Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 21, 2016

J. Lennox
Vidyo
K. Gross
AVA
S. Nandakumar
G. Salgueiro
Cisco Systems
B. Burman, Ed.
Ericsson

July 20, 2015

A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources

draft-ietf-avtext-rtp-grouping-taxonomy-08

Abstract

The terminology about, and associations among, Real-Time Transport Protocol (RTP) sources can be complex and somewhat opaque. This document describes a number of existing and proposed properties and relationships among RTP sources, and defines common terminology for discussing protocol entities and their relationships.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The existing taxonomy of sources in the Real-Time Transport Protocol (RTP) [RFC3550] has previously been regarded as confusing and inconsistent. Consequently, a deep understanding of how the different terms relate to each other becomes a real challenge. Frequently cited examples of this confusion are (1) how different protocols that make use of RTP use the same terms to signify different things and (2) how the complexities addressed at one layer are often glossed over or ignored at another.

This document improves clarity by reviewing the semantics of various aspects of sources in RTP. As an organizing mechanism, it approaches this by describing various ways that RTP sources are transformed on their way between sender and receiver, and how they can be grouped and associated together.

All non-specific references to ControLling mUltiple streams for tElepresence (CLUE) in this document map to [I-D.ietf-clue-framework] and all references to Web Real-Time Communications (WebRTC) map to [I-D.ietf-rtcweb-overview].

2. Concepts

This section defines concepts that serve to identify and name various transformations and streams in a given RTP usage. For each concept, alternate definitions and usages that co-exist today are listed along with various characteristics that further describes the concept. These concepts are divided into two categories, one related to the

chain of streams and transformations that media can be subject to, the other for entities involved in the communication.

2.1. Media Chain

In the context of this document, Media is a sequence of synthetic or Physical Stimuli (Section 2.1.1) (sound waves, photons, key-strokes), represented in digital form. Synthesized Media is typically generated directly in the digital domain.

This section contains the concepts that can be involved in taking Media at a sender side and transporting it to a receiver, which may recover a sequence of physical stimuli. This chain of concepts is of two main types, streams and transformations. Streams are time-based sequences of samples of the physical stimulus in various representations, while transformations changes the representation of the streams in some way.

The below examples are basic ones and it is important to keep in mind that this conceptual model enables more complex usages. Some will be further discussed in later sections of this document. In general the following applies to this model:

- A transformation may have zero or more inputs and one or more outputs.

- A stream is of some type, such as audio, video, real-time text, etc.

- A stream has one source transformation and one or more sink transformations (with the exception of Physical Stimulus (Section 2.1.1) that may lack source or sink transformation).

- Streams can be forwarded from a transformation output to any number of inputs on other transformations that support that type.

- If the output of a transformation is sent to multiple transformations, those streams will be identical; it takes a transformation to make them different.

- There are no formal limitations on how streams are connected to transformations.

It is also important to remember that this is a conceptual model. Thus real-world implementations may look different and have different structure.

To provide a basic understanding of the relationships in the chain we first introduce the concepts for the sender side (Figure 1). This covers physical stimuli until media packets are emitted onto the network.

```
               Physical Stimulus
                      |
                      V
           +----------------------+
           |     Media Capture    |
           +----------------------+
                      |
                 Raw Stream
                      V
           +----------------------+
           |     Media Source     |<- Synchronization Timing
           +----------------------+
                      |
                Source Stream
                      V
           +----------------------+
           |    Media Encoder     |
           +----------------------+
                      |
                Encoded Stream      +------------+
                      V             |            V
           +----------------------+ | +----------------------+
           |   Media Packetizer   | | | RTP-based Redundancy |
           +----------------------+ | +----------------------+
                      |             |            |
                      +-------------+  Redundancy RTP Stream
               Source RTP Stream                 |
                      V                          V
           +----------------------+   +----------------------+
           |  RTP-based Security  |   |  RTP-based Security  |
           +----------------------+   +----------------------+
                      |                          |
              Secured RTP Stream   Secured Redundancy RTP Stream
                      V                          V
           +----------------------+   +----------------------+
           |   Media Transport    |   |   Media Transport    |
           +----------------------+   +----------------------+
```

Figure 1: Sender Side Concepts in the Media Chain

In Figure 1 we have included a branched chain to cover the concepts for using redundancy to improve the reliability of the transport.

The Media Transport concept is an aggregate that is decomposed in Section 2.1.15.

In Figure 2 we review a receiver media chain matching the sender side, to look at the inverse transformations and their attempts to recover identical streams as in the sender chain, subject to what may be lossy compression and imperfect Media Transport. Note that the streams out of a reverse transformation, like the Source Stream out the Media Decoder are in many cases not the same as the corresponding ones on the sender side, thus they are prefixed with a "Received" to denote a potentially modified version. The reason for not being the same lies in the transformations that can be of irreversible type. For example, lossy source coding in the Media Encoder prevents the Source Stream out of the Media Decoder to be the same as the one fed into the Media Encoder. Other reasons include packet loss or late loss in the Media Transport transformation that even RTP-based Repair, if used, fails to repair. However, some transformations are not always present, like RTP-based Repair that cannot operate without Redundancy RTP Streams.

```
          +----------------------+   +----------------------+
          |   Media Transport    |   |   Media Transport    |
          +----------------------+   +----------------------+
            Received |                 Received | Secured
            Secured RTP Stream       Redundancy RTP Stream
                     V                          V
          +----------------------+   +----------------------+
          | RTP-based Validation |   | RTP-based Validation |
          +----------------------+   +----------------------+
                     |                          |
            Received RTP Stream   Received Redundancy RTP Stream
                     |                          |
                     |     +--------------------+
                     V     V
          +----------------------+
          |   RTP-based Repair   |
          +----------------------+
                     |
            Repaired RTP Stream
                     V
          +----------------------+
          |  Media Depacketizer  |
          +----------------------+
                     |
           Received Encoded Stream
                     V
          +----------------------+
          |    Media Decoder     |
          +----------------------+
                     |
           Received Source Stream
                     V
          +----------------------+
          |      Media Sink      |--> Synchronization Information
          +----------------------+
                     |
            Received Raw Stream
                     V
          +----------------------+
          |    Media Renderer    |
          +----------------------+
                     |
                     V
             Physical Stimulus
```

Figure 2: Receiver Side Concepts of the Media Chain

2.1.1. Physical Stimulus

The Physical Stimulus is a physical event in the analog domain that can be sampled and converted to digital form by an appropriate sensor or transducer. This include sound waves making up audio, photons in a light field, or other excitations or interactions with sensors, like keystrokes on a keyboard.

2.1.2. Media Capture

Media Capture is the process of transforming the analog Physical Stimulus (Section 2.1.1) into digital Media using an appropriate sensor or transducer. The Media Capture performs a digital sampling of the physical stimulus, usually periodically, and outputs this in some representation as a Raw Stream (Section 2.1.3). This data is considered "Media", because it includes data that is periodically sampled, or made up of a set of timed asynchronous events. The Media Capture is normally instantiated in some type of device, i.e. media capture device. Examples of different types of media capturing devices are digital cameras, microphones connected to A/D converters, or keyboards.

Characteristics:

- A Media Capture is identified either by hardware/manufacturer ID or via a session-scoped device identifier as mandated by the application usage.

- A Media Capture can generate an Encoded Stream (Section 2.1.7) if the capture device supports such a configuration.

- The nature of the Media Capture may impose constraints on the clock handling in some of the subsequent steps. For example, many audio or video capture devices are not completely free in selecting the sample rate.

2.1.3. Raw Stream

A Raw Stream is the time progressing stream of digitally sampled information, usually periodically sampled and provided by a Media Capture (Section 2.1.2). A Raw Stream can also contain synthesized Media that may not require any explicit Media Capture, since it is already in an appropriate digital form.

2.1.4. Media Source

A Media Source is the logical source of a time progressing digital media stream synchronized to a reference clock. This stream is called a Source Stream (Section 2.1.5). This transformation takes one or more Raw Streams (Section 2.1.3) and provides a Source Stream as output. The output is synchronized with a reference clock (Section 3.1), which can be as simple as a system local wall clock or as complex as an NTP synchronized clock.

The output can be of different types. One type is directly associated with a particular Media Capture's Raw Stream. Others are more conceptual sources, like an audio mix of multiple Source Streams (Figure 3). Mixing multiple streams typically requires that the input streams are possible to relate in time, meaning that they have to be Source Streams (Section 2.1.5) rather than Raw Streams. In Figure 3, the generated Source Stream is a mix of the three input Source Streams.

```
                Source    Source    Source
                Stream    Stream    Stream
                  |         |         |
                  V         V         V
              +--------------------------+
              |        Media Source      |<-- Reference Clock
              |           Mixer          |
              +--------------------------+
                            |
                            V
                      Source Stream
```

Figure 3: Conceptual Media Source in form of Audio Mixer

Another possible example of a conceptual Media Source is a video surveillance switch, where the input is multiple Source Streams from different cameras, and the output is one of those Source Streams based on some selection criteria, like a round-robin or based on some video activity measure.

