

Internet Engineering Task Force (IETF)  
Request for Comments: 6236  
Category: Standards Track  
ISSN: 2070-1721

I. Johansson  
Ericsson AB  
K. Jung  
Samsung Electronics Co., Ltd.  
May 2011

## **Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)**

### Abstract

This document proposes a new generic session setup attribute to make it possible to negotiate different image attributes such as image size. A possible use case is to make it possible for a low-end handheld terminal to display video without the need to rescale the image, something that may consume large amounts of memory and processing power. The document also helps to maintain an optimal bitrate for video as only the image size that is desired by the receiver is transmitted.

### Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6236>.

## Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">1.1.</a>	<a href="#">Requirements . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Conventions Used in This Document . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">Specification of the 'imageattr' SDP Attribute . . . . .</a>	<a href="#">5</a>
<a href="#">3.1.</a>	<a href="#">Attribute Syntax . . . . .</a>	<a href="#">5</a>
<a href="#">3.1.1.</a>	<a href="#">Overall View of Syntax . . . . .</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">Considerations . . . . .</a>	<a href="#">11</a>
<a href="#">3.2.1.</a>	<a href="#">No imageattr in First Offer . . . . .</a>	<a href="#">11</a>
<a href="#">3.2.2.</a>	<a href="#">Different Payload Type Numbers in Offer and Answer . . . . .</a>	<a href="#">11</a>
<a href="#">3.2.3.</a>	<a href="#">Asymmetry . . . . .</a>	<a href="#">12</a>
<a href="#">3.2.4.</a>	<a href="#">sendonly and recvonly . . . . .</a>	<a href="#">12</a>
<a href="#">3.2.5.</a>	<a href="#">Sample Aspect Ratio . . . . .</a>	<a href="#">13</a>
<a href="#">3.2.6.</a>	<a href="#">SDPCapNeg Support . . . . .</a>	<a href="#">13</a>
<a href="#">3.2.7.</a>	<a href="#">Interaction with Codec Parameters . . . . .</a>	<a href="#">14</a>
<a href="#">3.2.8.</a>	<a href="#">Change of Display in Middle of Session . . . . .</a>	<a href="#">16</a>
<a href="#">3.2.9.</a>	<a href="#">Use with Layered Codecs . . . . .</a>	<a href="#">16</a>
<a href="#">3.2.10.</a>	<a href="#">Addition of Parameters . . . . .</a>	<a href="#">16</a>
<a href="#">4.</a>	<a href="#">Examples . . . . .</a>	<a href="#">16</a>
<a href="#">4.1.</a>	<a href="#">A High-Level Example . . . . .</a>	<a href="#">16</a>
<a href="#">4.2.</a>	<a href="#">Detailed Examples . . . . .</a>	<a href="#">17</a>
<a href="#">4.2.1.</a>	<a href="#">Example 1 . . . . .</a>	<a href="#">17</a>
<a href="#">4.2.2.</a>	<a href="#">Example 2 . . . . .</a>	<a href="#">18</a>
<a href="#">4.2.3.</a>	<a href="#">Example 3 . . . . .</a>	<a href="#">19</a>
<a href="#">4.2.4.</a>	<a href="#">Example 4 . . . . .</a>	<a href="#">20</a>
<a href="#">5.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">20</a>
<a href="#">6.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">21</a>
<a href="#">7.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">21</a>
<a href="#">8.</a>	<a href="#">References . . . . .</a>	<a href="#">22</a>
<a href="#">8.1.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">22</a>
<a href="#">8.2.</a>	<a href="#">Informative References . . . . .</a>	<a href="#">22</a>



## 1. Introduction

This document proposes a new SDP attribute to make it possible to negotiate different image attributes, such as image size. The term image size is defined here, as it may differ from the physical screen size of, for instance, a hand-held terminal. As an example, it may be beneficial to display a video image on a part of the physical screen and leave space on the screen for other features such as menus and other info.

Allowing negotiation of the image size provides a number of benefits:

- o Less image distortion: Rescaling of images introduces additional distortion, something that can be avoided (at least on the receiver side) if the image size can be negotiated.
- o Reduced receiver complexity: Image rescaling can be quite computation intensive. For low-end devices, this can be a problem.
- o Optimal quality for the given bitrate: The sender does not need to encode an entire CIF (352x288) image if only an image size of 288x256 is displayed on the receiver screen.
- o Memory requirement: The receiver device will know the size of the image and can then allocate memory accordingly.
- o Optimal aspect ratio: The indication of the supported image sizes and aspect ratio allows the receiver to select the most appropriate combination based on its rescaling capabilities and the desired rendering. For example, if a sender proposes three resolutions in its SDP offer (100x200, 200x100, and 100x100) with sar=1.0 (1:1) etc., then the receiver can select the option that fits the receiver screen best.

