Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2013

M. Westerlund
B. Burman
L. Hamm
Ericsson

October 22, 2012

Codec Operation Point RTCP Extension
draft-westerlund-avtext-codec-operation-point-01

Abstract

The Audio-visual Profile with Feedback (AVPF) specification defines a framework and messages for fast feedback and media control over RTCP. The Codec Control Messages (CCM) specification defines an extension to AVPF, by specifying additional messages for codec control and feedback. This specification extends CCM, by specifying messages that let participants dynamically communicate a set of codec configuration parameters, which enables better optimization of resource efficiency and quality of media transmission.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Multimedia real-time communication services, such as video telephony and videoconferencing, use the real-time transport (RTP/RTCP) [RFC3550] protocol to transmit media streams, such as audio and video. A session establishment protocol, such as SIP [RFC3261], in combination with a capability negotiation protocol, such as SDP offer/answer [RFC3264] is normally used to establish the session and negotiate media capabilities. In some cases, a set of codec parameters is negotiated that does not express any specific limit or capability, but just describes a certain codec configuration.

During session establishment, the participating endpoints normally have limited knowledge about the session environment, e.g. whether the session will be point-to-point or contain some multiparty scenario, how users will interact with the application, how network conditions will vary during the session, etc. To take those variations into account, the participants can renegotiate session parameters to better suit the communication environment. At times, when variations or changes are frequent in nature, it will require the needed reaction time to be short, which may make repeated session renegotiation inefficient and/or too slow. In addition, variations may not even affect negotiated session parameters, if the variations occur within the negotiated boundaries.

The above scenario can become critical especially in cases where a given media stream is transmitted towards, and received by, multiple receivers. In multiparty environments, scalable encoding or simulcast can be used to make the system more efficient and provide better quality to participants that are capable of receiving and utilizing the higher quality. These use cases result in that a sending party is requested to deliver multiple encoder operation points.

The Audio-Visual Profile with Feedback (AVPF) specification [RFC4585] defines a framework and messages for fast feedback and media control over RTCP. The Codec Control Messages (CCM) specification [RFC5104] defines an extension to AVPF, by specifying additional messages for codec control and feedback. This specification extends CCM, by specifying messages that let participants dynamically communicate a set of codec configuration parameters, which enable better optimization of resource usage and quality of media transmission.

The codec configuration parameters specified in this document focus on some basic audio and video properties, such as video resolution, video frame rate, media stream bit-rate, audio sampling rate, number of audio channels, maximum RTP packet size and rate. Additional parameters can be standardized in the future.

The codec control messages are not meant to replace the configuration performed using e.g. SDP. Instead, the messages can be used to communicate dynamic and frequent changes that take place within boundaries that have been negotiated as part of the session establishment.

2. Definitions

2.1. Terminology

The following terms and abbreviations are used in this document:

Bandwidth: The network resource needed to transport a certain bitrate and any transport overhead, measured in bits per second. There will be spare network bandwidth when the (media) data bitrate and overhead is less than the available bandwidth. Similarly, data will have to be buffered when the available bandwidth excluding transport overhead is less than the bitrate used by the sender, or the excess data will be lost. The available bandwidth typically varies dynamically over time.

Bitrate: The amount of (media) data transmitted per time unit, measured in bits per second, utilizing some amount of the available network bandwidth resource. In the context of this specification and unless otherwise specified, it excludes IP/UDP/RTP overhead. Depending on the (media) data source, the bitrate can either be constant or vary dynamically over time.

Codec Configuration Parameter: The configurable value describing a certain codec property, which may impact user-perceived media fidelity, encoded media stream characteristics, or both. The parameter has a type (codec parameter type, see below) and a value, where the type describes what kind of codec property is controlled, and the value describes the property setting as well as how the value should be used in comparison operations. A single parameter value can express one specific value or an open-ended range. A pair of parameter values with different comparison types can describe a value range. Such value range can also be combined with a third, target value within that range.

Codec Operation Point: Also denoted just operation point. A set of codec configuration parameter values, describing the characteristics of one single encoding. For scalable encoding, it describes the resulting characteristics from combining a set of dependent sub-streams.

Codec Parameter Type: The specific type of a codec configuration parameter. Each parameter type defines what unit the value has. This specification defines a number of generally useful parameter types in Section 8 that can be used to control codec operation.

Encoding: A particular encoding is the resulting media stream from applying a certain choice of codec configuration parameters to the encoder. The media stream will have a certain fidelity (quality) from that encoding through the choice of sampling, bit-rate and other configuration parameters.

Endpoint: A host or node that has a presence in the RTP session with one or more Synchronization Sources (SSRC)s.

Mixer: An RTP session centralized node that generates media streams based on incoming media streams from other endpoints. See Topo-Mixer in RTP Topologies [RFC5117].

RTP Session: An association among a set of participants communicating with RTP. The distinguishing feature of an RTP session (defined in [RFC3550]) is that each RTP session maintains a full, separate space of SSRC identifiers. Each participant in the RTP session can see SSRC or CSRC identifiers from the other participants, either by RTP, RTCP, or both.

Sub-Stream: An individually decodeable part of a scalable media stream, including all dependent sub-streams. The characteristics of a certain sub-stream can be described by a codec operation point.

Translator: An RTP session centralized node that forwards all media streams from other endpoints, modified to some extent, e.g. addressing, encoding, fidelity. See Topo-Translator in RTP Topologies [RFC5117].

2.2. Abbreviations

AVC: Advanced Video Coding

AVPF: Extended RTP Profile for RTCP-Based Feedback

CCP: Codec Configuration Parameter

COP: Codec Operation Point

COPN: Codec Operation Point Notification

COPR: Codec Operation Point Request

COPS: Codec Operation Point Status

CPT: Codec Parameter Type

FCI: Feedback Control Information

FMT: Feedback Message Type

GUI: Graphical User Interface

MST: Multi-Session Transmission

MVC: Multiview Video Coding

OP: Operation Point

OPID: Operation Point Identification number

PPS: Picture Parameter Set

SPS: Sequence Parameter Set

SST: Single-Session Transmission

SVC: Scalable Video Coding

TLV: Type-Length-Value

2.3. Requirements Language

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

3. Motivation

3.1. Problem Description

Networks can contain endpoints with different capabilities, including CPU power, capture and render device fidelity (e.g. image resolution), and codecs. In addition, the characteristics and properties of networks can vary, which endpoints have to cope with. For example, in videoconferencing and telepresence services, a large

number of endpoints may participate, and there may be a large number of media streams associated with the session. Such multiparty scenarios typically use entities for media mixing, switching and transcoding. The aim is to provide the best possible quality to each endpoint, taking endpoint and network capabilities into consideration.

Many communication services today use codecs that can be configured in a number of different ways. Often, the codecs have multiple properties that can be configured and those properties may also be inter-related, often in complex ways. One example is the H.264 (AVC) [H264] video codec and its scalable (SVC) and multiview (MVC) versions. Most other video codecs, and codecs for many other types of media, also have multiple configurable properties. Such configurable properties will be referred to as "codec configuration parameters" in this specification.

There can be several reasons to change the media rate or other encoding or packetization properties during an ongoing communication session. Reasons can be that the available network bandwidth varies, or that other network properties change, such as effective MTU or packet rate limitations. Other reasons can be that the quality or representation of the media rendered to the end user changes, maybe as a direct result of the user manipulating the GUI (e.g. changing window position or size), or the relative importance of the received media stream changes (e.g. active or non-active speaker in a conferencing scenario), or the user selects to show some other content source that is available among the advertised media streams.

The codec changes above can be made directly between endpoints in a point-to-point scenario, or they may involve, and be acted upon, by media aware intermediaries (e.g. RTP mixers). An RTP mixer can do transcoding to provide each receiver with media streams of adapted quality, but transcoding has drawbacks as it always consumes processing power, typically impacts media quality in a negative way, and often introduces additional delays.

In order to avoid separate transcoding towards each endpoint, an RTP mixer can, by taking the capabilities of the endpoints into account, decide to request specific codec configurations from sending endpoints, which will minimize the need for transcoding. Also, in scenarios where no RTP mixers are used and transmitted media reaches multiple endpoints, the sender will have to take into account that each endpoint may have different capabilities. The use cases section (Section 4) shows different use cases, with and without RTP mixers.

Resource optimization involving bandwidth is expected to be one of the major reasons for changing encoding properties, since it is desirable to avoid using more bandwidth than absolutely necessary, especially considering that

- o the expectation for high media quality will continue to increase;

- o the bitrate required to transmit the media, despite increasingly efficient media coding, can due to the above also be expected to increase;

- o the available bandwidth is commonly a scarce and/or costly resource and will continue to be in the future;

- o the relation between media bitrate and media codec configuration, the used set of media codec property values, is typically complex and the mapping between each individual codec property and bitrate is not linear;

- o the used media bitrate does not uniquely identify the media codec configuration, but there are multiple codec configurations that can generate the same media bitrate;

- o the media receiver preferences how the codec property values should be set for a certain media bitrate will vary with the specific end-user service requirements (for example, but not limited to, users with special needs) and the current media stream role in the application;

- o the communication scenarios will not be limited to point-to-point, potentially involving multiple and at least partly conflicting constraints from different receivers.

Other resources that may be desirable to optimize include, but not limited to, endpoint and middle node processing (CPU) utilization, and transport quality (QoS).

A media receiver cannot be assumed to know exactly what codec configuration will be best for the media sender to use, given that the sender needs to take multiple aspects into account, including implementation limitations in the actual encoder. It should be more likely to find a value acceptable to both sender and receiver if the receiver can indicate an acceptable range instead of just a single value.

When an RTP mixer distributes streams to multiple receivers with different media quality requirements, it is sometimes possible to avoid targeted transcoding for every single receiver. That can be accomplished if the media sender has the ability to produce multiple media versions, such as for example scalable encoding or simulcast.

Thus, there is a need to both address specific media versions and describe the fact that multiple media versions with different configurations should be used.

3.2. Legacy Methods

3.2.1. Relation to SDP

The session description protocol (SDP) [RFC4566] is commonly used to negotiate and configure codecs, as well as to establish RTP/RTCP session parameters during session establishment and ongoing sessions, e.g. by using it in conjunction with SIP [RFC3261] and SDP Offer/Answer [RFC3264].

As described in Section 3.1, many of the underlying reasons which make media receivers desire certain codec encoding properties are highly dynamic in nature and using SIP/SDP to renegotiate the session will in many cases be too slow to be useful. SIP messages containing an SDP may become quite large for sessions containing many media types, and since there is no defined way to send a partial SDP, even very small changes require sending the entire SDP. Most of the current defined properties in SDP are oriented to be common for all media streams in the same RTP session, at least the ones sharing the same RTP Payload Type, rather than being specific to one media stream (e.g. "a=fmtp:98 profile-level-id=42C00C").

The mechanism in this specification does not replace SDP, or the SDP Offer/Answer mechanism. It is expected that SDP is used in order to negotiate and configure boundary values for codec properties, and COP can then be used to communicate specific values within those boundaries, as long as there is no impact on the values negotiated using SDP. It is possible to establish communication sessions even if one or more endpoints do not support COP.

3.2.2. Relation to RTCP

As discussed in CCM, regular RTCP reporting or extended reports [RFC3611] can to some extent be used to reconfigure an encoder, but the reported measures seldom map directly back to encoding properties and they typically cannot express an unwanted situation in terms of encoding properties and what the receiver would like to receive instead. Communicating codec properties indirectly as a set of network properties will require interpretation by both sender and receiver and will thus risk misinterpretations and ambiguity. Since it is likely that a decoder is able to identify unwanted characteristics of the media stream in terms of encoding properties, the most straight forward approach is to convey those properties directly to the encoder.

Responsive techniques to control encoding are already available, e.g. Codec Control Messages (CCM) [RFC5104]. Although highly applicable, the possibilities to control encoding is however not explicit enough, both in terms of the amount of available parameters to control, and the fact that they may be inter-related, alternative, or both.

Some codecs define codec-specific methods to enable receiver control of some encoding aspects, but it should be beneficial for interoperability to use codec agnostic signaling instead.

4. Use Cases for COP

This section discusses a number of use cases for codec operation points.

4.1. Point to Point

This set of use cases focuses on communication, which is directly point to point between a media sender and a receiver. There is no need for further forwarding of the media streams. Thus, the goal should be to produce a media stream, transport it to the media receiver, where it is consumed as optimal as possible for the application. Thanks to this one-to-one mapping between encoder and decoder, great flexibility exists to produce a media stream tailored to the receiver's needs, given the constraints that exist from media sender, transport network and the receiver.

Some constraints are static (and thus suitable for session configuration signalling), but others are highly dynamical and desirable to adapt to during the session:

Video Resolution in GUI: In a video communication application, including WebRTC based ones, the window where the media senders media stream is presented may change, for example due to the user modifying the size of the window. It might also be due to other application related actions, like selecting to show a collaborative work space and thus reducing the area used to show the remote video. In both of these cases it is the receiver side that knows how big the actual screen area is and what the most suitable resolution would be. It appears suitable to let the receiver request the media sender to send a media stream conforming to the displayed video size.

