

Audio/Video Transport Core Maintenance
Internet-Draft
Intended status: Standards Track
Expires: August 23, 2013

A. Williams
Audinate
K. Gross
AVA Networks
R. van Brandenburg
H. Stokking
TNO
February 19, 2013

RTP Clock Source Signalling
draft-ietf-avtcore-clksrc-02

Abstract

NTP timestamps are used by several RTP protocols for synchronisation and statistical measurements. This memo specifies SDP signalling identifying NTP timestamp clock sources and SDP signalling identifying the media clock sources in a multimedia session.

Requirements Language

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 [1].

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 August 23, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Applications	3
3. Definitions	4
4. Timestamp Reference Clock Source Signalling	5
4.1. Clock synchronization	5
4.2. Identifying NTP Reference Clocks	6
4.3. Identifying PTP Reference Clocks	6
4.4. Identifying Global Reference Clocks	7
4.5. Other Reference Clocks	8
4.6. Traceable Reference Clocks	8
4.7. SDP Signalling of Timestamp Clock Source	8
4.7.1. Examples	11
5. Media Clock Source Signalling	12
5.1. Asynchronously Generated Media Clock	12
5.2. Direct-Referenced Media Clock	12
5.3. Stream-Referenced Media Clock	13
5.4. SDP Signalling of Media Clock Source	14
5.5. Examples	16
6. Signalling considerations	17
6.1. Usage in Offer/Answer	18
6.2. Usage Outside of Offer/Answer	18
7. Security Considerations	18
8. IANA Considerations	19
9. References	20
9.1. Normative References	20
9.2. Informative References	21
Authors' Addresses	22

1. Introduction

RTP protocols use NTP format timestamps to facilitate multimedia session synchronisation and for providing estimates of round trip time (RTT) and other statistical parameters.

Information about media clock timing exchanged in NTP format timestamps may come from a clock which is synchronised to a global time reference, but this cannot be assumed nor is there a standardised mechanism available to indicate that timestamps are derived from a common reference clock. Therefore, RTP implementations typically assume that NTP timestamps are taken using unsynchronised clocks and must compensate for absolute time differences and rate differences. Without a shared reference clock, RTP can time align flows from the same source at a given receiver using relative timing, however tight synchronisation between two or more different receivers (possibly with different network paths) or between two or more senders is not possible.

High performance AV systems often use a reference media clock distributed to all devices in the system. The reference media clock is often distinct from the reference clock used to provide timestamps. A reference media clock may be provided along with an audio or video signal interface, or via a dedicated clock signal (e.g. genlock [19] or audio word clock [20]). If sending and receiving media clocks are known to be synchronised to a common reference clock, performance can be improved by minimising buffering and avoiding rate conversion.

This specification defines SDP signalling of timestamp clock sources and media reference clock sources.

2. Applications

Timestamp clock source and reference media clock signalling benefit applications requiring synchronised media capture or playout and low latency operation.

Examples include, but are not limited to:

Social TV : RTCP for inter-destination media synchronization [9] defines social TV as the combination of media content consumption by two or more users at different devices and locations and real-time communication between those users. An example of Social TV, is where two or more users are watching the same television broadcast at different devices and/or locations, while communicating with each other using text, audio and/or video. A

skew in the media playout of the two or more users can have adverse effects on their experience. A well-known use case here is one friend experiencing a goal in a football match well before or after other friends.

Video Walls : A video wall consists of multiple computer monitors, video projectors, or television sets tiled together contiguously or overlapped in order to form one large screen. Each of the screens reproduces a portion of the larger picture. In some implementations, each screen or projector may be individually connected to the network and receive its portion of the overall image from a network-connected video server or video scaler. Screens are refreshed at 50 or 60 hertz or potentially faster. If the refresh is not synchronized, the effect of multiple screens acting as one is broken.

Networked Audio : Networked loudspeakers, amplifiers and analogue I/O devices transmitting or receiving audio signals via RTP can be connected to various parts of a building or campus network. Such situations can for example be found in large conference rooms, legislative chambers, classrooms (especially those supporting distance learning) and other large-scale environments such as stadiums. Since humans are more susceptible to differences in audio delay, this use case needs even more accuracy than the video wall use case. Depending on the exact application, the need for accuracy can then be in the range of microseconds [21].

Sensor Arrays : Sensor arrays contain many synchronised measurement elements producing signals which are then combined to form an overall measurement. Accurate capture of the phase relationships between the various signals arriving at each element of the array is critically important for proper operation. Examples include towed or fixed sonar arrays, seismic arrays and phased arrays used in radar applications, for instance.

3. Definitions

The following definitions are used in this draft:

media level : Media level information applies to a single SDP media stream. In an SDP description, media-level information appears after each "m"-line.

multimedia session : A set of multimedia senders and receivers as well as the data streams flowing from senders to receivers. The Session Description Protocol (SDP) [2] describes multimedia sessions.

RTP media stream : A single stream of RTP packets identified by an RTP SSRC.

RTP media sender : The device generating an associated RTP media stream

SDP media stream : An RTP session potentially containing more than one RTP source. SDP media descriptions beginning with an "m"-line define the parameters of an SDP media stream.

session level : Session level information applies to an entire multimedia session. In an SDP description, session-level information appears before the first "m"-line.

source level : Source level information applies to a RTP media stream Source-Specific Media Attributes in the Session Description Protocol (SDP) [3] defines how source-level information is included into an SDP session description.

traceable time : A clock is considered to provide traceable time if it can be proven to be synchronised to International Atomic Time (TAI). Coordinated Universal Time (UTC) is a time standard synchronized to TAI. UTC is therefore also considered traceable time once leap seconds have been taken into account. GPS [10] is commonly used to provide a TAI traceable time reference. Some network time synchronisation protocols (e.g. PTP [11], NTP) can explicitly indicate that the master clock is providing a traceable time reference over the network.

4. Timestamp Reference Clock Source Signalling

The NTP timestamps used by RTP are taken by reading a local real-time clock at the sender or receiver. This local clock may be synchronised to another clock (time source) by some means or it may be unsynchronised. A variety of methods are available to synchronise local clocks to a reference time source, including network time protocols (e.g. NTP [12]) and radio clocks (e.g. GPS [10]).

The following sections describe and define SDP signalling, indicating whether and how the local timestamping clock in an RTP sender/receiver is synchronised to a reference clock.

4.1. Clock synchronization

Two or more local clocks that are sufficiently synchronised will produce timestamps for a given RTP event can be used as if they came from the same clock. Providing they are sufficiently synchronised,

timestamps produced in one RTP sender or receiver can be directly compared to a local clock in another RTP sender or receiver.

The accuracy of synchronisation required is application dependent. See Applications (Section 2) section for a discussion of applications and their corresponding requirements. To serve as a reference clock, clocks must minimally be syntonised (exactly frequency matched) to one another.

Sufficient synchronisation can typically be achieving by using a network time protocol (e.g. NTP, 802.1AS, IEEE 1588-2008) to synchronize all devices to a single master clock.

Another approach is to use clocks providing a global time reference (e.g. GPS, Galileo). This concept may be used in conjunction with network time protocols as some protocols (e.g. PTP, NTP) allow master clocks to indicate explicitly that they are providing traceable time.

4.2. Identifying NTP Reference Clocks

A single NTP server is identified by hostname (or IP address) and an optional port number. If the port number is not indicated, it is assumed to be the standard NTP port (123).

Two or more NTP servers may be listed at the same level in the session description to indicate that they are interchangeable. RTP senders or receivers can use any of the listed NTP servers to govern a local clock that is equivalent to a local clock slaved to a different server.

4.3. Identifying PTP Reference Clocks

The IEEE 1588 Precision Time Protocol (PTP) family of clock synchronisation protocols provides a shared reference clock in an network - typically a LAN. IEEE 1588 provides sub-microsecond synchronisation between devices on a LAN and typically locks within seconds at startup. With support from Ethernet switches, IEEE 1588 protocols can achieve nanosecond timing accuracy in LANs. Network interface chips and cards supporting hardware time-stamping of timing critical protocol messages are also available.

Three flavours of IEEE 1588 are in use today:

- o IEEE 1588-2002 [13]: the original "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems". This is also known as IEEE1588v1 or PTPv1.

- o IEEE 1588-2008 [11]: the second version of the "Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems". This is a revised version of the original IEEE1588-2002 standard and is also known as IEEE1588v2 or PTPv2. IEEE 1588-2008 is not protocol compatible with IEEE 1588-2002.
- o IEEE 802.1AS [14]: "Timing and Synchronization for Time Sensitive Applications in Bridged Local Area Networks". This is a Layer-2 only profile of IEEE 1588-2008 for use in Audio/Video Bridged LANs as described in IEEE 802.1BA-2011 [15].

Each IEEE 1588 clock is identified by a globally unique EUI-64 called a "ClockIdentity". A slave clock using one of the IEEE 1588 family of network time protocols acquires the ClockIdentity/EUI-64 of the grandmaster clock that is the ultimate source of timing information for the network. A boundary clock which is itself slaved to another boundary clock or the grandmaster passes the grandmaster ClockIdentity through to its slaves.

Several instances of the IEEE 1588 protocol may operate independently on a single network, forming distinct PTP domains, each of which may have a different grandmaster clock. As the IEEE 1588 standards have developed, the definition of PTP domains has changed. IEEE 1588-2002 identifies protocol subdomains by a textual name, but IEEE 1588-2008 identifies protocol domains using a numeric domain number. 802.1AS is a Layer-2 profile of IEEE 1588-2008 supporting a single numeric clock domain (0).

When PTP domains are signalled via SDP, senders and receivers SHOULD check that both grandmaster ClockIdentity and PTP domain match when determining clock equivalence.

