Abstract

This memo discusses use of RTP in the context of the RTC-Web activity. It discusses important features of RTP that need to be considered by other parts of the RTC-Web framework, describes which RTP profile to use in this environment, and outlines what RTP extensions should be supported.

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 December 7, 2011.

Copyright Notice

Copyright (c) 2011 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
Table of Contents

1.  Introduction . . . . . . . . . . . . . . . . . . . . . . . . .  3
    1.1.  Expected Topologies . . . . . . . . . . . . . . . . . . .  3
2.  Requirements from RTP . . . . . . . . . . . . . . . . . . . .  6
    2.1.  RTP Multiplexing Points . . . . . . . . . . . . . . . . .  6
    2.2.  Signalling for RTP sessions . . . . . . . . . . . . . . .  8
    2.3.  (Lack of) Signalling for Payload Format Changes . . . .  9
3.  RTP Profile . . . . . . . . . . . . . . . . . . . . . . . . . 10
4.  RTP and RTCP Guidelines . . . . . . . . . . . . . . . . . . . 10
5.  RTP Optimizations . . . . . . . . . . . . . . . . . . . . . . 11
    5.1.  RTP and RTCP Multiplexing . . . . . . . . . . . . . . . 11
    5.2.  Reduced Size RTCP . . . . . . . . . . . . . . . . . . . 11
    5.3.  Symmetric RTP/RTCP . . . . . . . . . . . . . . . . . . . 11
    5.4.  Generation of the RTCP Canonical Name (CNAME) . . . . . 12
6.  RTP Extensions . . . . . . . . . . . . . . . . . . . . . . . . 12
    6.1.  RTP Conferencing Extensions . . . . . . . . . . . . . . 12
        6.1.1.  RTCP Feedback Message: Full Intra Request . . . . . 13
        6.1.2.  RTCP Feedback Message: Picture Loss Indicator . . . 13
        6.1.3.  RTCP Feedback Message: Temporary Maximum Media
                Stream Bit Rate Request . . . . . . . . . . . . . . . 14
    6.2.  RTP Header Extensions . . . . . . . . . . . . . . . . . 14
    6.3.  Rapid Synchronisation Extensions . . . . . . . . . . . . 15
7.  Improving RTP Transport Robustness . . . . . . . . . . . . . . 15
    7.1.  RTP Retransmission . . . . . . . . . . . . . . . . . . . 15
    7.2.  Forward Error Correction (FEC) . . . . . . . . . . . . . 15
8.  RTP Rate Control and Media Adaptation . . . . . . . . . . . . . 16
9.  RTP Performance Monitoring . . . . . . . . . . . . . . . . . . 16
10. IANA Considerations . . . . . . . . . . . . . . . . . . . . . 16
11. Security Considerations . . . . . . . . . . . . . . . . . . . . 16
12. Acknowledgements . . . . . . . . . . . . . . . . . . . . . . 17
13. References . . . . . . . . . . . . . . . . . . . . . . . . . . 17
    13.1. Normative References . . . . . . . . . . . . . . . . . . 17
    13.2. Informative References . . . . . . . . . . . . . . . . . 19
Authors’ Addresses . . . . . . . . . . . . . . . . . . . . . . . 19

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.
1. Introduction

This memo discusses the Real-time Transport Protocol (RTP) [RFC3551] in the context of the RTC-Web activity. The work in the IETF Audio/Video Transport Working Group, and its successors, has been about providing building blocks for real-time multimedia transport, and has not specified who should use which building blocks. The selection of building blocks and functionalities can really only be done in the context of some application, for example RTC-Web. We have selected a set of RTP features and extensions that are suitable for a number of applications that fits the RTC-Web context. Thus applications such as VoIP, audio and video conferencing, and on-demand multimedia streaming are considered. Applications that rely on IP multicast have not been considered likely to be applicable to RTC-Web, thus extensions related to multicast have been excluded. We believe that RTC-Web will greatly benefit in interoperability if a reasonable set of RTP functionalities and extensions are selected. This memo is intended as a starting point for discussion of those features in the RTC-Web framework.

