**Multisource and Simulcast Negotiation for Scalable Conferencing**

**Version 9**

Cisco confidential – do not distribute outside the company

Modification History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Originator | Comments |
| 1 | Jan 25, 2013 | Robert Hansen | Initial Draft for review by attendees of the multistream workshop |
| 2 | Mar 10, 2013 | Robert Hansen | Draft revised based on feedback from initial review, with expanded examples and implementation notes separated from the specification |
| 3 | May 7, 2013 | Robert Hansen | Revision to RTCP message structures based on feedback and on decisions made in other areas, plus clarifications of other text. |
| 4 | June 14, 2013 | Robert Hansen | ACK for unsolicited SCA RTCP messages added. SDP restrictions on attribute use when CSI is set. Other text clarifications. |
| 5 | October 14, 2013 | Robert Hansen | SCR profile/level removed and id added. SDP syntax revised. New error code for provider temporarily unavailable. Other text clarifications. |
| 6 | January 6, 2014 | Robert Hansen | Initial ‘MSTR’ cookie added to the start of all RTCP feedback messages. Clarification of the id in CSIs. |
| 7 | June 23, 2014 | Robert Hansen | Mandatory SVC flavor changed from ‘NI-TC’ to ‘NI-T’. ‘source-count’ renamed ‘count’, auto-enumerates source IDs. Added duplication flag to active speaker policy. Other text clarifications. |
| 8 | July 9, 2014 | Robert Hansen  Nathan Buckles | Added CSI and flag to signal presence/absence of CSI to SCA invalid channel field. Added new error conditions for audio avatar and video placeholder. Renumbered error conditions. Support for H264 AVC added. |
| 9 | June 24, 2016 | Nathan Buckles | Added version 2 of the SCx messages. Added cvo, landscape rotation, and screen id fields to SCR. Added maximum references to the SCR H264 payload specific section. Added preferred bitrate to SCR. |

**Intro**

This document defines new methods allowing devices to advertise their support for H264 SVC, the range of sources they have available, and the temporal and spatial scalability available for those sources. It also defines new application-specific RTCP messages for subscribing to those sources, and to advertise temporary source availability on a more granular level. Implementer notes call out certain things initial Cisco implementations should do or may wish to do. Examples at the end indicate how the SDP can be used to fulfil a number of use-cases.

**Capture Source ID (CSI)**

A Capture Source ID (CSI) is used to uniquely identify a physical media capture source such as a camera or microphones within an RTP session. Transcoders or media switches should preserve the CSI of the original source, not create their own; in the case of a steam of media mixed/composed from multiple original streams it should be tagged with the CSIs of all of the contributors.

CSIs are conveyed in the CSRC field of RTP. Unlike a standard CSRC, however, they are not simply the value of the originating CSRC: they are not linked to SSRC and hence can remain constant, and all RTP streams that originate from the same source (representing different temporal, spatial or quality layers) share the same CSI. CSIs are determined by the sender of the static source, are 32 bits long and have the following format:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Scene ID | | | | | | | | | | | | | | | | | | | | | | | | Reserved | | | | | | | AV |

**Scene ID (24 bits):** This is a random value chosen when the source is first sent and unchanging after that.

**Reserved (7 bits):** These bits must be set to 0.

**AV (1 bit):** This must be set to 0 for audio sources, and 1 for video sources.

Capture sources that provide related views of the same scene (such as an endpoint’s camera and microphone) must share the same Scene ID section of their CSI. Capture sources the do not provide related views of the same scene (such as an endpoint’s camera and content source) must not shared the same Scene ID section of their CSI.

Devices that have static sources with CSIs must signal them as part of their sender source advertisement: see the next section on multi-source advertisement for this syntax. CSIs should be included in all media packets generated.

A device may change one or more of its CSIs at any time; when doing so it must send a signalling update to convey the new CSI information. A device that detects a CSI clash, where a CSI it is sending is also being sent by another entity (which it may do either by receiving an RTP packet with that CSI, or by seeing the same CSI used by another participant in the roster list or some other signalling), must immediately change that CSI to a new random value, and signal that it is doing so.

For CSIs to be more than minimally useful in a multipoint call their association with their source must be conveyed to the other devices in the conference; doing this is the responsibility of the infrastructure device or devices. The means for doing this is outside the scope of this document: a roster list is one obvious location.

CSIs should follow the normal rules for CSRCs when it comes to mixing and transcoding – transcoders should preserve the originating CSI, mixers combining multiple streams with CSIs should include all CSIs in the RTP packet (with the loudest speaker first, in the case of mixed audio).

**Implementers note**

Endpoints should only change CSIs due to a clash, and should otherwise keep them constant.

Infrastructure products should include CSIs in roster lists so that endpoints can immediately identify the originator when a new source is received.

Endpoints should include CSIs for all media sent as the response to an SCR. Infrastructure products operating in ‘hybrid’ mode by transcoding or otherwise manipulating non-multistream sources such that they can be switched to a multistream endpoint in response to an SCR should create CSIs for each of these endpoints and fill them in on the streams it is sending.

While as with SSRCs endpoints should monitor incoming CSRCs for CSI clashes, in practise endpoints will be able to more easily detect clashes via the roster list, as with this no media packet needs to be received before a collision can be detected.