In cases where rescaling is not implemented (for example, rescaling is not mandatory to implement in H.264 [[H.264](#)]), the indication of the image attributes may still provide an optimal use of bandwidth because the attribute will give the encoder a better indication about what image size is preferred anyway and will thus help to avoid wasting bandwidth by encoding with an unnecessarily large resolution.

For implementers that are considering rescaling issues, it is worth noting that there are several benefits to doing it on the sender side:

- o Rescaling on the sender/encoder side is likely to be easier to do as the camera-related software/hardware already contains the



necessary functionality for zooming/cropping/trimming/sharpening the video signal. Moreover, rescaling is generally done in RGB or YUV domains and should not depend on the codecs used.

- o The encoder may be able to encode in a number of formats but may not know which format to choose as, without the image attribute, it does not know the receiver's performance or preference.
- o The quality drop due to digital domain rescaling using interpolation is likely to be lower if it is done before the video encoding rather than after the decoding especially when low bitrate video coding is used.
- o If low-complexity rescaling operations such as simple cropping must be performed, the benefit with having this functionality on the sender side is that it is then possible to present a miniature "what you send" image on the display to help the user to frame the image correctly.

Several of the existing standards ([[H.263](#)], [[H.264](#)], and [[MPEG-4](#)]) have support for different resolutions at different framerates. The purpose of this document is to provide for a generic mechanism, which is targeted mainly at the negotiation of the image size. However, to make it more general, the attribute is named 'imageattr'.

This document is limited to point-to-point unicast communication scenarios. The attribute may be used in centralized conferencing scenarios as well but due to the abundance of configuration options, it may then be difficult to come up with a configuration that fits all parties.

### **1.1. Requirements**

The design of the image attribute is based on the following requirements, which are listed only for informational purposes:

- REQ-1: Support the indication of one or more set(s) of image attributes that the SDP endpoint wishes to receive or send. Each image attribute set must include a specific image size.
- REQ-2: Support setup/negotiation of image attributes, meaning that each side in the Offer/Answer should be able to negotiate the image attributes it prefers to send and receive.
- REQ-3: Interoperate with codec-specific parameters such as sprop-parameter-sets in [[H.264](#)] or config in [[MPEG-4](#)].



REQ-4: Make the attribute generic with as few codec specific details/tricks as possible in order to be codec agnostic.

Besides the above mentioned requirements, the requirement below may be applicable.

OPT-1: The image attribute should support the description of image-related attributes for various types of media, including video, pictures, images, etc.

## 2. Conventions Used in This Document

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. Specification of the 'imageattr' SDP Attribute

This section defines the SDP image attribute 'imageattr', which can be used in an SDP Offer/Answer exchange to indicate various image attribute parameters. In this document, we define the following image attribute parameters: image resolution, sample aspect ratio (sar), allowed range in picture aspect ratio (par) and the preference of a given parameter set over another (q). The attribute is extensible and guidelines for defining additional parameters are provided in [Section 3.2.10](#).

### 3.1. Attribute Syntax

In this section, the syntax of the 'imageattr' attribute is described. The 'imageattr' attribute is a media-level attribute. The section is split up in two parts: the first gives an overall view of the syntax, and the second describes how the syntax is used.