2.1.5. Source Stream

A Source Stream is a stream of digital samples that has been synchronized with a reference clock and comes from particular Media Source (Section 2.1.4).

2.1.6. Media Encoder

A Media Encoder is a transform that is responsible for encoding the media data from a Source Stream (Section 2.1.5) into another representation, usually more compact, that is output as an Encoded Stream (Section 2.1.7).

The Media Encoder step commonly includes pre-encoding transformations, such as scaling, resampling etc. The Media Encoder can have a significant number of configuration options that affects the properties of the Encoded Stream. This include properties such as codec, bit-rate, start points for decoding, resolution, bandwidth or other fidelity affecting properties.

Scalable Media Encoders need special attention as they produce multiple outputs that are potentially of different types. As shown in Figure 4, a scalable Media Encoder takes one input Source Stream and encodes it into multiple output streams of two different types; at least one Encoded Stream that is independently decodable and one or more Dependent Streams (Section 2.1.8). Decoding requires at least one Encoded Stream and zero or more Dependent Streams. A Dependent Stream's dependency is one of the grouping relations this document discusses further in Section 3.7.

```
                           Source Stream
                                 |
                                 V
                       +--------------------------+
                       |  Scalable Media Encoder  |
                       +--------------------------+
                          |         |   ...    |
                          V         V          V
                       Encoded  Dependent  Dependent
                       Stream    Stream     Stream
```

Figure 4: Scalable Media Encoder Input and Outputs

There are also other variants of encoders, like so-called Multiple Description Coding (MDC). Such Media Encoders produce multiple independent and thus individually decodable Encoded Streams. However, (logically) combining multiple of these Encoded Streams into a single Received Source Stream during decoding leads to an improvement in perceptual reproduced quality when compared to decoding a single Encoded Stream.

Creating multiple Encoded Streams from the same Source Stream, where the Encoded Streams are neither in a scalable nor in an MDC

relationship is commonly utilized in Simulcast [I-D.ietf-mmusic-sdp-simulcast] environments.

2.1.7. Encoded Stream

A stream of time synchronized encoded media that can be independently decoded.

Due to temporal dependencies, an Encoded Stream may have limitations in where decoding can be started. These entry points, for example Intra frames from a video encoder, may require identification and their generation may be event based or configured to occur periodically.

2.1.8. Dependent Stream

A stream of time synchronized encoded media fragments that are dependent on one or more Encoded Streams (Section 2.1.7) and zero or more Dependent Streams to be possible to decode.

Each Dependent Stream has a set of dependencies. These dependencies must be understood by the parties in a Multimedia Session that intend to use a Dependent Stream.

2.1.9. Media Packetizer

The transformation of taking one or more Encoded (Section 2.1.7) or Dependent Streams (Section 2.1.8) and putting their content into one or more sequences of packets, normally RTP packets, and output Source RTP Streams (Section 2.1.10). This step includes both generating RTP payloads as well as RTP packets. The Media Packetizer then selects which Synchronization source(s) (SSRC) [RFC3550] and RTP Sessions to use.

The Media Packetizer can combine multiple Encoded or Dependent Streams into one or more RTP Streams:

o The Media Packetizer can use multiple inputs when producing a single RTP Stream. One such example is SRST packetization when using Scalable Video Coding (SVC) (Section 3.7).

o The Media Packetizer can also produce multiple RTP Streams, for example when Encoded and/or Dependent Streams are distributed over multiple RTP Streams. One example of this is MRMT packetization when using SVC (Section 3.7).

2.1.10. RTP Stream

An RTP Stream is a stream of RTP packets containing media data, source or redundant. The RTP Stream is identified by an SSRC belonging to a particular RTP Session. The RTP Session is identified as discussed in Section 2.2.2.

A Source RTP Stream is an RTP Stream directly related to an Encoded Stream (Section 2.1.7), targeted for transport over RTP without any additional RTP-based Redundancy (Section 2.1.11) applied.

Characteristics:

- Each RTP Stream is identified by a Synchronization source (SSRC) [RFC3550] that is carried in every RTP and RTP Control Protocol (RTCP) packet header. The SSRC is unique in a specific RTP Session context.

- At any given point in time, a RTP Stream can have one and only one SSRC, but SSRCs for a given RTP Stream can change over time. SSRC collision and clock rate change [RFC7160] are examples of valid reasons to change SSRC for an RTP Stream. In those cases, the RTP Stream itself is not changed in any significant way, only the identifying SSRC number.

- Each SSRC defines a unique RTP sequence numbering and timing space.

- Several RTP Streams, each with their own SSRC, may represent a single Media Source.

- Several RTP Streams, each with their own SSRC, can be carried in a single RTP Session.

2.1.11. RTP-based Redundancy

RTP-based Redundancy is defined here as a transformation that generates redundant or repair packets sent out as a Redundancy RTP Stream (Section 2.1.12) to mitigate network transport impairments, like packet loss and delay. Note that this excludes the type of redundancy that most suitable Media Encoders (Section 2.1.6) may add to the media format of the Encoded Stream (Section 2.1.7) that makes it cope better with inevitable RTP packet losses.

The RTP-based Redundancy exists in many flavors; they may be generating independent Repair Streams that are used in addition to the Source Stream (like RTP Retransmission (Section 3.10) and some special types of Forward Error Correction, like RTP stream

duplication (Section 3.8)), they may generate a new Source Stream by combining redundancy information with source information (Using XOR FEC (Section 3.11) as a redundancy payload (Section 3.9)), or completely replace the source information with only redundancy packets.

2.1.12. Redundancy RTP Stream

A Redundancy RTP Stream is an RTP Stream (Section 2.1.10) that contains no original source data, only redundant data, which may either be used standalone or be combined with one or more Received RTP Streams (Section 2.1.23) to produce Repaired RTP Streams (Section 2.1.26).

2.1.13. RTP-based Security

The optional RTP-based Security transformation applies security services such as authentication, integrity protection and confidentiality to an input RTP Stream, like what is specified in The Secure Real-time Transport Protocol (SRTP) [RFC3711], producing a Secured RTP Stream (Section 2.1.14). Either an RTP Stream (Section 2.1.10) or a Redundancy RTP Stream (Section 2.1.12) can be used as input to this transformation.

In SRTP and the related Secure RTCP (SRTCP), all of the above mentioned security services are optional, except for integrity protection of SRTCP, which is mandatory. Also confidentiality (encryption) is effectively optional in SRTP, since it is possible to use a NULL encryption algorithm. As described in [RFC7201], the strength of SRTP data origin authentication depends on the cryptographic transform and key management used, for example in group communication where it is sometimes possible to authenticate group membership but not the actual RTP Stream sender.

RTP-based Security and RTP-based Redundancy can be combined in a few different ways. One way is depicted in Figure 1, where an RTP Stream and its corresponding Redundancy RTP Stream are protected by separate RTP-based Security transforms. In other cases, like when a Media Translator is adding FEC in Section 3.2.1.3 of [I-D.ietf-avtcore-rtp-topologies-update], a middlebox can apply RTP-based Redundancy to an already Secured RTP Stream instead of a Source RTP Stream. One example of that is depicted in Figure 5 below.

```
               Source RTP Stream    +------------+
                      V             |            V
           +----------------------+ | +----------------------+
           |  RTP-based Security  | | | RTP-based Redundancy |
           +----------------------+ | +----------------------+
                      |             |            |
                      |             |  Redundancy RTP Stream
                      +-------------+            |
                      |                          V
                      |                 +----------------------+
              Secured RTP Stream        |  RTP-based Security  |
                      |                 +----------------------+
                      |                          |
                      |            Secured Redundancy RTP Stream
                      V                          V
           +----------------------+   +----------------------+
           |   Media Transport    |   |   Media Transport    |
           +----------------------+   +----------------------+
```

Figure 5: Adding Redundancy to a Secured RTP Stream

In this case, the Redundancy RTP Stream may already have been secured for confidentiality (encrypted) by the first RTP-based Security, and it may therefore not be necessary to apply additional confidentiality protection in the second RTP-based Security. To avoid attacks and negative impact on RTP-based Repair (Section 2.1.25) and the resulting Repaired RTP Stream (Section 2.1.26), it is however still necessary to have this second RTP-based Security apply both authentication and integrity protection to the Redundancy RTP Stream.

2.1.14. Secured RTP Stream

A Secured RTP Stream is a Source or Redundancy RTP Stream that is protected through RTP-based Security (Section 2.1.13) by one or more of the confidentiality, integrity, or authentication security services.