Network Bit-rate Limitations: If the receiver discovers a network bandwidth limitation, it can choose to meet it by requesting media stream bit-rate limitations. Especially in cases where a media sender provides multiple media streams, the relative distribution

of available bit-rate can help the application to provide the most suitable experience in a constrained situation.

CPU Constraint: A media receiver may become constrained in the amount of available processing resources. This may occur in the middle of a session for example due to the user selecting a power saving mode, or starting additional applications requiring resources. When this occurs, the receiving application can select which and how much to constrain codec parameters to best suit the needs of the application. For example, if lower framerate is a better constraint than lower resolution.

4.2. Media Receiver to RTP Mixer

This section considers a multiparty session with a centralized media intermediary, like an RTP mixer, where the media receiver uses COP to affect the delivered media.

```
                   +------------+        +---+
                   |            |--RTP-->| B |
                   |            |<--COP--|   |
                   |            |        +---+
                   |            |
        +---+      |            |        +---+
        | A |-RTP->|   Mixer    |--RTP-->| C |
        +---+      |            |        +---+
```

```
                   |             |
                   |             |        +----+
                   |             |--RTP-->| D  |
                   +-------------+        +----+
```

Figure 1: Receiver (B) using COP to adapt a media stream

In the above Figure 1 we focus on the possible usages of COP by a media receiver, like B. Here the functional role of the intermediary becomes important (Topo-Mixer) [RFC5117]. An RTP mixer uses its own SSRC(s) to channel selected media streams to B from other participants like A. If the intermediary is instead a translator, the Receiver B can see A's SSRC(s) directly instead of possibly showing up as CSRC. We will in this section focus on the mixer case. The RTP translator case is further discussed in Section 4.4.

The RTP mixer's usage of its own SSRC allows mixer to receiver media flows to be associated with a role or purpose in the application rather than a given media source. Based on the assumption that the set of available stream roles are connected to the specific use case or application, it is likely that the set of stream roles (for example most active speaker) provided from a mixer will change less often than the original media source representing that role is

changed. It is further assumed that the desirable media characteristics related to a specific role will be fairly constant. To minimize the amount of signaling needed to modify stream characteristics, it could thus be appropriate to let a stream represent a role rather than limiting it to represent the original source. When there exist multiple RTP streams from the mixer to a receiver, the receiver can use COP to request an operations point that better suits the receiver's needs on each particular stream (role) of the media stream. COP also allows the receiver to select its desired trade-off in properties and quality between multiple delivered media streams.

There exist different reasons why B would need to indicate changes in its capabilities to receive a particular media stream:

Network Path: The receiver detects changes in the network that on a mid to long term will result in a new capability regarding the maximum bit-rate that can be supported.

Bandwidth Trade-off: In an application receiving multiple media streams, if the receiving application likes to change the relative bit-rate trade-off between the streams.

Presentation or GUI Changes: If the presentation or graphical user interface (GUI) changes on the receiving side this results in other requirements or needs on the media streams. For example if the application window is resized by the user, the amount of screen estate to present the different video elements changes. To optimize the video quality in relation to bit-rate the receiver indicates the new preferred video resolution.

In all the above cases the receiver sends a COP request to the mixer for new codec operation points on mixer controlled media stream(s). It then becomes the mixer's responsibility to determine if and how the requested COPs can be supported. For example by requesting new operations points from the media source as discussed in Section 4.3. The selection of another media source to deliver in a media stream can result in that the mixer may have to update the receiver on the properties of the operations point.

4.3. RTP Mixer to Media Sender

This section looks at the usage of COP in cases of multiparty with centralized media intermediary, like an RTP mixer, selecting and requesting tailored media stream or streams a media sender delivers to the intermediary for further forwarding or manipulation. This usage can be simplified to the media streams from one media sender (A), which is currently being delivered to multiple receivers (B-D)

as depicted in Figure 2.

```
                   +------------+        +---+
                   |            |--RTP-->| B |
                   |            |        +---+
              +---+|            |
              | A |<-COP-|      |        +---+
              |   |-RTP->| Mixer |--RTP-->| C |
              +---+|            |        +---+
                   |            |
                   |            |        +---+
                   |            |--RTP-->| D |
```

```
                        +------------+       +---+
```

Figure 2: Mixer using COP to adapt media streams to multiple receivers

The media path from the mixer to B, C and D are different and thus the available resources may vary between them. In addition B, C and D may have different capabilities when it comes to handling media streams. These limitations can be learned by the mixer through session configuration signalling, media transmission feedback (e.g. RTCP), or usage of COP by the receivers (See Section 4.2). Limitations are also expected to be updated during the session lifetime.

The media sender (A) has certain capabilities and what is possible to do will depend on A's capabilities and what has been configured between A and the mixer. Let's consider different capabilities of A and how they influence the usage of COP to affect the media stream(s) delivered to the mixer.

Single Media Encoding: If A can only provide a single media encoding of a particular media source, the mixer has to make a choice on what property it would like to request for that media stream. The most basic choice is to request the lowest common denominator across the receiver population. If the mixer has certain capabilities for media transcoding it could select to request another operation point for the media encoding with higher quality and then transcode to some few receivers. That enables a higher quality to several receivers while still being able to serve endpoints with the least capabilities. In these cases the mixer has to send COP requests that indicate only a single operation point with parameters matching the restrictions in the best possible way.

Scalable Media Encoding: If A is capable of producing a scalable media stream encoding, the mixer can request multiple operation points for the same media stream. For example, if A is capable of producing three different operation points, the mixer in the above

Figure 2 would be able to request scalability layers that match the capabilities of all three receivers B, C and D. If several receivers have similar capabilities, the mixer may choose to request fewer operation points. In this case, other than in the single media encoding, the mixer must determine which packets or parts of packets to send to each receiver based on their capabilities. This requires that the mixer is capable of identifying in the media stream which scalability layer matches a requested operation point. Thus, it is desirable that the media sender can indicate to the mixer which layer matches a given operation point.

Simulcast Media: If A and the mixer have negotiated the usage of simulcasted media encoding of the media source, then the mixer can adopt several operation points to best suit the receivers, just like for scalable encoding. When simulcasting, the mixer will however have to send one COP request per media stream it actually wants to affect. It is necessary to ensure that configuration changes over multiple media streams from the same media source take place. Compared to scalable media, the mixer does not need not strip away layers to match a particular operation point but can forward entirely self-contained media streams.

The use of COP as described above can be triggered by a multitude of reasons. We will here discuss some of them. We already mentioned that bit-rate adaptation (congestion control) on the mixer to receiver path can indicate a need to change an operation point. Another reason is when a new session participant joins that has certain receiver capabilities (both decoding or other hardware, as well as network path related), thus potentially changing the optimal set of operation points. There also exist a number of different cases where the desired application behavior results in changes in desired operation points, like change of active speakers, reconfiguration of the display layout, etc.

It is important to remember that Figure 2 only presents the view of a single media sender. In most communication sessions there are multiple media senders, and the mixer will need to take the combination of media streams from multiple media senders into account when choosing what is to be sent to a given receiver. Thus changes at one media sender can result in related changes of the operation points at the other media senders.

4.4. Media Receiver in Multicast or with RTP Transport Translator

This section covers the usage of COP in multicast transported RTP sessions, as well as when transport translators (Topo-Translator) [RFC5117] are used. Transport translators can be used to emulate any source multicast (ASM) over unicast. Multicast usages also include Source Specific Multicast (SSM) [RFC4607], which according to "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback" [RFC5760] has two main modes: simple mode, and summary feedback mode. SSM modes affect the usage of COP functionalities.

```
+---+      +------------+      +---+
| A |<---->|            |<---->| B |
+---+      |            |      +---+
           | Translator |
+---+      |            |      +---+
| C |<---->|            |<---->| D |
+---+      +------------+      +---+
```

Figure 3: RTP translator topology

A transport translator [RFC5117], which main purpose is to forward any incoming packets to all the other session participants, emulates an ASM session (see Figure 3). As anyone can send to all other in both cases, there are some properties in these large scale sessions with many participants which require extra consideration.

```
 +------+ +------+         +------+
 | MS1 |  | MS2 |   ....   | MSm |
 +------+ +------+         +------+
    ^        ^                ^
    |        |                |
    V        V                V
+----------------------------------+
|       Distribution Source       |
+---------+                       |
| FT Agg |                        |
+---------+------------------------+
  ^ ^           |
  :  .          |
  :   +...................+
  :             |          .
  :            / \          .
+-------+     /   \       +------+
| FT1  |<-----+    +----->| FT2 |
+-------+   /       \     +------+
  ^  ^     /         \     ^  ^
  :  :    /           \    :  :
  :  :   /             \   :  :
  :  :  /               \  :  :
  :   ./\               /\.   :
  :   /. \             / .\   :
  :  V  . V           V .  V  :
 +-----+ +-----+    +-----+ +-----+
 | R1 | | R2 | ... |Rn-1| | Rn |
 +-----+ +-----+    +-----+ +-----+
```

Figure 4: SSM based RTP session

In the above Figure 4, the media senders (MS1 ... MSm) send their media streams and RTCP traffic to the distribution source (DS). The DS forwards the RTP and RTCP traffic from the media senders to the SSM group. Using the RTCP extension for unicast RTCP feedback [RFC5760], the receivers (R1...Rn) send their RTCP traffic to their configured feedback target. This sample session has two feedback targets to scale with the amount of receivers. RTCP messages that need to go to a media sender are forwarded to the FT aggregator part of the distribution source for further forwarding over the unicast paths between the distribution source and the media senders. The feedback target and the feedback aggregator also forward all RTCP messages from receivers in simple mode, and aggregate it in summary mode. Some RTCP messages from a receiver may still have to be forwarded over the SSM group.

COP needs to support some reasonable functionality over the different

multiparty topologies described above and it is important that COP does not cause significant issues in any of the environments.

In the basic case, where only a single multicast group exists, there is a well known problem associated with adapting content and bit-rate to the receiver population. The more receivers, the larger the potential for non-matching requirements in requests from the different receivers. One strategy for meeting this is to use the lowest common denominator among the requests from the receiver population. This normally results in sub-optimal quality for a significant part of the session participants, the main benefit being that all participants will be able to receive some content.

Because of the above limitations of operation within a single group, the usage of COP in larger groups becomes difficult unless the parameters that can be adopted and affected by COP requests are such that a limited set of participants is expected to request them, and the impact for the others are limited or acceptable. The authors therefore expects the usage of COP in large groups to be limited and this specification focuses on operation in smaller groups. However, as it is not possible to define the threshold when a group changes from being small to be too large to work well with COP in the generic case, it is important that COP can operate safely in a large group, although the possibilities to satisfy the request may be severely limited.

There also exist use cases for COP where the media application uses multiple multicast groups to enable multiple operation points and allows each receiver to join the multicast groups that suits the participant's capabilities. An example of such usage would be Scalable Video Coding (SVC) using the Multi-Session Transport (MST) mode of the SVC RTP payload format [RFC6190]. The SVC MST RTP streams that are sent in each group can still contain multiple scalability layers. One could combine coarse-grained control on the operation points by having the receiver join a particular session with a more fine-grained control using COP to adjust the included scalability layers to the receiver's needs, such as lower CPU load.

5. Requirements

The solution outlined in this specification should fulfill the following requirements:

REQ-1: Enable dynamic control of possibly inter-related codec properties during an ongoing media session.

REQ-2: Be media type agnostic, to the furthest extent possible, and at least cover audio and video media.

REQ-3: Be codec agnostic (within the same media type), to the furthest extent possible.

REQ-4: Work with different media transmission types, i.e. single-stream, simulcast, single-stream scalable, and multi-stream scalable transmission.

REQ-5: Work with un-encrypted as well as encrypted media.

REQ-6: Be extensible, making it simple to add control and description of new codec properties.

REQ-7: Complement rather than conflict with other codec configuration methods such as other RTCP based techniques and SDP.

REQ-8: Support configurable parameters that are directly visible in the media stream as well as those that are not visible in the media stream.

In addition, Guidelines for Extending RTCP [RFC5968] should be followed.

6. Solution Overview

The mechanism described in this specification especially targets heterogeneous multiparty scenarios where different endpoints require differently encoded media from the same source, but its use in other situations is not precluded. In fact, point-to-point scenarios are considered to be of equal importance but not more demanding that the multiparty case. In the targeted scenario, the media stream from one encoder is sent to multiple decoders. Hence, the encoder must possibly provide an encoding with multiple operation points, suitable for the receivers. This is only possible with so-called scalable codecs, but some codecs may have inherent scalability features without being generally considered as scalable (e.g. H.264/AVC temporal scalability through non-reference frames). Multiparty services often involve a media mixer (Topo-Mixer) [RFC5117] as a central network node.


```
                             +----+
                             | S  |
                             +----+
                               |
                               v
                          +--------+
                          | Mixer  |
                          +--------+
                         /     |     \
                        v      v      v
                    +----+  +----+  +----+
                    | A  |  | B  |  | C  |
                    +----+  +----+  +----+
```

Figure 5: RTP mixer topology

The solution defined in this specification is targeted for automatic control of codec parameters, not as a direct result of user interaction, although the automatic control can in turn be triggered by user interaction. It can be used during an active session to quickly adapt to changes in media receiver available bandwidth and/or preferences for one or more codec properties, while still conforming to the session configuration, like SDP offer/answer negotiated minimum or maximum limits (depending on individual SDP property semantics). Some codec property changes will also motivate to renegotiate the SDP, but the scope of this specification intends to cover only changes that lie within the SDP negotiated set and thus do not impact the SDP.