This memo is structured into different topics. For each topic, one or several recommendations from the authors are done. When it comes to the importance of extensions, or the need for implementation support, we use three requirement levels to indicate the importance of the feature to the RTC-Web specification:

REQUIRED: Functionality that is absolutely needed to make the RTC-Web solution work well, or functionality of low complexity that provides high value.

RECOMMENDED: Should be included as it brings significant benefit, but the solution can potentially work without it.

OPTIONAL: Something that is useful in some cases, but not always a benefit.

When this memo discusses RTP, it includes the RTP Control Protocol (RTCP) unless explicitly stated otherwise. RTCP is a fundamental and integral part of the RTP protocol, and is REQUIRED to be implemented.

1.1. Expected Topologies

As RTC-Web is focused on peer to peer connections established from clients in web browsers the following topologies further discussed in RTP Topologies [RFC5117] are primarily considered. The topologies are depicted and briefly explained here for ease of the reader.
The point to point topology (Figure 1) is going to be very common in any single user to single user applications.

![Figure 1: Point to Point](image)

For small multiparty sessions it is practical enough to create RTP sessions by letting every participant send individual unicast RTP/UDP flows to each of the other participants. This is called multi-unicast and is unfortunately not discussed in the RTP Topologies [RFC5117]. This topology has the benefit of not requiring central nodes. On the downside is that it increase the used bandwidth by requiring one copy of the media streams for each participant part of the same session beyond the sender itself. Thus this is limited to scenarios with few end-points unless the media is very low bandwidth.

It needs to be noted that if this topology is to be supported by the RTC-Web framework it needs to be possible to connect one RTP session to multiple established peer to peer flows that are individually established.

![Figure 2: Multi-unicast](image)

![Figure 3: RTP Mixer with Only Unicast Paths](image)
An RTP mixer (Figure 3) is a centralized point that selects or mixes content in a conference to optimize the RTP session so that each end-point only needs to connect to one entity, the mixer. The mixer also reduces the bit-rate needs as the media sent from the mixer to the end-point can be optimized in different ways. These optimizations include methods like only choosing media from the currently most active speaker or mixing together audio so that only one audio stream is required instead of 3 in the depicted scenario. The downside of the mixer is that someone is required to provide the actual mixer.

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

**Figure 4: RTP Translator (Relay) with Only Unicast Paths**

If one wants a less complex central node it is possible to use a relay (called an Transport Translator) (Figure 4) that takes on the role of forwarding the media to the other end-points but doesn’t perform any media processing. It simply forwards the media from all other to all the other. Thus one endpoint A will only need to send a media once to the relay, but it will still receive 3 RTP streams with the media if B, C and D all currently transmits.

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

**Figure 5: Translator towards Legacy end-point**

To support legacy end-point (B) that don’t fulfill the requirements of RTC-Web it is possible to insert a Translator (Figure 5) that takes on the role to ensure that from A’s perspective B looks like a fully compliant end-point. Thus it is the combination of the Translator and B that looks like the end-point B. The intention is that the presence of the translator is transparent to A, however it is not certain that it is possible. Thus this case is included so that it can be discussed if any mechanism specified to be used for RTC-Web results in such issues and how to handle them.
2. Requirements from RTP

This section discusses some requirements RTP and RTCP [RFC3550] place on their underlying transport protocol, the signalling channel, etc.

2.1. RTP Multiplexing Points

There are three fundamental points of multiplexing within the RTP framework:

Use of separate RTP Sessions: The first, and the most important, multiplexing point is the RTP session. This multiplexing point does not have an identifier within the RTP protocol itself, but instead relies on the lower layer to separate the different RTP sessions. This is most often done by separating different RTP sessions onto different UDP ports, or by sending to different IP multicast addresses. The distinguishing feature of an RTP session is that it has a separate SSRC identifier space; a single RTP session can span multiple transport connections provided packets are gatewayed such that participants are known to each other. Different RTP sessions are used to separate different types of media within a multimedia session. For example, audio and video flows are sent on separate RTP sessions.

