport Area). That said, we can discuss some of the ideas here for completeness.

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). 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. Communications between enterprise networks will likely have to take advantage of MPLS.

Within a network core, where only MPLS is used, Diffserv typically does not apply. That said, Diffserv can be used to identify which traffic goes into which MPLS tunnels [RFC4124].

Labeling realtime traffic types using a layer 4 protocol would likely mean 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 data path. An advantage with this mechanism is for the label to inform each domain along the media path what type of traffic this traffic flow is, and allow each domain to adjust the appropriate DSCP (set by each domain for use within that domain). Meaning, if a DSCP is set by an endpoint or a router in the first domain and gets reset by a SP, the far end domain will be able to reset the DSCP 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 to 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 but one reason given). In addition, RFC 5897 recommends that service identification should be done in signaling, rather than guessing or deep packet inspection. The network will have to currently guess or perform deep packet inspection to classify 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 is only wanting to 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 parent 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.

The traffic class label will have the following structure,

parent.application(.adjective)(.adjective)(.admitted/non-admitted)

[Editor’s Note: the above is not exactly the ANBF to be used. The order is right. The parent and application MUST appear (each only once) and zero or more adjectives can appear.]

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; and
4) an optional 4th component is to classify this flow as a CAC admitted or non-admitted traffic flow. The default is non-admitted, whether present or not.

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

    Conversational.video.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 parent 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 parent components will be, as well as at least some application components per parent component. Some adjective components will also likely be proposed for IANA registration.

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

o Conversational
o Multimedia Conferencing
o Real-Time 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 Web (conference) chat/instant messaging
o Gaming

- o Virtual-desktop (interactive)
- o Remote-desktop
- o Telemetry (e.g., NORAD missile control)
- o Multiplex (i.e., combined streams)
- o (something to cover theater quality Netflix movies)
- o (something to cover YouTube)
- o Webcast
- o IPTV
- o Live-events (though not the buffered ones)
- o surveillance

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

- o Immersive
- o Desktop-video
- o Realtime-Text
- o web

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

2.1 Conversational Parent Traffic Class

The Conversational traffic class is best suited for applications that require very low delay variation and generally intended to enable real-time, bi-directional person-to-person or multi-directional via an MTP communication, such as the following application components:

- o Audio (voice)

- o Video

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

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

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.

Desktop-video - 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

The above substrings might also be used within Multimedia Conferencing

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

2.2 Multimedia-Conferencing Parent 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, such as the following application components:

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

2.3 Realtime-Interactive Parent Traffic Class

Realtime-Interactive traffic class is intended for interactive variable rate inelastic applications that require low jitter and loss and very low delay, such as the following application components:

- o Gaming - interactive player video games with other users on other hosts (e.g., Doom)
- o Virtual desktop (interactive) - similar to an X-windows station, has no local harddrive, 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	Tolerance to Delay	Tolerance to Jitter
Realtime Interactive	Inelastic, mostly variable rate, rate increases with user activity	Low	Very Low	Low

2.4 Multimedia-Streaming Parent 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, such as the following application components:

o Audio

o Video

o Multiplex (i.e., combined streams)

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

(something to cover theater quality Netflix movies)

(something to cover YouTube)

Webcast

The primary difference from the Multimedia-streaming parent class and the Broadcast parent class is about the length of time for buffering. Buffered streaming audio and/or video (e.g., Netflix or previously-recorded videos on web sites like CNN, ESPN or from an internal corporate server). Buffering here can be from seconds to hours (as opposed to Broadcast buffering which is minimal). The buffering aspect is what differentiates this parent class from the Broadcast class (which has minimal or no buffering).

Traffic Class Name	Traffic Characteristics	Tolerance to Loss	Tolerance to Delay	Tolerance to Jitter
Multimedia Streaming	Variable size packets, elastic with variable rate	Low - Medium	Medium - High	High

2.5 Broadcast Parent Traffic Class

Broadcast traffic class is best suited for inelastic streaming media applications that may be of constant or variable rate, requiring low jitter and very low packet loss, such as the following application components:

o IPTV

o Live events (though not the buffered ones)

o Video surveillance

Traffic Class Name	Traffic Characteristics	Tolerance to Loss	Tolerance to Delay	Tolerance to Jitter
Broadcast	Constant and variable rate, inelastic, generally non-bursty flows, generally sustained high packet rate, low inter-packet transmission interval, usually UDP framed in (S)RTP	Very Low	Low - Medium	Low - Medium

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-classification

traffic-classification  = "trafficclass" ":" [SP] parent-class
                           "." app-type *( app-param )

parent-class            = "Broadcast" /
                          "Realtime-Interactive" /
                          "Multimedia-Conferencing" /
                          "Multimedia-Streaming" /
                          "Conversational" /
                          extension-mech

extension-mech          = token

app-type                = "audio" / "video" / "text" /
                          "application-sharing" /
                          "presentation-data" /
                          "whiteboarding" / "webchat/IM" /
                          "gaming" / "virtual-desktop" /
                          "remote-desktop" / "telemetry"/
                          "multiplex" / "Netflix" / "youtube" /
                          "webcast" / "IPTV" / "live-events" /
                          "surveillance"

app-param               = "." adjective / "." cac-class

adjective               = "immersive" / "desktop-video" /
                          "Realtime-Text" /"web" /
                           generic-param ; from RFC3261

cac-class               = "admitted" / "non-admitted"
```

The attribute is named "trafficclass", for traffic classification, identifying which one of the five traffic classes applies to the media stream. There MUST NOT be more than one trafficclass attribute per media line. Confusion would result in where more than one exists per m= line.

The parent classes 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 Defined in RFC 4594	Parent Classes Defined in this document
Broadcast-video	Broadcast
Realtime-Interactive	Realtime-Interactive
Multimedia-Conferencing	Multimedia-Conferencing
Multimedia-Streaming	Multimedia-Streaming
Telephony	Conversational

Table 6. Label Changes 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 parent class is a human understandable categorization, and MUST NOT be the only part of the traffic class label present in the attribute. The parent class string MUST always be paired with an application type, with a "." as the string separator.

The application types (app-type) define the application of a particular traffic flow. The application types are listed both in the ABNF and defined in Section 2 of this document. Not every combination parent class is paired with application types, 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 strings 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 label, but these MUST be prefaced by the additional delimiter "_", creating a possibility such as

```
 parent-class.application-type.adjective._non-standard-adjective
                                        ^^^^
                                   See the underscore
```

For example, this is valid:

```
m=audio 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 is not permitted, such as the following:

```
m=audio 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 components matters, but only for the components. In other words, the parent class component MUST be before the application component, which MUST be before the adjective component, which MUST be before the cac-class component. If there are no adjective components, the cac-class component is immediately after the application component.

If there is more than one adjective component describing a traffic class, the order of the adjectives MUST NOT matter. Some algorithm such as alphabetizing the list and matching the understood strings SHOULD be used.

In addition to, or as an alternative to one or more adjectives , a cac-class value MAY be present indicating whether or not a session has had call admission control applied to it. The following two values are created by this document for the cac-class value:

- admitted
- nonadmitted

The default cac-class value for any trafficclass attribute is nonadmitted, even if not present. There MUST NOT be more than one cac-class sub-string per m=line.

Any application, adjective or cac-class 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.

Not understanding the parent class string SHOULD mean that this attribute is ignored.

The following is an example of media level description with a 'trafficclass' attribute:

```
m=audio 50000 RTP/AVP 112
a=trafficclass conversational.video.immersive.admitted
```

The above indicates a multiscreen telepresence session that has had call admission control applied to the 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 converstaional 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 well 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). SBCs 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 parent 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 parent component was not understand, i.e., the attribute SHOULD be ignored.

If yes, it checks there are any other component 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, see if there is a cac-class component, and before processing classification.

If yes, determine if there are more than one. Alphabetize all of the adjective components and match the traffic classification.

4 - does the answerer understand the cac-class component if present?

If not, consider the media flow for this m= line to be nonadmitted.

If yes, classify whether this component is CAC admitted or nonadmitted.

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 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 mis-classifying (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 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 Type Attribute Values
Reference: [this document]
Registration Procedures: Specification Required

Parent 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/instant messaging	[this document]
Gaming	[this document]
Virtual-desktop	[this document]
Remote-desktop	[this document]
Telemetry	[this document]
Multiplex	[this document]
Netflix*	[this document]
YouTube*	[this document]
Webcast	[this document]
IPTV	[this document]
Live-events	[this document]
surveillance	[this document]

[Editor’s Note: these are placeholders until a more generic string can be agreed to by the WG]

6.4 The Traffic Classification Adjective 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 Adjective Values
Reference: [this document]
Registration Procedures: Specification Required

Application Values	Reference
Immersive	[this document]
Desktop-video	[this document]
Realtime-Text	[this document]
web	[this document]

6.5 The Traffic Classification Attribute Call Admission Control Class Registration

This document requests IANA to create a new registry for the Call Admission Control Class similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Call Admission Control Class (cac-class) Attribute Values
Reference: [this document]
Registration Procedures: Specification Required

Attribute Values	Reference
Admitted	[this document]
Non-admitted	[this document]

7. Acknowledgments

To Dave Oran, Toerless Eckert, Henry Chen, David Benham, David Benham, Mo Zanty, Michael Ramalho, Glen Lavers, Charles Ganzhorn, 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

[RFC4566] M. Handley, V. Jacobson, C. Perkins, "SDP: Session

Description Protocol", RFC 4566, July 2006

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010

[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

[RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997

[RFC4080] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005

[RFC2872] Y. Bernet, R. Pabbati, "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000

[RFC5897] J. Rosenberg, "Identification of Communications Services in the Session Initiation Protocol (SIP)", RFC 5897, June 2010

[RFC4124] F. Le Faucheur, Ed., " Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering ", RFC 4124, June 2005

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

8.2. Informative References

[RFC4594] J. Babiarz, 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2012

M. Westerlund
B. Burman
Ericsson

July 4, 2011

RTP Multiple Stream Sessions and Simulcast
draft-westerlund-avtcore-multistream-and-simulcast-00

Abstract

RTP has always been a protocol that supports multiple participants each sending their own media streams in an RTP session. Unfortunately many implementations aimed only at point to point voice over IP with a single source in each end-point. Even client implementations aimed at video conferences have often been built with the assumption around central mixers that only deliver a single media stream per media type. Thus any application that wants to allow for more advance usage where multiple media streams are sent and received by an end-point has a problem with legacy. This issue is analyzed, and RTP clarifications and signalling extensions are proposed to handle this issue. A related issue is how to perform simulcast, in the meaning of sending multiple encodings or representations of the same media source, when using RTP for media transport. This is further analyzed and possible solutions discussed and we arrive at a conclusion for session multiplexing of simulcast versions. We also found a number of related issues when having multiple streams and simulcast.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This document looks at the issues of non basic usage of RTP where there is multiple media sources sent over an RTP session. This include multiple sources from the same end-point, multiple end-points each having a source, or due to an application that needs multiple encodings of a particular source. As will be shown these issues are interrelated and need a common discussion to ensure consistency.

After presenting the usages and the found issues the document goes on to discuss ways of solving the issues. These include both clarifications to the basic RTP behaviors and signalling extensions to be able to setup these session, also in the presence of legacy systems that are not assumed to have full support for multiple media streams within an RTP session.

This document proposes several general mechanisms that could be used independently in other use cases. We foresee that those proposals would in the end become independent but related documents in the relevant WGs of AVTCORE, AVTEXT and MMUSIC. However, at this stage when all these ideas are introduced we find it more useful to keep them together to ensure consistency and to make any relations clear, hopefully making it easier to find and resolve any issues in the area of multiple streams and simulcast.

1.1. Multiple Streams

RTP sessions are a concept which most fundamental part is a SSRC space. This space can encompass a number of network nodes and interconnect transport flows between these nodes. Each node may have zero, one or more source identifiers (SSRCs) used to either identify a real media source such as a camera or a microphone, a conceptual source, like the most active speaker selected by a RTP mixer that switches between incoming media streams based on the media stream or additional information, or simply as an identifier for a receiver that provides feedback and reports on reception. There are also RTP nodes, like translators that are manipulating, data, transport or session state without making their presence aware to the other session participants.

RTP was designed with multiple participants in a session from the beginning. This was not restricted to multicast as many believe but also unicast using either multiple transport flows below RTP or a network node that redistributes the RTP packets, either unchanged in the form of a transport translator (relay) or modified in an RTP mixer. In addition a single end-point may have multiple media sources of the same media type, like cameras or microphones.

However, the most common use cases has been point to point Voice over IP (VoIP) or streaming applications where there has commonly not been more than one media source per end-point. Even in conferencing applications, especially voice only, the conference focus or bridge has provided a single stream being a mix of the other participants to each participant. Thus there has been perceived little need for handling multiple SSRCs in implementations. This has resulted in an installed legacy base that isn't fully RTP specification compliant and will have different issues if they receive multiple SSRCs of media, either simultaneously or in sequence. These issues will manifest themselves in various ways, either by software crashes, or simply in limited functionality, like only decoding and playing back the first or latest SSRC received and discarding any other SSRCs.

The signalling solutions around RTP, especially SDP based, hasn't considered the fundamental issues around RTP session's theoretical support of up to 4 billion plus sources all sending media. No end-point has infinite processing resources to decode and mix any number of sources with media. In addition the memory for storing related state, especially decoder state is limited, and the network bandwidth to receive multiple streams is also limited. Today, the most likely limitations are processing and network bandwidth, although for some use cases memory or other limitations may exist. The point is that a given end-point will have some limitations in the number of streams it simultaneously can receive, decode and playback. These limitations needs to be possible to expose and enabling the session participants to take them into account.

In similar ways there is a need for an end-point to express if it intends to produce one or more media stream. Todays SDP signalling support for this is basically the directionality attribute which indicates an end-point intend to send media or not. No indication of how many media streams.

Taking these things together there exist a clear need to enable the usage of multiple simultaneous media streams within an RTP session in a way that allows a system to take legacy implementations into account in addition to negotiate the actual capabilities around the multiple streams in an RTP session.

In addition to address the above set of issues we will also identify a number of issues related to multiple streams that should be addressed in the most suitable way. These include both obscurities in the RTP specification and short-comings in various signalling mechanisms that are exposed by multi-stream use cases.

1.2. Simulcast

Simulcast is the act of simultaneously sending multiple different versions of a media content. This can be done in several ways and for different purposes. This document focuses on the case where one wants to provide multiple different encodings towards a intermediary so that the intermediary can select which version to forward to other participants in the session. More discussion on the different ways of doing simulcast, which is the focus of this document in "Simulcast Usage and Applicability" (Section 3).

The different versions of a source content that can be simulcasted and that are considered in this document are:

Bit-rate: The primary difference is the amount of bits spent to encode the source and thus primarily affects the media signal to noise ratio (SNR).

Codec: Different media codecs are used to ensure that different receivers that do not have a common set of decoders can decode at least one of the versions. This includes codec configuration options that aren’t compatible, like video encoder profiles, or the capability of receiving the transport packetization.

Sampling: Different sampling of media, in spatial as well as in temporal domain, may be used to suit different rendering capabilities or needs at receiving endpoints, as well as a method to achieve different bit-rates. For video streams, spatial sampling affects image resolution, and temporal sampling affects video framerate. For audio, spatial sampling relates to the number of audio channels, and temporal sampling affects audio bandwidth.

Different applications will have different reasons for providing a single media source in different versions. And as soon as an application have need for multiple versions for some reason, a potential need for simulcast is created. This need can arise even in media codecs that have scalability features built in to solve a set of variations.

The purpose of this document is to find the most suitable solution for the non-trivial variants of simulcast. To determine this, an analysis of different ways of multiplexing the different encodings are discussed in Section 6. Following the presentation of the alternatives, an analysis is performed in Section 7 on how different aspects like RTP mechanisms, signaling possibilities, and network features are affected by the alternatives.

The document ends with a recommendation for which solution is the most suitable and indicates what standardization work should be done if the WG agrees on the analysis and the suitability to define how simulcast should be done.

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 RFC 2119 [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.

3. Simulcast Usage and Applicability

This section discusses different usage scenarios the term simulcast may refer to, and makes it clear which of those this document focuses on. It also reviews why simulcast and scalable codecs can be a useful combination.

3.1. Simulcasting to RTP Mixer

The usage here is in a multi-party session where one uses one or more central nodes to help facilitate the media transport between the session participants. Thus, this targets the RTP topology defined in [RFC5117] of RTP Mixer (Section 3.4: Topo-Mixer). This usage is one which is targeted for further discussion in this document.

Simulcasting different media encodings of video that has both different resolution and bit-rate is highly applicable to video conferencing scenarios. For example an RTP mixer selects the most active speaker and sends that participant’s media stream as a high resolution stream to a receiver and in addition provides a number of small resolution video streams of any additional participants, thus enabling the receiving user to both see the current speaker in high quality and monitor the other participants. The active speaker gets a different combination of streams as it has limited use to get back the streams itself is sending. Thus, there can be several different combinations of high resolution and low resolution video in use simultaneously; requiring both a high and low resolution video from some sources at the same time.

For example, to provide both high and low resolution from an RTP Mixer there exist these potential alternatives:

Simulcast: The client sends one stream for the low resolution and another for the high resolution.

Scalable Video Coding: Using a video encoder that can provide one media stream that is both providing the high resolution and enables the mixer to extract a low resolution representation that has lower bit-rate than the full stream version.

Transcoding in the Mixer: The client transmits a high resolution stream to the RTP Mixer, which performs a transcoding to a lower resolution version of the video stream that is forwarded to the ones that need it.

The Transcoding requires that the mixer has sufficient amounts of transcoding resources to produce the number of low resolution versions required. This may in worst case be that all participants' streams needs transcoding. If the resources are not available, a different solution needs to be chosen.

The scalable video encoding requires a more complex encoder compared to non-scalable encoding. Also, if the resolution difference is big, the scalable codec may in fact be only marginally more bandwidth efficient, between the encoding client and the mixer, than a simulcast that sends the resolutions in separate streams, assuming equivalent video quality. At the same time, with scalable video encoding, the transmission of all but the lowest resolution will definitely consume more bandwidth from the mixer to the other participants than a non-scalable encoding, again assuming equivalent video quality.

Simulcasting has the benefit that it is conceptually simple. It enables use of any media codec that the participants agree on, allowing the mixer to be codec-agnostic. Considering today's video encoders, it is less bit-rate efficient in the path from the sending client to the mixer but more efficient in the mixer to receiver path compared to Scalable Video Coding.

3.1.1. Simulcast Combined with Scalable Encoding

Scalable codecs are often used in arguments to motivate why simulcast isn't needed. A single media encoding that is sent as one joint media stream or divided up in base layers and enhancement layers over multiple transport is sufficient to achieve the desired functionality. As explained above in reality scalable codec is often not more efficient, especially in the path from the mixer to the receiver.

There are however, good reasons to combine simulcast with scalable encoding. By using simulcast to cover encoding variations where the scalable codec least efficient one can optimize the efficiency of the complete system. So a low number of simulcast working points, where each working point is in its turn a scalable codec configuration providing medium and/or fine grained scalability allowing a mixer to further tune the bit-rate to the available towards particular receivers using a combination of selecting simulcast versions and the number of extensions layers from that source.

A good example of this usage would be to send video encoded using SVC, where each simulcast version is a different resolution, and each SVC media stream uses temporal scalability and SNR scalability within that single media stream. If only resolution and temporal variations are needed, this can be implemented using H.264, as each simulcast version provides the different resolution, and each media stream within a simulcast encoding has temporal scalability using no-reference frames.

3.2. Simulcasting to Consuming End-Point

This usage is based on an RTP Transport Translator (Section 3.3: Topo-Trn-Translator) [RFC5117]. The transport translator functions as a relay and transmits all the streams received from one participant to all the other participants. In this case, one would do downlink simulcasting such that all receivers would receive all the versions. However, this clearly increases the bit-rate consumed on the paths to the client. The only benefit for the receiving client would be reduced decoding complexity when needing to only display a low resolution version. Otherwise a single stream application which only transmits the high resolution stream would allow the receiver to decode it and scale it down to the needed resolution.

The usage of transport translator and simulcast becomes efficient if one allows each receiving client to control the relay to indicate which version it wants to receive. However such a usage of RTP has some potential issues with RTCP. From the sending end-point it will look like the transmitted stream isn't received by a receiver that is known to receive other streams from the sender. Thus some consideration and mechanism are needed to support such a use case so that it doesn't break RTCP reception reporting.

This document will continue to consider this case but with less emphasis than on the RTP mixer case.

3.3. Same Encoding to Multiple Destinations

One interpretation of simulcast is when one encoding is sent to multiple receivers. This is well supported in RTP by simply copying all outgoing RTP and RTCP traffic to several transport destinations as long as the intention is to create a common RTP session. As long as all participants do the same, a full mesh is constructed and everyone in the multi party session has a similar view of the joint RTP session. This is analog to an Any Source Multicast (ASM) session but without the traffic optimization as multiple copies of the same content is likely to have to pass over the same link.

```
+---+      +---+
| A |<---->| B |
+---+      +---+
  ^          ^
   \        /
    \      /
     v    v
     +---+
     | C |
     +---+
```

Full Mesh / Multi-unicast

As this type of simulcast is analog to ASM usage and RTP has good support for ASM sessions, no further consideration for this case is done.

3.4. Different Encoding to Independent Destinations

Another alternative interpretation of simulcast is with multiple destinations, where each destination gets a specifically tailored version, but where the destinations are independent. A typical example for this would be a streaming server distributing the same live session to a number of receivers, adapting the quality and resolution of the multi-media session to each receiver's capability and available bit-rate. This case can be solved in RTP by having independent RTP sessions between the sender and the receivers. Thus this case is not considered further.

4. Multiple Streams Issues

This section attempts to go a bit more in depth around the different issues when using multiple media streams in an RTP session to make it clear that although in theory multi-stream applications should already be possible to use, there are good reasons to create extensions for signalling. In addition, the RTP specification could benefit from clarifications on how certain mechanisms should be working when an RTP session contains more than two SSRCs.

4.1. Legacy behaviors

It is a common assumption among many applications using RTP that they don’t have a need to support more than one incoming and one outgoing media stream per RTP session. For a number of applications this assumption has been correct. For VoIP and Streaming applications it has been easiest to ensure that a given end-point only receives and/or sends a single stream. However, they should support a source switching SSRC, e.g due to collision.

Some RTP extension mechanisms require the RTP stacks to handle additional SSRCs, like SSRC multiplexed RTP retransmission [RFC4588]. However, that still has only required handling a single media decoding chain.

However, there are applications that clearly can benefit from receiving and using multiple media streams simultaneously. A very basic case would be T.140 conversational text, which is both low bandwidth and where there is no simple method for mixing multiple sources of text that is supposed to be transmitted and displayed as you type. An RTP session that contains more than 2 SSRC actively sending media streams has the potential to confuse a legacy client in various ways:

1. The receiving client needs to handle receiving more than one stream simultaneously rather than replacing the already existing stream with the new one.

2. Be capable of decoding multiple streams simultaneously

3. Be capable of rendering multiple streams simultaneously

These applications may be very similar to existing one media stream applications at signalling level. To avoid connecting two different implementations, one that is built to support multiple streams and one that isn't, it is important that the capabilities are signalled. It is also the legacy that makes us use a basic assumption in the solution. Anyone that doesn't explicitly indicate capability to receive multiple media streams is assumed to only handle a single media, to avoid affecting legacy clients.

4.2. Receiver Limitations

An RTP end-point that intends to process the media in an RTP session needs to have sufficient resources to receive and process all the incoming streams. It is extremely likely that no receiver is capable to handle the theoretical upper limit of an RTP session when it comes to more than 4 billion media sources. Instead, one or more properties will limit the end-points' capabilities to handle simultaneous media streams. These properties are for example memory, processing, network bandwidth, memory bandwidth, or rendering estate to mention a few possible limitations.

We have also considered the issue of how many simultaneous non-active sources an end-point can handle. We cannot see that inactive media sending SSRCs result in significant resource consumption and there should thus be no need to limit them.

A potential issue that needs to be acknowledged is where a limited set of simultaneously active sources varies within a larger set of session members. As each media decoding chain may contain state, it is important that this type of usage ensures that a receiver can flush a decoding state for an inactive source and if that source becomes active again it does not assume that this previous state exists.

Thus, we see need for a signalling solution that allows a receiver to indicate its upper limit in terms of capability to handle simultaneous media streams. We see little need for an upper limitation of RTP session members. Applications will need to have some considerations around how they use codecs.

4.3. Transmission Declarations

In an RTP based system where an end-point may either be legacy or has an explicit upper limit in the number of simultaneous streams, one will encounter situations where the end-point will not receive all simultaneous active streams in the session. Instead the end-points or central nodes, like RTP mixers, will provide the end-point with a selected set of streams based on various metrics, such as most active, most interesting, or user selected. In addition, the central node may combine multiple media streams using mixing or composition into a new media stream to enable an end-point to get a sufficient source coverage in the session, despite existing limitations.

For such a system to be able to correctly determine the need for central processing, the capabilities needed for such a central processing node, and the potential need for an end-point to do sender side limitations, it is necessary for an end-point to declare how many simultaneous streams it may send. Thus, enabling negotiation of the number of streams an end-point sends.

4.4. RTP and RTCP Issues

This section details a few RTP and RTCP issues identified in implementation work for supporting multiple streams.

4.4.1. Multiple Sender Reports in Compound

One potential interoperability issue is inclusion of multiple Sender Report blocks in the same RTCP compound packet. The RTP specification isn't clear if such stacking is allowed or not. Thus there might be RTCP receivers that might not correctly handle such message. There is also an uncertainty how one should calculate the RTCP transmission intervals in such cases.

4.4.2. Cross reporting within an end-point

When an end-point has more than one SSRC and sends media using them, a question arises if the different SSRCs needs to report on each other despite being local. It can be argued that it is needed due to that it might not be fully visible for any external observer that they are actually sent from the same end-point. Thus by reporting on each other there are no holes in the connectivity matrix between all sending SSRCs and all known SSRCs.

4.4.3. Which SSRC is providing feedback

When one has multiple SSRCs on an end-point and needs to send RTCP feedback messages some considerations around which SSRC is used as the source and if that is consistently used or not, may be needed.

4.5. SDP Signalling Issues

An existing issue with SDP is that the bandwidth parameters aren't specified to take asymmetric conditions into account. This becomes especially evident when we start using multiple streams in an RTP session. Such a use case can easily result in that an end-point maybe receive 5 streams of Full High Definition (HD) video but only sends one Standard Definition (SD) video stream. Thus easily having a 10:1 asymmetry in bit-rate.

If one uses the current SDP bandwidth parameters then one likely needs to set the session bandwidth to the sum of the most consuming direction. This can result in that there is no way of negotiating an upper bound for the lower band-width direction media stream(s). In addition, an end-point may conclude that it can't support the bit-rate despite being capable of actually receiving the media streams being sent. Thus making clear what bandwidth limitations a single stream has compared to the whole RTP session is important.

In the cases there is QoS, either by end-point reservation or done by systems like IMS, the requested bandwidth based on the signalled value will not represent what is actually needed.

Asymmetry in itself also create an issue, as RTCP bandwidth may be derived from the session bandwidth. It is important that all end-points have a common view on what the RTCP bandwidth is. Otherwise if the bandwidth values are more than 5 times different, an end-point with the high bandwidth value may time out an end-point that has a low value as it's minimal reporting interval can become more than 5 times longer than for the other nodes.

5. Multi-Stream Extensions

5.1. Signaling Support for Multi-Stream

There is a need to signal between RTP sender and receiver how many simultaneous RTP streams can be handled. The number of RTP streams that can be sent from a client should not have to match the number of streams that can be received by the same client. A multi-stream capable RTP sender MUST be able to adapt the number of sent streams to the RTP receiver capability.

For this purpose and for use in SDP, two new media-level SDP attributes are defined, max-send-ssrc and max-recv-ssrc, which can be used independently to establish a limit to the number of simultaneously active SSRCs for the send and receive directions, respectively. Active SSRCs are the ones counted as senders according to RFC3550, i.e. they have sent RTP packets during the last two regular RTCP reporting intervals.

The syntax for the attributes are in ABNF [RFC5234]:

```
max-ssrc = "a=" ("max-send-ssrc:" / "max-recv-ssrc:") PT 1*WSP limit
PT = "*" / 1*3DIGIT
limit = 1*8DIGIT
```

; WSP and DIGIT defined in [RFC5234]

A payload-agnostic upper limit to the total number of simultaneous SSRC that can be sent or received in this RTP session is signaled with a * payload type. A value of 0 MAY be used as maximum number of SSRC, but it is then RECOMMENDED that this is also reflected using the sendonly or recvonly attribute. There MUST be at most one payload-agnostic limit specified in each direction.

A payload-specific upper limit to the total number of simultaneous SSRC in the RTP session with that specific payload type is signaled with a defined payload type (static, or dynamic through rtpmap). Multiple lines with max-send-ssrc or max-recv-ssrc attributes specifying a single payload type MAY be used, each line providing a limitation for that specific payload type. Payload types that are not defined in the media block MUST be ignored.

If a payload-agnostic limit is present in combination with one or more payload-specific ones, the total number of payload-specific SSRCs are additionally limited by the payload-agnostic number. When there are multiple lines with payload-specific limits, the sender or receiver MUST be able to handle any combination of the SSRCs with different payload types that fulfill all of the payload specific limitations, with a total number of SSRCs up to the payload-agnostic limit.

When max-send-ssrc or max-recv-ssrc are not included in the SDP, it MUST be interpreted as equivalent to a limit of one, unless sendonly or recvonly attributes are specified, in which case the limit is implicitly zero for the corresponding unused direction.

5.1.1. Declarative Use

When used as a declarative media description, the specified limit in max-send-ssrc indicates the maximum number of simultaneous streams of the specified payload types that the configured end-point may send at any single point in time. Similarly, max-recv-ssrc indicates the maximum number of simultaneous streams of the specified payload types that may be sent to the configured end-point. Payload-agnostic limits MAY be used with or without additional payload-specific limits.

5.1.2. Use in Offer/Answer

When used in an offer, the specified limits indicates the agent's intent of sending and/or capability of receiving that number of simultaneous SSRC. The answerer MUST reverse the directionality of recognized attributes such that max-send-ssrc becomes max-recv-ssrc and vice versa. The answerer SHOULD decrease the offered limit in the answer to suit the answering client's capability. A sender MUST NOT send more simultaneous streams of the specified payload type than the receiver has indicated ability to receive, taking into account also any payload-agnostic limit.

In case an answer fails to include any of the limitation attributes, the agent MUST be interpreted as capable of supporting only a single stream in the direction for which attributes are missing. If the offer lacks attributes it MUST be assumed that the offerer only supports a single stream in each direction. In case the offer lack both max-send-ssrc and max-recv-ssrc, they MUST NOT be included in the answer.

5.1.3. Examples

The SDP examples below are not complete. Only relevant parts have been included.

```
m=video 49200 RTP/AVP 99
a=rtpmap:99 H264/90000
a=max-send-ssrc:* 2
a=max-recv-ssrc:* 4
```

An offer with a stated intention of sending 2 simultaneous SSRCs and a capability to receive 4 simultaneous SSRCs.

```
m=video 50324 RTP/AVP 96 97
a=rtpmap:96 H264/90000
a=rtpmap:97 H263-2000/90000
a=max-recv-ssrc:96 2
a=max-recv-ssrc:97 5
a=max-recv-ssrc:* 5
```

An offer to receive at most 5 SSRC, at most 2 of which using payload type 96 and the rest using payload type 97. By not including "max-send-ssrc" the value is implicitly set to 1.

```
m=video 50324 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H263-2000/90000
a=max-recv-ssrc:96 2
a=max-recv-ssrc:97 3
a=max-recv-ssrc:98 5
a=max-recv-ssrc:* 5
```

An offer to receive at most 5 SSRC, at most 2 of which using payload type 96, and at most 3 of which using payload type 97, and at most 5 using payload type 98. Permissible payload type combinations include those with no streams at all for one or more of the payload types, as well as a total number of SSRC less than 5, e.g. two SSRC with PT=96 and three SSRC with PT=97, or one SSRC with PT=96, one with PT=97 and two with PT=98.

5.2. Asymmetric SDP Bandwidth Modifiers

To resolve the issues around bandwidth, we propose new SDP bandwidth modifiers that supports directionality, possibility for payload specific values and clear semantics. A common problem for all the current SDP bandwidth modifiers is that they use a single bandwidth value without a clear specification. Uncertainty in how the bandwidth value is derived creates uncertainty on how bursty a media source can be.

Thus, we do consider what the design criteria are prior to providing a proposal for new SDP bandwidth attribute.

5.2.1. Design Criterias

The current b= SDP bandwidth syntax is very limited and only allows the following format:

```
bandwidth-fields  = *(%x62 "=" bwtype ":" bandwidth CRLF)
bwtype            = token
bandwidth         = 1*DIGIT
```

Thus we will need to specify a new SDP bandwidth attribute as that allows syntax of more complexity.

The functionalities we see from the new bandwidth attribute are the following:

Directionality: We need to be able to have different sets of attribute values depending on direction.

Bandwidth semantics: A semantics identifier so that new semantics can be defined in the future for other needed semantics. This part of the b= has been a very successful design feature. We do perceive a need for both single stream limitations and limitations for the aggregate of all streams in one direction.

Payload specific: The possibility to specify different bandwidth values for different RTP Payload types. This as some codecs have different characteristics and one may want to limit a specific codec and payload configuration to a particular bandwidth. Especially combined with codec negotiation there is a need to express intentions and limitations on usage for that particular codec. In addition, payload agnostic information is also needed.

Bandwidth specification method: To have a clear specification of what any bit-rate values mean we propose that Token bucket parameters should be used, i.e. bucket depth and bucket fill rate, where appropriate for the semantics. If single values are to be specified, a clear definition on how to derive that value must be specified, including averaging intervals etc.

We will use these design criteria next in an actual proposal.

5.2.2. Attribute Specification

We define a new SDP attribute ("a=") as the bandwidth modifier line syntax can't support the requirements and nor can it be changed in an interoperable way. Thus we define the "a=bw" attribute. This attribute is structured as follows. After the attribute name there is a directionality parameter, followed by a scope parameter and then a bandwidth semantics tag. The semantics tag defines what value(s) that follow and their interpretation.

The attribute is designed so that multiple instances of the line will be necessary to express the various bandwidth related configurations that are desired.

Scopes and semantics can be extended in the future at any point. To ensure that an end-point using SDP either in Offer/Answer or declarative truly understands these extensions, a required-prefix indicator ("!") can be added prior to any scope or semantics parameter.

5.2.2.1. Attribute Definition

The ABNF [RFC5234] for this attribute is the following:

```
bw-attrib     = "a=bw:" direction SP [req] scope SP
                [req] semantics ":" values
direction     = "send" / "recv" / "sendrecv"
scope         = payloadType / scope-ext
payloadType   = "PT=" ("*" / PT-value-list)
PT-value-list = PT-value *(";" PT-Value)
PT-value      = 1*3DIGIT
req           = "!"
semantics     = "SMT" / "AMT" / semantics-ext
values        = token-bucket / value-ext
token-bucket  = "tb=" br-value ":" bs-value
br-value      = 1*15DIGIT ; Bucket Rate
bs-value      = 1*15DIGIT ; Bucket Size

semantics-ext = token ; As defined in RFC 4566
scope-ext     = 1*VCHAR ; As defined in RFC 4566
value-ext     = 0*(WSP / VCHAR)
```

The a=bw attribute defines three possible directionalities:

send: In the send direction for SDP Offer/Answer agent or in case of declarative use in relation to the device that is being configured by the SDP.

recv: In the receiving direction for the SDP Offer/Answer agent providing the SDP or in case of declarative use in relation to the device that is being configured by the SDP.

sendrecv: The provided bandwidth values applies equally in send and recv direction, i.e. the values configures the directions symmetrically.

The Scope indicates what is being configured by the bandwidth semantics of this attribute line. This parameter is extensible and we begin with defining two different scopes based on payload type:

Payload Type: The bandwidth configuration applies to one or more specific payload type values.

PT=*: Applies independently of which payload type is being used.

This specification defines two semantics which are related. The Stream Maximum Token bucket based value (SMT) and the Aggregate Maximum Token bucket based value (AMT). Both semantics represent the bandwidth consumption of the stream or the aggregate as a token bucket. The token bucket values are the token bucket rate and the token bucket size, represented as two integer numbers. It is an open question exactly what this token bucket is measuring, if it is RTP payload only, like TIAS, or if it includes all headers down to the IP level as most of the other bandwidth modifiers do.

The definition of the semantics in more detail are:

SMT: The maximum intended or allowed bandwidth usage for each individual source (SSRC) in an RTP session as specified by a token bucket. The token bucket values are the token rate in bits per second and the bucket size in bytes. This semantics may be used both symmetrically or in a particular direction. It can be used either to express the maximum for a particular payload type or for any payload type (PT=*).

AMT: The maximum intended or allowed bandwidth usage for sum of all sources (SSRC) in an RTP session according to the specified directionality as specified by a token bucket. The token bucket values are the token rate in bits per second and the bucket size in bytes. Thus if using the sendrecv directionality parameter, both send and receive streams SHALL be included in the generated aggregate. If only a send or recv, then only the streams present in that direction are included in the aggregate. It can be used either to express the maximum for a particular payload type or for any payload type (PT=*).

5.2.2.2. Offer/Answer Usage

The offer/answer negotiation is done for each bw attribute line individually with the scope and semantics immutable. If an answerer would like to add additional bw configurations using other directionality, scope, and semantics combination, it may add them.

An agent responding to an offer will need to consider the directionality and reverse them when responding to media streams using unicast. If the transport is multicast the directionality is not affected.

For media stream offers over unicast with directionality send, the answerer will reverse the directionality and indicate its reception bandwidth capability, which may be lower or higher than what the sender has indicated as its intended maximum.

For media stream offers over unicast with directionality receive, these do indicate an upper limit, the answerer will reverse the directionality and may only reduce the bandwidth when producing the answer indicating the answerer intended maximum.

[Need to define how the required "!" prefix is used in Offer/Answer]

5.2.2.3. Declarative Usage

In declarative usage the SDP attribute is interpreted from the perspective of the end-point being configured by the particular SDP. An interpreter MAY ignore a=bw attribute lines that contains unknown scope or semantics that does not start with the required ("!") prefix. If a "required" prefix is present at an unknown scope or semantics, the interpreter SHALL NOT use this SDP to configure the end-point.

5.2.2.4. Example

Declarative example with stream asymmetry.