2.1.15. Media Transport

A Media Transport defines the transformation that the RTP Streams (Section 2.1.10) are subjected to by the end-to-end transport from one RTP sender to one specific RTP receiver (an RTP Session (Section 2.2.2) may contain multiple RTP receivers per sender). Each Media Transport is defined by a transport association that is normally identified by a 5-tuple (source address, source port, destination address, destination port, transport protocol), but a proposal exists for sending multiple transport associations on a single 5-tuple [I-D.westerlund-avtcore-transport-multiplexing].

Characteristics:

- Media Transport transmits RTP Streams of RTP Packets from a source transport address to a destination transport address.

- Each Media Transport contains only a single RTP Session.

- A single RTP Session can span multiple Media Transports.

The Media Transport concept sometimes needs to be decomposed into more steps to enable discussion of what a sender emits that gets transformed by the network before it is received by the receiver. Thus we provide also this Media Transport decomposition (Figure 6).

```
                               RTP Stream
                                    |
                                    V
                       +--------------------------+
                       |  Media Transport Sender  |
                       +--------------------------+
                                    |
                             Sent RTP Stream
                                    V
                       +--------------------------+
                       |    Network Transport     |
                       +--------------------------+
                                    |
                         Transported RTP Stream
                                    V
                       +--------------------------+
                       | Media Transport Receiver |
                       +--------------------------+
                                    |
                                    V
                           Received RTP Stream
```

Figure 6: Decomposition of Media Transport

2.1.16. Media Transport Sender

The first transformation within the Media Transport (Section 2.1.15) is the Media Transport Sender. The sending Endpoint (Section 2.2.1) takes an RTP Stream and emits the packets onto the network using the transport association established for this Media Transport, thereby creating a Sent RTP Stream (Section 2.1.17). In the process, it transforms the RTP Stream in several ways. First, it generates the necessary protocol headers for the transport association, for example IP and UDP headers, thus forming IP/UDP/RTP packets. In addition,

the Media Transport Sender may queue, intentionally pace or otherwise affect how the packets are emitted onto the network, thereby potentially introducing delay and delay variations [RFC5481] that characterize the Sent RTP Stream.

2.1.17. Sent RTP Stream

The Sent RTP Stream is the RTP Stream as entering the first hop of the network path to its destination. The Sent RTP Stream is identified using network transport addresses, like for IP/UDP the 5-tuple (source IP address, source port, destination IP address, destination port, and protocol (UDP)).

2.1.18. Network Transport

Network Transport is the transformation that subjects the Sent RTP Stream (Section 2.1.17) to traveling from the source to the destination through the network. This transformation can result in loss of some packets, delay and delay variation on a per packet basis, packet duplication, and packet header or data corruption. This transformation produces a Transported RTP Stream (Section 2.1.19) at the exit of the network path.

2.1.19. Transported RTP Stream

The Transported RTP Stream is the RTP Stream that is emitted out of the network path at the destination, subjected to the Network Transport's transformation (Section 2.1.18).

2.1.20. Media Transport Receiver

The Media Transport Receiver is the receiver Endpoint's (Section 2.2.1) transformation of the Transported RTP Stream (Section 2.1.19) by its reception process, which results in the Received RTP Stream (Section 2.1.23). This transformation includes transport checksums being verified. Sensible system designs typically either discard packets with mis-matching checksums, or pass them on while somehow marking them in the resulting Received RTP Stream so to alert subsequent transformations about the possible corrupt state. In this context it is worth noting that there is typically some probability for corrupt packets to pass through undetected (with a seemingly correct checksum). Other transformations can compensate for delay variations in receiving a packet on the network interface and providing it to the application (de-jitter buffer).

2.1.21. Received Secured RTP Stream

This is the Secured RTP Stream (Section 2.1.14) resulting from the Media Transport (Section 2.1.15) aggregate transformation.

2.1.22. RTP-based Validation

RTP-based Validation is the reverse transformation of RTP-based Security (Section 2.1.13). If this transformation fails, the result is either not usable and must be discarded, or may be usable but cannot be trusted. If the transformation succeeds, the result can be a Received RTP Stream (Section 2.1.23) or a Received Redundancy RTP Stream (Section 2.1.24), depending on what was input to the corresponding RTP-based Security transformation, but can also be a Received Secured RTP Stream (Section 2.1.21) in case several RTP-based Security transformations were applied.

2.1.23. Received RTP Stream

The Received RTP Stream is the RTP Stream (Section 2.1.10) resulting from the Media Transport's aggregate transformation (Section 2.1.15), i.e. subjected to packet loss, packet corruption, packet duplication, delay, and delay variation from sender to receiver.

2.1.24. Received Redundancy RTP Stream

The Received Redundancy RTP Stream is the Redundancy RTP Stream (Section 2.1.12) resulting from the Media Transport transformation, i.e. subjected to packet loss, packet corruption, delay, and delay variation from sender to receiver.

2.1.25. RTP-based Repair

RTP-based Repair is a Transformation that takes as input zero or more Received RTP Streams (Section 2.1.23) and one or more Received Redundancy RTP Streams (Section 2.1.24), and produces one or more Repaired RTP Streams (Section 2.1.26) that are as close to the corresponding sent Source RTP Streams (Section 2.1.10) as possible, using different RTP-based repair methods, for example the ones referred in RTP-based Redundancy (Section 2.1.11).

2.1.26. Repaired RTP Stream

A Repaired RTP Stream is a Received RTP Stream (Section 2.1.23) for which Received Redundancy RTP Stream (Section 2.1.24) information has been used to try to recover the Source RTP Stream (Section 2.1.10) as it was before Media Transport (Section 2.1.15).

2.1.27. Media Depacketizer

A Media Depacketizer takes one or more RTP Streams (Section 2.1.10), depacketizes them, and attempts to reconstitute the Encoded Streams (Section 2.1.7) or Dependent Streams (Section 2.1.8) present in those RTP Streams.

In practical implementations, the Media Depacketizer and the Media Decoder may be tightly coupled and share information to improve or optimize the overall decoding and error concealment process. It is, however, not expected that there would be any benefit in defining a taxonomy for those detailed (and likely very implementation-dependent) steps.

2.1.28. Received Encoded Stream

The Received Encoded Stream is the received version of an Encoded Stream (Section 2.1.7).

2.1.29. Media Decoder

A Media Decoder is a transformation that is responsible for decoding Encoded Streams (Section 2.1.7) and any Dependent Streams (Section 2.1.8) into a Source Stream (Section 2.1.5).

In practical implementations, the Media Decoder and the Media Depacketizer may be tightly coupled and share information to improve or optimize the overall decoding process in various ways. It is however not expected that there would be any benefit in defining a taxonomy for those detailed (and likely very implementation-dependent) steps.

A Media Decoder has to deal with any errors in the Encoded Streams that resulted from corruption or failure to repair packet losses. Therefore, it commonly is robust to error and losses, and includes concealment methods.

2.1.30. Received Source Stream

The Received Source Stream is the received version of a Source Stream (Section 2.1.5).

2.1.31. Media Sink

The Media Sink receives a Source Stream (Section 2.1.5) that contains, usually periodically, sampled media data together with associated synchronization information. Depending on application, this Source Stream then needs to be transformed into a Raw Stream

(Section 2.1.3) that is conveyed to the Media Render (Section 2.1.33), synchronized with the output from other Media Sinks. The Media Sink may also be connected with a Media Source (Section 2.1.4) and be used as part of a conceptual Media Source.

The Media Sink can further transform the Source Stream into a representation that is suitable for rendering on the Media Render as defined by the application or system-wide configuration. This include sample scaling, level adjustments etc.

2.1.32. Received Raw Stream

The Received Raw Stream is the received version of a Raw Stream (Section 2.1.3).

2.1.33. Media Render

A Media Render takes a Raw Stream (Section 2.1.3) and converts it into Physical Stimulus (Section 2.1.1) that a human user can perceive. Examples of such devices are screens, and D/A converters connected to amplifiers and loudspeakers.

An Endpoint can potentially have multiple Media Renders for each media type.