Multiplexing using the SSRC within an RTP session: The second multiplexing point is the SSRC that separates different sources of media within a single RTP session. An example might be different participants in a multiparty teleconference, or different camera views of a presentation. In most cases, each participant within an RTP session has a single SSRC, although this may change over time if collisions are detected. However, in some more complex scenarios participants may generate multiple media streams of the same type simultaneously (e.g., if they have two cameras, and so send two video streams at once) and so will have more than one SSRC in use at once. The RTCP CNAME can be used to distinguish between a single participant using two SSRC values (where the RTCP CNAME will be the same for each SSRC), and two participants (who will have different RTCP CNAMEs).

Multiplexing using the Payload Type within an RTP session: If different media encodings of the same type are to be used at different times within an RTP session, for example a single participant that can switch between two different audio codecs, the payload type is used to identify how the media from that particular source is encoded. When changing media formats within an RTP Session, the SSRC of the sender remains unchanged, but the RTP Payload Type changes to indicate the change in media format.
These multiplexing points are fundamental part of the design of RTP and are discussed in Section 5.2 of [RFC3550]. Of special importance is the need to separate different RTP sessions using a multiplexing mechanism at some lower layer than RTP, rather than trying to combine several RTP sessions into one lower layer flow.

The processing that can happen in an RTP mixer, translator or in an end-point is dependent on the purpose and media type of the stream, as determined by the RTP session on which it arrives. Hence, it is important to separate such RTP session from each other. This could of course be achieved by other methods, like tagging SSRC values with their purpose (this is not defined in any known specification), but there are reasons why this method isn’t defined. First of all it is not the simple solution, as this require additional signalling, and possibly synchronization between session peers. In addition, combining RTP sessions into a single lower-layer flow complicates quality of service and traffic engineering between the media flows in different RTP sessions. By using different transport layer ports, QoS mechanism that are capable of operating on the 5-tuple (Source address, port, destination address, port, and protocol) can be used without modification on RTP.

There are also various other RTP mechanism that become problematic if one doesn’t have a clear separation of RTP sessions:

Scalability: RTP was built with media scalability in consideration. The simplest way of achieving separation between different scalability layers are placing them in different RTP sessions, and using the same SSRC and CNAME in each session to bind them together. This is most commonly done in multicast, and not particular applicable to RTC-Web, but gatewaying of such a session would then require more alterations and likely stateful translation.

RTP Retransmission in Session Multiplexing mode: RTP Retransmission [RFC4588] does have a mode for session multiplexing. This would not be the main mode used in RTC-Web, but for interoperability and reduced cost in translation support for different RTP Sessions are required.

Forward Error Correction: The "An RTP Payload Format for Generic Forward Error Correction" [RFC2733] and its update [RFC5109] can only be used on media formats that produce RTP packets that are smaller than half the MTU if the FEC flow and media flow being protected are to be sent in the same RTP session, this is due to "RTP Payload for Redundant Audio Data" [RFC2198]. This is because the SSRC value of the original flow is recovered from the FEC packets SSRC field. So for anything that desires to use these
format with RTP payloads that are close to MTU needs to put the FEC data in a separate RTP session compared to the original transmissions.

RTCP behavior also becomes a factor in why overloading RTP sessions is problematic. The extension mechanisms used in RTCP depends on the media streams. For example the Extended RTCP report block for VoIP is of suitable for conversational audio, but clearly not useful for Video. This has three impacts, either one get unusable reports if they are generated for streams where there are little purpose. This is maybe less likely for the VoIP report, but for example the more detailed media agnostic reports it may occur. It otherwise makes the implementation of RTCP more complex as the SSRC purpose tagging needs not only to be one the media side, but also on the RTCP reporting. Also the RTCP reporting interval and transmission scheduling will be affected.

Due to these design principle implementors of various services or applications using RTP have not commonly violated this model. If one choses to violate it today, one fails to achieve interoperability with a number of existing services, applications and implementations.