```
m=video 50324 RTP/AVP 96 97 98
a=rtpmap:96 H264/90000
a=rtpmap:97 H263-2000/90000
a=rtpmap:98 MP4V-ES/90000
a=max-recv-ssrc:96 2
a=max-recv-ssrc:* 5
a=bw:send pt=* SMT:tb=1200000:16384
a=bw:recv pt=96 SMT:tb=1500000:16384
a=bw:recv pt=97:98 SMT:tb=2500000:16384
a=bw:recv pt=* AMT:tb=8000000:65535
```

In the above example the outgoing single stream is limited to bucket rate of 1.2 Mbps and bucket size of 16384 bytes. The up to 5 incoming streams can in total use maximum 8 Mbps bucket rate and with a bucket size of 65535 bytes. However, the individual streams maximum rate is depending on payload type. Payload type 96 (H.264) is limited to 1.5 Mbps with a bucket size of 16384 bytes, while the Payload types 97 (H.263) and 98 (MPEG-4) may use up top 2.5 Mbps with a bucket size of 16384 bytes.

5.3. Binding SSRCs Across RTP Sessions

When an end-point transmits multiple sources in the same RTP session there may be tight relations between two different media types and their SSRCs, for example a microphone and a camera that is co-located are tightly related. CNAME is not sufficient to express this relation although it is commonly inferred from end-points that has only one media stream per media type. CNAME primary use in multi-source usages is to indicate which end-point and what synchronization context a particular media stream relates to.

To enable a RTP session participant to determine that close binding across multiple sessions, despite the end-point sending multiple SSRCs a new method for identifying such sources are needed. We are not relying on using the same SSRC in all sessions for a particular media source as it is not robust against SSRC collision and forces potentially cascading SSRC changes between sessions.

5.3.1. SDES Item SRCNAME

Source Descriptions are a method that should work with all RTP topologies (assuming that any intermediary node is supporting this item) and existing RTP extensions. Thus we propose one defines a new SDES item called the SRCNAME which identifies with an unique identifier a single multi-media source, like a camera and a co-located microphone, or a truly individual media source such as a camera. That way any one receiving the SDES information from a set of interlinked RTP sessions can determine which are the same source.

We proposes that the SRCNAME would commonly be per communication session unique random identifiers generated according to "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)" [RFC6222] with the addition that a local counter enumerating the sources on the host also are concatenated to the key in step 4 prior to calculating the hash.

This SRCNAME’s relation to CNAME is the following. CNAME represents an end-point and a synchronization context. If the different sources identified by SRCNAMEs should be played out synchronized when receiving them in a multi-stream context, then the sources need to be in the same synchronization context. Thus in all cases, all SSRCs with the same SRCNAME will have the same CNAME. A given CNAME may contain multiple sets of sources using different SRCNAMEs.

5.3.2. SRCNAME in SDP

Source-Specific Media Attributes in the Session Description Protocol (SDP) [RFC5576] defines a way of declaring attributes for SSRC in each session in SDP. With a new SDES item, one can use this framework to define how also the SRCNAME can be provided for each SSRC in each RTP session, thus enabling an end-point to declare and learn the simulcast bindings ahead of receiving RTP/RTCP packets.

6. Simulcast Alternatives

Simulcast is the act of sending multiple alternative encodings of the same underlying media source. When transmitting multiple independent flows that originate from the same source, it could potentially be done in several different ways in RTP. The below sub-sections describe potential ways of achieving flow de-multiplexing and identification of which streams are alternative encodings of the same source.

In the below descriptions we also include how this interacts with multiple sources (SSRCs) in the same RTP session for other reasons than simulcast. So multiple SSRCs may occur for various reasons such as multiple participants in multipoint topologies such as multicast, transport relays or full mesh transport simulcasting, multiple source devices, such as multiple cameras or microphones at one end-point, or RTP mechanisms in use, such as RTP Retransmission [RFC4588].

6.1. Payload Type Multiplexing

Payload multiplexing uses only the RTP payload type to identify the different alternatives. Thus all alternative streams would be sent in the same RTP session using only a single SSRC per actual media source. So when having multiple SSRCs, each SSRC would be unique media sources or RTP mechanism-related SSRC. Each RTP payload type would then need to both indicate the particular encoding and its configuration in addition to being a stream identifier. When considering a mechanism like RTP retransmission using SSRC multiplexing, an SSRC may either be a media source with multiple encodings as provided by the payload type, or a retransmission packet as identified also by the payload type.

As some encoders, like video, produce large payloads one can not expect that multiple payload encodings can fit in the same RTP packet payload. Instead a payload type multiplexed simulcast will need to send multiple different packets with one version in each packet or sequence of packets.

6.2. SSRC Multiplexing

The SSRC multiplexing idea is based on using a unique SSRC for each alternative encoding of one actual media source within the same RTP session. The identification of how flows are considered to be alternative needs an additional mechanism, for example using SSRC grouping [RFC5576] and a new SDES item such as SRCNAME proposed in Section 5.3.1 with a semantics that indicate them as alternatives of a particular media source. When one have multiple actual media sources in a session, each media source will use a number of SSRCs to represent the different alternatives it produces. For example, if all actual media sources are similar and produce the same number of simulcast versions, one will have n*m SSRCs in use in the RTP session, where n is the number of actual media sources and m the number of simulcast versions they can produce. Each SSRC can use any of the configured payload types for this RTP session. All session level attributes and parameters which are not source specific will apply and must function with all the alternative encodings intended to be used.

6.3. Session Multiplexing

Session multiplexing means that each different version of an actual media source is transmitted in a different RTP session, using whatever session identifier to de-multiplex the different versions. This solution needs explicit session grouping [RFC5888] with a semantics that indicate them as alternatives. When there are multiple actual media sources in use, the SSRC representing a particular source will be present in the sessions for which it produces a simulcast version. It is also important to identify the SSRCs in the different sessions that are alternative encodings to each other, this can be accomplished using the same SSRC and/or a new SDES item identifying the media source across the session as the proposed SRCNAME SDES item (Section 5.3.1). Each RTP session will have its own set of configured RTP payload types where each SSRC in that session can use any of the configured ones. In addition all other attributes for sessions or sources can be used as normal to indicate the configuration of that particular alternative.

7. Simulcast Evaluation

This chapter evaluates the different multiplexing strategies in regard to several aspects.

7.1. Effects on RTP/RTCP

This section will be oriented around the different multiplexing mechanisms.

7.1.1. Payload Type Multiplexing

The simulcast solution needs to ensure that the negative impact on RTP/RTCP is minimal and that all the features of RTP/RTCP and its extensions can be used.

Payload type multiplexing for purposes like simulcast has well known negative effects on RTP. The basic issue is that all the different versions are being sent on the same SSRC, thus using the same timestamp and sequence number space. This has many effects:

1. Putting restraint between media encoding versions. For example, media encodings that uses different RTP timestamp rates cannot be combined as the timestamp values needs to be the same across all versions of the same media frame. Thus they are forced to use the same rate. When this is not possible, Payload Type

Multiplexing cannot be used.

2. Most RTP payload formats that may fragment a media object over multiple packets, like parts of a video frame, needs to determine the order of the fragments to correctly decode them. Thus it is important that one ensure that all fragments related to a frame or a similar media object are transmitted in sequence and without interruptions within the object. This can relatively simple be solved by ensuring that each version is sent in sequence.

3. Some media formats require uninterrupted sequence number space between media parts. These are media formats where any missing RTP sequence number will result in decoding failure or invoking of a repair mechanism within a single media context. The text/T140 payload format [RFC4103] is an example of such a format. These formats will be impossible to simulcast using payload multiplexing.

4. Sending multiple versions in the same sequence number space makes it more difficult to determine which version a packet loss may relate to. If one uses RTP Retransmission [RFC4588] one can ask for the missing packet. However, if the missing packet(s) do not belong to the version one is interested in, the retransmission request was in fact unnecessary.

5. The current RTCP feedback mechanisms are built around providing feedback on media streams based on stream ID (SSRC), packet (sequence numbers) and time interval (RTP Timestamps). There is almost never a field for indicating which payload type one is reporting on. Thus giving version specific feedback is difficult.

6. The current RTCP media control messages [RFC5104] specification is oriented around controlling particular media flows, i.e. requests are done addressing a particular SSRC. Thus such mechanisms needs to be redefined to support payload type multiplexing.

7. The number of payload types are inherently limited. Accordingly, using payload type multiplexing limits the number of simulcast streams and does not scale.

7.1.2. SSRC Multiplexing

As each version of the source has its own SSRC and thus explicitly unique flows, the negative effects above (Section 7.1.1) are not present for SSRC multiplexed simulcast.

The SSRC multiplexing of simulcast version requires a receiver to know that one is expected to only decode one of the versions and need not decode all of them simultaneously. This is currently a missing functionality as SDES CNAME cannot be used. The same CNAME has to be used for all flows connected to the same end-point and location. A clear example of this could be video conference where an end-point has 3 video cameras plus an audio mix being captured in the same room. As the media has a common timeline, it is important to be able to indicate that through the CNAME. Thus one cannot use CNAME to indicate that multiple SSRCs with the same CNAME are different versions of the same source. New semantics are required.

When one has all the versions in the same RTP session going to an RTP mixer and the mixer chooses to switch from forwarding one of the versions to forwarding another version, this creates an uncertainty in which SSRC one should use in the CSRC field (if used). As one is still delivering the same original source, such switch appears questionable to a receiver not having enabled simulcast in the direction to itself. Depending on what solution one chooses, one gets different effects here. If the CSRC is changed, then any message ensuring binding will need to be forwarded by the mixer, creating legacy issues. It has not been determined if there are downsides to not showing such a switch.

The impact of SSRC collisions on the SSRC multiplexing will be highly depending on what method is used to bind the SSRCs that provide different versions. Upon a collision and a forced change of the SSRC, a media sender will need to re-establish the binding to the other versions. By doing that, it will also likely be explicit when it comes to what the change was.

7.1.3. Session Multiplexing

Also session multiplexing does not have any of the negative effects that payload type multiplexing has (Section 7.1.1). As each flow is uniquely identified by RTP Session and SSRC, one can control and report on each flow explicitly. The great advantage of this method is that each RTP session appears just like if simulcast is not used thus minimal issues in RTP and RTCP including any extensions.

One potential downside of session multiplexing is that it becomes impossible without defining new RTCP message types to do truly synchronized media requests where one request goes to version A of source and another to version B of the same source. Due to the RTP session separation, one will be forced to send different RTCP packets to the different RTP session contexts, thus losing the ability to send two different RTCP packets in the same compound packet and RTP session context. This can be a minor inconvenience.

Using the same SSRC in all the RTP sessions allows for quick binding between the different versions. It also enables an RTP mixer that forwards one version to seamlessly decide to forward another version in a RTP session to a session participant that is not using simulcast in the direction from the mixer to the participant.

An SSRC collision forces a sender to change its SSRC in all sessions. Thus the collision-induced SSRC change may have bigger impact, as it affects all versions rather than a single version. But on the positive side, the binding between the versions will be immediate, rather than requiring additional signaling.

7.2. Signaling Impact

The method of multiplexing has significant impact on signaling functionality and how to perform it, especially if SDP [RFC4566] and SDP Offer/Answer [RFC3264] is used.

7.2.1. Negotiating the use of Simulcast

There will be a need for negotiating the usage of simulcast in general. For payload type multiplexing, one will need to indicate that different RTP payload types are intended as different simulcast versions. One likely has standalone SDP attributes that indicate the relation between the payload types, as one needs unique payload type numbers for the different versions. Thus, this increases the number of payload types needed within an RTP session. In worst case this may become a restriction as only 128 payload types are possible. This limitation is exacerbated if one uses solutions like RTP and RTCP multiplexing [RFC5761] where a number of payload types are blocked due to the overlap between RTP and RTCP.

SSRC multiplexing will likely use a standalone attribute to indicate the usage of simulcast. In addition, it may be possible to use a mechanism in SDP that binds the different SSRCs together. The first part is non-controversial. However the second one has significant impact on the signaling load in sessions with dynamic session participation. As each new participant joins a multiparty session, the existing participants that need to know the binding will need to receive an updated list of bindings. If that is done in SIP and SDP offer answer, a SIP re-Invite is required for each such transaction, invoking all the SIP nodes related to invites, and in systems like IMS also a number of policy nodes. If a receiver is required, which is likely, to receive the SSRC bindings prior to being able to decode any new source, then the signaling channel may introduce additional delay before a receiver can decode the media.

Session multiplexing results in one media description per version.

It will be necessary to indicate which RTP sessions are in fact simulcast versions. For example, using a Media grouping semantics specific for this. Each of these sessions will be focused on the particular version they intend to transport.

Legacy fallback, the impact on an end-point that isn't simulcast enabled, also needs to be considered. For a payload type multiplex solution, a legacy end-point that doesn't understand the indication that different RTP payload types are for different purpose may be slightly confused by the large amount of possibly overlapping or identical RTP payload types. In addition, as payload multiplexing isn't backwards compatible within a single media stream, the signalling needs to ensure that such a legacy client doesn't join a session using simulcast.

For an SSRC multiplexed session, a legacy end-point will ignore the SSRC binding signaling. From its perspective, this session will look like an ordinary session and it will setup to handle all the versions simultaneously. Thus, a legacy client is capable of decoding and rendering a simulcast enabled RTP session, but it will consume more resources and result in a duplication of the same source.