2.2. Communication Entities

This section contains concepts for entities involved in the communication.

```
      +------------------------------------------------------------+
      | Communication Session                                      |
      |                                                            |
      | +----------------+                      +----------------+ |
      | | Participant A  |    +------------+    | Participant B  | |
      | |                |    | Multimedia |    |                | |
      | | +------------+ |<==>| Session    |<==>| +------------+ | |
      | | | Endpoint A | |    |            |    | | Endpoint B | | |
      | | |            | |    +------------+    | |            | | |
      | | | +----------+-+----------------------+-+----------+ | | |
      | | | | RTP      | |                      | |          | | | |
      | | | | Session  |-+---Media Transport----+>|          | | | |
      | | | | Audio    |<+---Media Transport----+-|          | | | |
      | | | |          | |          ^           | |          | | | |
      | | | +----------+-+----------|-----------+-+----------+ | | |
      | | |            | |          v           | |            | | |
      | | |            | | +-----------------+  | |            | | |
      | | |            | | | Synchronization |  | |            | | |
      | | |            | | |     Context     |  | |            | | |
      | | |            | | +-----------------+  | |            | | |
      | | |            | |          ^           | |            | | |
      | | | +----------+-+----------|-----------+-+----------+ | | |
      | | | | RTP      | |          v           | |          | | | |
      | | | | Session  |<+---Media Transport----+-|          | | | |
      | | | | Video    |-+---Media Transport----+>|          | | | |
      | | | |          | |                      | |          | | | |
      | | | +----------+-+----------------------+-+----------+ | | |
      | | +------------+ |                      | +------------+ | |
      | +----------------+                      +----------------+ |
      +------------------------------------------------------------+
```

Figure 7: Example Point to Point Communication Session with two RTP Sessions

Figure 7 shows a high-level example representation of a very basic point-to-point Communication Session between Participants A and B. It uses two different audio and video RTP Sessions between A's and B's Endpoints, where each RTP Session is a group communications channel that can potentially carry a number of RTP Streams. It is using separate Media Transports for those RTP Sessions. The Multimedia Session shared by the Participants can, for example, be established using SIP (i.e., there is a SIP Dialog between A and B). The terms used in Figure 7 are further elaborated in the sub-sections below.

2.2.1. Endpoint

An Endpoint is a single addressable entity sending or receiving RTP packets. It may be decomposed into several functional blocks, but as long as it behaves as a single RTP stack entity it is classified as a single "Endpoint".

Characteristics:

- Endpoints can be identified in several different ways. While RTCP Canonical Names (CNAMEs) [RFC3550] provide a globally unique and stable identification mechanism for the duration of the Communication Session (see Section 2.2.5), their validity applies exclusively within a Synchronization Context (Section 3.1). Thus one Endpoint can handle multiple CNAMEs, each of which can be shared among a set of Endpoints belonging to the same Participant (Section 2.2.3). Therefore, mechanisms outside the scope of RTP, such as application defined mechanisms, must be used to provide Endpoint identification when outside this Synchronization Context.

- An Endpoint can be associated with at most one Participant (Section 2.2.3) at any single point in time.

- In some contexts, an Endpoint would typically correspond to a single "host", for example a computer using a single network interface and being used by a single human user. In other contexts, a single "host" can serve multiple Participants, in which case each Participant's Endpoint may share properties, for example the IP address part of a transport address.

2.2.2. RTP Session

An RTP Session is an association among a group of Participants communicating with RTP. It is a group communications channel which can potentially carry a number of RTP Streams. Within an RTP Session, every Participant can find meta-data and control information (over RTCP) about all the RTP Streams in the RTP Session. The bandwidth of the RTCP control channel is shared between all Participants within an RTP Session.

Characteristics:

- An RTP Session can carry one ore more RTP Streams.

- An RTP Session shares a single SSRC space as defined in RFC3550 [RFC3550]. That is, the Endpoints participating in an RTP Session can see an SSRC identifier transmitted by any of the other Endpoints. An Endpoint can receive an SSRC either as SSRC or as a

Contributing source (CSRC) in RTP and RTCP packets, as defined by the Endpoints' network interconnection topology.

- An RTP Session uses at least two Media Transports (Section 2.1.15), one for sending and one for receiving. Commonly, the receiving Media Transport is the reverse direction of the Media Transport used for sending. An RTP Session may use many Media Transports and these define the session's network interconnection topology.

- A single Media Transport always carries a single RTP Session.

- Multiple RTP Sessions can be conceptually related, for example originating from or targeted for the same Participant (Section 2.2.3) or Endpoint (Section 2.2.1), or by containing RTP Streams that are somehow related (Section 3).

2.2.3. Participant

A Participant is an entity reachable by a single signaling address, and is thus related more to the signaling context than to the media context.

Characteristics:

- A single signaling-addressable entity, using an application-specific signaling address space, for example a SIP URI.

- A Participant can participate in several Multimedia Sessions (Section 2.2.4).

- A Participant can be comprised of several associated Endpoints (Section 2.2.1).

2.2.4. Multimedia Session

A Multimedia Session is an association among a group of Participants (Section 2.2.3) engaged in the communication via one or more RTP Sessions (Section 2.2.2). It defines logical relationships among Media Sources (Section 2.1.4) that appear in multiple RTP Sessions.

Characteristics:

- A Multimedia Session can be composed of several RTP Sessions with potentially multiple RTP Streams per RTP Session.

- Each Participant in a Multimedia Session can have a multitude of Media Captures and Media Rendering devices.

- A single Multimedia Session can contain media from one or more Synchronization Contexts (Section 3.1). An example of that is a Multimedia Session containing one set of audio and video for communication purposes belonging to one Synchronization Context, and another set of audio and video for presentation purposes (like playing a video file) with a separate Synchronization Context that has no strong timing relationship and need not be strictly synchronized with the audio and video used for communication.

2.2.5. Communication Session

A Communication Session is an association among two or more Participants (Section 2.2.3) communicating with each other via one or more Multimedia Sessions (Section 2.2.4).

Characteristics:

- Each Participant in a Communication Session is identified via an application-specific signaling address.

- A Communication Session is composed of Participants that share at least one Multimedia Session, involving one or more parallel RTP Sessions with potentially multiple RTP Streams per RTP Session.

For example, in a full mesh communication, the Communication Session consists of a set of separate Multimedia Sessions between each pair of Participants. Another example is a centralized conference, where the Communication Session consists of a set of Multimedia Sessions between each Participant and the conference handler.

3. Concepts of Inter-Relations

This section uses the concepts from previous sections, and looks at different types of relationships among them. These relationships occur at different abstraction levels and for different purposes, but the reason for the needed relationship at a certain step in the media handling chain may exist at another step. For example, the use of Simulcast (Section 3.6)) implies a need to determine relations at RTP Stream level, but the underlying reason is that multiple Media Encoders use the same Media Source, i.e. to be able to identify a common Media Source.

3.1. Synchronization Context

A Synchronization Context defines a requirement on a strong timing relationship between the Media Sources, typically requiring alignment of clock sources. Such a relationship can be identified in multiple ways as listed below. A single Media Source can only belong to a

single Synchronization Context, since it is assumed that a single Media Source can only have a single media clock and requiring alignment to several Synchronization Contexts (and thus reference clocks) will effectively merge those into a single Synchronization Context.

3.1.1. RTCP CNAME

RFC3550 [RFC3550] describes Inter-media synchronization between RTP Sessions based on RTCP CNAME, RTP and Network Time Protocol (NTP) [RFC5905] formatted timestamps of a reference clock. As indicated in [RFC7273], despite using NTP format timestamps, it is not required that the clock be synchronized to an NTP source.

3.1.2. Clock Source Signaling

[RFC7273] provides a mechanism to signal the clock source in Session Description Protocol (SDP) [RFC4566] both for the reference clock as well as the media clock, thus allowing a Synchronization Context to be defined beyond the one defined by the usage of CNAME source descriptions.

3.1.3. Implicitly via RtcMediaStream

WebRTC defines "RtcMediaStream" with one or more "RtcMediaStreamTracks". All tracks in a "RtcMediaStream" are intended to be synchronized when rendered, implying that they must be generated such that synchronization is possible.

3.1.4. Explicitly via SDP Mechanisms

The SDP Grouping Framework [RFC5888] defines an m= line (Section 4.2) grouping mechanism called "Lip Synchronization" (with LS identification-tag) for establishing the synchronization requirement across m= lines when they map to individual sources.

Source-Specific Media Attributes in SDP [RFC5576] extends the above mechanism when multiple Media Sources are described by a single m= line.

3.2. Endpoint

Some applications requires knowledge of what Media Sources originate from a particular Endpoint (Section 2.2.1). This can include such decisions as packet routing between parts of the topology, knowing the Endpoint origin of the RTP Streams.

In RTP, this identification has been overloaded with the Synchronization Context (Section 3.1) through the usage of the RTCP source description CNAME (Section 3.1.1). This works for some usages, but in others it breaks down. For example, if an Endpoint has two sets of Media Sources that have different Synchronization Contexts, like the audio and video of the human Participant as well as a set of Media Sources of audio and video for a shared movie, CNAME would not be an appropriate identification for that Endpoint. Therefore, an Endpoint may have multiple CNAMEs. The CNAMEs or the Media Sources themselves can be related to the Endpoint.

