

AVT Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 21, 2008

J. Ott  
Helsinki University of Technology  
C. Perkins  
University of Glasgow  
February 18, 2008

Guidelines for Extending the RTP Control Protocol (RTCP)  
draft-ott-avt-rtcp-guidelines-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 21, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

The RTP Control Protocol (RTCP) is used along with the Real-time Transport Protocol (RTP) to provide a control channel between media senders and receivers. This allows constructing a feedback loop to enable application adaptivity and monitoring, among other uses. The basic reporting mechanisms offered by RTCP are generic, yet quite powerful and suffice to cover a range of uses. This document provides guidelines on extending RTCP if those basic mechanisms prove

insufficient.

Table of Contents

- 1. Introduction . . . . . 3
- 2. Terminology . . . . . 4
- 3. RTP and RTCP Operation Overview . . . . . 4
  - 3.1. RTCP Capabilities . . . . . 5
  - 3.2. RTCP Limitations . . . . . 7
- 4. Issues with RTCP Extensions . . . . . 7
- 5. Guidelines . . . . . 9
- 6. Security Considerations . . . . . 11
- 7. IANA Considerations . . . . . 11
- 8. Acknowledgements . . . . . 11
- 9. References . . . . . 11
  - 9.1. Normative References . . . . . 11
  - 9.2. Informative References . . . . . 12
- Authors' Addresses . . . . . 12
- Intellectual Property and Copyright Statements . . . . . 14

## 1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is used to carry time-dependent (often continuous) media such as audio or video across a packet network in an RTP session. RTP usually runs on top of an unreliable transport such as UDP, DTLS, or DCCP, so that RTP packets are susceptible to loss, re-ordering, or duplication. Associated with RTP is the RTP Control Protocol (RTCP) which provides a control channel for each session: media senders provide information about their current sending activities ("feed forward") and media receivers report on their reception statistics ("feedback") in terms of received packets, losses, and jitter. Senders and receivers provide self-descriptions allowing to disambiguate all entities in an RTP session and correlate SSRC identifiers with specific application instances. RTCP is carried over the same transport as RTP and is hence inherently best-effort and hence the RTCP reports are designed for such an unreliable environment, e.g., by making them "for information only".

The RTCP control channel provides coarse-grained information about the session in two respects: 1) the RTCP SR and RR packets contain only cumulative information or means over a certain period of time and 2) the time period is in the order of seconds and thus neither has a high resolution nor does the feedback come back instantaneously. Both these restrictions have their origin in RTP being scalable and generic. Even these basic mechanisms (which are still not implemented everywhere despite their simplicity and very precise specification, including sample code) offer substantial information for designing adaptive applications and for monitoring purposes, among others.

Recently, numerous extensions have been proposed in different contexts to RTCP which significantly increase the complexity of the protocol and the reported values, mutate it toward a command channel, and/or attempt turning it into a reliable messaging protocol. While the reasons for such extensions may be legitimate, many of the resulting designs appear ill-advised in the light of the RTP architecture. Moreover, extensions are often badly motivated and thus appear unnecessary given what can be achieved with the RTCP mechanisms in place today.

This document is intended to provide some guidelines for designing RTCP extensions. It is particularly intended to avoid an extension creep for corner cases which can only harm interoperability and future evolution of the protocol at large. We first outline the basic operation of RTCP and constructing feedback loops using the basic RTCP mechanisms. Subsequently, we outline categories of extensions proposed (and partly already accepted) for RTCP and

discuss issues and alternative ways of thinking by example. Finally, we provide some guidelines and highlight a number of questions to ask (and answer!) before writing up an RTCP extension.

## 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 BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

The terminology defined in RTP [RFC3550], the RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551], and the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] apply.

## 3. RTP and RTCP Operation Overview

One of the twelve networking truths states: "In protocol design, perfection has been reached not when there is nothing left to add, but when there is nothing left to take away" [RFC1925]. Despite (or because of) this being an April, 1st, RFC, this specific truth is very valid and it applies to RTCP as well.

