

AVTCore
Internet-Draft
Obsoletes: RFC5285 (if approved)
Intended status: Standards Track
Expires: April 10, 2016

R. Even, Ed.
Huawei Technologies
D. Singer
Apple, Inc.
H. Desineni
Qualcomm
October 8, 2015

A General Mechanism for RTP Header Extensions
draft-even-avtcore-rfc5285-bis-01.txt

Abstract

This document provides a general mechanism to use the header extension feature of RTP (the Real-Time Transport Protocol). It provides the option to use a small number of small extensions in each RTP packet, where the universe of possible extensions is large and registration is de-centralized. The actual extensions in use in a session are signaled in the setup information for that session.

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 10, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	2
2. Requirements Notation	3
3. Design Goals	3
4. Packet Design	3
4.1. General	3
4.2. One-Byte Header	5
4.3. Two-Byte Header	7
5. SDP Signaling Design	8
6. SDP Signaling for support of mixed one byte and two bytes header extensions.	10
7. Offer/Answer	11
8. BNF Syntax	13
9. Security Considerations	14
10. IANA Considerations	14
10.1. Identifier Space for IANA to Manage	15
10.2. Registration of the SDP extmap Attribute	16
10.3. Registration of the SDP Attribute	17
11. Acknowledgments	17
12. References	17
12.1. Normative References	17
12.2. Informative References	18
Authors' Addresses	18

1. Introduction

The RTP specification [RFC3550] provides a capability to extend the RTP header. It defines the header extension format and rules for its use in Section 5.3.1. The existing header extension method permits at most one extension per RTP packet, identified by a 16-bit identifier and a 16-bit length field specifying the length of the header extension in 32-bit words.

This mechanism has two conspicuous drawbacks. First, it permits only one header extension in a single RTP packet. Second, the specification gives no guidance as to how the 16-bit header extension identifiers are allocated to avoid collisions.

This specification removes the first drawback by defining a backward-compatible and extensible means to carry multiple header extension elements in a single RTP packet. It removes the second drawback by defining that these extension elements are named by URIs, defining an IANA registry for extension elements defined in IETF specifications,

and a Session Description Protocol (SDP) method for mapping between the naming URIs and the identifier values carried in the RTP packets.

This header extension applies to RTP/AVP (the Audio/Visual Profile) and its extensions.

This document removes a limitation from RFC5285 that did not allow sending both one byte and two bytes header extensions in the same RTP stream

2. Requirements Notation

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

3. Design Goals

The goal of this design is to provide a simple mechanism whereby multiple identified extensions can be used in RTP packets, without the need for formal registration of those extensions but nonetheless avoiding collision.

This mechanism provides an alternative to the practice of burying associated metadata into the media format bit stream. This has often been done in media data sent over fixed-bandwidth channels. Once this is done, a decoder for the specific media format is required to extract the metadata. Also, depending on the media format, the metadata may need to be added at the time of encoding the media so that the bit-rate required for the metadata is taken into account. But the metadata may not be known at that time. Inserting metadata at a later time can require a decode and re-encode to meet bit-rate requirements.

In some cases, a more appropriate, higher-level mechanism may be available, and if so, it should be used. For cases where a higher-level mechanism is not available, it is better to provide a mechanism at the RTP level than have the metadata be tied to a specific form of media data.

4. Packet Design

4.1. General

The following design is fit into the "header extension" of the RTP extension, as described above.

The presence and format of this header extension and its contents are negotiated or defined out-of-band, such as through signaling (see below for SDP signaling). The value defined for an RTP extension (defined below for the one-byte and two-byte header forms) is only an architectural constant (e.g., for use by network analyzers); it is the negotiation/definition (e.g., in SDP) that is the definitive indication that this header extension is present.

This specification inherits the requirement from the RTP specification that the header extension "is designed so that the header extension may be ignored". To be specific, header extensions using this specification MUST only be used for data that can safely be ignored by the recipient without affecting interoperability, and MUST NOT be used when the presence of the extension has changed the form or nature of the rest of the packet in a way that is not compatible with the way the stream is signaled (e.g., as defined by the payload type). Valid examples might include metadata that is additional to the usual RTP information.

The RTP header extension is formed as a sequence of extension elements, with possible padding. Each extension element has a local identifier and a length. The local identifiers may be mapped to a larger namespace in the negotiation (e.g., session signaling).

As is good network practice, data should only be transmitted when needed. The RTP header extension should only be present in a packet if that packet also contains one or more extension elements, as defined here. An extension element should only be present in a packet when needed; the signaling setup of extension elements indicates only that those elements may be present in some packets, not that they are in fact present in all (or indeed, any) packets.

Each extension element in a packet has a local identifier (ID) and a length. The local identifiers present in the stream MUST have been negotiated or defined out-of-band. There are no static allocations of local identifiers. Each distinct extension MUST have a unique ID. The value 0 is reserved for padding and MUST NOT be used as a local identifier.

There are two variants of the extension: one-byte and two-byte headers. Since it is expected that (a) the number of extensions in any given RTP session is small and (b) the extensions themselves are small, the one-byte header form is preferred and MUST be supported by all receivers. A stream MUST contain only one-byte or two-byte headers unless it is known that all recipients support mixing, either by offer/answer negotiation (see section 6) or by out-of-band knowledge. One-byte and two-byte headers MUST NOT be mixed in a single RTP

packet. Transmitters SHOULD NOT use the two-byte form when all extensions are small enough for the one-byte header form.

A sequence of extension elements, possibly with padding, forms the header extension defined in the RTP specification. There are as many extension elements as fit into the length as indicated in the RTP header extension length. Since this length is signaled in full 32-bit words, padding bytes are used to pad to a 32-bit boundary. The entire extension is parsed byte-by-byte to find each extension element (no alignment is required), and parsing stops at the earlier of the end of the entire header extension, or in one-byte headers only case, on encountering an identifier with the reserved value of 15.

In both forms, padding bytes have the value of 0 (zero). They may be placed between extension elements, if desired for alignment, or after the last extension element, if needed for padding. A padding byte does not supply the ID of an element, nor the length field. When a padding byte is found, it is ignored and the parser moves on to interpreting the next byte.

Note carefully that the one-byte header form allows for data lengths between 1 and 16 bytes, by adding 1 to the signaled length value (thus, 0 in the length field indicates 1 byte of data follows). This allows for the important case of 16-byte payloads. This addition is not performed for the two-byte headers, where the length field signals data lengths between 0 and 255 bytes.

Use of RTP header extensions will reduce the efficiency of RTP header compression, since the header extension will be sent uncompressed unless the RTP header compression module is updated to recognize the extension header. If header extensions are present in some packets, but not in others, this can also reduce compression efficiency by requiring an update to the fixed header to be conveyed when header extensions start or stop being sent. The interactions of the RTP header extension and header compression is explored further in [RFC2508] and [RFC3095].

4.2. One-Byte Header

In the one-byte header form of extensions, the 16-bit value required by the RTP specification for a header extension, labeled in the RTP specification as "defined by profile", takes the fixed bit pattern 0xBEDE (the first version of this specification was written on the feast day of the Venerable Bede).

Each extension element starts with a byte containing an ID and a length:

```

0
0 1 2 3 4 5 6 7
+-----+
|  ID   | len  |
+-----+
```

The 4-bit ID is the local identifier of this element in the range 1-14 inclusive. In the signaling section, this is referred to as the valid range.

The local identifier value 15 is reserved for future extension and MUST NOT be used as an identifier. If the ID value 15 is encountered, its length field should be ignored, processing of the entire extension should terminate at that point, and only the extension elements present prior to the element with ID 15 considered.

The 4-bit length is the number minus one of data bytes of this header extension element following the one-byte header. Therefore, the value zero in this field indicates that one byte of data follows, and a value of 15 (the maximum) indicates element data of 16 bytes. (This permits carriage of 16-byte values, which is a common length of labels and identifiers, while losing the possibility of zero-length values -- which would often be padded anyway.)