3.3. Participant

In communication scenarios, it is commonly needed to know which Media Sources originate from which Participant (Section 2.2.3). One reason is, for example, to enable the application to display Participant Identity information correctly associated with the Media Sources. This association is handled through the signaling solution to point at a specific Multimedia Session where the Media Sources may be explicitly or implicitly tied to a particular Endpoint.

Participant information becomes more problematic due to Media Sources that are generated through mixing or other conceptual processing of Raw Streams or Source Streams that originate from different Participants. This type of Media Sources can thus have a dynamically varying set of origins and Participants. RTP contains the concept of CSRC that carry information about the previous step origin of the included media content on RTP level.

3.4. RtcMediaStream

An RtcMediaStream in WebRTC is an explicit grouping of a set of Media Sources (RtcMediaStreamTracks) that share a common identifier and a single Synchronization Context (Section 3.1).

3.5. Multi-Channel Audio

There exist a number of RTP payload formats that can carry multi-channel audio, despite the codec being a single-channel (mono) encoder. Multi-channel audio can be viewed as multiple Media Sources sharing a common Synchronization Context. These are independently encoded by a Media Encoder and the different Encoded Streams are packetized together in a time synchronized way into a single Source RTP Stream, using the used codec's RTP Payload format. Examples of codecs that support multi-channel audio are PCMA and PCMU [RFC3551], AMR [RFC4867], and G.719 [RFC5404].

3.6. Simulcast

A Media Source represented as multiple independent Encoded Streams constitutes a Simulcast [I-D.ietf-mmusic-sdp-simulcast] or MDC of that Media Source. Figure 8 shows an example of a Media Source that is encoded into three separate Simulcast streams, that are in turn sent on the same Media Transport flow. When using Simulcast, the RTP Streams may be sharing RTP Session and Media Transport, or be separated on different RTP Sessions and Media Transports, or any combination of these two. One major reason to use separate Media Transports is to make use of different Quality of Service for the different Source RTP Streams. Some considerations on separating related RTP Streams are discussed in Section 3.12.

```
                            +----------------+
                            |  Media Source  |
                            +----------------+
                     Source Stream  |
             +----------------------+----------------------+
             |                      |                      |
             V                      V                      V
    +------------------+   +------------------+   +------------------+
    |  Media Encoder   |   |  Media Encoder   |   |  Media Encoder   |
    +------------------+   +------------------+   +------------------+
             | Encoded              | Encoded              | Encoded
             | Stream               | Stream               | Stream
             V                      V                      V
    +------------------+   +------------------+   +------------------+
    | Media Packetizer |   | Media Packetizer |   | Media Packetizer |
    +------------------+   +------------------+   +------------------+
             | Source               | Source               | Source
             | RTP                  | RTP                  | RTP
             | Stream               | Stream               | Stream
             +-----------------+    |    +-----------------+
                               |    |    |
                               V    V    V
                          +-------------------+
                          |  Media Transport  |
                          +-------------------+
```

Figure 8: Example of Media Source Simulcast

The Simulcast relation between the RTP Streams is the common Media Source. In addition, to be able to identify the common Media Source, a receiver of the RTP Stream may need to know which configuration or encoding goals that lay behind the produced Encoded Stream and its properties. This enables selection of the stream that is most useful in the application at that moment.

3.7. Layered Multi-Stream

Layered Multi-Stream (LMS) is a mechanism by which different portions of a layered or scalable encoding of a Source Stream are sent using separate RTP Streams (sometimes in separate RTP Sessions). LMSs are useful for receiver control of layered media.

A Media Source represented as an Encoded Stream and multiple Dependent Streams constitutes a Media Source that has layered dependencies. Figure 9 represents an example of a Media Source that is encoded into three dependent layers, where two layers are sent on the same Media Transport using different RTP Streams, i.e. SSRCs, and the third layer is sent on a separate Media Transport.

```
                            +----------------+
                            |  Media Source  |
                            +----------------+
                                    |
                                    |
                                    V
       +---------------------------------------------------------+
       |                      Media Encoder                      |
       +---------------------------------------------------------+
               |                    |                     |
        Encoded Stream       Dependent Stream     Dependent Stream
               |                    |                     |
               V                    V                     V
       +----------------+   +----------------+   +----------------+
       |Media Packetizer|   |Media Packetizer|   |Media Packetizer|
       +----------------+   +----------------+   +----------------+
               |                    |                     |
          RTP Stream           RTP Stream            RTP Stream
               |                    |                     |
               +------+      +------+                     |
                      |      |                            |
                      V      V                            V
                +-----------------+              +-----------------+
                | Media Transport |              | Media Transport |
                +-----------------+              +-----------------+
```

Figure 9: Example of Media Source Layered Dependency

It is sometimes useful to make a distinction between using a single Media Transport or multiple separate Media Transports when (in both cases) using multiple RTP Streams to carry Encoded Streams and Dependent Streams for a Media Source. Therefore, the following new terminology is defined here:

SRST: Single RTP Stream on a Single Media Transport

MRST: Multiple RTP Streams on a Single Media Transport

MRMT: Multiple RTP Streams on Multiple Media Transports

MRST and MRMT relations needs to identify the common Media Encoder origin for the Encoded and Dependent Streams. When using different RTP Sessions (MRMT), a single RTP Stream per Media Encoder, and a single Media Source in each RTP Session, common SSRC and CNAMEs can be used to identify the common Media Source. When multiple RTP Streams are sent from one Media Encoder in the same RTP Session (MRST), then CNAME is the only currently specified RTP identifier that can be used. In cases where multiple Media Encoders use multiple Media Sources sharing Synchronization Context, and thus having a common CNAME, additional heuristics or identification need to be applied to create the MRST or MRMT relationships between the RTP Streams.

3.8. RTP Stream Duplication

RTP Stream Duplication [RFC7198], using the same or different Media Transports, and optionally also delaying the duplicate [RFC7197], offers a simple way to protect media flows from packet loss in some cases (see Figure 10). This is a specific type of redundancy. All but one Source RTP Stream (Section 2.1.10) are effectively Redundancy RTP Streams (Section 2.1.12), but since both Source and Redundant RTP Streams are the same, it does not matter which one is which. This can also be seen as a specific type of Simulcast (Section 3.6) that transmits the same Encoded Stream (Section 2.1.7) multiple times.

```
                            +----------------+
                            |  Media Source  |
                            +----------------+
                     Source Stream  |
                                    V
                            +----------------+
                            | Media Encoder  |
                            +----------------+
                    Encoded Stream  |
                        +-----------+-----------+
                        |                       |
                        V                       V
               +------------------+    +------------------+
               | Media Packetizer |    | Media Packetizer |
               +------------------+    +------------------+
                 Source | RTP Stream     Source | RTP Stream
                        |                       V
                        |                +-------------+
                        |                | Delay (opt) |
                        |                +-------------+
                        |                       |
                        +-----------+-----------+
                                    |
                                    V
                          +-------------------+
                          |  Media Transport  |
                          +-------------------+
```

Figure 10: Example of RTP Stream Duplication

3.9. Redundancy Format

The RTP Payload for Redundant Audio Data [RFC2198] defines a transport for redundant audio data together with primary data in the same RTP payload. The redundant data can be a time delayed version of the primary or another time delayed Encoded Stream using a different Media Encoder to encode the same Media Source as the primary, as depicted in Figure 11.

```
              +--------------------+
              |    Media Source    |
              +--------------------+
                        |
                   Source Stream
                        |
                        +------------------------+
                        |                        |
                        V                        V
              +--------------------+   +--------------------+
              |   Media Encoder    |   |   Media Encoder    |
              +--------------------+   +--------------------+
                        |                        |
                        |                 +------------+
                  Encoded Stream          | Time Delay |
                        |                 +------------+
                        |                        |
                        |     +------------------+
                        V     V
              +--------------------+
              |  Media Packetizer  |
              +--------------------+
                        |
                        V
                   RTP Stream
```

Figure 11: Concept for usage of Audio Redundancy with different Media Encoders

The Redundancy format is thus providing the necessary meta information to correctly relate different parts of the same Encoded Stream. The case depicted above (Figure 11) relates the Received Source Stream fragments coming out of different Media Decoders, to be able to combine them together into a less erroneous Source Stream.