SSRC and CSI should be chosen independently – this allows easy differentiation of self-view media and a real collision without needing to wait for RTCP SDES packets to be received (if both SSRC and CSI match that of stream the device is sending then the chance both have collided at the same time is essentially zero).

**H264 SVC SDP negotiation**

Support for H264 SVC should be indicated by including a codec with payload type *H264-SVC* with a clock rate of 90000:

* **a=rtpmap:98 H264-SVC/90000**

The codec has all the same optional media parameters as standard H264 defined in [RFC6184]. These parameters specify the receive limits of the device *per stream*. There are also a number of additional parameters that can be included:

The optional parameter *mst-mode* is as defined in [RFC6190], and signals support for multi-stream transmission of temporal layers. Cisco endpoints must support NI-T mode.

* **...;mst-mode=NI-T;...** (MST mode NI-T supported)

The optional parameter *uc-mode* defines the level of UCConfig mode support, as per the [Unified Communications Specification for H264 AVC and SVC UCConfig Modes]. Cisco endpoints must support mode 1 (temporal scalability).

* **...;uc-mode=1;...** (UCConfig mode 1 supported)

**Implementers note**

To ensure interoperability all implementations supporting H264-SVC should advertise support for receiving streams with baseline profile, MST mode NI-T, packetization mode 1 and UCConfig mode 1. They may also include other variants of H264-SVC if they wish.

Scalability of calls involving B2BUAs or conferencing bridges may suffer if the middle boxes need to rewrite dynamic payload types between participants. A CITG-wide initiative to agree on default dynamic payloads exists, and publishes the document ‘*Default Dynamic Payload Numbers Used by UC Endpoints and 3rd Parties’* to define these defaults, which includes an allocation for H264-SVC. As such, all implementations should use a dynamic payload type of **98** for H264-SVC (baseline profile, MST mode NI-T, packetization mode 1 and UCConfig mode 1) in their initial offer.

**Multi-source SDP advertisement**

With scalable conferencing, devices are able to send media from multiple, independent sources, either from static, physical sources they possess or by switching in streams they have received from other sources. These independent sources are signalled using a new media attribute, defined below. Devices may also be able to simulcast media from such a device with different spatial encodings or using different compression algorithms; these are *not* advertised as independent sources - see the next section on source encoding for how these are advertised.

A source can be signalled in SDP via a new media attribute *sprop-source*. This attribute has the following format:

* **a=sprop-source:<id> <optional parameters>**

The id of the source is used to identify stream capabilities (see the next section on source encoding) and must be an 8-bit value not be used by any other *sprop-source* attribute in this m-line.

The optional parameter *csi* signals that this is a static source, and provides the CSI of this source in decimal form. If not present there is no static CSI for this source.

* **...;csi=4049318657;...** (CSI of this source is 0xF15BB301)

The optional parameter *count* signals that multiple independent sources are available with these parameters. The value of *count* must be a positive integer. If not present, the value defaults to 1. If greater than 1, it indicates that N sources with identical parameters are available from this sender, and is equivalent to N *a=sprop-source* lines with the same parameters and the same source encodings (see next section). The ids of these additional sources are equal to the base *sprop-*source id + n, where n is the zero-indexed index of the source. For example, an *sprop-source* with an id of 5 and a count of 3 represents three identical independent sources with ids 5, 6 and 7. If *csi* is specified the value of this parameter cannot be greater than 1.

* **...;count=3;...** (three sources with these parameters are available)

The optional parameter *policies* signals the policies that the sender supports for this source and that can be included in the SCR. Policies define the manner in which the sender will decide between different video captures available to send to the recipient. A supported policy is listed, followed by a colon, followed by the id of that policy (to be used in an SCR – see later in the document). Each id must be a unique value between 1 and 65535. If multiple policies are supported they should be included as a comma-separated list. Currently defined policies are *as* (active-speaker) and *rs* (receiver-selected); new policies may be defined in later documents. If not present there are no policies available for this source. If *csi* is specified no policies may be defined for this source.

* **...;policies=as:1,rs:2;...** (this source supports active-speaker switching and receiver selection)

The optional parameter *simul* signals the simulcast encodings (see next section) of this source that can be sent simultaneously. If not present **all** simulcast encodings can be sent simultaneously. The value of the parameter is a list of alternate sets of simulcast encodings that can be sent, specified by *simulcast id* (see next section). The alternate encoding sets are separated by a vertical bar, while the simultaneous encodings within each set are comma-separated.

* **...;simul=1,2|1,3|1,4;...** (this source can be simulcast in encoding 1 and *either* 2, 3 or 4)

The optional parameter *lrotation* signals support for requesting a landscape display aspect ratio (e.g. 4:3 or 16:9) of this source. If not present then the source cannot be requested in a landscape aspect ratio. The value of the parameter is either a 0 (not supported) or a 1 (supported).

* **...;lrotation=1;… (this source supports a landscape rotation request)**

The optional parameter *cvo* (coordination of video orientation, clause 6.2.3 of <http://www.3gpp.org/DynaReport/26114.htm>) signals support for the cvo field of the SCR message. If not present then the cvo field is not supported. The value of the parameter is either a 0 (not supported) or a 1 (supported).

