Network Working Group J. Lennox
Internet-Draft Vidyo
Intended status: Informational K. Gross
Expires: December 29, 2014 AVA
S. Nandakumar
G. Salgueiro
Cisco Systems
B. Burman
Ericsson
June 27, 2014

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

draft-ietf-avtext-rtp-grouping-taxonomy-02

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 relationships among RTP sources, and attempts to define 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 RTP is often 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 attempts to provide some 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 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 an attempt is made to list any alternate definitions and usages that co-exist today 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 memo, Media is a sequence of synthetic or Physical Stimulus (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 stimulus. 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:

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

- o A stream is of some type.

- o 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).

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

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

- o There are no formal limitations on how streams are connected to transformations, this may include loops if required by a particular transformation.

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 below first introduce the concepts for the sender side (Figure 1). This covers physical stimulus 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  | | |  Media Redundancy  |
              +--------------------+ | +--------------------+
                        |            |           |
                        +------------+ Redundancy RTP Stream
                 Source 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 below in Section 2.1.13.

Below we review a receiver media chain (Figure 2) matching the sender side to look at the inverse transformations and their attempts to recover possibly identical streams as in the sender chain. 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 Media Repair, if used, fails to repair. It should be noted that some transformations are not always present, like Media Repair that cannot operate without Redundancy RTP Streams.

```
        +--------------------+   +--------------------+
        |  Media Transport   |   |  Media Transport   |
        +--------------------+   +--------------------+
                  |                         |
         Received RTP Stream  Received Redundancy RTP Stream
                  |                         |
                  |    +-------------------+
                  V    V
        +--------------------+
        |    Media 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 that can be measured and converted to digital form by an appropriate sensor or transducer. This include sound waves making up audio, photons in a light field that is visible, 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 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 due to its periodical sampling, or at least being timed asynchronous events, some form of a stream of media data. 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 support such a configuration.

2.1.3. Raw Stream

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 reference clock synchronized, time progressing, digital media stream, 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. This output has been synchronized with some reference clock, even if just a system local wall 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 Raw Streams (Figure 3), a mixed selection of the three loudest inputs regarding speech activity, a selection of a particular video based on the current speaker, i.e. typically based on other Media Sources.

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

Figure 3: Conceptual Media Source in form of Audio Mixer

Characteristics:

- o At any point, it can represent a physical captured source or conceptual source.

2.1.5. Source Stream

A time progressing 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 bit-rate, start points for decoding, resolution, bandwidth or other fidelity affecting properties. The actually used codec is also an important factor in many communication systems, not only its parameters.

Scalable Media Encoders need special mentioning as they produce multiple outputs that are potentially of different types. 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) that requires at least one Encoded Stream and zero or more Dependent Streams to be possible to decode. A Dependent Stream's dependency is one of the grouping relations this document discusses further in Section 3.8.

```
                 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 Encoder produce multiple independent and thus individually decodable Encoded Streams that are possible to combine into a Received Source Stream that is somehow a better representation of the original Source Stream than using only a single Encoded Stream.

Characteristics:

- o A Media Source can be multiply encoded by different Media Encoders to provide various encoded representations.

2.1.7. Encoded Stream

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

Characteristics:

- o 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.

Characteristics:

- Each Dependent Stream has a set of dependencies. These dependencies must be understood by the parties in a multi-media 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 Stream (Section 2.1.8) and put 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 can use multiple inputs when producing a single RTP Stream. One such example is SST packetization when using SVC (Section 3.5).

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 MST packetization when using SVC (Section 3.5).

Characteristics:

- The Media Packetizer will select which Synchronization source(s) (SSRC) [RFC3550] in which RTP sessions that are used.

- Media Packetizer can combine multiple Encoded or Dependent Streams into one or more RTP Streams.

2.1.10. RTP Stream

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 a RTP Stream containing at least some content from an Encoded Stream. Source material is any media material that is produced for transport over RTP without any additional redundancy

applied to cope with network transport losses. Compare this with the Redundancy RTP Stream (Section 2.1.12).

Characteristics:

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

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

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

- Several RTP Streams may map to a single Media Source via the source transformations.

- Several RTP Streams can be carried over a single RTP Session.

2.1.11. Media Redundancy

Media redundancy is a transformation that generates redundant or repair packets sent out as a Redundancy RTP Stream to mitigate network transport impairments, like packet loss and delay.

The Media Redundancy exists in many flavors; they may be generating independent Repair Streams that are used in addition to the Source Stream (RTP Retransmission [RFC4588] and some FEC [RFC5109]), they may generate a new Source Stream by combining redundancy information with source information (Using XOR FEC [RFC5109] as a redundancy payload [RFC2198]), or completely replace the source information with only redundancy packets.

2.1.12. Redundancy RTP Stream

A RTP Stream (Section 2.1.10) that contains no original source data, only redundant data that may be combined with one or more Received RTP Stream (Section 2.1.19) to produce Repaired RTP Streams (Section 2.1.22).

2.1.13. 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 may contain multiple RTP receivers per sender). Each Media Transport is defined by a transport association that is identified by a 5-tuple (source address, source port, destination address, destination port, transport protocol). Each transport association normally contains only a single RTP session, although a proposal exists for sending multiple RTP sessions over one transport association [I-D.westerlund-avtcore-transport-multiplexing].

