Audio Video Transport Working Group
Internet-Draft
Expires: August 28, 2006

M. Westerlund
Ericsson

February 24, 2006

How to Write an RTP Payload Format
draft-westerlund-avt-rtp-howto-00.txt

Status of this Memo

Copyright Notice

Abstract

This document contains information on how to best write an RTP payload format. Reading tips, design practices, and practical tips on how to quickly and with good results produce an RTP payload format specification. A template is also included with instructions that can be used when writing an RTP payload format.

Table of Contents

1. Introduction

RTP [RFC3550] payload formats define how a specific real-time data format is structured in the payload of an RTP packet. A real-time data format without a payload format specification can't be transported using RTP. This creates an interest from many individuals/organizations with media encoders or other types of real-time data to define RTP payload formats. The specification of a well designed RTP payload format is non-trivial and requires knowledge of both RTP and the real-time data format.

This document intends to help any author of an RTP payload format to make important design decisions, consider important features of RTP, security, etc. The document is also intended to be a good starting point for any person with little experience in IETF and/or RTP to learn the necessary steps.

This document extends and updates the information that are available in "Guidelines for Writers of RTP Payload Format Specifications" [RFC2736]. Since this RFC was written further experience has been gained on the design and specification of RTP payload format. Several new RTP profiles, and robustness tools has also been defined, which needs to be considered.

We also discuss the possible venues of defining an RTP payload format, in IETF, by other standard bodies and proprietary ones. Independent on the intended venue of specification, all will gain from this document.

1.1. Structure

This document has several different parts discussing different aspects of the creation of an RTP payload format specification. After the introduction and definitions there are a section discussing the preparations the author(s) should do before start writing. The following section discusses the different processes used when specifying and completing an payload format, with focus on working inside the IETF. Section 5 discusses the design of payload formats themselves in detail. Section 6 discusses the current design trends and provides good examples of practices that should be followed when applicable. Following that there is a discussion on important sections in the RTP payload format specification itself, like security and IANA considerations. This document ends with an appendix containing an template that can be used when writing RTP payload formats.

2. Terminology

2.1. Definitions

Media Stream A sequence of RTP packets that together provides all or parts of a media. It is scoped in RTP by the RTP session and a single sender source.

RTP Session: An association among a set of participants communicating with RTP. The distinguishing feature of an RTP session is that each maintains a full, separate space of SSRC identifiers. See also Section3.2.1.

RTP Payload Format: The RTP Payload format specifies how a specific media format is put into the RTP Payloads. Thus enabling the format to be used in RTP sessions.

2.2. Acronyms

ABNF Augmented Backus-Naur Form

ADU Application Data Unit

ALF Application Level Framing

ASM Any-Source Multicast

AVT: Audio Video Transport

BCP Best Current Practice

ID: Internet Draft

MTU Maximum Transmission Unit

WG: Working Group

QoS: Quality of Service

RFC: Request For Comment

RTP: Real-time Transport Protocol

RTCP: RTP Control Protocol

RTT: Round Trip Time

SSM Source Specific Multicast

3. Preparations

RTP is a complex real-time media delivery framework and it has a lot of details to consider when writing an RTP payload format. There is also important to have a good understanding of the media codec/format so that all its important features and properties are considered. First when one has sufficient understanding of both part can one produce an RTP payload format of high quality. On top of this, one needs to understand the process within IETF and especially the AVT WG to quickly go from initial idea to a finished RFC. This and the next section helps an author prepare himself in those regards.

3.1. Recommend Reading

In the below sub sections there are a number of documents listed. Not all needs to be read in full detail. However basically everything listed below does an author need to be aware of.

3.1.1. IETF Process and Publication

To understand the IETF process an draft author should start by reading RFC 2026 [RFC2026] that describes the standards process of IETF. In addition an author needs to understands the IETF rules and rights associated with copyright and IPR documented in BCP 78 [RFC3978] and BCP 79 [RFC3979]. In RFC 2418 [RFC2418] is the WG process, the relation between the IESG and the WG, and the responsibilities of WG chairs and participants described.

It is important to note that the RFC series contains documents of several different classifications; standards track, informational, experimental, best current practice (BCP), and historic. The standard tracks contains documents of three different maturity classifications, proposed, draft and Internet Standard. A standards track document must start as proposed, after proved interoperability of all the features it can be moved to draft standard, and final when further experience has been gathered it can be moved to Internet standard. As the content of the RFCs are not allowed to be changed, the only way of updating an RFC is to write and publish a new one that either updates or replaces the old one. Therefore it is important to both consider the Category field in the header and check if the RFC one is reading or going to reference is the latest and valid. One way of checking the current status of an RFC is to use the RFC-editor's RFC search engine, which displays the current status and which if any RFCs that updates or obsolete it.

Before starting to write an draft one should also read the Internet Draft writing guidelines (http://www.ietf.org/ietf/1id-guidelines.txt), the ID checklist

(http://www.ietf.org/ID-Checklist.html) and "Instructions to Request for Comments (RFC) Authors" [rfc2223bis].

There are also a number of documents to consider in process of writing of drafts intended to become RFCs. These are important when writing certain type of text.

RFC 2606: When writing examples using DNS names in Internet drafts, those name shall be using the example.com, example.net, and example.org domains.

RFC 3849: Defines the range of IPv6 unicast addresses (2001:DB8::/32) that should be used in any examples.

RFC 3330: Defines the range of IPv4 unicast addresses reserved for documentation and examples: 192.0.2.0/24.

RFC 4234: Augmented Backus-Naur Form (ABNF) is often used when writing text field specifications. Not that commonly used in RTP payload formats but may be useful when defining Media Type parameters of some complexity.

3.1.2. RTP

The recommended reading for RTP consist of several different parts; design guidelines, the RTP protocol, profiles, robustness tools, and media specific recommendations.

Any author of RTP payload formats should start with reading RFC 2736 [RFC2736] which contains an introduction to the application layer framing (ALF) principle, the channel characteristics of IP channels, and design guidelines for RTP payload formats. The goal of ALF is to be able to transmit Application Data Units (ADUs) that are independently usable by the receiver in individual RTP packets. Thus minimizing dependencies between RTP packets and the effects of packet loss.

Then it is suitable to learn more about the RTP protocol, by studying the RTP specification RFC 3550 [RFC3550] and the existing profiles. As a complement to the standards document there exist a book totally dedicated to RTP [CSP-RTP]. There exist several profiles for RTP today, but all are based on the "RTP Profile for Audio and Video Conferences with Minimal Control" (RFC 3551) [RFC3551] (abbreviated AVP). The other profiles that one should known about are Secure RTP (SAVP) [RFC3711], "Extended RTP Profile for RTCP-based Feedback" [rfc-avpf] and "Extended Secure RTP Profile for RTCP-based Feedback (RTP/SAVPF)" [rfc-savpf]. It is important to understand RTP and the AVP profile in detail. For the other profiles it is sufficient to

have an understanding on what functionality they provided and the limitations they create.