The PTP protocols employ a distributed election protocol called the "Best Master Clock Algorithm" (BMCA) to determine the active clock master. The clock master choices available to BMCA can be restricted or biased by configuration parameters to influence the election process. In some systems it may be desirable to limit the number of possible PTP clock masters to avoid the need to re-signal timestamp clock sources when the clock master changes.

4.4. Identifying Global Reference Clocks

Global reference clocks provide a source of traceable time, typically via a hardware radio receiver interface. Examples include GPS and Galileo. Apart from the name of the reference clock system, no further identification is required.

4.5. Other Reference Clocks

RFC 3550 allows senders and receivers to either use a local wallclock reference for their NTP timestamps or, by setting the timestamp field to 0, to supply no timestamps at all. Both are common practice in embedded RTP implementations. These clocks are identified as "local" and can only be assumed to be equivalent to clocks originating from the same device.

In other systems, all RTP senders and receivers may use a timestamp clock synchronised to a reference clock that is not provided by one of the methods listed above. Examples may include the reference time information provided by digital television or cellular services. These sources are identified as "private" reference clocks. All RTP senders and receivers in a session using a private reference clock are assumed to have a mechanism outside this specification for determining whether their timestamp clocks are equivalent.

4.6. Traceable Reference Clocks

A timestamp clock source may be labelled "traceable" if it is known to be delivering traceable time. Providing adjustments are made for differing epochs, timezones and leap seconds, timestamps taken using clocks synchronised to a traceable time source can be directly compared even if the clocks are synchronised to different sources or via different mechanisms.

Since all NTP and PTP servers providing traceable time can be directly compared, it is not necessary to identify traceable time servers by protocol address or other identifiers.

4.7. SDP Signalling of Timestamp Clock Source

Specification of the timestamp reference clock source may be at any or all levels (session, media or source) of an SDP description (see level definitions (Section 3) earlier in this document for more information).

Timestamp clock source signalling included at session-level provides default parameters for all RTP sessions and sources in the session description. More specific signalling included at the media level overrides default session level signalling. More specific signalling included at the source level overrides default media level signalling.

If timestamp clock source signalling is included anywhere in an SDP description, it must be properly defined for all levels in the description. This may simply be achieved by providing default

signalling at the session level.

Timestamp reference clock parameters may be repeated at a given level (i.e. for a session or source) to provide information about additional servers or clock sources. If the attribute is repeated at a given level, all clocks described at that level are assumed to be equivalent. Traceable time sources **MUST NOT** be mixed with non-traceable time sources at any given level.

Note that clock source parameters may change from time to time, for example, as a result of a PTP clock master election. The SIP [4] protocol supports re-signalling of updated SDP information, however other protocols may require additional notification mechanisms.

Figure 1 shows the ABNF [5] grammar for the SDP reference clock source information.

```

timestamp-refclk = "a=ts-refclk:" clksrc CRLF
clksrc = ntp / ptp / gps / gal / local / private / clksrc-ext

ntp          = "ntp=" ntp-server-addr
ntp-server-addr = host [ ":" port ]
ntp-server-addr =/ "traceable"

ptp          = "ptp=" ptp-version ":" ptp-server
ptp-version  = "IEEE1588-2002"
ptp-version  =/ "IEEE1588-2008"
ptp-version  =/ "IEEE802.1AS-2011"
ptp-version  =/ ptp-version-ext
ptp-version-ext = token

ptp-server   = ptp-gmid [ ":" ptp-domain ] / "traceable"
ptp-gmid     = EUI64
ptp-domain   = ptp-domain-name / ptp-domain-nmbr
ptp-domain-name = "domain-name=" 16ptp-domain-char
ptp-domain-char = %x21-7E / %x00
                ; allowed characters: 0x21-0x7E (IEEE 1588-2002)
ptp-domain-nmbr = "domain-nmbr=" %x00-7f
                ; allowed number range: 0-127 (IEEE 1588-2008)

gps          = "gps"
gal          = "gal"
local        = "local"
private      = "private" [ ":" "traceable" ]

clksrc-ext   = token

host         = hostname / IPv4address / IPv6reference
hostname     = *( domainlabel "." ) toplabel [ "." ]
toplabel     = ALPHA / ALPHA *( alphanum / "-" ) alphanum
domainlabel  = alphanum
              / alphanum *( alphanum / "-" ) alphanum
IPv4address  = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6reference = "[" IPv6address "]"
IPv6address  = hexpart [ ":" IPv4address ]
hexpart      = hexseq / hexseq ":@" [ hexseq ] / ":@" [ hexseq ]
hexseq       = hex4 *( ":" hex4)
hex4         = 1*4HEXDIG

port = 1*DIGIT

EUI64 = 7(2HEXDIG "-") 2HEXDIG

```

Figure 1: Timestamp Reference Clock Source Signalling

4.7.1. Examples

Figure 2 shows an example SDP description with a timestamp reference clock source defined at the session level.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 192.0.2.1
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 233.252.0.1/64
t=2873397496 2873404696
a=recvonly
a=ts-refclk:ntp=traceable
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

Figure 2: Timestamp reference clock definition at the session level

Figure 3 shows an example SDP description with timestamp reference clock definitions at the media level overriding the session level defaults. Note that the synchronisation confidence timestamp appears on the first attribute at the media level only.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 192.0.2.1
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 233.252.0.1/64
t=2873397496 2873404696
a=recvonly
a=ts-refclk:local
m=audio 49170 RTP/AVP 0
a=ts-refclk:ntp=203.0.113.10 2011-02-19 21:03:20.345+01:00
a=ts-refclk:ntp=198.51.100.22
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
a=ts-refclk:ptp=IEEE802.1AS-2011:39-A7-94-FF-FE-07-CB-D0
```

Figure 3: Timestamp reference clock definition at the media level

Figure 4 shows an example SDP description with a timestamp reference clock definition at the source level overriding the session level default.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 192.0.2.1
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 233.252.0.1/64
t=2873397496 2873404696
a=recvonly
a=ts-refclk:local
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
a=ssrc:12345 ts-refclk:ptp=IEEE802.1AS-2011:39-A7-94-FF-FE-07-CB-D0
```

Figure 4: Timestamp reference clock signalling at the source level

5. Media Clock Source Signalling

The media clock source for a stream determines the timebase used to advance the RTP timestamps included in RTP packets. The media clock may be asynchronously generated by the sender, it may be generated in fixed relationship to the reference clock or it may be generated with respect to another stream on the network (which is presumably being received by the sender).

5.1. Asynchronously Generated Media Clock

In the simplest sender implementation, the sender generates media by sampling audio or video according to a free-running local clock. The RTP timestamps in media packets are advanced according to this media clock and packet transmission is typically timed to regular intervals on this timeline. The sender may or may not include an NTP timestamp in sender reports to allow mapping of this asynchronous media clock to a reference clock.

The asynchronously generated media clock is the assumed mode of operation when there is no signalling of media clock source. Alternatively, asynchronous media clock may be explicitly signalled.

```
a=mediaclock:sender
```

5.2. Direct-Referenced Media Clock

A media clock may be directly derived from a reference clock. For this case it is required that a reference clock be specified with an `a=ts-refclk` attribute (Section 4.7).

The signalling optionally indicates a media clock offset value. The offset indicates the RTP timestamp value at the epoch (time of origin) of the reference clock. If no offset is signalled, the offset can be inferred at the receiver by examining RTCP sender reports which contain NTP and RTP timestamps which combined define a mapping.

A rate modifier may be specified. The modifier is expressed as the ratio of two integers and modifies the rate specified or implied by the media description by this ratio. If omitted, the rate is assumed to be the exact rate specified or implied by the media format. For example, without a rate specification, the media clock for an 8 kHz G.711 audio stream will advance exactly 8000 units for each second advance in the reference clock from which it is derived.

The rate modifier is primarily useful for accommodating certain "oddball" audio sample rates associated with NTSC video (see Figure 7). Modified rates are not advised for video streams which generally use a 90 kHz RTP clock regardless of frame rate or sample rate used for embedded audio.

```
a=mediaclock:direct[=<offset>] [rate=<rate numerator>/<rate
denominator>]
```

5.3. Stream-Referenced Media Clock

A common synchronisation architecture for audio/visual systems involves distributing a reference media clock from a master device to a number of slave devices, typically by means of a cable. Examples include audio word clock distribution and video black burst distribution. In this case, the media clock is locally generated, often by a crystal oscillator and is not locked to a timestamp reference clock.

To support this architecture across a network, a master clock identifier is associated with an RTP media stream carrying media clock timing information from a master device. The master clock identifier represents a media clock source in the master device. Slave devices in turn associate the master media clock identifier with streams they transmit, signalling the synchronisation relationship between the master and slave devices.

Slave devices recover media clock timing from the clock master stream, using it to synchronise the slave media clock with the master. Timestamps in the master clock RTP media stream are taken using the timestamp reference clock shared by the master and slave devices. The timestamps communicate information about media clock timing (rate, phase) from the master to the slave devices.

Timestamps are communicated in the usual RTP fashion via RTCP SRs, or via the RFC6051 [6] header extension. The stream media format may indicate other clock information, such as the nominal rate.

Note that slaving of a device media clock to a master device does not affect the usual RTP lip sync / time alignment algorithms. Time aligned playout of two or more RTP sources still relies upon NTP timestamps supplied via RTCP SRs or by the RFC6051 timestamp header extension.

In a given system, master clock identifiers must be unique. Such identifiers MAY be manually configured, however 17 octet string identifiers SHOULD be generated according to the "short-term persistent RTCP CNAME" algorithm as described in RFC6222 [7].

A reference stream can be an RTP stream or AVB stream based on the IEEE 1722 [16] standard.

