Network Working Group
INTERNET-DRAFT
Expires: April 2008
Intended Status: Proposed Standard

Stephan Wenger
Umesh Chandra
Nokia
Magnus Westerlund
Bo Burman
Ericsson
October 26, 2007

Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)

<draft-ietf-avt-avpf-ccm-10.txt>

Status of this Memo

Copyright Notice

Abstract

This document specifies a few extensions to the messages defined in the Audio-Visual Profile with Feedback (AVPF). They are helpful primarily in conversational multimedia scenarios where centralized multipoint functionalities are in use. However, some are also usable in smaller multicast environments and point-to-point calls.

The extensions discussed are messages related to the ITU-T H.271 Video Back Channel, Full Intra Request, Temporary Maximum Media Stream Bit Rate and Temporal Spatial Trade-off.

TABLE OF CONTENTS

1. Introduction

When the Audio-Visual Profile with Feedback (AVPF) [RFC4585] was developed, the main emphasis lay in the efficient support of point-to-point and small multipoint scenarios without centralized multipoint control. However, in practice, many small multipoint conferences operate utilizing devices known as Multipoint Control Units (MCUs). Long-standing experience of the conversational video conferencing industry suggests that there is a need for a few additional feedback messages, to support centralized multipoint conferencing efficiently. Some of the messages have applications beyond centralized multipoint, and this is indicated in the description of the message. This is especially true for the message intended to carry ITU-T Rec. H.271 [H.271] bit strings for Video Back Channel messages.

In Real-time Transport Protocol (RTP) [RFC3550] terminology, MCUs comprise mixers and translators. Most MCUs also include signaling support. During the development of this memo, it was noticed that there is considerable confusion in the community related to the use of terms such as mixer, translator, and MCU. In response to these concerns, a number of topologies have been identified that are of practical relevance to the industry, but are not documented in sufficient detail in [RFC3550]. These topologies are documented in [Topologies], and understanding this memo requires previous or parallel study of [Topologies].

Some of the messages defined here are forward only, in that they do not require an explicit notification to the message emitter that they have been received and/or indicating the message receiver's actions. Other messages require a response, leading to a two way communication model that one could view as useful for control purposes. However, it is not the intention of this memo to open up RTP Control Protocol (RTCP) to a generalized control protocol. All mentioned messages have relatively strict real-time constraints, in the sense that their value diminishes with increased delay. This makes the use of more traditional control protocol means, such as Session Initiation Protocol (SIP) [RFC3261], undesirable when used for the same purpose. That is why this solution is recommended instead of "XML Schema for Media Control" [XML-MC], which uses SIP Info to transfer XML messages with similar semantics to what are defined in this memo. Furthermore, all messages are of a very simple format that can be easily processed by an RTP/RTCP sender/receiver. Finally, and most importantly, all messages relate only to the RTP stream with which they are associated, and not to

any other property of a communication system. In particular, none of them relate to the properties of the access links traversed by the session.

2. Definitions

2.1. Glossary

AIMD - Additive Increase Multiplicative Decrease
AVPF - The extended RTP profile for RTCP-based feedback
FEC - Forward Error Correction
FCI - Feedback Control Information [RFC4585]
FIR - Full Intra Request
MCU - Multipoint Control Unit
MPEG - Moving Picture Experts Group
TMMBN - Temporary Maximum Media Stream Bit Rate Notification
TMMBR - Temporary Maximum Media Stream Bit Rate Request
PLI - Picture Loss Indication
PR - Packet rate
QP - Quantizer Parameter
RTT - Round trip time
SSRC - Synchronization Source
TSTN - Temporal Spatial Trade-off Notification
TSTR - Temporal Spatial Trade-off Request
VBCM - Video Back Channel Message indication.

2.2. Terminology

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

Message:
An RTCP feedback message [RFC4585] defined by this specification, of one of the following types:

Request:
Message that requires acknowledgement

Command:
Message that forces the receiver to an action

Indication:
Message that reports a situation

Notification:
Message that provides a notification that an event has occurred. Notifications are commonly generated in response to a Request.

Note that, with the exception of "Notification", this terminology is in alignment with ITU-T Rec. H.245 [H245].

Decoder Refresh Point:
A bit string, packetized in one or more RTP packets, which completely resets the decoder to a known state.

Examples for "hard" decoder refresh points are Intra pictures in H.261, H.263, MPEG-1, MPEG-2, and MPEG-4 part 2, and Instantaneous Decoder Refresh (IDR) pictures in H.264. "Gradual" decoder refresh points may also be used; see for example [AVC]. While both "hard" and "gradual" decoder refresh points are acceptable in the scope of this specification, in most cases the user experience will benefit from using a "hard" decoder refresh point.

A decoder refresh point also contains all header information above the picture layer (or equivalent, depending on the video compression standard) that is conveyed in-band. In H.264, for example, a decoder refresh point contains parameter set Network Adaptation Layer (NAL) units that generate parameter sets necessary for the decoding of the following slice/data partition NAL units (and that are not conveyed out of band).

Decoding:
The operation of reconstructing the media stream.

Rendering:
The operation of presenting (parts of) the reconstructed media stream to the user.

Stream thinning:
The operation of removing some of the packets from a media stream. Stream thinning, preferably, is media-aware, implying that media packets are removed in the order of increasing relevance to the reproductive quality. However, even when employing media-aware stream thinning, most media streams quickly lose quality when subjected to increasing

levels of thinning. Media-unaware stream thinning leads to even worse quality degradation. In contrast to transcoding, stream thinning is typically seen as a computationally lightweight operation.

Media:
: Often used (sometimes in conjunction with terms like bit rate, stream, sender ...) to identify the content of the forward RTP packet stream (carrying the codec data), to which the codec control message applies.

Media Stream:
: The stream of RTP packets labeled with a single Synchronization Source (SSRC) carrying the media (and also in some cases repair information such as retransmission or Forward Error Correction (FEC) information).

Total media bit rate:
: The total bits per second transferred in a media stream, measured at an observer-selected protocol layer and averaged over a reasonable timescale, the length of which depends on the application. In general, a media sender and a media receiver will observe different total media bit rates for the same stream, first because they may have selected different reference protocol layers, and second, because of changes in per-packet overhead along the transmission path. The goal with bit rate averaging is to be able to ignore any burstiness on very short timescales, below for example 100 ms, introduced by scheduling or link layer packetization effects.

Maximum total media bit rate:
: The upper limit on total media bit rate for a given media stream at a particular receiver and for its selected protocol layer. Note that this value cannot be measured on the received media stream, instead it needs to be calculated or determined through other means, such as QoS negotiations or local resource limitations. Also note that this value is an average (on a timescale that is reasonable for the application) and that it may be different from the instantaneous bit-rate seen by packets in the media stream.

Overhead:
: All protocol header information required to convey a packet with media data from sender to receiver, from the application layer down to a pre-defined protocol level (for example down to, and including, the IP header). Overhead may include, for example, IP, UDP, and RTP headers, any layer 2 headers, any

Contributing Sources (CSRCs), RTP-Padding, and RTP header extensions. Overhead excludes any RTP payload headers and the payload itself.

Net media bit rate:
The bit rate carried by a media stream, net of overhead. That is, the bits per second accounted for by encoded media, any applicable payload headers, and any directly associated meta payload information placed in the RTP packet. A typical example of the latter is redundancy data provided by the use of RFC 2198 [RFC2198]. Note that, unlike the total media bit rate, the net media bit rate will have the same value at the media sender and at the media receiver unless any mixing or translating of the media has occurred.

For a given observer, the total media bit rate for a media stream is equal to the sum of the net media bit rate and the per-packet overhead as defined above multiplied by the packet rate.

Feasible region:
The set of all combinations of packet rate and net media bit rate that do not exceed the restrictions in maximum media bit rate placed on a given media sender by the Temporary Maximum Media Stream Bit-rate Request (TMMBR) messages it has received. The feasible region will change as new TMMBR messages are received.

Bounding set:
The set of TMMBR tuples, selected from all those received at a given media sender, that define the feasible region for that media sender. The media sender uses an algorithm such as that in section 3.5.4.2 to determine or iteratively approximate the current bounding set, and reports that set back to the media receivers in a Temporary Maximum Media Stream Bit-rate Notification (TMMBN) message.

2.3. Topologies

Please refer to [Topologies] for an in depth discussion. The topologies referred to throughout this memo are labeled (consistently with [Topologies]) as follows:

Topo-Point-to-Point . . . . . Point-to-point communication
Topo-Multicast . . . . . Multicast communication
Topo-Translator . . . . . Translator based
Topo-Mixer . . . . . Mixer based
Topo-RTP-switch-MCU . . . . RTP stream switching MCU,

Topo-RTCP-terminating-MCU . . Mixer but terminating RTCP

3. Motivation

This section discusses the motivation and usage of the different video and media control messages. The video control messages have been under discussion for a long time, and a requirement draft was drawn up [Basso]. This draft has expired; however we quote relevant sections of it to provide motivation and requirements.

3.1. Use Cases

There are a number of possible usages for the proposed feedback messages. Let us begin by looking through the use cases Basso et al. [Basso] proposed. Some of the use cases have been reformulated and comments have been added.

1. An RTP video mixer composes multiple encoded video sources into a single encoded video stream. Each time a video source is added, the RTP mixer needs to request a decoder refresh point from the video source, so as to start an uncorrupted prediction chain on the spatial area of the mixed picture occupied by the data from the new video source.

2. An RTP video mixer receives multiple encoded RTP video streams from conference participants, and dynamically selects one of the streams to be included in its output RTP stream. At the time of a bit stream change (determined through means such as voice activation or the user interface), the mixer requests a decoder refresh point from the remote source, in order to avoid using unrelated content as reference data for inter picture prediction. After requesting the decoder refresh point, the video mixer stops the delivery of the current RTP stream and monitors the RTP stream from the new source until it detects data belonging to the decoder refresh point. At that time, the RTP mixer starts forwarding the newly selected stream to the receiver(s).

3. An application needs to signal to the remote encoder that the desired trade-off between temporal and spatial resolution has changed. For example, one user may prefer a higher frame rate and a lower spatial quality, and another user may prefer the opposite. This choice is also highly content dependent. Many current video conferencing systems offer in the user interface a mechanism to make this selection, usually in the form of a slider. The mechanism is helpful in point-to-point, centralized

multipoint and non-centralized multipoint uses.

4. Use case 4 of the Basso draft applies only to Picture Loss Indication (PLI) as defined in AVPF [RFC4585] and is not reproduced here.

5. Use case 5 of the Basso draft relates to a mechanism known as "freeze picture request". Sending freeze picture requests over a non-reliable forward RTCP channel has been identified as problematic. Therefore, no freeze picture request has been included in this memo, and the use case discussion is not reproduced here.

6. A video mixer dynamically selects one of the received video streams to be sent out to participants and tries to provide the highest bit rate possible to all participants, while minimizing stream trans-rating. One way of achieving this is to set up sessions with endpoints using the maximum bit rate accepted by each endpoint, and accepted by the call admission method used by the mixer. By means of commands that reduce the maximum media stream bit rate below what has been negotiated during session set up, the mixer can reduce the maximum bit rate sent by endpoints to the lowest of all the accepted bit rates. As the lowest accepted bit rate changes due to endpoints joining and leaving or due to network congestion, the mixer can adjust the limits at which endpoints can send their streams to match the new value. The mixer then requests a new maximum bit rate, which is equal to or less than the maximum bit rate negotiated at session setup for a specific media stream, and the remote endpoint can respond with the actual bit rate that it can support.

The picture Basso et al draws up covers most applications we foresee. However, we would like to extend the list with two additional use cases:

7. Currently deployed congestion control algorithms (AIMD and TFRC [RFC3448]) probe for additional available capacity as long as there is something to send. With congestion control algorithms using packet loss as the indication for congestion, this probing generally results in reduced media quality (often to a point where the distortion is large enough to make the media unusable), due to packet loss and increased delay.

   In a number of deployment scenarios, especially cellular ones, the bottleneck link is often the last hop link. That cellular link also commonly has some type of QoS negotiation enabling the cellular device to learn the maximal bit rate available over this last hop. A media receiver behind this link can, in most (if not

all) cases, calculate at least an upper bound for the bit rate available for each media stream it presently receives. How this is done is an implementation detail and not discussed herein. Indicating the maximum available bit rate to the transmitting party for the various media streams can be beneficial to prevent that party from probing for bandwidth for this stream in excess of a known hard limit. For cellular or other mobile devices, the known available bit rate for each stream (deduced from the link bit rate) can change quickly, due to handover to another transmission technology, QoS renegotiation due to congestion, etc. To enable minimal disruption of service, quick convergence is necessary, and therefore media path signaling is desirable.

8. The use of reference picture selection (RPS) as an error resilience tool has been introduced in 1997 as NEWPRED [NEWPRED], and is now widely deployed. When RPS is in use, simplistically put, the receiver can send a feedback message to the sender, indicating a reference picture that should be used for future prediction. ([NEWPRED] mentions other forms of feedback as well.) AVPF contains a mechanism for conveying such a message, but did not specify for which codec and according to which syntax the message should conform. Recently, the ITU-T finalized Rec. H.271 which (among other message types) also includes a feedback message. It is expected that this feedback message will fairly quickly enjoy wide support. Therefore, a mechanism to convey feedback messages according to H.271 appears to be desirable.