3.10. RTP Retransmission

Figure 12 shows an example where a Media Source's Source RTP Stream is protected by a retransmission (RTX) flow [RFC4588]. In this example the Source RTP Stream and the Redundancy RTP Stream share the same Media Transport.

```
          +--------------------+
          |    Media Source    |
          +--------------------+
                    |
                    V
          +--------------------+
          |   Media Encoder    |
          +--------------------+
                    |                              Retransmission
              Encoded Stream     +--------+     +---- Request
                    V            |        V     V
          +--------------------+ | +--------------------+
          |  Media Packetizer  | | | RTP Retransmission |
          +--------------------+ | +--------------------+
                    |            |           |
                    +------------+  Redundancy RTP Stream
             Source RTP Stream               |
                    |                        |
                    +---------+    +---------+
                              |    |
                              V    V
                       +-----------------+
                       | Media Transport |
                       +-----------------+
```

Figure 12: Example of Media Source Retransmission Flows

The RTP Retransmission example (Figure 12) illustrates that this mechanism works purely on the Source RTP Stream. The RTP Retransmission transform buffers the sent Source RTP Stream and, upon request, emits a retransmitted packet with an extra payload header as a Redundancy RTP Stream. The RTP Retransmission mechanism [RFC4588] is specified such that there is a one to one relation between the Source RTP Stream and the Redundancy RTP Stream. Therefore, a Redundancy RTP Stream needs to be associated with its Source RTP Stream. This is done based on CNAME selectors and heuristics to match requested packets for a given Source RTP Stream with the original sequence number in the payload of any new Redundancy RTP Stream using the RTX payload format. In cases where the Redundancy RTP Stream is sent in a different RTP Session than the Source RTP Stream, the RTP Session relation is signaled by using the SDP Media Grouping's [RFC5888] Flow Identification (FID identification-tag) semantics.

3.11. Forward Error Correction

Figure 13 shows an example where two Media Sources' Source RTP Streams are protected by Forward Error Correction (FEC). Source RTP Stream A has a RTP-based Redundancy transformation in FEC Encoder 1. This produces a Redundancy RTP Stream 1, that is only related to Source RTP Stream A. The FEC Encoder 2, however, takes two Source RTP Streams (A and B) and produces a Redundancy RTP Stream 2 that protects them jointly, i.e. Redundancy RTP Stream 2 relates to two Source RTP Streams (a FEC group). FEC decoding, when needed due to packet loss or packet corruption at the receiver, requires knowledge about which Source RTP Streams that the FEC encoding was based on.

In Figure 13 all RTP Streams are sent on the same Media Transport. This is however not the only possible choice. Numerous combinations exist for spreading these RTP Streams over different Media Transports to achieve the communication application's goal.

```
       +--------------------+                +--------------------+
       |   Media Source A   |                |   Media Source B   |
       +--------------------+                +--------------------+
                 |                                     |
                 V                                     V
       +--------------------+                +--------------------+
       |   Media Encoder A  |                |   Media Encoder B  |
       +--------------------+                +--------------------+
                 |                                     |
           Encoded Stream                        Encoded Stream
                 V                                     V
       +--------------------+                +--------------------+
       | Media Packetizer A |                | Media Packetizer B |
       +--------------------+                +--------------------+
                 |                                     |
        Source RTP Stream A                   Source RTP Stream B
                 |                                     |
           +-----+---------+-------------+         +---+---+
           |               V             V         V       |
           |       +---------------+  +---------------+    |
           |       | FEC Encoder 1 |  | FEC Encoder 2 |    |
           |       +---------------+  +---------------+    |
           |  Redundancy   |     Redundancy   |            |
           |  RTP Stream 1 |     RTP Stream 2 |            |
           V               V                  V            V
       +----------------------------------------------------------+
       |                    Media Transport                       |
       +----------------------------------------------------------+
```

Figure 13: Example of FEC Redundancy RTP Streams

As FEC Encoding exists in various forms, the methods for relating FEC Redundancy RTP Streams with its source information in Source RTP Streams are many. The XOR based RTP FEC Payload format [RFC5109] is defined in such a way that a Redundancy RTP Stream has a one to one relation with a Source RTP Stream. In fact, the RFC requires the Redundancy RTP Stream to use the same SSRC as the Source RTP Stream. This requires the use of either a separate RTP Session, or the Redundancy RTP Payload format [RFC2198]. The underlying relation requirement for this FEC format and a particular Redundancy RTP Stream is to know the related Source RTP Stream, including its SSRC.

3.12. RTP Stream Separation

RTP Streams can be separated exclusively based on their SSRCs, at the RTP Session level, or at the Multi-Media Session level.

When the RTP Streams that have a relationship are all sent in the same RTP Session and are uniquely identified based on their SSRC only, it is termed an SSRC-Only Based Separation. Such streams can be related via RTCP CNAME to identify that the streams belong to the same Endpoint. SSRC-based approaches [RFC5576], when used, can explicitly relate various such RTP Streams.

On the other hand, when RTP Streams that are related are sent in the context of different RTP Sessions to achieve separation, it is known as RTP Session-based separation. This is commonly used when the different RTP Streams are intended for different Media Transports.

Several mechanisms that use RTP Session-based separation rely on it to enable an implicit grouping mechanism expressing the relationship. The solutions have been based on using the same SSRC value in the different RTP Sessions to implicitly indicate their relation. That way, no explicit RTP level mechanism has been needed, only signaling level relations have been established using semantics from Grouping of Media lines framework [RFC5888]. Examples of this are RTP Retransmission [RFC4588], SVC Multi-Session Transmission [RFC6190] and XOR Based FEC [RFC5109]. RTCP CNAME explicitly relates RTP Streams across different RTP Sessions, as explained in the previous section. Such a relationship can be used to perform inter-media synchronization.

RTP Streams that are related and need to be associated can be part of different Multimedia Sessions, rather than just different RTP Sessions within the same Multimedia Session context. This puts further demand on the scope of the mechanism(s) and its handling of identifiers used for expressing the relationships.

3.13. Multiple RTP Sessions over one Media Transport

[I-D.westerlund-avtcore-transport-multiplexing] describes a mechanism that allows several RTP Sessions to be carried over a single underlying Media Transport. The main reasons for doing this are related to the impact of using one or more Media Transports (using a common network path or potentially have different ones). The fewer Media Transports used, the less need for NAT/FW traversal resources and smaller number of flow based Quality of Service (QoS).

However, Multiple RTP Sessions over one Media Transport imply that a single Media Transport 5-tuple is not sufficient to express in which RTP Session context a particular RTP Stream exists. Complexities in the relationship between Media Transports and RTP Session already exist as one RTP Session contains multiple Media Transports, e.g. even a Peer-to-Peer RTP Session with RTP/RTCP Multiplexing requires two Media Transports, one in each direction. The relationship between Media Transports and RTP Sessions as well as additional levels of identifiers need to be considered in both signaling design and when defining terminology.

4. Mapping from Existing Terms

This section describes a selected set of terms from some relevant IETF RFC and Internet Drafts (at the time of writing), using the concepts from previous sections.

4.1. Telepresence Terms

The terms in this sub-section are used in the context of CLUE [I-D.ietf-clue-framework]. Note that some terms listed in this sub-section use the same names as terms defined elsewhere in this document. Unless explicitly stated (as "RTP Taxonomy") and in this sub-section, they are to be read as references to the CLUE-specific term within this sub-section.

4.1.1. Audio Capture

Defined in CLUE as a Media Capture (Section 4.1.7) for audio. Describes an audio Media Source (Section 2.1.4).

4.1.2. Capture Device

Defined in CLUE as a device that converts physical input into an electrical signal. Identifies a physical entity performing an RTP Taxonomy Media Capture (Section 2.1.2) transformation.

4.1.3. Capture Encoding

Defined in CLUE as a specific encoding (Section 4.1.6) of a Media Capture (Section 4.1.7). Describes an Encoded Stream (Section 2.1.7) related to CLUE specific semantic information.

4.1.4. Capture Scene

Defined in CLUE as a structure representing a spatial region captured by one or more Capture Devices (Section 4.1.2), each capturing media representing a portion of the region. Describes a set of spatially related Media Sources (Section 2.1.4).

4.1.5. Endpoint

Defined in CLUE as a CLUE-capable device which is the logical point of final termination through receiving, decoding and rendering and/or initiation through capturing, encoding, and sending of media streams (Section 4.1.10). CLUE further defines it to consist of one or more physical devices with source and sink media streams, and exactly one [RFC4353] Participant. Describes exactly one Participant (Section 2.2.3) and one or more RTP Taxonomy Endpoints (Section 2.2.1).

4.1.6. Individual Encoding

Defined in CLUE as a set of parameters representing a way to encode a Media Capture (Section 4.1.7) to become a Capture Encoding (Section 4.1.3). Describes the configuration information needed to perform a Media Encoder (Section 2.1.6) transformation.

4.1.7. Media Capture

Defined in CLUE as a source of media, such as from one or more Capture Devices (Section 4.1.2) or constructed from other media streams (Section 4.1.10). Describes either an RTP Taxonomy Media Capture (Section 2.1.2) or a Media Source (Section 2.1.4), depending on in which context the term is used.

4.1.8. Media Consumer

Defined in CLUE as a CLUE-capable device that intends to receive Capture Encodings (Section 4.1.3). Describes the media receiving part of an RTP Taxonomy Endpoint (Section 2.2.1).