In this section, we will briefly review what is available from the basic RTP/RTCP specifications. As specifications, we include those which are generic, i.e., do not have dependencies on particular media types. This includes the RTP base specification [RFC3550] and profile [RFC3551], the RTCP bandwidth modifiers for session descriptions [RFC3556], the timely feedback extensions (RFC 4585), and the extensions to run RTCP over SSM networks. RTCP XR [RFC3611] provides extended reporting mechanisms which are partly generic in nature, partly specific to a certain media stream.

We do not cover other RTP-related documents: SRTP [RFC3711] is orthogonal as is the mapping of RTP/RTCP to other transports (such as RFC 4571). The description of RTP topologies is useful knowledge (RFC 5117) but functionally not relevant here. Various RTP error control mechanisms (such as RFC 2198, RFC 2733, RFC 4588, and RFC 5109) are useful mechanisms for RTP packets, some of which may be worthwhile for RTCP if appropriate.

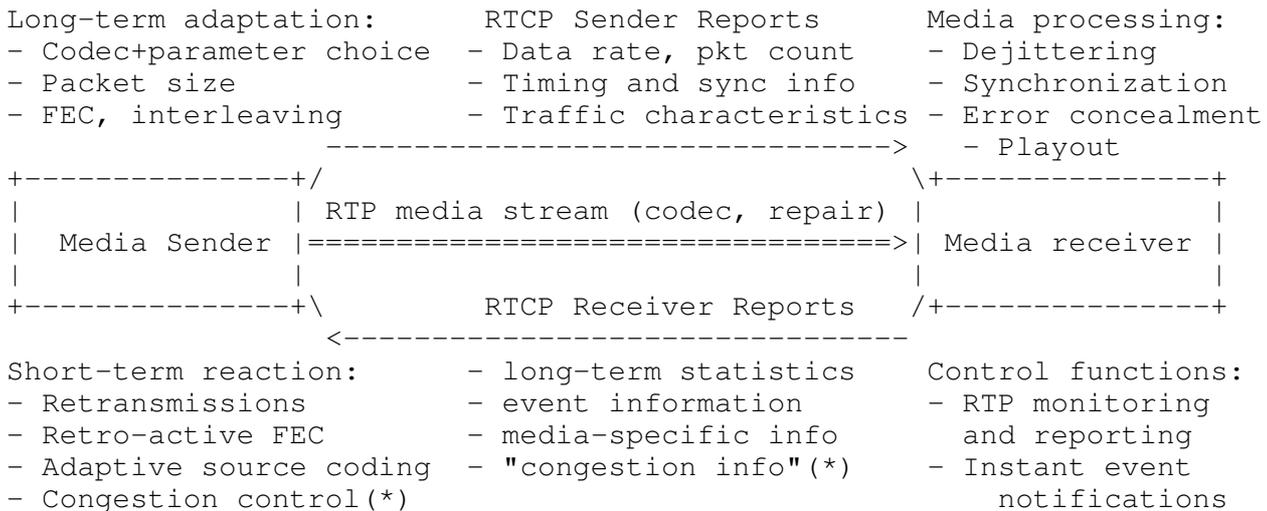
### 3.1. RTCP Capabilities

The RTP/RTCP specifications quoted above provide feedback mechanisms with the following properties, which can be considered as "building blocks" for adaptive real-time applications for IP networks.