3.2. Using the Media Path

There are two reasons why we use the media path for the codec control messages.

First, systems employing MCUs often separate the control and media processing parts. As these messages are intended for or generated by the media part rather than the signaling part of the MCU, having them on the media path avoids transmission across interfaces and unnecessary control traffic between signaling and processing. If the MCU is physically decomposed, the use of the media path avoids the need for media control protocol extensions (e.g. in MEGACO [RFC3525]).

Secondly, the signaling path quite commonly contains several signaling entities, e.g. SIP proxies and application servers. Avoiding going through signaling entities avoids delay for several reasons. Proxies have less stringent delay requirements than media processing and due to their complex and more generic nature may result in significant processing delay. The topological locations of the signaling entities are also commonly not optimized for

minimal delay, but rather towards other architectural goals. Thus, the signaling path can be significantly longer in both geographical and delay sense.

3.3. Using AVPF

The AVPF feedback message framework [RFC4585] provides the appropriate framework to implement the new messages. AVPF implements rules controlling the timing of feedback messages to avoid congestion through network flooding by RTCP traffic. We re-use these rules by referencing AVPF.

The signaling setup for AVPF allows each individual type of function to be configured or negotiated on an RTP session basis.

3.3.1. Reliability

The use of RTCP messages implies that each message transfer is unreliable, unless the lower layer transport provides reliability. The different messages proposed in this specification have different requirements in terms of reliability. However, in all cases, the reaction to an (occasional) loss of a feedback message is specified.

3.4. Multicast

The codec control messages might be used with multicast. The RTCP timing rules specified in [RFC3550] and [RFC4585] ensure that the messages do not cause overload of the RTCP connection. The use of multicast may result in the reception of messages with inconsistent semantics. The reaction to inconsistencies depends on the message type, and is discussed for each message type separately.

3.5. Feedback Messages

This section describes the semantics of the different feedback messages and how they apply to the different use cases.

3.5.1. Full Intra Request Command

A Full Intra Request (FIR) Command, when received by the designated media sender, requires that the media sender sends a Decoder Refresh Point (see 2.2) at the earliest opportunity. The evaluation of such

opportunity includes the current encoder coding strategy and the current available network resources.

FIR is also known as an "instantaneous decoder refresh request", "fast video update request" or "video fast update request".

Using a decoder refresh point implies refraining from using any picture sent prior to that point as a reference for the encoding process of any subsequent picture sent in the stream. For predictive media types that are not video, the analogue applies. For example, if in MPEG-4 systems scene updates are used, the decoder refresh point consists of the full representation of the scene and is not delta-coded relative to previous updates.

Decoder refresh points, especially Intra or IDR pictures, are in general several times larger in size than predicted pictures. Thus, in scenarios in which the available bit rate is small, the use of a decoder refresh point implies a delay that is significantly longer than the typical picture duration.

Usage in multicast is possible; however aggregation of the commands is recommended. A receiver that receives a request closely after sending a decoder refresh point -- within 2 times the longest Round Trip Time (RTT) known, plus and AVPF-induced RTCP packet sending delays -- should await a second request message to ensure that the media receiver has not been served by the previously delivered decoder refresh point. The reason for the specified delay is to avoid sending unnecessary decoder refresh points. A session participant may have sent its own request while another participant's request was in-flight to them. Suppressing those requests that may have been sent without knowledge about the other request avoids this issue.

Using the FIR command to recover from errors is explicitly disallowed, and instead the PLI message defined in AVPF [RFC4585] should be used. The PLI message reports lost pictures and has been included in AVPF for precisely that purpose.

Full Intra Request is applicable in use-cases 1 and 2.

3.5.1.1. Reliability

The FIR message results in the delivery of a decoder refresh point, unless the message is lost. Decoder refresh points are easily identifiable from the bit stream. Therefore, there is no need for protocol-level notification, and a simple command repetition mechanism is sufficient for ensuring the level of reliability required. However, the potential use of repetition does require a

mechanism to prevent the recipient from responding to messages already received and responded to.

To ensure the best possible reliability, a sender of FIR may repeat the FIR request until the desired content has been received. The repetition interval is determined by the RTCP timing rules applicable to the session. Upon reception of a complete decoder refresh point or the detection of an attempt to send a decoder refresh point (which got damaged due to a packet loss), the repetition of the FIR must stop. If another FIR is necessary, the request sequence number must be increased. A FIR sender shall not have more than one FIR request (different request sequence number) outstanding at any time per media sender in the session.

The receiver of FIR (i.e. the media sender) behaves in complementary fashion to ensure delivery of a decoder refresh point. If it receives repetitions of the FIR more than 2*RTT after it has sent a decoder refresh point, it shall send a new decoder refresh point. Two round trip times allow time for the decoder refresh point to arrive back to the requestor and for the end of repetitions of FIR to reach and be detected by the media sender.

An RTP mixer or RTP switching MCU that receive a FIR from a media receiver is responsible to ensure that a decoder refresh point is delivered to the requesting receiver. It may be necessary for the mixer/MCU to generate FIR commands. From a reliability perspective, the two legs (FIR-requesting endpoint to mixer/MCU, and mixer/MCU to decoder refresh point generating endpoint) are handled independently from each other.

3.5.2. Temporal Spatial Trade-off Request and Notification

The Temporal Spatial Trade-off Request (TSTR) instructs the video encoder to change its trade-off between temporal and spatial resolution. Index values from 0 to 31 indicate monotonically a desire for higher frame rate. That is, a requester asking for an index of 0 prefers a high quality and is willing to accept a low frame rate, whereas a requester asking for 31 wishes a high frame rate, potentially at the cost of low spatial quality.

In general the encoder reaction time may be significantly longer than the typical picture duration. See use case 3 for an example. The encoder decides whether and to what extent the request results in a change of the trade-off. It returns a Temporal Spatial Trade-Off Notification (TSTN) message to indicate the trade-off that it will use henceforth.

TSTR and TSTN have been introduced primarily because it is believed that control protocol mechanisms, e.g. a SIP re-invite, are too heavyweight and too slow to allow for a reasonable user experience. Consider, for example, a user interface where the remote user selects the temporal/spatial trade-off with a slider. An immediate feedback to any slider movement is required for a reasonable user experience. A SIP re-INVITE [RFC3261] would require at least two round-trips more (compared to the TSTR/TSTN mechanism) and may involve proxies and other complex mechanisms. Even in a well-designed system, it could take a second or so until the new trade-off is finally selected. Furthermore the use of RTCP solves the multicast use case very efficiently.

The use of TSTR and TSTN in multipoint scenarios is a non-trivial subject, and can be achieved in many implementation-specific ways. Problems stem from the fact that TSTRs will typically arrive unsynchronized, and may request different trade-off values for the same stream and/or endpoint encoder. This memo does not specify a translator's, mixer's or endpoint's reaction to the reception of a suggested trade-off as conveyed in the TSTR. We only require the receiver of a TSTR message to reply to it by sending a TSTN, carrying the new trade-off chosen by its own criteria (which may or may not be based on the trade-off conveyed by the TSTR). In other words, the trade-off sent in TSTR is a non-binding recommendation, nothing more.

Three TSTR/TSTN scenarios need to be distinguished, based on the topologies described in [Topologies]. The scenarios are described in the following sub-clauses.

3.5.2.1. Point-to-Point

In this most trivial case (Topo-Point-to-Point), the media sender typically adjusts its temporal/spatial trade-off based on the requested value in TSTR, subject to its own capabilities. The TSTN message conveys back the new trade-off value (which may be identical to the old one if, for example, the sender is not capable of adjusting its trade-off).

3.5.2.2. Point-to-Multipoint Using Multicast or Translators

RTCP Multicast is used either with media multicast according to Topo-Multicast, or following RFC 3550's translator model according to Topo-Translator. In these cases, unsynchronized TSTR messages from different receivers may be received, possibly with different requested trade-offs (because of different user preferences). This

memo does not specify how the media sender tunes its trade-off. Possible strategies include selecting the mean or median of all trade-off requests received, giving priority to certain participants, or continuing to use the previously selected trade-off (e.g. when the sender is not capable of adjusting it). Again, all TSTR messages need to be acknowledged by TSTN, and the value conveyed back has to reflect the decision made.

3.5.2.3. Point-to-Multipoint Using RTP Mixer

In this scenario (Topo-Mixer) the RTP mixer receives all TSTR messages, and has the opportunity to act on them based on its own criteria. In most cases, the mixer should form a "consensus" of potentially conflicting TSTR messages arriving from different participants, and initiate its own TSTR message(s) to the media sender(s). As in the previous scenario, the strategy for forming this "consensus" is up to the implementation, and can, for example, encompass averaging the participants' request values, giving priority to certain participants, or using session default values.

Even if a mixer or translator performs transcoding, it is very difficult to deliver media with the requested trade-off, unless the content the mixer or translator receives is already close to that trade-off. Thus, if the mixer changes its trade-off, it needs to request the media sender(s) to use the new value, by creating a TSTR of its own. Upon reaching a decision on the used trade-off it includes that value in the acknowledgement to the downstream requestors. Only in cases where the original source has substantially higher quality (and bit rate) is it likely that transcoding alone can result in the requested trade-off.

3.5.2.4. Reliability

A request and reception acknowledgement mechanism is specified. The Temporal Spatial Trade-off Notification (TSTN) message informs the requester that its request has been received, and what trade-off is used henceforth. This acknowledgment mechanism is desirable for at least the following reasons:

- A change in the trade-off cannot be directly identified from the media bit stream.
- User feedback cannot be implemented without knowing the chosen trade-off value, according to the media sender's constraints.
- Repetitive sending of messages requesting an unimplementable trade-off can be avoided.

3.5.3. H.271 Video Back Channel Message

ITU-T Rec. H.271 defines syntax, semantics, and suggested encoder reaction to a video back channel message. The structure defined in this memo is used to transparently convey such a message from media receiver to media sender. In this memo, we refrain from an in-depth discussion of the available code points within H.271 and refer to the specification text [H.271] instead.

However, we note that some H.271 messages bear similarities with native messages of AVPF and this memo. Furthermore, we note that some H.271 message are known to require caution in multicast environments -- or are plainly not usable in multicast or multipoint scenarios. Table 1 provides a brief, oversimplifed overview of the messages currently defined in H.271, their roughly corresponding AVPF or CCM messages (the latter as specified in this memo), and an indication of our current knowledge of their multicast safety.

H.271 msg type	AVPF/CCM msg type	multicast-safe
0 (when used for reference picture selection)	AVPF RPSI	No (positive ACK of pictures)
1 picture loss	AVPF PLI	Yes
2 partial loss	AVPF SLI	Yes
3 one parameter CRC	N/A	Yes (no required sender action)
4 all parameter CRC	N/A	Yes (no required sender action)
5 refresh point	CCM FIR	Yes

Table 1: H.271 messages and their AVPF/CCM equivalents

> Note: H.271 message type 0 is not a strict equivalent to AVPF's Reference Picture Selection Indication (RPSI); it is an indication of known-as-correct reference picture(s) at the decoder. It does not command an encoder to use a defined reference picture (the form of control information envisioned to be carried in RPSI). However, it is believed and intended that H.271 message type 0 will be used for the same purpose as AVPF's RPSI -- although other use forms are also possible.

In response to the opaqueness of the H.271 messages, especially with respect to the multicast safety, the following guidelines MUST be followed when an implementation wishes to employ the H.271 video back channel message:

1. Implementations utilizing the H.271 feedback message MUST stay in compliance with congestion control principles, as outlined in section 5.

2. An implementation SHOULD utilize the IETF-native messages as defined in [RFC4585] and in this memo instead of similar messages defined in [H.271]. Our current understanding of similar messages is documented in Table 1 above. One good reason to divert from the SHOULD statement above would be if it is clearly understood that, for a given application and video compression standard, the aforementioned "similarity" is not given, in contrast to what the table indicates.

3. It has been observed that some of the H.271 code points currently in existence are not multicast-safe. Therefore, the sensible thing to do is not to use the H.271 feedback message type in multicast environments. It MAY be used only when all the issues mentioned later are fully understood by the implementer, and properly taken into account by all endpoints. In all other cases, the H.271 message type MUST NOT be used in conjunction with multicast.

4. It has been observed that even in centralized multipoint environments, where the mixer should theoretically be able to resolve issues as documented below, the implementation of such a mixer and cooperative endpoints is a very difficult and tedious task. Therefore, H.271 messages MUST NOT be used in centralized multipoint scenarios, unless all the issues mentioned below are fully understood by the implementer, and properly taken into account by both mixer and endpoints.

Issues to be taken into account when considering the use of H.271 in multipoint environments:

1. Different state on different receivers. In many environments it cannot be guaranteed that the decoder state of all media receivers is identical at any given point in time. The most obvious reason for such a possible misalignment of state is a loss that occurs on the path to only one of many media receivers. However, there are other not so obvious reasons, such as recent joins to the multipoint conference (be it by joining the multicast group or through additional mixer output). Different states can lead the media receivers to issue potentially contradicting H.271 messages (or one media receiver issuing an H.271 message that, when observed by the media sender, is not helpful for the other media receivers). A naive reaction of the media sender to these contradicting messages can lead to

unpredictable and annoying results.