An RTP clock master stream SHOULD be identified at the source level by an SSRC and master clock identifier. If master clock identifiers are declared at the media or session level, all RTP sources at or below the level of declaration MUST provide equivalent timing to a slave receiver.

```
a=ssrc:<media-clock-master-ssrc-id> mediack:master-id=<media-  
clock-master-id>
```

An RTP media sender indicates that it is slaved to a clock master via a clock master identifier:

```
a=mediack:master-id=<media-clock-master-id>
```

An RTP media sender indicates that it is slaved to an IEEE 1722 clock master via a stream identifier (an EUI-64):

```
a=mediack:IEEE1722=<StreamID>
```

5.4. SDP Signalling of Media Clock Source

Specification of the media clock source may be at any or all levels (session, media or source) of an SDP description (see level definitions (Section 3) earlier in this document for more information).

Media clock source signalling included at session level provides default parameters for all RTP sessions and sources in the session description. More specific signalling included at the media level overrides default session level signalling. Further, source-level

signalling overrides media clock source signalling at the enclosing media level and session level.

Media clock source signalling may be present or absent on a per-stream basis. In the absence of media clock source signals, receivers assume an asynchronous media clock generated by the sender.

Media clock source parameters may be repeated at a given level (i.e. for a session or source) to provide information about additional clock sources. If the attribute is repeated at a given level, all clocks described at that level are comparable clock sources and may be used interchangeably.

Figure 5 shows the ABNF [5] grammar for the SDP media clock source information.

```
mediaclock-master = "a=ssrc:" integer SP clk-master-id

clk-master-id = "mediaclock:master-id=" master-id

timestamp-mediaclock = "a=mediaclock:" mediaclock

mediaclock = sender / refclk / streamid / mediaclock-ext

sender = "sender" sender-ext

sender-ext = token

refclk = "direct" [ "=" 1*DIGIT ] [rate] [direct-ext]

rate = [ SP "rate=" integer "/" integer ]

direct-ext = token

streamid = "master-id=" master-id
streamid =/ "IEEE1722=" avb-stream-id
streamid =/ streamid-ext

master-id = EUI48
avb-stream-id = EUI64

EUI48 = 5(2HEXDIG ":") 2HEXDIG
EUI64 = 7(2HEXDIG ":") 2HEXDIG

streamid-ext = token

mediaclock-ext = token
```

Figure 5: Media Clock Source Signalling

5.5. Examples

Figure 6 shows an example SDP description 8 channels of 24-bit, 48 kHz audio transmitted as a multicast stream. Media clock is derived directly from an IEEE 1588-2008 reference.

```
v=0
o=- 1311738121 1311738121 IN IP4 192.0.2.1
c=IN IP4 233.252.0.1/64
s=
t=0 0
m=audio 5004 RTP/AVP 96
a=rtpmap:96 L24/48000/8
a=sendonly
a=ts-refclk:ptp=IEEE1588-2008:39-A7-94-FF-FE-07-CB-D0:0
a=mediaclock:direct=963214424
```

Figure 6: Media clock directly referenced to IEEE 1588-2008

Figure 7 shows an example SDP description 2 channels of 24-bit, 44056 kHz NTSC "pull-down" media clock derived directly from an IEEE 1588-2008 reference clock

```
v=0
o=- 1311738121 1311738121 IN IP4 192.0.2.1
c=IN IP4 233.252.0.1/64
s=
t=0 0
m=audio 5004 RTP/AVP 96
a=rtpmap:96 L24/44100/2
a=sendonly
a=ts-refclk:ptp=IEEE1588-2008:39-A7-94-FF-FE-07-CB-D0:0
a=mediaclock:direct=963214424 rate=1000/1001
```

Figure 7: "Oddball" sample rate directly referenced to IEEE 1588-2008

Figure 8 shows the same 48 kHz audio transmission from Figure 6 with media clock derived from another RTP stream.


```
v=0
o=- 1311738121 1311738121 IN IP4 192.0.2.1
c=IN IP4 233.252.0.1/64
s=
t=0 0
m=audio 5004 RTP/AVP 96
a=rtpmap:96 L24/48000/2
a=sendonly
a=ts-refclk:ptp=IEEE1588-2008:39-A7-94-FF-FE-07-CB-D0:0
a=mediaclock:master-id=00:60:2b:20:12:1f
```

Figure 8: RTP stream with media clock slaved to a master device

Figure 9 shows the same 48 kHz audio transmission from Figure 6 with media clock derived from an IEEE 1722 AVB stream.

```
v=0
o=- 1311738121 1311738121 IN IP4 192.0.2.1
c=IN IP4 233.252.0.1/64
s=
t=0 0
m=audio 5004 RTP/AVP 96
a=rtpmap:96 L24/48000/2
a=sendonly
a=ts-refclk:ptp=IEEE1588-2008:39-A7-94-FF-FE-07-CB-D0:0
a=mediaclock:IEEE1722=38-D6-6D-8E-D2-78-13-2F
```

Figure 9: RTP stream with media clock slaved to an IEEE1722 master device

6. Signalling considerations

Signaling for timestamp clock source (Section 4.7) and media clock source (Section 5.4) is defined to be used either by applications that implement the SDP Offer/Answer model [8] or by applications that use SDP to describe media and transport configurations.

A description or offer may include reference clock signalling, media clock signalling or both. If no reference clock is specified, the direct-referenced media clock (Section 5.2) is not allowed. If no media clock is specified, an asynchronous media clock (Section 5.1) is assumed. stream-referenced media clock (Section 5.3) may be used with or without a reference clock specification. If a reference clock is not signalled, the stream may be established as rate synchronized however time synchronisation is not guaranteed.

6.1. Usage in Offer/Answer

An answer to an offer with direct-referenced media clock and reference clock specification must include the same media clock and reference clock signalling in which case a connection is established using the specified synchronisation. Alternatively the answer may omit both the signals or return only the reference clock specification. In this case, a connection is established assuming an asynchronous media clock.

An answer to an offer with media-referenced media clock specification must include the same media clock specification. The answer **MUST** include the same reference clock signalling or may drop the reference clock signalling. If reference clock signalling is not present in the answer, either due to not being present in the offer or due to being dropped by the answerer, the stream may be established as rate synchronized but not time synchronized.

An asynchronous media clock is the default media clock mode. This mode may be explicitly signalled or presumed due to lack of signalling. Asynchronous media clocking does not require reference clock signalling. An offer with asynchronous media clocking **MAY** include reference clock signalling. Because the asynchronous media clock is the default mode, the answerer is not required to explicitly signal this even if it is explicitly signalled in the offer.

6.2. Usage Outside of Offer/Answer

SDP can be employed outside of the Offer/Answer context, for instance for multimedia sessions that are announced through the Session Announcement Protocol (SAP) [17], or streamed through the Real Time Streaming Protocol (RTSP) [18]. The signaling model is simpler, as the sender does not negotiate parameters, but the functionality expected from specifying media clock and reference clock attributes is the same as in Offer/Answer.

7. Security Considerations

Entities receiving and acting upon an SDP message **SHOULD** be aware that a session description cannot be trusted unless it has been obtained by an authenticated transport protocol from a known and trusted source. Many different transport protocols may be used to distribute session description, and the nature of the authentication will differ from transport to transport. For some transports, security features are often not deployed. In case a session description has not been obtained in a trusted manner, the endpoint **SHOULD** exercise care because, among other attacks, the media sessions

received may not be the intended ones, the destination where media is sent to may not be the expected one, any of the parameters of the session may be incorrect.

Incorrect reference or media clock parameters may cause devices or streams to synchronize to unintended clock sources. Normally this simply results in failure to make a media connection or failure to synchronize once connected. Enough devices fraudulently assigned to a specific clock source (e.g. a particular IEEE 1588 grandmaster) may, however, constitute a successful denial of service attack on that source. Devices MAY wish to validate the integrity of the clock description through some means before connecting to unfamiliar clock sources.

8. IANA Considerations

The SDP attribute "ts-refclk" defined by this document is registered with the IANA registry of SDP Parameters as follows:

SDP Attribute ("att-field"):

Attribute name:	ts-refclk
Long form:	Timestamp reference clock source
Type of name:	att-field
Type of attribute:	session, media and source level
Subject to charset:	no
Purpose:	See section 4 of this document
Reference:	This document
Values:	see this document and registrations below

The attribute has an extensible parameter field and therefore a registry for these parameters is required. This document creates an IANA registry called the Timestamp Reference Clock Source Parameters Registry. It contains the six parameters defined in Figure 1: "ntp", "ptp", "gps", "gal", "local", "private".

The SDP attribute "mediaclock" defined by this document is registered with the IANA registry of SDP Parameters as follows:

SDP Attribute ("att-field"):

Attribute name: mediaclk

Long form: Media clock source

Type of name: att-field

Type of attribute: session and media level

Subject to charset: no

Purpose: See section 5 of this document

Reference: This document

Values: see this document and registrations below

The attribute has an extensible parameter field and therefore a registry for these parameters is required. This document creates an IANA registry called the Media Clock Source Parameters Registry. It contains the three parameters defined in Figure 5: "sender", "direct", "master", "slave" and "IEEE1722".

9. References

9.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [3] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.
- [4] 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.
- [5] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [6] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP

Flows", RFC 6051, November 2010.

- [7] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.
- [8] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

9.2. Informative References

- [9] Brandenburg, R., Stokking, H., Deventer, O., Boronat, F., Montagud, M., and K. Gross, "Inter-destination Media Synchronization using the RTP Control Protocol (RTCP)", draft-ietf-avtcore-idms-08 (work in progress), January 2013.
- [10] Global Positioning Systems Directorate, "Navstar GPS Space Segment/Navigation User Segment Interfaces", September 2011.
- [11] Institute of Electrical and Electronics Engineers, "1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008, 2008, <<http://standards.ieee.org/findstds/standard/1588-2008.html>>.
- [12] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [13] Institute of Electrical and Electronics Engineers, "1588-2002 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2002, 2002, <<http://standards.ieee.org/findstds/standard/1588-2002.html>>.
- [14] "Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", <<http://standards.ieee.org/findstds/standard/802.1AS-2011.html>>.
- [15] "Audio Video Bridging (AVB) Systems", <<http://standards.ieee.org/findstds/standard/802.1BA-2011.html>>.
- [16] "IEEE Standard for Layer 2 Transport Protocol for Time Sensitive Applications in a Bridged Local Area Network", <<http://standards.ieee.org/findstds/standard/1722-2011.html>>.