For session multiplexing, a legacy end-point will not understand the grouping semantic. It might either understand the grouping framework and thus determine that they are grouped for some purpose, or not understand grouping at all and then the offer simply looks like several different media sessions. This enables a simple fallback solution to exclude a legacy client from all simulcast versions except one, whichever is most suitable for the application.

7.2.2. Bandwidth negotation

The payload type multiplexed session cannot negotiate bandwidth for the individual versions without extensions. The regular SDP bandwidth attributes can only negotiate the overall bandwidth that all versions will consume. This makes it difficult to determine that one should drop one or more versions due to lack of bandwidth between the peers.

SSRC multiplexing suffers the same issues as payload type multiplexing, unless additional signaling (SSRC level attributes) is added.

Session multiplexing can negotiate bandwidth for each individual version and determine to exclude a particular version, and have the full knowledge on what it excludes to avoid consuming an excessive amount of bandwidth.

7.2.3. Negotation of media parameters

The negotiation and setting of the media codec, the codec parameters and RTP payload parameters for the payload type multiplexing is possible for each individual version as each has a unique payload type. The same is true for the session multiplexing where each version negotiates the parameters in the context of it's RTP session. The SSRC multiplexed version would need additional signaling to enable a binding between the payload types and which versions they are used for. Otherwise, the RTP payload types are negotiated without any context of which version intends to use which payload type.

However, the above assumes that there are no issues with defining different payload types for different alternative encodings. If that is not possible or it is intended to use the same payload type for multiple encodings, then additional signalling becomes necessary which isn't possible for payload multiplexing. For SSRC multiplexing, this signalling needs to redefine already existing session attributes, like imageattr [RFC6236] to have a per-SSRC scope. Session multiplexing can use existing attributes as they automatically get per-encoding scope thanks to the session multiplexing.

7.2.4. Negotation of RTP/RTCP Extensions

When one negotiates or configures the existing RTP and RTCP extensions, that can be done on either session level or in direct relation to one or several RTP payload types. They are not negotiated in the context of an SSRC. Thus payload type multiplexing will need to negotiate any session level extensions for all the versions without version specific consideration, unless extensions are deployed. It can also negotiate payload specific versions at a version individual level. SSRC multiplexing cannot negotiate any extension related to a certain version without extensions. Session multiplexing will have the full freedom of negotiating extensions for each version individually without any additional extensions.

7.3. Network Aspects

The multiplexing choice has impact on network level mechanisms.

7.3.1. Quality of Service

When it comes to Quality of Service mechanisms, they are either flow based or marking based. RSVP [RFC2205] is an example of a flow based mechanism, while Diff-Serv [RFC2474] is an example of a Marking based one. If one uses a marking based scheme, the method of multiplexing will not affect the possibility to use QoS. However, if one uses a flow based one, there is a clear difference between the methods. Both Payload Type and SSRC multiplexing will result in all versions being part of the same 5-tuple (protocol, source address, destination address, source port, destination port) which is the most common selector for flow based QoS. Thus, separation of the level of QoS between versions is not possible. That is however possible if one uses session based multiplexing, where each different version will be in a different RTP context and thus commonly being sent over different 5-tuples.

7.3.2. NAT Traversal

Both the payload and SSRC multiplexing will have only one RTP session, not introducing any additional NAT traversal complexities compared to not using simulcast and only have a single version. The session multiplexing is using one RTP session per simulcast version. Thus additional lower layer transport flows will be required unless an explicit de-multiplexing layer is added between RTP and the transport protocol.

Below we analyzed and comment on the impact of requiring more underlying transport flows in the presence of NATs and Firewalls:

End-Point Port Consumption: A given IP address only has 65536 available local ports per transport protocol for any consumer of ports that exist on the machine. This is normally never an issue for a end-user machine. It can become an issue for servers that have large number of simultaneous flows. However, if the application uses ICE, which authenticated STUN requests, a server can serve multiple end-point from the same local port, and use the whole 5-tuple (source and destination address, source and destination port, protocol) as identifier of flows after having securely bound them to end-points using the STUN request. Thus in theory the minimal number of media server ports needed are the maximum number of simultaneous RTP sessions a single end-point may use, when in practice implementation will probably benefit from using more.

NAT State: If an end-point is behind a NAT each flow it generates to an external address will result in a state on that NAT. That state is a limited resource, either from memory or processing stand-point in home or SOHO NATs, or for large scale NATs serving many internal end-points, the available external ports run-out. We see this primarily as a problem for larger centralized NATs where end-point independent mapping do require each flow mapping to use one port for the external IP address. Thus affecting the maximum aggregation of internal users per external IP address.

However, we would like to point out that a real-time video conference session with audio and video are likely using less than 10 UDP flows, it is not like certain web applications that can result that 100+ TCP flows are opened to various servers from a single browser instance.

NAT Traversal taking additional time: When doing the NAT/FW traversal it takes additional time. And it takes time in a phase of communication between accepting to communicate and the media path being established which is fairly critical. The best case scenario for how much extra time it can take following the specified ICE procedures are: 1.5*RTT + Ta*(Additional_Flows-1), where Ta is the pacing timer, which ICE specifies to be no smaller than 20 ms. That assumes a message in one direction, and then an immediate triggered check back. This as ICE first finds one candidate pair that works prior to establish multiple flows. Thus, there are no extra time until one has found a working candidate pair. Based on that working pair the extra time it takes, is what it takes to in parallel establish the additional flows which in most case are 2-3 additional flows.

NAT Traversal Failure Rate: Due to that one need more than a single flow to be established through the NAT there is some risk that one succeed in establishing the first flow but fails with one or more of the additional flows. The risk that this happens are hard to quantify. However, that risk should be fairly low as one has just prior successfully established one flow from the same interfaces. Thus only rare events as NAT resource overload, or selecting particular port numbers that are filtered etc, should be reasons for failure.

As most simulcast solutions will anyway not use a very large number of simulcast versions due to the cost in encoding resources etc. one can discuss if the extra transport flows are a significant cost. We perceive the cost as low, if others are concluding that the cost is higher, a more generalized mechanism for multiplexing RTP sessions onto the same underlying transport flow should be considered.

7.4. Summary

It is quite clear from the analysis that payload type multiplexing is not at all a realistic option for using simulcast. It has many issues, especially on RTP/RTCP level. Thus, we will not consider it a viable solution in further discussions below.

Both SSRC and session multiplexing are viable to use. However, session multiplexing provides increased flexibility in usage, better support for network QoS, signalling flexibility, and support compared to SSRC multiplexing, without defining additional extensions. Session multiplexing does however require additional NAT/FW pinholes to be opened or some other solution to allow multiple RTP sessions to share the same transport flow, but that is anyway something that already happens in today's applications.

The authors consider the impact on the signalling one of the most significant issues when it comes to SSRC multiplexing. For many use cases, selecting SSRC multiplexing will require us to define numerous signalling mechanisms to support binding such properties to specific SSRCs or encoding groups. This signalling already exists today for non simulcast RTP sessions or for simulcast in a session multiplexing context.

Session multiplexing is in the authors view clearly the best choice and is therefore recommended to be pursued as the single solution for simulcast.

8. Simulcast Extensions

This section discusses various extensions that either are required or could provide system performance gains if they where specified.

8.1. Signalling Support for Simulcast

To enable the usage of simulcast using session multiplexing some minimal signalling support is required. That support is discussed in this section. First of all, there is need for a mechanism to identify the RTP sessions carrying simulcast alternatives to each other. Secondly, a receiver needs to be able to identify the SSRC in the different sessions that are of the same media source but in different encodings.

Beyond the necessary signalling support for simulcast we look at some very useful optimizations in regards to the transmission of media streams and to help RTP mixers to select which stream alternatives to deliver to a specific client, or request a client to encode in a particular way.

8.1.1. Grouping Simulcast RTP Sessions

The proposal is to define a new grouping semantics for the session groupings framework [RFC5888]. There is a need to separate the semantics of intent to send simulcast streams from the capability to recognize and receive them. For that reason two new simulcast grouping tags are defined, "SimulCast Receive" (SCR) and "SimulCast Send" (SCS). They both act as an indicator that session level simulcast is occurring and which sets of RTP sessions that carries simulcast alternatives to each other.

The grouping semantics SCR and SCS SHOULD be combined with the SDP attributes "a=max-send-ssrc" and "a=max-recv-ssrc" Section 5.1 to indicate the number of simultaneous streams of each encoding that may be sent or capable of receiving.

8.1.1.1. Declarative Use

When used as a declarative media description, SCR indicates the configured end-points required capability to recognize and receive a specified set of RTP streams as simulcast streams. In the same fashion, SCS request the end-point to send a specified set of RTP streams as simulcast streams. SCR and SCS MAY be used independently and at the same time and they need not specify the same or even the same number of RTP sessions in the group.

8.1.1.2. Offer/Answer Use

When used in an offer, SCS indicates the SDP providing agent’s intent of sending simulcast, and SCR indicates the agent’s capability of receiving simulcast streams. SCS and SCR MAY be used independently and at the same time and they need not specify the same or even the same number of RTP sessions in the group. The answerer MUST change SCS to SCR and SCR to SCS in the answer, given that it has and wants to use the corresponding (reverse) capability. An answerer not supporting the SCS or SCR direction, or not supporting SCS or SCR grouping semantics at all, will remove that grouping attribute altogether, according to [RFC5888]. An offerer that receives an answer indicating lack of simulcast support in one or both directions, where SCR and/or SCS grouping are removed, MUST NOT use simulcast in the non-supported direction(s).

8.1.2. Binding SSRCs Across RTP Sessions

When one performs simulcast, a transmitting end-point will for each actual media source have one SSRC in each session for which it currently provides an encoding alternative. As a receiver or a mixer will receive one or more of these, it is important that any RTP session participant beyond the sender can explicitly identify which SSRCs in the set of RTP sessions providing a simulcast service for a particular media type that originate from the same media source and thus belong together in the simulcast.

To accomplish this we extend the usage of SRCNAME as defined in Section 5.3.1. Within a particular media type the different RTP session carrying the different encodings will have the same SRCNAME identifier. That way even if multiple encodings or representations are produced, any one receiving the SDES information from a set of interlinked RTP sessions can determine which are the same source.

8.2. Mixer Requests of Client streams

To increase the efficiency of simulcast systems, it is highly desirable that an RTP middlebox can signal to the client encoding and transmitting the streams if a particular stream is currently needed or not. This needs to be a quick and media plane oriented solution as it changes based on for example the user’s speech activity or the user’s selection in the user interface. Although several SIP and SDP-based methods would be possible, the required responsiveness suggests use of TMMBR from [RFC5104] with a bandwidth value of 0 to temporarily pause a certain SSRC and re-establishing transmission through TMMBR with a non-zero value.

8.3. Client to Mixer and Mixer to Client limiations

When a client has known limitations, for example based on local display layout between sources or if there is a better combination of streams from the available set of different encodings, then it is desirable to make these limitations known to the mixer delivering the streams. These limitations are also clearly dynamic, as sources may come or leave the session, making it prefer a different layout with another set of limitations in the delivered streams.

The Codec Control Messages in [RFC5104] defines some controls. However, with the addition of simulcast and scalable video there are more parameters that would be desired to control in a way similar to the Temporary Maximum Media Stream Bit Rate (TMMBR) messages, beyond just bit-rate. Factors such as largest image dimension and frame rate will also be needed, for example. In the context of simulcast, one also needs to consider if a limitation is not specific to an SSRC, but rather which encoding and scalability variation is most suitable from a particular media source (SRCNAME).

Thus we propose that new RTCP messages are defined to temporarily limit media source with respect to a combination of media stream properties such as for example bit-rate, frame-rate, image resolution, and audio channels. Such a message should be flexible enough to allow for additional limitation attributes.

8.4. Multiplexing Multiple RTP Sessions on Single Flow

It should be considered for RTP in non-legacy cases if multiple RTP sessions could be multiplexed in a standardized way on top of a single transport layer flow. That way the cost of opening additional transport flows and the needed NAT/FW traversal would be avoided. We acknowledge that this has impact on use cases using a flow based QoS mechanism that needs differentiated service levels between sessions. Such a mechanism should thus be optional to use, but as there is likely a general interest in such a mechanism, work on this should be started.

8.5. Examples

This section contains some SDP examples combining the proposals in this document to accomplish actual usages. We have skipped both NAT traversal tools as well as using the AVPF RTP profile [RFC4585] and Codec Control Messages [RFC5104] to save space in the SDPs, they are bulky enough. However, all these tools are likely to be part of a real SDP.

8.5.1. Multi-stream Signaling

This section contains examples of signalling for an application using multiple streams within an RTP session in two different contexts. In both these cases, the end-point that is involved in the signalling receives multiple streams, while only in the second case will the end-point transmit multiple streams.

8.5.1.1. Local Rendering in Video Conference Client

This example assumes a transport translator that enables the end-point to receive multiple streams from the other participants without using multiple destinations on transport level.