#### 3.1.1. Overall View of Syntax

The syntax for the image attribute is in ABNF [\[RFC5234\]](#):

```
image-attr = "imageattr:" PT 1*2( 1*WSP ( "send" / "recv" )
                                   1*WSP attr-list )
PT = 1*DIGIT / "*"
attr-list = ( set *(1*WSP set) ) / "*"
           ; WSP and DIGIT defined in \[RFC5234\]

set= "[" "x=" xyrange "," "y=" xyrange *( "," key-value ) "]"
      ; x is the horizontal image size range (pixel count)
      ; y is the vertical image size range (pixel count)
```





```

key-value = ( "sar=" srange )
            / ( "par=" prange )
            / ( "q=" qvalue )
            ; Key-value MAY be extended with other keyword
            ; parameters.
            ; At most, one instance each of sar, par, or q
            ; is allowed in a set.
            ;
            ; sar (sample aspect ratio) is the sample aspect ratio
            ; associated with the set (optional, MAY be ignored)
            ; par (picture aspect ratio) is the allowed
            ; ratio between the display's x and y physical
            ; size (optional)
            ; q (optional, range [0.0..1.0], default value 0.5)
            ; is the preference for the given set,
            ; a higher value means a higher preference

onetonine = "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"
            ; Digit between 1 and 9

xyvalue = onetonine *5DIGIT
            ; Digit between 1 and 9 that is
            ; followed by 0 to 5 other digits

step = xyvalue

xyrange = ( "[" xyvalue ":" [ step ":" ] xyvalue "]" )
            ; Range between a lower and an upper value
            ; with an optional step, default step = 1
            ; The rightmost occurrence of xyvalue MUST have a
            ; higher value than the leftmost occurrence.
            / ( "[" xyvalue 1*( "," xyvalue ) "]" )
            ; Discrete values separated by ','
            / ( xyvalue )
            ; A single value

spvalue = ( "0" "." onetonine *3DIGIT )
            ; Values between 0.1000 and 0.9999
            / ( onetonine "." 1*4DIGIT )
            ; Values between 1.0000 and 9.9999

srange = ( "[" spvalue 1*( "," spvalue ) "]" )
            ; Discrete values separated by ','.
            ; Each occurrence of spvalue MUST be
            ; greater than the previous occurrence.
            / ( "[" spvalue "-" spvalue "]" )
            ; Range between a lower and an upper level (inclusive)
            ; The second occurrence of spvalue MUST have a higher
            ; value than the first
            / ( spvalue )
            ; A single value

```



```
prange = ( "[" spvalue "-" spvalue "]" )  
          ; Range between a lower and an upper level (inclusive)  
          ; The second occurrence of spvalue MUST have a higher  
          ; value than the first
```

```
qvalue = ( "0" "." 1*2DIGIT )  
          / ( "1" "." 1*2("0") )  
          ; Values between 0.00 and 1.00
```

- o The attribute typically contains a "send" and a "recv" keyword. These specify the preferences for the media once the session is set up, in the send and receive direction respectively from the point of view of the sender of the session description. One of the keywords ("send" or "recv") MAY be omitted; see [Section 3.2.4](#) and [Section 3.2.2](#) for a description of cases when this may be appropriate.
- o The "send" keyword and corresponding attribute list (attr-list) MUST NOT occur more than once per image attribute.
- o The "recv" keyword and corresponding attribute list (attr-list) MUST NOT occur more than once per image attribute.
- o PT is the payload type number; it MAY be set to "\*" (wild card) to indicate that the attribute applies to all payload types in the media description.
- o For sendrecv streams, both of the send and recv directions SHOULD be present in the SDP.
- o For inactive streams it is RECOMMENDED that both of the send and recv directions are present in the SDP.

#### [3.1.1.1](#). Parameter Rules

The following rules apply for the parameters.

Payload type number (PT): The image attribute is bound to a specific codec by means of the payload type number. A wild card (\*) can be specified for the payload type number to indicate that it applies to all payload types in the media description. Several image attributes can be defined, for instance for different video codec alternatives. This however requires that the payload type numbers differ. Note that the attribute is associated to the codec(s), for instance an SDP offer may specify payload type number 101 while the answer may specify 102, this may make it troublesome to



specify a payload type number with the 'imageattr' attribute. See [Section 3.2.2](#) for a discussion and recommendation how this is solved.

Preference (q): The preference for each set is 0.5 by default; setting the optional q parameter to another value makes it possible to set different preferences for the sets. A higher value gives a higher preference for the given set.

sar: The sar (storage aspect ratio) parameter specifies the sample aspect ratio associated to the given range of x and y values. The sar parameter is defined as  $dx/dy$  where dx and dy are the physical size of the pixels. Square pixels gives a sar=1.0. The parameter sar MAY be expressed as a range or as a single value.

If this parameter is not present, a default sar value of 1.0 is assumed.

The interpretation of sar differs between the send and the receive directions.

- \* In the send direction, sar defines a specific sample aspect ratio associated to a given x and y image size (range).
- \* In the recv direction, sar expresses that the receiver of the given medium prefers to receive a given x and y resolution with a given sample aspect ratio.

See [Section 3.2.5](#) for a more detailed discussion.

The sar parameter will likely not solve all the issues that are related to different sample aspect ratios, but it can help to solve them and reduce aspect ratio distortion.

The response MUST NOT include a sar parameter if there is no acceptable value given. The reason for this is that if the response includes a sar parameter it is interpreted as "sar parameter accepted", while removal of the sar parameter is treated as "sar parameter not accepted". For this reason, it is safer to remove an unacceptable sar parameter altogether.

par: The par (width/height = x/y ratio) parameter indicates a range of allowed ratios between x and y physical size (picture aspect ratio). This is used to limit the number of x and y image size combinations; par is given as

par=[ratio\_min-ratio\_max]



where `ratio_min` and `ratio_max` are the min and max allowed picture aspect ratios.