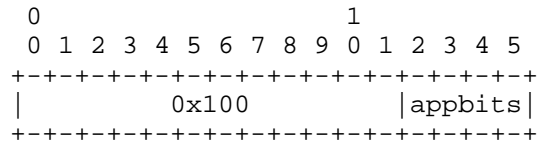
An example header extension, with three extension elements, some padding, and including the required RTP fields, follows:

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|          0xBE          |          0xDE          |          length=3          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  ID  | L=0 |          data          |  ID  | L=1 |          data...          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          ...data          |          0 (pad)          |          0 (pad)          | ID  | L=3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          data          |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

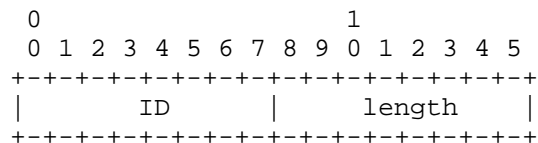
4.3. Two-Byte Header

In the two-byte header form, the 16-bit value required by the RTP specification for a header extension, labeled in the RTP specification as "defined by profile", is defined as shown below.



The appbits field is 4 bits that are application-dependent and may be defined to be any value or meaning, and are outside the scope of this specification. For the purposes of signaling, this field is treated as a special extension value assigned to the local identifier 256. If no extension has been specified through configuration or signaling for this local identifier value 256, the appbits field SHOULD be set to all 0s by the sender and MUST be ignored by the receiver.

Each extension element starts with a byte containing an ID and a byte containing a length:



The 8-bit ID is the local identifier of this element in the range 1-255 inclusive. In the signaling section, the range 1-256 is referred to as the valid range, with the values 1-255 referring to extension elements, and the value 256 referring to the 4-bit field 'appbits' (above).

The 8-bit length field is the length of extension data in bytes not including the ID and length fields. The value zero indicates there is no data following.

An example header extension, with three extension elements, some padding, and including the required RTP fields, follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|           0x10          |           0x00          |           length=3          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           ID           |           L=0           |           ID           |           L=1           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           data         |           0 (pad)        |           ID           |           L=4           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     data                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

5. SDP Signaling Design

The indication of the presence of this extension, and the mapping of local identifiers used in the header extension to a larger namespace, **MUST** be performed out-of-band, for example, as part of a SIP offer/answer exchange using SDP. This section defines such signaling in SDP.

A usable mapping **MUST** use IDs in the valid range, and each ID in this range **MUST** be used only once for each media (or only once if the mappings are session level). Mappings that do not conform to these rules **MAY** be presented, for instance, during offer/answer negotiation as described in the next section, but remapping to conformant values is necessary before they can be applied.

Each extension is named by a URI. That URI **MUST** be absolute, and precisely identifies the format and meaning of the extension. URIs that contain a domain name **SHOULD** also contain a month-date in the form `mm/yyyy`. The definition of the element and assignment of the URI **MUST** have been authorized by the owner of the domain name on or very close to that date. (This avoids problems when domain names change ownership.) If the resource or document defines several extensions, then the URI **MUST** identify the actual extension in use, e.g., using a fragment or query identifier (characters after a '#' or '?' in the URI).

Rationale: the use of URIs provides for a large, unallocated space, and gives documentation on the extension. The URIs are not required to be de-referencable, in order to permit confidential or experimental use, and to cover the case when extensions continue to be used after the organization that defined them ceases to exist.

An extension URI with the same attributes MUST NOT appear more than once applying to the same stream, i.e., at session level or in the declarations for a single stream at media level. (The same extension may, of course, be used for several streams, and may appear differently parameterized for the same stream.)

For extensions defined in RFCs, the URI used SHOULD be a URN starting "urn:ietf:params:rtp-hdext:" and followed by a registered, descriptive name.

The registration requirements are detailed in the IANA Considerations section, below.

An example (this is only an example), where 'avt-example-metadata' is the hypothetical name of a header extension, might be:

```
urn:ietf:params:rtp-hdext:avt-example-metadata
```

An example name not from the IETF (this is only an example) might be:

```
http://example.com/082005/ext.htm#example-metadata
```

The mapping may be provided per media stream (in the media-level section(s) of SDP, i.e., after an "m=" line) or globally for all streams (i.e., before the first "m=" line, at session level). The definitions MUST be either all session level or all media level; it is not permitted to mix the two styles. In addition, as noted above, the IDs used MUST be unique for each stream type for a given media, or for the session for session-level declarations.

Each local identifier potentially used in the stream is mapped to a string using an attribute of the form:

```
a=extmap:<value>["/"<direction>] <URI> <extensionattributes>
```

where <URI> is a URI, as above, <value> is the local identifier (ID) of this extension and is an integer in the valid range inclusive (0 is reserved for padding in both forms, and 15 is reserved in the one-byte header form, as noted above), and <direction> is one of "sendonly", "recvonly", "sendrecv", or "inactive" (without the quotes).

The formal BNF syntax is presented in a later section of this specification.

Example:

```
a=extmap:1 http://example.com/082005/ext.htm#ttime
```

```
a=extmap:2/sendrecv http://example.com/082005/ext.htm#xmeta short
```

When SDP signaling is used for the RTP session, it is the presence of the 'extmap' attribute(s) that is diagnostic that this style of header extensions is used, not the magic number indicated above.

6. SDP Signaling for support of mixed one byte and two bytes header extensions.

In order to allow for backward interoperability with systems that do not support mixing of one byte and two bytes header extensions this document defines the "a=extmap-allow-mixed" Session Description Protocol (SDP) [RFC4566] attribute to indicate if the participant is capable of supporting this new mode. The attribute takes no value. This attribute can be used at the session or media levels. A participant that proposes the use of this mode SHALL itself support the reception of mixed one byte and two bytes header extensions.

The negotiation for mixed one byte and two bytes extension MUST be negotiated in offer/answer [RFC3264]. In the absence of negotiation using offer/answer, mixed headers MUST NOT occur unless the transmitter has some (out of band) knowledge that all potential recipients support this mode.

The formal definition of this attribute is:

Name: extmap-allow-mixed

Value:

Usage Level: session, media

Charset Dependent: no

Example:

```
a=extmap-allow-mixed
```

When doing SDP Offer/Answer [RFC3264] an offering client that wishes to use both one and two bytes extensions MUST include the attribute "a= extmap-allow-mixed " in the SDP offer. If "a= extmap-allow-mixed " is present in the offer SDP, the answerer that supports this mode and wishes to use it SHALL include the "a=extmap-allow-mixed " attribute in the answer. In cases the answer has been excluded, neither clients SHALL use mixed one bytes and two bytes extensions in the same RTP stream.

7. Offer/Answer

The simple signaling described above may be enhanced in an offer/answer context, to permit:

- o asymmetric behavior (extensions sent in only one direction),
- o the offer of mutually exclusive alternatives, or
- o the offer of more extensions than can be sent in a single session.

A direction attribute MAY be included in an extmap; without it, the direction implicitly inherits, of course, from the stream direction, or is "sendrecv" for session-level attributes or extensions of "inactive" streams. The direction MUST be one of "sendonly", "recvonly", "sendrecv", or "inactive". A "sendonly" direction indicates an ability to send; a "recvonly" direction indicates a desire to receive; a "sendrecv" direction indicates both. An "inactive" direction indicates neither, but later re-negotiation may make an extension active.

Extensions, with their directions, may be signaled for an "inactive" stream. It is an error to use an extension direction incompatible with the stream direction (e.g., a "sendonly" attribute for a "recvonly" stream).

If an offer or answer contains session-level mappings (and hence no media-level mappings), and different behavior is desired for each stream, then the entire set of extension map declarations may be moved into the media-level section(s) of the SDP. (Note that this specification does not permit mixing global and local declarations, to make identifier management easier.)

If an extension map is offered as "sendrecv", explicitly or implicitly, and asymmetric behavior is desired, the SDP may be modified to modify or add direction qualifiers for that extension.

If an extension is marked as "sendonly" and the answerer desires to receive it, the extension MUST be marked as "recvonly" in the SDP answer. An answerer that has no desire to receive the extension or does not understand the extension SHOULD remove it from the SDP answer.

If an extension is marked as "recvonly" and the answerer desires to send it, the extension MUST be marked as "sendonly" in the SDP answer. An answerer that has no desire to, or is unable to, send the extension SHOULD remove it from the SDP answer.

Local identifiers in the valid range inclusive in an offer or answer must not be used more than once per media section (including the session-level section). A session update MAY change the direction qualifiers of extensions under use. A session update MAY add or remove extension(s). Identifiers values in the valid range MUST NOT be altered (remapped).

Note that, under this rule, the same local identifier cannot be used for two extensions for the same media, even when one is "sendonly" and the other "recvonly", as it would then be impossible to make either of them sendrecv (since re-numbering is not permitted either).

If a party wishes to offer mutually exclusive alternatives, then multiple extensions with the same identifier in the (unusable) range 4096-4351 may be offered; the answerer should select at most one of the offered extensions with the same identifier, and remap it to a free identifier in the valid range, for that extension to be usable.