- o Sender Reports (SR) indicate to the receivers the total number of packets and octets have been sent (since the beginning of the session or the last change of the sender's SSRC). These values allow deducing the mean data rate and mean packet size for both the entire session and, if continuously monitored, for every transmission interval.
- o Receiver Reports (RR) and SRs indicate reception statistics from each receiver for every sender. These statistics include:
  - \* The packet loss rate since the last SR or RR was sent.
  - \* The total number of packets lost since the beginning of the session which may again be broken down to each reporting period.
  - \* The highest sequence number received so far -- which allows a sender to roughly estimate how much data is in flight when used together with the SR and RR timestamps (and also allows observing whether the path still works and at which rate packets are delivered to the receiver).
  - \* The moving average of the inter-arrival jitter of media packets.
- o These values are sampled in "regular" interval 0.5 ... 1.5 times the deterministic calculated interval T. For sessions with a constant number of RTP entities, T will remain stable and timer reconsideration will not have any effect. T is determined by the media bit rate, the mean RTCP packet size, whether the sampling node is a sender or a receiver, and its respective bitrate budget. T has a lower limit of 5s per [RFC3550] (although this may be scaled to 360 seconds divided by the session bandwidth in kilobits/second in some circumstances, giving an interval smaller than 5 seconds for bandwidths greater than 72 kb/s).
- o The lower limit can be eliminated and more frequent feedback can be provided when using the early feedback profile for RTCP (RFC 4585). In this case, the RTCP frequency is only limited by the available bitrate (usually 5% of the media stream bit rate is allocated for RTCP). If this is insufficient, the RTCP bitrate may be increased in the session description to enable more frequent feedback (RFC 3556). (Work in progress suggests to reduce the mean RTCP packet size [I-D.ietf-avt-rtcp-non-compound] which may help increasing the feedback frequency further.)
- o RFC 4585 even allows -- statistically -- to provide close-to-instant feedback from a receiver to a sender about any observed event in the media stream.

- o Sender Reports also contain timestamps which allow (in conjunction with Receiver Reports) the sender to calculate the current RTT. This value can be monitored over time and thus may be used to infer trends at coarse granularity. RFC 3611 provides a similar RTT mechanism for the receivers.
- o RTCP is suitable for unicast and multicast communications and all basic functions are designed with group communications in mind. While traditional (any-source) multicast (ASM) is clearly not available in the Internet at large, source-specific multicast (SSM) and overlay multicast are -- and both are commercially relevant. RTCP extensions have been defined to operate over SSM, and complex topologies may be created by interconnecting RTP mixers and translators.

These mechanisms can be used to implement a quite flexible feedback loop and enable short-term reaction to observed events as well as long term adaptation to changes in the networking environment. Adaptation mechanisms available on the sender side include (but are not limited to) choosing different codecs, different parameters for codecs (spatial or temporal resolution for video, audible quality for audio and voice), and different packet sizes to adjust the bit rate. Furthermore, various forward error correction mechanisms and, if RTTs are short and the application permits extra delays, even reactive error control such as retransmissions. Long-term feedback can be provided in regular RTCP reports at configurable intervals, whereas (close-to-)instant feedback is available by means of the early feedback profile. Figure 1 below outlines this idea graphically:



(\*) RTCP feedback has been seen as insufficient for congestion control purposes due to the infrequent nature of reporting (which should be in the order of once per RTT).

Figure 1: Outline of an RTCP Feedback Loop

It is important to note that not all information needs to be signalled explicitly -- ever or upon every RTCP packet -- but can be derived locally from other pieces of information and from the evolution of the information over time.

### 3.2. RTCP Limitations

The design of RTP limits what can meaningfully be done (and hence should be done) with RTCP. In particular, the design favours scalability and loose coupling over tightly controlled feedback loops. Some of these limitations are listed below (they need to be taken into account when designing extensions):

- o One response per media packet. RTCP is designed to provide occasional feedback which is unlike, e.g., TCP ACKs which can be sent in response to every (other) packet.
- o RTCP is not capable of providing truly instant feedback.
- o RTCP is inherently unreliable.