Three message types are defined to support the solution: a request, a notification, and a status report:

Request (COPR): A media receiver requesting a media sender to adjust one or more of it's media encoding parameters for a media stream. The request is normally based on a specific set of media encoding parameters that the media sender has explicitly notified the media receiver about in a notification.

Notification (COPN): A media sender notifying a media receiver of the currently used media encoding parameters for a media stream. The notification is initiated by the media sender, typically whenever the media encoding parameters changed significantly from what was previously used. The reason for the change can either be local to the media sender (user, endpoint or network), or it can be the result of one or more requests from remote endpoints.

Status Report (COPA): A media sender reporting to a request sender (media receiver) on request reception status, which specific request from the media receiver that was received and considered in setting current media encoding parameters, and the identification of the media stream that is considered to fulfill the request. The status report can also indicate various error conditions, such as reception of invalid or failing requests.

More details about the individual messages are found in the following sub-sections.

6.1. Message Structure

A COP message is sent from an RTP session participant in its role either as media receiver or media sender. Each message can contain one or more message items of one or more message types, all originating from a single media source.

The individual message items each relate only to a single operation point, describing part of an atomic notification or request.

The general structure is outlined below:

```
+---------------------------------------+
| AVPF PSFB FMT="COP"                   |
| SSRC of Packet Sender                 |
| SSRC of Media Source                  |
| +-----------------------------------+ |
| | COP Message Item 0                | |
| +-----------------------------------+ |
| | (Codec Configuration Parameters) | |
| +-----------------------------------+ |
| +-----------------------------------+ |
| | COP Message Item 1                | |
| +-----------------------------------+ |
| | (Codec Configuration Parameters) | |
| +-----------------------------------+ |
| ...                                   |
+---------------------------------------+
```

Figure 6: COP message structure

Note that the request is the only COP message item defined in this specification that is sent in the media receiver role and makes use of "SSRC of media source" as the targeted media stream for the request. Both the notification and the status report message items are sent in the media sender role, reporting on the message sender's own configuration and thus relate only to the "SSRC of packet sender", being agnostic to the "SSRC of media source" field.

It is for example possible to collocate COPS and COPN messages for the same media source in the same COP FCI. It is also possible to co-locate one or more COPR referring to a single "SSRC of media source" with one or more COPN and/or COPS relating to a single "SSRC of packet sender" within a single COP message.

Multiple message items of the same type in the same COP message are used to describe a notification, status or request for a media stream containing multiple operation points (see Section 6.3).

Multiple COP messages are needed to be able to refer to multiple different "SSRC of packet sender" and/or "SSRC of media source".

6.2. Codec Configuration Parameter Use

The codec configuration parameters that are applicable to a certain codec may be specific to the media type (audio, video, ...), and may also be codec specific. Some codec properties (described by codec configuration parameters) have to be explicitly enabled by (non-RTCP based) capability signaling to be possible or permitted to use.

An endpoint implementing this specification does not need to support all available codec configuration parameters defined herein or in extensions to this specification. A certain parameter could be unnecessary for a certain codec or media stream, even if it is generally supported by the endpoint. This specification therefore defines capability signaling that allows a COP receiver to declare explicit support per parameter type on a per codec level. The set of codec configuration parameters that can be used for a certain media stream by a COP sender is thus restricted by the combination of applicability, capability signaling and explicit receiver parameter support signaling.

Any codec configuration parameter that is applicable and feasible to use, but is not included as part of an operation point, has a default value. This default value is defined for each parameter type, but should preferably whenever possible be taken from capability signaling. It is not necessary to use all defined parameter types in a media stream description. Some parameter types can, depending on media type or codec, either be unnecessary, or not possible to describe or control in detail, in which case they can be left out. This means that the effective value is "undefined" within the limits set by capability signaling (outside the scope of this specification).

6.3. Operation Point

The codec configuration parameters contained in a single message item jointly constitute a description of an operation point for a specific media stream from a media sender.

For the purpose of COP signaling, each operation point is identified with an identity number (OPID), which is scoped by the media sender's RTP SSRC identification, and can be chosen freely by the media sender. The need for this media sub-stream identification only appears with scalable coding or other media encoding methods that introduce separable and configurable sub-streams within the same SSRC. An OPID thus refers to such configurable sub-stream, described by a set of related codec configuration parameters.

```
                        +--RTP Session 1 --------------------+
   Media Source 1----+-+-> SSRC1 --> Sub-Stream 1 -> OPID1 |
   (MIC, Camera)     |           \-> Sub-Stream 2 -> OPID2 |
                     |                                     |
   Media Source 2-+--+---> SSRC2 --> Sub-Stream 1 -> OPID3 |
                  |  |           \-> Sub-Stream 2 -> OPID4 |
                  |  |           \-> Sub-Stream 3 -> OPID5 |
                  |  +-------------------------------------+
                  |
                  |  +--RTP Session 2 --------------------+
                  +--+---> SSRC3 --> Sub-Stream 1 -> OPID6 |
                     |           \-> Sub-Stream 2 -> OPID7 |
                     +-------------------------------------+
```

Figure 7: Relation of OPID to media source, RTP session and SSRC

Figure 7 depicts the possible relations between media sources, RTP sessions, RTP streams (SSRCs), RTP sub-streams, and the OPID.

For example, a single video camera may be encoded using SVC for a combined SST and MST transmission configuration. In that case a subset of scalability layers is sent as SST in the first RTP session using SSRC2. Another set of scalability layers is transported in the second RTP session as another SST using SSRC3. The RTP packet stream from each SSRC can thus contain several sub-streams, each identified with its own OPID. As a result, a single media source is present in two RTP sessions, using two different SSRCs (2 and 3) containing a total of five sub-streams (OPID 3 to 7).

Since an operation point is expected to change over time, as a result of media receiver requests (Section 6.4), resulting from local media sender considerations (Section 6.5), or both, the operation point (OPID) is version handled. The version is scoped by SSRC and OPID.

It is expected that all encoders dividing a media stream into sub-streams will include some means to identify those sub-streams in the media stream. However, it is also expected that such identification is in general codec specific. There is thus a need to map the codec agnostic COP OPID identification to codec specific identification, and this specification therefore includes a method for such mapping (Section 10).

6.4. Request

The request is sent by a media receiver, which can be either an endpoint or a middle node such as an RTP mixer. The receiver of the request may similarly be either the original media sender or a RTP mixer. Included in the request is a description of the desired codec configuration for a specific media (sub-)stream. The parameter values communicated in a notification (Section 6.5) of that (sub-)stream are taken as a starting point when deciding what parameters and parameter values to choose for the request, and only parameters with changed values need to be included the request. The media receiver can of course use other sources of information when choosing parameters and values, for example observation of the received media stream and capability signaling.

It is not required to receive a notification beforehand to be able to create a meaningful request. The request can include a set of changed properties for existing streams, but it can also request the addition or removal of one or more media sub-streams having certain properties, in which case there will be no notification to base the request on. A media receiver may also want to send a request prior to having received any notifications for existing streams, and can then base the request on other information such as for example observing the media stream or use information from the capability signaling. In case there is no existing stream and OPID to refer to in the request, a "provisional" OPID MUST be chosen in the request, which will have to be mapped back to an existing (sub-)stream and "real" OPID through methods defined in this specification (Section 10).

The media sender receiving a specific request is not required to reconfigure the encoder accordingly, even if it should try to do so. The media sender is allowed to take other (previous or concurrent) requests and any local considerations into account, possibly modifying some of the parameter values, or even to reject the request completely if it is not seen as feasible. It is thus not possible for a media receiver to uniquely see from the media stream or even from a notification if the media sender received the request or if the request was lost and needs to be resent.

A request should be based on a notification, but there may be situations where a request is sent approximately simultaneously with a new notification for the same stream. In that case, there is a risk that the request is based on the wrong set of codec properties compared to the new notification. It is therefore necessary to have the set of codec properties version controlled, identified by an OPID. If a notification announces a specific version of the operation point, where the version is updated every time it is changed, the request can refer to that specific version and any mis-reference can be clearly identified and resolved. In addition, it allows for easy identification of repeated notifications and requests by checking the operation point identification and the version, without the need to parse through all codec properties for changes.

6.5. Notification

The notification is sent by a media sender and describes a media stream or sub-stream in terms of a defined, finite set of codec properties. That same set of codec properties can also be used in a request (Section 6.4). The notification and the set of defined properties is important to be known at the media receiver since it is rarely possible to see from the media stream itself what controllable properties were used to generate the stream. The set of codec properties and their values used to describe a certain media stream at a certain point in time are henceforth called a codec configuration. Each operation point in this codec configuration is implemented using an RTP payload type, defined by capability signaling outside the scope of this specification.

It must be possible for a media sender to change the codec configuration not only based on requests from media receivers, but also based on local limitations, considerations, or user actions. This implies that the notification can be sent standalone and not only as a response to a request (compare TMMBR and TMMBN [RFC5104]). To avoid that media receivers have to guess what codec configuration is used, a media sender should always send a notification when the codec configuration for a stream changes. Loss of a notification messages should not be critical since a media receiver could either fall back to infer the approximate codec configuration from the media stream itself, or simply wait with a request until the next notification is sent.

A notification can potentially contain a large amount of codec properties. However, parameters that are not enabled by codec and COP capability signaling, or inherently are not part of the used codec will not be included. The notification only describes the currently used codec configuration, and each parameter of an operation point will be described by a single value. To further

limit the amount of properties to be sent, it is possible to rely on parameter defaults (listed by individual parameter type definitions) whenever those values are acceptable.

The media receiver could want to take local action at the time when the codec configuration in the media stream changes. Using the same reasoning as above, this may not be possible to see from the media stream itself. This functionality is explicitly enabled by including the RTP time stamp in the notification, where the time stamp describes a time (possibly in the future) when the codec configuration is (estimated to be) effective.

It is not required that a media sender sends notifications for all media streams or sub-streams. However, the non-announced streams or sub-streams will then not be accessible to media receiver control (Section 6.4). Any media or transport resources occupied by those non-announced streams (in COP terms) must be excluded from the total amount of available resources when deciding feasible parameter value ranges for the announced streams.

6.6. Status Report

The status report is sent by a media sender and is needed to confirm reception of a request OPID to avoid unnecessary retransmission of requests. Loss of a status report will likely trigger a request retransmission, except when the request sender can infer from the media stream or a notification that the stream is now acceptable.

The status report is not a required acknowledgement of every request, but instead reports on the last received request, identified by a request sequence number in addition to the OPID. This decoupling of requests and status reports reduces the needed amount of status reports in case of frequently updated requests and/or lack of resources to send status reports.

If a request is somehow not acceptable to a media sender, the status report can also indicate failure and a reason for failure.

In case the OPID in the request is a "provisional" OPID (Section 6.4), the status report responds with that exact OPID, but also includes a reference to a "real" media (sub-)stream identification or OPID that the media sender considers appropriate for the request.

No description of any codec configuration is included in a status report, even if the corresponding request was successful. The codec configuration is only carried in the notification (Section 6.5) message. Multiple status reports targeted for multiple request

senders can through media (sub-)stream identification and OPID point to the same notification message, reducing the need to repeat applicable codec configuration parameters with every accepted request.

6.7. Adding and Removing Operation Points

A media sender can unilaterally create a new operation point by simply selecting a free OPID identifier and use COPN to announce it.

To remove an operation point, the media sender simply stops announcing it in COPN. This procedure can be used both for entire media streams containing a single operation point and to add/remove sub-streams in media streams containing multiple operation points.

The media receiver can request a new operation point to be created by using a COPR with an unused identifier and by setting a flag to indicate that this requests a new OPID. The media sender then decides if it honors the request or not, and announces the new OPID as described above.

The media receiver can indicate that it is no longer interested in receiving an operation point corresponding to a media sub-stream by not including any COPR message item for it in a single COP message. The media receiver can indicate a wish to continue to receive an unmodified operation point using a COPR without any codec properties (no change).

7. Codec Control Message Extension

This specification specifies a new feedback message, COP, for codec control of real-time media, as an extension to the AVPF [RFC4585] and CCM [RFC5104] specifications. The AVPF specification outlines a mechanism for fast feedback messages over RTCP, which is applicable for IP based real-time media transport and communication services. It defines both transport layer and payload-specific feedback messages. This specification targets the payload-specific type, since a certain codec is typically described by a payload type.

AVPF defines three and CCM defines four payload-specific feedback messages (PSFB). All AVPF and CCM messages are identified by means of the feedback message type (FMT) parameter. This specification specifies one additional payload-specific feedback message.