* **...;cvo=1;… (this source supports a cvo request)**

The optional parameter *mavatar* (mute with avatar) signals support for requesting an encoding of a video avatar in place of a live video feed should the live video feed become unavailable (e.g. the user mutes their video). If not present then the source cannot generate encoded video based on an avatar. The value of the parameter is either a 0 (not supported) or a 1 (supported)

* **...;mavatar=1;… (this source supports a mute with avatar request)**

**Implementers note**

Every *sprop-source* attribute should have at least one corresponding *sprop-simul* attribute (else the attribute is meaningless).

Endpoints should include CSIs for all sources.

Infrastructure products should always advertise support for active-speaker switching, and may choose to support receiver selection. Endpoints should not advertise any policies (unless they actually have multiple hidden sources, such as a three-camera system only sending one stream).

**Source encoding SDP advertisement**

A sender may be able to simulcast media from a single source with a variety of different spatial encodings and codecs, or they may only be able to send it with one particular encoding. Each encoding format they can support for a source (or the fact they support all advertised codecs) should be advertised using a new media attribute *sprop-simul*. This attribute has the following format:

* **a=sprop-simul:<source id> <simulcast id> <format> <format specific parameters>**

The *source id* of the attribute must match the id of an *sprop-source* attribute in this m-line. The simulcast id must be unique to *sprop-simul* attributes in this m-line and is used by the *sprop-source* attribute to identify specific encodings (see previous section).

The *format* defines a format or formats in which the sender is able to send this source. The format must be equal to a payload type advertised in this m-line, or to *\**, which allows the format of *any* payload type advertised on the m-line. Finally, there are a set of optional parameters specific to this format (if defined) that give the send limits of the encoding. Currently these are only defined for H264 and H264-SVC. If \* is used, no send limits are specified.

***H264 specific parameters***

The format-specific parameters for H264 (AVC) are the same as those defined for the receive side advertisement in RFC 6184. These parameters specify the send limits of this encoding. However profile and packetization mode must match those of the receive format if included (if not included they are implicit).

***H264-SVC specific parameters***

The format-specific parameters for H264-SVC are the same as those defined for the receive side advertisement earlier in the document. These parameters specify the send limits of this encoding. However profile, packetization mode, SST mode and UCConfig mode must match those of the receive format if included (if not included they are implicit).

The new optional parameter *fr-layers* gives the SVC temporal layers available from this source. This is a comma-separated list of the cumulative frame-rate of the temporal layers available, expressed in hundredths of a frame per second. If this parameter is not present only a single temporal layering is supported.

* **...;fr-layers=750,1500,3000;...** (this source supports 7.5/15/30fps temporal layering)

**Implementers note**

Only H264 AVC and SVC format specific parameters are defined; H265 will likely follow at a later date.

For audio endpoints implementations should generally simply advertise sources with \*, unless they have some encoder restriction with regards to certain codecs.

**Other SDP attributes**

Devices should use and understand overall call bandwidth and per session bandwidth limits using standard parameters as defined in [RFC3261]. There is no requirement for bandwidth symmetry: the receive bandwidth advertised in an SDP answer may exceed that of the offerer.

Support for the SCR, SCA and SCA-ACK application-specific feedback messages (see later sections) must be advertised in SDP following the syntax defined in [RFC4585]. Such support is advertised with the following media attributes:

* **a=rtcp-fb:\* ccm cisco-scr**
* **a=rtcp-fb:\* ccm cisco-scr-v2**

The first version of the RTCP feedback attribute advertises support for version 1 of the SCx messages. The second version advertises support for version 2 of the SCx messages. An SCx message sender, when sending an SCx, may send that SCx message using any of the versions supported by the receiver (in the receiver’s SDP). This receiver must support receiving SCx messages for all versions that it advertises.

The new media attribute *sprop-total* can be used to advertise some additional encoder- and codec-independent totals for all transmitted encodings. This attribute has the following format:

* **a=sprop-total:<optional parameters>**

The optional parameter *max-channels* defines the maximum number of sub-session channels the sender can support. The value gives the maximum number of channel requests that can be made in an SCR (see next section). If not present there is no additional constraint on the maximum number of streams that can be sent, and an SCR may contain requests up to the maximum allowed by the number of sources and allowable simulcast qualities.

* **...;max-channels=3;...** (a maximum of three sub-session channels can be requested via SCR)

The optional parameter *­max-bitrate* defines the maximum send bitrate available. The value is given as an integer in kbits per second. If not present, the max transmit bandwidth is undefined.

* **...;max-bitrate=2000;...** (the maximum send bitrate for all encodings is 2Mbits/s)

The optional parameter *max-pps* defines the maximum pixels per second available to be sent. The value codec-independent and is expressed as an integer in pixels per second. If not present the max pixels per second is undefined.

* **...;max-pps=27648000;...** (the max pixels per second, equivalent to a 720p stream at 30fps)

**Implementers note**

All implementations should signal send/receive support for Cisco multi-stream RTCP. Absence of signalling for Cisco multi-stream RTCP on a media line or its presence at inactive, sendonly or recvonly means that no multi-stream will be done on this media line. Implementations should avoid using sendonly or recvonly, since they don’t achieve anything without the corresponding receive or send capabilities.