Similarly, if more extensions are offered than can be fit in the valid range, identifiers in the range 4096-4351 may be offered; the answerer should choose those that are desired, and remap them to a free identifier in the valid range.

It is always allowed to place the offered identifier value "as is" in the SDP answer (for example, due to lack of a free identifier value in the valid range). Extensions with an identifier outside the valid range cannot, of course, be used. If required, the offerer or answerer can update the session to make space for such an extension.

Rationale: the range 4096-4351 for these negotiation identifiers is deliberately restricted to allow expansion of the range of valid identifiers in future.

Either party MAY include extensions in the stream other than those negotiated, or those negotiated as "inactive", for example, for the benefit of intermediate nodes. Only extensions that appeared with an identifier in the valid range in SDP originated by the sender can be sent.

Example (port numbers, RTP profiles, payload IDs and rtpmaps, etc. all omitted for brevity):

The offer:

```
a=extmap:1 URI-toffset
a=extmap:14 URI-obscure
a=extmap:4096 URI-gps-string
a=extmap:4096 URI-gps-binary
a=extmap:4097 URI-frametype
m=video
a=sendrecv
m=audio
a=sendrecv
```

The answerer is interested in receiving GPS in string format only on video, but cannot send GPS at all. It is not interested in transmission offsets on audio, and does not understand the URI-obscure extension. It therefore moves the extensions from session level to media level, and adjusts the declarations:

```
m=video
a=sendrecv
a=extmap:1 URI-toffset
a=extmap:2/recvonly URI-gps-string
a=extmap:3 URI-frametype
m=audio
a=sendrecv
a=extmap:1/sendonly URI-toffset
```

8. BNF Syntax

The syntax definition below uses ABNF according to [RFC5234]. The syntax element 'URI' is defined in [RFC3986] (only absolute URIs are permitted here). The syntax element 'extmap' is an attribute as defined in [RFC4566], i.e., "a=" precedes the extmap definition. Specific extension attributes are defined by the specification that defines a specific extension name; there may be several.

```
extmap = mapentry SP extensionname [SP extensionattributes]
extensionname = URI
direction = "sendonly" / "recvonly" / "sendrecv" / "inactive"
mapentry = "extmap:" 1*5DIGIT [ "/" direction]
extensionattributes = byte-string
URI = <Defined in RFC 3986>
byte-string = <Defined in RFC 4566>
SP = <Defined in RFC 5234>
DIGIT = <Defined in RFC 5234>
```

9. Security Considerations

This defines only a place to transmit information; the security implications of the extensions must be discussed with those extensions.

Care should be taken when defining extensions. Clearly, they should be solely informative, but even when the information is extracted, should not cause security concerns.

Header extensions have the same security coverage as the RTP header itself. When Secure Real-time Transport Protocol (SRTP) [RFC3711] is used to protect RTP sessions, the RTP payload may be both encrypted and integrity protected, while the RTP header is either unprotected or integrity protected. Therefore, it is inappropriate to place information in header extensions that cause security problems if disclosed, unless the entire RTP packet is protected by a lower-layer security protocol providing both confidentiality and integrity capability.

10. IANA Considerations

This document updates the IANA consideration to reference this document and adds a new SDP attribute in section 10.3

Note to IANA : change RFCxxxx to this RFC number and remove the note.

10.1. Identifier Space for IANA to Manage

The mapping from the naming URI form to a reference to a specification is managed by IANA. Insertion into this registry is under the requirements of "Expert Review" as defined in [RFC5226].

The IANA will also maintain a server that contains all of the registered elements in a publicly accessible space.

Here is the formal declaration required by the IETF URN Sub-namespace specification [RFC3553].

- o Registry name: RTP Compact Header Extensions
- o Specification: RFC 5285 and RFCs updating RFC 5285.
- o Information required:
 - A. The desired extension naming URI
 - B. A formal reference to the publicly available specification
 - C. A short phrase describing the function of the extension
 - D. Contact information for the organization or person making the registration

For extensions defined in RFCs, the URI is recommended to be of the form urn:ietf:params:rtp-hdext:, and the formal reference is the RFC number of the RFC documenting the extension.

- o Review process: Expert review is required. The expert review should check the following requirements:
 - 1. that the specification is publicly available;
 - 2. that the extension complies with the requirements of RTP and this specification, for extensions (notably, that the stream is still decodable if the extension is ignored or not recognized);
 - 3. that the extension specification is technically consistent (in itself and with RTP), complete, and comprehensible;
 - 4. that the extension does not duplicate functionality in existing IETF specifications (including RTP itself), or other extensions already registered;

5. that the specification contains a security analysis regarding the content of the header extension;
 6. that the extension is generally applicable, for example point-to-multipoint safe, and the specification correctly describes limitations if they exist; and
 7. that the suggested naming URI form is appropriately chosen and unique.
- o Size and format of entries: a mapping from a naming URI string to a formal reference to a publicly available specification, with a descriptive phrase and contact information.
 - o Initial assignments: none.

10.2. Registration of the SDP extmap Attribute

This section contains the information required by [RFC4566] for an SDP attribute.

- o contact name, email address, and telephone number:

D. Singer
singer@apple.com
+1 408-974-3162
- o attribute name (as it will appear in SDP): extmap
- o long-form attribute name in English: generic header extension map definition
- o type of attribute (session level, media level, or both): both
- o whether the attribute value is subject to the charset attribute: not subject to the charset attribute
- o a one-paragraph explanation of the purpose of the attribute: This attribute defines the mapping from the extension numbers used in packet headers into extension names as documented in specifications and appropriately registered.
- o a specification of appropriate attribute values for this attribute: see RFC 5285.

10.3. Registration of the SDP Attribute

The IANA is requested to register one new SDP attribute:

```
SDP Attribute ("att-field"):  
  Attribute name:      extmap-allow-mixed  
  Long form:          One and Two bytes mixed mode  
  Type of name:       att-field  
  Type of attribute:  Media or session level  
  Subject to charset: No  
  Purpose:            Negotiate the use of One and Two bytes  
                     in the same RTP stream.  
  Reference:          [RFCXXXX]  
  Values:             None
```

11. Acknowledgments

Both Brian Link and John Lazzaro provided helpful comments on an initial draft of this document. Colin Perkins was helpful in reviewing and dealing with the details. The use of URNs for IETF-defined extensions was suggested by Jonathan Lennox, and Pete Cordell was instrumental in improving the padding wording. Dave Oran provided feedback and text in the review. Mike Dolan contributed the two-byte header form. Magnus Westerlund and Tom Taylor were instrumental in managing the registration text.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.
- [RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, June 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

12.2. Informative References

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

Authors' Addresses

Roni Even (editor)
Huawei Technologies
Shabazi 12A
Tel Aviv
Israel

EMail: Roni.even@mail01.huawei.com

David Singer
Apple, Inc.
1 Infinite Loop
Cupertino, CA 95014
USA

Phone: +1 408 996 1010
EMail: singer@apple.com
URI: <http://www.apple.com/quicktime>

Harikishan Desineni
Qualcomm
5775 Morehouse Drive
San Diego, CA 92126
USA

Phone: +1 858 845 8996
EMail: hd@qualcomm.com
URI: <http://www.qualcomm.com>

AVTCORE Working Group
Internet-Draft
Updates: 3550 (if approved)
Intended status: Standards Track
Expires: April 18, 2016

C. Perkins
University of Glasgow
V. Singh
Aalto University
October 16, 2015

Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions
draft-ietf-avtccore-rtp-circuit-breakers-11

Abstract

The Real-time Transport Protocol (RTP) is widely used in telephony, video conferencing, and telepresence applications. Such applications are often run on best-effort UDP/IP networks. If congestion control is not implemented in the applications, then network congestion will deteriorate the user's multimedia experience. This document does not propose a congestion control algorithm; instead, it defines a minimal set of RTP circuit-breakers. Circuit-breakers are conditions under which an RTP sender needs to stop transmitting media data in order to protect the network from excessive congestion. It is expected that, in the absence of severe congestion, all RTP applications running on best-effort IP networks will be able to run without triggering these circuit breakers. Any future RTP congestion control specification will be expected to operate within the constraints defined by these circuit breakers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 18, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Background	3
3. Terminology	6
4. RTP Circuit Breakers for Systems Using the RTP/AVP Profile	7
4.1. RTP/AVP Circuit Breaker #1: RTCP Timeout	9
4.2. RTP/AVP Circuit Breaker #2: Media Timeout	10
4.3. RTP/AVP Circuit Breaker #3: Congestion	11
4.4. RTP/AVP Circuit Breaker #4: Media Usability	15
4.5. Ceasing Transmission	16
5. RTP Circuit Breakers and the RTP/AVPF and RTP/SAVPF Profiles	17
6. Impact of RTCP Extended Reports (XR)	18
7. Impact of Explicit Congestion Notification (ECN)	18
8. Impact of Bundled Media and Layered Coding	18
9. Security Considerations	19
10. IANA Considerations	19
11. Acknowledgements	20
12. References	20
12.1. Normative References	20
12.2. Informative References	20
Authors' Addresses	23

1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is widely used in voice-over-IP, video teleconferencing, and telepresence systems. Many of these systems run over best-effort UDP/IP networks, and can suffer from packet loss and increased latency if network congestion occurs. Designing effective RTP congestion control algorithms, to adapt the transmission of RTP-based media to match the available network capacity, while also maintaining the user experience, is a difficult but important problem. Many such congestion control and

media adaptation algorithms have been proposed, but to date there is no consensus on the correct approach, or even that a single standard algorithm is desirable.