2. Combining messages from different media receivers in a media sender is a non-trivial task. As reasons, we note that these messages may be contradicting each other, and that their transport is unreliable (there may well be other reasons). In case of many H.271 messages (i.e. types 0, 2, 3, and 4), the algorithm for combining must be aware both of the network/protocol environment (i.e. with respect to congestion) and of the media codec employed, as H.271 messages of a given type can have different semantics for different media codecs.

3. The suppression of requests may need to go beyond the basic mechanisms described in AVPF (which are driven exclusively by timing and transport considerations on the protocol level). For example, a receiver is often required to refrain from (or delay) generating requests, based on information it receives from the media stream. For instance, it makes no sense for a receiver to issue a FIR when a transmission of an Intra/IDR picture is ongoing.

4. When using the non-multicast-safe messages (e.g. H.271 type 0 positive ACK of received pictures/slices) in larger multicast groups, the media receiver will likely be forced to delay or even omit sending these messages. For the media sender this looks like data has not been properly received (although it was received properly), and a naively implemented media sender reacts to these perceived problems where it should not.

3.5.3.1. Reliability

H.271 Video Back Channel messages do not require reliable transmission, and confirmation of the reception of a message can be derived from the forward video bit stream. Therefore, no specific reception acknowledgement is specified.

With respect to re-sending rules, clause 3.5.1.1 applies.

3.5.4. Temporary Maximum Media Stream Bit Rate Request and Notification

A receiver, translator or mixer uses the Temporary Maximum Media Stream Bit Rate Request (TMMBR, "timber") to request a sender to limit the maximum bit rate for a media stream (see 2.2) to, or below, the provided value. The Temporary Maximum Media Stream Bit Rate Notification (TMMBN) contains the media sender's current view of the most limiting subset of the TMMBR-defined limits it has received, to help the participants to suppress TMMBR requests that

would not further restrict the media sender. The primary usage for the TMMBR/TMMBN messages is in a scenario with an MCU or mixer (use case 6), corresponding to Topo-Translator or Topo-Mixer, but also to Topo-Point-to-Point.

Each temporary limitation on the media stream is expressed as a tuple. The first component of the tuple is the maximum total media bit rate (as defined in section 2.2) that the media receiver is currently prepared to accept for this media stream. The second component is the per-packet overhead that the media receiver has observed for this media stream at its chosen reference protocol layer.

As indicated in section 2.2, the overhead as observed by the sender of the TMMBR (i.e. the media receiver) may differ from the overhead observed at the receiver of the TMMBR (i.e. the media sender) due to use of a different reference protocol layer at the other end or due to the intervention of translators or mixers that affect the amount of per packet overhead. For example, a gateway in between the two that converts between IPv4 and IPv6 affects the per-packet overhead by 20 bytes. Other mechanisms that change the overhead include tunnels. The problem with varying overhead is also discussed in [RFC3890]. As will be seen in the description of the algorithm for use of TMMBR, the difference in perceived overhead between the sending and receiving ends presents no difficulty because calculations are carried out in terms of variables that have the same value at the sender as at the receiver -- for example, packet rate and net media rate.

Reporting both maximum total media bit rate and per-packet overhead allows different receivers to provide bit rate and overhead values for different protocol layers, for example at the IP level, at the outer part of a tunnel protocol, or at the link layer. The protocol level a peer reports on depends on the level of integration the peer has, as it needs to be able to extract the information from that protocol level. For example, an application with no knowledge of the IP version it is running over can not meaningfully determine the overhead of the IP header, and hence will not want to include IP overhead in the overhead or maximum total media bit rate calculation.

It is expected that most peers will be able to report values at least for the IP layer. In certain implementations it may be advantageous to also include information pertaining to the link layer, which in turn allows for a more precise overhead calculation and a better optimization of connectivity resources.

The Temporary Maximum Media Stream Bit Rate messages are generic messages that can be applied to any RTP packet stream. This separates them from the other codec control messages defined in this specification, which apply only to specific media types or payload formats. The TMMBR functionality applies to the transport, and the requirements the transport places on the media encoding.

The reasoning below assumes that the participants have negotiated a session maximum bit rate, using a signaling protocol. This value can be global, for example in case of point-to-point, multicast, or translators. It may also be local between the participant and the peer or mixer. In either case, the bit rate negotiated in signaling is the one that the participant guarantees to be able to handle (depacketize and decode). In practice, the connectivity of the participant also influences the negotiated value -- it does not make much sense to negotiate a total media bit rate that one's network interface does not support.

It is also beneficial to have negotiated a maximum packet rate for the session or sender. RFC 3890 provides an SDP [RFC4566] attribute that can be used for this purpose; however, that attribute is not usable in RTP sessions established using offer/answer [RFC3264]. Therefore an optional maximum packet rate signaling parameter is specified in this memo.

An already established maximum total media bit rate may be changed at any time, subject to the timing rules governing the sending of feedback messages. The limit may change to any value between zero and the session maximum, as negotiated during session establishment signaling. However, even if a sender has received a TMMBR message allowing an increase in the bit rate, all increases must be governed by a congestion control mechanism. TMMBR indicates known limitations only, usually in the local environment, and does not provide any guarantees about the full path. Furthermore, any increases in TMMBR-established bit rate limits are to be executed only after a certain delay from the sending of the TMMBN message that notifies the world about the increase in limit. The delay is specified as at least twice the longest RTT as known by the media sender, plus the media sender's calculation of the required wait time for the sending of another TMMBR message for this session based on AVPF timing rules. This delay is introduced to allow other session participants to make known their bit rate limit requirements, which may be lower.

If it is likely that the new value indicated by TMMBR will be valid for the remainder of the session, the TMMBR sender is expected to perform a renegotiation of the session upper limit using the session signaling protocol.

3.5.4.1. Behavior for media receivers using TMMBR

This section is an informal description of behaviour described more precisely in section 4.2.

A media sender begins the session limited by the maximum media bit rate and maximum packet rate negotiated in session signaling, if any. Note that this value may be negotiated for another protocol layer than the one the participant uses in its TMMBR messages. Each media receiver selects a reference protocol layer, forms an estimate of the overhead it is observing (or estimating it if no packets has been seen yet) at that reference level, and determines the maximum total media bit rate it can accept, taking into account its own limitations and any transport path limitations of which it may be aware. In case the current limitations are more restricting than what was agreed on in the session signaling, the media receiver reports its initial estimate of these two quantities to the media sender using a TMMBR message. Overall message traffic is reduced by the possibility of including tuples for multiple media senders in the same TMMBR message.

The media sender applies an algorithm such as that specified in section 3.5.4.2 to select which of the tuples it has received are most limiting (i.e. the bounding set as defined in section 2.2). It modifies its operation to stay within the feasible region (as defined in section 2.2), and also sends out a TMMBN notification to the media receivers indicating the selected bounding set. That notification also indicates who was responsible for the tuples in the bounding set, i.e. the "owner"(s) of the limitation. A session participant that owns no tuple in the bounding set is called a "non-owner".

If a media receiver does not own one of the tuples in the bounding set reported by the TMMBN, it applies the same algorithm as the media sender to determine if its current estimated (maximum total media bit rate, overhead) tuple would enter the bounding set if known to the media sender. If so, it issues a TMMBR request reporting the tuple value to the sender. Otherwise it takes no action for the moment. Periodically, its estimated tuple values may change or it may receive a new TMMBN. If so, it reapplies the algorithm to decide whether it needs to issue a TMMBR request.

If, alternatively, a media receiver owns one of the tuples in the reported bounding set, it takes no action until such time as its estimate of its own tuple values changes. At that time it sends a TMMBR request to the media sender to report the changed values.

A media receiver may change status between owner and non-owner of a bounding tuple between one TMMBN message and the next. Thus, it must check the contents of each TMMBN to determine its subsequent actions.

Implementations may use other algorithms of their choosing, as long as the bit rate limitations resulting from the exchange of TMMBR and TMMBN messages are at least as strict (at least as low, in the bit rate dimension) as the ones resulting from the use of the aforementioned algorithm.

Obviously, in point-to-point cases, when there is only one media receiver, this receiver becomes "owner" once it receives the first TMMBN in response to its own TMMBR, and stays "owner" for the rest of the session. Therefore, when it is known that there will always be only a single media receiver, the above algorithm is not required. Media receivers that are aware they are the only ones in a session can send TMMBR messages with bit rate limits both higher and lower than the previously notified limit, at any time (subject to the AVPF [RFC4585] RTCP RR send timing rules). However, it may be difficult for a session participant to determine if it is the only receiver in the session. Because of this any implementation of TMMBR is required to include the algorithm described in the next section or a stricter equivalent.

3.5.4.2. Algorithm for establishing current limitations

This section introduces an example algorithm for the calculation of a session limit. Other algorithms can be employed, as long as the result of the calculation is at least as restrictive as the result that is obtained by this algorithm.

First, it is important to consider the implications of using a tuple for limiting the media sender's behavior. The bit rate and the overhead value result in a two-dimensional solution space for the calculation of the bit rate of media streams. Fortunately, the two variables are linked. Specifically, the bit rate available for RTP payloads is equal to the TMMBR reported bit rate minus the packet rate used, multiplied by the TMMBR reported overhead converted to bits. As a result, when different bit rate/overhead combinations need to be considered, the packet rate determines the correct limitation. This is perhaps best explained by an example:

Example:

Receiver A: TMMBR_max total BR = 35 kbps, TMMBR_OH = 40 bytes
Receiver B: TMMBR_max total BR = 40 kbps, TMMBR_OH = 60 bytes

For a given packet rate (PR) the bit rate available for media payloads in RTP will be:

Max_net media_BR_A =
TMMBR_max total BR_A - PR * TMMBR_OH_A * 8 ... (1)

Max_net media_BR_B =
TMMBR_max total BR_B - PR * TMMBR_OH_B * 8 ... (2)

For a PR = 20 these calculations will yield a Max_net media_BR_A = 28600 bps and Max_net media_BR_B = 30400 bps, which suggests that receiver A is the limiting one for this packet rate. However, at a certain PR there is a switchover point at which receiver B becomes the limiting one. The switchover point can be identified by setting Max_media_BR_A equal to Max_media_BR_B and breaking out PR:

$$PR = \frac{\text{TMMBR\_max total BR\_A} - \text{TMMBR\_max total BR\_B}}{8*(\text{TMMBR\_OH\_A} - \text{TMMBR\_OH\_B})} \quad \ldots (3)$$

which, for the numbers above yields 31.25 as the switchover point between the two limits. That is, for packet rates below 31.25 per second, receiver A is the limiting receiver, and for higher packet rates, receiver B is more limiting. The implications of this behavior have to be considered by implementations that are going to control media encoding and its packetization. As exemplified above, multiple TMMBR limits may apply to the trade-off between net media bit rate and packet rate. Which limitation applies depends on the packet rate being considered.

This also has implications for how the TMMBR mechanism needs to work. First, there is the possibility that multiple TMMBR tuples are providing limitations on the media sender. Secondly there is a need for any session participant (media sender and receivers) to be able to determine if a given tuple will become a limitation upon the media sender, or if the set of already given limitations is stricter than the given values. In the absence of the ability to make this determination the suppression of TMMBR requests would not work.

The basic idea of the algorithm is as follows. Each TMMBR tuple can be viewed as the equation of a straight line (cf. equations (1) and (2)) in a space where packet rate lies along the X-axis and maximum bit rate lies along the Y-axis. The lower envelope of the set of lines corresponding to the complete set of TMMR tuples, together with the X and Y axes, defines a polygon. Points lying within this polygon are combinations of packet rate and bit rate that meet all of the TMMBR constraints. The highest feasible packet rate within

this region is the minimum of the rate at which the bounding polygon meets the X-axis or the session maximum packet rate (SMAXPR, measured in packets per second) provided by signaling, if any. Typically a media sender will prefer to operate at a lower rate than this theoretical maximum, so as to increase the rate at which actual media content reaches the receivers. The purpose of the algorithm is to distinguish the TMMBR tuples constituting the bounding set and thus delineate the feasible region, so that the media sender can select its preferred operating point within that region

Figure 1 below shows a bounding polygon formed by TMMBR tuples A and B. A third tuple C lies outside the bounding polygon and is therefore irrelevant in determining feasible tradeoffs between media rate and packet rate. The line labeled ss..s represents the limit on packet rate imposed by the session maximum packet rate (SMAXPR) obtained by signaling during session setup. In Figure 1 the limit determined by tuple B happens to be more restrictive than SMAXPR. The situation could easily be the reverse, meaning that the bounding polygon is terminated on the right by the vertical line representing the SMAXPR constraint.