Implementations may or may not wish to include an *sprop-total* attribute, depending on whether they wish to express further send limitations.

Implementations that receive an *sprop-total* don’t actually need to process it if they don’t want to – the limitations expressed provide guidance for the receiver to make a more optimal choice by understanding the sender’ limits (eg, not using a particular layout if it knows the streams it needs to make it up would be lower resolution than it wants).

**RTCP Sub-session Channel Request (SCR)**

The Sub-session Channel Request (SCR) feedback message is an application layer RTCP feedback message as defined in RFC4585 used by the receiver to request specific sub-session channels within the main RTP session. Each new SCR (identifiable by unique sequence number) invalidates all previous SCRs.

The SCR contains zero or more sub-session channel requests – each request corresponds to the sending of a single media stream that falls within the parameters of that request. Each of these media streams must contain an RTP extension header including the ID specified in the request (in cases where a single media stream fulfils two or more requests only a single stream need be send, tagged with multiple IDs).

As an application-level message the RTCP packet must have FMT of 15 and PT of PSFB (206) as per [RFC4585]. The format of the first eight bytes of the FCI of the RTCP packet is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Application Feedback Identifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Multistream Message Type | | | | | | | | | | | | Version | | | | Sequence Number | | | | | | | | | | | | | | | |

**Application Feedback Identifier (32 bits):** The id of application-specific feedback message. Must be set to 0x4D535452 (the hexadecimal value of ‘MSTR’ in ASCII).

**Multistream Message Type (12 bits):** The type of multisource/simulcast feedback message. Must be set to 1.

**Version (4 bits):** The version of the SCR message. Must be set to 1 or 2, based on SDP negotiation.

**Sequence Number** **(16 bits):** The sequence number of the current request. For retransmissions of an SCR this must be the same as that of the SCR being retransmitted. For new SCRs this value must be higher than that of the previous SCR sent. This value must not be set to 0. If support for SCR feedback is disabled mid-call all existing state is deprecated, including sequence numbers.

The SCR may then include any number of requests (note that an SCR with 0 requests is valid), each of which must have the following format:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Sub-session Channel ID | | | | | | | | | Source ID | | | | | | | | Length | | | | | | | | | | | | | | | |
| Maximum Bitrate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Preferred Bitrate | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Policy Type | | | | | | | | | | | | | | | | | Policy Info Length | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Sub-session Channel ID (8 bits):** The id with which the media sender should tag all RTP packets that match this request. This is conveyed in the *virtual identifier* RTP extension header (see the *‘multistream-signalling*’ document for a definition of this element).

**Source ID (8 bits):** The ID of the *sprop-source* attribute advertised by the media sender in their most recent SDP offer or answer that this request is related to.

**Length (16 bits):** Length of the request in bytes, including the 4 bytes for channel id and length.

**Maximum Bitrate (32 bits):** The maximum sub-session channel media bitrate, expressed in bits per second.

**Preferred Bitrate (32 bits):** The preferred sub-session channel media bitrate, expressed in bits per second and only present in v2 SCR packets. The preferred bitrate indicates a bitrate less than or equal to the maximum bitrate that is preferred based on current network conditions or other criteria. A sender may send a bitrate higher than the preferred bitrate if necessary, but must always send a bitrate lower than the requested maximum bitrate.

**Policy Type (16 bits):** The policy type ID for the request. If no specific policy is desired by the requester this field must be set to 0. If non-zero, the value must match a policy ID specified by the sender in their most recent SDP message for this source.

**Policy Info Length (16 bits):** The number of bytes of policy-specific information, **not including** the policy type and length fields. If the policy type is set to 0, this must be set to 0.

**Policy-Specific Info (N bits):** The length and format of this information varies depending on the specific policy. The length of the policy-specific info field must be a multiple of 32 bits. Two policies are defined below, and additional policies may be defined later:

* **Active Speaker:** Length 32 bits.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Priority | | | | | | | | ScreenGroupID | | | | | | | | D | Reserved | | | | | | | | | | | | | | |

* + **Priority (8 bits):** A value between 0 and 255. A higher value indicates a higher switching priority.
  + **ScreenGroupID (8 bits):** In version 2 of the SCR, the most significant 4 bits are the *grouping adjacency ID* and the least significant 4 bits are the *screen ID*. In version 1 of the SCR, all 8 bits are the *grouping adjacency ID* and the *screen ID* has a default value of 0. Requests with the same grouping adjacency ID will be rendered adjacent to each other with the lowest sub-session channel id most screen-left. Requests with the same screen ID will be rendered on the same physical or logical screen, influencing relationships between streams such that an individual source will either be displayed on one screen or another but not both (at the same time).
  + **Duplication Flag (1 bit):** If the duplication flag is set to ‘1’ this indicates that the sender should, where appropriate, provide duplicates of sources that are being displayed in higher-priority groups, whereas if set to ‘0’ the sender should avoid duplicating these streams.
  + **Reserved (15 bits):** The last 15 bits are reserved, should be set to 0 and ignored when received.
* **Receiver-Selected Source:** Length 32 bits. The value is equal to the CSI of the source the requester wants to receive in this channel.