One new PSFB FMT value is assigned in this specification:

TBA1:   Codec Operation Point (COP)

This section defines the feedback message structure, message items and their semantics with the exception of the actual codec configuration parameters which are defined in the next section (Section 8).

7.1. COP Message

The COP message is a payload-specific AVPF CCM message identified by the PSFB FMT value listed above. It carries one or more COP message items, each with either a request for, a description of a certain "operation point"; a set of codec parameters, or a request status indication.

Not all message items makes use of the "SSRC of media source" in the common packet header. "SSRC of media source" SHALL be set to 0 if no message item that makes use of it is included in the FCI.

7.2. FCI Format

The COP FCI MUST contain one or more codec operation point message items. The maximum number of COP message items in a COP message is limited by the [RFC4585] Common Packet Format 'length' field.

The definition of the AVPF feedback message format mandates that the FCI part is a multiple of 32-bit words. The below defined message items will not be 32-bit word aligned. Therefore it is sometimes necessary to insert one to three padding bytes at the end of the FCI. The number of padding bytes are determined by a receiver by comparing the sum of the message items and the feedback message length fields. The padding byte MUST be set to zero (0) and ignored on reception.


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|FMT=TBA1 |      PT=206   |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      SSRC of packet sender                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|                      SSRC of media source                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    COP message item header #1                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    COP message item payload #1                :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:               |         COP message item header #2            :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:               |         COP message item payload #2           :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                              ...                              :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
: COP message item payload #N   |          Padding (0)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: COP RTCP Message Structure

7.2.1. Message Item Format

All codec operation point message items share a common header format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type |       Payload Length    |      OPID     |N|   Version   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                    (Message Item Payload)                     :
```

Figure 9: COP message item header format

The message header fields are:

Type (3 bits): Message item type. Three item types are defined in this specification, COPR, COPN and COPS, with values as listed in Table 1 below. More item types MAY be defined in extensions to this specification. Message items with a type field that has an unknown value SHALL be ignored by the receiver.

Payload Length (13 bits): The total length in bytes of all data belonging to this message, following the message item header, i.e. anything following the Version field.

OPID (8 bits): Operation point ID. Some (typically scalable) codecs are capable of encoding into multiple simultaneous operation points using the same SSRC, and each operation point can then be referenced by OPID. MUST be unique within the scope of an SSRC when N flag is not set. MUST be set to 0 for message items not using the field. See also Section 7.2.3.

N (1 bit): A "New OPID" flag, indicating that the OPID value is chosen arbitrarily and is not meant to refer to any existing operation point. The message sender SHOULD NOT use an already known OPID in combination with the N flag. See also individual message item definitions.

Version (7 bits): Referencing a specific version of the codec configuration identified by the OPID.

7.2.2. Message Item Types

The message types defined in this specification are:

Value	Message Item Type
0	Codec Operation Point Notification (COPN)
1	Codec Operation Point Request (COPR)
2	Codec Operation Point Status (COPS)
3-6	Unassigned
7	Reserved for future extensions

Table 1: Message Item Type Values

Each message type defined in this specification is described in detail in subsequent sections.

7.2.3. Operation Point Identification

All RTP media streams belonging to the same session can per definition be identified by the SSRC. However, identification of any sub-streams contained in the same RTP media stream (SSRC) needs to use some other identification method, scoped by the SSRC. This is the case for a media stream containing more than one operation point, like for example SVC [RFC6190] streams being sent using Single Stream Transport (SST) RTP packetization.

The encoding of and restrictions for such sub-stream (operation point) identification will in general be codec specific. Therefore, the OPID used in this specification is merely an SSRC-unique identification number. It is however necessary to create a mapping between this generic number and the codec specific sub-stream identification that can be found in the media stream. This mapping is achieved by including the ID parameter (Section 8.3) in a message item carrying a certain OPID.

In Section 10, codec specific ID parameter formats are defined for a few of the most common codecs that supports scalability.

7.3. Codec Operation Point Notification

7.3.1. Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type |      Payload Length     |     OPID      |N|   Version   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Transition Time Stamp                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|Payload Type |    Codec Configuration Parameters             :
+-+-+-+-+-+-+-+-+                                               :
:                                                               :
```

Figure 10: COPN format

The COPN-specific message fields are (see also message item format (Section 7.2.1)):

Type (3 bits): Set to 0, as listed in Table 1.

OPID (8 bits): The OPID which is described by the codec configuration parameters.

N (1 bit): Not used by COPN and SHALL be set to 0 by senders.

Version (7 bits): Referencing a specific version of the codec configuration identified by the OPID. SHALL be increased by 1 modulo 2^8 whenever the used codec configuration referenced by the OPID is changed. A repeated message SHALL NOT increase the Version. The initial value SHOULD be chosen randomly.

Transition Time Stamp (32 bits): The RTP Time Stamp value when the listed codec configuration parameters will be effective in the media stream, using the same time line as RTP packets for the referenced SSRC (media sender SSRC). The Time Stamp value MAY express either a time in the past or in the future, and need not map exactly to an actual RTP Time Stamp present in an RTP packet

for that SSRC. The same timestamp value SHOULD be used for subsequent transmissions of the identical set of codec configuration parameters for the same OPID and version.

R (1 bit): Reserved. MUST be set to 0 by senders and MUST be ignored by receivers implementing this specification. MAY be defined differently by extensions to this specification.

Payload Type (7 bits): SHALL be identical to the RTP header Payload Type valid for the (sub-)stream described by this OPID.

Codec Configuration Parameters (variable length): Contains zero or more TLV carrying codec configuration parameters as defined in parameter types (Section 8).

7.3.2. Semantics

This message is used to inform the media receiver(s) about used codec configuration parameters at the media sender. The available codec parameter types that can be used to describe the codec configuration are defined in Section 8.

Some codecs may have clear inband indications in the encoded media stream of how one or more of the codec configuration parameters are configured. For those codecs and codec configuration parameters, COPN is not strictly necessary. Still, for some codecs and / or for some codec configuration parameters, it is not unambiguously possible to see individual codec configuration parameter values from the encoded media stream, or even possible to see some codec configuration parameters at all, motivating use of COPN.

COPN SHOULD be scheduled for transmission when it becomes known that there are media receivers in the RTP session that did not yet receive any codec configuration parameters for an active operation point, or whenever the effective codec configuration parameters has changed significantly, but MAY be scheduled for transmission at any time.

The media sender decides what amount of change is required to be considered significant.

The reason for a codec configuration parameter change can either be local to the sending terminal, for example as a result of user interaction or some algorithmic decision, or resulting from reception of one or more COPR messages (Section 7.4).

If a media sender can no longer fulfill the established codec configuration parameter restrictions of a operation point that was previously described by a COPN, it MAY change any codec configuration parameter or even remove the entire operation point, and SHOULD then

signal this at the earliest opportunity by sending an updated COPN to the media receiver(s).

An OPID can implicitly be indicated as no longer being used by omitting that OPID from the set of COPN message items in the COP PSFB message. All OPIDs that the media sender intends to use at the latest time indicated by any transition timestamp value in the set of COPN present in the COP PSFB message, MUST be included in that COP message.

All operation points referred by a COPS (Section 7.5) SHOULD also be detailed by a COPN message contained in the same or in a subsequent COP feedback message, even if the operation point did not change significantly from previous COPN.

Note that the OPID Version of that COPN, subsequent to COPS, will be equal or larger than the Version indicated in the COPS. The Version difference may be larger than one (taking field wraparound into account) depending on the number of updated COPN sent since the COPR that triggered the COPS. See also description of those messages below.

Note: COPN may be seen as a more explicit and elaborate version of the TSTN message of [RFC5104] and most of the considerations detailed there for TSTN also apply to COPN.

7.3.2.1. Parameters

The media sender decides what codec configuration parameters to use in the COPN to describe an operation point. It is RECOMMENDED that all codec configuration parameters that were accepted as restrictions based on received COPR messages are included. All codec configuration parameters significantly more restrictive than implicit or explicit restrictions set by capability signaling (outside the scope of this specification) SHOULD also be included. Any codec configuration parameter that are either not applicable to the Payload Type or not enabled by capability signaling MUST NOT be included. All codec configuration parameters not covered by the above restrictions MAY be included.

When the operation point has dependency to other operation points (such as in scalable coding), the values to use for codec configuration parameters MUST describe the result when all dependencies are utilized. For example, assume an operation point describing a base layer with 15 Hz framerate, and a dependent operation point describing an enhancement layer adding another 15 Hz to the base layer, resulting in 30 Hz framerate when both layers are combined. The correct parameter value to use for that latter,

dependent "enhancement" operation point is 30 Hz, not the 15 Hz difference.

The value of a codec configuration parameter that was not included in a COPN message SHOULD either be inferred from other signaling, e.g. session setup or capability negotiation, outside the scope of this specification, or if such signaling is not available or not applicable, use the default value as defined per parameter type (Section 8).

An operation point describes one specific setting of codec parameters, and a COPN message therefore MUST NOT include the ALT parameter type (Section 8.2) in the codec parameters describing the operation point.

7.3.2.2. Relation to COPR

To limit RTCP bandwidth and avoid bandwidth expansion, COPN is not mandated as response to every received COPR (Section 7.4).

A media sender implementing this specification SHOULD take requested operation points from COPR messages into account for future encoding, but MAY decide to use other codec configuration parameter values than those requested, e.g. as a result of multiple (possibly contradicting) COPR messages from different media receivers, or any media sender policies, rules or limitations. Thus, a COPN message operation point MAY use other codec configuration parameters and other values than those requested in a COPR.

The media sender SHOULD try to maintain OPIDs between COPR and COPN when COPR sender suggests a new OPID value (N flag is set) in the COPR, but MAY use another OPID in COPN. Examples where other OPID values have to be chosen are for example when the suggested OPID conflicts with an already existing OPID, or when the media sender decides that a the suggested new OPID can be fulfilled by an already existing OPID.

Even if a COPR references an existing OPID (N flag cleared), the media sender may have to take other aspects than a specific COPR into account when choosing how many operation points to use, and the exact contents of those operation points. See the description on COPS (Section 7.5) on how to achieve mapping between a suggested new OPID and what OPID will actually be used.

When OPID cannot be kept the same between COPN and COPR, the mapping SHALL be done using identical ID parameters (Section 8.3) in the COPS and COPN resulting from the COPR. Further details are described in the section on COPS (Section 7.5).

Since COPR references a certain COPN OPID, Version, and COPN is send unreliably and may be lost, COPN senders MUST keep at least the two last COPN Versions for each SSRC, OPID tuple and SHOULD keep at least four.

7.3.3. Timing Rules

The timing follows the rules outlined in section 3 of AVPF [RFC4585]. This notification message may be time critical and SHOULD be sent using early or immediate feedback RTCP timing, but MAY be sent using regular RTCP timing.

A typical example when regular RTCP timing can be appropriate is when the sent media stream is further restricted from what was described by the most recent COPN, which should not cause any problems in the media receivers. Similarly, it is likely appropriate to use early or immediate timing when effective media stream restrictions urgently needs to be removed, which may require media receivers to increase their resource usage.