If `sar` and the sample aspect ratio that the receiver actually uses in the display are the same (or close), the relation between the x and y pixel resolution and the physical size of the image is straightforward. If however `sar` differs from the sample aspect ratio of the receiver display, this must be taken into consideration when the x and y pixel resolution alternatives are sorted out. See [Section 4.2.4](#) for an example of this.

#### **3.1.1.2. Offer/Answer Rules**

In accordance with [\[RFC3264\]](#), offer/answer exchange of the image attribute is as follows.

o Offerer sending the offer:

- \* The offerer must be able to support the image attributes that it offers, unless the offerer has expressed a wild card (\*) in the attribute list.
- \* It is recommended that a device that sees no reason to use the image attribute includes the attribute with wild cards (\*) in the attribute lists anyway for the send and recv directions. An example of this looks like:

```
a=imageattr:97 send * recv *
```

This gives the answerer the possibility of expressing its preferences. The use of wild cards introduces a risk that the message size can increase in an uncontrolled way. To reduce this risk, these wild cards SHOULD only be replaced by an as small set as possible.

o Answerer receiving the offer and sending the answer:

- \* The answerer may choose to keep the image attribute but is not required to do so.
- \* The answerer may, for its receive and send direction, include one or more entries that it can support from the set of entries proposed in the offer.
- \* The answerer may also, for its receive and send direction, replace the entries with a complete new set of entries different from the original proposed by the offerer. The





implementor of this feature should however be aware that this may cause extra offer/answer exchanges.

- \* The answerer may also remove its send direction completely if it is deemed that it cannot support any of the proposed entries.
- \* The answerer should not include an image attribute in the answer if it was not present in the offer.

o Offerer receiving the answer:

- \* If the image attribute is not included in the SDP answer the offerer SHOULD continue to process the answer as if this mechanism had not been offered.
- \* If the image attribute is included in the SDP answer but none of the entries are usable or acceptable, the offerer MUST resort to other methods to determine the appropriate image size. In this case, the offerer must also issue a new offer/answer without the image attribute to avoid misunderstandings between the offerer and answerer. This will avoid the risk of infinite negotiations.

## **3.2. Considerations**

### **3.2.1. No imageattr in First Offer**

When the initial offer does not contain the 'imageattr' attribute, the rules in [Section 3.1.1.2](#) require the attribute to be absent in the answer. The reasons for this are:

- o The offerer of the initial SDP is not likely to understand the image attribute if it did not include it in the offer, bearing in mind that [Section 3.1.1](#) recommends that the offerer provide the attribute with wild carded parameters if it has no preference.
- o Inclusion of the image attribute in the answer may come in conflict with the rules in [Section 3.1.1.2](#), especially the rules that apply to "offerer receiving the answer".

For the above reasons, it is RECOMMENDED that a device that sees no reason to use the image attribute includes the attribute with wild cards (\*) in the attribute lists anyway for the send and recv directions.



### 3.2.2. Different Payload Type Numbers in Offer and Answer

In some cases, the answer may specify a different media payload type number than the offer. As an example, the offer SDP may have the m-line

```
m=video 49154 RTP/AVP 99
```

while the answer SDP may have the m-line

```
m=video 49154 RTP/AVP 100
```

If the image attribute in the offer specifies payload type number 99, this attribute will then have no meaning in the answerers receive direction as the m-line specifies media payload type number 100.

There are a few ways to solve this.

1. Use a wild card "\*" as the payload type number in the image attribute in the offer SDP. The answer SDP also uses the wild card. The drawback with this approach is that this attribute then applies to all payload type numbers in the media description.
2. Specify a wild card "\*" as the payload type number in the image attribute in the answer SDP. The offer SDP may contain a defined payload type number in the image attribute but the answer SDP replaces this with a wild card. The drawback here is similar to what is listed above.
3. The image attribute is split in two parts in the SDP answer. For example the offer SDP (only the parts of interest in this discussion) looks like:

```
m=video 49154 RTP/AVP 99  
a=imageattr:99 send ... recv ...
```

The answer SDP looks like:

```
m=video 49154 RTP/AVP 100  
a=imageattr:99 send ...  
a=imageattr:100 recv ...
```

This alternative does not pose any drawbacks. Moreover, it allows specification of different image attributes if more than one payload type is specified in the offer SDP.



Of the alternatives listed above, the last one **MUST** be used as it is the most safe. The other alternatives **MUST NOT** be used.