Finally, the request must include one or more payload sections. If multiple payload sections are included, these are *alternatives* – a recipient receiving a request containing multiple payload sections responds with a single media stream with parameters matching one of the payload specifications (the sender can pick which they prefer to send, and may change mid-stream to a different set of codec parameters that match a different payload specification). The payload section has the following format:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Payload Type | | | | | | | | Reserved | | | | | A | C | L | Payload Info Length | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Payload Type (8 bits):** A payload type specifying a media type which the receiver is willing to receive on this sub-session channel. For a specific media payload type, the first bit must be set to 0, and the bottom seven bits set to a payload signalled in the requester’s SDP. Alternatively, the requester can set the first bit to 1 and the lower seven bits to 0 to signal *all payloads*: this specifies that the requester is willing to receive any payload type they have specified that they can receive in SDP with no additional constraints *with the exception of any payload type they have also explicitly listed in this request* (which they can receive with the constraints listed).

**Reserved (5 bits):** These bits should be set to 0 and ignored when received.

**A (1 bit):** When set this bit requests that the sender send an encoded avatar (image) if the live video stream is not available. When cleared the sender should not send an encoded avatar if the live video is not available (instead the sender should send an SCA warning). This field should be set to 0 when the source being requested has not indicated support for avatar video in the sprop-source attribute.

**C (1 bit):** When set this bit requests that the sender include CVO (coordination of video orientation) information in the transmitted stream. When cleared the sender should not include CVO information. This field should be set to 0 when the source being requested has not indicated support for CVO in the sprop-source attribute.

**L (1 bit):** When set this bit requests that the sender always encode the media source in a landscape display aspect ration (e.g. 4:3 or 16:9), padding as required. When the L bit is set, the value of the C bit can be ignored and treated as if it was 0. When cleared the sender may send the native aspect ratio of the encoded source. This field should be set to 0 when the source being requested has not indicated support for landscape rotation in the sprop-source attribute.

**Payload Info Length (16 bits):** The number of bytes of payload-specific information, **not including** the policy type and length fields. If the payload type is set to all payloads, this MUST be set to 0. For any specific payload, this must either be set to 0 (which signals that no additional constraints exist beyond those listed in SDP) or to the length appropriate for the codec that payload refers to. The length of the payload-specific info field must be a multiple of 32 bits.

**Payload-Specific Info (N bits):** The length and format of this information varies depending on the specific codec. The length of the payload-specific info field must be a multiple of 32 bits. The constraints for H.264 and H264-SVC are defined below (they share the same set of constraints), while constraints for additional payloads may be defined later.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Max MBPS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Max FS | | | | | | | | | | | | | | | | | Max FPS | | | | | | | | | | | | | | | | |
| Temporal Layers | | | | | | | | | ...Temporal Layer Blocks… | | | | | | | | | | | | | | | | | | | | | | | | |
| TLV Type | | | | | | | | | TLV Length | | | | | | | | | TLV Value… | | | | | | | | | | | | | | | |
| TLV Value… | | | | | | | | | Padding | | | | | | | | | | | | | | | | | | | | | | | | |

**Max MBPS (32 bits):** A limit on the maximum number of receivable macroblocks per second, in units of macroblocks per second.

**Max FS (16 bits):** A limit on the maximum receivable framesize, in units of macroblocks.

**Max FPS (16 bits):** A limit on the maximum receivable frame-rate, in units of hundredths of a frame per second.

**Temporal Layers (8 bits):** The number of specific temporal layers requested. If 0, the sender of the SCR is willing to receive the media for this sub-session channel in any temporal layering within the receive parameters defined in their SDP. For H264 AVC this should be set to 0.

This is followed by a number of 16-bit **Temporal Layer** blocks equal to the number of temporal layers defined; each block gives the frame-rate of the temporal layer requested in hundreds of a frame per seconds.

**TLV Section (Variable Length):** In version 1 of the SCR no TLVs are allowed and this section must be left out. In version 2 of the SCR zero or more TLVs are allowed. A receiver must ignore TLV with type fields that it does not understand. TLVs have the following format.

**TLV Type (8 bits):** The type of the TLV, valid types are:

**Reserved (0):** Not allowed, reserved for padding.

**Maximum Number of Reference Frames (1):** The maximum number of allowed reference frames for this H264 stream. The length of this TLV must be 1 and the value is an unsigned 8 bit integer. If not present, the value defaults to 1. If present and the value is set to 0, then the value defaults to 1.

**TLV Length (8 bits):** The length of the TLV in bytes not including the type and length fields. A value of 0 is valid and indicates that there is no value field for this TLV.

**TLV Value (Variable Length):** The value of the TLV, interpretation is defined by the type and length of the TLV.

Finally, the H.264-specific info must be padded with 0s to be 32-bit aligned.

**Implementers note**

Implementations may send single-stream media per m-line before receiving an SCR and as per [RFC3264] must at least have their receive ports open to receive it. Applications that do not wish to send cut-through media are free not to do so, and those that prefer not to receive it should simply not decode or not render the received cut-through media.

No matter the payload-specific parameters of an SCR, implementations should not send media exceeding the receive parameters of any given stream defined in SDP.

If SCR feedback is disabled mid-call and then later re-enabled, the receiver must cope with receiving a new initial sequence number.