4.1.9. Media Provider

Defined in CLUE as a CLUE-capable device that intends to send Capture Encodings (Section 4.1.3). Describes the media sending part of an RTP Taxonomy Endpoint (Section 2.2.1).

4.1.10. Stream

Defined in CLUE as a Capture Encoding (Section 4.1.3) sent from a Media Provider (Section 4.1.9) to a Media Consumer (Section 4.1.8) via RTP. Describes an RTP Stream (Section 2.1.10).

4.1.11. Video Capture

Defined in CLUE as a Media Capture (Section 4.1.7) for video. Describes a video Media Source (Section 2.1.4).

4.2. Media Description

A single Session Description Protocol (SDP) [RFC4566] media description (or media block; an m-line and all subsequent lines until the next m-line or the end of the SDP) describes part of the necessary configuration and identification information needed for a Media Encoder transformation, as well as the necessary configuration and identification information for the Media Decoder to be able to correctly interpret a received RTP Stream.

A Media Description typically relates to a single Media Source. This is for example an explicit restriction in WebRTC. However, nothing prevents that the same Media Description (and same RTP Session) is re-used for multiple Media Sources [I-D.ietf-avtcore-rtp-multi-stream]. It can thus describe properties of one or more RTP Streams, and can also describe properties valid for an entire RTP Session (via [RFC5576] mechanisms, for example).

4.3. Media Stream

RTP [RFC3550] uses media stream, audio stream, video stream, and stream of (RTP) packets interchangeably, which are all RTP Streams.

4.4. Multimedia Conference

A Multimedia Conference is a Communication Session (Section 2.2.5) between two or more Participants (Section 2.2.3), along with the software they are using to communicate.

4.5. Multimedia Session

SDP [RFC4566] defines a Multimedia Session as a set of multimedia senders and receivers and the data streams flowing from senders to receivers, which would correspond to a set of Endpoints and the RTP Streams that flow between them. In this document, Multimedia Session (Section 2.2.4) also assumes those Endpoints belong to a set of Participants that are engaged in communication via a set of related RTP Streams.

RTP [RFC3550] defines a Multimedia Session as a set of concurrent RTP Sessions among a common group of Participants. For example, a video conference may contain an audio RTP Session and a video RTP Session. This would correspond to a group of Participants (each using one or more Endpoints) sharing a set of concurrent RTP Sessions. In this document, Multimedia Session also defines those RTP Sessions to have some relation and be part of a communication among the Participants.

4.6. Multipoint Control Unit (MCU)

This term is commonly used to describe the central node in any type of star topology [I-D.ietf-avtcore-rtp-topologies-update] conference. It describes a device that includes one Participant (Section 2.2.3) (usually corresponding to a so-called conference focus) and one or more related Endpoints (Section 2.2.1) (sometimes one or more per conference Participant).

4.7. Multi-Session Transmission (MST)

One of two transmission modes defined in H.264 based SVC [RFC6190], the other mode being SST (Section 4.13). In Multi-Session Transmission (MST), the SVC Media Encoder sends Encoded Streams and Dependent Streams distributed across two or more RTP Streams in one or more RTP Sessions. The term "MST" is ambiguous in RFC 6190, especially since the name indicates the use of multiple "sessions", while MST type packetization is in fact required whenever two or more RTP Streams are used for the Encoded and Dependent Streams, regardless if those are sent in one or more RTP Sessions. Corresponds either to MRST or MRMT (Section 3.7) stream relations defined in this document. The SVC RTP Payload RFC [RFC6190] is not particularly explicit about how the common Media Encoder (Section 2.1.6) relation between Encoded Streams (Section 2.1.7) and Dependent Streams (Section 2.1.8) is to be implemented.

4.8. Recording Device

WebRTC specifications use this term to refer to locally available entities performing a Media Capture (Section 2.1.2) transformation.

4.9. RtcMediaStream

A WebRTC RtcMediaStream is a set of Media Sources (Section 2.1.4) sharing the same Synchronization Context (Section 3.1).

4.10. RtcMediaStreamTrack

A WebRTC RtcMediaStreamTrack is a Media Source (Section 2.1.4).

4.11. RTP Sender

RTP [RFC3550] uses this term, which can be seen as the RTP protocol part of a Media Packetizer (Section 2.1.9).

4.12. RTP Session

Within the context of SDP, a singe m= line can map to a single RTP Session (Section 2.2.2) or multiple m= lines can map to a single RTP Session. The latter is enabled via multiplexing schemes such as BUNDLE [I-D.ietf-mmusic-sdp-bundle-negotiation], for example, which allows mapping of multiple m= lines to a single RTP Session.

4.13. Single Session Transmission (SST)

One of two transmission modes defined in H.264 based SVC [RFC6190], the other mode being MST (Section 4.7). In Single Session Transmission (SST), the SVC Media Encoder sends Encoded Streams (Section 2.1.7) and Dependent Streams (Section 2.1.8) combined into a single RTP Stream (Section 2.1.10) in a single RTP Session (Section 2.2.2), using the SVC RTP Payload format. The term "SST" is ambiguous in RFC 6190, in that it sometimes refers to the use of a single RTP Stream, like in sections relating to packetization, and sometimes appears to refer to use of a single RTP Session, like in the context of discussing SDP. Closely corresponds to SRST (Section 3.7) defined in this document.

4.14. SSRC

RTP [RFC3550] defines this as "the source of a stream of RTP packets", which indicates that an SSRC is not only a unique identifier for the Encoded Stream (Section 2.1.7) carried in those packets, but is also effectively used as a term to denote a Media Packetizer (Section 2.1.9). In [RFC3550], it is stated that "a

synchronization source may change its data format, e.g., audio encoding, over time". The related Encoded Stream data format in an RTP Stream (Section 2.1.10) is identified by the RTP Payload Type. Changing data format for an Encoded Stream effectively also changes what Media Encoder (Section 2.1.6) that is used for the Encoded Stream. No ambiguity is introduced to SSRC as Encoded Stream identifier by allowing RTP Payload Type changes, as long as only a single RTP Payload Type is valid for any given RTP Time Stamp. This is aligned with and further described by Section 5.2 of [RFC3550].

5. Security Considerations

The purpose of this document is to make clarifications and reduce the confusion prevalent in RTP taxonomy because of inconsistent usage by multiple technologies and protocols making use of the RTP protocol. It does not introduce any new security considerations beyond those already well documented in the RTP protocol [RFC3550] and each of the many respective specifications of the various protocols making use of it.

Having a well-defined common terminology and understanding of the complexities of the RTP architecture will help lead us to better standards, avoiding security problems.

6. Acknowledgement

This document has many concepts borrowed from several documents such as WebRTC [I-D.ietf-rtcweb-overview], CLUE [I-D.ietf-clue-framework], and Multiplexing Architecture [I-D.westerlund-avtcore-transport-multiplexing]. The authors would like to thank all the authors of each of those documents.

The authors would also like to acknowledge the insights, guidance and contributions of Magnus Westerlund, Roni Even, Paul Kyzivat, Colin Perkins, Keith Drage, Harald Alvestrand, Alex Eleftheriadis, Mo Zanaty, Stephan Wenger, and Bernard Aboba.

7. Contributors

Magnus Westerlund has contributed the concept model for the media chain using transformations and streams model, including rewriting pre-existing concepts into this model and adding missing concepts. The first proposal for updating the relationships and the topologies based on this concept was also performed by Magnus.

8. IANA Considerations

This document makes no request of IANA.

9. Informative References

[I-D.ietf-avtcore-rtp-multi-stream]
Lennox, J., Westerlund, M., Wu, W., and C. Perkins, "Sending Multiple Media Streams in a Single RTP Session", draft-ietf-avtcore-rtp-multi-stream-08 (work in progress), July 2015.

[I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-10 (work in progress), July 2015.

[I-D.ietf-clue-framework]
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-22 (work in progress), April 2015.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-23 (work in progress), July 2015.

[I-D.ietf-mmusic-sdp-simulcast]
Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in SDP and RTP Sessions", draft-ietf-mmusic-sdp-simulcast-00 (work in progress), January 2015.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-14 (work in progress), June 2015.

[I-D.westerlund-avtcore-transport-multiplexing]
Westerlund, M. and C. Perkins, "Multiplexing Multiple RTP Sessions onto a Single Lower-Layer Transport", draft-westerlund-avtcore-transport-multiplexing-07 (work in progress), October 2013.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <http://www.rfc-editor.org/info/rfc2198>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <http://www.rfc-editor.org/info/rfc3551>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, DOI 10.17487/RFC4353, February 2006, <http://www.rfc-editor.org/info/rfc4353>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <http://www.rfc-editor.org/info/rfc4588>.