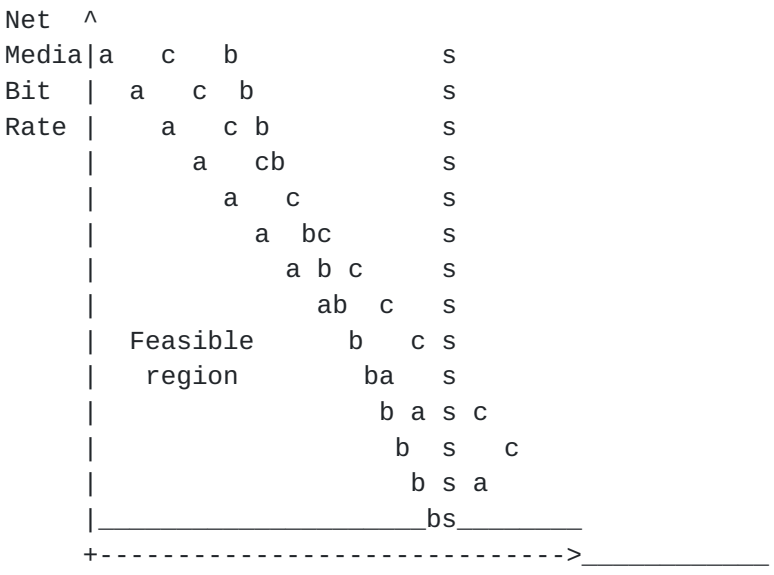


Figure 1 - Geometric Interpretation of TMMBR Tuples

Note that the slopes of the lines making up the bounding polygon are increasingly negative as one moves in the direction of increasing

packet rate. Note also that with slight rearrangement, equations (1) and (2) have the canonical form:

$$y = mx + b$$

where
m is the slope and has value equal to the negative of the tuple overhead (in bits),
and
b is the y-intercept and has value equal to the tuple maximum total media bit rate.

These observations lead to the conclusion that when processing the TMMBR tuples to select the initial bounding set, one should sort and process the tuples by order of increasing overhead. Once a particular tuple has been added to the bounding set, all tuples not already selected and having lower overhead can be eliminated, because the next side of the bounding polygon has to be steeper (i.e. the corresponding TMMBR must have higher overhead) than the latest added tuple.

Line cc..c in Figure 1 illustrates another principle. This line is parallel to line aa..a, but has a higher Y-intercept. That is, the corresponding TMMBR tuple contains a higher maximum total media bit rate value. Since line cc..c is outside the bounding polygon, it illustrates the conclusion that if two TMMBR tuples have the same overhead value, the one with higher maximum total media bit rate value cannot be part of the bounding set and can be set aside.

Two further observations complete the algorithm. Obviously, moving from the left, the successive corners of the bounding polygon (i.e. the intersection points between successive pairs of sides) lie at successively higher packet rates. On the other hand, again moving from the left, each successive line making up the bounding set crosses the X-axis at a lower packet rate.

The complete algorithm can now be specified. The algorithm works with two lists of TMMBR tuples, the candidate list X and the selected list Y, both ordered by increasing overhead value. The algorithm terminates when all members of X have been discarded or removed for processing. Membership of the selected list Y is probationary until the algorithm is complete. Each member of the selected list is associated with an intersection value, which is the packet rate at which the line corresponding to that TMMBR tuple intersects with the line corresponding to the previous TMMBR tuple in the selected list. Each member of the selected list is also associated with a maximum packet rate value, which is the lesser of

the session maximum packet rate SMAXPR (if any) and the packet rate at which the line corresponding to that tuple crosses the X-axis.

When the algorithm terminates, the selected list is equal to the bounding set as defined in section 2.2.

Initial Algorithm

This algorithm is used by the media sender when it has received one or more TMMBR requests and before it has determined a bounding set for the first time.

1. Sort the TMMBR tuples by order of increasing overhead. This is the initial candidate list X.

2. When multiple tuples in the candidate list have the same overhead value, discard all but the one with the lowest maximum total media bit rate value.

3. Select and remove from the candidate list the TMMBR tuple with the lowest maximum total media bit rate value. If there is more than one tuple with that value, choose the one with the highest overhead value. This is the first member of the selected list Y. Set its intersection value equal to zero. Calculate its maximum packet rate as the minimum of SMAXPR (if available) and the value obtained from the following formula, which is the packet rate at which the corresponding line crosses the X-axis.

    Max PR = TMMBR max total BR / (8 * TMMBR OH) ... (4)

4. Discard from the candidate list all tuples with a lower overhead value than the selected tuple.

5. Remove the first remaining tuple from the candidate list for processing. Call this the current candidate.

6. Calculate the packet rate PR at the intersection of the line generated by the current candidate with the line generated by the last tuple in the selected list Y, using equation (3).

7. If the calculated value PR is equal to or lower than the intersection value stored for the last tuple of the selected list, discard the last tuple of the selected list and go back to step 6 (retaining the same current candidate).

   Note that the choice of the initial member of the selected list Y in step 3 guarantees that the selected list will never be emptied by this process, meaning that the algorithm must eventually (if

not immediately) fall through to the step 8.

8. (This step is reached when the calculated PR value of the current candidate is greater than the intersection value of the current last member of the selected list Y.) If the calculated value PR of the current candidate is lower than the maximum packet rate associated with the last tuple in the selected list, add the current candidate tuple to the end of the selected list. Store PR as its intersection value. Calculate its maximum packet rate as the lesser of SMAXPR (if available) and the maximum packet rate calculated using equation (4).

9. If any tuples remain in the candidate list, go back to step 5.

Incremental Algorithm

The previous algorithm covered the initial case, where no selected list had previously been created. It also applied only to the media sender. When a previously-created selected list is available at either the media sender or media receiver, two other cases can be considered:

- when a TMMBR tuple not currently in the selected list is a candidate for addition;
- when the values change in a TMMBR tuple currently in the selected list.

At the media receiver these cases correspond respectively to those of the non-owner and owner of a tuple in the TMMBN-reported bounding set.

In either case, the process of updating the selected list to take account of the new/changed tuple can use the basic algorithm described above, with the modification that the initial candidate set consists only of the existing selected list and the new or changed tuple. Some further optimization is possible (beyond starting with a reduced candidate set) by taking advantage of the following observations.

The first observation is that if the new/changed candidate becomes part of the new selected list, the result may be to cause zero or more other tuples to be dropped from the list. However, if more than one other tuple is dropped, the dropped tuples will be consecutive. This can be confirmed geometrically by visualizing a new line that cuts off a series of segments from the previously-existing bounding polygon. The cut-off segments are connected one to the next, the geometric equivalent of consecutive tuples in a

list ordered by overhead value. Beyond the dropped set in either direction all of the tuples that were in the earlier selected list will be in the updated one. The second observation is that, leaving aside the new candidate, the order of tuples remaining in the updated selected list is unchanged because their overhead values have not changed.

The consequence of these two observations is that, once the placement of the new candidate and the extent of the dropped set of tuples (if any) has been determined, the remaining tuples can be copied directly from the candidate list into the selected list, preserving their order. This conclusion suggests the following modified algorithm:

- Run steps 1-4 of the basic algorithm.

- If the new candidate has survived steps 2 and 4 and has become the new first member of the selected list, run steps 5-9 on subsequent candidates until another candidate is added to the selected list. Then move all remaining candidates to the selected list, preserving their order.

- If the new candidate has survived steps 2 and 4 and has not become the new first member of the selected list, start by moving all tuples in the candidate list with lower overhead values than that of the new candidate to the selected list, preserving their order. Run steps 5 through 9 for the new candidate, with the modification that the intersection values and maximum packet rates for the tuples on the selected list have to be calculated on the fly because they were not previously stored. Continue processing only until a subsequent tuple has been added to the selected list, then move all remaining candidates to the selected list, preserving their order.

  Note that the new candidate could be added to the selected list only to be dropped again when the next tuple is processed. It can easily be seen that in this case the new candidate does not displace any of the earlier tuples in the selected list. The limitations of ASCII art make this difficult to show in a figure. Line cc..c in Figure 1 would be an example if it had a steeper slope (tuple C had a higher overhead value), but still intersected line aa..a beyond where line aa..a intersects line bb..b.

The algorithm just described is approximate, because it does not take account of tuples outside the selected list. To see how such tuples can become relevant, consider Figure 1 and suppose that the

maximum total media bit rate in tuple A increases to the point that line aa..a moves outside line cc..c. Tuple A will remain in the bounding set calculated by the media sender. However, once it issues a new TMMBN, media receiver C will apply the algorithm and discover that its tuple C should now enter the bounding set. It will issue a TMMBR request to the media sender, which will repeat its calculation and come to the appropriate conclusion.

The rules of section 4.2 require that the media sender refrain from raising its sending rate until media receivers have had a chance to respond to the TMMBN. In the example just given, this delay ensures that the relaxation of tuple A does not actually result in an attempt to send media at a rate exceeding the capacity at C.

3.5.4.3. Use of TMMBR in a Mixer Based Multipoint Operation

Assume a small mixer-based multiparty conference is ongoing, as depicted in Topo-Mixer of [Topologies]. All participants have negotiated a common maximum bit rate that this session can use. The conference operates over a number of unicast paths between the participants and the mixer. The congestion situation on each of these paths can be monitored by the participant in question and by the mixer, utilizing, for example, RTCP receiver reports (RR) or the transport protocol, e.g. DCCP [RFC4340]. However, any given participant has no knowledge of the congestion situation of the connections to the other participants. Worse, without mechanisms similar to the ones discussed in this draft, the mixer (which is aware of the congestion situation on all connections it manages) has no standardized means to inform media senders to slow down, short of forging its own receiver reports (which is undesirable). In principle, a mixer confronted with such a situation is obliged to thin or transcode streams intended for connections that detected congestion.

In practice, unfortunately, media-aware streaming thinning is a very difficult and cumbersome operation and adds undesirable delay. If media-unaware, it leads very quickly to unacceptable reproduced media quality. Hence, a means to slow down senders even in the absence of congestion on their connections to the mixer is desirable.

To allow the mixer to throttle traffic on the individual links, without performing transcoding, there is a need for a mechanism that enables the mixer to ask a participant's media encoders to limit the media stream bit rate they are currently generating. TMMBR provides the required mechanism. When the mixer detects congestion between itself and a given participant, it executes the following procedure:

1. It starts thinning the media traffic to the congested participant to the supported bit rate.

2. It uses TMMBR to request the media sender(s) to reduce the total media bit rate sent by them to the mixer, to a value that is in compliance with congestion control principles for the slowest link. Slow refers here to the available bandwidth / bit rate / capacity and packet rate after congestion control.

3. As soon as the bit rate has been reduced by the sending part, the mixer stops stream thinning implicitly, because there is no need for it once the stream is in compliance with congestion control.

This use of stream thinning as an immediate reaction tool followed up by a quick control mechanism appears to be a reasonable compromise between media quality and the need to combat congestion.

3.5.4.4. Use of TMMBR in Point-to-Multipoint Using Multicast or Translators

In these topologies, corresponding to Topo-Multicast or Topo-Translator, RTCP RRs are transmitted globally. This allows all participants to detect transmission problems such as congestion, on a medium timescale. As all media senders are aware of the congestion situation of all media receivers, the rationale for the use of TMMBR in the previous section does not apply. However, even in this case the congestion control response can be improved when the unicast links are using congestion controlled transport protocols (such as TCP or DCCP). A peer may also report local limitations to the media sender.

3.5.4.5. Use of TMMBR in Point-to-point operation

In use case 7 it is possible to use TMMBR to improve the performance when the known upper limit of the bit rate changes. In this use case the signaling protocol has established an upper limit for the session and total media bit rates. However, at the time of transport link bit rate reduction, a receiver can avoid serious congestion by sending a TMMBR to the sending side. Thus, TMMBR is useful for putting restrictions on the application and thus placing the congestion control mechanism in the right ballpark. However, TMMBR is usually unable to provide the continuously quick feedback loop required for real congestion control. Nor do its semantics match those of congestion control given its different purpose. For these reasons TMMBR SHALL NOT be used as a substitute for congestion control.

3.5.4.6. Reliability

The reaction of a media sender to the reception of a TMMBR message is not immediately identifiable through inspection of the media stream. Therefore, a more explicit mechanism is needed to avoid unnecessary re-sending of TMMBR messages. Using a statistically based retransmission scheme would only provide statistical guarantees of the request being received. It would also not avoid the retransmission of already received messages. In addition, it would not allow for easy suppression of other participants' requests. For these reasons, a mechanism based on explicit notification is used.

Upon the reception of a request a media sender sends a TMMBN notification containing the current bounding set, and indicating which session participants own that limit. In multicast scenarios, that allows all other participants to suppress any request they may have, if their limitations are less strict than the current ones (i.e. define lines lying outside the feasible region as defined in section 2.2). Keeping and notifying only the bounding set of tuples allows for small message sizes and media sender states. A media sender only keeps state for the SSRCs of the current owners of the bounding set of tuples; all other requests and their sources are not saved. Once the bounding set has been established, new TMMBR messages should be generated only by owners of the bounding tuples and by other entities that determine (by applying the algorithm of section 3.5.4.2 or its equivalent) that their limitations should now be part of the bounding set.

4. RTCP Receiver Report Extensions

This memo specifies six new feedback messages. The Full Intra Request (FIR), Temporal-Spatial Trade-off Request (TSTR), Temporal-Spatial Trade-off Notification (TSTN), and Video Back Channel Message (VBCM) are "Payload Specific Feedback Messages" as defined in Section 6.3 of AVPF [RFC4585]. The Temporary Maximum Media Stream Bit Rate Request (TMMBR) and Temporary Maximum Media Stream Bit Rate Notification (TMMBN) are "Transport Layer Feedback Messages" as defined in Section 6.2 of AVPF.

The new feedback messages are defined in the following subsections, following a similar structure to that in sections 6.2 and 6.3 of the AVPF specification [RFC4585].