7.4. Codec Operation Point Request

7.4.1. Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Type |       Payload Length      |      OPID     |N|   Version   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sequence No   | Codec Configuration Parameters                :
+-+-+-+-+-+-+-+-+                                               :
:                                                               :
```

Figure 11: COPR format

The COPR-specific message fields are:

Type (3 bits): Set to 1, as listed in Table 1.

OPID (8 bits): The OPID this request refers to for an existing OPID, and an arbitrarily chosen but unique value in requests for new operations points, i.e. with the N flag set.

N (1 bit): MUST be set to 0 when OPID references an existing OPID announced in a COPN received from the targeted media sender, and MUST be set to 1 otherwise.

Version (7 bits): When N flag is not set (0), referencing a specific version of the codec configuration identified by the OPID in a COPN received from the targeted media sender. Not used and MUST be set to 0 when N flag is set (1).

Sequence No (8 bits): Sequence Number. SHALL be incremented by 1 modulo 2^8 for every COPR that includes an updated set of requested codec configuration parameters described by the same

OPID and Version as was used with the previous Sequence Number. Sequence Number SHALL be kept unchanged in repetitions of this message. Initial value SHOULD be chosen randomly.

Codec Configuration Parameters (variable length): Contains zero or more TLV carrying codec configuration parameters as defined in parameter types (Section 8).

7.4.2. Semantics

This message item is sent by a media receiver wanting to control one or more codec configuration parameters of the targeted media sender. The requested values MUST stay within the media capability negotiated by other means than this specification. The available codec configuration parameters that can be controlled are listed in Section 8.

Note: COPR may be seen as a more explicit and elaborate version of the TSTR message of [RFC5104] and most of the considerations detailed there for TSTR also apply to COPR.

7.4.2.1. Sender Behavior

If at least one COPN (Section 7.3) is received for the targeted stream, the codec configuration parameters for that stream (SSRC) with defined OPID and Version are known to the COPR sender. The COPR MUST refer to the OPID and Version of the most recently received COPN (if any) for the targeted stream. Since it references a defined set of codec configuration parameters from a COPN, the COPR SHOULD only include the codec configuration parameters it wishes to change in the message, but it MAY include also unchanged codec configuration parameters.

If no COPN is received for the targeted stream, the COPR sender MUST choose an arbitrary OPID and set the N flag to indicate that the OPID does not refer to any existing operation point. In this case the Version field is not used and MUST be set to 0. The OPID value SHALL NOT be identical to any OPID from the same media source that the media receiver is aware of and has received COPN for. Since in this case no COPN reference exist, the COPR sender SHOULD include all

codec configuration parameters that it wishes to include a specific restriction for (other than the default). Note that for some codecs, some codec configuration parameters may be possible to infer from the media stream, but if the wanted restriction includes also those and lacking a describing COPN, they SHOULD anyway be included explicitly in the COPR.

Any codec configuration parameter that are not enabled by capability signaling MUST NOT be included.

A COPR sender MUST increment the SN field modulo 2^8 with every new COPR that includes any update to the codec configuration parameters (referring to a specific version of an OPID compared to the previously sent SN, as long as it does not receive any COPS (Section 7.5) with the same OPID, Version, and SN as was used in the most recently sent COPR. COPR having a later SN MUST be interpreted as replacing any COPR with identical OPID and Version but with lower SN, taking field wrap into account.

A COPR sender that did not receive any corresponding COPS, but did receive a COPN with the same OPID and with a higher Version than was used in the last COPR SHALL reconsider the COPR and MAY send an updated COPR referencing the new Version.

If the capability negotiation has established that a codec supporting scalable operation is used, and if the media receiver wishes to request that scalability is used, it MAY do so by sending multiple COPR with different OPID to the same media sender. The OPID and Version used in such request MAY be based on an existing operation point, but it MAY also indicate a desire to introduce scalability into a previously non-scalable stream by choosing a new OPID (indicated by setting the N flag). In any case, the resulting OPIDs and sub-streams are identified through use of the ID parameter (Section 8.3) in subsequent COPS and COPN. See also the description of COPS (Section 7.5).

An operation point without any codec configuration parameters MAY be used and MUST be interpreted as a request to keep the operation point unchanged. This is especially useful when modifying some but not all in a set of sub-streams.

When a COPR sender is receiving multiple operation points and wants to continue to do so, it MUST include all operation points it still wishes to receive in the COPR, also those that can be left unchanged.

An COPR MAY also describe alternative operation points that the media sender can choose from, through use of one or more ALT parameters (Section 8.2).

Since COPR references a specific COPN using SSRC, OPID and Version, a COPR sender typically needs to keep the latest Version of received COPN for each SSRC and OPID, also including the codec configuration parameters.

7.4.2.2. Media Sender Behavior

A media sender receiving a COPR SHOULD take the request into account for future encoding, but MAY also take COPR from other media receivers and other information available to the media sender into account when deciding how to change encoding properties.

A media receiver sending COPR thus cannot always expect that all parameter values of the request are fully honored, or even honored at all. It can only know that the COPR was taken into account when receiving a COPS (Section 7.5) from the media sender with a matching OPID, Version and SN.

To what extent a COPR is honored is described by the chosen codec configuration parameter values contained in a subsequent COPN message (Section 7.3) with a later (taking wraparound into account) Version than the one referred by the COPR.

7.4.3. Timing Rules

The timing follows the rules outlined in section 3 of [RFC4585]. This request message MAY be sent using Immediate, Early or Regular timing depending on the application's needs.

A COPR sender that did not receive a corresponding COPS MAY choose to retransmit the COPR, without increasing the SN.

When an RTP media receiver (SSRC) is timing out or leaves (BYE received) from the RTP session, it SHALL implicitly imply that all COPR restrictions put by that media receiver are removed.

7.5. Codec Operation Point Status

7.5.1. Message Format

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Type |      Payload Length     |     OPID      |N|   Version   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      SSRC of COPR sender                      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Sequence No   | RC  | Reason  |Codec Configuration Parameters :
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               :
    :                                                               :
```

Figure 12: COPS format

The COPS-specific message fields are:

Type (3 bits): Set to 2, as listed in Table 1.

OPID (8 bits): MUST be set identical to the same field in the COPR being reported on.

N (1 bit): MUST be set identical to the same field in the COPR being reported on.

Version (7 bits): MUST be set identical to the same field in the COPR being reported on.

SSRC of COPR sender (32 bits): MUST be set identical to the SSRC of packet sender field in the common AVPF header part of the COPR being reported on.

Sequence No (8 bits): MUST be set identical to the same field in the COPR being reported on.

RC (3 bits): Return Code. Indicates degree of success or failure of the COPR being reported on, as described in Table 2.

Reason (5 bits): Contains more detailed information on the reason for success or failure, as described in Table 3 or extensions to this specification.

Codec Configuration Parameters (variable): MAY contain an ID codec configuration parameter providing codec specific media identification of the OPID, subject to conditions outlined in the text below, or MAY be empty.

7.5.2. Semantics

The COPS message item indicates the request status related to a certain SSRC OPID tuple by listing the latest received COPR (Section 7.4) SN. It effectively informs the COPR sender that it no longer needs to resend that COPR SN (or any previous SN).

COPS indicates that the specified COPR was successfully received by the media sender targeted in the request. If the COPR suggested codec configuration parameters could be understood (Table 2), they may be taken into account, possibly together with COPR messages from other receivers and other aspects applicable to the specific media sender. The Return Code carries an indication to which extent the COPR could be honored.

Value	Meaning
0	Success
1	Partial success
2	Failure
3-6	Unassigned
7	Reserved for future extension

Table 2: Return Code Values

A Success Return Code indicates that the resulting media configuration is fully in line with the COPR.

A Partial Success Return Code indicates that the resulting media configuration is not fully in line with the COPR, but that the media sender regards the COPR to be sufficiently well represented by one or more of the existing operation points.

A Failure Return code indicates that the media sender failed to take the COPR into account, either due to some error condition or because no media stream could be created or changed to comply.

The Reason Values defined below are independent of Return Code, but all reasons may not be meaningful with all return codes. More reasons MAY be defined in extensions to this specification.

Value	Meaning
0	Success
1	Unknown OPID
2	Too many operation points
3	Request violates capability limits
4	Too old operation point version
5	Unknown parameter type
6	Parameter value too long
7	Invalid comparison type
8	One or more parameter values in the request were changed
9-31	Unassigned

Table 3: Reason Values

COPS is typically sent without any codec configuration parameters. When the N flag was set in the related COPR, a non-failing COPS MUST include an ID parameter (Section 8.3) identifying the actual sub-stream that the media sender considers applicable to the COPR. The OPID used by that sub-stream can be found through examining ID parameters of subsequent COPN from the same media source for ID values matching the one in COPS.

Senders implementing this specification MUST NOT use any other codec configuration parameter types than ID in a COPS message. The contained ID parameter points to the specific media (sub-)stream that the media sender regards as applicable to the COPR.

When a COPR receiver has received multiple COPR messages from a single COPR source with the same OPID but with several different values of Version and/or SN, and for which it has not yet sent a COPS, it SHALL only send COPS for the COPR with the Highest SN, taking field wrap of those two fields into account.

7.5.3. Timing Rules

COPS SHALL be sent at the earliest opportunity after having received a COPR, with the following exception:

> A media sender that receives a COPR with a previously received OPID, Version, and SN closely after sending a COPS for that same OPID, Version, and SN (within 2 times the longest observed round trip time, plus any AVPF-induced packet sending delays), SHOULD await a repeated COPR before scheduling another COPS transmission for that OPID, Version, and SN.

The exception is introduced to avoid unnecessary COPS transmission when there is a chance that already sent COPS or COPN may satisfy or invalidate the COPR.

7.6. Handling in Mixers and Translators

7.6.1. COPN

Any media sender, including mixers and translators, that sends RTP media marked with it's own SSRC and that implements this specification SHALL also be prepared to send COPN, even if it is not the originating media source. As a result of that, such media sender may have to send updated COPN whenever the included media sources (CSRC) changes, subject to rules laid out above (Section 7.3.2). Note that this can be achieved in different ways, for example by forwarding (possibly cached) COPN from the included CSRC when the mixer is not performing transcoding.

In cases where a mixer or translator needs to forward a COPR from one side (A) to the other (B) (as described in Section 7.6.2), the COPN sent to the A side MAY need to be delayed until the mixer or translator has received a corresponding COPN from the B side, as indicated in Figure 13 below.

```
+--------+ 1. COPR +--------+ 2. COPR +--------+
|        |-------->|        |-------->|        |
|   A    | 4. COPN | Mixer  | 3. COPN |   B    |
|        |<--------|        |<--------|        |
+--------+         +--------+         +--------+
```

Figure 13: Mixer delay of COPN

If a mixer or translator has decided to act partially (modify the media stream with respect to some parameter types, but not all) on a received COPR from the A side, and a COPN is received from the B side indicating that the current media modifications are no longer necessary, the mixer or translator SHOULD cease it's own actions that are no longer needed. It SHOULD then also issue a COPN describing the new situation to the A side, as indicated in Figure 14 below.

```
+--------+ 1. COPR +--------+         +--------+
|        |-------->|        | 2. COPR |        |
|        | 3. COPN |        |-------->|        |
|   A    |<--------| Mixer  | 4. COPN |   B    |
|        | 5. COPN |        |<--------|        |
|        |<--------|        |         |        |
+--------+         +--------+         +--------+
```

Figure 14: Mixer update of COPN

7.6.2. COPR

A mixer or media translator that implements this specification and encodes content sent to the media receiver issuing the COPR SHALL consider the request to determine if it can fulfill it by changing its own encoding parameters. A mixer encoding for multiple session participants will need to consider the joint needs of all participants when generating a COPR on its own behalf towards the media sender.

A mixer or translator able to fulfill the COPR partially MAY act on the parts it can fulfill (and SHALL then send COPS and COPN accordingly), but SHOULD anyway forward the unaltered COPR towards the media sender, since it is likely most efficient to make the necessary codec configuration parameter changes directly at the original media source.

A media translator that does not act on COP messages will forward them unaltered, according to normal translator rules.

7.6.3. COPS

A mixer or media translator that implements this specification, encoding content sent to media receivers and that acts on COPR SHALL also report using COPS, just like any other media sender. An RTP translator not knowing or acting on COPR will forward all COP messages unaltered, according to normal RTP translator rules.

8. Parameter Types

This section defines the general codec configuration parameter (CCP) TLV format. Then a number of different parameter formats are defined. It is expected that a number of additional CCPs will be defined in the future as the needs of different codecs are explored or developed.

8.1. Parameter Format

COP message items MAY contain one or more codec configuration parameters, encoded in TLV (Type-Length-Value) format, which SHOULD then be interpreted as simultaneously applicable to the defined operation point. Parameter values MUST be byte-aligned.


```
    0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | ParamType       | C |  Length    |                                |
```

```
   +--------------+---+-----------+                                |
   |                                                                |
   /                        Parameter Value                         /
   /                                              +--------------+
   |                                              |
   +----------------------------------------------+
```

Figure 15: Codec parameter format

ParamType (8 bits): The codec configuration parameter type, encoded as defined in Table 4 and possible extensions to this specification. A parameter with an unknown ParamType SHALL be ignored on reception in a COPN and SHALL either be reported as unknown in COPS or be ignored when received in COPR.

C (2 bits): Comparison Type, encoded as defined in Table 5, unless specified otherwise by individual ParamType definitions. The Comparison Type specifies what type of restriction the codec configuration parameter value expresses and how it should be compared to other codec configuration parameter values of the same ParamType.

Exact: The parameter value is an exact value, and no other values are acceptable. MUST NOT be used together with any other Comparison Types for the same ParamType.

Minimum: The parameter value is an inclusive minimum restriction. MAY be used together with Maximum and/or Target Comparison Types for the same ParamType. If no minimum restriction is specified, no specific minimum restriction exist.

Maximum: The parameter value is an inclusive maximum restriction. MAY be used together with Minimum and/or Target Comparison Types for the same ParamType. If no maximum restriction is specified, no specific maximum restriction exist.

Target: The parameter value is a preferred target value, but other values within a specified range are acceptable. This type MUST be used together with at least one of Minimum and Maximum Comparison Types for the same ParamType. If no target is specified, no specific preference exist.

Length (6 bits): The parameter value Length in bytes, excluding the ParamType and the Length field itself. A Length of 0 indicates that the parameter has no value, effectively constituting a wild-carded parameter that can take on any value (expresses no specific restriction). This is also the RECOMMENDED way to explicitly remove a previously effective restriction.

Parameter Value (variable length): The actual parameter value, encoded in a format defined by the specific ParamType definition.

The meaning of Multiple codec configuration parameters with the same ParamType and the same Comparison Type included as part of the same operation point is undefined and SHALL NOT be used.

A codec configuration parameter that is encoded in a way (including incorrectly) that cannot be interpreted by the receiver SHALL be ignored.