This memo does not attempt to propose a new RTP congestion control algorithm. Instead, we propose a small set of RTP circuit breakers. These are conditions under which there is general agreement that an RTP flow is causing serious congestion, and hence ought to cease transmission. The RTP circuit breakers proposed in this memo are a specific instance of the general class of network transport circuit breakers [I-D.ietf-tsvwg-circuit-breaker], designed to act as a protection mechanism of last resort to avoid persistent congestion. It is expected that future standards-track congestion control algorithms for RTP will operate within the envelope defined by this memo.

2. Background

We consider congestion control for unicast RTP traffic flows. This is the problem of adapting the transmission of an audio/visual data flow, encapsulated within an RTP transport session, from one sender to one receiver, so that it matches the available network bandwidth. Such adaptation needs to be done in a way that limits the disruption to the user experience caused by both packet loss and excessive rate changes. Congestion control for multicast flows is outside the scope of this memo. Multicast traffic needs different solutions, since the available bandwidth estimator for a group of receivers will differ from that for a single receiver, and because multicast congestion control has to consider issues of fairness across groups of receivers that do not apply to unicast flows.

Congestion control for unicast RTP traffic can be implemented in one of two places in the protocol stack. One approach is to run the RTP traffic over a congestion controlled transport protocol, for example over TCP, and to adapt the media encoding to match the dictates of the transport-layer congestion control algorithm. This is safe for the network, but can be suboptimal for the media quality unless the transport protocol is designed to support real-time media flows. We do not consider this class of applications further in this memo, as their network safety is guaranteed by the underlying transport.

Alternatively, RTP flows can be run over a non-congestion controlled transport protocol, for example UDP, performing rate adaptation at the application layer based on RTP Control Protocol (RTCP) feedback. With a well-designed, network-aware, application, this allows highly effective media quality adaptation, but there is potential to disrupt the network's operation if the application does not adapt its sending

rate in a timely and effective manner. We consider this class of applications in this memo.

Congestion control relies on monitoring the delivery of a media flow, and responding to adapt the transmission of that flow when there are signs that the network path is congested. Network congestion can be detected in one of three ways: 1) a receiver can infer the onset of congestion by observing an increase in one-way delay caused by queue build-up within the network; 2) if Explicit Congestion Notification (ECN) [RFC3168] is supported, the network can signal the presence of congestion by marking packets using ECN Congestion Experienced (CE) marks; or 3) in the extreme case, congestion will cause packet loss that can be detected by observing a gap in the received RTP sequence numbers.

Once the onset of congestion is observed, the receiver has to send feedback to the sender to indicate that the transmission rate needs to be reduced. How the sender reduces the transmission rate is highly dependent on the media codec being used, and is outside the scope of this memo.

There are several ways in which a receiver can send feedback to a media sender within the RTP framework:

- o The base RTP specification [RFC3550] defines RTCP Reception Report (RR) packets to convey reception quality feedback information, and Sender Report (SR) packets to convey information about the media transmission. RTCP SR packets contain data that can be used to reconstruct media timing at a receiver, along with a count of the total number of octets and packets sent. RTCP RR packets report on the fraction of packets lost in the last reporting interval, the cumulative number of packets lost, the highest sequence number received, and the inter-arrival jitter. The RTCP RR packets also contain timing information that allows the sender to estimate the network round trip time (RTT) to the receivers. RTCP reports are sent periodically, with the reporting interval being determined by the number of SSRCs used in the session and a configured session bandwidth estimate (the number of SSRCs used is usually two in a unicast session, one for each participant, but can be greater if the participants send multiple media streams). The interval between reports sent from each receiver tends to be on the order of a few seconds on average, although it varies with the session bandwidth, and sub-second reporting intervals are possible in high bandwidth sessions, and it is randomised to avoid synchronisation of reports from multiple receivers. RTCP RR packets allow a receiver to report ongoing network congestion to the sender. However, if a receiver detects the onset of congestion part way through a reporting interval, the base RTP specification contains

no provision for sending the RTCP RR packet early, and the receiver has to wait until the next scheduled reporting interval.

- o The RTCP Extended Reports (XR) [RFC3611] allow reporting of more complex and sophisticated reception quality metrics, but do not change the RTCP timing rules. RTCP extended reports of potential interest for congestion control purposes are the extended packet loss, discard, and burst metrics [RFC3611], [RFC7002], [RFC7097], [RFC7003], [RFC6958]; and the extended delay metrics [RFC6843], [RFC6798]. Other RTCP Extended Reports that could be helpful for congestion control purposes might be developed in future.
- o Rapid feedback about the occurrence of congestion events can be achieved using the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] (or its secure variant, RTP/SAVPF [RFC5124]) in place of the RTP/AVP profile [RFC3551]. This modifies the RTCP timing rules to allow RTCP reports to be sent early, in some cases immediately, provided the RTCP transmission rate keeps within its bandwidth allocation. It also defines transport-layer feedback messages, including negative acknowledgements (NACKs), that can be used to report on specific congestion events. RTP Codec Control Messages [RFC5104] extend the RTP/AVPF profile with additional feedback messages that can be used to influence that way in which rate adaptation occurs, but do not further change the dynamics of how rapidly feedback can be sent. Use of the RTP/AVPF profile is dependent on signalling.
- o Finally, Explicit Congestion Notification (ECN) for RTP over UDP [RFC6679] can be used to provide feedback on the number of packets that received an ECN Congestion Experienced (CE) mark. This RTCP extension builds on the RTP/AVPF profile to allow rapid congestion feedback when ECN is supported.

In addition to these mechanisms for providing feedback, the sender can include an RTP header extension in each packet to record packet transmission times [RFC5450]. Accurate transmission timestamps can be helpful for estimating queuing delays, to get an early indication of the onset of congestion.

Taken together, these various mechanisms allow receivers to provide feedback on the senders when congestion events occur, with varying degrees of timeliness and accuracy. The key distinction is between systems that use only the basic RTCP mechanisms, without RTP/AVPF rapid feedback, and those that use the RTP/AVPF extensions to respond to congestion more rapidly.

3. 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]. This interpretation of these key words applies only when written in ALL CAPS. Mixed- or lower-case uses of these key words are not to be interpreted as carrying special significance in this memo.

The definition of the RTP circuit breaker is specified in terms of the following variables:

- o Td is the deterministic RTCP reporting interval, as defined in Section 6.3.1 of [RFC3550].
- o Tdr is the sender's estimate of the deterministic RTCP reporting interval, Td, calculated by a receiver of the data it is sending. Tdr is not known at the sender, but can be estimated by executing the algorithm in Section 6.2 of [RFC3550] using the average RTCP packet size seen at the sender, the number of members reported in the receiver's SR/RR report blocks, and whether the receiver is sending SR or RR packets. Tdr is recalculated when each new RTCP SR/RR report is received, but the media timeout circuit breaker (see Section 4.2) is only reconsidered when Tdr increases.
- o Tr is the network round-trip time, calculated by the sender using the algorithm in Section 6.4.1 of [RFC3550] and smoothed using an exponentially weighted moving average as $Tr = (0.8 * Tr) + (0.2 * Tr_new)$ where Tr_new is the latest RTT estimate obtained from an RTCP report. The weight is chosen so old estimates decay over k intervals.
- o k is the non-reporting threshold (see Section 4.2).
- o Tf is the media framing interval at the sender. For applications sending at a constant frame rate, Tf is the inter-frame interval. For applications that switch between a small set of possible frame rates, for example when sending speech with comfort noise, where comfort noise frames are sent less often than speech frames, Tf is set to the longest of the inter-frame intervals of the different frame rates. For applications that send periodic frames but dynamically vary their frame rate, Tf is set to the largest inter-frame interval used in the last 10 seconds. For applications that send less than one frame every 10 seconds, or that have no concept of periodic frames (e.g., text conversation [RFC4103], or pointer events [RFC2862]), Tf is set to the time interval since the previous frame when each frame is sent.