As a conclusion not ensuring that RTP sessions are used for its intended purpose as a multiplexing point does violate the RTP design philosophy. It prevents the use of certain RTP extensions. It will require additional extensions to function and will significantly increase the complexity of the implementation. At the same time it will significantly reduce the interoperability with current implementations. Thus the authors considered it REQUIRED that RTP sessions are multiplexed using a mechanism outside of RTP. The RECOMMENDED mechanism to accomplish that would be to use unique UDP flows. If the WG comes to a consensus that due to NAT/Firewall traversal aspects would be greatly simplified with a single flow between peers and accept that flow based QoS can only be done on the aggreage of all RTP sessions then the authors RECOMMEND that some type of multiplexing layer is inserted between UDP flow and the RTP/RTCP header to separate the RTP sessions.

2.2. Signalling for RTP sessions

RTP is built with the assumption of an external to RTP/RTCP signalling channel to configure the RTP sessions and its functions. The basic configuration of an RTP session consists of the following parameters:
RTP Profile: The name of the RTP profile to be used in session. The RTP/AVP [RFC3551] and RTP/AVPF [RFC4585] profiles can interoperate on basic level, as can their secure variants RTP/SAVP [RFC3711] and RTP/SAVPF [RFC5124]. The secure variants of the profiles do not directly interoperate with the non-secure variants, due to the presence of additional header fields in addition to any cryptographic transformation of the packet content.

Transport Information: Source and destination address(s) and ports for RTP and RTCP must be signalled for each RTP session. If RTP and RTCP multiplexing [RFC5761] is to be used, such that a single port is used for RTP and RTCP flows, this must be signalled.

RTP Payload Types, media formats, and media format parameters: The mapping between media type names (and hence the RTP payload formats to be used) and the RTP payload type numbers must be signalled. Each media type may also have a number of media type parameters that must also be signalled to configure the codec and RTP payload format (the "a=fmtp:" line from SDP).

RTP Extensions: The RTP extensions one intends to use needs to be agreed on, including any parameters for that extension. In some case just to avoid spending bit-rate on features that the other end-point will ignore. But for certain mechanisms there is requirement for this to happen as interoperability failure otherwise happens.

RTCP Bandwidth: Support for exchanging RTCP Bandwidth values to the end-points will be necessary, as described in "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth" [RFC3556], or something semantically equivalent. This also ensures that the end-points have a common view of the RTCP bandwidth, this is important as too different view of the bandwidths may lead to failure to interoperate.

These parameters are often expressed in SDP messages conveyed within an offer/answer exchange. RTP does not depend on SDP or on the offer/answer model, but does require all the necessary parameters to be negotiated somehow, and provided to the RTP implementation.

2.3. (Lack of) Signalling for Payload Format Changes

As discussed in Section 2.2, the mapping between media type name, and its associated RTP payload format, and the RTP payload type number to be used for that format must be signalled as part of the session setup. An endpoint may signal support for multiple media formats, or multiple configurations of a single format, each using a different RTP payload type number. If multiple formats are signalled by an
endpoint, that endpoint is REQUIRED to be prepared to receive data encoded in any of those formats at any time. RTP does not require advance signalling for changes between formats that were signalled during the session setup. This is needed for rapid rate adaptation.

3. RTP Profile

The "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)" [RFC5124] is REQUIRED to be implemented. This builds on the basic RTP/AVP profile [RFC3551], the RTP/AVPF feedback profile [RFC4585], and the secure RTP/SAVP profile [RFC3711].

The RTP/AVPF part of RTP/SAVPF is required to get the improved RTCP timer model, that allows more flexible transmission of RTCP packets in response to events, rather than strictly according to bandwidth. This also saves RTCP bandwidth and will commonly only utilize the full amount when there is a lot of events on which to send feedback. This functionality is needed to make use of the RTP conferencing extensions discussed in Section 6.1.