There has been developed a number of robustness tools for RTP. The tools are for different use cases and real-time requirements.

RFC 2198: The "RTP Payload for Redundant Audio Data" [RFC2198] provides functionalities to provided redundant copies of audio or text payloads. These redundant copies are sent together with an primary format in the same RTP payload. This format relies on the RTP timestamp to determine where data belongs in a sequence and therefore is usually primarily suitable to be used with audio. However also the RTP Payload format for T.140 [RFC4103] text format uses this format. The formats major property is that it only preserves the timestamp of the redundant payloads, not the original sequence number. Thus making it unusable for most video formats. This format is also only suitable for media formats that produce relatively small RTP payloads.

RFC 2733: The "An RTP Payload Format for Generic Forward Error Correction" provides an XOR based FEC over a number of RTP packets. These FEC packets are sent in a separate stream or as a redundant encoding using RFC 2198. This FEC scheme has certain restrictions in the number of packets it can protect. It is suitable for low to medium delay tolerable applications with limited amount of RTP packets.

RTP Retransmission: The RTP retransmission scheme [rtp-rtx] is used for semi-reliability of the most important RTP packets in a media stream. The scheme is not intended, nor suitable, to provide full reliability. It requires the application to be quite delay tolerable as a minimum of a round-trip time plus processing delay is required to perform an retransmission. Thus it is mostly suitable for streaming applications but may also be usable in certain cases when operating on networks with short RTT.

There also exist some management and monitoring extensions.

RFC 2959: The RTP protocol Management Information Database (MIB) [RFC2959] that is used with SNMP to configure and retrieve information about RTP sessions.

RFC 3611: The RTCP Extended Reports (RTCP XR) [RFC3611] consist of a framework for reports sent within RTCP. It can easily be extended by defining new report formats in future. The report formats that are defined are providing report information on; packet loss vectors, packet duplication, packet reception times, RTCP statistics summary and VoIP Quality. It also defines a mechanism

that allows receivers to calculate the RTT to other session participants when used.

RMONMIB: The remote monitoring work group has defined a mechanism [RFC3577] based on usage of the MIB that can be an alternative to RTCP XR.

There has also been developed a number of transport optimizations that are used in certain environments. They are all intended to be transparent and not need special consideration by the RTP payload format writer. Thus they are primarily listed here for informational reasons and do not require deeper studies.

RFC 2508: Compressing IP/UDP/RTP headers for slow serial links (CRTP) [RFC2508] is the first IETF developed RTP header compression mechanism. It provides quite good compression however it has clear performance problems when subject to packet loss between compressor and decompressor.

RFC 3095: Is the base specification of the robust header compression (ROHC) protocol [RFC3095]. This solution was created as a result of CRTP's lack of performance when subject to losses.

RFC 3545: Enhanced compressed RTP (E-CRTP) [RFC3545] was also developed to provide extensions to CRTP that allows for better performance over links with long RTTs, packet loss and/or reordering.

RFC 4170: Tunneling Multiplexed Compressed RTP (TCRTP) [RFC4170] is a solution that allows header compression within a tunnel carrying multiple multiplexed RTP flows. This is primarily used in voice trunking.

There exist a couple of different security mechanisms that may be used with RTP. All generic mechanisms need to be transparent for the RTP payload format and nothing that needs special consideration. The main reason that there exist different solutions is that different applications have different requirements thus different solutions have been developed. The main properties for a RTP security mechanism are to provide confidentiality for the RTP payload, integrity protection to detect manipulation of payload and headers, and source authentication. Not all mechanism provides all of these features which will need to be considered when used.

RTP Encryption: Section 9 of RFC 3550 describes a mechanism to provide confidentiality of the RTP and RTCP packets, using per default DES encryption. It may use other encryption algorithms if both end-points agree on it. This mechanism is not recommend due to its weak security properties of the used encryption algorithms. It also lacks integrity and source authentication mechanisms.

SRTP: The profile for Secure RTP (SAVP) [RFC3711] and the derived profile (SAVPF [rfc-savpf]) is a solution that provides confidentiality, integrity protection and partial source authentication.

IPsec: IPsec may also be used to protect RTP and RTCP packet.

TLS: TLS may also be used to provide transport security between two end-point of the TLS connection for a flow of RTP packets that are framed over TCP.

3.2. Important RTP details

This section does not remove the necessity of reading up on RTP. However it does point out a couple of important details to remember when designing the payload format.

3.2.1. The RTP Session

The definition of the RTP session from RFC 3550 is:

"An association among a set of participants communicating with RTP. A participant may be involved in multiple RTP sessions at the same time. In a multimedia session, each medium is typically carried in a separate RTP session with its own RTCP packets unless the encoding itself multiplexes multiple media into a single data stream. A participant distinguishes multiple RTP sessions by reception of different sessions using different pairs of destination transport addresses, where a pair of transport addresses comprises one network address plus a pair of ports for RTP and RTCP. All participants in an RTP session may share a common destination transport address pair, as in the case of IP multicast, or the pairs may be different for each participant, as in the case of individual unicast network addresses and port pairs. In the unicast case, a participant may receive from all other participants in the session using the same pair of ports, or may use a distinct pair of ports for each.

The distinguishing feature of an RTP session is that each maintains a full, separate space of SSRC identifiers (defined next). The set of participants included in one RTP session consists of those that can receive an SSRC identifier transmitted by any one of the participants

either in RTP as the SSRC or a CSRC (also defined below) or in RTCP. For example, consider a three-party conference implemented using unicast UDP with each participant receiving from the other two on separate port pairs. If each participant sends RTCP feedback about data received from one other participant only back to that participant, then the conference is composed of three separate point-to-point RTP sessions. If each participant provides RTCP feedback about its reception of one other participant to both of the other participants, then the conference is composed of one multi-party RTP session. The latter case simulates the behavior that would occur with IP multicast communication among the three participants.

The RTP framework allows the variations defined here, but a particular control protocol or application design will usually impose constraints on these variations."

3.2.2. RTP Header

The RTP header contains two fields that require additional specification by the RTP payload format, namely the RTP Timestamp and the marker bit. Certain RTP payload formats also uses the RTP sequence number to realize certain functionalities. The payload type is used to indicate the used payload format.

Marker bit: A single bit normally used to provide important indications. In audio it is normally used to indicate the start of an talk burst. This to enable jitter buffer adaptation prior to this with minimal audio quality impact. In video the marker bit is normally used to indicate the last packet part of an frame. This enables an decoder to finish decoding the picture, where it otherwise may need to wait for the next packet to explicitly know that.

Timestamp: The RTP timestamp indicate the time instance the media belongs to. For discrete media, like video it normally indicates when the media (frame) was sampled. For continuous media it normally indicates the first time instance the media present in the payload represents. For audio this is the sampling time of the first sample. All RTP payload formats must specify the meaning of the timestamp value and which clock rates that are allowed. Note that clock rates below 1000 Hz is not appropriate due to RTCP measurements function that in that case loose resolution.