Implementations are free to pick the RTCP encapsulation and SSRC with which to associate the SCR message so long as it follows the RTCP specifications.

Receivers that support multiple types of application-layer feedback should use the initial ‘MSTR’ value to differentiate multi-stream control messages from other application-layer feedback messages (such as MARI). The message type field can then be used to differentiate the SCR from other multi-stream control messages.

Max MBPS, Max FS and Max FPS are additional constraints which the sender should not exceed; as with all H.264 limitations, the sender may send *less* than the values requested. These are simple limitations (not adjustments from a level), so they should not be set to 0 unless the implementation is actually requesting that no video is sent.

More information on the Max FPS parameter can be obtained from the analogous ‘max-fps’ parameter defined in draft-kristensen-avt-rtp-h241param.

**RTCP Sub-session Channel Announce (SCA)**

The Sub-session Channel Announce (SCA) feedback message is an application-level RTCP feedback message used by the sender to announce the number of sendable, independent sources they have for this session, along with identifying any currently configured sub-session channel requests that they are unable to fulfil.

This message must be sent in response to the receipt of an SCR with a sequence number equal to or higher than the most recent SCR seen. If the sequence number is higher than the highest previous seen then the SCR should be processed and an SCA with a new sequence number sent in response. If the sequence number indicates an SCR retransmission an SCA message should be sent either with a new sequence number or as a retransmission.

An SCA may also be sent independently by a sender to notify the receiver of a change of some kind.

The recipient of an SCA must listen for media from each sub-session channel request that was successful or that received a ‘*Stream Temporarily Unavailable’* response in the most recent SCA received. The recipient may stop listening for streams that received any other error response.

If no SCA has been sent the number of sub-session channels available and max adjacent sources are undefined.

As an application-level message the RTCP packet must have FMT of 15 and PT of PSFB (206) as per [RFC4585]. The format of the FCI of the RTCP packet is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Application Feedback Identifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Multistream Message Type | | | | | | | | | | | | Version | | | | Sequence Number | | | | | | | | | | | | | | | |
| Current Request | | | | | | | | | | | | | | | | Sources Available | | | | | | | | Max Adj. | | | | Reserv. | | | A. |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Application Feedback Identifier (32 bits):** The id of application-specific feedback message. Must be set to 0x4D535452 (the hexadecimal value of ‘MSTR’ in ASCII).

**Multistream Message Type (12 bits):** The type of application-specific feedback message. Must be set to 2.

**Version (4 bits):** The version of the SCA message. Must be set to either 1 or 2 based on SDP version negotiation.

**Sequence Number** **(16 bits):** The sequence number of the current announcement. For retransmissions of an SCA this must be the same as that of the SCA being retransmitted. For new SCAs this value must be higher than that of the previous SCA sent. This value must not be set to 0. If support for SCR feedback is disabled mid-call all existing state is deprecated, including sequence numbers.

**Current Request (16 bits):** The sequence number of the most recent SCR processed. If no SCR has been received, this value must be set to 0.

**Sources Available (8 bits):** The maximum number of independent sources that can currently be supplied for this session. This value should be less than or equal to the number of independent sources advertised via sprop-source lines in the SCA sender’s SDP.

**Max Adjacent Sources (4 bits):** This gives the maximum number of spatially adjacent streams from any single source that will be sent during the call (eg., 3 for a multipoint conference including at least one three camera system). If the sender does not support spatial adjacency in sources on this session this value should be set to 0.

**Reserved (3 bits):** These bits should be set to 0 and ignored when received.

**ACK Required flag (1 bit):** If set to 1, this indicates that the recipient should send an SCA-ACK packet when it receives this packet. If set to 0 no SCA-ACK packet should be sent in response to this.

Finally, there are 0 or more *invalid channel warnings*, each of which is 32-64 bits with the following format:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | | 4 | | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Invalid Channel ID | | | | | | | | Reserved | | | | | | | | | | | | | | | C | | Error Code | | | | | | | | |
| Capture Source ID | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Invalid Channel ID (8 bits):** The ID of the invalid channel requested.

**Reserved (15 bits):** These bits should be set to 0 and ignored when received.

**C Flag (1 bit):** If set to 1 then the Capture Source ID field is present, otherwise the Capture Source ID field is not present.

**Error Code (8 bits):** This provides a reason that the channel is invalid. The error codes currently defined are:

* **0x0:** Undefined Error
* **0x20:** Invalid Request
* **0x31:** Stream Temporarily Unavailable: Insufficient sources
* **0x32:** Stream Temporarily Unavailable: Current source has no media of this format
* **0x40:** Stream Source Change Avatar: The normal source for this stream has been replaced with an encoding of an avatar.

**Capture Source ID (32 bit):** The CSI of the source that is currently unavailable.

**Implementers note**

Max Adjacent Sources allows endpoints to optimise the layout they wish to use – for instance, if a 3-screen system is present in the conference, endpoint implementations may wish to avoid requesting a 2x2 layout, for instance.

Implementations should record the highest SCR sequence number processed so far. SCRs with a higher sequence number should be processed, an SCA sent and the highest sequence number processed so far updated. SCRs with the same sequence number as the highest processed should not be processed, but an SCA should be sent. SCRs with a lower sequence number than the highest processed should not be processed and no SCA should be sent.