The RTP/SAVP part of RTP/SAVPF is for support for Secure RTP (SRTP) [RFC3711]. This provides media encryption, integrity protection, replay protection and a limited form of source authentication. It does not contain a specific keying mechanism, so that, and the set of security transforms, will be required to be chosen. It is possible that a security mechanism operating on a lower layer than RTP can be used instead and that should be evaluated. However, the reasons for the design of SRTP should be taken into consideration in that discussion.

4. RTP and RTCP Guidelines

RTP and RTCP are two flexible and extensible protocols that allow, on the one hand, choosing from a variety of building blocks and combining those to meet application needs, and on the other hand, create extensions where existing mechanisms are not sufficient: from new payload formats to RTP extension headers to additional RTCP control packets.

Different informational documents provide guidelines to the use and particularly the extension of RTP and RTCP, including the following: Guidelines for Writers of RTP Payload Format Specifications [RFC2736] and Guidelines for Extending the RTP Control Protocol [RFC5968].
5. RTP Optimizations

This section discusses some optimizations that makes RTP/RTCP work better and more efficient and therefore are considered.

5.1. RTP and RTCP Multiplexing

Historically, RTP and RTCP have been run on separate UDP ports. With the increased use of Network Address/Port Translation (NAPT) this has become problematic, since maintaining multiple NAT bindings can be costly. It also complicates firewall administration, since multiple ports must be opened to allow RTP traffic. To reduce these costs and session setup times, support for multiplexing RTP data packets and RTCP control packets on a single port [RFC5761] is REQUIRED. Supporting this specification is generally a simplification in code, since it relaxes the tests in [RFC3550].

Note that the use of RTP and RTCP multiplexed on a single port ensures that there is occasional traffic sent on that port, even if there is no active media traffic. This may be useful to keep-alive NAT bindings.

5.2. Reduced Size RTCP

RTCP packets are usually sent as compound RTCP packets; and RFC 3550 demands that those compound packets always start with an SR or RR packet. However, especially when using frequent feedback messages, these general statistics are not needed in every packet and unnecessarily increase the mean RTCP packet size and thus limit the frequency at which RTCP packets can be sent within the RTCP bandwidth share.

RFC5506 "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences" [RFC5506] specifies how to reduce the mean RTCP message and allow for more frequent feedback. Frequent feedback, in turn, is essential to make real-time application quickly aware of changing network conditions and allow them to adapt their transmission and encoding behavior. Supporting this specification is generally a simplification in code, since it relaxes the tests in [RFC3550].

Support for RFC5506 is REQUIRED.

5.3. Symmetric RTP/RTCP

RTP entities choose the RTP and RTCP transport addresses, i.e., IP addresses and port numbers, to receive packets on and bind their respective sockets to those. When sending RTP packets, however, they
may use a different IP address or port number for RTP, RTCP, or both; e.g., when using a different socket instance for sending and for receiving. Symmetric RTP/RTCP requires that the IP address and port number for sending and receiving RTP/RTCP packets are identical.

The reasons for using symmetric RTP is primarily to avoid issues with NAT and Firewalls by ensuring that the flow is actually bi-directional and thus kept alive and registered as flow the intended recipient actually wants. In addition it saves resources in the form of ports at the end-points, but also in the network as NAT mappings or firewall state is not unnecessary bloated. Also the number of QoS state are reduced.

Using Symmetric RTP and RTCP [RFC4961] is REQUIRED.

5.4. Generation of the RTCP Canonical Name (CNAME)

The RTCP Canonical Name (CNAME) provides a persistent transport-level identifier for an RTP endpoint. While the Synchronization Source (SSRC) identifier for an RTP endpoint may change if a collision is detected, or when the RTP application is restarted, it’s RTCP CNAME is meant to stay unchanged, so that RTP endpoints can be uniquely identified and associated with their RTP media streams. For proper functionality, RTCP CNAMEs should be unique within the participants of an RTP session.

The RTP specification [RFC3550] includes guidelines for choosing a unique RTP CNAME, but these are not sufficient in the presence of NAT devices. In addition, some may find long-term persistent identifiers problematic from a privacy viewpoint. Accordingly, support for generating the RTCP CNAME as specified in "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)" [RFC6222] is RECOMMENDED, since this addresses both concerns.