Sequence number: The sequence number are monotonically increasing and set as packets are sent. That property is used in many payload formats to recover the order of everything from the whole stream down to fragments of ADUs and the order they shall be decoded.

Payload Type: Commonly the same payload type is used for a media stream for the whole duration of a session. However in some cases it may be required to change the payload format or its configuration during the session. The payload type is used to indicate on a per packet basis which format is used. Thus certain major configuration information can be bound to a payload type value by out-of-band signalling. Examples of this would be video decoder configuration information.

SSRC: The Sender Source ID is normally not used by a payload format other than identifying the RTP timestamp and sequence number space a packet belongs to, allowing the simultaneously reception of multiple senders. However there are certain of the RTP robustification mechanisms that are RTP payloads that have used multiple SSRCs and bound them together to correctly separate original data and repair or robustification data.

The remaining fields are commonly not influencing the RTP payload format. The padding bit is worth clarifying as it indicates that one or more bytes are appended after the RTP payload. This padding must be removed by a receiver before payload format processing can occur. Thus it is completely separate from any padding that may occur within the payload format itself.

3.2.3. RTP Multiplexing

RTP has three multiplexing points that are used for different purposes. A proper understanding of this is important to correctly utilized them.

The first one is separation of media streams of different types, which is accomplished using different RTP sessions. So for example in the common multi-media session with audio and video, RTP multiplex audio and video on different RTP sessions. To achieve this separation, transport level functionalities are use, normally UDP port numbers. Different RTP sessions are also used to realize layered scalability as it allows a receiver to select one or more layers for multicasted RTP sessions simply by joining the multicast groups the desired layers are transported over. This also allows different Quality of Service (QoS) be applied to different media.

The next point is separation of different sources within a RTP session. Here RTP uses the SSRC (Sender Source) which identifies individual sources. An example of individual sources in audio RTP session, would be different microphones, independent of if they are from the same host or different hosts. For each SSRC a unique RTP sequence number and timestamp space is used.

The third multiplexing point is the RTP headers payload type field. The payload type identifies what format the content in the RTP payload has. This includes different payload format configurations, different codecs, and also usage of robustness mechanisms like the one described in RFC 2198 [RFC2198].

3.2.4. RTP Synchronization

There are several types of synchronization and we will here describe how RTP handles the different types:

Intra media: The synchronization within a media stream from a source is accomplished using the RTP timestamp field. Each RTP packet carry the RTP timestamp that specifies the media contained in this packets position in relation to other media on the time line. This is especially useful in cases of discontinues transmissions. Discontinues can also be caused by the network and with extensive losses the RTP timestamp tells the receiver how much later than previously received media the media shall be played out.

Inter media: As applications commonly has a desire to use several media types at the same time there exist a need to synchronize also the different medias from the same source. This puts two requirements on RTP; possibility to determine which media is from the same source and if they should be synchronized with each other; and the functionality to facilitate the synchronization itself.

The first part of Inter media synchronization is to determine which SSRCs in each session that should be synchronized with each other. This is accomplished by comparing the RTCP SDES CNAME field. SSRCs with the same CNAME in different RTP session should be synchronized.

The actual RTCP mechanism for inter media synchronization is based on that each media stream provide a position on the media specific time line (measured in RTP timestamp ticks) and a common reference time line. The common reference time line is in RTCP expressed as an wall clock time in the Network Time Protocol (NTP) format. It is important to notice that the wall clock time is not required to be synchronized between hosts, for example by using NTP [RFC1305] . It can even have nothing at all to do with the actual time, for example the host system's uptime can be used for this purpose. The important factor is that all media streams from a particular source that are being synchronized uses the same reference clock to derive there relative RTP timestamp time scales.

In the below Figure (Figure 1) it is depicted how if one receives RTCP Sender Report (SR) packet P1 in one media stream and RTCP SR

packet P2 in the other session, then one can calculate the corresponding RTP timestamp values for any arbitrary point in time T. However to be able to do that it is also required to know the RTP timestamp rates for each media currently used in the sessions.

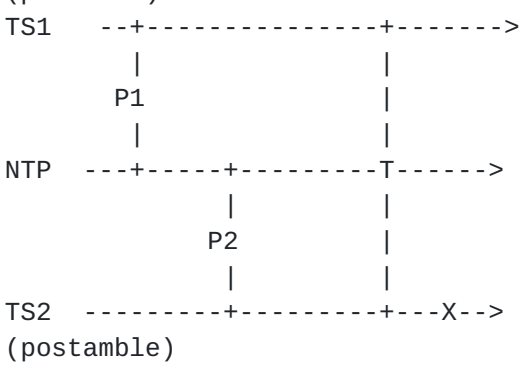

Figure 1: RTCP Synchronization

Lets assume that media one uses a RTP Timestamp clock rate of 16000 Hz, and media 2 a rate of 90 kHz. Then the TS1 and TS2 for point T can be calculated in the following way: TS1(T) = TS1(P1) + 16000 * (NTP(T)-NTP(P1)) and TS2(T) = TS2(P2) + 90000 * (NTP(T)-NTP(P2)). This calculation is useful as it allows to generate a common synchronization point for which all time values are provided (TS1(T), TS2(T) and T). So when one like to calculate at which NTP time the TS present in packet X corresponds to one can do that in the following way: NTP(X) = NTP(T) + (TS2(X) - TS2(T))/90000.

3.3. Signalling Aspects

RTP payload formats are used in the context of application signalling protocols such as SIP [RFC3261] using SDP [sdp-new] with Offer/Answer [RFC3264], RTSP [RFC2326] or SAP [RFC2326]. These examples all uses SDP to indicate which and how many media streams that are desired to be used in the session and their configuration. To be able to declare or negotiate which media format and RTP payload packetization the payload format must be given an identifier. In addition to the identifier many payload formats also have the need to carry further configuration information out-of-band in regards to the RTP payloads prior to the media transport session.

The above examples of session establishing protocols all use SDP, however also other session description formats may be used. For example there have been discussion on a new Session Description format within IETF (SDP-NG). To prevent locking the usage of RTP to SDP based out-of-band signalling, the payload formats are identified using an separate definition format for the identifier and

parameters. That format is the Media Type.