- [17] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.
- [18] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

URIs

- [19] <<http://en.wikipedia.org/wiki/Genlock>>
- [20] <http://en.wikipedia.org/wiki/Word_clock>
- [21] <<http://www.ieee802.org/1/files/public/docs2007/as-dolsen-time-accuracy-0407.pdf>>

Authors' Addresses

Aidan Williams
Audinate
Level 1, 458 Wattle St
Ultimo, NSW 2007
Australia

Phone: +61 2 8090 1000
Fax: +61 2 8090 1001
Email: aidan.williams@audinate.com
URI: <http://www.audinate.com/>

Kevin Gross
AVA Networks
Boulder, CO
US

Email: kevin.gross@avanw.com
URI: <http://www.avanw.com/>

Ray van Brandenburg
TNO
Brassersplein 2
Delft 2612CT
the Netherlands

Phone: +31-88-866-7000
Email: ray.vanbrandenburg@tno.nl

Hans Stokking
TNO
Brassersplein 2
Delft 2612CT
the Netherlands

Email: hans.stokking@tno.nl

AVTCore
Internet-Draft
Updates: 3550 (if approved)
Intended status: Standards Track
Expires: August 23, 2013

K. Gross
AVA Networks
R. van Brandenburg
TNO
February 19, 2013

RTP and Leap Seconds
draft-ietf-avtcore-leap-second-02

Abstract

This document discusses issues that arise when RTP sessions span Universal Coordinate Time (UTC) leap seconds. It updates RFC 3550 to describe how RTP senders and receivers should behave in the presence of leap seconds.

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 August 23, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Leap seconds	3
3.1. UTC behavior during leap second	4
3.2. NTP behavior during leap second	4
3.3. POSIX behavior during leap second	4
3.4. Summary of leap-second behaviors	4
4. Recommendations	5
4.1. RTP Sender Reports	6
4.2. RTP Packet Playout	6
5. Security Considerations	6
6. IANA Considerations	7
7. Acknowledgements	7
8. References	7
8.1. Normative References	7
8.2. Informative References	7
Authors' Addresses	8

1. Introduction

In some media networking applications, RTP streams are referenced to a wall-clock time (absolute date and time). This is accomplished through use of the NTP timestamp field in the RTCP sender report (SR) to create a mapping between RTP timestamps and the wall clock. When a wall-clock reference is used, the playout time for RTP packets is referenced to the wall clock. Smooth and continuous media playout requires a smooth and continuous time base. The time base used by the wall clock may include leap seconds which are not rendered smoothly.

This document updates RFC 3550 [1] providing recommendations for smoothly rendering streamed media referenced to common wall clocks which do not have smooth or continuous behavior in the presence of leap seconds.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2] and indicate requirement levels for compliant implementations.

3. Leap seconds

The world scientific time standard is International Atomic Time (TAI) which is based on vibrations of cesium atoms in an atomic clock. The world civil time is based on the rotation of the Earth. In 1972 the civil time standard, Coordinated Universal Time (UTC), was redefined in terms of TAI and the concept of leap seconds was introduced to allow UTC to remain synchronized with with the rotation of the Earth.

Leap seconds are scheduled by the International Earth Rotation and Reference Systems Service. Leap seconds may be scheduled at the last day of any month but are preferentially scheduled for December and June and secondarily March and September.[3] Because Earth's rotation is unpredictable, leap seconds are typically not scheduled more than six months in advance.

Leap seconds do not respect local time and always occur at the end of the UTC day. Leap seconds can be scheduled to either add or remove a second from the day. A leap second that adds an extra second is known as a positive leap second. A leap second that skips a second is known as a negative leap second. All leap seconds since their introduction in 1972 have been scheduled in June or December and all

have been positive.

NOTE- The ITU is studying a proposal which could eventually eliminate leap seconds from UTC. As of January 2012, this proposal is expected to be decided no earlier than 2015.[4]

3.1. UTC behavior during leap second

UTC clocks insert a 61st second at the end of the day when a leap second is scheduled. The leap second is designated "23h 59m 60s".

3.2. NTP behavior during leap second

Under NTP [5] a leap second is inserted at the beginning of the last second of the day. This results in the clock freezing or slowing for one second immediately prior to the last second of the affected day. This results in the last second of the day having a real-time duration of two seconds. Timestamp accuracy is compromised during this period because the clock's rate is not well defined.

3.3. POSIX behavior during leap second

Most POSIX systems insert the leap second at the end of the last second of the day. This results in repetition of the last second. A timestamp within the last second of the day is therefore ambiguous in that it can refer to a moment in time in either of the last two seconds of a day containing a leap second.

3.4. Summary of leap-second behaviors

Table 1 summarizes behavior across a leap second for the wall clocks discussed above.

Table 1 illustrates the leap second that occurred June 30, 2012 when the offset between International Atomic time (TAI) and UTC changed from 34 to 35 seconds. The first column shows RTP timestamps for an 8 kHz audio stream. The second column shows the TAI reference. Following columns show behavior for the leap-second-bearing wall clocks described above. Time values are shown at half-second intervals.

RTP	TAI	UTC	POSIX	NTP
8000	00:00:32.500	23:59:58.500	23:59:58.500	23:59:58.500
12000	00:00:33.000	23:59:59.000	23:59:59.000	23:59:59.000
16000	00:00:33.500	23:59:59.500	23:59:59.500	23:59:59.500
20000	00:00:34.000	23:59:60.000	23:59:59.000	00:00:00.000
24000	00:00:34.500	23:59:60.500	23:59:59.500	00:00:00.000
28000	00:00:35.000	00:00:00.000	00:00:00.000	00:00:00.000
32000	00:00:35.500	00:00:00.500	00:00:00.500	00:00:00.500

Table 1

NOTE- Some NTP implementations do not entirely freeze the clock while the leap second is inserted. Successive calls to retrieve system time return infinitesimally larger (e.g. 1 microsecond or 1 nanosecond) time values. This behavior is designed to satisfy assumptions applications may make that time increases monotonically. This behavior occurs in the least-significant bits of the time value and so is not typically visible in the human-readable format shown in the table.

4. Recommendations

Senders and receivers which are not referenced to a wall clock are not affected by issues associated with leap seconds and no special accommodation is required.

RTP implementation using a wall-clock reference is simplified by using a clock with a timescale which does not include leap seconds. IEEE 1588, [6] GPS [7] and other TAI [8] references do not include leap seconds. NTP time, operating system clocks and other UTC references include leap seconds.

All participants working to a leap-second-bearing reference SHOULD recognize leap seconds and have a working communications channel to receive notification of leap second scheduling. Without prior knowledge of leap second schedule, NTP servers and clients may become offset by exactly one second with respect to their UTC reference. This potential discrepancy begins when a leap second occurs and ends when all participants receive a time update from a server or peer. Depending on the system implementation, the offset can last anywhere from a few seconds to a few days. A long-lived discrepancy can be particularly disruptive to RTP operation.

Because of the timestamp ambiguity, positive leap seconds can

introduce and the inconsistent manner in which different systems accommodate leap seconds, generating or using NTP timestamps during the entire last second of a day on which a positive leap second has been scheduled SHOULD be avoided. Note that the period to be avoided has a real-time duration of two seconds. In the Table 1 example, the region to be avoided is indicated by RTP timestamps 12000 through 28000

Negative leap seconds do not introduce timestamp ambiguity or other complications. No special treatment with respect to RTP timestamps is required in the presence of a negative leap second.

4.1. RTP Sender Reports

RTP Senders working to a leap-second-bearing reference SHOULD NOT generate sender reports containing an originating NTP timestamp in the vicinity of a positive leap second. To maintain a consistent RTCP schedule and avoid the risk of unintentional timeouts, such senders MAY send receiver reports in place of sender reports in the vicinity of the leap second.

For the purpose of suspending sender reports in the vicinity of a leap second, senders MAY assume a positive leap second occurs at the end of the last day of every month.

Receivers working to a leap-second-bearing reference SHOULD ignore timestamps in any sender reports generated in the vicinity of a positive leap second.

For the purpose of ignoring sender reports in the vicinity of a leap second, receivers MAY assume a positive leap second occurs at the end of the last day of every month.

4.2. RTP Packet Playout

Receivers working to a leap-second-bearing reference SHOULD take both positive and negative leap seconds in the reference into account in determining playout time based on RTP timestamps for data in RTP packets.

5. Security Considerations

RTP streams using a wall-clock reference as discussed here present an additional attack vector compared to self-clocking streams. Manipulation of the wall clock at either sender or receiver can potentially disrupt streaming.

For an RTP stream operating to an leap-second-bearing reference to operate reliably across a leap second, sender and receiver must both be aware of the leap second. It is possible to disrupt a stream by blocking or delaying leap second notification to one of the participants. Streaming can be similarly affected if one of the participants can be tricked into believing a leap second has been scheduled where there is not one. These vulnerabilities are present in RFC 3550 [1] and these new recommendations neither heighten or diminish them. Integrity of the leap second schedule is the responsibility of the operating system and time distribution mechanism both of which are outside the scope of RFC 3550 [1] and these recommendations.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgements

The authors would like to thank Steve Allen for his valuable comments in helping to improve this document.

8. References

8.1. Normative References

- [1] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

- [3] ITU-R, "Recommendation ITU-R TF.460-4 - Standard-frequency and time-signal emissions", February 2002.
- [4] ITU-R Working Party 7A, "Question SG07.236", February 2012.
- [5] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [6] IEEE, "IEEE Standard for a Precision Clock Synchronization

Protocol for Networked Measurement and Control Systems",
July 2008.