### **3.2.3. Asymmetry**

While the image attribute supports asymmetry, there are some limitations. One important limitation is that the codec being used can only support up to a given maximum resolution for a given profile level.

As an example, H.264 [[H.264](#)] with profile level 1.2 does not support higher resolution than 352x288 (CIF). The offer/answer rules imply that the same profile level must be used in both directions. This means that in an asymmetric scenario where Alice wants an image size of 580x360 and Bob wants 150x120, profile level 2.2 is needed in both directions even though profile level 1 would have been sufficient in one direction.

Currently, the only solution to this problem is to specify two unidirectional media descriptions. Note however that the asymmetry issue for the H.264 codec is solved by means of the level-asymmetry-allowed parameter in [[RFC6184](#)].

### **3.2.4. sendonly and recvonly**

If the directional attributes `a=sendonly` or `a=recvonly` are given for a medium, there is of course no need to specify the image attribute for both directions. Therefore, one of the directions in the attribute may be omitted. However, it may be good to do the image attribute negotiation in both directions in case the session is updated for media in both directions at a later stage.

### **3.2.5. Sample Aspect Ratio**

The relationship between the `sar` parameter and the `x` and `y` pixel resolution deserves some extra discussion. Consider the offer from Alice to Bob (we set the `recv` direction aside for the moment):

```
a=imageattr:97 send [x=720,y=576,sar=1.1]
```

If the receiver display has square pixels, the 720x576 image would need to be rescaled to for example 792x576 or 720x524 to ensure a correct image aspect ratio. This in practice means that rescaling would need to be performed on the receiver side, something that is contrary to the spirit of this document.



To avoid this problem Alice may specify a range of values for the sar parameter like:

```
a=imageattr:97 send [x=720,y=576,sar=[0.91,1.0,1.09,1.45]]
```

Meaning that Alice can encode with any of the mentioned sample aspect ratios, leaving Bob to decide which one he prefers.

### **3.2.6. SDPCapNeg Support**

The image attribute can be used within the SDP Capability Negotiation [RFC5939] framework and its use is then specified using the "a=acap" parameter. An example is

```
a=acap:1 imageattr:97 send [x=720,y=576,sar=[0.91,1.0,1.09,1.45]]
```

For use with SDP Media Capability Negotiation extension [SDPMediaCapNeg], where it is no longer possible to specify payload type numbers, it is possible to use the parameter substitution rule, an example of this is

```
...
a=mcap:1 video H264/90000
a=acap:1 imageattr:%1% send [x=720,y=576,sar=[0.91,1.0,1.09,1.45]]
...
```

where %1% maps to media capability number 1.

It is also possible to use the a=mscap attribute like in the example below.

```
...
a=mcap:1 video H264/90000
a=mscap:1 imageattr send [x=720,y=576,sar=[0.91,1.0,1.09,1.45]]
...
```

### **3.2.7. Interaction with Codec Parameters**

As the SDP for most codecs already specifies some kind of indication of, for example, the image size, at session setup, measures must be taken to avoid conflicts between the image attribute and this already existing information.

The following subsections describe the most well known codecs and how they define image-size related information. [Section 3.2.7.4](#) outlines a few possible solutions, but this document does not make a recommendation for any of them.





#### **3.2.7.1. H.263**

The payload format for H.263 [[H.263](#)] is described in [[RFC4629](#)].

H.263 defines (on the fntp line) a list of image sizes and their maximum frame rates (profiles) that the offerer can receive. The answerer is not allowed to modify this list and must reject a payload type that contains an unsupported profile. The CUSTOM profile may be used for image size negotiation but support for asymmetry requires the specification of two unidirectional media descriptions using the sendonly/recvonly attributes.

#### **3.2.7.2. H.264**

The payload format for H.264 [[H.264](#)] is described in [[RFC6184](#)].

H.264 defines information related to image size in the fntp line by means of sprop-parameter-sets. According to the specification, several sprop-parameter-sets may be defined for one payload type. The sprop-parameter-sets describe the image size (+ more) that the offerer sends in the stream and need not be complete. This means that sprop-parameter-sets does not represent any negotiation and the answer is not allowed to change the sprop-parameter-sets.

This configuration may be changed later inband if for instance image sizes need to be changed or added.

#### **3.2.7.3. MPEG-4**

The payload format for MPEG-4 [[MPEG-4](#)] is described in [[RFC3016](#)].

MPEG-4 defines a config parameter on the fntp line, which is a hexadecimal representation of the MPEG-4 visual configuration information. This configuration does not represent any negotiation and the answer is not allowed to change the parameter.

It is not possible to change the configuration using inband signaling.