3.3.1. Media Types

Media types [RFC4288] was originally created for identifying media formats included in email. Media types are today also used in HTTP, MSRP and many other protocols to identify arbitrary content carried within the protocols. Media types also provide a media hierarchy that fits RTP payload formats well. Media type names are two-part and consist of content type and sub-type separated with a slash, e.g. "audio/PCMA" or "video/h263-2000". It is important to choose the correct content-type when creating the media type identifying an RTP payload format. However in most cases there is little doubt what content type the format belongs to. Guidelines for choosing the correct media type and registration rules are present in RFC 4288 [RFC4288]. The additional rules for media types for RTP payload formats are present in RFC XXXX. [rfc3555bis]

Media types are allowed any number of parameters which are divided into two groups, required and optional parameters. They are always on the form name=value. There exist no restriction on how the value is defined from media types perspective, except that parameters must have value. However the carrying of media types in SDP etc. has resulted in the following restrictions that needs to be followed to make media types for RTP payload format usable:

1. Arbitrary binary content in the parameters are allowed but needs to be encoded so that they can be placed within text based protocols. Base64 [RFC3548] is recommended, but for shorter content BASE16 may be more appropriate as it is simpler to interpret by humans. This needs to be explicitly stated when defining a media type parameter with binary content.

2. The end of the value needs to be easily found when parsing a message. Thus parameter values that are continuous and non interrupted by common text separators, such as space and semi-colon are recommended. If that is not possible some type of escaping should be used. Usage of <"> is recommended.

3. A common representation form of the media type and its parameters is on a single line. In those cases the media type is followed by a semi-colon separated list of the parameter value pair, e.g. audio/amr octet-align=0; mode-set=0,2,5,7; mode-change-period=2.

3.3.2. Mapping to SDP

As SDP [sdp-new] is so commonly used as an out-of-band signalling channel, a mapping of the media type exist. The details on how to

map the media type and its parameters into SDP are described in RFC YYYY [rfc3555bis]. However this is not sufficient to explain how certain parameter shall be interpreted for example in the context of Offer/Answer negotiation [RFC3264].

3.3.2.1. The Offer/Answer Model

The Offer/Answer (O/A) model allows SIP to negotiate media formats and which payload formats and their configuration is used in a session. However O/A does not define a default behavior and instead points out the need to define how parameters behave. To make things even more complex the direction of media within a session do have impact on these rules, thus some cases may require description separately for peers that are send only, receiver only or both senders and receivers as identified by the SDP attributes a=sendonly, a=recvonly, and a=sendrecv. In addition any usage of multicast puts a further limitations as the same media stream is delivered to all participants. If those restrictions are to limiting also to be used in unicast then separate rules for unicast and multicast will be required.

The most common O/A interpretation and the simplest is for declarative parameters, i.e. the sending entity can declare a value and that has no direct impact on the other agents values. This declared value applies to all media that are going to be sent to the declaring entity. For example most video codecs has level parameter which tells the other participants the highest complexity the video decoder supports. The level parameter can be declared independently by two participants in a unicast session as it will be the media sender responsibility to transmit a video stream that fulfills the limitation the other has declared. However in multicast it will be necessary to send a stream that follows the limitation of the weakest receiver, i.e. the one that has supports the lowest level. To simplify the negotiation in these cases it is common to require any answerer to a multicast session to take a yes or no approach to parameters.

"Negotiated" parameters are another type of parameters, for which both sides needs to agree on their values. Such parameter requires that the answerer either accept as they are offered or remove the payload type the parameter belonged to. The removal of the payload type from the answer indicates to the offerer the lack of support. An unfortunate implications of the need to use complete payload types to indicate each configuration possible to achieve interoperability, is that the number of payload types necessary can quickly grow big. This is one reason to keep the total number of set of capabilities that may be implemented limited.

The most problematic type of parameters are those that relates with the transmission the entity performs. They do not really fit the O/A model but can be shoe-horned in. Example of such parameters can be found in the H.264 video code's payload format [RFC3984], where the name of all parameters with this property starts sprop-. The issue that exist is that they declare properties for a media stream one don't yet know if the other party accept. The best one can make of the situation is to explain the assumption that the other party will accept the same reception parameter as the offerer of the session. If the answerer needs to change any declarative parameter then the offerer may be required to make an new offer to update the parameter values for its outgoing media stream.

Another issue to consider is the sendonly media streams in offers. For all parameters that relates to what one accepts to receive those don't have any meaning other than provide a template for the answering entity. It is worth pointing out in the specification that these provides recommended set of parameter values by the sender. Note that sendonly streams in answers will need to indicate the offerers parameters to ensure that the offerer can match the answer to the offer.

A further issue with offer/answer which complicates things is that it is allowed to renumber the payload types between offer and answer. This is not recommended but allowed for support of gateways to the ITU conferencing suit. Which means that answers for payload types needs to be possible to bind to the ones in the offer even when the payload type number has been changed, and some of the proposed payload types have been removed. This must normally be done based on configurations offered, thus negotiated parameters becomes vital.

3.3.2.2. Declarative usage in RTSP and SAP

SAP (Session Announcement Protocol) [RFC2974] is used for announcing multicast sessions. Independently of the usage of Source Specific Multicast (SSM) [RFC3569] or Any-Source Multicast (ASM), the SDP provided by SAP applies to all participants. All media that is sent to the session must follow the media stream definition as specified by the SDP. Thus enabling everyone to receive the session if they support the configuration. Here SDP provides a one way channel with no possibility to affect the configuration defined by SDP that the session creator has decided upon. Any RTP Payload format that requires parameters for the send direction and which needs individual values per implementation or instance will fail in a SAP session for a multicast session allowing anyone to send.

Real-Time Streaming Protocol (RTSP) [RFC2326] allows the negotiation of transport parameters for media streams part of a streaming session

between a server and client. RTSP has divided the transport parameters from the media configuration. SDP is commonly used for media configuration in RTSP and is sent to the client prior to session establishment, either through the usage of the DESCRIBE method or an out-of-band channel like HTTP, email etc. The SDP is used to determine which media streams and what formats are being used before the session establishment.

Thus both SAP and RTSP uses SDP to configure receivers and senders with a predetermined configuration including the payload format and any of its parameters of a media stream. Thus all parameters are used in a declarative fashion. This can result in different treatment of parameters between offer/answer and declarative usage in RTSP and SAP. This will then need to be pointed out by the payload format specification.

3.4. Transport Characteristics

The general channel characteristics that RTP flows are experiencing are documented in Section 3 of RFC2736 [RFC2736]. Below additional information is discussed.

3.4.1. Path MTU

At the time of writing the most common IP Maximum Transmission Unit (MTU) of used link layers is 1500 bytes (Ethernet data payload). However there exist links with both smaller MTU and much larger MTUs. Certain parts of Internet do already today support IP MTU of 9000 bytes or more. There is an slow ongoing evolution towards larger MTU sizes. This should be considered in the design, especially in regards to features such as aggregation of independently decodable data units.

4. Specification Process

This section discusses the recommended process to produce an RTP payload format in the described venues. This is to document the best current practice on how to get a well designed and specified payload format as quickly as possible. For specifications that are proprietary or defined by other standards bodies than IETF the primary milestone is registration of the RTP payload format name. However there is also the issue of ensuring best possible quality of any specification.