The below parameters encoded as signed or unsigned integers uses a variable size representation in the value field. It is RECOMMENDED to only include the minimal number of bytes necessary to represent the value that is to be included in the parameter TLV. The length field in the parameter TLV will explicitly indicate how many bytes are present in the value field. All parameters using a variable size representation of their value MUST define the maximum number of bytes possible to include in the value field.

The ParamType values and the SDP tags (see Section 9) for the codec configuration parameter types defined in this specification are listed below.

Value	Meaning	Tag
0	ALT	alt
1	ID	id
2	Payload Type	pt
3	Bitrate	bitrate
4	Token Bucket Size	token-bucket
5	Framerate	framerate
6	Horizontal Pixels	hor-size
7	Vertical Pixels	ver-size
8	Sample Aspect Ratio	sar
9	Picture Aspect Ratio	par
10	Channels	channels
11	Sampling Rate	sampling
12	Maximum RTP Packet Size	max-rtp-size
13	Maximum RTP Packet Rate	max-rtp-rate
14	Frame Aggregation	aggregate
15-254	Undefined	
255	Reserved for future extension	

Table 4: Parameter Type Values

The values of the defined parameter value comparison type are listed below.

Value	Meaning
0	Exact
1	Minimum
2	Maximum
3	Target

Table 5: Comparison Type Values

The following sub-sections describe the syntax and semantics of the different codec configuration parameter types defined in this specification.

Unless explicitly specified in the sub-sections below, or in extensions to this specification, all parameter type values are binary encoded unsigned integers, most significant byte first (for multi-byte values).

8.2. ALT

This codec parameter type is a special parameter, separating the codec configuration parameters preceding it from the ones that follow into two separate, alternative operation points.

Type Value: 0

Tag: alt

Unit: Not applicable.

Semantics: A special parameter expressing an "alternative" relation between the parameters preceding it and the parameters following it. This SHOULD be interpreted as describing two alternate operation points where one and only one SHALL be chosen, with the operation point preceding ALT in the parameter list being preferred. Multiple ALT parameters MAY be used in the same parameter list, in which case each set of parameters to evaluate can be either before the first ALT parameter, between two ALT parameters, or after the last ALT parameter. Evaluating from the top of the list and obeying the above preference rule, the first acceptable set of parameters (not containing any ALT parameter) is the one to choose.

Encoding: Not applicable.

Media Types: All.

Value Restrictions: MUST be used with the Length field set to 0. Two ALT parameters MUST be separated by at least one parameter other than ALT.

Default Value: Not applicable.

Comparison Types: MUST be set to 0.

Note:

8.3. ID

This codec parameter type is a special parameter that enables codec specific identification of sub-streams, for example when there are multiple sub-streams in a single SSRC. It can also be used to reference OPID, when the used codec does not support or use sub-streams. When used, it SHALL be listed first among the codec parameters used to describe the sub-stream.

Type Value: 1

Tag: id

Unit: Not applicable.

Semantics: A special parameter describing the, possibly codec specific, media identification for the OPID.

Encoding: If used with non-scalable encoding, it MUST contain an OPID (Section 7.2.1). If used with scalable encoding, this codec specific encoding MUST be defined by Section 10. It MUST be defined to occupy an integer number of bytes, where all bits in the bytes are defined as part of the format.

Media Types: All.

Value Restrictions: If used with non-scalable encoding, any OPID restrictions apply. If used with scalable encoding, any restrictions MUST be defined by the definition of the codec specific sub-stream identification definition (Section 10).

Default Value: Not set.

Comparison Types: MUST be set to 0.

Note: MAY be used whenever there is a need to identify an operation point in codec native format, or when there is a need to map that against an OPID.

8.4. Payload Type

Type Value: 2

Tag: pt

Unit: Not applicable.

Semantics: Referencing the RTP Payload Type to use for the OPID.

Encoding: The least significant 7 bits MUST use the same encoding as the RTP Payload Type field in the RTP header. The most significant bit MUST be set to 0.

Media Types: All.

Value Restrictions: The same restrictions valid for RTP Payload Type apply, i.e. 7-bit values 0-127. MUST be represented by a single byte in the value field.

Default Value: Not set.

Comparison Types: MUST be set to 0.

Note: MAY be used whenever there is a need to specify codec configuration parameters valid only for a certain RTP Payload Type. What media type, codec and possible parameters that are described by the RTP Payload Type is outside the scope of this specification, but is typically defined in capability or call setup signaling, for example SDP.

8.5. Bitrate

Type Value: 3

Tag: bitrate

Unit: Bits per second.

Semantics: Media level per second average media bitrate, excluding IP/UDP/RTP overhead, but including RTP payload headers (similar to b=TIAS from SDP signaling [RFC3890]), rounded up to the closest integer.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: All.

Value Restrictions: A value of 0 MAY be used. The largest value allowed is what is possible to represent in a 64-bit unsigned integer value, i.e. a value between 0 and 18,446,744,073,709,551,615.

Default Value: Maximum value computed from capability or call setup signaling, e.g. b= parameter from SDP. Note that it is often not possible to achieve more than a rough estimation from such computation.

Comparison Types: All. The Exact comparison type is meaningful only for streams that are able to produce a set of predictable (e.g. constant) packet sizes, sent at predictable (e.g. constant) inter-packet intervals.

Note: This parameter used with a maximum comparison type parameter is significantly similar to CCM Temporary Maximum Media Bit Rate (TMMBR). When being used with a maximum or exact comparison type value of 0, it is also significantly similar to PAUSE [I-D.westerlund-avtext-rtp-stream-pause]. Compared to those, this parameter conveys significant extra information through the relation to other parameters applied to the same operation point, as well as the possibility to express other restrictions than a maximum limit. When CCM TMMBR is supported in addition to this specification, the Bitrate parameters from all operation points within each SSRC should be considered and CCM TMMBR messages SHOULD be sent for those SSRC that are found to be in the bounding set (see CCM [RFC5104], section 3.5.4.2). When PAUSE is supported in addition to this specification, the Bitrate parameters from all operation points within each SSRC should be considered and CCM PAUSE messages SHOULD be sent for those SSRC that contain only operation points that are limited by a Bitrate maximum value of 0. There only difference between setting the bitrate to 0 and removing the OPID entirely is that increasing the bitrate from 0 just requires the bitrate parameter to be sent again, while re-activating a removed OPID requires it to be fully re-defined including all other parameters that are included in the OPID.

8.6. Token Bucket Size

Type Value: 4

Tag: token-bucket

Unit: Bytes.

Semantics: Media level token bucket [RFC2212] size excluding IP/UDP/RTP overhead, but including RTP payload headers, describing the bitrate variability over time as described in [I-D.westerlund-mmusic-sdp-bw-attribute]. This parameter can be combined with the parameter bitrate (Section 8.5) (above) to provide token bucket fill rate plus bucket size for a complete token bucket model.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: All.

Value Restrictions: A value of 0 is generally not meaningful and SHOULD NOT be used. Values that can be represented using a 32-bit unsigned integer, i.e. 0 to 4,294,967,295.

Default Value: 4096 bytes.

Comparison Types: Maximum, Target.

Note: Changing the token bucket size does not imply changing the average bitrate, it just changes the acceptable average bitrate variation over time.

8.7. Framerate

Type Value: 5

Tag: framerate

Unit: 100th of a Hz. This definition allows e.g. distinguishing between video encoded at 30 Hz (two-byte value 3000) and 29.97 Hz (two-byte value 2997). It also allows for high speed video cameras, like 1000 Hz (three-byte value 100000), and slow-scan down to one frame every 100 seconds (one-byte value 1).

Semantics: The number of media frames to render per second.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: Mainly intended for video and timed image media types, but MAY be used also for other media types.

Value Restrictions: A value of 0 MAY be used, meaning single-frame, request based encoding (request procedure is out of scope for this specification). Values that can be represented using a 32-bit unsigned integer, i.e. 0 to 42,949,672.95 Hz.

Default Value: Maximum allowed by call setup and/or capability signaling, e.g. a=framerate parameter from SDP [RFC4566], or codec-specific configuration.

Comparison Types: All.

Note: A media frame is typically a set of semantically grouped samples, e.g. the relation that a video image has to its individual pixels, or the relation that an audio frame has to individual audio samples. The value applies to encoded media framerate, not the packet rate (Section 8.15) that may also be changed as a result of different Frame Aggregation (Section 8.16). When the COP end-point also makes use of CCM [RFC5104] TSTR/TSTN, COPN with this parameter MAY be used in combination with TSTN to explicitly indicate what framerate setting the TSTR resulted in, making it possible for the TSTR sender to adjust the used, relative TSTR scale to more closely match what framerate was actually received.

8.8. Horizontal Pixels

Type Value: 6

Tag: hor-size

Unit: Pixels.

Semantics: Horizontal image size.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: Video and image.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used.

Default Value: Maximum allowed by call setup and/or capability signaling. Values that can be represented using a 32-bit unsigned integer, i.e. 1 to 4,294,967,295.

Comparison Types: All.

Note: The pixel and picture aspect ratios cannot be changed with this parameter. Video encoders can typically describe both pixel and picture aspect ratios as part of the encoded media stream. If the COP end-point supports imageattr signaling [RFC6236], values for this parameter SHOULD be chosen only among the negotiated set in the SDP, and should be done so both for the media receiving COPR sender and the media sending COPN sender, according to imageattr values for the affected media stream direction.

8.9. Vertical Pixels

Type Value: 7

Tag: ver-size

Unit: Pixels.

Semantics: Vertical image size.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: Video and image.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used. Values that can be represented using a 32-bit unsigned integer, i.e. 1 to 4,294,967,295.

Default Value: Maximum allowed by call setup and/or capability signaling.

Comparison Types: All.

Note: See Note in Section 8.8.

8.10. Sample Aspect Ratio

Type Value: 8

Tag: sar

Unit: Unit-less value pair.

Semantics: The ratio between the intended horizontal distance between the columns and the intended vertical distance between the rows of the luma sample array in a frame, similar to what is defined in [H241].

Encoding: Two binary encoded, unsigned 8-bit integers in order horizontal, vertical.

Media Types: Video and image.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used as value in either the horizontal or vertical component. Component values that can be represented using an 8-bit unsigned integer, i.e. 1 to 255.

Default Value: The same as defined in [H241] when there is no explicit indication, based on image size.

Comparison Types: All.

Note: If the COP end-point supports imageattr signaling [RFC6236], values for this parameter SHOULD be chosen only among the negotiated set in the SDP, and should be done so both for the media receiving COPR sender and the media sending COPN sender, according to imageattr values for the affected media stream direction.

8.11. Picture Aspect Ratio

Type Value: 9

Tag: par

Unit: Unit-less value pair.

Semantics: The ratio between the intended horizontal width and the intended vertical height of a displayed picture, similar to what is defined in [H241].

Encoding: Two binary encoded, unsigned 8-bit integers in order horizontal, vertical.

Media Types: Video and image.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used as value in either the horizontal or vertical component. Component values that can be represented using an 8-bit unsigned integer, i.e. 1 to 255.

Default Value: The same as defined in [H241] when there is no explicit indication, based on image size.

Comparison Types: All.

Note: If the COP end-point supports imageattr signaling [RFC6236], values for this parameter SHOULD be chosen only among the negotiated set in the SDP, and should be done so both for the media receiving COPR sender and the media sending COPN sender, according to imageattr values for the affected media stream direction.

8.12. Channels

Type Value: 10

Tag: channels

Unit: Unit-less.

Semantics: The number of media channels.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: All.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used. Values that can be represented using a 16-bit unsigned integer, i.e. 1 to 65,535.

Default Value: Taken from call setup or capability signaling, or 1 if no other value is available.

Comparison Types: All.

Note: This codec configuration parameter SHOULD NOT be used if the capability negotiation did not establish that suitable multi-channel coding is supported by both ends. For audio, the interpretation and spatial mapping SHALL follow the one for the indicated payload format. If no such channel mapping is defined in the payload format, and if not specifically signalled by other means, e.g. SDP, the channel configurations defined in [RFC3551] SHALL be used. For video, it SHALL be interpreted as the number of views in multiview coding, where the number 2 SHOULD represent stereo (3D) coding, unless negotiated otherwise by means outside of this specification, e.g. SDP. If multiple payload formats are defined and if those do not share channel configurations, the Payload Type parameter (Section 8.4) MUST be included as one of the parameters for the OPID.

8.13. Sampling Rate

Type Value: 11

Tag: sampling

Unit: Hz.

Semantics: Frequency of the media sampling clock in Hz, as input to the codec, per channel (Section 8.12).

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: Mainly intended for audio media, but MAY be used for other media types.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used. Values that can be represented using a 32-bit unsigned integer, i.e. 1 to 4,294,967,295.

Default Value: Taken from call setup or capability signaling, e.g. RTP TS rate from SDP m-line.

Comparison Types: All.