#### **3.2.7.4. Possible Solutions**

The subsections above clearly indicate that this kind of information must be aligned well with the image attribute to avoid conflicts. There are a number of possible solutions, listed below without any preference:



- o Ignore payload format parameters: This may not work well in the presence of bad channel conditions especially in the beginning of a session. Moreover, this is not a good option for MPEG-4.
- o Second session-wide offer/answer round: In the second offer/answer, the parameters specific to codec payload format are defined based on the outcome of the 'imageattr' negotiation. The drawback with this is that setup of the entire session (including audio) may be delayed considerably, especially as the 'imageattr' negotiation can already itself cost up to two offer/answer rounds. Also, the conflict between the 'imageattr' negotiation and the parameters specific to payload format is still present after the first offer/answer round and a fuzzy/buggy implementation may start media before the second offer/answer is completed with unwanted results.
- o Second session-wide offer/answer round only for video: This is similar to the alternative above with the exception that setup time for audio is not increased; moreover, the port number for video is set to 0 during the first offer answer round to avoid the flow of media.

This has the effect that video will blend in some time after the audio is started (up to 2 seconds delay). This alternative is likely the most clean-cut and failsafe. The drawback is, as the port number in the first offer is always zero, the media startup will always be delayed even though it would in fact have been possible to start media after the first offer/answer round.

Note that according to [[RFC3264](#)], a port number of zero means that the whole media line is rejected, meaning that a new offer for the same port number should be treated as a completely new stream and not as an update. The safest way to solve this problem is to use preconditions; this is however outside the scope of this document.

#### **3.2.8. Change of Display in Middle of Session**

A very likely scenario is that a user switches to another phone during a video telephony call or plugs a cellphone into an external monitor. In both cases, it is very likely that a renegotiation is initiated using the SIP-REFER [[RFC3515](#)] or SIP-UPDATE [[RFC3311](#)] methods. It is RECOMMENDED to negotiate the image size during this renegotiation.



### **3.2.9. Use with Layered Codecs**

As the image attribute is a media-level attribute, its use with layered codecs causes some concern. If the layers are transported in different RTP streams, the layers are specified on different media descriptions, and the relation is specified using the grouping framework [RFC5888] and the depend attribute [RFC5583]. As it is not possible to specify only one image attribute for several media descriptions the solution is either to specify the same image attribute for each media description, or to only specify the image attribute for the base layer.

### **3.2.10. Addition of Parameters**

The image attribute allows for the addition of parameters in the future. To make backwards adaptation possible, an entity that processes the attribute MUST ignore any unknown parameters in the offer and MUST NOT include them in the answer it generates. Addition of future parameters that are not understood by the receiving endpoint may lead to ambiguities if mutual dependencies between parameters exist; therefore, addition of parameters must be done with great care.

## **4. Examples**

This section gives some more information on how to use the attribute by means of a high-level example and a few detailed examples.

### **4.1. A High-Level Example**

Assume that Alice wishes to set up a session with Bob and that Alice takes the first initiative. The syntactical white-space delimiters (1\*WSP) and double-quotes are removed to make reading easier.

In the offer, Alice provides information for both the send and receive (recv) directions. For the send direction, Alice provides a list that the answerer can select from. For the receive direction, Alice may either specify a desired image size range right away or a \* to instruct Bob to reply with a list of image sizes that Bob can support for sending. Using the overall high level syntax the image attribute may then look like

```
a=imageattr:PT send attr-list recv attr-list
```

or

```
a=imageattr:PT send attr-list recv *
```



In the first alternative, the `recv` direction may be a full list of desired image size formats. It may however (and most likely) just be a list with one alternative for the preferred `x` and `y` resolution.

If Bob supports an `x` and `y` resolution in at least one of the `X` and `Y` ranges given in the `send attr-list` and in the `recv attr-list` of the offer, the answer from Bob will look like:

```
a=imageattr:PT send attr-list recv attr-list
```

and the offer/answer negotiation is done. Note that the `attr-list` will likely be pruned in the answer. While it may contain many different alternatives in the offer, it may in the end contain just one or two alternatives.

If Bob does not support any `x` and `y` resolution in one of the provided `send` or `recv` ranges given in the `send attr-list` or in the `recv attr-list`, the corresponding part is removed completely. For instance, if Bob doesn't support any of the offered alternatives in the `recv attr-list` in the offer, the answer from Bob would look like:

```
a=imageattr:PT recv attr-list
```

## 4.2. Detailed Examples

This section gives a few detailed examples. It is assumed where needed that Alice initiates a session with Bob.

### 4.2.1. Example 1

Two image resolution alternatives are offered with 800x640 with `sar=1.1` having the highest preference.