[7] Global Positioning Systems Directorate, "Navstar GPS Space
Segment/Navigation User Segment Interfaces", September 2011.

[8] BIPM, "Circular T", May 2012.

Authors' Addresses

Kevin Gross
AVA Networks
Boulder, CO
US

Email: kevin.gross@avanw.com

Ray van Brandenburg
TNO
Brassersplein 2
Delft 2612CT
the Netherlands

Phone: +31-88-866-7000
Email: ray.vanbrandenburg@tno.nl

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2013

C. Perkins
University of Glasgow
V. Singh
Aalto University
February 22, 2013

Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions
draft-ietf-avtcore-rtcp-circuit-breakers-02

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 26, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Background	3
4. RTP Circuit Breakers for Systems Using the RTP/AVP Profile . .	6
4.1. RTP/AVP Circuit Breaker #1: Media Timeout	8
4.2. RTP/AVP Circuit Breaker #2: RTCP Timeout	8
4.3. RTP/AVP Circuit Breaker #3: Congestion	9
4.4. Ceasing Transmission	12
5. RTP Circuit Breakers for Systems Using the RTP/AVPF Profile .	12
6. Impact of RTCP XR	13
7. Impact of Explicit Congestion Notification (ECN)	14
8. Security Considerations	14
9. IANA Considerations	14
10. Acknowledgements	14
11. References	15
11.1. Normative References	15
11.2. Informative References	15
Authors' Addresses	17

1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is widely used in voice-over-IP, video teleconferencing, and telepresence systems. Many of these systems run over best-effort UDP/IP networks, and can suffer from packet loss and increased latency if network congestion occurs. Designing effective RTP congestion control algorithms, to adapt the transmission of RTP-based media to match the available network capacity, while also maintaining the user experience, is a difficult but important problem. Many such congestion control and media adaptation algorithms have been proposed, but to date there is no consensus on the correct approach, or even that a single standard algorithm is desirable.

This memo does not attempt to propose a new RTP congestion control algorithm. Rather, it proposes a minimal set of "circuit breakers"; conditions under which there is general agreement that an RTP flow is causing serious congestion, and ought to cease transmission. It is expected that future standards-track congestion control algorithms for RTP will operate within the envelope defined by this memo.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. This interpretation of these key words applies only when written in ALL CAPS. Mixed- or lower-case uses of these key words are not to be interpreted as carrying special significance in this memo.

3. Background

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

Congestion control for unicast RTP traffic can be implemented in one

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

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

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

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

- o The base RTP specification [RFC3550] defines RTCP Reception Report (RR) packets to convey reception quality feedback information, and Sender Report (SR) packets to convey information about the media transmission. RTCP SR packets contain data that can be used to reconstruct media timing at a receiver, along with a count of the total number of octets and packets sent. RTCP RR packets report on the fraction of packets lost in the last reporting interval, the cumulative number of packets lost, the highest sequence number received, and the inter-arrival jitter. The RTCP RR packets also contain timing information that allows the sender to estimate the network round trip time (RTT) to the receivers. RTCP reports are

sent periodically, with the reporting interval being determined by the number of SSRCs used in the session and a configured session bandwidth estimate (the number of SSRCs used is usually two in a unicast session, one for each participant, but can be greater if the participants send multiple media streams). The interval between reports sent from each receiver tends to be on the order of a few seconds on average, and it is randomised to avoid synchronisation of reports from multiple receivers. RTCP RR packets allow a receiver to report ongoing network congestion to the sender. However, if a receiver detects the onset of congestion partway through a reporting interval, the base RTP specification contains no provision for sending the RTCP RR packet early, and the receiver has to wait until the next scheduled reporting interval.

- o The RTCP Extended Reports (XR) [RFC3611] allow reporting of more complex and sophisticated reception quality metrics, but do not change the RTCP timing rules. RTCP extended reports of potential interest for congestion control purposes are the extended packet loss, discard, and burst metrics [RFC3611], [I-D.ietf-xrblock-rtcp-xr-discard], [I-D.ietf-xrblock-rtcp-xr-discard-rle-metrics], [I-D.ietf-xrblock-rtcp-xr-burst-gap-discard], [I-D.ietf-xrblock-rtcp-xr-burst-gap-loss]; and the extended delay metrics [RFC6843], [RFC6798]. Other RTCP Extended Reports that could be helpful for congestion control purposes might be developed in future.
- o Rapid feedback about the occurrence of congestion events can be achieved using the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] in place of the more common RTP/AVP profile [RFC3551]. This modifies the RTCP timing rules to allow RTCP reports to be sent early, in some cases immediately, provided the average RTCP reporting interval remains unchanged. It also defines new transport-layer feedback messages, including negative acknowledgements (NACKs), that can be used to report on specific congestion events. The use of the RTP/AVPF profile is dependent on signalling, but is otherwise generally backwards compatible with the RTP/AVP profile, as it keeps the same average RTCP reporting interval as the base RTP specification. The RTP Codec Control Messages [RFC5104] extend the RTP/AVPF profile with additional feedback messages that can be used to influence that way in which rate adaptation occurs. The dynamics of how rapidly feedback can be sent are unchanged.
- o Finally, Explicit Congestion Notification (ECN) for RTP over UDP [RFC6679] can be used to provide feedback on the number of packets that received an ECN Congestion Experienced (CE) mark. This RTCP

extension builds on the RTP/AVPF profile to allow rapid congestion feedback when ECN is supported.

In addition to these mechanisms for providing feedback, the sender can include an RTP header extension in each packet to record packet transmission times. There are two methods: [RFC5450] represents the transmission time in terms of a time-offset from the RTP timestamp of the packet, while [RFC6051] includes an explicit NTP-format sending timestamp (potentially more accurate, but a higher header overhead). Accurate sending timestamps can be helpful for estimating queuing delays, to get an early indication of the onset of congestion.

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

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

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

RTCP is a fundamental part of the RTP protocol, and the mechanisms described here rely on the implementation of RTCP. Implementations which claim to support RTP, but that do not implement RTCP, cannot use the circuit breaker mechanisms described in this memo. Such implementations SHOULD NOT be used on networks that might be subject to congestion unless equivalent mechanisms are defined using some non-RTCP feedback channel to report congestion and signal circuit breaker conditions.

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

1. The sender can estimate the network round-trip time once per RTCP reporting interval, based on the contents and timing of RTCP SR and RR packets.

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

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

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

Increased jitter can be a signal of transient network congestion, but in the highly aggregated form reported in RTCP RR packets, it offers insufficient information to estimate the extent or persistence of congestion. Jitter reports are a useful early warning of potential network congestion, but provide an insufficiently strong signal to be used as a circuit breaker.

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

Two packet loss regimes can be observed: 1) RTCP RR packets show a non-zero packet loss fraction, while the extended highest sequence number received continues to increment; and 2) RR packets show a loss fraction of zero, but the extended highest sequence number received

does not increment even though the sender has been transmitting RTP data packets. The former corresponds to the TCP congestion avoidance state, and indicates a congested path that is still delivering data; the latter corresponds to a TCP timeout, and is most likely due to a path failure. A third condition is that data is being sent but no RTCP feedback is received at all, corresponding to a failure of the reverse path. We derive circuit breaker conditions for these loss regimes in the following.

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

If RTP data packets are being sent, but the RTCP SR or RR packets reporting on that SSRC indicate a non-increasing extended highest sequence number received, this is an indication that those RTP data packets are not reaching the receiver. This could be a short-term issue affecting only a few packets, perhaps caused by a slow-to-open firewall or a transient connectivity problem, but if the issue persists, it is a sign of a more ongoing and significant problem. Accordingly, if a sender of RTP data packets receives two or more consecutive RTCP SR or RR packets from the same receiver, and those packets correspond to its transmission and have a non-increasing extended highest sequence number received field (i.e., the sender receives at least three RTCP SR or RR packets that report the same value in the extended highest sequence number received field for an SSRC, but the sender has sent RTP data packets for that SSRC that would have caused an increase in the reported value of the extended highest sequence number received if they had reached the receiver), then that sender SHOULD cease transmission (see Section 4.4).

The reason for waiting for two or more consecutive RTCP packets with a non-increasing extended highest sequence number is to give enough time for transient reception problems to resolve themselves, but to stop problem flows quickly enough to avoid causing serious ongoing network congestion. A single RTCP report showing no reception could be caused by a transient fault, and so will not cease transmission. Waiting for more than two consecutive RTCP reports before stopping a flow might avoid some false positives, but could lead to problematic flows running for a long time period (potentially tens of seconds, depending on the RTCP reporting interval) before being cut off.

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

In addition to media timeouts, as were discussed in Section 4.1, an RTP session has the possibility of an RTCP timeout. This can occur when RTP data packets are being sent, but there are no RTCP reports returned from the receiver. This is either due to a failure of the receiver to send RTCP reports, or a failure of the return path that is preventing those RTCP reporting from being delivered. In either

case, it is not safe to continue transmission, since the sender has no way of knowing if it is causing congestion. Accordingly, an RTP sender that has not received any RTCP SR or RTCP RR packets reporting on the SSRC it is using for three or more RTCP reporting intervals SHOULD cease transmission (see Section 4.4). When calculating the timeout, the fixed minimum RTCP reporting interval SHOULD be used (based on the rationale in Section 6.2 of RFC 3550 [RFC3550]).

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

4.3. RTP/AVP Circuit Breaker #3: Congestion

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

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

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

The phase "order of magnitude" in the above means within a factor of ten, approximately. In order to implement this, it is necessary to estimate the throughput a TCP connection would achieve over the path. For a long-lived TCP Reno connection, Padhye et al. [Padhye] showed that the throughput can be estimated using the following equation:

$$X = \frac{s}{R \cdot \sqrt{2 \cdot b \cdot p / 3} + (t_{\text{RTO}} \cdot (3 \cdot \sqrt{3 \cdot b \cdot p / 8} \cdot p \cdot (1 + 32 \cdot p^2)))}$$

where:

X is the transmit rate in bytes/second.

s is the packet size in bytes. If data packets vary in size, then the average size is to be used.