Implementations should use receipt of an SCA as acknowledgement that an SCR has been processed, as this may well be simpler than monitoring the media for appropriate changes – if an SCA is not received with a certain period the implementation should retransmit the SCR.

Implementations must not assume that they will receive an SCA response to an SCR before the sender changes their media streams – implementations may send an SCA before they change the media streams they are sending, or may send it after. Similarly, implementations must not assume that they will receive an unsolicited SCA before the sender changes their media streams; a media receiver must be prepared for any requested stream to be resumed at any time (particularly true of those marked Temporarily Unavailable).

If SCA feedback is disabled mid-call and then later re-enabled, the receiver must cope with receiving a new initial sequence number.

Implementations sending an unsolicited SCA (not in response to an SCR) should set the ‘ACK Required’ flag to 1 and use the receipt of an SCA-ACK to confirm reception at the far end (otherwise the SCA should be resent).

Implementations are not required to send an unsolicited SCA at the start of a call. If an implementation requires receiving an SCA before it can send a meaningful SCR (eg, because it wishes to know the number of channels available or the max adjacency) it should solicit an SCA by sending an empty SCR.

Implementations are free to pick the RTCP encapsulation and SSRC with which to associate the SCA message so long as it follows the RTCP specifications.

Receivers that support multiple types of application-layer feedback should use the initial ‘MSTR’ value to differentiate multi-stream control messages from other application-layer feedback messages (such as MARI). The message type field can then be used to differentiate the SCA from other multi-stream control messages.

Receivers should be prepared to receive SCAs in rapid succession to provide CSIs on video streams to provide the CSI of audio-only participants. This allows audio avatar placeholders to be rendered with the correct association to roster list information.

Error 0x20 should be used when an SCR entry refers to an invalid source id or is otherwise invalid.

Error 0x31 should be used when the number of requests in an incoming SCR is within the limits of what is advertised in the SDP, but not enough sources are currently available.

Error 0x32 should be used when enough sources are available, but media for a particular source can’t be sent in the format requested in the incoming SCR. Upon receiving this error code, a receiver can choose to render an alternative, such as an audio avatar.

Error 0x40 should be used when the media sender has changed the encoding source based on user input or some other event.

Error code 0x0 should only be used when no other error code is available to describe the reason a particular request can’t be fulfilled.

Each error code applies to the referenced SCR entry only. The SCA should contain a separate invalid channel warning for every SCR entry that is invalid or can otherwise not be fulfilled, with the ‘Invalid Channel ID’ field having the same value as the SCI in the SCR entry.

All error codes except 0x40 indicate that no media (RTP) will be transmitted on the referenced SCR entry. It is possible (due to race condition, packet loss, etc) that RTP may arrive for a short period of time before or after an SCA warning is added or removed. It is recommended that an implementation consume this media, decode it as applicable, but not render that media to the end user.

**RTCP Sub-session Channel Announce Acknowledgement (SCA-ACK)**

The Sub-session Channel Announce Acknowledgement (SCA-ACK) feedback message is an application-level RTCP feedback message used by the sender to acknowledge receipt of an SCA message.

As an application-level message the RTCP packet must have FMT of 15 and PT of PSFB (206) as per [RFC4585]. The format of the FCI of the RTCP packet is as follows:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 | 2 | |
| Application Feedback Identifier | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Multistream Message Type | | | | | | | | | | | | Version | | | | SCA Sequence Number | | | | | | | | | | | | | | | |

**Application Feedback Identifier (32 bits):** The id of application-specific feedback message. Must be set to 0x4D535452 (the hexadecimal value of ‘MSTR’ in ASCII).

**Multistream Message Type (12 bits):** The type of application-specific feedback message. Must be set to 3.

**Version (4 bits):** The version of the SCA-ACK message. Must be set to 1 or 2 based on SDP negotiation.

**Sequence Number** **(16 bits):** The sequence number of the SCA this message is acknowledging.

**Implementers note**

Implementations receiving an SCA with the ‘ACK required’ flag set should send an SCA-ACK packet with a corresponding SCA sequence number. Implementations can send this packet even if the SCA received has a lower sequence number than the previous SCA received (obviating a need to maintain state).

Implementations are free to pick the RTCP encapsulation and SSRC with which to associate the SCA-ACK message so long as it follows the RTCP specifications.

Receivers that support multiple types of application-layer feedback should use the initial ‘MSTR’ value to differentiate multi-stream control messages from other application-layer feedback messages (such as MARI). The message type field can then be used to differentiate the SCA-ACK from other multi-stream control messages.

**SDP Examples**

A set of examples follow, giving the SDPs of a range of hypothetical devices to illustrate how the SDP syntax works, and how it can be used to solve a wide range of use cases relevant to Cisco. To save space incomplete SDP are used, with only the relevant main audio and video portions. Lines that represent new SDP syntax defined in this document are bolded.