6. RTP Extensions

There are a number of RTP extensions that could be very useful in the RTC-Web context. One set is related to conferencing, others are more generic in nature.

6.1. RTP Conferencing Extensions

RTP is inherently defined for group communications, whether using IP multicast, multi-unicast, or based on a centralised server. In today’s practice, however, overlay-based conferencing dominates, typically using one or a few so-called conference bridges or servers to connect endpoints in a star or flat tree topology. Quite diverse
conferencing topologies can be created using the basic elements of RTP mixers and translators as defined in RFC 3550.

An number of conferencing topologies are defined in [RFC5117] out of the which the following ones are the more common (and most likely in practice workable) ones:

1) RTP Translator (Relay) with Only Unicast Paths (RFC 5117, section 3.3)

2) RTP Mixer with Only Unicast Paths (RFC 5117, section 3.4)

3) Point to Multipoint Using a Video Switching MCU (RFC 5117, section 3.5)

4) Point to Multipoint Using Content Modifying MCUs (RFC 5117, section 3.6)

We note that 3 and 4 are not well utilizing the functions of RTP and in some cases even violates the RTP specifications. Thus we recommend that one focus on 1 and 2.

RTP protocol extensions to be used with conferencing are included because they are important in the context of centralized conferencing, where one RTP Mixer (Conference Focus) receives a participants media streams and distribute them to the other participants. These messages are defined in the Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF) [RFC4585] and the "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)" (CCM) [RFC5104] and are fully usable by the Secure variant of this profile (RTP/SAVPF) [RFC5124].

6.1.1. RTCP Feedback Message: Full Intra Request

The Full Intra Request is defined in Sections 3.5.1 and 4.3.1 of CCM [RFC5104]. It is used to have the mixer request from the currently distributed session participants a new Intra picture. This is used when switching between sources to ensure that the receivers can decode the video or other predicted media encoding with long prediction chains. It is RECOMMENDED that this feedback message is supported.

6.1.2. RTCP Feedback Message: Picture Loss Indicator

The Picture Loss Indicator is defined in Section 6.3.1 of AVPF [RFC4585]. It is used by a receiver to tell the encoder that it lost the decoder context and would like to have it repaired somehow. This is semantically different from the Full Intra Request above. It is
RECOMMENDED that this feedback message is supported as a loss
tolerance mechanism.

6.1.3. RTCP Feedback Message: Temporary Maximum Media Stream Bit Rate
Request

This feedback message is defined in Section 3.5.4 and 4.2.1 in CCM
[RFC5104]. This message and its notification message is used by a
media receiver, to inform the sending party that there is a current
limitation on the amount of bandwidth available to this receiver.
This can be for various reasons, and can for example be used by an
RTP mixer to limit the media sender being forwarded by the mixer
(without doing media transcoding) to fit the bottlenecks existing
towards the other session participants. It is RECOMMENDED that this
feedback message is supported.

6.2. RTP Header Extensions

The RTP specification [RFC3550] provides a capability to extend the
RTP header with in-band data, but the format and semantics of the
extensions are poorly specified. Accordingly, if header extensions
are to be used, it is REQUIRED that they be formatted and signalled
according to the general mechanism of RTP header extensions defined
in [RFC5285].

As noted in [RFC5285], the requirement from the RTP specification
that header extensions are "designed so that the header extension may
be ignored" [RFC3550] stands. To be specific, header extensions 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 rapid synchronisation header extension is recommended, as
discussed in Section 6.3.

Currently no other header extensions are recommended. But we do
include a list of the available ones for consideration below:

Transmission Time offsets: [RFC5450] defines a format for including
an RTP timestamp offset of the actual transmission time of the RTP
packet in relation to capture/display timestamp present in the RTP
header. This can be used to improve jitter determination and
buffer management.
Associating Time-Codes with RTP Streams: [RFC5484] defines how to associate SMPTE times codes with the RTP streams.