```
+---+      +------------+      +---+
| A |<---->|            |<---->| B |
+---+      |            |      +---+
           | Translator |
+---+      |            |      +---+
| C |<---->|            |<---->| D |
+---+      +------------+      +---+
```

Four-party Translator-based Conference

Example of Media plane for RTP transport translator based multi-party conference with 4 participants.

Client A (Alice) in above figure is a desktop video conference client with a single camera and microphone. It uses a central transport translator to relay its media streams to the other participants, and in the same way it receives media streams from all other participants from the relay. This enables the client to locally render and present other participants in a layout selected by the local client.

The network path between client A and the translator has certain known limitations, leading to a client needing to express its upper bounds in simultaneous streams that can be supported. That allows the conference server to know when it needs to tell the media plane relay to change its behavior from relaying to switching the media streams.

Alice invites herself into the conference by sending the following SDP offer:

```
v=0
o=alice 2890844526 2890842807 IN IP4 192.0.2.156
s=Multi stream Invite
c=IN IP4 192.0.2.156
b=AS:3530
t=0 0
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:1450
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:send pt=8;9 SMT:tb=64000:1500
a=bw:recv pt=* AMT:tb=1288000:1500
a=max-recv-ssrc:* 10
a=ssrc:834512974 cname:alice@foo.example.com
m=video 49300 RTP/AVP 96
b=AS:2080
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=bw:send pt=96 SMT:tb=500000:8192
a=bw:recv pt=96 SMT:tb=500000:8192
a=max-recv-ssrc:* 4
a=ssrc:451297483 cname:alice@foo.example.com
a=content:main
```

Alice Offer for a Multi-stream Conference

In the above SDP, Alice proposes one audio and one video RTP session. The audio session has 4 payload types being configured and the different payload configurations also show Alice’s intentions of their different bandwidth usage. For the audio receive direction, Alice accepts an aggregate bandwidth of 1288 kbps with a 1500 byte bucket depth. This is sufficient bandwidth for 10 simultaneous streams. This limit of up to 10 streams being received is additionally indicated on SSRC level using the a=max-recv-ssrc attribute. The send limitation is implicitly set to one by excluding the a=max-send-ssrc attribute. Alice also declares the cname for the SSRC she intends to use.

The video session has only a single payload format using H.264. The configured profile and level is sufficient to support multiple resolutions of interest for the application. Alice indicates the intention to send 640x360 resolution and requests to receive either 640x360 or 320x180. The bandwidth for the video is expressed as the same 500 kbps upper limit in both send and receive directions, with an 8192 bytes bucket depth. There is no explicit limitation on the aggregate bandwidth. Alice does however express that she cannot handle receiving more than 4 simultaneous active SSRCs, so there is an implicit limit.

The application server controlling the conference receives the Offer and constructs a response based on knowledge about the conference and the available translator.

```
v=0
o=server 39451234544 39451234578 IN IP4 198.51.100.2
s=Multi stream Alice Answer
c=IN IP4 198.51.100.43
b=AS:2950
t=0 0
m=audio 49200 RTP/AVP 96 97 9
b=AS:870
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=bw:recv pt=96 SMT:tb=128800:1500
a=bw:recv pt=97 SMT:tb=64800:1500
a=bw:recv pt=9 SMT:tb=64000:1500
a=bw:send pt=* AMT:tb=500000:1500
a=max-send-ssrc:* 6
a=ssrc:239245219 cname:bob@foo.example.com
a=ssrc:986545121 cname:dave@foo.example.com
a=ssrc:2199983234 cname:fred@foo.example.com
m=video 49300 RTP/AVP 96
b=AS:2080
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* recv [x=640,y=360] send [x=640,y=360] [x=320,y=180]
a=bw:recv pt=96 SMT:tb=500000:8192
a=bw:send pt=96 SMT:tb=500000:8192
a=max-send-ssrc:* 4
a=ssrc:924521923 cname:bob@foo.example.com
a=ssrc:654512198 cname:dave@foo.example.com
a=ssrc:3234219998 cname:fred@foo.example.com
a=content:main
```

SDP Answer to Alice from application server

The application server accepts both audio and video RTP sessions. It removed the a-law PCM format as it isn’t needed in this conference. It also reduces the number of simultaneous streams that may occur to 6 by setting the a=max-send-ssrc attribute to 6. The aggregate bandwidth that the client may receive, i.e. what the server declares as send, is limited down 500 kbps with a bucket depth of 1500 bytes. The SSRC values and their CNAMEs from the 3 already connected clients, bob, dave and fred are also included.

The video session is accepted as is, indicated by reversing the directions on the parts that indicates direction in the bw attribute and the imageattr. The max-recv-ssrc is changed to max-send-ssrc to indicate that there may be up to 4 simultaneous sources from the translator down to alice. The SSRCs and the corresponding CNAMEs are also declared for video allowing for audio and video to be bound together, enabling synchronization before receiving the first RTCP sender reports.

8.5.1.2. Multiple Sources from Telepresence Room

In this use case Alice is an end-point which is a telepresence room. It has 3 cameras to cover different parts of the room's table. It also has directional microphones for each camera sector, such that it requests to send 3 streams of audio to maintain audio to screen bindings. If this is not possible, a stereo field sound mix can be provided instead that covers all three cameras.

Alice communicates directly with another single telepresence room end-point, Bob, but with only 2 cameras and microphones. However, Bob can receive 3 simultaneous streams and can use them in the local playout layout.

Alice invites herself into the conference by sending the following SDP offer:

```
v=0
o=alice 2890844526 2890842807 IN IP4 192.0.2.156
s=Telepresence Alice Invite
c=IN IP4 192.0.2.156
b=AS:8965
t=0 0
m=audio 49200 RTP/AVP 97 96
b=AS:725
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:recv pt=* AMT:tb=644000:1500
a=max-recv-ssrc:* 5
a=max-send-ssrc:97 3
a=max-send-ssrc:96 1
a=ssrc:239245219 cname:alice@foo.example.com
a=ssrc:239245219 srcname:a3:d3:4b:f1:22:12
a=ssrc:986545121 cname:alice@foo.example.com
a=ssrc:986545121 srcname:12:3f:ab:d2:ec:32
a=ssrc:2199983234 cname:alice@foo.example.com
a=ssrc:2199983234 srcname:7f:12:db:87:2d:52
m=video 49300 RTP/AVP 96
b=AS:8240
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=1280,y=720] recv [x=1280,y=720]
a=bw:send pt=96 SMT:tb=2500000:8192
a=bw:recv pt=96 SMT:tb=3000000:8192
a=bw:send pt=* AMT:tb=8000000:16384
a=max-recv-ssrc:* 5
a=max-send-ssrc:* 3
a=ssrc:245219239 cname:alice@foo.example.com
a=ssrc:245219239 srcname:a3:d3:4b:f1:22:12
a=ssrc:545121986 cname:alice@foo.example.com
a=ssrc:545121986 srcname:12:3f:ab:d2:ec:32
a=ssrc:199983234 cname:alice@foo.example.com
a=ssrc:199983234 srcname:7f:12:db:87:2d:52
a=content:main
```

Telepresence room Offer for a point to point session

Alice invites Bob into a session where Alice proposes one audio and one video RTP session, both with multiple streams. The audio session is proposing to use 3 mono streams of G.719 (pt=97) as being more prioritized than a single stereo G.719 (pt=96). It also states that it is willing to accept up to 5 simultaneous audio streams from Bob independent of payload type. The end-point also declares the SSRC it intends to use with bindings to CNAME and SRCNAME, enabling Bob to bind together the audio and the video streams that come from the same part of the conference table.

The video session only configures H.264 payload format and states that it intends to send 1280x720 resolution and requests to receive the same. Alice also states that she will put the upper limit of the streams it sends to 2500 kbps with 8192 bytes bucket depth, while it will accept to receive individual streams that are up to 3000 kbps with 8192 bytes bucket depth. However, it also promises to limit the aggregate to no more than 8000 kbps and 16384 of bucket depth for the combination of all three streams it intends to send. Alice is willing to receive up to 5 streams of video simultaneous. Also here Alice informs Bob of the SSRC and their bindings to CNAME and SRCNAME.

Bob process this invite and constructs a SDP answer to be delivered to Alice. As Bob only has two cameras and microphones it will indicate this from its side. However, it is capable of receiving Alice 3 streams without any issues.

```
v=0
o=bob 2890847754 28908477889 IN IP4 198.51.100.21
s=Telepresence Bob Response
c=IN IP4 198.51.100.21
b=AS:8528
t=0 0
m=audio 49200 RTP/AVP 97 96
b=AS:288
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:send pt=* AMT:tb=136000:1500
a=bw:recv pt=* AMT:tb=240000:1500
a=max-recv-ssrc:* 3
a=max-send-ssrc:97 2
a=max-send-ssrc:96 1
a=ssrc:52037639 cname:bob@foo.example.com
a=ssrc:52037639 srcname:37:ee:ca:38:01:3c
a=ssrc:820545843 cname:bob@foo.example.com
a=ssrc:820545843 srcname:20:85:17:48:75:a4
m=video 49300 RTP/AVP 96
b=AS:8240
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=1280,y=720] recv [x=1280,y=720]
a=bw:recv pt=96 SMT:tb=2500000:8192
a=bw:send pt=96 SMT:tb=3000000:8192
a=bw:send pt=* AMT:tb=6000000:16384
a=bw:recv pt=* AMT:tb=8000000:16384
a=max-recv-ssrc:* 3
a=max-send-ssrc:* 2
a=ssrc:911548031 cname:bob@foo.example.com
a=ssrc:911548031 srcname:37:ee:ca:38:01:3c
a=ssrc:586599792 cname:bob@foo.example.com
a=ssrc:586599792 srcname:20:85:17:48:75:a4
a=content:main
```

Telepresence room Answer for a point to point session

So Bob accepts the audio codec configurations but changes the aggregate bandwidths to what it is going to send itself and creates a limitation for Alice based on three mono streams. It confirms the number of streams Alice intends to be sending by including a=max-recv-ssrc:* 3. It also declares that it intends to send either two mono or one stereo stream. Bob also provides its configuration for SSRC and their mapping of CNAME and SRCNAME.

For video it is very similar, the number of streams Bob intends to send is stated as 2 and it also accept the 3 streams Alice intended to send in the max-recv-ssrc attribute. The bandwidth for these streams is accepted as suggested by Bob, keeping the upper limit for the individual streams at 3000 kbps and 8192 bytes depth. It also adds a total in Bob send direction that is twice the individual streams. It also confirms Alice’s limitation for the aggregate. Finally the SSRCs for video are also declared and their bindings to CNAME and SRCNAME.

8.5.2. Simulcast Signaling

This example is for a case of client to video conference service using a centralized media topology with an RTP mixer. Alice, Bob calls into a conference server for a conference call with audio and video to the RTP mixer, these clients being capable to send a few video simulcast versions. The conference server also dials out to Fred, which is a legacy client resulting in fallback behavior. When dialing out to Joe more success is achieved as Joe is a client similar to Alice.