[RFC4867] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 4867, DOI 10.17487/RFC4867, April 2007, <http://www.rfc-editor.org/info/rfc4867>.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <http://www.rfc-editor.org/info/rfc5109>.

[RFC5404] Westerlund, M. and I. Johansson, "RTP Payload Format for G.719", RFC 5404, DOI 10.17487/RFC5404, January 2009, <http://www.rfc-editor.org/info/rfc5404>.

[RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <http://www.rfc-editor.org/info/rfc5481>.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <http://www.rfc-editor.org/info/rfc5576>.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, DOI 10.17487/RFC5888, June 2010, <http://www.rfc-editor.org/info/rfc5888>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <http://www.rfc-editor.org/info/rfc5905>.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, DOI 10.17487/RFC6190, May 2011, <http://www.rfc-editor.org/info/rfc6190>.

[RFC7160] Petit-Huguenin, M. and G. Zorn, Ed., "Support for Multiple Clock Rates in an RTP Session", RFC 7160, DOI 10.17487/RFC7160, April 2014, <http://www.rfc-editor.org/info/rfc7160>.

[RFC7197] Begen, A., Cai, Y., and H. Ou, "Duplication Delay Attribute in the Session Description Protocol", RFC 7197, DOI 10.17487/RFC7197, April 2014, <http://www.rfc-editor.org/info/rfc7197>.

[RFC7198] Begen, A. and C. Perkins, "Duplicating RTP Streams", RFC 7198, DOI 10.17487/RFC7198, April 2014, <http://www.rfc-editor.org/info/rfc7198>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <http://www.rfc-editor.org/info/rfc7201>.

[RFC7273] Williams, A., Gross, K., van Brandenburg, R., and H. Stokking, "RTP Clock Source Signalling", RFC 7273, DOI 10.17487/RFC7273, June 2014, <http://www.rfc-editor.org/info/rfc7273>.

Appendix A. Changes From Earlier Versions

NOTE TO RFC EDITOR: Please remove this section prior to publication.

A.1. Modifications Between WG Version -07 and -08

Addresses comments from IESG evaluation.

- Made text more firm around what improvements this document introduces.

- Clarified the distinction between analog and digital in sections 2.1.1 and 2.1.2.

- Removed the explicit requirement that a Source RTP Stream must send at least some data from an Encoded Stream, replacing it with a statement that it is directly related to the Encoded Stream.

- Moved the clarification that RTP-based Redundancy excludes Media Encoder redundancy data in an Encoded Stream from Section 2.1.10 (RTP Stream) to 2.1.11 (RTP-based Redundancy), since that statement applies to RTP-based Redundancy rather than to RTP Stream.

- Added clarification that a Media Transport Sender can intentionally pace packet transmission.

- Aligned text around delay variation to use this term throughout, and added a reference to RFC 5481.

- Added that RTP Session is a group communications channel that can potentially carry a number of RTP Streams, as an additional clarification below Figure 7.

- Added a clarification in Section 4.1 around Telepresence Terms on which references are to CLUE terms and which are to other sections of this document, for terms that have the same name in CLUE as in this document.

- Clarified in Section 4.14 what SSRC data format changes means, since the RFC 3550 SSRC definition mentions this possibility.

- Editorial improvements.

A.2. Modifications Between WG Version -06 and -07

Addresses comments from AD review and GenArt review.

- Added RTP-based Security and RTP-based Validation transform sections, as well as Secured RTP Stream and Received Secured RTP Stream sections.

- Improved wording in Abstract and Introduction sections.

- Clarified what is considered "media" in section 2.1.2 Media Capture.

- Changed a number of "Characteristics" lists to more suitable prose text.

- Re-worded text around use of Encoded and Dependent RTP Streams in section 2.1.9 Media Packetizer.

- Clarified description of Source RTP Stream in section 2.1.10.

- Clarified motivation to use separate Media Transports for Simulcast in section 3.6.

- Added local descriptions of terms imported from CLUE framework.

- Editorial improvements.

A.3. Modifications Between WG Version -05 and -06

- Clarified that a Redundancy RTP Stream can be used standalone to generate Repaired RTP Streams.

- Clarified that (in accordance with above) RTP-based Repair takes zero or more Received RTP Streams and one or more Received Redundancy RTP Streams as input.

- Changed Figure 6 to more clearly show that Media Transport is terminated in the Endpoint, not in the Participant.

- Added a sentence to Endpoint section that clarifies there may be contexts where a single "host" can serve multiple Participants, making those Endpoints share some properties.

- Merged previous section 3.5 on SST/MST with previous section 3.8 on Layered Multi-Stream into a common section discussing the scalable/layered stream relation, and moved improved, descriptive

text on SST and MST to new sub-sections 4.7 and 4.13, describing them as existing terms.

- Editorial improvements.

A.4. Modifications Between WG Version -04 and -05

- Editorial improvements.

A.5. Modifications Between WG Version -03 and -04

- Changed "Media Redundancy" and "Media Repair" to "RTP-based Redundancy" and "RTP-based Repair", since those terms are more specific and correct.

- Changed "End Point" to "Endpoint" and removed Editor's Note on this.

- Clarified that a Media Capture may impose constraints on clock handling.

- Clarified that mixing multiple Raw Streams into a Source Stream is not possible, since that requires mixed streams to have a timing relation, requiring them to be Source Streams, and added an example.

- Clarified that RTP-based Redundancy excludes the type of encoding redundancy found within the encoded media format in an Encoded Stream.

- Clarified that a Media Transport contains only a single RTP Session, but a single RTP Session can span multiple Media Transports.

- Clarified that packets with seemingly correct checksum that are received by a Media Transport Receiver may still be corrupt.

- Clarified that a corrupt packet in a Media Transport Receiver is typically either discarded or somehow marked and passed on in the Received RTP Stream.

- Added Synchronization Context to Figure 6.

- Editorial improvements and clarifications.

A.6. Modifications Between WG Version -02 and -03

- Changed section 3.5, removing SST-SS/MS and MST-SS/MS, replacing them with SRST, MRST, and MRMT.

- Updated section 3.8 to align with terminology changes in section 3.5.

- Added a new section 4.12, describing the term Multimedia Conference.

- Changed reference from I-D to now published RFC 7273.

- Editorial improvements and clarifications.

A.7. Modifications Between WG Version -01 and -02

- Major re-structure

- Moved media chain Media Transport detailing up one section level

- Collapsed level 2 sub-sections of section 3 and thus moved level 3 sub-sections up one level, gathering some introductory text into the beginning of section 3

- Added that not only SSRC collision, but also a clock rate change [RFC7160] is a valid reason to change SSRC value for an RTP stream

- Added a sub-section on clock source signaling

- Added a sub-section on RTP stream duplication

- Elaborated a bit in section 2.2.1 on the relation between End Points, Participants and CNAMEs

- Elaborated a bit in section 2.2.4 on Multimedia Session and synchronization contexts

- Removed the section on CLUE scenes defining an implicit synchronization context, since it was incorrect

- Clarified text on SVC SST and MST according to list discussions

- Removed the entire topology section to avoid possible inconsistencies or duplications with draft-ietf-avtcore-rtp-topologies-update, but saved one example overview figure of Communication Entities into that section

- Added a section 4 on mapping from existing terms with one sub-section per term, mainly by moving text from sections 2 and 3

- Changed all occurrences of Packet Stream to RTP Stream

- Moved all normative references to informative, since this is an informative document

- Added references to RFC 7160, RFC 7197 and RFC 7198, and removed unused references

A.8. Modifications Between WG Version -00 and -01

- WG version -00 text is identical to individual draft -03

- Amended description of SVC SST and MST encodings with respect to concepts defined in this text

- Removed UML as normative reference, since the text no longer uses any UML notation

- Removed a number of level 4 sections and moved out text to the level above

A.9. Modifications Between Version -02 and -03

- Section 4 rewritten (and new communication topologies added) to reflect the major updates to Sections 1-3

- Section 8 removed (carryover from initial -00 draft)

- General clean up of text, grammar and nits

A.10. Modifications Between Version -01 and -02

- Section 2 rewritten to add both streams and transformations in the media chain.

- Section 3 rewritten to focus on exposing relationships.

A.11. Modifications Between Version -00 and -01

- Too many to list

- Added new authors

- Updated content organization and presentation

Authors' Addresses

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

Email: jonathan@vidyo.com

Kevin Gross
AVA Networks, LLC
Boulder, CO
US

Email: kevin.gross@avanw.com

Suhas Nandakumar
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
US

Email: snandaku@cisco.com

Gonzalo Salgueiro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
US

Email: gsalguei@cisco.com

Bo Burman (editor)
Ericsson
Kistavagen 25
SE-16480 Stockholm
Sweden

Email: bo.burman@ericsson.com