4.1. Design Principles of the Extension Mechanism

RTCP was originally introduced as a channel to convey presence, reception quality statistics and hints on the desired media coding. A limited set of media control mechanisms were introduced in early RTP payload formats for video formats, for example in RFC 2032 [RFC2032]. However, this specification, for the first time, suggests a two-way handshake for some of its messages. There is danger that this introduction could be misunderstood as a precedent for the use of RTCP as an RTP session control protocol. To prevent such a misunderstanding, this subsection attempts to clarify the scope of the extensions specified in this memo, and strongly suggests that future extensions follow the rationale spelled out here, or compellingly explain why they divert from the rationale.

In this memo, and in AVPF [RFC4585], only such messages have been included as:

a) have comparatively strict real-time constraints, which prevent the use of mechanisms such as a SIP re-invite in most application scenarios. The real-time constraints are explained separately for each message where necessary.

b) are multicast-safe in that the reaction to potentially contradicting feedback messages is specified, as necessary for each message; and

c) are directly related to activities of a certain media codec, class of media codecs (e.g. video codecs), or a given RTP packet stream.

In this memo, a two-way handshake is introduced only for messages for which:

a) a notification or acknowledgement is required due to their nature. An analysis to determine whether this requirement exists has been performed separately for each message.

b) the notification or acknowledgement cannot be easily derived from the media bit stream.

All messages in AVPF [RFC4585] and in this memo present their contents in a simple, fixed binary format. This accommodates media receivers which have not implemented higher control protocol functionalities (SDP, XML parsers and such) in their media path.

Messages that do not conform to the design principles just described are not an appropriate use of RTCP or of the Codec Control Framework defined in this document.

4.2. Transport Layer Feedback Messages

As specified in section 6.1 of RFC 4585 [RFC4585], Transport Layer Feedback messages are identified by the RTCP packet type value RTPFB (205).

In AVPF, one message of this category had been defined. This memo specifies two more such messages. They are identified by means of the FMT parameter as follows:

Assigned in AVPF [RFC4585]:

1: Generic NACK
31: reserved for future expansion of the identifier number space

Assigned in this memo:

2: reserved (see note below)
3: Temporary Maximum Media Stream Bit Rate Request (TMMBR)
4: Temporary Maximum Media Stream Bit Rate Notification (TMMBN)

Note: early drafts of AVPF [RFC4585] reserved FMT=2 for a code point that has later been removed. It has been pointed out that there may be implementations in the field using this value in accordance with the expired draft. As there is sufficient numbering space available, we mark FMT=2 as

reserved so to avoid possible interoperability problems with any such early implementations.

Available for assignment:

0: unassigned
5-30: unassigned

The following subsection defines the formats of the Feedback Control Information (FCI) entries for the TMMBR and TMMBN messages respectively and specify the associated behaviour at the media sender and receiver.

4.2.1. Temporary Maximum Media Stream Bit Rate Request (TMMBR)

The Temporary Maximum Media Stream Bit Rate Request is identified by RTCP packet type value PT=RTPFB and FMT=3.

The FCI field of a Temporary Maximum Media Stream Bit-Rate Request (TMMBR) message SHALL contain one or more FCI entries.

4.2.1.1. Message Format

The Feedback Control Information (FCI) consists of one or more TMMBR FCI entries with the following syntax:

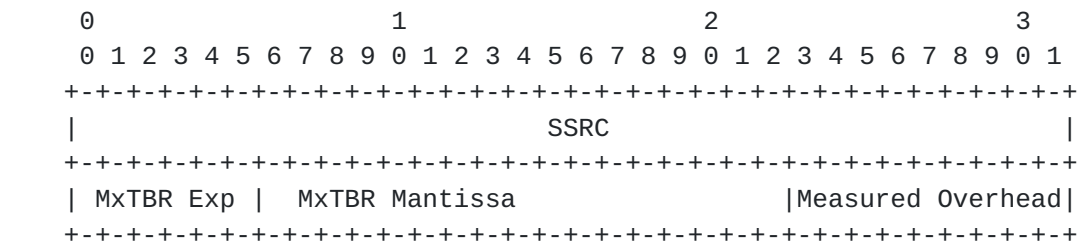


Figure 2 - Syntax of an FCI entry in the TMMBR message

SSRC (32 bits): The SSRC value of the media sender that is requested to obey the new maximum bit rate.

MxTBR Exp (6 bits): The exponential scaling of the mantissa for the maximum total media bit rate value. The value is an unsigned integer [0..63].

MxTBR Mantissa (17 bits): The mantissa of the maximum total media bit rate value as an unsigned integer.

Measured Overhead (9 bits): The measured average packet overhead value in bytes. The measurement SHALL be done according to the description in section 4.2.1.2. The value is an unsigned integer [0..511].

The maximum total media bit rate (MxTBR) value in bits per second is calculated from the MxTBR exponent (exp) and mantissa in the following way:

MxTBR = mantissa * 2^exp

This allows for 17 bits of resolution in the range 0 to 131072*2^63 (approximately 1.2*10^24).

The length of the TMMBR feedback message SHALL be set to 2+2*N where N is the number of TMMBR FCI entries.

4.2.1.2. Semantics

Behaviour at the Media Receiver (Sender of the TMMBR)

TMMBR is used to indicate a transport related limitation at the reporting entity acting as a media receiver. TMMBR has the form of a tuple containing two components. The first value is the highest bit rate per sender of a media stream, available at a receiver-chosen protocol layer, which the receiver currently supports in this RTP session. The second value is the measured header overhead in bytes as defined in section 2.2 and measured at the chosen protocol layer in the packets received for the stream. The measurement of the overhead is a running average that is updated for each packet received for this particular media source (SSRC), using the following formula:

avg_OH (new) = 15/16*avg_OH (old) + 1/16*pckt_OH,

where avg_OH is the running (exponentially smoothed) average and pckt_OH is the overhead observed in the latest packet.

If a maximum bit rate has been negotiated through signaling, the maximum total media bit rate that the receiver reports in a TMMBR message MUST NOT exceed the negotiated value converted to a common basis (i.e. with overheads adjusted to bring it to the same reference protocol layer).

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of the packet sender" field indicates the source of the request, and the "SSRC of media source" is not used and SHALL be set to 0. Within a particular TMMBR FCI entry, the "SSRC of media sender" in the FCI field denotes the media sender the tuple applies to. This is useful in the multicast or translator topologies where the reporting entity may address all of the media senders in a single TMMBR message using multiple FCI entries.

The media receiver SHALL save the contents of the latest TMMBN message received from each media sender.

The media receiver MAY send a TMMBR FCI entry to a particular media sender under the following circumstances:

- before any TMMBN message has been received from that media sender;
- when the media receiver has been identified as the source of a bounding tuple within the latest TMMBN message received from that media sender, and the value of the maximum total media bit rate or the overhead relating to that media sender has changed;
- when the media receiver has not been identified as the source of a bounding tuple within the latest TMMBN message received from that media sender, and, after the media receiver applies the incremental algorithm from section 3.5.4.2 or a stricter equivalent, the media receiver's tuple relating to that media sender is determined to belong to the bounding set.

A TMMBR FCI entry MAY be repeated in subsequent TMMBR messages if no Temporary Maximum Media Stream Bit-Rate Notification (TMMBN) FCI has been received from the media sender at the time of transmission of the next RTCP packet. The bit rate value of a TMMBR FCI entry MAY be changed from one TMMBR message to the next. The overhead measurement SHALL be updated to the current value of avg_OH each time the entry is sent.

If the value set by a TMMBR message is expected to be permanent, the TMMBR setting party SHOULD renegotiate the session parameters to reflect that using session setup signaling, e.g. a SIP re-invite.

Behaviour at the Media Sender (Receiver of the TMMBR)

When it receives a TMMBR message containing an FCI entry relating to it, the media sender SHALL use an initial or incremental algorithm

as applicable to determine the bounding set of tuples based on the new information. The algorithm used SHALL be at least as strict as the corresponding algorithm defined in section 3.5.4.2. The media sender MAY accumulate TMMBR requests over a small interval (relative to the RTCP sending interval) before making this calculation.

Once it has determined the bounding set of tuples, the media sender MAY use any combination of packet rate and net media bit rate within the feasible region that these tuples describe to produce a lower total media stream bit rate, as it may need to address a congestion situation or other limiting factors. See section 5 (congestion control) for more discussion.

If the media sender concludes that it can increase the maximum total media bit rate value, it SHALL wait before actually doing so, for a period long enough to allow a media receiver to respond to the TMMBN if it determines that its tuple belongs in the bounding set. This delay period is estimated by the formula:

```
2 * RTT + T_Dither_Max,
```

where RTT is the longest round trip time known to the media sender and T_Dither_Max is defined in section 3.4 of [RFC4585]. Even in point-to-point sessions a media sender MUST obey to the aforementioned rule, as it is not guaranteed that a participant is able to determine correctly whether all the sources are co-located in a single node, and are coordinated.

A TMMBN message SHALL be sent by the media sender at the earliest possible point in time, in response to any TMMBR messages received since the last sending of TMMBN. The TMMBN message indicates the calculated set of bounding tuples and the owners of those tuples at the time of the transmission of the message.

An SSRC may time out according to the default rules for RTP session participants, i.e. the media sender has not received any RTP or RTCP packets from the owner for the last five regular reporting intervals. An SSRC may also explicitly leave the session, with the participant indicating this through the transmission of an RTCP BYE packet or using an external signaling channel. If the media sender determines that the owner of a tuple in the bounding set has left the session, the media sender SHALL transmit a new TMMBN containing the previously-determined set of bounding tuples but with the tuple belonging to the departed owner removed.

A media sender MAY proactively initiate the equivalent to a TMMBR message to itself, when it is aware that its transmission path is more restrictive than the current limitations. As a result, a TMMBN

indicating the media source itself as the owner of a tuple is being sent, thereby avoiding unnecessary TMMBR messages from other participants. However, like any other participant, when the media sender becomes aware of changed limitations, it is required to change the tuple, and to send a corresponding TMMBN.

Discussion

Due to the unreliable nature of transport of TMMBR and TMMBN, the above rules may lead to the sending of TMMBR messages which appear to disobey those rules. Furthermore, in multicast scenarios it can happen that more than one "non-owning" session participant may determine, rightly or wrongly, that its tuple belongs in the bounding set. This is not critical for a number of reasons:

a) If a TMMBR message is lost in transmission, either the media sender sends a new TMMBN message in response to some other media receiver or it does not send a new TMMBN message at all. In the first case, the media receiver applies the incremental algorithm and, if it determines that its tuple should be part of the bounding set, sends out another TMMBR. In the second case, it repeats the sending of a TMMBR unconditionally. Either way, the media sender eventually gets the information it needs.

b) Similarly, if a TMMBN message gets lost, the media receiver that has sent the corresponding TMMBR request does not receive the notification and is expected to re-send the request and trigger the transmission of another TMMBN.

c) If multiple competing TMMBR messages are sent by different session participants, then the algorithm can be applied taking all of these messages into account, and the resulting TMMBN provides the participants with an updated view of how their tuples compare with the bounded set.

d) If more than one session participant happens to send TMMBR messages at the same time and with the same tuple component values, it does not matter which of those tuples is taken into the bounding set. The losing session participant will determine, after applying the algorithm, that its tuple does not enter the bounding set, and will therefore stop sending its TMMBR request.

It is important to consider the security risks involved with faked TMMBRs. See the security considerations in Section 6.

As indicated already, the feedback messages may be used in both multicast and unicast sessions in any of the specified topologies. However, for sessions with a large number of participants, using the

lowest common denominator, as required by this mechanism, may not be the most suitable course of action. Large sessions may need to consider other ways to adapt the bit rate to participants' capabilities, such as partitioning the session into different quality tiers, or using some other method of achieving bit rate scalability.

4.2.1.3. Timing Rules

The first transmission of the TMMBR request message MAY use early or immediate feedback in cases when timeliness is desirable. Any repetition of a request message SHOULD use regular RTCP mode for its transmission timing.

4.2.1.4. Handling in Translator and Mixers

Media translators and mixers will need to receive and respond to TMMBR messages as they are part of the chain that provides a certain media stream to the receiver. The mixer or translator may act locally on the TMMBR request and thus generate a TMMBN to indicate that it has done so. Alternatively, in the case of a media translator it can forward the request, or in the case of a mixer generate one of its own and pass it forward. In the latter case, the mixer will need to send a TMMBN back to the original requestor to indicate that it is handling the request.

4.2.2. Temporary Maximum Media Stream Bit Rate Notification (TMMBN)

The Temporary Maximum Media Stream Bit Rate Notification is identified by RTCP packet type value PT=RTPFB and FMT=4.

The FCI field of the TMMBN Feedback message may contain zero, one or more TMMBN FCI entries.

4.2.2.1. Message Format

The Feedback Control Information (FCI) consists of zero, one or more TMMBN FCI entries with the following syntax:

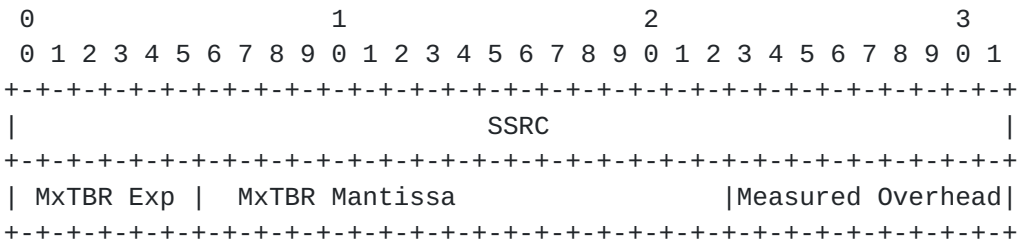


Figure 3 - Syntax of an FCI entry in the TMMBN message

SSRC (32 bits): The SSRC value of the "owner" of this tuple.