R is the round trip time in seconds.

p is the loss event rate, between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted.

t_RTO is the TCP retransmission timeout value in seconds, approximated by setting t_RTO = 4 * R.

b is the number of packets acknowledged by a single TCP acknowledgement ([RFC3448] recommends the use of b=1 since many TCP implementations do not use delayed acknowledgements).

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

$$X = \frac{s}{R \cdot \sqrt{p \cdot 2 / 3}}$$

It is RECOMMENDED that this simplified throughput equation be used, since the reduction in accuracy is small, and it is much simpler to calculate than the full equation.

Given this TCP equation, two parameters need to be estimated and reported to the sender in order to calculate the throughput: the round trip time, R, and the loss event rate, p (the packet size, s, is known to the sender). The round trip time can be estimated from RTCP SR and RR packets. This is done too infrequently for accurate statistics, but is the best that can be done with the standard RTCP mechanisms.

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

The congestion circuit breaker is therefore: when a sender receives an RTCP SR or RR packet that contains a report block for an SSRC it is using, that sender has to check the fraction lost field in that report block to determine if there is a non-zero packet loss rate. If the fraction lost field is zero, then continue sending as normal. If the fraction lost is greater than zero, then estimate the TCP throughput using the simplified equation above, and the measured R , p (approximated by the fraction lost), and s . Compare this with the actual sending rate. If the actual sending rate is more than ten times the estimated sending rate derived from the TCP throughput equation for two consecutive RTCP reporting intervals, the sender SHOULD cease transmission (see Section 4.4). If the RTP sender is using a reduced minimum RTCP reporting interval (as specified in Section 6.2 of RFC 3550 [RFC3550] or the RTP/AVPF profile [RFC4585]), then that reduced RTCP reporting interval is used when determining if the circuit breaker is triggered, since that interval scales with the media data rate.

Systems that usually send at a high data rate, but that can reduce their data rate significantly (i.e., by at least a factor of ten), MAY first reduce their sending rate to this lower value to see if this resolves the congestion, but MUST then cease transmission if the problem does not resolve itself within a further two RTCP reporting intervals (see Section 4.4). An example of this might be a video conferencing system that backs off to sending audio only, before completely dropping the call. If such a reduction in sending rate resolves the congestion problem, the sender MAY gradually increase the rate at which it sends data after a reasonable amount of time has passed, provided it takes care not to cause the problem to recur ("reasonable" is intentionally not defined here).

As in Section 4.1, we use two reporting intervals to avoid triggering the circuit breaker on transient failures. This circuit breaker is a worst-case condition, and congestion control needs to be performed to keep well within this bound. It is expected that the circuit breaker will only be triggered if the usual congestion control fails for some

reason.

4.4. Ceasing Transmission

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

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

5. RTP Circuit Breakers for Systems Using the RTP/AVPF Profile

Use of the Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [RFC4585] allows receivers to send early RTCP reports in some cases, to inform the sender about particular events in the media stream. There are several use cases for such early RTCP reports, including providing rapid feedback to a sender about the onset of congestion.

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

If the extension for Reduced-Size RTCP [RFC5506] is not used, early RTCP feedback packets sent according to the RTP/AVPF profile will be compound RTCP packets that include an RTCP SR/RR packet. That RTCP SR/RR packet MUST be processed as if it were sent as a regular RTCP

report and counted towards the circuit breaker conditions specified in Section 4 of this memo. This will potentially make the RTP circuit breaker fire earlier than it would if the RTP/AVPF profile was not used.

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

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

These rules are intended to allow the use of low-overhead early RTP/AVPF feedback for generic NACK messages without triggering the RTP circuit breaker. This is expected to make such feedback suitable for RTP congestion control algorithms that need to quickly report loss events in between regular RTCP reports. The reaction to reduced-size RTCP SR/RR packets is to allow such algorithms to send feedback that can trigger the circuit breaker, when desired.

6. Impact of RTCP XR

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

7. Impact of Explicit Congestion Notification (ECN)

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

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

8. Security Considerations

The security considerations of [RFC3550] apply.

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

If non-authenticated RTCP reports are used, an on-path attacker can trivially generate fake RTCP packets that indicate high packet loss rates, causing the circuit breaker to trigger and disrupting an RTP session. This is somewhat more difficult for an off-path attacker, due to the need to guess the randomly chosen RTP SSRC value and the RTP sequence number. This attack can be avoided if RTCP packets are authenticated, for example using the Secure RTP profile [RFC3711].

9. IANA Considerations

There are no actions for IANA.

10. Acknowledgements

The authors would like to thank Bernard Aboba, Harald Alvestrand, Kevin Gross, Cullen Jennings, Randell Jesup, Jonathan Lennox, Matt Mathis, Stephen McQuistin, Eric Rescorla, and Abheek Saha for their valuable feedback.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 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.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

11.2. Informative References

- [Floyd] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", Proc. ACM SIGCOMM 2000, DOI 10.1145/347059.347397, August 2000.
- [I-D.ietf-xrblock-rtcp-xr-burst-gap-discard] Clark, A., Huang, R., and W. Wu, "RTP Control Protocol(RTCP) Extended Report (XR) Block for Burst/Gap Discard metric Reporting", draft-ietf-xrblock-rtcp-xr-burst-gap-discard-10 (work in progress), January 2013.
- [I-D.ietf-xrblock-rtcp-xr-burst-gap-loss] Clark, A., Zhang, S., Zhao, J., and W. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Burst/Gap Loss metric Reporting", draft-ietf-xrblock-rtcp-xr-burst-gap-loss-08 (work in progress), January 2013.

- [I-D.ietf-xrblock-rtcp-xr-discard]
Clark, A., Zorn, G., and W. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Discard Count metric Reporting", draft-ietf-xrblock-rtcp-xr-discard-11 (work in progress), December 2012.
- [I-D.ietf-xrblock-rtcp-xr-discard-rle-metrics]
Ott, J., Singh, V., and I. Curcio, "RTP Control Protocol (RTCP) Extended Reports (XR) for Run Length Encoding (RLE) of Discarded Packets", draft-ietf-xrblock-rtcp-xr-discard-rle-metrics-05 (work in progress), December 2012.
- [Padhye] Padhye, J., Firoiu, V., Towsley, D., and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation", Proc. ACM SIGCOMM 1998, DOI 10.1145/285237.285291, August 1998.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.
- [RFC5450] Singer, D. and H. Desineni, "Transmission Time Offsets in RTP Streams", RFC 5450, March 2009.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.
- [RFC6051] Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", RFC 6051, November 2010.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, August 2012.
- [RFC6798] Clark, A. and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Packet Delay Variation Metric Reporting", RFC 6798, November 2012.

[RFC6843] Clark, A., Gross, K., and Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for Delay Metric Reporting", RFC 6843, January 2013.

Authors' Addresses

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

Email: csp@csperkins.org

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

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

AVTCORE
Internet-Draft
Updates: 3550 (if approved)
Intended status: Standards Track
Expires: August 29, 2013

J. Lennox
Vidyo
M. Westerlund
Ericsson
Q. Wu
Huawei
C. Perkins
University of Glasgow
February 25, 2013

RTP Considerations for Endpoints Sending Multiple Media Streams
draft-lennox-avtcore-rtp-multi-stream-02

Abstract

This document expands and clarifies the behavior of the Real-Time Transport Protocol (RTP) endpoints when they are sending multiple media streams in a single RTP session. In particular, issues involving Real-Time Transport Control Protocol (RTCP) messages are described.

This document updates RFC 3550 in regards to handling of multiple SSRCs per endpoint in RTP sessions.

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 August 29, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Use Cases For Multi-Stream Endpoints	3
3.1. Multiple-Capturer Endpoints	3
3.2. Multi-Media Sessions	4
3.3. Multi-Stream Mixers	4
4. Issue Cases	4
4.1. Cascaded Multi-party Conference with Source Projecting Mixers	5
5. Multi-Stream Endpoint RTP Media Recommendations	5
6. Multi-Stream Endpoint RTCP Recommendations	5
6.1. RTCP Reporting Requirement	6
6.2. Initial Reporting Interval	6
6.3. Compound RTCP Packets	6
7. RTCP Bandwidth Considerations for Sources with Disparate Rates	7
8. Grouping of RTCP Reception Statistics and Other Feedback . .	7
8.1. Semantics and Behavior of Reporting Groups	8
8.2. Determine the Report Group	9
8.3. RTCP Packet Reporting Group's Reporting Sources	9
8.4. RTCP Source Description (SDES) item for Reporting Groups	11
8.5. Middlebox Considerations	11
8.6. SDP signaling for Reporting Groups	11
8.7. Bandwidth Benefits of RTCP Reporting Groups	11
8.8. Consequences of RTCP Reporting Groups	12
9. Security Considerations	13
10. Open Issues	13
11. IANA Considerations	13
11.1. RTCP SDDES Item	13
11.2. RTCP Packet Type	14
12. References	14
12.1. Normative References	14
12.2. Informative References	14
Appendix A. Changes From Earlier Versions	15
A.1. Changes From Draft -01	15
A.2. Changes From Draft -00	16

Authors' Addresses	16
--------------------	----

1. Introduction

At the time The Real-Time Transport Protocol (RTP) [RFC3550] was originally written, and for quite some time after, endpoints in RTP sessions typically only transmitted a single media stream per RTP session, where separate RTP sessions were typically used for each distinct media type.