- o G is the frame group size. That is, the number of frames that are coded together based on a particular sending rate setting. If the codec used by the sender can change its rate on each frame, $G = 1$; otherwise G is set to the number of frames before the codec can adjust to the new rate. For codecs that have the concept of a group-of-pictures (GoP), G is likely the GoP length.
- o $T_{rr_interval}$ is the minimal interval between RTCP reports, as defined in Section 3.4 of [RFC4585]; it is only meaningful for implementations of RTP/AVPF profile [RFC4585] or the RTP/SAVPF profile [RFC5124].
- o X is estimated throughput a TCP connection would achieve over a path, in bytes per second.
- o s is the size of RTP packets being sent, in bytes. If the RTP packets being sent vary in size, then the average size over the packet comprising the last $4 * G$ frames MUST be used (this is intended to be comparable to the four loss intervals used in [RFC5348]).
- o p is the loss event rate, between 0.0 and 1.0, that would be seen by a TCP connection over a particular path. When used in the RTP congestion circuit breaker, this is approximated as described in Section 4.3.
- o t_{RTO} is the retransmission timeout value that would be used by a TCP connection over a particular path, in seconds. This MUST be approximated using $t_{RTO} = 4 * Tr$ when used as part of the RTP congestion circuit breaker.
- o b is the number of packets that are acknowledged by a single TCP acknowledgement. Following [RFC3448], it is RECOMMENDED that the value $b = 1$ is used as part of the RTP congestion circuit breaker.

4. RTP Circuit Breakers for Systems Using the RTP/AVP Profile

The feedback mechanisms defined in [RFC3550] and available under the RTP/AVP profile [RFC3551] are the minimum that can be assumed for a baseline circuit breaker mechanism that is suitable for all unicast applications of RTP. Accordingly, for an RTP circuit breaker to be useful, it needs to be able to detect that an RTP flow is causing excessive congestion using only basic RTCP features, without needing RTCP XR feedback or the RTP/AVPF profile for rapid RTCP reports.

RTCP is a fundamental part of the RTP protocol, and the mechanisms described here rely on the implementation of RTCP. Implementations that claim to support RTP, but that do not implement RTCP, cannot use

the circuit breaker mechanisms described in this memo. Such implementations SHOULD NOT be used on networks that might be subject to congestion unless equivalent mechanisms are defined using some non-RTCP feedback channel to report congestion and signal circuit breaker conditions. The RTCP timeout circuit breaker (Section 4.1) will trigger if an implementation of this memo attempts to interwork with an endpoint that does not support RTCP. Implementations that sometimes need to interwork with endpoints that do not support RTCP need to disable the RTP circuit breakers if they don't receive some confirmation via signalling that the remote endpoint implements RTCP (the presence of an SDP "a=rtcp:" attribute in an answer might be such an indication).

Three potential congestion signals are available from the basic RTCP SR/RR packets and are reported for each synchronisation source (SSRC) in the RTP session:

1. The sender can estimate the network round-trip time once per RTCP reporting interval, based on the contents and timing of RTCP SR and RR packets.
2. Receivers report a jitter estimate (the statistical variance of the RTP data packet inter-arrival time) calculated over the RTCP reporting interval. Due to the nature of the jitter calculation ([RFC3550], section 6.4.4), the jitter is only meaningful for RTP flows that send a single data packet for each RTP timestamp value (i.e., audio flows, or video flows where each packet comprises one video frame).
3. Receivers report the fraction of RTP data packets lost during the RTCP reporting interval, and the cumulative number of RTP packets lost over the entire RTP session.

These congestion signals limit the possible circuit breakers, since they give only limited visibility into the behaviour of the network.

RTT estimates are widely used in congestion control algorithms, as a proxy for queuing delay measures in delay-based congestion control or to determine connection timeouts. RTT estimates derived from RTCP SR and RR packets sent according to the RTP/AVP timing rules are too infrequent to be useful for congestion control, and don't give enough information to distinguish a delay change due to routing updates from queuing delay caused by congestion. Accordingly, we cannot use the RTT estimate alone as an RTP circuit breaker.

Increased jitter can be a signal of transient network congestion, but in the highly aggregated form reported in RTCP RR packets, it offers insufficient information to estimate the extent or persistence of

congestion. Jitter reports are a useful early warning of potential network congestion, but provide an insufficiently strong signal to be used as a circuit breaker.

The remaining congestion signals are the packet loss fraction and the cumulative number of packets lost. If considered carefully, these can be effective indicators that congestion is occurring in networks where packet loss is primarily due to queue overflows, although loss caused by non-congestive packet corruption can distort the result in some networks. TCP congestion control [RFC5681] intentionally tries to fill the router queues, and uses the resulting packet loss as congestion feedback. An RTP flow competing with TCP traffic will therefore expect to see a non-zero packet loss fraction that has to be related to TCP dynamics to estimate available capacity. This behaviour of TCP is reflected in the congestion circuit breaker below, and will affect the design of any RTP congestion control protocol.

Two packet loss regimes can be observed: 1) RTCP RR packets show a non-zero packet loss fraction, while the extended highest sequence number received continues to increment; and 2) RR packets show a loss fraction of zero, but the extended highest sequence number received does not increment even though the sender has been transmitting RTP data packets. The former corresponds to the TCP congestion avoidance state, and indicates a congested path that is still delivering data; the latter corresponds to a TCP timeout, and is most likely due to a path failure. A third condition is that data is being sent but no RTCP feedback is received at all, corresponding to a failure of the reverse path. We derive circuit breaker conditions for these loss regimes in the following.

4.1. RTP/AVP Circuit Breaker #1: RTCP Timeout

An RTCP timeout can occur when RTP data packets are being sent, but there are no RTCP reports returned from the receiver. This is either due to a failure of the receiver to send RTCP reports, or a failure of the return path that is preventing those RTCP reporting from being delivered. In either case, it is not safe to continue transmission, since the sender has no way of knowing if it is causing congestion.

An RTP sender that has not received any RTCP SR or RTCP RR packets reporting on the SSRC it is using, for a time period of at least three times its deterministic RTCP reporting interval, T_d , without the randomization factor, and using the fixed minimum interval of $T_{min}=5$ seconds, SHOULD cease transmission (see Section 4.5). The rationale for this choice of timeout is as described in Section 6.2 of [RFC3550] ("so that implementations which do not use the reduced value for transmitting RTCP packets are not timed out by other

participants prematurely"), as updated by Section 6.1.4 of [I-D.ietf-avtcore-rtp-multi-stream] to account for the use of the RTP/AVPF profile [RFC4585] or the RTP/SAVPF profile [RFC5124].

To reduce the risk of premature timeout, implementations SHOULD NOT configure the RTCP bandwidth such that T_d is larger than 5 seconds. Similarly, implementations that use the RTP/AVPF profile [RFC4585] or the RTP/SAVPF profile [RFC5124] SHOULD NOT configure $T_{rr_interval}$ to values larger than 4 seconds (the reduced limit for $T_{rr_interval}$ follows Section 6.1.3 of [I-D.ietf-avtcore-rtp-multi-stream]).

The choice of three RTCP reporting intervals as the timeout is made following Section 6.3.5 of RFC 3550 [RFC3550]. This specifies that participants in an RTP session will timeout and remove an RTP sender from the list of active RTP senders if no RTP data packets have been received from that RTP sender within the last two RTCP reporting intervals. Using a timeout of three RTCP reporting intervals is therefore large enough that the other participants will have timed out the sender if a network problem stops the data packets it is sending from reaching the receivers, even allowing for loss of some RTCP packets.

If a sender is transmitting a large number of RTP media streams, such that the corresponding RTCP SR or RR packets are too large to fit into the network MTU, the receiver will generate RTCP SR or RR packets in a round-robin manner. In this case, the sender SHOULD treat receipt of an RTCP SR or RR packet corresponding to any SSRC it sent on the same 5-tuple of source and destination IP address, port, and protocol, as an indication that the receiver and return path are working, preventing the RTCP timeout circuit breaker from triggering.