MxTBR Exp (6 bits): The exponential scaling of the mantissa for the maximum total media bit rate value. The value is an unsigned integer [0..63].

MxTBR Mantissa (17 bits): The mantissa of the maximum total media bit rate value as an unsigned integer.

Measured Overhead (9 bits): The measured average packet overhead value in bytes represented as an unsigned integer [0..511].

Thus, the FCI within the TMMBN message contains entries indicating the bounding tuples. For each tuple, the entry gives the owner by the SSRC, followed by the applicable maximum total media bit rate and overhead value.

The length of the TMMBN message SHALL be set to 2+2*N where N is the number of TMMBN FCI entries.

4.2.2.2. Semantics

This feedback message is used to notify the senders of any TMMBR message that one or more TMMBR messages have been received or that an owner has left the session. It indicates to all participants the current set of bounding tuples and the "owners" of those tuples.

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of the packet sender" field indicates the source of the notification. The "SSRC of media source" is not used and SHALL be set to 0.

A TMMBN message SHALL be scheduled for transmission after the reception of a TMMBR message with an FCI entry identifying this media sender. Only a single TMMBN SHALL be sent, even if more than one TMMBR message is received between the scheduling of the transmission and the actual transmission of the TMMBN message. The TMMBN message indicates the bounding tuples and their owners at the time of transmitting the message. The bounding tuples included SHALL be the set arrived at through application of the applicable algorithm of section 3.5.4.2 or an equivalent, applied to the

previous bounding set, if any, and tuples received in TMMBR messages since the last TMMBN was transmitted.

The reception of a TMMBR message SHALL still result in the transmission of a TMMBN message even if, after application of the algorithm, the newly reported TMMBR tuple is not accepted into the bounding set. In such a case the bounding tuples and their owners are not changed, unless the TMMBR was from an owner of a tuple within the previously calculated bounding set. This procedure allows session participants that did not see the last TMMBN message to get a correct view of this media sender's state.

As indicated in section 4.2.1.2, when a media sender determines that an "owner" of a bounding tuple has left the session, then that tuple is removed from the bounding set, and the media sender SHALL send a TMMBN message indicating the remaining bounding tuples. If there are no remaining bounding tuples a TMMBN without any FCI SHALL be sent to indicate this. Without a remaining bounding tuple, the maximum media bit rate and maximum packet rate negotiated in session signaling, if any, apply.

> Note: if any media receivers remain in the session, this last will be a temporary situation. The empty TMMBN will cause every remaining media receiver to determine that its limitation belongs in the bounding set and send a TMMBR in consequence.

In unicast scenarios (i.e. where a single sender talks to a single receiver), the aforementioned algorithm to determine ownership degenerates to the media receiver becoming the "owner" of the one bounding tuple as soon as the media receiver has issued the first TMMBR message.

4.2.2.3. Timing Rules

The TMMBN acknowledgement SHOULD be sent as soon as allowed by the applied timing rules for the session. Immediate or early feedback mode SHOULD be used for these messages.

4.2.2.4. Handling by Translators and Mixers

As discussed in Section 4.2.1.4 mixers or translators may need to issue TMMBN messages as responses to TMMBR messages for SSRC's handled by them.

4.3. Payload Specific Feedback Messages

As specified by section 6.1 of RFC 4585 [RFC4585], Payload-Specific FB messages are identified by the RTCP packet type value PSFB (206).

AVPF [RFC4585] defines three payload-specific feedback messages and one application layer feedback message. This memo specifies four additional payload-specific feedback messages. All are identified by means of the FMT parameter as follows:

Assigned in [RFC4585]:

```
  1:     Picture Loss Indication (PLI)
  2:     Slice Lost Indication (SLI)
  3:     Reference Picture Selection Indication (RPSI)
  15:    Application layer FB message
  31:    reserved for future expansion of the number space
```

Assigned in this memo:

```
  4:     Full Intra Request Command (FIR)
  5:     Temporal-Spatial Trade-off Request (TSTR)
  6:     Temporal-Spatial Trade-off Notification (TSTN)
   7:     Video Back Channel Message (VBCM)
```

Unassigned:

```
  0:     unassigned
   8-14:  unassigned
  16-30: unassigned
```

The following subsections define the new FCI formats for the payload-specific feedback messages.

4.3.1. Full Intra Request (FIR)

The FIR message is identified by RTCP packet type value PT=PSFB and FMT=4.

The FCI field MUST contain one or more FIR entries. Each entry applies to a different media sender, identified by its SSRC.

4.3.1.1. Message Format

The Feedback Control Information (FCI) for the Full Intra Request consists of one or more FCI entries, the content of which is depicted in Figure 4. The length of the FIR feedback message MUST be set to 2+2*N, where N is the number of FCI entries.

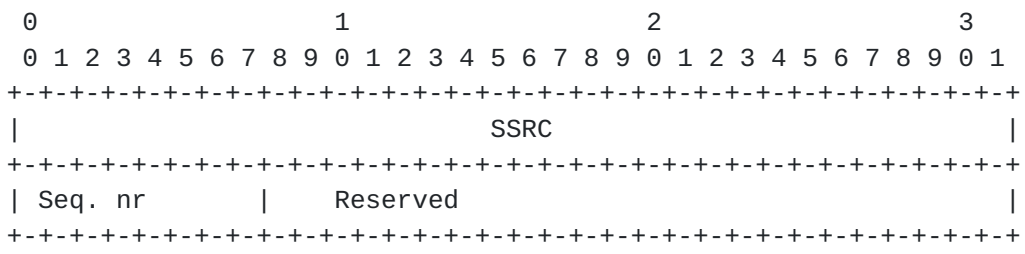


Figure 4 - Syntax of an FCI entry in the FIR message

SSRC (32 bits): The SSRC value of the media sender which is requested to send a decoder refresh point.

Seq. nr (8 bits): Command sequence number. The sequence number space is unique for each pairing of the SSRC of command source and the SSRC of the command target. The sequence number SHALL be increased by 1 modulo 256 for each new command. A repetition SHALL NOT increase the sequence number. The initial value is arbitrary.

Reserved (24 bits): All bits SHALL be set to 0 by the sender and SHALL be ignored on reception.

The semantics of this feedback message is independent of the RTP payload type.

4.3.1.2. Semantics

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of the packet sender" field indicates the source of the request, and the "SSRC of media source" is not used and SHALL be set to 0. The SSRCs of the media senders to which the FIR command applies are in the corresponding FCI entries. A FIR message MAY contain requests to multiple media senders, using one FCI entry per target media sender.

Upon reception of FIR, the encoder MUST send a decoder refresh point (see section 2.2) as soon as possible.

The sender MUST consider congestion control as outlined in section 5, which MAY restrict its ability to send a decoder refresh point quickly.

FIR SHALL NOT be sent as a reaction to picture losses -- it is RECOMMENDED to use PLI [RFC4585] instead. FIR SHOULD be used only in situations where not sending a decoder refresh point would render the video unusable for the users.

A typical example where sending FIR is appropriate is when, in a multipoint conference, a new user joins the session and no regular decoder refresh point interval is established. Another example would be a video switching MCU that changes streams. Here, normally, the MCU issues a FIR to the new sender so to force it to emit a decoder refresh point. The decoder refresh point normally includes a Freeze Picture Release (defined outside this specification), which re-starts the rendering process of the receivers. Both techniques mentioned are commonly used in MCU-based multipoint conferences.

Other RTP payload specifications such as RFC 2032 [RFC2032] already define a feedback mechanism for certain codecs. An application supporting both schemes MUST use the feedback mechanism defined in this specification when sending feedback. For backward compatibility reasons such an application SHOULD also be capable of receiving and reacting to the feedback scheme defined in the respective RTP payload format, if this is required by that payload format.

4.3.1.3. Timing Rules

The timing follows the rules outlined in section 3 of [RFC4585]. FIR commands MAY be used with early or immediate feedback. The FIR feedback message MAY be repeated. If using immediate feedback mode the repetition SHOULD wait at least one RTT before being sent. In early or regular RTCP mode the repetition is sent in the next regular RTCP packet.

4.3.1.4. Handling of FIR Message in Mixer and Translators

A media translator or a mixer performing media encoding of the content for which the session participant has issued a FIR is responsible for acting upon it. A mixer acting upon a FIR SHOULD NOT forward the message unaltered; instead it SHOULD issue a FIR itself.

4.3.1.5. Remarks

Currently, video appears to be the only useful application for FIR, as it appears to be the only RTP payload widely deployed that relies heavily on media prediction across RTP packet boundaries. However, use of FIR could also reasonably be envisioned for other media types that share essential properties with compressed video, namely cross-frame prediction (whatever a frame may be for that media type). One possible example may be the dynamic updates of MPEG-4 scene descriptions. It is suggested that payload formats for such media types refer to FIR and other message types defined in this specification and in AVPF [RFC4585], instead of creating similar mechanisms in the payload specifications. The payload specifications may have to explain how the payload-specific terminologies map to the video-centric terminology used herein.

In conjunction with video codecs, FIR messages typically trigger the sending of full intra or IDR pictures. Both are several times larger then predicted (inter) pictures. Their size is independent of the time they are generated. In most environments, especially when employing bandwidth-limited links, the use of an intra picture implies an allowed delay that is a significant multiple of the typical frame duration. An example: if the sending frame rate is 10 fps, and an intra picture is assumed to be 10 times as big as an inter picture, then a full second of latency has to be accepted. In such an environment there is no need for a particularly short delay in sending the FIR message. Hence, waiting for the next possible time slot allowed by RTCP timing rules as per [RFC4585] should not have an overly negative impact on the system performance.

Mandating a maximum delay for completing the sending of a decoder refresh point would be desirable from an application viewpoint, but is problematic from a congestion control point of view. "As soon as possible" as mentioned above appears to be a reasonable compromise.

In environments where the sender has no control over the codec (e.g. when streaming pre-recorded and pre-coded content), the reaction to this command cannot be specified. One suitable reaction of a sender would be to skip forward in the video bit stream to the next decoder refresh point. In other scenarios, it may be preferable not to react to the command at all, e.g. when streaming to a large multicast group. Other reactions may also be possible. When deciding on a strategy, a sender could take into account factors such as the size of the receiving group, the "importance" of the sender of the FIR message (however "importance" may be defined in this specific application), the frequency of decoder refresh points in the content, and so on. However, a session which predominately handles pre-coded content is not expected to use FIR at all.

The relationship between the Picture Loss Indication and FIR is as follows. As discussed in section 6.3.1 of AVPF [RFC4585], a Picture Loss Indication informs the decoder about the loss of a picture and hence the likelihood of misalignment of the reference pictures between the encoder and decoder. Such a scenario is normally related to losses in an ongoing connection. In point-to-point scenarios, and without the presence of advanced error resilience tools, one possible option for an encoder consists in sending a decoder refresh point. However, there are other options. One example is that the media sender ignores the PLI, because the embedded stream redundancy is likely to clean up the reproduced picture within a reasonable amount of time. The FIR, in contrast, leaves a (real-time) encoder no choice but to send a decoder refresh point. It does not allow the encoder to take into account any considerations such as the ones mentioned above.

4.3.2. Temporal-Spatial Trade-off Request (TSTR)

The TSTR feedback message is identified by RTCP packet type value PT=PSFB and FMT=5.

The FCI field MUST contain one or more TSTR FCI entries.

4.3.2.1. Message Format

The content of the FCI entry for the Temporal-Spatial Trade-off Request is depicted in Figure 5. The length of the feedback message MUST be set to 2+2*N, where N is the number of FCI entries included.

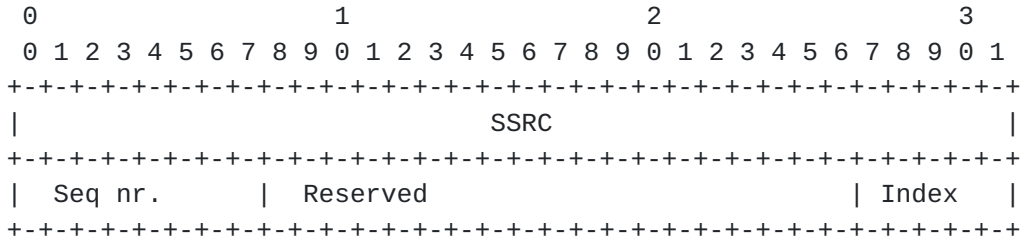


Figure 5 - Syntax of an FCI Entry in the TSTR Message

SSRC (32 bits): The SSRC of the media sender which is requested to apply the tradeoff value given in Index.

Seq. nr (8 bits): Request sequence number. The sequence number space is unique for pairing of the SSRC of request source and the SSRC of the request target. The sequence number

SHALL be increased by 1 modulo 256 for each new command. A repetition SHALL NOT increase the sequence number. The initial value is arbitrary.

Reserved (19 bits): All bits SHALL be set to 0 by the sender and SHALL be ignored on reception.

Index (5 bits): An integer value between 0 and 31 that indicates the relative trade-off that is requested. An index value of 0 indicates highest possible spatial quality, while 31 indicates highest possible temporal resolution.