4.1. IETF

Specification in IETF is recommended for all standardized media formats. The main reason is to provide an openly available RTP payload format specification that also has been reviewed by people experienced with RTP Payload formats. This also assumes that the AVT WG exist.

4.1.1. Steps from Idea to Publication

There are a number of steps that an RTP payload format should go through from the initial idea until it is published. This also documents the process that the AVT working group applies when working with RTP payload formats.

1. Idea: Determined the need for an RTP payload format as an IETF specification.

2. Initial effort: Using this document as guideline one should be able to get started on the work. If one's media codec doesn't fit any of the common design patterns or one has problems understanding what the most suitable way forward is, then one should contact the AVT working group and/or the WG chairs. The goal of this stage is to have an initial individual draft. This draft needs to focus on the introduction parts that describe the real-time media format and the basic idea on how to packetize it. All the details are not required to be filled in. However security chapter is not something that one should skip even initially. It is important to consider already from the start any serious security risks that needs to be solved. This step is completed when one has a draft that is sufficient detailed for a first review by the WG. The less confident one is of the solution, the less work should be spent on details, instead concentrate on the codec properties and what is required to make it work.

3. Submission of first version. When one has performed the above one submits the draft as an individual draft. This can be done at any time except the 3 weeks (current deadline at the time of writing, consult current announcements) just before an IETF meeting. When the IETF draft announcement has been sent out on the draft announcement list, forward it to the AVT WG and request that it is reviewed. In the email outline any issues the authors currently have with the design.

4. Iterative improvements: Taking the feedback into account one updates the draft and try resolve any issues. New revision of the draft can be submitted at any time. It is recommended to do it whenever one has made major updates or have new issues that are easiest to discuss in the context of a new draft version.

5. Becoming WG document: Due to that the definition of RTP payload formats are part of the AVT's charter, RTP payload formats that are going to be published as standards track RFCs needs to become WG documents. Becoming WG document means that the chairs are responsible for administrative handling, like publication requests. However be aware that making a document into a WG document changes the formal ownership and responsibility from the individual authors to the WG. The initial authors will continue being document editor, unless unusual circumstances occur. The AVT WG accepts new RTP payload formats based on their suitability and document maturity. The document maturity is a requirement to ensure that there are dedicated document editors and that there exist a good solution.

6. Iterative improvements: The updates and review cycles continues until the draft the has reached the maturity suitable for publication.

7. WG last call: WG last call of at least 2 weeks are always performed for AVT WG documents. The authors request WG last call for a draft when they think it i mature enough for publication. The chairs perform a review to check if they agree with the authors assessment. If the chairs agree on the maturity, the WG last call is announced on the WG mailing list. If there are issues raised these needs to be addressed with an updated draft version. For any more substantial updates of draft, a new WG last call is announced for the updated version. Minor changes, like editorial on can be progressed without an additional WG last call.

8. Publication Requested: For WG documents the chairs request publication of the draft. After this the approval and publication process described in RFC 2026 [RFC2026] are

performed. The status after the publication has been requested can be tracked using the IETF data tracker. Documents do not expire as normal after publication has been requested. In addition any submission of document updates requires the approval of WG chair(s). The authors are commonly asked to address comments or issues raised by the IESG. The authors also review the document prior to publication as an RFC to ensure its correctness.

4.1.2. WG meetings

WG meetings are for discussing issues, not presentations. This means that most RTP payload format should never need to be discussed in a WG meeting. RTP payload formats that would be discussed are either controversial issues that failed to be resolved on the mailing list, or includes new design concepts worth a general discussion.

There exist no requirement to present or discuss a draft at a WG meeting before it becoming published as an RFC. Thus even authors that lack the possibility to go to WG meetings should be able to successfully specify an RTP payload format in IETF. WG meetings may only become required if the draft get stuck in a serious debate that isn't easily resolved.

4.1.3. Draft Naming

To simplify the work of the AVT WG chairs and its WG members a specific draft file naming convention shall be used for RTP payload formats. Individual submissions shall be named draft-<lead author family name>-avt-rtp-<descriptive name>-<version>. The WG documents shall be named according to this template: draft-ietf-avt-rtp-<descriptive name>-<version>. The inclusion of "avt" in the draft filename ensures that the search for "avt-" will find all AVT related drafts. Inclusion of "rtp" tells us that it is an RTP payload format draft. The descriptive name should be as short as possible while still describe what the payload format is for. It is recommended to use the media format or codec acronym. Please note that the version must start at 00 and is increased by one for each submission to the IETF secretary of the draft. No version numbers may be skipped.

4.1.4. How to speed up the process

There a number of ways of losing a lot of time in the above process. This section discuss what to do and what to avoid.

- Do not only update the draft to the meeting deadline. An update to each meeting automatically limits the draft to 3 updates per

year. Instead ignore the meeting schedule and publish new versions as soon as possible.

- Try to avoid requesting review when people are busy, like the weeks before a meeting. Review should be asked at all possible times and it is actually more likely that people has more time for them directly after a meeting.

- Perform draft updates quickly. A common mistake is that the authors lets the draft slip. By performing updates to the draft text directly after getting resolution on an issue, speeds things up. This as it minimizes the delay that the author has direct control over. Waiting for reviews, responses from area directors and chairs, etc can be much harder to speed up.

- Failing to take the human nature into account. It happens that people forget or needs to be reminded about tasks. Send people you are waiting for a kindly reminder if things takes longer than expected. To avoid annoying people ask for a time estimate from people when they expect to full fill the requested task.

- Not enough review. It is common that documents take a long time and many iterations because not enough review is performed in each iteration. To improve the amount of review you get on your own document, trade review time with other document authors. Make a deal with some other document authors that you will review his draft(s) if he reviews yours. Even inexperience reviewers can help with language, editorial or clarity issues. Try also approaching the more experienced people in the WG and get them to commit to a review. The WG chairs cannot, even if desirable, be expected to review all versions. Due to workload the chairs may need to concentrate on key points in a draft evolution, like initial submissions, if ready to become WG document and WG last call.

4.2. Other Standards bodies

Other standard bodies may define RTP payload in their own specifications. When they do this they are strongly recommend to contact the AVT WG chairs and request review of the work. It is recommended that at least two review steps are performed. One early in the process when more fundamental issues easily can be resolved without abandoning a lot of effort. Then when nearing completion, but while still possible to update the specification as second review should be scheduled. In that pass the quality can be assessed and hopefully no updates are needed. Using this procedure can avoids both conflicting definitions and serious mistakes, like breaking certain aspects of the RTP model.

RTP payload Media Types may be registered in the standards tree by other standard bodies. The requirements on the organization are outlined in the media types registration document (RFC 3555 [RFC3555] and RFC 4288 [RFC4288]). This registration requires a request to the IESG, which ensures that the registration template is acceptable. To avoid last minute problems with these registration the registration template should be sent for review both to the AVT WG and the media types list (ietf-types@iana.org) and is something that should be included in the IETF reviews of the payload format specification.