4.2. RTP/AVP Circuit Breaker #2: Media Timeout

If RTP data packets are being sent, but the RTCP SR or RR packets reporting on that SSRC indicate a non-increasing extended highest sequence number received, this is an indication that those RTP data packets are not reaching the receiver. This could be a short-term issue affecting only a few RTP packets, perhaps caused by a slow to open firewall or a transient connectivity problem, but if the issue persists, it is a sign of a more ongoing and significant problem (a "media timeout").

The time needed to declare a media timeout depends on the parameters T_{dr} , T_r , T_f , and on the non-reporting threshold k . The value of k is chosen so that when T_{dr} is large compared to T_r and T_f , receipt of at least k RTCP reports with non-increasing extended highest sequence number received gives reasonable assurance that the forward path has

failed, and that the RTP data packets have not been lost by chance. The RECOMMENDED value for k is 5 reports.

When $T_{dr} < T_f$, then RTP data packets are being sent at a rate less than one per RTCP reporting interval of the receiver, so the extended highest sequence number received can be expected to be non-increasing for some receiver RTCP reporting intervals. Similarly, when $T_{dr} < T_r$, some receiver RTCP reporting intervals might pass before the RTP data packets arrive at the receiver, also leading to reports where the extended highest sequence number received is non-increasing. Both issues require the media timeout interval to be scaled relative to the threshold, k .

The media timeout RTP circuit breaker is therefore as follows. When starting sending, calculate MEDIA_TIMEOUT using:

$$\text{MEDIA_TIMEOUT} = \text{ceil}(k * \max(T_f, T_r, T_{dr}) / T_{dr})$$

When a sender receives an RTCP packet indicating that the media it's sending is being received, then it cancels the media timeout circuit breaker. If it is still sending, then it MUST calculate a new value for MEDIA_TIMEOUT, and set a new media timeout circuit breaker.

If a sender receives an RTCP packet indicating that its media was not received, it MUST calculate a new value for MEDIA_TIMEOUT. If the new value is larger than the previous, it replaces MEDIA_TIMEOUT with the new value, extending the media timeout circuit breaker; otherwise it keeps the original value of MEDIA_TIMEOUT. This process is known as reconsidering the media timeout circuit breaker.

If MEDIA_TIMEOUT consecutive RTCP packets are received indicating that the media being sent was not received, and the media timeout circuit breaker has not been cancelled, then the media timeout circuit breaker triggers. When the media timeout circuit breaker triggers, the sender SHOULD cease transmission (see Section 4.5).

When stopping sending an RTP stream, a sender MUST cancel the corresponding media timeout circuit breaker.

4.3. RTP/AVP Circuit Breaker #3: Congestion

If RTP data packets are being sent, and the corresponding RTCP SR or RR packets show non-zero packet loss fraction and increasing extended highest sequence number received, then those RTP data packets are arriving at the receiver, but some degree of congestion is occurring. The RTP/AVP profile [RFC3551] states that:

If best-effort service is being used, RTP receivers SHOULD monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve an average throughput, measured on a reasonable time scale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast session), or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

The comparison to TCP cannot be specified exactly, but is intended as an "order-of-magnitude" comparison in time scale and throughput. The time scale on which TCP throughput is measured is the round-trip time of the connection. In essence, this requirement states that it is not acceptable to deploy an application (using RTP or any other transport protocol) on the best-effort Internet which consumes bandwidth arbitrarily and does not compete fairly with TCP within an order of magnitude.

The phrase "order of magnitude" in the above means within a factor of ten, approximately. In order to implement this, it is necessary to estimate the throughput a TCP connection would achieve over the path. For a long-lived TCP Reno connection, it has been shown that the TCP throughput, X , in bytes per second, can be estimated using [Padhye]:

$$X = \frac{S}{Tr \cdot \sqrt{2 \cdot b \cdot p / 3} + (t_{RTO} \cdot (3 \cdot \sqrt{3 \cdot b \cdot p / 8} \cdot p \cdot (1 + 32 \cdot p \cdot p)))}$$

This is the same approach to estimated TCP throughput that is used in [RFC3448]. Under conditions of low packet loss the second term on the denominator is small, so this formula can be approximated with reasonable accuracy as follows [Mathis]:

$$X = \frac{S}{Tr \cdot \sqrt{2 \cdot b \cdot p / 3}}$$

It is RECOMMENDED that this simplified throughput equation be used, since the reduction in accuracy is small, and it is much simpler to calculate than the full equation. Measurements have shown that the simplified TCP throughput equation is effective as an RTP circuit breaker for multimedia flows sent to hosts on residential networks using ADSL and cable modem links [Singh]. The data shows that the full TCP throughput equation tends to be more sensitive to packet loss and triggers the RTP circuit breaker earlier than the simplified

equation. Implementations that desire this extra sensitivity MAY use the full TCP throughput equation in the RTP circuit breaker. Initial measurements in LTE networks have shown that the extra sensitivity is helpful in that environment, with the full TCP throughput equation giving a more balanced circuit breaker response than the simplified TCP equation [Sarker]; other networks might see similar behaviour.

No matter what TCP throughput equation is chosen, two parameters need to be estimated and reported to the sender in order to calculate the throughput: the round trip time, T_r , and the loss event rate, p (the packet size, s , is known to the sender). The round trip time can be estimated from RTCP SR and RR packets. This is done too infrequently for accurate statistics, but is the best that can be done with the standard RTCP mechanisms.

Report blocks in RTCP SR or RR packets contain the packet loss fraction, rather than the loss event rate, so p cannot be reported (TCP typically treats the loss of multiple packets within a single RTT as one loss event, but RTCP RR packets report the overall fraction of packets lost, and does not report when the packet losses occurred). Using the loss fraction in place of the loss event rate can overestimate the loss. We believe that this overestimate will not be significant, given that we are only interested in order of magnitude comparison ([Floyd] section 3.2.1 shows that the difference is small for steady-state conditions and random loss, but using the loss fraction is more conservative in the case of bursty loss).

The congestion circuit breaker is therefore: when a sender that is transmitting at least one RTP packet every $\max(T_{dr}, T_r)$ seconds receives an RTCP SR or RR packet that contains a report block for an SSRC it is using, the sender MUST record the value of the fraction lost field in the report block and the time since the last report block was received for that SSRC. If more than $CB_INTERVAL$ (see below) report blocks have been received for that SSRC, the sender MUST calculate the average fraction lost over the last $CB_INTERVAL$ reporting intervals, and then estimate the TCP throughput that would be achieved over the path using the chosen TCP throughput equation and the measured values of the round-trip time, T_r , the loss event rate, p (approximated by the average fraction lost, as is described below), and the packet size, s . The estimate of the TCP throughput, X , is then compared with the actual sending rate of the RTP stream. If the actual sending rate of the RTP stream is more than $10 * X$, then the congestion circuit breaker is triggered.

The average fraction lost is calculated based on the sum, over the last $CB_INTERVAL$ reporting intervals, of the fraction lost in each reporting interval multiplied by the duration of the corresponding reporting interval, divided by the total duration of the last

CB_INTERVAL reporting intervals. The CB_INTERVAL parameter is set to:

```
CB_INTERVAL =  
    ceil(3*min(max(10*G*Tf, 10*Tr, 3*Tdr), max(15, 3*Td))/(3*Tdr))
```

The parameters that feed into CB_INTERVAL are chosen to give the congestion control algorithm time to react to congestion. They give at least three RTCP reports, ten round trip times, and ten groups of frames to adjust the rate to reduce the congestion to a reasonable level. It is expected that a responsive congestion control algorithm will begin to respond with the next group of frames after it receives indication of congestion, so CB_INTERVAL ought to be a much longer interval than the congestion response.

If the RTP/AVPF profile [RFC4585] or the RTP/SAVPF [RFC5124] is used, and the T_rr_interval parameter is used to reduce the frequency of regular RTCP reports, then the value Tdr in the above expression for the CB_INTERVAL parameter MUST be replaced by max(T_rr_interval, Tdr).

The CB_INTERVAL parameter is calculated on joining the session, and recalculated on receipt of each RTCP packet, after checking whether the media timeout circuit breaker or the congestion circuit breaker has been triggered.