4.3.2.2. Semantics

A decoder can suggest a temporal-spatial trade-off level by sending a TSTR message to an encoder. If the encoder is capable of adjusting its temporal-spatial trade-off, it SHOULD take into account the received TSTR message for future coding of pictures. A value of 0 suggests a high spatial quality and a value of 31 suggests a high frame rate. The progression of values from 0 to 31 indicate monotonically a desire for higher frame rate. The index values do not correspond to precise values of spatial quality or frame rate.

The reaction to the reception of more than one TSTR message by a media sender from different media receivers is left open to the implementation. The selected trade-off SHALL be communicated to the media receivers by the means of the TSTN message.

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of the packet sender" field indicates the source of the request, and the "SSRC of media source" is not used and SHALL be set to 0. The SSRCs of the media senders to which the TSTR applies are in the corresponding FCI entries.

A TSTR message MAY contain requests to multiple media senders, using one FCI entry per target media sender.

4.3.2.3. Timing Rules

The timing follows the rules outlined in section 3 of [RFC4585]. This request message is not time critical and SHOULD be sent using regular RTCP timing. Only if it is known that the user interface requires quick feedback, the message MAY be sent with early or immediate feedback timing.

4.3.2.4. Handling of message in Mixers and Translators

A mixer or media translator that encodes content sent to the session participant issuing the TSTR SHALL consider the request to determine if it can fulfill it by changing its own encoding parameters. A media translator unable to fulfill the request MAY forward the request unaltered towards the media sender. A mixer encoding for multiple session participants will need to consider the joint needs of these participants before generating a TSTR on its own behalf towards the media sender. See also the discussion in Section 3.5.2.

4.3.2.5. Remarks

The term "spatial quality" does not necessarily refer to the resolution as measured by the number of pixels the reconstructed video is using. In fact, in most scenarios the video resolution stays constant during the lifetime of a session. However, all video compression standards have means to adjust the spatial quality at a given resolution, often influenced by the Quantizer Parameter or QP. A numerically low QP results in a good reconstructed picture quality, whereas a numerically high QP yields a coarse picture. The typical reaction of an encoder to this request is to change its rate control parameters to use a lower frame rate and a numerically lower (on average) QP, or vice versa. The precise mapping of Index value to frame rate and QP is intentionally left open here, as it depends on factors such as the compression standard employed, spatial resolution, content, bit rate, and so on.

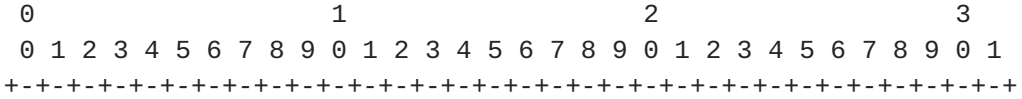
4.3.3. Temporal-Spatial Trade-off Notification (TSTN)

The TSTN message is identified by RTCP packet type value PT=PSFB and FMT=6.

The FCI field SHALL contain one or more TSTN FCI entries.

4.3.3.1. Message Format

The content of an FCI entry for the Temporal-Spatial Trade-off Notification is depicted in Figure 6. The length of the TSTN message MUST be set to 2+2*N, where N is the number of FCI entries.



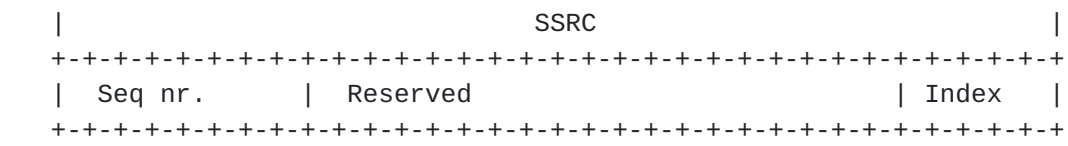


Figure 6 - Syntax of the TSTN

SSRC (32 bits): The SSRC of the source of the TSTR request which resulted in this Notification.

Seq. nr (8 bits): The sequence number value from the TSTR request that is being acknowledged.

Reserved (19 bits): All bits SHALL be set to 0 by the sender and SHALL be ignored on reception.

Index (5 bits): The trade-off value the media sender is using henceforth.

Informative note: The returned trade-off value (Index) may differ from the requested one, for example in cases where a media encoder cannot tune its trade-off, or when pre-recorded content is used.

4.3.3.2. Semantics

This feedback message is used to acknowledge the reception of a TSTR. For each TSTR received targeted at the session participant, a TSTN entry SHALL be sent included in a TSTN feedback message. A single TSTN message MAY acknowledge multiple requests using multiple FCI entries. The index value included SHALL be the same in all FCI entries of the TSTN message. Including a FCI for each requestor allows each requesting entity to determine that the media sender received the request. The Notification SHALL also be sent in response to TSTR repetitions received. If the request receiver has received TSTR with several different sequence numbers from a single requestor it SHALL only respond to the request with the highest (modulo 256) sequence number. Note that the highest sequence number may be a smaller integer value due to the wrapping of the field. Section A.1 of [RFC3550] has an algorithm for keeping track of the highest received sequence number for RTP packets, this could be adapted for this usage.

The TSTN SHALL include the Temporal-Spatial Trade-off index that will be used as a result of the request. This is not necessarily the same index as requested, as the media sender may need to

aggregate requests from several requesting session participants. It may also have some other policies or rules that limit the selection.

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of the packet sender" field indicates the source of the Notification, and the "SSRC of media source" is not used and SHALL be set to 0. The SSRCs of the requesting entities to which the Notification applies are in the corresponding FCI entries.

4.3.3.3. Timing Rules

The timing follows the rules outlined in section 3 of [RFC4585]. This acknowledgement message is not extremely time critical and SHOULD be sent using regular RTCP timing.

4.3.3.4. Handling of TSTN in Mixer and Translators

A mixer or translator that acts upon a TSTR SHALL also send the corresponding TSTN. In cases where it needs to forward a TSTR itself the notification message MAY need to be delayed until the TSTR has been responded to.

4.3.3.5. Remarks

None

4.3.4. H.271 Video Back Channel Message (VBCM)

The VBCM is identified by RTCP packet type value PT=PSFB and FMT=7.

The FCI field MUST contain one or more VBCM FCI entries.

4.3.4.1. Message Format

The syntax of an FCI entry within the VBCM indication is depicted in Figure 7.

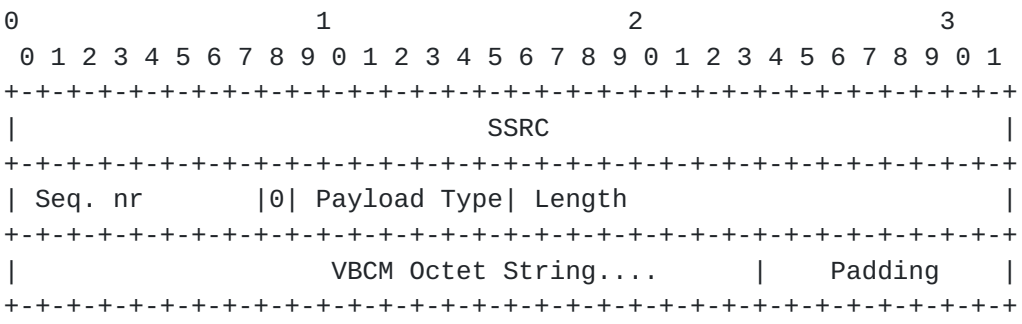


Figure 7 - Syntax of an FCI Entry in the VBCM Message

SSRC (32 bits): The SSRC value of the media sender that is requested to instruct its encoder to react to the VBCM message

Seq. nr (8 bits): Command sequence number. The sequence number space is unique for pairing of the SSRC of command source and the SSRC of the command target. The sequence number SHALL be increased by 1 modulo 256 for each new command. A repetition SHALL NOT increase the sequence number. The initial value is arbitrary.

0: Must be set to 0 by the sender and should not be acted upon by the message receiver.

Payload Type (7 bits): The RTP payload type for which the VBCM bit stream must be interpreted.

Length (16 bits): The length of the VBCM octet string in octets exclusive of any padding octets

VBCM Octet String (Variable length): This is the octet string generated by the decoder carrying a specific feedback sub-message.

Padding (Variable length): Bits set to 0 to make up a 32 bit boundary.

4.3.4.2. Semantics

The "payload" of the VBCM indication carries different types of codec-specific, feedback information. The type of feedback information can be classified as a 'status report' (such as an indication that a bit stream was received without errors, or that a

partial or complete picture or block was lost) or 'update requests' (such as complete refresh of the bit stream).

> Note: There are possible overlaps between the VBCM sub-messages and CCM/AVPF feedback messages, such as FIR. Please see section 3.5.3 for further discussion.

The different types of feedback sub-messages carried in the VBCM are indicated by the "payloadType" as defined in [VBCM]. These sub-message types are reproduced below for convenience. "payloadType", in ITU-T Rec. H.271 terminology, refers to the sub-type of the H.271 message and should not be confused with an RTP payload type.

Payload Type	Message Content
0	One or more pictures without detected bit stream error mismatch
1	One or more pictures that are entirely or partially lost
2	A set of blocks of one picture that is entirely or partially lost
3	CRC for one parameter set
4	CRC for all parameter sets of a certain type
5	A "reset" request indicating that the sender should completely refresh the video bit stream as if no prior bit stream data had been received
> 5	Reserved for future use by ITU-T

Table 2: H.271 message types ("payloadTypes")

The bit string or the "payload" of a VBCM message is of variable length and is self-contained and coded in a variable length, binary format. The media sender necessarily has to be able to parse this optimized binary format to make use of VBCM messages.

Each of the different types of sub-messages (indicated by payloadType) may have different semantics depending on the codec used.

Within the common packet header for feedback messages (as defined in section 6.1 of [RFC4585]), the "SSRC of the packet sender" field indicates the source of the request, and the "SSRC of media source" is not used and SHALL be set to 0. The SSRCs of the media senders to which the VBCM message applies to are in the corresponding FCI

entries. The sender of the VBCM message MAY send H.271 messages to multiple media senders and MAY send more than one H.271 message to the same media sender within the same VBCM message.

4.3.4.3. Timing Rules

The timing follows the rules outlined in section 3 of [RFC4585]. The different sub-message types may have different properties in regards to the timing of messages that should be used. If several different types are included in the same feedback packet then the requirements for the sub-message type with the most stringent requirements should be followed.

4.3.4.4. Handling of message in Mixer or Translator

The handling of VBCM in a mixer or translator is sub-message type dependent.

4.3.4.5. Remarks

Please see section 3.5.3 for a discussion of the usage of H.271 messages and messages defined in AVPF [RFC4585] and this memo with similar functionality.

> Note: There has been some discussion whether the RTP payload type field in this message is needed. It will be needed if there is potentially more than one VBCM-capable RTP payload type in the same session, and the semantics of a given VBCM message changes between payload types. For example, the picture identification mechanism in messages of H.271 type 0 is fundamentally different between H.263 and H.264 (although both use the same syntax). Therefore, the payload field is justified here. There was a further comment that for TSTR and FIR such a need does not exist, because the semantics of TSTR and FIR are either loosely enough defined, or generic enough, to apply to all video payloads currently in existence/envisioned.

5. Congestion Control

The correct application of the AVPF [RFC4585] timing rules prevents the network from being flooded by feedback messages. Hence, assuming a correct implementation and configuration, the RTCP channel cannot break its bit rate commitment and introduce congestion.

The reception of some of the feedback messages modifies the behaviour of the media senders or, more specifically, the media encoders. Thus, modified behaviour MUST respect the bandwidth limits that the application of congestion control provides. For example, when a media sender is reacting to a FIR, the unusually high number of packets that form the decoder refresh point have to be paced in compliance with the congestion control algorithm, even if the user experience suffers from a slowly transmitted decoder refresh point.

A change of the Temporary Maximum Media Stream Bit Rate value can only mitigate congestion, but not cause congestion as long as congestion control is also employed. An increase of the value by a request REQUIRES the media sender to use congestion control when increasing its transmission rate to that value. A reduction of the value results in a reduced transmission bit rate, thus reducing the risk for congestion.

6. Security Considerations

The defined messages have certain properties that have security implications. These must be addressed and taken into account by users of this protocol.

The defined setup signaling mechanism is sensitive to modification attacks that can result in session creation with sub-optimal configuration, and, in the worst case, session rejection. To prevent this type of attack, authentication and integrity protection of the setup signaling is required.

Spoofed or maliciously created feedback messages of the type defined in this specification can have the following implications:

a. severely reduced media bit rate due to false TMMBR messages that sets the maximum to a very low value;

b. assignment of the ownership of a bounding tuple to the wrong participant within a TMMBN message, potentially causing unnecessary oscillation in the bounding set as the mistakenly identified owner reports a change in its tuple and the true owner possibly holds back on changes until a correct TMMBN message reaches the participants;

c. sending TSTR requests that result in a video quality different from the user's desire, rendering the session less useful;

d. sending multiple FIR commands to reduce the frame-rate, and make the video jerky, due to the frequent usage of decoder refresh points.

To prevent these attacks there is a need to apply authentication and integrity protection of the feedback messages. This can be accomplished against threats external to the current RTP session using the RTP profile that combines SRTP [SRTP] and AVPF into SAVPF [SAVPF]. In the mixer cases, separate security contexts and filtering can be applied between the mixer and the participants, thus protecting other users on the mixer from a misbehaving participant.