```
+---+      +-----------+      +---+
| A |<---->|           |<---->| B |
+---+      |           |      +---+
           |   Mixer   |
+---+      |           |      +---+
| F |<---->|           |<---->| J |
+---+      +-----------+      +---+
```

Four-party Mixer-based Conference

Example of Media plane for RTP mixer based multi-party conference with 4 participants.

8.5.2.1. Alice: Desktop Client

Alice is calling in to the mixer with an audiovisual single stream desktop client, only adding capability to send simulcast, announce SRCNAME and use of the new directional bandwidth attribute from Section 5.2 compared to a legacy client. The offer from Alice looks like

```
v=0
o=alice 2362969037 2362969040 IN IP4 203.0.113.156
s=Simulcast enabled Desktop Client
t=0 0
c=IN IP4 203.0.113.156
b=AS:825
a=group:SCS 2 3
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:145
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:send pt=8;9 SMT:tb=64000:1500
a=bw:recv pt=* AMT:tb=128800:1500
a=ssrc:521923924 cname:alice@foo.example.com
a=ssrc:521923924 srcname:a3:d3:4b:f1:22:12
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:520
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=bw:send pt=96 SMT:tb=500000:8192
a=bw:recv pt=96 SMT:tb=500000:8192
a=ssrc:192392452 cname:alice@foo.example.com
a=ssrc:192392452 srcname:a3:d3:4b:f1:22:12
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:160
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=bw:send pt=96 SMT:tb=128000:4096
a=bw:recv pt=96 SMT:tb=128000:4096
a=ssrc:239245219 cname:alice@foo.example.com
a=ssrc:239245219 srcname:a3:d3:4b:f1:22:12
a=mid:3
a=sendonly
```

Alice Offer for a Simulcast Conference

As can be seen from the SDP, Alice has a simulcast-enabled client and offers two different session-multiplexed simulcast versions sent from her single camera, indicated by the SCS grouping tag and the two media ID's (2 and 3). The first video version with media ID 2 prefers 360p resolution (signaled via imageattr) and the second video version with media ID 3 prefers 180p resolution. The first video media line also acts as the single receive video (making media line sendrecv), while the second video media line is only related to simulcast transmission and is thus offered sendonly. The two simulcast encoding streams and its related audio stream are bound together using SRCNAME SDES item. We also declare the end-point CNAME as all sources belong to the same synchronization context.

Alice uses the a=bw attribute defined in this document, but also uses the less exact, legacy b-line for interoperability. For video in this example, the client offers to send and receive a bandwidth lower than the video codec level maximum, which could for example have been set via some client or user preference, based on known transport limitations or knowledge what bandwidth is reasonable from a quality perspective given that specific codec at the proposed image resolution. The bitrates given in this example are supposed to be aligned with Section 5.2 and are thus based on the RTP payload level, but could also be designed based on another network layer according to the discussion in that section.

8.5.2.2. Bob: Telepresence Room

Bob is calling in to the mixer with a telepresence client that has capability for both sending multi-stream, receiving and local rendering of those multiple streams, as well as sending simulcast versions of the uplink video. More specifically, in this example the client has three cameras, each being sent in three different simulcast versions. In the receive direction, up to two main screens can show video from a (multi-stream) conference participant being active speaker, and still more screen estate can be used to show videos from up to 16 other conference listeners. Each camera has a corresponding (stereo) microphone that can also be negotiated down to mono by removing the stereo payload type from the answer.

```
v=0
o=bob 129384719 9834727 IN IP4 203.0.113.35
s=Simulcast enabled Multi stream Telepresence Client
t=0 0
c=IN IP4 203.0.113.35
b=AS:6035
a=group:SCS 2 3 4
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:435
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
```

```
a=rtpmap:8 PCMA/8000
a=max-send-ssrc:* 3
a=max-recv-ssrc:* 3
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:send pt=8;9 SMT:tb=64000:1500
a=bw:send pt=* AMT:tb=386400:1500
a=bw:recv pt=* AMT:tb=386400:1500
a=ssrc:724847850 cname:bob@foo.example.com
a=ssrc:724847850 srcname:37:ee:ca:38:01:3c
a=ssrc:2847529901 cname:bob@foo.example.com
a=ssrc:2847529901 srcname:20:85:17:48:75:a4
a=ssrc:57289389 cname:bob@foo.example.com
a=ssrc:57289389 srcname:1e:23:97:ab:9e:0c
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:4500
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:* send [x=1280,y=720] recv [x=1280,y=720]
     [x=640,y=360] [x=320,y=180]
a=max-send-ssrc:96 3
a=max-recv-ssrc:96 2
a=bw:send pt=96 SMT:tb=1500000:16384
a=bw:send pt=* AMT:tb=4500000:16384
a=bw:recv pt=96 SMT:tb=1500000:16384
a=bw:recv pt=* AMT:tb=3000000:16384
a=ssrc:75384768 cname:bob@foo.example.com
a=ssrc:75384768 srcname:37:ee:ca:38:01:3c
a=ssrc:2934825991 cname:bob@foo.example.com
a=ssrc:2934825991 srcname:20:85:17:48:75:a4
a=ssrc:3582594238 cname:bob@foo.example.com
a=ssrc:3582594238 srcname:1e:23:97:ab:9e:0c
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:1560
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* send [x=640,y=360]
a=max-send-ssrc:96 3
a=bw:send pt=96 SMT:tb=500000:8192
a=ssrc:1371234978 cname:bob@foo.example.com
a=ssrc:1371234978 srcname:37:ee:ca:38:01:3c
a=ssrc:897234694 cname:bob@foo.example.com
a=ssrc:897234694 srcname:20:85:17:48:75:a4
a=ssrc:239263879 cname:bob@foo.example.com
a=ssrc:239263879 srcname:1e:23:97:ab:9e:0c
```

```
a=mid:3
a=sendonly
m=video 49500 RTP/AVP 96
b=AS:420
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=max-send-ssrc:96 3
a=bw:send pt=96 SMT:tb=128000:4096
a=ssrc:485723998 cname:bob@foo.example.com
a=ssrc:485723998 srcname:37:ee:ca:38:01:3c
a=ssrc:2345798212 cname:bob@foo.example.com
a=ssrc:2345798212 srcname:20:85:17:48:75:a4
a=ssrc:1295729848 cname:bob@foo.example.com
a=ssrc:1295729848 srcname:1e:23:97:ab:9e:0c
a=mid:4
a=sendonly
m=video 49600 RTP/AVP 96 97 98
b=AS:2600
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:96 recv [x=1280,y=720]
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42c01e
a=imageattr:97 recv [x=640,y=360]
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42c00d
a=imageattr:98 recv [x=320,y=180]
a=max-recv-ssrc:96 1
a=max-recv-ssrc:97 4
a=max-recv-ssrc:98 16
a=max-recv-ssrc:* 16
a=bw:recv pt=96 SMT:tb=1500000:16384
a=bw:recv pt=97 SMT:tb=500000:8192
a=bw:recv pt=98 SMT:tb=128000:4096
a=bw:recv pt=* AMT:tb=2500000:16384
a=mid:5
a=recvonly
a=content:alt
```

Bob Offer for a Multi-stream and Simulcast Telepresence Conference

Bob has a three-camera, three-screen, simulcast-enabled client with even higher performance than Alice’s and can additionally support 720p video, as well as multiple receive streams of various resolutions. The client implementor has thus decided to offer three simulcast streams for each camera, indicated by the SCS grouping tag and the three media ID’s (2, 3, and 4) in the SDP.

The first video media line with media ID 2 indicates the ability to send video from three simultaneous video sources (cameras) through the max-send-ssrc attribute with value 3. This media line is also marked as the main video by using the content attribute from [RFC4796]. Also the receive direction has declared ability to handle multiple video sources, and in this example it is 2. The interpretation of content:main for those two streams in the receive direction is that the client expects and can present (in prime position) at most two main (active speaker) video streams from another multi-camera client.

The second and third video media lines with media ID 3 and 4 are the sendonly simulcast streams. They can implicitly through the grouping be interpreted as also being content:main for the send direction, but is not marked as such since multiple media blocks with content:main could be confusing for a legacy client.

The fourth video media line with media ID 5 is recvonly and is marked with content:alt. That media line should, as was intended for that content attribute value, receive alternative content to the main speaker, such as "audience". In a multi-party conference, that could for example be the next-to-most-active speakers. The SDP describes that those streams can be presented in a set of different resolutions, indicated through the different payload types. The maximum number of streams per payload type is indicated through the max-recv-ssrc attribute. In this example, at most one stream can have payload type 96, preferably 720p, as indicated by the related imageattr line. Similarly, at most 4 streams can have payload type 97, preferably using 360p resolution, and at most 16 streams can have payload type 98, preferably of 180p resolution. In any case, there must never be more than 16 simultaneous streams of any payload type, but combinations of payload types may occur, such as for example two streams using payload type 97 and 8 streams using payload type 98.

To be able to relate the three cameras with the three microphones, all media lines that send audio or video use the ssrc attribute from [RFC5576], specifying the same SRCNAME from Section 5.3.2 for the audio and video versions that belong together. The use of this attribute is optional and the information can be retrieved from RTCP reporting, but it will then not be possible to correctly relate audio and video sources until the first RTCP report is received and participants may then seemingly make uncorrelated moves between screens and/or speakers when adjusting possible false correlation assumptions.

The legacy bandwidth reflects only the bandwidth in the receive direction, while the new bw attribute is very specific per direction and per media stream. We do note that the offered bandwidth for transmission express as AS on session level woad be 6985. It is unclear what is the correct interpretation of the legacy bandwidth when there is bandwidth asymmetry.

The answer from a simulcast-enabled RTP mixer to this last SDP could look like:

```
v=0
o=server 238947290 239573929 IN IP4 198.51.100.2
s=Multi stream and Simulcast Telepresence Bob Answer
c=IN IP4 198.51.100.43
b=AS:7065
a=group:SCR 2 3 4
m=audio 49200 RTP/AVP 96
b=AS:435
a=rtpmap:96 G719/48000/2
a=max-send-ssrc:96 3
a=max-recv-ssrc:96 3
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:recv pt=96 SMT:tb=128800:1500
a=bw:send pt=* AMT:tb=386400:1500
a=bw:recv pt=* AMT:tb=386400:1500
a=ssrc:4111848278 cname:server@conf1.example.com
a=ssrc:4111848278 srcname:87:e9:19:29:c1:bb
a=ssrc:835978294 cname:server@conf1.example.com
a=ssrc:835978294 srcname:1f:83:b3:85:62:7a
a=ssrc:2938491278 cname:server@conf1.example.com
a=ssrc:2938491278 srcname:99:76:b4:bb:90:52
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:4650
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:* send [x=1280,y=720] [x=640,y=360] [x=320,y=180]
     recv [x=1280,y=720]
a=max-recv-ssrc:96 3
a=max-send-ssrc:96 2
a=bw:recv pt=96 SMT:tb=1500000:16384
a=bw:recv pt=* AMT:tb=4500000:16384
a=bw:send pt=96 SMT:tb=1500000:16384
a=bw:send pt=* AMT:tb=3000000:16384
a=ssrc:2938746293 cname:server@conf1.example.com
a=ssrc:2938746293 srcname:87:e9:19:29:c1:bb
a=ssrc:1207102398 cname:server@conf1.example.com
a=ssrc:1207102398 srcname:1f:83:b3:85:62:7a
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
```