To ensure a timely response to persistent congestion, implementations SHOULD NOT configure the RTCP bandwidth such that Tdr is larger than 5 seconds. Similarly, implementations that use the RTP/AVPF profile [RFC4585] or the RTP/SAVPF profile [RFC5124] SHOULD NOT configure T_rr_interval to values larger than 4 seconds (the reduced limit for T_rr_interval follows Section 6.1.3 of [I-D.ietf-avtcore-rtp-multi-stream]).

The rationale for enforcing a minimum sending rate below which the congestion circuit breaker will not trigger is to avoid spurious circuit breaker triggers when the number of packets sent per RTCP reporting interval is small, and hence the fraction lost samples are subject to measurement artefacts. The bound of at least one packet every max(Tdr, Tr) seconds is derived from the one packet per RTT minimum sending rate of TCP [RFC5405], adapted for use with RTP where the RTCP reporting interval is decoupled from the network RTT.

When the congestion circuit breaker is triggered, the sender SHOULD cease transmission (see Section 4.5). However, if the sender is able to reduce its sending rate by a factor of (approximately) ten, then it MAY first reduce its sending rate by this factor (or some larger amount) to see if that resolves the congestion. If the sending rate

is reduced in this way and the congestion circuit breaker triggers again after the next CB_INTERVAL RTCP reporting intervals, the sender MUST then cease transmission. An example of such a rate reduction might be a video conferencing system that backs off to sending audio only, before completely dropping the call. If such a reduction in sending rate resolves the congestion problem, the sender MAY gradually increase the rate at which it sends data after a reasonable amount of time has passed, provided it takes care not to cause the problem to recur ("reasonable" is intentionally not defined here).

The RTCP reporting interval of the media sender does not affect how quickly congestion circuit breaker can trigger. The timing is based on the RTCP reporting interval of the receiver that generates the SR/RR packets from which the loss rate and RTT estimate are derived (note that RTCP requires all participants in a session to have similar reporting intervals, else the participant timeout rules in [RFC3550] will not work, so this interval is likely similar to that of the sender). If the incoming RTCP SR or RR packets are using a reduced minimum RTCP reporting interval (as specified in Section 6.2 of RFC 3550 [RFC3550] or the RTP/AVPF profile [RFC4585]), then that reduced RTCP reporting interval is used when determining if the circuit breaker is triggered.

If there are more media streams that can be reported in a single RTCP SR or RR packet, or if the size of a complete RTCP SR or RR packet exceeds the network MTU, then the receiver will report on a subset of sources in each reporting interval, with the subsets selected round-robin across multiple intervals so that all sources are eventually reported [RFC3550]. When generating such round-robin RTCP reports, priority SHOULD be given to reports on sources that have high packet loss rates, to ensure that senders are aware of network congestion they are causing (this is an update to [RFC3550]).

4.4. RTP/AVP Circuit Breaker #4: Media Usability

Applications that use RTP are generally tolerant to some amount of packet loss. How much packet loss can be tolerated will depend on the application, media codec, and the amount of error correction and packet loss concealment that is applied. There is an upper bound on the amount of loss can be corrected, however, beyond which the media becomes unusable. Similarly, many applications have some upper bound on the media capture to play-out latency that can be tolerated before the application becomes unusable. The latency bound will depend on the application, but typical values can range from the order of a few hundred milliseconds for voice telephony and interactive conferencing applications, up to several seconds for some video-on-demand systems.

As a final circuit breaker, RTP senders SHOULD monitor the reported packet loss and delay to estimate whether the media is likely to be suitable for the intended purpose. If the packet loss rate and/or latency is such that the media has become unusable, and has remained unusable for a significant time period, then the application SHOULD cease transmission. Similarly, receivers SHOULD monitor the quality of the media they receive, and if the quality is unusable for a significant time period, they SHOULD terminate the session. This memo intentionally does not define a bound on the packet loss rate or latency that will result in unusable media, nor does it specify what time period is deemed significant, as these are highly application dependent.

Sending media that suffers from such high packet loss or latency that it is unusable at the receiver is both wasteful of resources, and of no benefit to the user of the application. It also is highly likely to be congesting the network, and disrupting other applications. As such, the congestion circuit breaker will almost certainly trigger to stop flows where the media would be unusable due to high packet loss or latency. However, in pathological scenarios where the congestion circuit breaker does not stop the flow, it is desirable that the RTP application cease sending useless traffic. The role of the media usability circuit breaker is to protect the network in such cases.

4.5. Ceasing Transmission

What it means to cease transmission depends on the application, but the intention is that the application will stop sending RTP data packets to a particular destination 3-tuple (transport protocol, destination port, IP address), until the user makes an explicit attempt to restart the call. It is important that a human user is involved in the decision to try to restart the call, since that user will eventually give up if the calls repeatedly trigger the circuit breaker. This will help avoid problems with automatic redial systems from congesting the network. Accordingly, RTP flows halted by the circuit breaker SHOULD NOT be restarted automatically unless the sender has received information that the congestion has dissipated.

It is recognised that the RTP implementation in some systems might not be able to determine if a call set-up request was initiated by a human user, or automatically by some scripted higher-level component of the system. These implementations SHOULD rate limit attempts to restart a call to the same destination 3-tuple as used by a previous call that was recently halted by the circuit breaker. The chosen rate limit ought to not exceed the rate at which an annoyed human caller might redial a misbehaving phone.

5. RTP Circuit Breakers and the RTP/AVPF and RTP/SAVPF Profiles

Use of the Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [RFC4585] allows receivers to send early RTCP reports in some cases, to inform the sender about particular events in the media stream. There are several use cases for such early RTCP reports, including providing rapid feedback to a sender about the onset of congestion. The RTP/SAVPF Profile [RFC5124] is a secure variant of the RTP/AVPF profile, that is treated the same in the context of the RTP circuit breaker. These feedback profiles are often used with non-compound RTCP reports [RFC5506] to reduce the reporting overhead.

Receiving rapid feedback about congestion events potentially allows congestion control algorithms to be more responsive, and to better adapt the media transmission to the limitations of the network. It is expected that many RTP congestion control algorithms will adopt the RTP/AVPF profile or the RTP/SAVPF profile for this reason, defining new transport layer feedback reports that suit their requirements. Since these reports are not yet defined, and likely very specific to the details of the congestion control algorithm chosen, they cannot be used as part of the generic RTP circuit breaker.

Reduced-size RTCP reports sent under the RTP/AVPF early feedback rules that do not contain an RTCP SR or RR packet MUST be ignored by the congestion circuit breaker (they do not contain the information needed by the congestion circuit breaker algorithm), but MUST be counted as received packets for the RTCP timeout circuit breaker. Reduced-size RTCP reports sent under the RTP/AVPF early feedback rules that contain RTCP SR or RR packets MUST be processed by the congestion circuit breaker as if they were sent as regular RTCP reports, and counted towards the circuit breaker conditions specified in Section 4 of this memo. This will potentially make the RTP circuit breaker trigger earlier than it would if the RTP/AVPF profile was not used.

When using ECN with RTP (see Section 7), early RTCP feedback packets can contain ECN feedback reports. The count of ECN-CE marked packets contained in those ECN feedback reports is counted towards the number of lost packets reported if the ECN Feedback Report report is sent in an compound RTCP packet along with an RTCP SR/RR report packet. Reports of ECN-CE packets sent as reduced-size RTCP ECN feedback packets without an RTCP SR/RR packet MUST be ignored.

These rules are intended to allow the use of low-overhead RTP/AVPF feedback for generic NACK messages without triggering the RTP circuit breaker. This is expected to make such feedback suitable for RTP congestion control algorithms that need to quickly report loss events

in between regular RTCP reports. The reaction to reduced-size RTCP SR/RR packets is to allow such algorithms to send feedback that can trigger the circuit breaker, when desired.

The RTP/AVPF and RTP/SAVPF profiles include the `T_rr_interval` parameter that can be used to adjust the regular RTCP reporting interval. The use of the `T_rr_interval` parameter changes the behaviour of the RTP circuit breaker, as described in Section 4.

6. Impact of RTCP Extended Reports (XR)

RTCP Extended Report (XR) blocks provide additional reception quality metrics, but do not change the RTCP timing rules. Some of the RTCP XR blocks provide information that might be useful for congestion control purposes, others provided non-congestion-related metrics. With the exception of RTCP XR ECN Summary Reports (see Section 7), the presence of RTCP XR blocks in a compound RTCP packet does not affect the RTP circuit breaker algorithm. For consistency and ease of implementation, only the reception report blocks contained in RTCP SR packets, RTCP RR packets, or RTCP XR ECN Summary Report packets, are used by the RTP circuit breaker algorithm.