Note: The value refers to the media sample clock, not the media Framerate (Section 8.7). It does not specify any codec-internal up- or down-sampling that may take place as part of the encoding process. If multiple channels (Section 8.12) are used and different channels use different sampling rates, then this parameter MUST NOT be used unless there is a known sampling rate relationship and an ordering between the channels, in which case the specified sampling rate value SHALL be taken as applicable to the first channel of the ordered set. The relationship may e.g. be known implicitly by each party through some specification, or be negotiated using other means than this specification. Typically only a limited subset of sampling frequencies makes sense to the media encoder, and sometimes it is not possible to change at all. For video, the sampling rate is very closely connected to the image horizontal (Section 8.8), vertical (Section 8.9) resolution, and framerate (Section 8.7), which are more explicit and meaningful and SHOULD therefore be used instead. For audio, changing sampling rate may require changing codec and thus changing RTP payload type. The actual media sampling rate may not be identical to the sampling rate specified for RTP Time Stamps for that RTP Payload Type. E.g. almost all video codecs use only 90 000 Hz sampling clock for RTP Time Stamps, while the actual pixel sampling clock is typically in the range from a few to several hundred MHz. Also some recent audio codecs use an RTP Time Stamp rate that differ from the actual media sampling rate. Aspects related to mid-stream changes of RTP Time Stamp rate is described in [I-D.ietf-avtext-multiple-clock-rates].

8.14. Maximum RTP Packet Size

Type Value: 12

Tag: max-rtp-size

Unit: Bytes.

Semantics: The maximum size of an RTP packet, including the RTP header but excluding lower layers.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: All.

Value Restrictions: The meaning of a value less than the size of the RTP header (12 bytes for current RTP specification [RFC3550]) is not defined and SHOULD NOT be used. Values that can be represented using a 32-bit unsigned integer, i.e. 0 to 4,294,967,295.

Default Value: 1400 bytes for IPv4, 1280 bytes for IPv6 or if IP version cannot be determined.

Comparison Types: Maximum.

Note: The parameter should typically be used to adapt encoding to a known or assumed MTU limitation, and MAY be used to assist MTU path discovery in point-to-point as well as in RTP mixer or translator topologies.

8.15. Maximum RTP Packet Rate

Type Value: 13

Tag: max-rtp-rate

Unit: RTP packets per second.

Semantics: Maximum number of RTP packets per second, calculated or estimated as the largest value appearing during a one-second sliding window, similar to the definition of "maxprate" [RFC3890].

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: All.

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used. Values that can be represented using a 32-bit unsigned integer, i.e. 1 to 4,294,967,295.

Default Value: Not set.

Comparison Types: Maximum.

Note: The parameter should typically be used to adapt encoding on a network that is packet rate rather than bitrate limited, if such property is known. This codec configuration parameter MUST NOT exceed any negotiated "maxprate" [RFC3890] value, if present.

8.16. Application Data Unit Aggregation

Type Value: 14

Tag: aggregate

Unit: Milliseconds.

Semantics: The amount of non-redundant application data unit (ADU) representing different RTP Time Stamps that should be included in the RTP payload, henceforth in this specification called an "ADU aggregate". An ADU aggregation value of 1 is equivalent to no aggregation.

Encoding: Binary encoded unsigned integer, most significant byte first.

Media Types: Mainly intended for audio, but MAY be used also for other media, e.g. Real-Time Text [RFC4103].

Value Restrictions: The meaning of the value 0 is not defined and SHALL NOT be used. Values that can be represented using a 16-bit unsigned integer, i.e. 1 to 65,535.

Value Default Value: 1.

Comparison Types: All.

Note: To use this parameter, there MUST exist a defined way of including multiple ADUs into the same RTP payload for the used RTP Payload Type. There MUST also exist a known internal timing relationship between individual ADUs within the RTP payload for the used RTP Payload Type. Some payload formats (typically video) do not allow multiple ADUs (representing different sampling times) in the RTP payload. This codec configuration parameter SHOULD NOT be used unless the "maxprate" [RFC3890] and/or "ptime" parameters are included in the SDP. The requested ADU aggregation level MUST NOT cause exceeding the negotiated "maxprate" value, if present, and SHOULD NOT exceed the negotiated "ptime" value, if present. The requested frame aggregation level MUST NOT be in conflict with any Maximum RTP Packet Size (Section 8.14) or Maximum RTP Packet Rate (Section 8.15) parameters. The packet rate that may result from different frame aggregation values is related to, but semantically not the same as, media Framerate (Section 8.7).

9. SDP Extensions

As described in [RFC4585] and [RFC5104], the rtcp-fb attribute may be used to negotiate capability to handle specific AVPF commands and indications, and specifically the "ccm" feedback value is used for codec control. All rules defined there related to use of "rtcp-fb" and "ccm" also apply to the new feedback message defined in this specification.

9.1. Extension of the rtcp-fb Attribute

In this document, a new "ccm" rtcp-fb-ccm-param is defined, according to the method of extension described in [RFC5104]:

- o "cop" indicates support for all COP message items defined in this specification, and one or more of the codec configuration parameters defined in this specification

The ABNF [RFC5234] for the new rtcp-fb-ccm-param is:

```
rtcp-fb-ccm-param =/ SP "cop" 1*rtcp-fb-ccm-cop-param
; rtcp-fb-ccm-param defined in [RFC5104]

rtcp-fb-ccm-cop-param = SP "alt"
                      / SP "id"
                      / SP "pt"
                      / SP "bitrate"
                      / SP "token-bucket"
                      / SP "framerate"
                      / SP "hor-size"
                      / SP "ver-size"
                      / SP "sar"
                      / SP "par"
                      / SP "channels"
                      / SP "sampling"
                      / SP "max-rtp-size"
                      / SP "max-rtp-rate"
                      / SP "aggregate"
                      / SP token ; for future extensions
; token defined in [RFC4566]
```

Figure 16: ABNF for cop

Token values for rtcp-fb-ccm-cop-param are defined in Table 4. Their semantics are described in Section 8.

Supported parameter types are indicated by including one or more rtcp-fb-ccm-cop-param.

9.2. Offer/Answer Usage

The usage of Offer/Answer [RFC3264] in this specification inherits all applicable usage defined in [RFC5104].

In order to announce support, and willingness to use, the CCM "cop" feedback message, an offerer or answerer SHALL indicate that capability through the extended SDP rtcp-fb attribute, defined in Section 9.1. The offerer or answerer MUST include a list of the parameter types that it is willing to receive.

If an SDP offer does not indicate support of the CCM "cop" feedback message, the answerer MUST NOT indicate support in the associated SDP answer.

The answerer MAY add and/or remove parameter types that were not present in the associated SDP offer. If the answerer adds parameter types to the SDP answer, it MUST be able to receive such messages, but the answerer MUST NOT send such messages towards the offerer.

If an SDP answer does not indicate support of the CCM "cop" feedback message, the offerer MUST NOT send such messages towards the answerer.

The offerer and the answerer SHOULD NOT send any parameter types that the remote party did not indicate receive support for. As described in Section 8, a parameter with an unknown ParamType SHALL be ignored on reception in a COPN and SHALL either be reported as unknown in COPS or be ignored when received in COPR.

Entities MUST list all supported parameter types in every subsequent SDP offer or answer associated with the session. If a parameter type is not listed, it is an indication that the offerer or answerer is no longer willing to receive such messages within the session.

9.3. Declarative Usage

Declarative use of the CCM "cop" does not differ from the Offer/Answer usage.

10. Codec Sub-Stream Identification

The defined mechanism is not bound to a specific codec. It uses the main characteristics of a chosen set of media types, including audio and video. To what extent this mechanism can be applied depends on which specific codec is used.

When using a codec that can produce separate sub-streams within a single SSRC, those sub-streams can only be referred with a COP OPID if there is a defined relation to the codec-specific sub-stream identification. This is accomplished in this specification by defining an ID parameter format using codec-specific sub-stream identification for each such codec.

If such sub-streams have dependencies, the OPID describes the characteristics of the sub-stream including all it's dependencies, but excluding any sub-streams that are dependent on this sub-stream. The sub-stream identification describes a single, payload specific node in a dependency tree, and does in general not include any identification of the sub-streams it depends on, or the dependency structure between sub-streams. Any dependency structure must thus be described by the media stream payload format and is out of scope for this specification.

This section contains ID parameter format definitions for a few selected codecs. The format definitions MUST use an integer number of bytes and MUST define all bits in those bytes. Note, the ID

parameter is interpreted in the context of a given SSRC and a specific RTP payload type.

Extensions to this specification MAY add more codec-specific definitions than the ones described in the sub-sections below. Such definitions made in extensions to this specification SHOULD be considered as an integrated part of this section, with respect to usage with other mechanisms defined in this specification.

10.1. H.264 AVC

Some non-scalable video codecs such as H.264 AVC [H264] and corresponding RTP payload format [RFC6184] can accomplish simultaneous encoding of multiple operation points. H.264 AVC can encode a video stream using limited-reference and non-reference frames such that it enables limited temporal scalability, by use of the nal_ref_id syntax element.

The ID parameter type is defined below:

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|  Reserved | N |
+-+-+-+-+-+-+-+-+
```

Figure 17: ID definition for AVC

Reserved (6 bits): Reserved. SHALL be set to 0 by senders and SHALL be ignored by receivers implementing this specification. MAY be defined differently by extensions to this specification.

N (2 bits): SHALL be identical to the highest value of the nal_ref_idc H.264 NAL header syntax element valid for the sub-bitstream described by this OPID, with the exception of nal_ref_idc value 3 that is valid for and is part of all sub-bitstreams.

10.2. H.264 SVC

This document specifies the usage of multiple, simultaneous codec operation points and therefore maps well to scalable video coding. Scalable video coding such as H.264 SVC (Annex G) [H264] uses three scalability dimensions: temporal, spatial, and quality. It also includes the possibility to use redundant encodings and priority among sub-streams.

The ID SHALL be considered describing an SVC sub-bitstream, which is defined in G.3.59 of H.264 [H264] and corresponding RTP payload

format [RFC6190]. For use with H.264 SVC, ID SHALL be constructed as defined below:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|    PID    |      RPC      | DID |  QID  | TID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: ID definition for SVC

R (1 bit): Reserved. SHALL be set to 0 by senders and SHALL be ignored by receivers implementing this specification. MAY be defined differently by extensions to this specification.

PID (6 bits): SHALL be identical to an unsigned binary integer representation of the priority_id H.264 syntax element valid for the sub-bitstream described by this OPID. SHALL be set to 0 if no priority_id is available.

RPC (7 bits): SHALL be identical to an unsigned binary integer representation of the redundant_pic_cnt H.264 syntax element valid for the sub-bitstream described by this OPID. SHALL be set to 0 if no redundant_pic_cnt is available.

DID (3 bits): SHALL be identical to the dependency_id H.264 syntax element valid for the sub-bitstream described by this OPID.

QID (4 bits): SHALL be identical to the quality_id H.264 syntax element valid for the sub-bitstream described by this OPID.

TID (3 bits): SHALL be identical to the temporal_id H.264 syntax element valid for the sub-bitstream described by this OPID

11. Examples

COP messages are binary encoded. However, in the following examples, all COP messages are for clarity listed in symbolic, pseudo-code form, where only COP message fields of interest to the example are included, along with the COP parameters.

11.1. SDP Offer/Answer

The SDP capabilities for COP are defined as receiver capabilities, meaning that there is no explicit indication what COP messages an endpoint will use in the send direction. It is however reasonable to expect that an endpoint can also send the same messages that it can understand and act on when received. This is assumed in all the SDP

examples below, but note that symmetric COP capabilities is not a requirement.

The example below shows an SDP Offer, where support of CCM "cop" message is announced for the video codecs.

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example
s=-
c=IN IP4 host.atlanta.example
t=0 0
m=audio 50000 RTP/AVP 0 8 97
b=AS:80
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
```

```
m=video 50010 RTP/AVPF 31 32
b=AS:600
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
a=rtcp-fb:31 ccm cop framerate bitrate token-rate
a=rtcp-fb:32 ccm cop hor-size ver-size framerate bitrate \
             token-rate
```

Figure 19: SDP offer (COP support indicated)

Note that the offer contains two different video payload types, and that the COP parameters differ between them, meaning that the possibility for codec configuration also differ. In this case, the MPEG-1 codec can control both framerate and image size, but for H.261 only the framerate can be controlled.

In the SDP Answer below, responding to the above offer, the answerer supports CCM "cop" messages.

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.example
s=-
c=IN IP4 host.biloxi.example
t=0 0
m=audio 52000 RTP/AVP 0
b=AS:80
a=rtpmap:0 PCMU/8000
m=video 52100 RTP/AVPF 32
b=AS:600
a=rtpmap:32 MPV/90000
a=rtcp-fb:32 ccm cop hor-size ver-size framerate bitrate \
             token-rate packet-size
```

Figure 20: SDP answer (COP support indicated)

Note that the answerer indicates support for more parameter types than the offerer.

Below is another SDP Answer, also responding to the same offer above, where the answerer does not support "cop".