**Endpoint example (simulcast, local composition)**

m=audio 4298 RTP/AVP 113 108 0 15

a=rtpmap:113 MP4A-LATM/90000

a=fmtp:113 profile-level-id=24;object=23;bitrate=96000

a=rtpmap:108 MP4A-LATM/90000

a=fmtp:108 profile-level-id=24;object=23;bitrate=64000

a=rtpmap:0 PCMU/8000/1

a=rtpmap:15 G728/8000

**a=sprop-source:1 csi=15847168**

**a=sprop-simul:1 100 \***

**a=sprop-simul:1 101 \***

**a=rtcp-fb:\* ccm cisco-scr**

a=sendrecv

m=video 4300 RTP/AVP 126 98

b=AS:4000

a=rtpmap:126 H264/90000

a=fmtp:126 profile-level-id=42e016;max-mbps=244800;max-fs=8160;packetization-mode=1

**a=rtpmap:98 H264-SVC/90000**

**a=fmtp:98 profile-level-id=42e016;max-mbps=244800;max-fs=8160;packetization-mode=1;mst-mode=NI-T;uc-mode=1**

**a=sprop-source:1 csi=15847169;simul=100|101,102**

**a=sprop-simul:1 100 98 profile-level-id=42e016;max-mbps=244800;max-fs=8160;fr-layers=750,1500,3000**

**a=sprop-simul:1 101 98 profile-level-id=42e016;max-mbps=108000;max-fs=3600;fr-layers=750,1500,3000**

**a=sprop-simul:1 102 98 profile-level-id=42e016;max-mbps=108000;max-fs=3600;fr-layers=750,1500,3000**

**a=rtcp-fb:\* ccm cisco-scr**

a=sendrecv

An SDP snippet from a hypothetical endpoint with a single microphone and single camera. In audio the endpoint supports a range of audio codecs and can simulcast the audio twice using any two formats the receiver wants (allowing a switch to forward AAC to some endpoints and G728 to others, for instance). It can also simulcast the video twice, if desired, but here it has further limitations: while it is able to encode a 1080p@30fps stream, it is more limited if required to encode two streams. In this case it is only able to support up to 720p@30fps. Each stream is available with three temporal layerings, at 7.5fps, 15fps and 30fps. The CSIs of the audio and video are specified. Support for SCR and SCA in RTCP is signalled.

**Conferencing device example (switching of multiple, independent audio and video streams)**

m=audio 6014 RTP/AVP 113 15

a=rtpmap:113 MP4A-LATM/90000

a=fmtp:113 profile-level-id=24;object=23;bitrate=96000

a=rtpmap:15 G728/8000

**a=sprop-source:1 source-count=3;policies=as:10**

**a=sprop-simul:1 1 \***

**a=rtcp-fb:\* ccm cisco-scr**

a=sendrecv

m=video 6016 RTP/AVP 126 98

b=AS:8000

a=rtpmap:126 H264/90000

a=fmtp:126 profile-level-id=42e016;max-mbps=244800;max-fs=8160;packetization-mode=1

**a=rtpmap:98 H264-SVC/90000**

**a=fmtp:98 profile-level-id=42e016;max-mbps=244800;max-fs=8160;packetization-mode=1;mst-mode=NI-T;uc-mode=1**

**a=sprop-source:1 policies=as:10,rs:11**

**a=sprop-simul:1 1 98 profile-level-id=42e016;max-mbps=244800;max-fs=8160;fr-layers=1500,3000**

**a=sprop-source:2 source-count=5; policies=as:10,rs:11**

**a=sprop-simul:2 2 98 profile-level-id=42e016;max-mbps=108000;max-fs=3600;fr-layers=1500,3000**

**a=rtcp-fb:\* ccm cisco-scr**

a=sendrecv

An SDP snippet from a hypothetical switching MCU. In audio the MCU can supply up to three streams, encoded in either AAC or G728. In video the device can supply up to six streams, one at up to 1080p@30fps and five at up to 720p@30fps. As the streams are switched, not static, sources no CSIs are specified. The MCU supports active speaker switching and receiver source selection for video, but only supports active speaker switching in audio. Support for SCR and SCA in RTCP is signalled.

**H265-capable endpoint example (simulcast of H264 and H265)**

m=video 4300 RTP/AVP 126 98 104

b=AS:4000

a=rtpmap:126 H264/90000

a=fmtp:126 profile-level-id=42e016;max-mbps=244800;max-fs=8160;packetization-mode=1

**a=rtpmap:98 H264-SVC/90000**

**a=fmtp:98 profile-level-id=42e016;max-mbps=244800;max-fs=8160;packetization-mode=1;mst-mode=NI-T;uc-mode=1**

**a=rtpmap:104 H265/90000**

**a=fmtp:104 example-caps=1080p30;packetization-mode=1**

**a=sprop-source:1 csi=15847169;simul=100|101,102**

**a=sprop-simul:1 1 98 profile-level-id=42e016;max-mbps=244800;max-fs=8160;fr-layers=750,1500,3000**

**a=sprop-simul:1 2 104 example-caps=1080p30**

**a=sprop-total:max-pps=62668800**

**a=rtcp-fb:\* ccm cisco-scr**

a=sendrecv

An SDP snippet from an endpoint capable of encoding H265. Only the video portion is shown. The endpoint can simulcast two streams, one in H264-SVC and one in H265. Each stream can be sent at up to 1080p30, but if both are requested the endpoint will be unable to supply them at both at full resolution, as it has advertised a max pixels per second equivalent to a single 1080p30 stream.