Recently, however, a number of scenarios have emerged (discussed further in Section 3) in which endpoints wish to send multiple RTP media streams, distinguished by distinct RTP synchronization source (SSRC) identifiers, in a single RTP session. Although RTP's initial design did consider such scenarios, the specification was not consistently written with such use cases in mind. The specifications are thus somewhat unclear.

The purpose of this document is to expand and clarify [RFC3550]'s language for these use cases. The authors believe this does not result in any major normative changes to the RTP specification, however this document defines how the RTP specification is to be interpreted. In these cases, this document updates RFC3550.

The document starts with terminology and some use cases where multiple sources will occur. This is followed by some case studies to try to identify issues that exist and need considerations. This is followed by RTP and RTCP recommendations to resolve issues. Next are security considerations and remaining open issues.

2. Terminology

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

3. Use Cases For Multi-Stream Endpoints

This section discusses several use cases that have motivated the development of endpoints that send multiple streams in a single RTP session.

3.1. Multiple-Capturer Endpoints

The most straightforward motivation for an endpoint to send multiple media streams in a session is the scenario where an endpoint has

multiple capture devices of the same media type and characteristics. For example, telepresence endpoints, of the type described by the CLUE Telepresence Framework [I-D.ietf-clue-framework] is designed, often have multiple cameras or microphones covering various areas of a room.

3.2. Multi-Media Sessions

Recent work has been done in RTP [I-D.ietf-avtcore-multi-media-rtp-session] and SDP [I-D.ietf-mmusic-sdp-bundle-negotiation] to update RTP's historical assumption that media streams of different media types would always be sent on different RTP sessions. In this work, a single endpoint's audio and video media streams (for example) are instead sent in a single RTP session.

3.3. Multi-Stream Mixers

There are several RTP topologies which can involve a central device that itself generates multiple media streams in a session.

One example is a mixer providing centralized compositing for a multi-capture scenario like that described in Section 3.1. In this case, the centralized node is behaving much like a multi-capturer endpoint, generating several similar and related sources.

More complicated is the Source Projecting Mixer, see Section 3.6 of [I-D.westerlund-avtcore-rtp-topologies-update]. This is a central box that receives media streams from several endpoints, and then selectively forwards modified versions of some of the streams toward the other endpoints it is connected to. Toward one destination, a separate media source appears in the session for every other source connected to the mixer, "projected" from the original streams, but at any given time many of them can appear to be inactive (and thus are receivers, not senders, in RTP). This sort of device is closer to being an RTP mixer than an RTP translator, in that it terminates RTCP reporting about the mixed streams, and it can re-write SSRCs, timestamps, and sequence numbers, as well as the contents of the RTP payloads, and can turn sources on and off at will without appearing to be generating packet loss. Each projected stream will typically preserve its original RTCP source description (SDS) information.

4. Issue Cases

This section illustrates some scenarios that have shown areas where the RTP specification is unclear.

4.1. Cascaded Multi-party Conference with Source Projecting Mixers

This issue case tries to illustrate the effect of having multiple SSRCs sent by an endpoint, by considering the traffic between two source-projecting mixers in a large multi-party conference.

For concreteness, consider a 200-person conference, where 16 sources are viewed at any given time. Assuming participants are distributed evenly among the mixers, each mixer would have 100 sources "behind" (projected through) it, of which at any given time eight are active senders. Thus, the RTP session between the mixers consists of two endpoints, but 200 sources.

The RTCP bandwidth implications of this scenario are discussed further in Section 8.7.

(TBD: Other examples? Can this section be removed?)

5. Multi-Stream Endpoint RTP Media Recommendations

While an endpoint MUST (of course) stay within its share of the available session bandwidth, as determined by signalling and congestion control, this need not be applied independently or uniformly to each media stream. In particular, session bandwidth MAY be reallocated among an endpoint's media streams, for example by varying the bandwidth use of a variable-rate codec, or changing the codec used by the media stream, up to the constraints of the session's negotiated (or declared) codecs. This includes enabling or disabling media streams as more or less bandwidth becomes available.

6. Multi-Stream Endpoint RTCP Recommendations

This section contains a number of different RTCP clarifications or recommendations that enables more efficient and simpler behavior without loss of functionality.

The RTP Control Protocol (RTCP) is defined in Section 6 of [RFC3550], but it is largely documented in terms of "participants". In many cases, the specification's recommendations for "participants" are to be interpreted as applying to individual media streams, rather than to endpoints. This section describes several concrete cases where this applies.

(tbd: rather than think in terms of media streams, it might be clearer to refer to SSRC values, where a participant with multiple active SSRC values counts as multiple participants, once per SSRC)

6.1. RTCP Reporting Requirement

For each of an endpoint's media streams, whether or not it is currently sending media, SR/RR and SD/ES packets MUST be sent at least once per RTCP report interval. (For discussion of the content of SR or RR packets' reception statistic reports, see Section 8.)

6.2. Initial Reporting Interval

When a new media stream is added to a unicast session, the sentence in [RFC3550]'s Section 6.2 applies: "For unicast sessions ... the delay before sending the initial compound RTCP packet MAY be zero." This applies to individual media sources as well. Thus, endpoints MAY send an initial RTCP packet for an SSRC immediately upon adding it to a unicast session.

This allowance also applies, as written, when initially joining a unicast session. However, in this case some caution needs to be exercised if the end-point or mixer has a large number of sources (SSRCs) as this can create a significant burst. How big an issue this depends on the number of source to send initial SR or RR and Session Description CNAME items for in relation to the RTCP bandwidth.

(tbd: Maybe some recommendation here? The aim in restricting this to unicast sessions was to avoid this burst of traffic, which the usual RTCP timing and reconsideration rules will prevent)

6.3. Compound RTCP Packets

Section 6.1 gives the following advice to RTP translators and mixers:

It is RECOMMENDED that translators and mixers combine individual RTCP packets from the multiple sources they are forwarding into one compound packet whenever feasible in order to amortize the packet overhead (see Section 7). An example RTCP compound packet as might be produced by a mixer is shown in Fig. 1. If the overall length of a compound packet would exceed the MTU of the network path, it SHOULD be segmented into multiple shorter compound packets to be transmitted in separate packets of the underlying protocol. This does not impair the RTCP bandwidth estimation because each compound packet represents at least one distinct participant. Note that each of the compound packets MUST begin with an SR or RR packet.

Note: To avoid confusion, an RTCP packet is an individual item, such as a Sender Report (SR), Receiver Report (RR), Source Description (SD/ES), Goodbye (BYE), Application Defined (APP), Feedback [RFC4585]

or Extended Report (XR) [RFC3611] packet. A compound packet is the combination of two or more such RTCP packets where the first packet has to be an SR or an RR packet, and which contains a SDES packet containing a CNAME item. Thus the above results in compound RTCP packets that contain multiple SR or RR packets from different sources as well as any of the other packet types. There are no restrictions on the order in which the packets can occur within the compound packet, except the regular compound rule, i.e., starting with an SR or RR.

This advice applies to multi-media-stream endpoints as well, with the same restrictions and considerations. (Note, however, that the last sentence does not apply to AVPF [RFC4585] or SAVPF [RFC5124] feedback packets if Reduced-Size RTCP [RFC5506] is in use.)

Due to RTCP's randomization of reporting times, there is a fair bit of tolerance in precisely when an endpoint schedules RTCP to be sent. Thus, one potential way of implementing this recommendation would be to randomize all of an endpoint's sources together, with a single randomization schedule, so an MTU's worth of RTCP all comes out simultaneously.

(tbd: Multiplexing RTCP packets from multiple different sources might require some adjustment to the calculation of RTCP's avg_rtcp_size, as the RTCP group interval is proportional to avg_rtcp_size times the group size).

7. RTCP Bandwidth Considerations for Sources with Disparate Rates

It is possible for an RTP session to carry sources of greatly differing bandwidths. One example is the scenario of [I-D.ietf-avtcore-multi-media-rtp-session], when audio and video are sent in the same session. However, this can occur even within a single media type, for example a video session carrying both 5 fps QCIF and 60 fps 1080p HD video, or an audio session carrying both G.729 and L24/48000/6 audio.

(tbd: recommend how RTCP bandwidths are to be chosen in these scenarios. Likely, these recommendations will be different for sessions using AVPF-based profiles (where the trr-int parameter is available) than for those using AVP.)

8. Grouping of RTCP Reception Statistics and Other Feedback

As specified by [RFC3550], an endpoint MUST send reception reports about every active media stream it is receiving, from at least one local source.

However, a naive application of the RTP specification's rules could be quite inefficient. In particular, if a session has N SSRCs (active and inactive, i.e., participant SSRCs), and the session has S active senders in each reporting interval, there will either be $N \times S$ report blocks per reporting interval, or (per the round-robin recommendations of [RFC3550] Section 6.1) reception sources would be unnecessarily round-robbined. In a session where most media sources become senders reasonably frequently, this results in quadratically many reception report blocks in the conference, or reporting delays proportional to the number of session members.

Since traffic is received by endpoints, however, rather than by media sources, there is not actually any need for this quadratic expansion. All that is needed is for each endpoint to report all the remote sources it is receiving.

Thus, this document defines a new RTCP mechanism, Reporting Groups, to indicate sources which originate from the same endpoint, and which therefore would have identical reception reports.

8.1. Semantics and Behavior of Reporting Groups

An RTCP Reporting Group indicates that a set of sources (SSRCs) that originate from a single entity (endpoint or middlebox) in an RTP session, and therefore all the sources in the group's view of the network is identical. If reporting groups are in use, two sources SHOULD be put into the same reporting group if their view of the network is identical; i.e., if they report on traffic received at the same interface of an RTP endpoint. Sources with different views of the network MUST NOT be put into the same reporting group.

If reporting groups are in use, an endpoint MUST NOT send reception reports from one source in a reporting group about another one in the same group ("self-reports"). Similarly, an endpoint MUST NOT send reception reports about a remote media source from more than one source in a reporting group ("cross-reports"). Instead, it MUST pick one of its local media sources as the "reporting" source for each remote media source, and use it to send reception reports about that remote source; all the other media sources in the reporting group MUST NOT send any reception reports about that remote media source.