```
b=AS:1560
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* recv [x=640,y=360]
a=max-recv-ssrc:96 3
a=bw:recv pt=96 SMT:tb=500000:8192
a=mid:3
a=recvonly
m=video 49500 RTP/AVP 96
b=AS:420
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:96 3
a=bw:recv pt=96 SMT:tb=128000:4096
a=mid:4
a=recvonly
m=video 49600 RTP/AVP 96 97 98
b=AS:2600
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:96 send [x=1280,y=720]
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42c01e
a=imageattr:97 send [x=640,y=360]
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42c00d
a=imageattr:98 send [x=320,y=180]
a=max-send-ssrc:96 1
a=max-send-ssrc:97 4
a=max-send-ssrc:98 8
a=max-send-ssrc:* 8
a=bw:send pt=96 SMT:tb=1500000:16384
a=bw:send pt=97 SMT:tb=500000:8192
a=bw:send pt=98 SMT:tb=128000:4096
a=bw:send pt=* AMT:tb=2500000:16384
a=ssrc:2981523948 cname:server@conf1.example.com
a=ssrc:2938237 cname:server@conf1.example.com
a=ssrc:1230495879 cname:server@conf1.example.com
a=ssrc:74835983 cname:server@conf1.example.com
a=ssrc:3928594835 cname:server@conf1.example.com
a=ssrc:948753 cname:server@conf1.example.com
a=ssrc:1293456934 cname:server@conf1.example.com
a=ssrc:4134923746 cname:server@conf1.example.com
a=mid:5
a=sendonly
a=content:alt
```

Server Answer for Bob Multi-stream and Simulcast Telepresence Conference

In this SDP answer, the grouping tag is changed to SCR, confirming that the sent simulcast streams will be received. The directionality of the streams themselves as well as the directionality of multi-stream and bandwidth attributes are changed. Note that the session level legacy bandwidth can be calculated more correctly with support from the bw attribute in the offer than would have been the case if only legacy media level bandwidth was present. Bandwidth bucket size can be adjusted down between the offer and the answer for streams sent from the answerer, indicating a more strict constant bitrate than really needed. The bucket size can be adjusted up or down for streams received by the answerer, indicating a more strict or flexible bitrate constraint, respectively, for the receiver compared to what the sender offered. The number of allowed streams in the content:alt video session has been reduced to 8 in the answer from 16 offered.

Note that the two video sources in the media block with mid:2 correspond to the two first audio sources (matching SRCNAME). The last audio source correspond to all video sources in the media block with mid:5, however SRCNAME can not be used to perform this binding as its semantic doesn’t match.

8.5.2.3. Fred: Dial-out to Legacy Client

Fred has a simple legacy client that know nothing of the new signaling means discussed in this document. In this example, the multi-stream and simulcast aware RTP mixer is calling out to Fred. Even though it is never actually sent, this would be Fred’s offer SDP, should he have called in. It is included here to improve the reader’s understanding of Fred’s response to the conference SDP.

```
v=0
o=fred 82342187 237429834 IN IP4 192.0.2.213
s=Legacy Client
t=0 0
c=IN IP4 192.0.2.213
m=audio 50132 RTP/AVP 9 8
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
m=video 50134 RTP/AVP 96 97
b=AS:405
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
a=rtpmap:97 H263-2000/90000
a=fmtp:97 profile=0;level=30
```

Legacy Client Hypothetical Offer

Fred would offer a single mono audio and a single video, each with a couple of different codec alternatives.

The same conference server as in the previous example is calling out to Fred, offering the full set of multi-stream and simulcast features, with maximum stream and bandwidth limits based on what the server itself can support.

```
v=0
o=server 323439283 2384192332 IN IP4 198.51.100.2
s=Multi stream and Simulcast Dial-out Offer
c=IN IP4 198.51.100.43
b=AS:7065
a=group:SCR 2 3 4
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:435
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=max-send-ssrc:* 4
a=max-recv-ssrc:* 3
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:send pt=8;9 SMT:tb=64000:1500
a=bw:send pt=* AMT:tb=515200:1500
a=bw:recv pt=* AMT:tb=386400:1500
a=ssrc:3293472833 cname:server@conf1.example.com
a=ssrc:3293472833 srcname:28:23:54:39:7a:0e
a=ssrc:1734728348 cname:server@conf1.example.com
a=ssrc:1734728348 srcname:83:88:be:19:a6:15
a=ssrc:1054453769 cname:server@conf1.example.com
a=ssrc:1054453769 srcname:76:91:cc:23:02:68
a=ssrc:3923447729 cname:server@conf1.example.com
a=ssrc:3923447729 srcname:be:73:a6:03:00:82
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:4650
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:* send [x=1280,y=720] [x=640,y=360] [x=320,y=180]
    recv [x=1280,y=720]
a=max-recv-ssrc:96 3
a=max-send-ssrc:96 3
a=bw:recv pt=96 SMT:tb=1500000:16384
a=bw:recv pt=* AMT:tb=4500000:16384
```

```
a=bw:send pt=96 SMT:tb=1500000:16384
a=bw:send pt=* AMT:tb=4500000:16384
a=ssrc:78456398 cname:server@conf1.example.com
a=ssrc:78456398 srcname:28:23:54:39:7a:0e
a=ssrc:3284726348 cname:server@conf1.example.com
a=ssrc:3284726348 srcname:83:88:be:19:a6:15
a=ssrc:2394871293 cname:server@conf1.example.com
a=ssrc:2394871293 srcname:76:91:cc:23:02:68
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:1560
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:* recv [x=640,y=360]
a=max-recv-ssrc:96 3
a=bw:recv pt=96 SMT:tb=500000:8192
a=mid:3
a=recvonly
m=video 49500 RTP/AVP 96
b=AS:420
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:96 3
a=bw:recv pt=96 SMT:tb=128000:4096
a=mid:4
a=recvonly
m=video 49600 RTP/AVP 96 97 98
b=AS:2600
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01f
a=imageattr:96 send [x=1280,y=720]
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42c01e
a=imageattr:97 send [x=640,y=360]
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42c00d
a=imageattr:98 send [x=320,y=180]
a=max-send-ssrc:96 1
a=max-send-ssrc:97 4
a=max-send-ssrc:98 8
a=max-send-ssrc:* 8
a=bw:send pt=96 SMT:tb=1500000:16384
a=bw:send pt=97 SMT:tb=500000:8192
a=bw:send pt=98 SMT:tb=128000:4096
a=bw:send pt=* AMT:tb=2500000:16384
a=ssrc:2342872394 cname:server@conf1.example.com
```

```
a=ssrc:1283741823 cname:server@conf1.example.com
a=ssrc:3294823947 cname:server@conf1.example.com
a=ssrc:1020408838 cname:server@conf1.example.com
a=ssrc:1999343791 cname:server@conf1.example.com
a=ssrc:2934192349 cname:server@conf1.example.com
a=ssrc:2234347728 cname:server@conf1.example.com
a=ssrc:3224283479 cname:server@conf1.example.com
a=mid:5
a=sendonly
a=content:alt
```

Server Dial-out Offer with Multi-stream and Simulcast

The answer from Fred to this offer would look like:

```
v=0
o=fred 9842793823 239482793 IN IP4 192.0.2.213
s=Legacy Client Answer to Server Dial-out
t=0 0
c=IN IP4 192.0.2.213
m=audio 50132 RTP/AVP 9
b=AS:80
a=rtpmap:9 G722/8000
m=video 50134 RTP/AVP 96
b=AS:405
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
m=video 0 RTP/AVP 96
m=video 0 RTP/AVP 96
m=video 0 RTP/AVP 96
```

Legacy Client Answer to Server Dial-out

as can be seen from the hypothetical offer, Fred does not understand any of the multistream or simulcast attributes, and does also not understand the grouping framework. Thus, all those lines are removed from the answer SDP and any surplus video media blocks except for the first are rejected. The media bandwidth are adjusted down to what Fred actually accepts to receive.

8.5.2.4. Joe: Dial-out to Desktop Client

This example is almost identical to the one above, with the difference that the answering end-point has some limited simulcast and multi-stream capability. As above this is the offer SDP that Joe would have used, should he have called in.

```
v=0
```

```
o=joe 82342187 237429834 IN IP4 192.0.2.213
s=Simulcast and Multistream enabled Desktop Client
t=0 0
c=IN IP4 192.0.2.213
b=AS:985
a=group:SCS 2 3
m=audio 49200 RTP/AVP 96 97 9 8
b=AS:145
a=rtpmap:96 G719/48000/2
a=rtpmap:97 G719/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:send pt=97 SMT:tb=64800:1500
a=bw:send pt=8;9 SMT:tb=64000:1500
a=bw:recv pt=* AMT:tb=128800:1500
a=ssrc:1223883729 cname:joe@foo.example.com
a=ssrc:1223883729 srcname:12:88:07:cf:81:65
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:520
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:96 send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=bw:send pt=96 SMT:tb=500000:8192
a=bw:recv pt=96 SMT:tb=500000:8192
a=ssrc:3842394823 cname:joe@foo.example.com
a=ssrc:3842394823 srcname:12:88:07:cf:81:65
a=mid:2
a=content:main
m=video 49400 RTP/AVP 96
b=AS:160
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
a=imageattr:96 send [x=320,y=180]
a=bw:send pt=96 SMT:tb=128000:4096
a=bw:recv pt=96 SMT:tb=128000:4096
a=ssrc:1214232284 cname:joe@foo.example.com
a=ssrc:1214232284 srcname:12:88:07:cf:81:65
a=mid:3
a=sendonly
m=video 49300 RTP/AVP 96
b=AS:320
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:* 2
a=bw:recv pt=96 SMT:tb=128000:4096
```

```
a=bw:recv pt=96 AMT:tb=256000:4096
a=mid:4
a=recvonly
a=content:alt
```

Desktop Client Hypothetical Offer

Joe would send two versions of simulcast, 360p and 180p, from a single camera and can receive three sources of multi-stream, one 360p and two 180p streams.

Again, the same conference server is calling out to Joe and the offer SDP from the server would be almost identical to the one in the previous example. It is therefore not included here. The response from Joe would look like:

```
v=0
o=joe 239482639 4702341992 IN IP4 192.0.2.213
s=Answer from Desktop Client to Server Dial-out
t=0 0
c=IN IP4 192.0.2.213
b=AS:985
a=group:SCS 2 3
m=audio 49200 RTP/AVP 96
b=AS:145
a=rtpmap:96 G719/48000/2
a=bw:send pt=96 SMT:tb=128800:1500
a=bw:recv pt=* AMT:tb=128800:1500
a=ssrc:1223883729 cname:joe@foo.example.com
a=ssrc:1223883729 srcname:12:88:07:cf:81:65
a=mid:1
m=video 49300 RTP/AVP 96
b=AS:520
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c01e
a=imageattr:96 send [x=640,y=360] recv [x=640,y=360] [x=320,y=180]
a=bw:send pt=96 SMT:tb=500000:8192
a=bw:recv pt=96 SMT:tb=500000:8192
a=ssrc:3842394823 cname:joe@foo.example.com
a=ssrc:3842394823 srcname:12:88:07:cf:81:65
a=mid:2
a=content:main
m=video 0 RTP/AVP 96
a=mid:3
m=video 49400 RTP/AVP 96
b=AS:160
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00d
```

```
a=imageattr:96 send [x=320,y=180]
a=bw:send pt=96 SMT:tb=128000:4096
a=bw:recv pt=96 SMT:tb=128000:4096
a=ssrc:1214232284 cname:joe@foo.example.com
a=ssrc:1214232284 srcname:12:88:07:cf:81:65
a=mid:4
a=sendonly
m=video 49300 RTP/AVP 96
b=AS:320
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42c00c
a=imageattr:96 recv [x=320,y=180]
a=max-recv-ssrc:* 2
a=bw:recv pt=96 SMT:tb=128000:4096
a=bw:recv pt=96 AMT:tb=256000:4096
a=mid:5
a=recvonly
a=content:alt
```

Desktop Client Answer to Server Dial-out

Since the RTP mixer support all of the features that Joe does and more, the SDP does not differ much from what it should have been in an offer. It can be noted that as stated in [RFC5888], all media lines need mid attributes, even the rejected ones, which is why mid:3 is present even though the mid quality simulcast version is rejected by Joe.

9. IANA Considerations

Following the guidelines in [RFC4566], in [RFC5888], and in [RFC3550], the IANA is requested to register:

1. The SID grouping tag to be used with the grouping framework, as defined in Section 8.1.1

2. A new SDES Item named SRCNAME, as defined in Section 5.3.1

3. The max-send-ssrc and max-recv-ssrc SDP attributes as defined in Section 5.1

4. The bw attribute as defined in Section 5.2

5. The bw attribute scope registry rules

6. The bw attribute semantics registry rules

10. Security Considerations

There is minimal difference in security between the simulcast solutions. Session multiplexing may have some additional overhead in the key-management, but that is minor as most key management schemes can be performed in parallel.

The multi-stream signalling has as other SDP based signalling issues with man in the middles that may modify the SDP as an attack on either the service in general or a particular end-point. This can as usual be resolved by a security mechanism that provides integrity and source authentication between the signalling peers.

The SDES SRCNAME being opaque identifiers could potentially carry additional meanings or function as overt channel. If the SRCNAME would be permanent between sessions, they have the potential for compromising the users privacy as they can be tracked between sessions. See RFC6222 for more discussion.

11. Acknowledgements

12. References

12.1. 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.

[RFC6222] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.

12.2. Informative References

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

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

[RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.

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

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

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

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

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

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

[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.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, May 2011.

Authors' Addresses

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 7141311
Email: bo.burman@ericsson.com