It is also indicated that Alice wishes to display video with a resolution of 330x250 on her display.

```
a=imageattr:97 send [x=800,y=640,sar=1.1,q=0.6] [x=480,y=320] \  
recv [x=330,y=250]
```

In case Bob accepts the "`recv [x=330,y=250]`", the answer may look like

```
a=imageattr:97 recv [x=800,y=640,sar=1.1] \  
send [x=330,y=250]
```

indicating that the receiver (Bob) wishes the encoder (on Alice's side) to compensate for a sample aspect ratio of 1.1 (11:10) and desires an image size on its screen of 800x640.





There is however a possibility that "recv [x=330,y=250]" is not supported. If the case, Bob may completely remove this part or replace it with a list of supported image sizes.

```
a=imageattr:97 recv [x=800,y=640,sar=1.1] \  
                send [x=[320:16:640],y=[240:16:480],par=[1.2-1.3]]
```

Alice can then select a valid image size that is closest to the one that was originally desired (336x256) and performs a second offer/answer.

```
a=imageattr:97 send [x=800,y=640,sar=1.1] \  
                recv [x=336,y=256]
```

Bob replies with:

```
a=imageattr:97 recv [x=800,y=640,sar=1.1] \  
                send [x=336,y=256]
```

#### **4.2.2. Example 2**

Two image resolution sets are offered with the first having a higher preference (q=0.6).

```
a=imageattr:97 \  
    send [x=[480:16:800],y=[320:16:640],par=[1.2-1.3],q=0.6] \  
        [x=[176:8:208],y=[144:8:176],par=[1.2-1.3]] \  
    recv *
```

The x-axis resolution can take the values 480 to 800 in 16 pixels steps and 176 to 208 in 8 pixels steps. The par parameter limits the set of possible x and y screen resolution combinations such that 800x640 (ratio=1.25) is a valid combination while 720x608 (ratio=1.18) or 800x608 (ratio=1.31) are invalid combinations.

For the recv direction (Bob->Alice), Bob is requested to provide a list of supported image sizes.



#### 4.2.3. Example 3

In this example, more of the SDP offer is shown. A complicating factor is that the answerer changes the media payload type number in the offer/answer exchange.

```
m=video 49154 RTP/AVP 99
a=rtpmap:99 H264/900000
a=fmtp:99 packetization-mode=0;profile-level-id=42e011; \
  sprop-parameter-sets=Z0LgC5ZUCg/I,aM4BrFSAA
a=imageattr:99 \
  send [x=176,y=144] [x=224,y=176] [x=272,y=224] [x=320,y=240] \
  recv [x=176,y=144] [x=224,y=176] [x=272,y=224,q=0.6] [x=320,y=240]
```

In the send direction, sprop-parameter-sets is defined for a resolution of 320x240, which is the largest image size offered in the send direction. This means that if 320x240 is selected, no additional offer/answer is necessary. In the receive direction, four alternative image sizes are offered with 272x224 being the preferred choice.

The answer may look like:

```
m=video 49154 RTP/AVP 100
a=rtpmap:100 H264/900000
a=fmtp:100 packetization-mode=0;profile-level-id=42e011; \
  sprop-parameter-sets=Z0LgC5ZUCg/I,aM4BrFSAA
a=imageattr:99 send [x=320,y=240]
a=imageattr:100 recv [x=320,y=240]
```

indicating (in this example) that the image size is 320x240 in both directions. Although the offerer preferred 272x224 for the receive direction, the answerer might not be able to offer 272x224 or not allow encoding and decoding of video of different image sizes simultaneously. The answerer sets new sprop-parameter-sets, constructed for both send and receive directions at the restricted conditions and image size of 320x240.

Note also that, because the payload type number is changed by the answerer, the image attribute is also split in two parts according to the recommendation in [Section 3.2.2](#).



#### 4.2.4. Example 4

This example illustrates in more detail how compensation for different sample aspect ratios can be negotiated with the image attribute.

We set up a session between Alice and Bob; Alice is the offerer of the session. The offer (from Alice) contains the image attribute below:

```
a=imageattr:97 \  
  send [x=400:16:800],y=[320:16:640],sar=[1.0-1.3],par=[1.2-1.3]] \  
  recv [x=800,y=600,sar=1.1]
```

First we consider the recv direction: The offerer (Alice) explicitly states that she wishes to receive the screen resolution 800x600. However, she also indicates that the screen on her display does not use square pixels; the sar value=1.1 means that Bob must (preferably) compensate for this.

So, if Bob's video camera produces square pixels, and if Bob wishes to satisfy Alice's sar requirement, the image processing algorithm must rescale a 880x600 pixel image ( $880=800*1.1$ ) to 800x600 pixels (could be done other ways).