Registration of the RTP payload name is something that is required to avoid name collision in the future. Do also note that "x-" names are not suitable for any documented format as they have the same problem with name collision and can't be registered. The list of already registered media types can be found at IANA (http://www.iana.org).

4.3. Propreitary and Vendor Specific

Proprietary RTP payload formats are commonly specified when the real-time media format is proprietary and not intended to be part of any standardized system. However there exist many reasons why also proprietary formats should be correctly documented and registered;

- Usage in standardized signalling environment such as SIP/SDP. RTP needs to be configured regarding used RTP profiles, payload formats and their payload types. To accomplish this there is an need for registered names to ensure that the names do not collide with other formats.

- Sharing with business partners. As RTP payload formats are used for communication, situations where business partners like to support one proprietary format often arises. Having a well written specification of the format will save time and money for both one selves and ones partner, as interoperability will much easier to accomplish.

- To ensure interoperability between different implementations on different platforms.

To avoid name collisions there is a central register keeping tracks of the registered Media Type names used by different RTP payload formats. When it comes to proprietary formats they should be registered in the vendors own tree. All vendor specific registrations uses names that start with "vnd.vendor-name". All names that uses names in the vendors own trees are not required to be registered with IANA. However registration is recommended if used at all in public environments.

5. Designing Payload Formats

The best summary of payload format design is KISS (Keep It Simple, Stupid). A simple payload format makes it easy to review for correctness, implement, and have low complexity. Unfortunately contradicting requirements sometime makes it hard to do things simple. Complexity issues and problems that occur for RTP payload formats are:

To many configurations: Contradicting requirements results in that one configuration for each conceivable case is created. Such contradicting requirements are often between functionality and bandwidth. This has two big negatives. First all configurations needs to be implemented. Secondly the using application must select the most suitable configuration. Selecting the best configuration can be very difficult and in negotiating applications, this can create interoperability problems. The recommendation is to try to select a very limited (preferable one) configuration that preforms the most common case well and is capable of handling the other cases, but maybe less well.

Hard to implement: Certain payload formats may become difficult to implement both correctly and efficient. This needs to be considered in the design.

Interaction with general mechanisms: Special solutions may create issues with deployed tools for RTP, like robustification tools. For example the requirement of non broken sequence space creates issues with using both payload type switching and interleaving any robustification mechanism within the stream.

5.1. Features of RTP payload formats

There are number of common features in RTP payload formats. There are no general requirement to support these features, instead their applicability must be considered for each payload format. It might in fact be that certain features are not even applicable.

5.1.1. Aggreagation

Aggregation allows for the inclusion of multiple ADUs within the same RTP payload. This is commonly supported for codec that produce ADUs of sizes smaller than the IP MTU. Do remember that the MTU may be significantly larger than 1500 bytes, 9000 bytes is available today and a MTU of 64k may be available in the future. Many speech codecs have the property of ADUs of a few fixed sizes. Video encoders generally may produce ADUs of quite flexible size. Thus the need for aggregation may be less. However in certain use cases the

possibility to aggregate multiple ADUs especially for different playback times are useful.

The main disadvantage of aggregation is the extra delay introduced, due to buffering until sufficient amount of ADUs have been collected. It also introduces buffering requirements on the receiver.

5.1.2. Fragmentation

If the real-time media format has the property that it may produce ADUs that are larger than common MTUs sizes then fragmentation support should be considered. An RTP Payload format may always fallback on IP fragmentation, however as discussed in RFC 2736 this have some drawbacks. The usage of RTP payload format level fragmentation, does primarily allow for more efficient usage of RTP packet loss recovery mechanisms.

5.1.3. Interleaving and Transmission Re-Scheduling

Interleaving has been implemented in a number of payload formats to allow for less quality reduction when packet loss occurs and data is aggregated. A loss of an RTP packet with several ADUs in the payload has the same affect as a burst loss if the ADUs would have been transmitted in individual packets. To reduce the burstiness of the loss, the data present in an aggregated payload may be interleaved, thus spread the loss over a longer time period.

A requirement for doing interleaving within an RTP payload format is the aggregation of multiple ADUs. For formats that don't use aggregation there is still the possibility to implement an transmission order re-scheduling mechanism. That have the effect that packets transmitted next to each other originates from different points in the media stream. This can be used to mitigate burst losses, which may be useful if one transmit packets with small intervals. However it may also be used to transmit more significant data earlier in combination with RTP retransmission to allow for more graceful degradation and increased possibilities to receive the most important data, e.g. Intra frames of video.

The drawbacks of interleaving is the significantly increased transmission buffering delay, making it mostly useless for low delay applications. It also creates significant buffering requirements on the receiver. That buffering also is problematic as it is usually difficult to indicate when a receiver may start consume data and still avoid buffer underrun caused by the interleaving mechanism itself. The transmission re-scheduling is only useful in a few specific cases, like in streaming with retransmissions. This must be weighted against the complexity of these schemes.

5.1.4. Media Back Channels

A few RTP payload format have implemented back channels within the media format. Those have been for specific features, like the AMR [RFC3267] codec mode request (CMR) field. The CMR field is used in gateway operations to circuit switched voice to allow an IP terminal to react to the CS networks need for a specific encoder mode. A common property for the media back channels is the need to have this signalling in direct relation to the media or the media path.

If back channels are considered for an RTP payload format they should be for specific mechanism and which can't be easily satisfied by more generic mechanisms within RTP or RTCP.

6. Current Trends in Payload Format Design

This section provides a few examples of payload formats that is worth noting for good design in general or specific details.

6.1. Audio Payloads

To be written

6.2. Video

To be written

6.3. Text

To be written

7. Important Specification Sections

There a number of sections in the payload format draft that needs some special considerations. These include security and IANA considerations.

7.1. Security Consideration

All Internet drafts requires a Security Consideration section. The security consideration section in an RTP payload format needs to concentrate on the security properties this particular format has. Some payload format has very little specific issues or properties and can fully fall back on the general RTP and used profile's security considerations. Due to that these are always applicable a reference to these are normally placed first in the security consideration section.

The security issues of confidentiality, integrity protection and source authentication are common issues for all payload formats. These should be solved by payload external mechanism and does not need any special consideration in the payload format except for an reminder on these issues. A suitable stock text to inform people about this is included in the template.

Potential security issues with an RTP payload format and the media encoding that needs to be considered are:

1. That the decoding of the payload format or its media shows substantial non-uniformity, either in output or in complexity to perform the decoding operation. For example a generic non-destructive compression algorithm may provide an output of almost infinite size for a very limited input. Thus consuming memory or storage space out of proportion with what the receiving application expected causing some sort of disruption, i.e. a denial of service attack on the receiver by preventing that host to produce any good put. Certain decoding operations may also have variable consumption of amount of processing needed to perform such operations dependent on the input. This may also be a security risk if that processing load is possible to raise significantly from nominal simply by designing a malicious input sequence. If such potential exist this must be expressed in the security consideration section to make implementers aware of the need to take precautions against such behavior.