Audio Levels indications: There is ongoing work to define RTP header extensions for providing audio levels both from a media sender to an mixer [I-D.ietf-avtext-client-to-mixer-audio-level], and from a mixer to a receiver [I-D.ietf-avtext-mixer-to-client-audio-level].

6.3. Rapid Synchronisation Extensions

Many RTP sessions require synchronisation between audio, video, and other content. This synchronisation is performed by receivers, using information contained in RTCP SR packets, as described in the RTP specification [RFC3550]. This basic mechanism can be slow, however, so it is RECOMMENDED that the rapid RTP synchronisation extensions described in [RFC6051] be implemented. The rapid synchronisation extensions use the general RTP header extension mechanism [RFC5285], which requires signalling, but are otherwise backwards compatible.

7. Improving RTP Transport Robustness

There are some tools that can robustify RTP flows against Packet loss and reduce the impact on media quality. However they all add extra bits compared to a non-robustified stream. These extra bits need to be considered and the aggregate bit-rate needs to be rate controlled. Thus robustification might require a lower base encoding quality but has the potential to give that quality with fewer errors in it.

7.1. RTP Retransmission

Support for RTP retransmission as defined by "RTP Retransmission Payload Format" [RFC4588] is RECOMMENDED.

The retransmission scheme in RTP allows flexible application of retransmissions. Only selected missing packets can be requested by the receiver. It also allows for the sender to prioritize between missing packets based on senders knowledge about their content. Compared to TCP, RTP retransmission also allows one to give up on a packet that despite retransmission(s) still has not been received within a time window.

7.2. Forward Error Correction (FEC)

Support of some type of FEC to combat the effects of packet loss is beneficial, but is heavily application dependent. However, some FEC mechanisms are encumbered.
8. RTP Rate Control and Media Adaptation

It is REQUIRED to have an RTP Rate Control mechanism using Media adaptation to ensure that the generated RTP flows are network friendly, and maintain the user experience in the presence of network problems.

The biggest issue is that there are no standardized and ready to use mechanism that can simply be included in RTC-Web. Thus there will be need for the IETF to produce such a specification. A potential starting point for defining a solution is "RTP with TCP Friendly Rate Control" [rtp-tfrc].

9. RTP Performance Monitoring

RTCP does contains a basic set of RTP flow monitoring points like packet loss and jitter. There exist a number of extensions that could be included in the set to be supported. However, in most cases which RTP monitoring that is needed depends on the application, which makes it difficult to select which to include when the set of applications is very large.

10. IANA Considerations

This memo makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

11. Security Considerations

RTP and its various extensions each have their own security considerations. These should be taken into account when considering the security properties of the complete suite. We currently don’t think this suite creates any additional security issues or properties. The use of SRTP will provide protection or mitigation against all the fundamental issues by offering confidentiality, integrity and partial source authentication. We don’t discuss the key-management aspect of SRTP in this memo, that needs to be done taking the RTC-Web communication model into account.

In the context of RTC-Web the actual security properties required
from RTP are currently not fully understood. Until security goals and requirements are specified it will be difficult to determine what security features in addition to SRTP and a suitable key-management, if any, that are needed.

12. Acknowledgements

13. References

13.1. Normative References

[I-D.ietf-avtext-client-to-mixer-audio-level]
draft-ietf-avtext-client-to-mixer-audio-level-00 (work in progress), February 2011.

[I-D.ietf-avtext-mixer-to-client-audio-level]
draft-ietf-avtext-mixer-to-client-audio-level-00 (work in progress), February 2011.


13.2. Informative References


[rtp-tfrc]
Gharai, L., "RTP with TCP Friendly Rate Control (draft-gharai-avtcore-rtp-tfrc-00)", March 2011.

Authors’ Addresses

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
United Kingdom
Email: csp@csperkins.org

Magnus Westerlund
Ericsson
Farorgatan 6
SE-164 80 Kista
Sweden
Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Joerg Ott
Aalto University
School of Electrical Engineering
Espoo 02150
Finland
Email: jorg.ott@aalto.fi