... and now the send direction: Alice indicates that she can (in the image processing algorithms) rescale the image for sample aspect ratios in the range 1.0 to 1.3. She can also provide a number of different image sizes (in pixels) ranging from 400x320 to 800x640. Bob inspects the offered sar and image sizes and responds with the modified image attribute.

```
a=imageattr:97 \  
  recv [x=464,y=384,sar=1.15] \  
  send [x=800,y=600,sar=1.1]
```

Alice will (in order to satisfy Bob's request) need to rescale the image from her video camera from 534x384 ( $534=464*1.15$ ) to 464x384.

## 5. IANA Considerations

Following the guidelines in [RFC4566], the IANA is requested to register one new SDP attribute:

Attribute name:      imageattr

Long form name:      Image attribute



Type of attribute: Media-level

Subject to charset: No

Purpose: This attribute defines the ability to negotiate various image attributes such as image sizes. The attribute contains a number of parameters which can be modified in an offer/answer exchange.

Appropriate values: See [Section 3.1.1 of RFC 6236](#)

Contact name: Authors of [RFC 6236](#)

## 6. Security Considerations

The image attribute and especially the parameters that denote the image size can take on values that may cause memory or CPU exhaustion problems. This may happen either as a consequence of a mistake by the sender of the SDP or as a result of an attack issued by a malicious SDP sender. This issue is similar to the case where the a=fmtp line(s) may take on extreme values for the same reasons outlined above.

A receiver of the SDP containing the image attribute MUST ensure that the parameters have values that are reasonable and that the device can handle the implications in terms of memory and CPU usage. Failure to do a sanity check on the parameters may result in memory or CPU exhaustion.

In principle, for some SDPs containing the image attribute and for some deployments, it could be the case that simply checking the parameters is not sufficient to detect all potential Denial-of-Service (DoS) problems. Implementers ought to consider whether there are any potential DoS attacks that would not be detected by simply checking parameters.

## 7. Acknowledgements

The authors would like to thank the people who have contributed with objections and suggestions to this document and provided valuable guidance in the amazing video-coding world. Special thanks to Clinton Priddle, Roni Even, Randell Jesup, and Dan Wing. Thanks also to Robert Sparks and Paul Kyzivat for the help with the last fixes to get the attribute to work well with the offer/answer model.





## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [RFC5583] Schierl, T. and S. Wenger, "Signaling Media Decoding Dependency in the Session Description Protocol (SDP)", [RFC 5583](#), July 2009.
- [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", [RFC 5888](#), June 2010.

### 8.2. Informative References

- [H.263] ITU-T, ITU-T Recommendation H.263 (2005): "Video coding for low bit rate communication".
- [H.264] ITU-T, ITU-T Recommendation H.264: "Advanced video coding for generic audiovisual services", <<http://www.itu.int/rec/T-REC-H.264-200711-S/en>>.
- [MPEG-4] ISO/IEC, ISO/IEC 14496-2:2004: "Information technology - Coding of audio-visual objects - Part 2: Visual".
- [RFC3016] Kikuchi, Y., Nomura, T., Fukunaga, S., Matsui, Y., and H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams", [RFC 3016](#), November 2000.
- [RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", [RFC 3311](#), October 2002.
- [RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", [RFC 3515](#), April 2003.



- [RFC4629] Ott, H., Bormann, C., Sullivan, G., Wenger, S., and R. Even, "RTP Payload Format for ITU-T Rec", [RFC 4629](#), January 2007.
- [RFC5939] Andreassen, F., "Session Description Protocol (SDP) Capability Negotiation", [RFC 5939](#), September 2010.
- [RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", [RFC 6184](#), May 2011.
- [SDPMediaCapNeg] Gilman, R., Even, R., and F. Andreassen, "SDP Media Capabilities Negotiation", Work in Progress, February 2011.

#### Authors' Addresses

Ingemar Johansson  
Ericsson AB  
Laboratoriegatan 11  
SE-971 28 Luleå  
SWEDEN

Phone: +46 73 0783289  
EMail: [ingemar.s.johansson@ericsson.com](mailto:ingemar.s.johansson@ericsson.com)

Kyunghun Jung  
Samsung Electronics Co., Ltd.  
Dong Suwon P.O. Box 105  
416, Maetan-3Dong, Yeongtong-gu  
Suwon-city, Gyeonggi-do  
Korea 442-600

Phone: +82 10 9909 4743  
EMail: [kyunghun.jung@samsung.com](mailto:kyunghun.jung@samsung.com)