```
v=0
o=bob 2808844564 2808844564 IN IP4 host.biloxi.example
s=-
```

```
c=IN IP4 host.biloxi.example
t=0 0
m=audio 52000 RTP/AVP 0
b=AS:80
a=rtpmap:0 PCMU/8000
m=video 52100 RTP/AVPF 32
b=AS:600
a=rtpmap:32 MPV/90000
```

Figure 21: SDP answer (COP support not indicated)

11.2. Dynamic Video Re-sizing

In this example, two COP-enabled endpoints communicate in an audio/video session. The receiving endpoint has a graphical user interface that can be dynamically changed by the user. This user interaction includes the ability to change the size of the receiving video window, which is also indicated in the previous SDP example (Section 11.1).

At some point during the established communication, a notification about current video stream codec operation point is sent to the resizable window endpoint that receives the video stream.

```
COPN {SSRC:123456, OPID:123, Version:5,
      bitrate(max):325000,
      token-bucket(exact):1000,
      framerate(exact):15,
      hor-size(exact):320,
      ver-size(exact):240}
```

Figure 22: COPN for QVGA 15 Hz

Some time later the user of the resizable window endpoint reduces the size of the video window. As a result of the resize operation, the video window can no longer make full use of the received video resolution, wasting bandwidth and decoder processing resources. The resizable window endpoint thus decides to notify the video stream sender about the changed conditions by sending a request for a video stream of smaller size:


```
COPR {SSRC:123456, OPID:123, Version:5,
```

```
                     hor-size(target):243,
                     ver-size(target):185}
```

Figure 23: COPR for 243x185

The COPR refers to the previously received COPN with the same OPID and Version, and thus need only list parameters that need be changed. The request could arguably contain also other parameters that are potentially affected by the spatial resolution, such as the bitrate, but that can be omitted since the media sender is not slaved to the request but is allowed to make it's own decisions based on the request.

The request sender has chosen to use target type values instead of an exact value for the horizontal and vertical sizes, which can be interpreted as "anything sufficiently similar is acceptable". The target values is in this example chosen to correspond exactly to the resized video display area. Many video coding algorithms operate most efficiently when the image size is some even multiple, and this way of expressing the request explicitly leaves room for the media sender to take such aspect into account.

The media sender (COPR receiver) responds with the following:

```
   COPS {SSRC:123456, OPID:123, Version:5,
         Partial Success,
         One or more parameter values in the request were changed}

   COPN {SSRC:123456, OPID:123, Version:6,
         bitrate(max):240000,
         token-bucket(exact):1000,
         framerate(exact):15,
         hor-size(exact):240,
         ver-size(exact):176}
```

Figure 24: COPS and COPN for partial success

It can be noted that the updated COPN (version 6) indicates that the media sender has, in addition to reducing the video horizontal and vertical size, chosen to also reduce the bitrate. This bitrate reduction was not in the request, but is a reasonable decision taken by the media sender. It can also be seen that the horizontal and vertical sizes are not chosen identical to the request, but is in fact adjusted to be even multiples of 16, which is a local restriction of the fictitious video encoder in this example. To handle the mismatch of the request and the resulting video stream, the video receiver can perform some local action such as for example automatic readjustment of the resized window, image scaling (possibly combined with cropping), or padding.

11.3. Illegal Request

In this example, the sent request is asking the media sender to go beyond what is negotiated in the SDP. The SDP Offer below indicates to use video with H.264 Constrained Baseline Profile at level 1.1.

```
v=0
o=alice 2893746526 2893746526 IN IP4 host.atlanta.example
s=-
c=IN IP4 host.atlanta.example
t=0 0
m=audio 49160 RTP/AVP 96
b=AS:80
a=rtpmap:96 G722/16000
m=video 51920 RTP/AVPF 97
b=AS:200
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42e00b
a=rtcp-fb:97 ccm cop framerate bitrate token-rate
```

Figure 25: SDP offer with H.264 level 1.1

Assuming this offer is accepted and that the answerer also supports COP, further assume that this COP message exchange occurs at some time during the established communication:

```
     Media Sender                      Media Receiver
     ------------                      --------------

     COPN {SSRC:9876, OPID:67,      ->
           Version:2,
           bitrate(exact):190000,
           token-bucket(exact):500,
           framerate(exact):10,
           hor-size(exact):320,
           ver-size(exact):240}

                                    <-  COPR {SSRC:9876, OPID:67,
                                              Version:2,
                                              framerate(exact):10,
                                              hor-size(exact):352,
                                              ver-size(exact):288}

     COPS {SSRC:9876, OPID:67,      ->
           Version:2,
           Failure,
           Request violates capability limits}
```

Figure 26: COP message exchange indicating failure

The failure above is due to a combination of frame size and frame rate that exceeds H.264 level 1.1, which would thus exceed the limits established by SDP Offer/Answer. The maximum permitted framerate for 352x288 pixels (CIF) is 7.6 Hz for H.264 level 1.1, as defined in Annex A of [H264].

11.4. Reference Response to Modification of Scalable Layer

When scalable coding is used, each layer correspond to a codec operation point. A media receiver can thus target a request towards a single layer. Assume a video encoding with three framerate layers, announced in a (multiple operation point) notification as:

```
                 COPN {SSRC:9876, OPID:67, Version:2, ID:2
                       bitrate(exact):190000,
                       token-bucket(exact):500,
                       framerate(exact):10,
                       hor-size(exact):320,
                       ver-size(exact):240}

                 COPN {SSRC:9876, OPID:73, Version:1,
                       bitrate(exact):350000, ID:1
                       token-bucket(exact):600,
                       framerate(exact):30,
                       hor-size(exact):320,
                       ver-size(exact):240}

                 COPN {SSRC:9876, OPID:95, Version:5, ID:0
                       bitrate(exact):400000,
                       token-bucket(exact):800,
                       framerate(exact):60,
                       hor-size(exact):320,
                       ver-size(exact):240}
```

Figure 27: COPN indicating three framerate layers

Assume further that the media receiver is not pleased with the low framerate of OPID 67, wanting to increase it from 10 Hz to 25-30 Hz. Note that the media receiver still wants to receive the other layers unchanged, not remove them, and thus has to explicitly indicate this by including them without parameters.

```
                    COPR {SSRC:9876, OPID:67, Version:2,
                          framerate(greater):25,
                          framerate(less):30}
```

```
                 COPR {SSRC:9876, OPID:73, Version:1}

                 COPR {SSRC:9876, OPID:95, Version:5}
```

Figure 28: COPR requesting to change one layer

The media sender decides it cannot meet the request for OPID 67, but instead considers (an unmodified) OPID 73 (with ID 1) to be a sufficiently good match:


```
   COPS {SSRC:9876, OPID:67, Version:2,
         Partial Success,
         One or more parameter values in the request were changed,
         ID:1}

   (COPN for the other two OPIDs omitted here for brevity)

   COPN {OSSRC:9876, OPID:73, Version:1, ID:1
         bitrate(exact):350000,
         token-bucket(exact):600,
         framerate(exact):30,
         hor-size(exact):320,
         ver-size(exact):240}
```

Figure 29: COPS and COPN with layer modification partial success

The COPS indicates partial success and uses the ID number to refer another OPID, describing the best compromise that can currently be used to meet the request. COPS does not contain the referred OPID, but ID should be defined in a codec-specific way that makes it possible to identify the layer directly in the media stream. If the corresponding OPID is needed, for example to attempt another request targeting that, it can be found by searching the active set of COPN for matching ID values.

11.5. Successful Request to Add Codec Operation Point

In this example, the media receiver is receiving a non-scalable stream from a codec that can support scalability, and wishes to add a scalability layer. Assume the existing OPID from the media sender is announced as:

```
COPN {SSRC:3492, OPID:4, Version:2,
      bitrate(exact):350000,
      token-bucket(exact):600,
      framerate(exact):30,
      hor-size(exact):320,
      ver-size(exact):240}
```

Figure 30: COPN with single operation point

The media receiver constructs a request for multiple streams by including multiple requests for different OPID. Since the new stream does not exist, it has no OPID from the media sender and the receiver chooses a random value as reference and indicates that it is a new, temporary OPID. The request for the new stream includes all parameters that the media receiver has an opinion on, and leaves the other parameters to be chosen by the media sender. In this case it is a request for identical frame size and doubled framerate.


```
COPR {SSRC:3492, OPID:4, Version:2}

COPR {SSRC:3492, OPID:237, New, Version:0,
      framerate(exact):60,
      hor-size(exact):320,
      ver-size(exact):240}
```

Figure 31: COPR requesting to add operation point

The media sender decides it can start layered encoding with the requested parameters. The status response to the new OPID contains a reference to an ID that is included as part of the matching, subsequent COPN. Note that since both the original and the new streams are now part of a scalable set, they must both be identified with ID parameters to be able to distinguish between them. The media sender has chosen an OPID for the new stream in the COPN, which need not be identical to the temporary one in the request, but the new stream can anyway be uniquely identified through the ID that is announced in both the COPS and COPN.

Note that since the ID has a defined relation to the media sub-stream identification, decoding of that new sub-stream can start immediately after receiving the COPS. It may however not be possible to describe the new stream in COP parameter terms until the COPN is received (depending on COP parameter visibility directly in the media stream).

```
              COPS {SSRC:3492, OPID:4, Version:2,
                    Success, Success,
                    ID:1}

              COPS {SSRC:3492, OPID:237, New, Version:0,
                    Success, Success,
                    ID:0}

              COPN {SSRC:3492, OPID:4, Version:2, ID:1,
                    bitrate(exact):350000,
                    token-bucket(exact):600,
                    framerate(exact):30,
                    hor-size(exact):320,
                    ver-size(exact):240}

              COPN {SSRC:3492, OPID:9, Version:0, ID:0,
                    bitrate(exact):390000,
                    token-bucket(exact):600,
                    framerate(exact):60,
                    hor-size(exact):320,
                    ver-size(exact):240}
```

Figure 32: COPS and COPN indicating operation point added

12. IANA Considerations

Following the guidelines in [RFC4566], in [RFC4585], and in [RFC3550], the IANA is requested to register:

1. The 'cop' tag to be used with ccm under rtcp-fb AVPF attribute in SDP.

2. The FMT number TBA1 to be allocated to the COP feedback message from this specification.

3. A registry listing registered values for 'cop' message item type, with initial values from Table 1.

4. A registry listing registered values and tag names for 'cop' parameter type, with initial values from Table 4.

13. Security Considerations

This document extends the CCM [RFC5104] and defines new messages, i.e. COPR, COPN and COPS. The exchange of these new messages MAY have some security implications, which need to be addressed by the user. Following are some important implications,

1. Identity spoofing - An attacker can spoof him/herself as an authenticated user and can falsely control or indicate the codec parameters of any source transmission. In order to prevent this type of attack, a strong authentication and integrity protection mechanism is needed.

2. Denial of Service (DoS) - An attacker can falsely set codec parameters for all the source streams which MAY result in Denial of Service (DoS). An Authentication protocol MAY save from this attack.

3. Man-in-Middle Attack (MiMT) - The codec configuration and notification of changes of the RTP source is prone to a Man-in-Middle attack. The public key authentication May be used to prevent MiMT.

14. Open Issues

There is currently no defined way for a media receiver to indicate that it wants to release the restrictions it previously had on an operation point, if the media stream contains only a single operation point.

15. Acknowledgements

The authors would like to thank Prof. Dr.-Ing. Markus Kampmann at Fachhochschule Koblenz University of Applied Sciences and Prof. Dr.-

Ing. Frank Hartung at Multimediatechnik, Audio- und Videotechnik at Fachhochschule Aachen for fruitful contributions and discussions during the initial stages of writing this specification. The authors would also like to thank Christer Holmberg for feedback on the specification.

16. References

16.1. Normative References

[H241] ITU-T Recommendation H.241, "Extended video procedures and control signals for H.300 series terminals", May 2006.

[H264] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services", March 2010.

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

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3890] Westerlund, M., "A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)", RFC 3890, September 2004.

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

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC6184] Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, May 2011.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, May 2011.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, May 2011.

16.2. Informative References

[I-D.ietf-avtext-multiple-clock-rates] Petit-Huguenin, M., "Support for multiple clock rates in an RTP session", draft-ietf-avtext-multiple-clock-rates-02 (work in progress), January 2012.

[I-D.westerlund-avtext-rtp-stream-pause] Akram, A., Burman, B., Grondal, D., and M. Westerlund, "RTP Media Stream Pause and Resume", draft-westerlund-avtext-rtp-stream-pause-02 (work in progress), July 2012.

[I-D.westerlund-mmusic-sdp-bw-attribute] Frankkila, T., Westerlund, M., and B. Burman, "Extensible Bandwidth Attribute for SDP", draft-westerlund-mmusic-sdp-bw-attribute-00 (work in progress), October 2011.

[RFC2212] Shenker, S., Partridge, C., and R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997.

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

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control

Protocol Extended Reports (RTCP XR)", RFC 3611,

November 2003.

[RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, June 2005.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

[RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.

[RFC5968] Ott, J. and C. Perkins, "Guidelines for Extending the RTP Control Protocol (RTCP)", RFC 5968, September 2010.

Authors' Addresses

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Bo Burman
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 13 11
Email: bo.burman@ericsson.com

Laurits Hamm
Ericsson
Ericsson Allee 1
DE-52134 Herzogenrath
Germany

Phone: +49 2407 575 6779
Email: laurits.hamm@ericsson.com