### 4. Issues with RTCP Extensions

Issues that came up in the past with extensions to RTP and RTCP include (but are probably not limited to) the following:

- o Defined only or primarily for unicast. RTCP has been defined for multicast. Extensions may become useful in the future well outside their originally intended area of application and should consider this. Stating that something works for unicast only is not an option in many cases, particularly as various flavours of multicast have become relevant again.
- o Assuming reliable (instant) state synchronization. RTCP reports are sent irregularly and may be lost. Hence, there may be a significant time lag between intending to send a state update to the RTP peer(s) and the packet being received, in some cases, the packet may not be received at all.
- o Reliable RTCP. Where needed, reliability is implemented on top of RTCP using acknowledgements. While acknowledgements and retransmission suggest to overcome the above reliability issue (and, in fact, they probably can), this is likely to come at the cost of significant additional delay which may defeat the purpose of providing the feedback in the first place. Moreover, for scalability reasons, if multicasting is used, these ACKs need to be targeted to a subgroup or an individual entity to avoid implosion. The result are typically specific mechanisms with limited re-usability.
- o Commands are issued rather than hints given. RTCP is about reporting observations -- in a best-effort manner -- between RTP entities. Causing actions on the remote side requires some form of reliability (see above) and adherence cannot be verified.
- o RTCP reporting is expanded to become a network management tool. RTCP is sensitive to the size of RTCP reports as the latter determines the mean reporting interval given a certain bit rate share for RTCP. The amount of information going into RTCP reports should primarily target the peer (and thus include information that can be meaningfully reacted upon). Gathering and reporting statistics beyond this is not an RTCP task and should be addressed by out-of-band protocols.
- o Serious complexity is created. Related to the previous item, RTCP reports that convey all kinds of data first need to gather and calculate/infer this information to begin with (which requires very precise specifications). Given that it already seems to be difficult to even implement baseline RTCP, any added complexity can only discourage implementers, may lead to buggy implementations (in which case the reports do not serve the purpose they were intended to), and hinder interoperability.
- o Architectural issues. Extensions are written without considering the architectural concepts of RTP. For example, point-to-point communication is assumed, yet third party monitors are expected to listen in. Besides being a bad idea to rely on eavesdropping entities on the path, this is obviously not possible if SRTP is being used with encrypted SRTCP packets.

This list is surely not exhaustive. Also, the authors do not claim that the suggested extensions (even if using acknowledgements) would not serve a legitimate purpose. We rather want to draw attention to the fact that the same results may be achievable in an architecturally cleaner and conceptually RTP/RTCP-compliant way. The following section contains a first attempt to provide some guidelines on what to consider when thinking about extensions to RTP and RTCP.

## 5. Guidelines

Designing RTCP extensions requires consideration of a number aspects as well as in-depth understanding of the operation of RTP and its basic mechanisms. While it is expected that there are many aspects not yet covered by RTCP reporting and operation, quite a bit of functionality is readily available for use. Quite a few further mechanisms should probably never become part of the RTP family of specifications. In the following, we provide a bit of guidance (in its first iteration) what to consider when (and before!) developing an extension to RTCP.

Note: In this first revision of this draft, this section is still very rough.

We begin with a short check list concerning the applicability of RTCP in the first place:

- o Check what you can do with the existing mechanisms and exploit the information readily available. Is the need for an extension only perceived (e.g., because of lazy implementers, artificial constraints in endpoints) or is the function or information really not available. It is worthwhile remembering that any piece of redundant information supplied by a protocol runs the risk of being inconsistent at some point and different implementation may choose to handle such situations differently (e.g., give precedence to different values). Similarly, every function should only be doable in exactly one (well-specified) way.
- o Is the extension really applicable to RTP entities running anywhere in the Internet or is this a link-specific extension? In the latter case, link-specific extensions (e.g., using header compression) may be preferable. RTCP should also not be used to carry information specific to a particular (access) link.