This reporting source MUST also be the source for any RTP/AVPF Feedback [RFC4585] or Extended Report (XR) [RFC3611] packets about the corresponding remote sources as well. If a reporting source leaves the session (i.e., if it sends a BYE, or leaves the group without sending BYE under the rules of [RFC3550] section 6.3.7), another reporting source MUST be chosen if any members of the group still exist.

An endpoint or middlebox MAY use multiple sources as reporting sources; however, each reporting source MUST have non-overlapping sets of remote SSRCs it reports on. This is primarily to be done when the reporting source's number of reception report blocks is so large that it would be forced to round robin around the sources. Thus, by splitting the reports among several reporting SSRCs more consistent reporting can be achieved.

If RTP/AVPF feedback is in use, a reporting source MAY send immediate or early feedback at any point when any member of the reporting group could validly do so.

An endpoint SHOULD NOT create single-source reporting groups, unless it is anticipated that the group might have additional sources added to it in the future.

8.2. Determine the Report Group

A remote RTP entity, such as an endpoint or a middlebox needs to be able to determine the report group used by another RTP entity. To achieve this goal two RTCP extensions has been defined. For the SSRCs that are reporting on behalf of the reporting group an SDES item RGRP has been defined for providing the report group with an identifier. For SSRCs that aren't reporting on any peer SSRC a new RTCP packet type is defined. This RTCP packet type "Reporting Sources", lists the SSRC that are reporting on this SSRC's behalf.

This divided approach has been selected for the following reasons:

1. Enable an explicit indication of who reports on this SSRC's behalf. Being explicit prevents the remote entity from detecting that is missing the reports if there issues with the reporting SSRC's RTCP packets.
2. Enable explicit identification of the SSRCs that are actively reporting as one entity.

8.3. RTCP Packet Reporting Group's Reporting Sources

This section defines a new RTCP packet type called "Reporting Group's Reporting Sources" (RGRS). It identifies the SSRC(s) that report on behalf of the SSRC that is the sender of the RGRS packet.

This packet consists of the fixed RTCP packet header which indicates the packet type, the number of reporting sources included and the SSRC which behalf the reporting SSRCs report on. This is followed by the list of reporting SSRCs.

Any RTP mixer or translator which forwards SR or RR packets from members of a reporting group MUST forward the corresponding RGRS RTCP packet as well.

8.4. RTCP Source Description (SDES) item for Reporting Groups

A new RTCP Source Description (SDES) item is defined for the purpose of identifying reporting groups.

The Source Description (SDES) item "RGRP" is sent by a reporting group's reporting SSRC. Syntactically, its format is the same as the RTCP SDES CNAME item [RFC6222], and MUST be chosen with the same global-uniqueness and privacy considerations as CNAME. This name MUST be stable across the lifetime of the reporting group, even if the SSRC of a reporting source changes.

Every source which belongs to a reporting group MUST either include an RGRP SDES item in an SDES packet (if it is a reporting source), or an RGRS packet (if it is not), in every compound RTCP packet in which it sends an RR or SR packet (i.e., in every RTCP packet it sends, unless Reduced-Size RTCP [RFC5506] is in use).

Any RTP mixer or translator which forwards SR or RR packets from members of a reporting group MUST forward the corresponding RGRP SDES items as well, even if it otherwise strips SDES items other than CNAME.

8.5. Middlebox Considerations

This section discusses middlebox considerations for Reporting groups.

To be expanded.

8.6. SDP signaling for Reporting Groups

TBD

8.7. Bandwidth Benefits of RTCP Reporting Groups

To understand the benefits of RTCP reporting groups, consider the scenario described in Section 4.1. This scenario describes an environment in which the two endpoints in a session each have a hundred sources, of which eight each are sending within any given reporting interval.

For ease of analysis, we can make the simplifying approximation that the duration of the RTCP reporting interval is equal to the total size of the RTCP packets sent during an RTCP interval, divided by the

RTCP bandwidth. (This will be approximately true in scenarios where the bandwidth is not so high that the minimum RTCP interval is reached.) For further simplification, we can assume RTCP senders are following the recommendations of Section 6.3; thus, the per-packet transport-layer overhead will be small relative to the RTCP data. Thus, only the actual RTCP data itself need be considered.

In a report interval in this scenario, there will, as a baseline, be 200 SDES packets, 184 RR packets, and 16 SR packets. This amounts to approximately 6.5 kB of RTCP per report interval, assuming 16-byte CNAMEs and no other SDES information.

Using the original [RFC3550] everyone-reports-on-every-sender feedback rules, each of the 184 receivers will send 16 report blocks, and each of the 16 senders will send 15. This amounts to approximately 76 kB of report block traffic per interval; 92% of RTCP traffic consists of report blocks.

If reporting groups are used, however, there is only 0.4 kB of reports per interval, with no loss of useful information. Additionally, there will be (assuming 16-byte RGRPs, and a single reporting source per reporting group) an additional 2.4 kB per cycle of RGRP SDES items and RGRS packets. Put another way, the unmodified [RFC3550] reporting interval is approximately 8.9 times longer than if reporting groups are in use.

8.8. Consequences of RTCP Reporting Groups

The RTCP traffic generated by receivers using RTCP Reporting Groups might appear, to observers unaware of these semantics, to be generated by receivers who are experiencing a network disconnection, as the non-reporting sources appear not to be receiving a given sender at all.

This could be a potentially critical problem for such a sender using RTCP for congestion control, as such a sender might think that it is sending so much traffic that it is causing complete congestion collapse.

However, such an interpretation of the session statistics would require a fairly sophisticated RTCP analysis. Any receiver of RTCP statistics which is just interested in information about itself needs to be prepared that any given reception report might not contain information about a specific media source, because reception reports in large conferences can be round-robin.

Thus, it is unclear to what extent such backward compatibility issues would actually cause trouble in practice.

9. Security Considerations

In the secure RTP protocol (SRTP) [RFC3711], the cryptographic context of a compound SRTCP packet is the SSRC of the sender of the first RTCP (sub-)packet. This could matter in some cases, especially for keying mechanisms such as Mikey [RFC3830] which use per-SSRC keying.

Other than that, the standard security considerations of RTP apply; sending multiple media streams from a single endpoint does not appear to have different security consequences than sending the same number of streams.

10. Open Issues

At this stage this document contains a number of open issues. The below list tries to summarize the issues:

1. Further clarifications on how to handle the RTCP scheduler when sending multiple sources in one compound packet.
2. How is the use of reporting groups be signaled in SDP?
3. How is the RTCP avg_rtcp_size be calculated when RTCP packets are routinely multiplexed among multiple RTCP senders?
4. Do we need to provide a recommendation for unicast session joiners with many sources to not use 0 initial minimal interval from bit-rate burst perspective?

11. IANA Considerations

This document make several requests to IANA for registering new RTP/RTCP identifiers.

(Note to the RFC-Editor: please replace "TBA" with the IANA-assigned value, and "XXXX" with the number of this document, prior to publication as an RFC.)

11.1. RTCP SDES Item

This document adds an additional SDES types to the IANA "RTCP SDES Item Types" Registry, as follows:

Value	Abbrev	Name	Reference
TBA	RGRP	Reporting Group	[RFCXXXX]

Figure 1: Item for the IANA Source Attribute Registry

11.2. RTCP Packet Type

This document defines one new RTCP Control Packet types (PT) to be registered as follows:

Value	Abbrev	Name	Reference
TBA	RGRR	Reporting Group Reporting Sources	[RFCXXXX]

Figure 2: Item for the IANA RTCP Control Packet Types (PT) Registry

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.
- [RFC6222] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.

12.2. Informative References

- [I-D.ietf-avtcore-multi-media-rtp-session] Westerlund, M., Perkins, C., and J. Lennox, "Multiple Media Types in an RTP Session", draft-ietf-avtcore-multi-media-rtp-session-01 (work in progress), October 2012.

[I-D.ietf-clue-framework]

Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-09 (work in progress), February 2013.

[I-D.ietf-mmusic-sdp-bundle-negotiation]

Holmberg, C., Alvestrand, H., and C. Jennings, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-bundle-negotiation-03 (work in progress), February 2013.

[I-D.westerlund-avtcore-rtp-topologies-update]

Westerlund, M. and S. Wenger, "RTP Topologies", draft-westerlund-avtcore-rtp-topologies-update-01 (work in progress), October 2012.

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

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

Appendix A. Changes From Earlier Versions

Note to the RFC-Editor: please remove this section prior to publication as an RFC.

A.1. Changes From Draft -01

- o Merged with draft-wu-avtcore-multisrc-endpoint-adver.
- o Changed how Reporting Groups are indicated in RTCP, to make it clear which source(s) is the group's reporting sources.
- o Clarified the rules for when sources can be placed in the same reporting group.
- o Clarified that mixers and translators need to pass reporting group SDES information if they are forwarding RR and SR traffic from members of a reporting group.

A.2. Changes From Draft -00

- o Added the Reporting Group semantic to explicitly indicate which sources come from a single endpoint, rather than leaving it implicit.
- o Specified that Reporting Group semantics (as they now are) apply to AVPF and XR, as well as to RR/SR report blocks.
- o Added a description of the cascaded source-projecting mixer, along with a calculation of its RTCP overhead if reporting groups are not in use.
- o Gave some guidance on how the flexibility of RTCP randomization allows some freedom in RTCP multiplexing.
- o Clarified the language of several of the recommendations.
- o Added an open issue discussing how avg_rtcp_size ought to be calculated for multiplexed RTCP.
- o Added an open issue discussing how RTCP bandwidths are to be chosen for sessions where source bandwidths greatly differ.

Authors' Addresses

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

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

Email: csp@csp Perkins.org