7. Impact of Explicit Congestion Notification (ECN)

The use of ECN for RTP flows does not affect the media timeout RTP circuit breaker (Section 4.2) or the RTCP timeout circuit breaker (Section 4.1), since these are both connectivity checks that simply determinate if any packets are being received.

ECN-CE marked packets SHOULD be treated as if it were lost for the purposes of congestion control, when determining the optimal media sending rate for an RTP flow. If an RTP sender has negotiated ECN support for an RTP session, and has successfully initiated ECN use on the path to the receiver [RFC6679], then ECN-CE marked packets SHOULD be treated as if they were lost when calculating if the congestion-based RTP circuit breaker (Section 4.3) has been met. The count of ECN-CE marked RTP packets is returned in RTCP XR ECN summary report packets if support for ECN has been initiated for an RTP session.

8. Impact of Bundled Media and Layered Coding

The RTP circuit breaker operates on a per-RTP session basis. An RTP sender that participates in several RTP sessions MUST treat each RTP session independently with regards to the RTP circuit breaker.

An RTP sender can generate several media streams within a single RTP session, with each stream using a different SSRC. This can happen if bundled media are in use, when using simulcast, or when using layered

media coding. By default, each SSRC will be treated independently by the RTP circuit breaker. However, the sender MAY choose to treat the flows (or a subset thereof) as a group, such that a circuit breaker trigger for one flow applies to the group of flows as a whole, and either causes the entire group to cease transmission, or the sending rate of the group to reduce by a factor of ten, depending on the RTP circuit breaker triggered. Grouping flows in this way is expected to be especially useful for layered flows sent using multiple SSRCs, as it allows the layered flow to react as a whole, ceasing transmission on the enhancement layers first to reduce sending rate if necessary, rather than treating each layer independently.

9. Security Considerations

The security considerations of [RFC3550] apply.

If the RTP/AVPF profile is used to provide rapid RTCP feedback, the security considerations of [RFC4585] apply. If ECN feedback for RTP over UDP/IP is used, the security considerations of [RFC6679] apply.

If non-authenticated RTCP reports are used, an on-path attacker can trivially generate fake RTCP packets that indicate high packet loss rates, causing the circuit breaker to trigger and disrupting an RTP session. This is somewhat more difficult for an off-path attacker, due to the need to guess the randomly chosen RTP SSRC value and the RTP sequence number. This attack can be avoided if RTCP packets are authenticated; authentication options are discussed in [RFC7201].

Timely operation of the RTP circuit breaker depends on the choice of RTCP reporting interval. If the receiver has a reporting interval that is overly long, then the responsiveness of the circuit breaker decreases. In the limit, the RTP circuit breaker can be disabled for all practical purposes by configuring an RTCP reporting interval that is many minutes duration. This issue is not specific to the circuit breaker: long RTCP reporting intervals also prevent reception quality reports, feedback messages, codec control messages, etc., from being used. Implementations are expected to impose an upper limit on the RTCP reporting interval they are willing to negotiate (based on the session bandwidth and RTCP bandwidth fraction) when using the RTP circuit breaker, as discussed in Section 4.3.

10. IANA Considerations

There are no actions for IANA.

11. Acknowledgements

The authors would like to thank Bernard Aboba, Harald Alvestrand, Gorrry Fairhurst, Nazila Fough, Kevin Gross, Cullen Jennings, Randell Jesup, Jonathan Lennox, Matt Mathis, Stephen McQuistin, Simon Perreault, Eric Rescorla, Abheek Saha, Fabio Verdicchio, and Magnus Westerlund for their valuable feedback.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, DOI 10.17487/RFC3448, January 2003, <<http://www.rfc-editor.org/info/rfc3448>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<http://www.rfc-editor.org/info/rfc3551>>.
- [RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <<http://www.rfc-editor.org/info/rfc3611>>.
- [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, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.

12.2. Informative References

- [Floyd] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", Proceedings of the ACM SIGCOMM conference, 2000, DOI 10.1145/347059.347397, August 2000.
- [I-D.ietf-avtcore-rtp-multi-stream] Lennox, J., Westerlund, M., Wu, W., and C. Perkins, "Sending Multiple Media Streams in a Single RTP Session", draft-ietf-avtcore-rtp-multi-stream-09 (work in progress), September 2015.
- [I-D.ietf-tsvwg-circuit-breaker] Fairhurst, G., "Network Transport Circuit Breakers", draft-ietf-tsvwg-circuit-breaker-05 (work in progress), October 2015.
- [Mathis] Mathis, M., Semke, J., Mahdavi, J., and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm", ACM SIGCOMM Computer Communication Review 27(3), DOI 10.1145/263932.264023, July 1997.
- [Padhye] Padhye, J., Firoiu, V., Towsley, D., and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", Proceedings of the ACM SIGCOMM conference, 1998, DOI 10.1145/285237.285291, August 1998.
- [RFC2862] Civanlar, M. and G. Cash, "RTP Payload Format for Real-Time Pointers", RFC 2862, DOI 10.17487/RFC2862, June 2000, <<http://www.rfc-editor.org/info/rfc2862>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005, <<http://www.rfc-editor.org/info/rfc4103>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, DOI 10.17487/RFC5104, February 2008, <<http://www.rfc-editor.org/info/rfc5104>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<http://www.rfc-editor.org/info/rfc5124>>.

- [RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, DOI 10.17487/RFC5348, September 2008, <<http://www.rfc-editor.org/info/rfc5348>>.
- [RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, DOI 10.17487/RFC5405, November 2008, <<http://www.rfc-editor.org/info/rfc5405>>.
- [RFC5450] Singer, D. and H. Desineni, "Transmission Time Offsets in RTP Streams", RFC 5450, DOI 10.17487/RFC5450, March 2009, <<http://www.rfc-editor.org/info/rfc5450>>.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, DOI 10.17487/RFC5506, April 2009, <<http://www.rfc-editor.org/info/rfc5506>>.
- [RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <<http://www.rfc-editor.org/info/rfc5681>>.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <<http://www.rfc-editor.org/info/rfc6679>>.
- [RFC6798] Clark, A. and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Packet Delay Variation Metric Reporting", RFC 6798, DOI 10.17487/RFC6798, November 2012, <<http://www.rfc-editor.org/info/rfc6798>>.
- [RFC6843] Clark, A., Gross, K., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Delay Metric Reporting", RFC 6843, DOI 10.17487/RFC6843, January 2013, <<http://www.rfc-editor.org/info/rfc6843>>.
- [RFC6958] Clark, A., Zhang, S., Zhao, J., and Q. Wu, Ed., "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Loss Metric Reporting", RFC 6958, DOI 10.17487/RFC6958, May 2013, <<http://www.rfc-editor.org/info/rfc6958>>.
- [RFC7002] Clark, A., Zorn, G., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Discard Count Metric Reporting", RFC 7002, DOI 10.17487/RFC7002, September 2013, <<http://www.rfc-editor.org/info/rfc7002>>.

- [RFC7003] Clark, A., Huang, R., and Q. Wu, Ed., "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Discard Metric Reporting", RFC 7003, DOI 10.17487/RFC7003, September 2013, <<http://www.rfc-editor.org/info/rfc7003>>.
- [RFC7097] Ott, J., Singh, V., Ed., and I. Curcio, "RTP Control Protocol (RTCP) Extended Report (XR) for RLE of Discarded Packets", RFC 7097, DOI 10.17487/RFC7097, January 2014, <<http://www.rfc-editor.org/info/rfc7097>>.
- [RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <<http://www.rfc-editor.org/info/rfc7201>>.
- [Sarker] Sarker, Z., Singh, V., and C. Perkins, "An Evaluation of RTP Circuit Breaker Performance on LTE Networks", Proceedings of the IEEE Infocom workshop on Communication and Networking Techniques for Contemporary Video, 2014, April 2014.
- [Singh] Singh, V., McQuistin, S., Ellis, M., and C. Perkins, "Circuit Breakers for Multimedia Congestion Control", Proceedings of the International Packet Video Workshop, 2013, DOI 10.1109/PV.2013.6691439, December 2013.

Authors' Addresses

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
United Kingdom

Email: csp@csperkins.org

Varun Singh
Aalto University
School of Electrical Engineering
Otakaari 5 A
Espoo, FIN 02150
Finland

Email: varun@comnet.tkk.fi
URI: <http://www.netlab.tkk.fi/~varun/>