From a conceptual viewpoint, every RTCP extension should ask and answer(!) at least the following questions:

- o How will this new building block complement and work with the others? Are all interactions specified?
- o Will this work with all different profiles? (Security?) Are any feature interactions expected?
- o Should this extension be kept in-line with baseline RTP and the existing RTP profiles? Or does the extension deviate so much from the basic operation (and the answer to the previous question is "no") so that a new profile should be defined which is allowed to be incompatible with others? If so, how do nodes using different profiles fail safely?
- o How does this extension interoperate with other nodes if the extension is not understood by the peer(s)?
- o How will the extension deal with different networking conditions (e.g., degrade with increase in losses or latency, possibly across orders of magnitude)?
- o How will this extension work with multicast? Will this degrade gracefully with increasing group sizes? What will be the impact on the RTCP report frequency and bitrate allocation?

For the specific design, the following considerations should be taken into account (they form a mixture of common protocol design guidelines and specifics for RTCP):

- o First of all, if there is (and for RTCP this applies quite often) an old-style mechanisms from a different networking environment, don't try to recreate this in RTCP. The networking environment is heterogeneous and will often be drastically different. Instead, ask what the actual semantics and the result perceived by the application or the user are. Then, design a mechanism that achieves this result compatible with RTP/RTCP. (And do not forget that every mechanism will break if no packets get through.)
- o Target re-usability for the specification, i.e., think broader than a specific use case. (Well, there is a balance to be struck: in order to get anything done, there needs to be sufficient focus.) Point solutions need a really good motivation to be dealt with in the IETF in the first place. This essentially suggests to develop buildings blocks whenever possible.
- o For everything (value, procedure, timer, etc.) being defined, make sure that it is defined properly, so that independent interoperable implementation can be done.
- o When defining field, mechanisms, etc., remember that all field values need to be both generated and reacted upon, that mechanisms need to be implemented, etc., and that all of this increases the complexity of an implementation. Things which are too complex won't get implemented in the first place. Extensions defining new metrics and parameters should reference existing standards where possible, rather than invent something new and/or proprietary.

- o Remember that not every bit or every action needs to be represented or signalled explicitly. It may be possible to infer the necessary pieces of information from other values or their evolution (a very prominent example is TCP congestion control). As a result, it may be possible to decouple bits on the wire from local actions and reduce the overhead.
- o Particularly with media streams, reliability can often be "soft". Rather than implementing explicit acknowledgements, receipt of a hint may also be observed from the altered behaviour (e.g., sending the requested intra-frame or changing the reference frame for video, changing the codec, etc.). The semantics of messages should be idempotent so that the respective message may be sent repeatedly.

TBD: discuss and cite the ALF paper (Clark and Tennenhouse, SIGCOMM 1990)

## 6. Security Considerations

TBD.

Congestion control and denial of service issues are likely key points to raise here.

## 7. IANA Considerations

There are no specific IANA action necessary for this document.

## 8. Acknowledgements

This draft has been motivated by many discussions in the AVT WG. The authors would like to acknowledge the active members in the group for providing the inspiration.

## 9. References

### 9.1. Normative References

- [RFC1925] Callon, R., "The Twelve Networking Truths", RFC 1925, April 1996.
- [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.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.
- [RFC3556] Casner, S., "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", RFC 3556, July 2003.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

## 9.2. Informative References

- [I-D.ietf-avt-rtcp-non-compound]  
Johansson, I. and M. Westerlund, "Support for non-compound RTCP, opportunities and consequences",  
draft-ietf-avt-rtcp-non-compound-02 (work in progress),  
February 2008.

## Authors' Addresses

Joerg Ott  
Helsinki University of Technology  
Otakaari 5 A  
Espoo, FIN 02150  
Finland

Email: jo@netlab.hut.fi

Colin Perkins  
University of Glasgow  
Department of Computing Science  
Sir Alwyn Williams Building  
Lilybank Gardens  
Glasgow G12 8QQ  
United Kingdom

Email: [csp@csperkins.org](mailto:csp@csperkins.org)

## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgment

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