2. The inclusion of active content in the media format or its transport. With active content means scripts etc that allows an attacker to perform potentially arbitrary operations on the receiver. Most active content have limited possibility to access

the system or perform operations outside a protected sandbox. However if active content may be included the potential must be noted. It is also strongly recommend that references to any security model applicable for such content is referenced.

3. Some media formats allows for the carrying of "user data", or types of data which is not known at the time of the specification of the payload format. Such data may be a security risk and should be mentioned.

Suitable stock text for the security consideration is provided in the template. However the authors do need to actively consider any security issues from the start. Failure to address these issues is blocking approval and publication.

7.2. Congestion Control

RTP and its profiles do discuss congestion control. Congestion control is an important issue in any usage in non-dedicated networks. For that reason all RTP payload formats are recommended to discuss the possibilities that exist to regulate the bit-rate of the transmissions using the described RTP payload format. Some formats may have limited or step wise regulation of bit-rate. Such limiting factor should be discussed.

7.3. IANA Consideration

Due to that all RTP Payload format contains a Media Type specification they also need an IANA consideration section. The media type name must be registered and this is done by requesting that IANA register that media name. When that registration request is written it shall also be requested that the media type is included under the "RTP Payload Format MIME types" list part of the RTP registry.

In addition to the above request for media type registration some payload formats may have parameters where in the future new parameter values needs to be added. In these cases a registry for that parameter must be created. This is done by defining the registry in the IANA consideration section. BCP 26 (RFC 2434) [RFC2434] provides guidelines to writing such registries. Care should be taken when defining the policy for new registrations.

Before writing a new registry it is worth checking the existing ones in the IANA "MIME Media Type Sub-Parameter Registries". For example video formats needing a media parameter expressing color sub-sampling may be able to reuse those defined for video/raw [RFC4175].

8. Authoring Tools

This section informs and recommends some tools that may be used. Don't be pressured to follow these recommendation. There exist a number of alternatives. But these suggestion is worth checking out before deciding that the field is greener somewhere else.

8.1. Editing Tools

There is many choices when it comes to tools to choose for authoring Internet drafts. However in the end they needs to be able to produce a draft that conforms to the Internet drafts requirements. If you don't have any previous experience with authoring Internet drafts XML2RFC do have some advantages. It helps creating a lot of the necessary boiler plate in accordance with the latest rules. Thus reducing the effort. It also speeds up the publication after approval as the RFC-editor can use the source XML document to quicker produce the RFC.

Another common choice is to use Microsoft Word and a suitable template, see [RFC3285] to produce the draft and print that using the generic text printer. It has some advantage when it comes to spell checking and change bars. However Word may also produce some problems, like changing formating, inconsistent result between what one sees in the editor and in the generated text document, at least according to the authors personal experience.

8.2. Verification Tools

There are few tools that are very good to know about when writing an draft. These help check and verify parts of ones work. These tools can be found at http://tools.ietf.org.

- ID Nits checker. It checks that the boiler plate and some other things that are easily verifiable by machine is okay in your draft. Always use it before submitting a draft to avoid direct refusal in the submission step.

- ABNF Parser and verification. Used to check that your ABNF parses correctly and warns about loose ends, like undefined symbols. However the actual content can only be verified by humans knowing what it intends to describe.

- RFC diff. A diff tool that is optimized for drafts and RFC. For example it doesn't point out that the foot and header has moved in relation to the text on every page.

9. Open Issues

This document currently has a few open issues that needs resolving before publication:

- Should any procedure for the future when the AVT WG is closed be described?
- The section of current examples of good work needs to be filled in.
-

10. IANA Considerations

This document makes no request of IANA.

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

11. Security Considerations

As this is an informational document on the writing of drafts intended to be RFCs there is no direct security considerations. However the document does discuss the writing of security consideration sections and what should be particular considered for RTP payload formats.

12. RFC Editor Consideration

Note to RFC Editor: This section may be removed after carrying out all the instructions of this section.

Please replace all References to RFC XXXX with the RFC number that the update of RFC 3555 [rfc3555bis] receives.

13. Acknowledgements

14. Informative References

[CSP-RTP] Colin , "RTP: Audio and Video for the Internet", June 2003.

[MACOSFILETYPES] Apple Knowledge Base Article 55381<http://www.info.apple.com/kbnum/n55381>, "Mac OS: File Type and Creator Codes, and File Formats", 1993.

[RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

[RFC2327] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[RFC2418] Bradner, S., "IETF Working Group Guidelines and Procedures", BCP 25, RFC 2418, September 1998.

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

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[RFC2736] Handley, M. and C. Perkins, "Guidelines for Writers of RTP Payload Format Specifications", BCP 36, RFC 2736, December 1999.

[RFC2959] Baugher, M., Strahm, B., and I. Suconick, "Real-Time Transport Protocol Management Information Base", RFC 2959, October 2000.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[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.

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

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

[RFC3267] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 3267, June 2002.

[RFC3285] Gahrns, M. and T. Hain, "Using Microsoft Word to create Internet Drafts and RFCs", RFC 3285, May 2002.

[RFC3545] Koren, T., Casner, S., Geevarghese, J., Thompson, B., and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, July 2003.

[RFC3548] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.

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

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

[RFC3555] Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Formats", RFC 3555, July 2003.

[RFC3569] Bhattacharyya, S., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, July 2003.

[RFC3577] Waldbusser, S., Cole, R., Kalbfleisch, C., and D. Romascanu, "Introduction to the Remote Monitoring (RMON) Family of MIB Modules", RFC 3577, August 2003.

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3978] Bradner, S., "IETF Rights in Contributions", BCP 78, RFC 3978, March 2005.

[RFC3979] Bradner, S., "Intellectual Property Rights in IETF Technology", BCP 79, RFC 3979, March 2005.

[RFC3984] Wenger, S., Hannuksela, M., Stockhammer, T., Westerlund, M., and D. Singer, "RTP Payload Format for H.264 Video", RFC 3984, February 2005.

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

[RFC4170] Thompson, B., Koren, T., and D. Wing, "Tunneling Multiplexed Compressed RTP (TCRTP)", BCP 110, RFC 4170, November 2005.

[RFC4175] Gharai, L. and C. Perkins, "RTP Payload Format for Uncompressed Video", RFC 4175, September 2005.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[rfc-avpf] Joerg, "Extended RTP Profile for RTCP-based Feedback (RTP/AVPF)", August 2004.

[rfc-savpf]
Joerg, "Extended Secure RTP Profile for RTCP-based Feedback (RTP/SAVPF)", July 2004.

[rfc2223bis]
Reynolds, K., "Instructions to Request for Comments (RFC) Authors", August 2004.

[rfc3555bis]
Casner, L., "MIME Type Registration of RTP Payload Formats", October 2005.