7. SDP Definitions

Section 4 of [RFC4585] defines a new SDP [RFC4566] attribute, rtcp-fb, that may be used to negotiate the capability to handle specific AVPF commands and indications, such as Reference Picture Selection, Picture Loss Indication etc. The ABNF for rtcp-fb is described in section 4.2 of [RFC4585]. In this section we extend the rtcp-fb attribute to include the commands and indications that are described for codec control in the present document. We also discuss the Offer/Answer implications for the codec control commands and indications.

7.1. Extension of the rtcp-fb Attribute

As described in AVPF [RFC4585], the rtcp-fb attribute indicates the capability of using RTCP feedback. AVPF specifies that the rtcp-fb attribute must only be used as a media level attribute and must not be provided at session level. All the rules described in [RFC4585] for rtcp-fb attribute relating to payload type and to multiple rtcp-fb attributes in a session description also apply to the new feedback messages defined in this memo.

The ABNF [RFC4234] for rtcp-fb as defined in [RFC4585] is

```
"a=rtcp-fb: " rtcp-fb-pt SP rtcp-fb-val CRLF
```

where rtcp-fb-pt is the payload type and rtcp-fb-val defines the type of the feedback message such as ack, nack, trr-int and rtcp-fb-id. For example, to indicate the support of feedback of picture loss indication, the sender declares the following in SDP

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
```

```
s=Media with feedback
t=0 0
c=IN IP4 host.example.com
m=audio 49170 RTP/AVPF 98
a=rtpmap:98 H263-1998/90000
a=rtcp-fb:98 nack pli
```

In this document we define a new feedback value "ccm" which indicates the support of codec control using RTCP feedback messages. The "ccm" feedback value SHOULD be used with parameters that indicate the specific codec control commands supported. In this draft we define four such parameters, namely:

- o "fir" indicates support of the Full Intra Request (FIR).
- o "tmmbr" indicates support of the Temporary Maximum Media Stream Bit Rate Request/Notification (TMMBR/TMMBN). It has an optional sub parameter to indicate the session maximum packet rate (measured in packets per second) to be used. If not included this defaults to infinity.
- o "tstr" indicates support of the Temporal-Spatial Trade-off Request/Notification (TSTR/TSTN).
- O "vbcm" indicates support of H.271 video back channel messages (VBCM). It has zero or more subparameters identifying the supported H.271 "payloadType" values.

In the ABNF for rtcp-fb-val defined in [RFC4585], there is a placeholder called rtcp-fb-id to define new feedback types. "ccm" is defined as a new feedback type in this document and the ABNF for the parameters for ccm are defined here (please refer to section 4.2 of [RFC4585] for complete ABNF syntax).

```
rtcp-fb-param = SP "app" [SP byte-string]
              / SP rtcp-fb-ccm-param
              /     ; empty

rtcp-fb-ccm-param = "ccm" SP ccm-param


ccm-param  = "fir"   ; Full Intra Request
           / "tmmbr" [SP "smaxpr=" MaxPacketRateValue]
                     ; Temporary max media bit rate
           / "tstr"  ; Temporal Spatial Trade Off
           / "vbcm" *(SP subMessageType) ; H.271 VBCM messages
           / token [SP byte-string]
                      ; for future commands/indications
subMessageType = 1*8DIGIT
byte-string = <as defined in section 4.2 of [RFC4585] >
```

```
MaxPacketRateValue = 1*15DIGIT
```

7.2. Offer-Answer

The Offer/Answer [RFC3264] implications for codec control protocol feedback messages are similar to those described in [RFC4585]. The offerer MAY indicate the capability to support selected codec commands and indications. The answerer MUST remove all ccm parameters corresponding to the CCM messages that it does not wish to support in this particular media session (for example because it does not implement the message in question, or because its application logic suggests the support of the message adds no value). The answerer MUST NOT add new ccm parameters in addition to what has been offered. The answer is binding for the media session and both offerer and answerer MUST NOT use any feedback messages other than what both sides have explicitly indicated as being supported. In others words only the joint subset of CCM parameters from the offer and answer may be used.

Note, that including a CCM parameter in an offer or answer indicates that the party (offerer or answerer) is at least capable of receiving the corresponding CCM message(s) and act upon them. In cases when the reception of a negotiated CCM messages mandates the party to respond with another CCM message, it must also have that capability. Although it is not mandated to initiate CCM messages of any negotiated type, it is generally expected that an party will initiate CCM messages when appropriate.

The session maximum packet rate parameter part of the TMMBR indication is declarative and everyone SHALL use the highest value indicated in a response. If the session maximum packet rate parameter is not present in an offer it SHALL NOT be included by the answerer.

7.3. Examples

Example 1: The following SDP describes a point-to-point video call with H.263, with the originator of the call declaring its capability to support the FIR and TSTR/TSTN codec control messages. The SDP is carried in a high level signaling protocol like SIP.

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
s=Point-to-Point call
c=IN IP4 192.0.2.124
m=audio 49170 RTP/AVP 0
```

```
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVPF 98
a=rtpmap:98 H263-1998/90000
a=rtcp-fb:98 ccm tstr
a=rtcp-fb:98 ccm fir
```

In the above example, when the sender receives a TSTR message from the remote party it is capable of adjusting the trade off as indicated in the RTCP TSTN feedback message.

Example 2: The following SDP describes a SIP end point joining a video mixer that is hosting a multiparty video conferencing session. The participant supports only the FIR (Full Intra Request) codec control command and it declares it in its session description.

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
s=Multiparty Video Call
c=IN IP4 192.0.2.124
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVPF 98
a=rtpmap:98 H263-1998/90000
a=rtcp-fb:98 ccm fir
```

When the video MCU decides to route the video of this participant it sends an RTCP FIR feedback message. Upon receiving this feedback message the end point is required to generate a full intra request.

Example 3: The following example describes the Offer/Answer implications for the codec control messages. The Offerer wishes to support "tstr", "fir" and "tmmbr". The offered SDP is

```
-------------> Offer
      v=0
      o=alice 3203093520 3203093520 IN IP4 host.example.com
      s=Offer/Answer
      c=IN IP4 192.0.2.124
      m=audio 49170 RTP/AVP 0
      a=rtpmap:0 PCMU/8000
      m=video 51372 RTP/AVPF 98
      a=rtpmap:98 H263-1998/90000
      a=rtcp-fb:98 ccm tstr
      a=rtcp-fb:98 ccm fir
      a=rtcp-fb:* ccm tmmbr smaxpr=120
```

The answerer wishes to support only the FIR and TSTR/TSTN messages and the answerer SDP is

<---------------- Answer

```
v=0
o=alice 3203093520 3203093524 IN IP4 otherhost.example.com
s=Offer/Answer
c=IN IP4 192.0.2.37
m=audio 47190 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 53273 RTP/AVPF 98
a=rtpmap:98 H263-1998/90000
a=rtcp-fb:98 ccm tstr
a=rtcp-fb:98 ccm fir
```

Example 4: The following example describes the Offer/Answer implications for H.271 Video back channel messages (VBCM). The Offerer wishes to support VBCM and the sub-messages of payloadType 1 (one or more pictures that are entirely or partially lost) and 2 (a set of blocks of one picture that are entirely or partially lost).

-------------> Offer

```
v=0
o=alice 3203093520 3203093520 IN IP4 host.example.com
s=Offer/Answer
c=IN IP4 192.0.2.124
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 51372 RTP/AVPF 98
a=rtpmap:98 H263-1998/90000
a=rtcp-fb:98 ccm vbcm 1 2
```

The answerer only wishes to support sub-messages of type 1 only

<---------------- Answer

```
v=0
o=alice 3203093520 3203093524 IN IP4 otherhost.example.com
s=Offer/Answer
c=IN IP4 192.0.2.37
m=audio 47190 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 53273 RTP/AVPF 98
```

```
a=rtpmap:98 H263-1998/90000
a=rtcp-fb:98 ccm vbcm 1
```

So, in the above example, only VBCM indications comprised of "payloadType" 1 will be supported.

8. IANA Considerations

The new value "ccm" needs to be registered with IANA in the "rtcp-fb" Attribute Values registry located at the time of publication at: http://www.iana.org/assignments/sdp-parameters

Value name: ccm
Long Name: Codec Control Commands and Indications
Reference: RFC XXXX

A new registry "Codec Control Messages" needs to be created to hold "ccm" parameters located at time of publication at: http://www.iana.org/assignments/sdp-parameters

New registration in this registry follows the "Specification required" policy as defined by [RFC2434]. In addition they are required to indicate which, if any additional RTCP feedback types, such as "nack", "ack".

The initial content of the registry is the following values:

Value name: fir
Long name: Full Intra Request Command
Usable with: ccm
Reference: RFC XXXX

Value name: tmmbr
Long name: Temporary Maximum Media Stream Bit Rate
Usable with: ccm
Reference: RFC XXXX

Value name: tstr
Long name: temporal Spatial Trade Off
Usable with: ccm
Reference: RFC XXXX

Value name: vbcm
Long name: H.271 video back channel messages
Usable with: ccm
Reference: RFC XXXX

The following values need to be registered as FMT values in the "FMT Values for RTPFB Payload Types" registry located at the time of publication at: http://www.iana.org/assignments/rtp-parameters

RTPFB range

Name	Long Name	Value	Reference
	Reserved	2	[RFCxxxx]
TMMBR	Temporary Maximum Media Stream Bit Rate Request	3	[RFCxxxx]
TMMBN	Temporary Maximum Media Stream Bit Rate Notification	4	[RFCxxxx]

The following values need to be registered as FMT values in the "FMT Values for PSFB Payload Types" registry located at the time of publication at: http://www.iana.org/assignments/rtp-parameters

PSFB range

Name	Long Name	Value	Reference
FIR	Full Intra Request Command	4	[RFCxxxx]
TSTR	Temporal-Spatial Trade-off Request	5	[RFCxxxx]
TSTN	Temporal-Spatial Trade-off Notification	6	[RFCxxxx]
VBCM	Video Back Channel Message	7	[RFCxxxx]

9. Contributors

Tom Taylor has made a very significant contribution, for which the authors are very grateful, to this specification by helping rewrite the specification. Especially the parts regarding the algorithm for determining bounding sets for TMMBR have benefited.

10. Acknowledgements

The authors would like to thank Andrea Basso, Orit Levin, Nermeen Ismail for their work on the requirement and discussion draft [Basso].

Drafts of this memo were reviewed and extensively commented by Roni Even, Colin Perkins, Randell Jesup, Keith Lantz, Harikishan Desineni, Guido Franceschini and others. The authors appreciate these reviews.

Funding for the RFC Editor function is currently provided by the Internet Society.

11. References

11.1. Normative references

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

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

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

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

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

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

11.2. Informative references

[Basso] A. Basso, et. al., "Requirements for transport of video control commands", draft-basso-avt-videoconreq-02.txt, expired Internet Draft, October 2004.

[AVC] Joint Video Team of ITU-T and ISO/IEC JTC 1, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, March 2003.

[H245] ITU-T Rec. HG.245, "Control protocol for multimedia communication", MAY 2006

[NEWPRED] S. Fukunaga, T. Nakai, and H. Inoue, "Error Resilient Video Coding by Dynamic Replacing of Reference Pictures," in Proc. Globcom'96, vol. 3, pp. 1503 - 1508, 1996.

[SRTP] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC2032] Turletti, T. and C. Huitema, "RTP Payload Format for H.261 Video Streams", RFC 2032, October 1996.

[SAVPF] J. Ott, E. Carrara, "Extended Secure RTP Profile for RTCP-based Feedback (RTP/SAVPF)," draft-ietf-avt-profile-savpf-11.txt, February, 2007.

[RFC3525] Groves, C., Pantaleo, M., Anderson, T., and T. Taylor, "Gateway Control Protocol Version 1", RFC 3525, June 2003.

[RFC3448] M. Handley, S. Floyd, J. Padhye, J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, Jan 2003

[VBCM] ITU-T Rec. H.271, "Video Back Channel Messages", June 2006

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

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[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.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, September 1997.

[Topologies] M. Westerlund, and S. Wenger, "RTP Topologies", draft-ietf-avt-topologies-06, work in progress, Aug 2007.

[XML-MC] O. Levin, R. Even, P. Hagendorf, "XML Schema for Media Control," draft-levin-mmusic-xml-media-control-11, work in progress, July 2007.

12. Authors' Addresses

Stephan Wenger
Nokia Corporation
975, Page Mill Road,
Palo Alto,CA 94304
USA

Phone: +1-650-862-7368
EMail: stewe@stewe.org

Umesh Chandra
Nokia Research Center
975, Page Mill Road,
Palo Alto,CA 94304
USA

Phone: +1-650-796-7502
Email: Umesh.1.Chandra@nokia.com

Magnus Westerlund
Ericsson Research
Ericsson AB
SE-164 80 Stockholm, SWEDEN

Phone: +46 8 7190000
EMail: magnus.westerlund@ericsson.com

Bo Burman
Ericsson Research
Ericsson AB
SE-164 80 Stockholm, SWEDEN

Phone: +46 8 7190000
EMail: bo.burman@ericsson.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

RFC Editor Considerations

The RFC editor is requested to replace all occurrences of XXXX with the RFC number this document receives.