[rtp-rtx] Jose, "RTP Retransmission Payload Format", March 2005.

[sdp-new] Perkins, "SDP: Session Description Protocol", July 2005.

Appendix A. RTP Payload Format Template

This section contains a template for writing an RTP payload format in form as a Internet draft. Text within [...] are instructions and must be removed. Some text proposals that are included are conditional. "..." is used to indicate where further text should be written.

A.1. Title

[The title shall be descriptive but as compact as possible. RTP is allowed and recommended abbreviation in the title]

RTP Payload format for ...

A.2. Front page boilerplate

Status of this Memo

[Insert the IPR notice boiler plate from BCP 79 that applies to this draft.]

[Insert the current Internet Draft document explanation. At the time of publishing it was:]

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

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

[Insert the ID list and shadow list reference. At the time of publishing it was:]

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

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

[Optionally: Select either of these paragraphs depending on draft status]

This document is an individual submission to the IETF. Comments

should be directed to the authors.

This document is a submission of the IETF AVT WG. Comments should be directed to the AVT WG mailing list, avt@ietf.org.

A.3. Abstract

[An payload format abstract should mention the capabilities of the format, for which media format is used, and a little about that codec formats capabilities. Any abbreviation used in the payload format must be spelled out here except the very well known like RTP. No references are allowed, no use of RFC 2119 language either.]

A.4. Table of Content

[All drafts over 15 pages in length must have an Table of Content.]

A.5. Introduction

[The introduction should provide a background and overview of the payload formats capabilities. No normative language in this section, i.e. no MUST, SHOULDS etc.]

A.6. Conventions, Definitions and Acronyms

[Define conventions, definitions and acronyms used in the document in this section. The most common definition used in RTP Payload formats are the RFC 2119 definitions of the upper case normative words, e.g. MUST and SHOULD.]

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.

A.7. Media Format Background

[The intention of this section is to enable reviewers and persons to get an overview of the capabilities and major properties of the media format. It should be kept short and concise and is not a complete replacement for reading the media format specification.]

A.8. Payload format

[Overview of payload structure]

A.8.1. RTP Header Usage

[RTP header usage needs to be defined. The fields that absolutely

need to be defined are timestamp and marker bit. Further field may be specified if used. All the rest should be left to their RTP specification definition]

The remaining RTP header fields are used as specified in RFC 3550.

A.8.2. Payload Header

[Define how the payload header if it exist are structured and used.]

A.8.3. Payload Data

[The payload data, i.e. what the media codec has produced. Commonly done through reference to media codec specification which defines how the data is structured. Rules for padding may need to be defined to bring data to octet alignment.]

A.9. Payload Examples

[One or more examples are good to help ease the understanding of the RTP payload format.]

A.10. Congestion Control Considerations

[This section is to describe the possibility to vary the bit-rate as a response to congestion. Below is also a proposal for an initial text that reference RTP and profiles definition of congestion control.]

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. An additional requirement if best-effort service is being used is: users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters.

A.11. Payload Format Parameters

This RTP payload format is identified using the ... media type which is registered in accordance with RFC XXXX [rfc3555bis] and using the template of RFC 4288 [RFC4288].

A.11.1. Media Type Definition

[Here the media type registration template from RFC 4288 is placed and filled out. This template is provided with some common RTP boilerplate.]

Type name:

Subtype name:

Required parameters:

Optional parameters:

Encoding considerations:

This media type is framed and binary, see section 4.8 in RFC4288 [RFC4288].

Security considerations:

Interoperability considerations:

Published specification:

Applications that use this media type:

Additional information:

Magic number(s):

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Intended usage: (One of COMMON, LIMITED USE or OBSOLETE.)

Restrictions on usage:

[The below text is for media types that is only defined for RTP payload formats. There exist certain media types that are defined both as RTP payload formats and file transfer. The rules for such types are documented in RFC 3555bis [rfc3555bis].]

This media type depends on RTP framing, and hence is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author:

Change controller:

IETF Audio/Video Transport working group delegated from the IESG.

(Any other information that the author deems interesting may be added below this line.)

[From RFC 4288: Some discussion of Macintosh file type codes and their purpose can be found in [MACOSFILETYPES]. Additionally, please refrain from writing "none" or anything similar when no file extension or Macintosh file type is specified, lest "none" be confused with an actual code value.]

A.11.2. Mapping to SDP

The mapping of the above defined payload format media type and its parameters SHALL be done according to Section 3 of RFC XXXX [rfc3555bis].

[More specific rules only need to be included if some parameter does not match these rules.]

A.11.2.1. Offer/Answer Considerations

[Here write your offer/answer consideration section, please see Section Section 3.3.2.1 for help.]

A.11.2.2. Declarative SDP Considerations

[Here write your considerations for declarative SDP, please see Section Section 3.3.2.2 for help.]

A.12. IANA Considerations

This memo requests that IANA registers [insert media type name here] as specified in Appendix A.11.1. The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" (http://www.iana.org/assignments/rtp-parameters).

[See Section Section 7.3 and consider if any of the parameter needs a registered name space.]

A.13. Securtiy Considerations

[See Section Section 7.1]

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [4], and in any applicable RTP profile. The main security considerations for the RTP packet carrying the RTP payload format defined within this memo are confidentiality, integrity and source authenticity. Confidentiality is achieved by encryption of

the RTP payload. Integrity of the RTP packets through suitable cryptographic integrity protection mechanism. Cryptographic system may also allow the authentication of the source of the payload. A suitable security mechanism for this RTP payload format should provide confidentiality, integrity protection and at least source authentication capable of determining if an RTP packet is from a member of the RTP session or not.

Note that the appropriate mechanism to provide security to RTP and payloads following this memo may vary. It is dependent on the application, the transport, and the signalling protocol employed. Therefore a single mechanism is not sufficient, although if suitable the usage of SRTP [RFC3711] is recommended. Other mechanism that may be used are IPsec [RFC4301] and TLS [RFC2246] (RTP over TCP), but also other alternatives may exist.

[Fill in here any further potential security threats]

A.14. References

[References must be classified as either normative or informative and added to the relevant section. References should use descriptive reference tags.]

A.14.1. Normative References

[Normative references are those that are required to be used to correctly implement the payload format.]

A.14.2. Informative References

[All other references.]

A.15. Author Addresses

[All Authors need to include their Name and email addresses as a minimal. Commonly also surface mail and possibly phone numbers are included.]

A.16. IPR Notice

[Use the appropriate boilerplate from Section 5 of BCP 79 [RFC3979].]

A.17. Copyright Notice

[Use the boilerplate from Section 5.4 and 5.5 of BCP 78 [RFC3978].]

Author's Address

Magnus Westerlund
Ericsson
Trondheimsgatan 23
Stockholm, SE-164 80
SWEDEN

Phone: +46 8 4048287
Fax: +46 8 757 55 50
Email: magnus.westerlund@ericsson.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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