Characteristics:

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

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 5).

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

Figure 5: Decomposition of Media Transport

2.1.14. Media Transport Sender

The first transformation within the Media Transport (Section 2.1.13) is the Media Transport Sender, where the sending End-Point (Section 2.2.1) takes a RTP Stream and emits the packets onto the network using the transport association established for this Media Transport thus creating a Sent RTP Stream (Section 2.1.15). In this process it transforms the RTP Stream in several ways. First, it gains 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, pace or otherwise affect how the packets are emitted onto the network. Thus adding delay, jitter and inter packet spacings that characterize the Sent RTP Stream.

2.1.15. 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)).

2.1.16. Network Transport

Network Transport is the transformation that the Sent RTP Stream (Section 2.1.15) is subjected to by traveling from the source to the destination through the network. These transformations include, loss of some packets, varying delay on a per packet basis, packet duplication, and packet header or data corruption. These transformations produces a Transported RTP Stream (Section 2.1.17) at the exit of the network path.

2.1.17. Transported RTP Stream

The RTP Stream that is emitted out of the network path at the destination, subjected to the Network Transport's transformation (Section 2.1.16).

2.1.18. Media Transport Receiver

The receiver End-Point's (Section 2.2.1) transformation of the Transported RTP Stream (Section 2.1.17) by its reception process that result in the Received RTP Stream (Section 2.1.19). This transformation includes transport checksums being verified and if non-matching, causing discarding of the corrupted packet. Other transformations can include delay variations in receiving a packet on the network interface and providing it to the application.

2.1.19. Received RTP Stream

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

2.1.20. Received Redundancy RTP Stream

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

2.1.21. Media Repair

A Transformation that takes as input one or more Source RTP Streams (Section 2.1.10) as well as Redundancy RTP Streams (Section 2.1.12) and attempts to combine them to counter the transformations introduced by the Media Transport (Section 2.1.13) to minimize the difference between the Source Stream (Section 2.1.5) and the Received Source Stream (Section 2.1.26) after Media Decoder (Section 2.1.25). The output is a Repaired RTP Stream (Section 2.1.22).

2.1.22. Repaired RTP Stream

A Received RTP Stream (Section 2.1.19) for which Received Redundancy RTP Stream (Section 2.1.20) information has been used to try to re-create the RTP Stream (Section 2.1.10) as it was before Media Transport (Section 2.1.13).

2.1.23. Media Depacketizer

A Media Depacketizer takes one or more RTP Streams (Section 2.1.10) and 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.

It should be noted that in practical implementations, the Media Depacketizer and the Media Decoder 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.

2.1.24. Received Encoded Stream

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

2.1.25. 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).

It should be noted that 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.

Characteristics:

- A Media Decoder is the entity that will have to deal with any errors in the encoded streams that resulted from corruptions or failures to repair packet losses. This as a media decoder generally is forced to produce some output periodically. It thus commonly includes concealment methods.

2.1.26. Received Source Stream

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

2.1.27. 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 sent in synchronization with the output from other Media Sinks to a Media Render (Section 2.1.29). 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.

Characteristics:

- 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.

2.1.28. Received Raw Stream

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

2.1.29. 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, D/A converters connected to amplifiers and loudspeakers.

Characteristics:

- o An End Point can potentially have multiple Media Renders for each media type.

2.2. Communication Entities

This section contains concept for entities involved in the communication.

```
+------------------------------------------------------------+
| Communication Session                                      |
|                                                            |
| +----------------+                   +----------------+ |
| | Participant A  |   +------------+  | Participant B  | |
| |                |   | Multimedia |  |                | |
| | +-------------+|<=>| Session    |<=>|+-------------+ | |
| | | End Point A ||   |            |   || End Point B | | |
| | |             ||   +------------+   ||             | | |
| | | +-----------++-------------------++-----------+ | | |
| | | | RTP Session|                   |           | | | |
| | | | Audio      |--Media Transport-->|           | | | |
| | | |            |<--Media Transport--|           | | | |
| | | +-----------++-------------------++-----------+ | | |
| | |             ||                   ||             | | |
| | | +-----------++-------------------++-----------+ | | |
| | | | RTP Session|                   |           | | | |
| | | | Video      |--Media Transport-->|           | | | |
| | | |            |<--Media Transport--|           | | | |
| | | +-----------++-------------------++-----------+ | | |
| | +-------------+|                   |+-------------+ | |
| +----------------+                   +----------------+ |
+------------------------------------------------------------+
```

Figure 6: Example Point to Point Communication Session with two RTP Sessions

The figure above 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 End Points, 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 that figure are further elaborated in the sub-sections below.

2.2.1. End Point

Editor's note: Consider if a single word, "Endpoint", is preferable

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 "End Point".

Characteristics:

- End Points 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 End Point can handle multiple CNAMEs, each of which can be shared among a set of End Points 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 ensure End Point identification when outside this Synchronization Context.

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

- In some contexts, an End Point would typically correspond to a single "host".

2.2.2. RTP Session

Editor's note: Re-consider if this is really a Communication Entity, or if it is rather an existing concept that should be described in Section 4.

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:

- Typically, 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 End Points participating in an RTP Session can see an SSRC identifier transmitted by any of the other End Points. An End Point 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 2.1.13), one for sending and one for receiving. Commonly, the receiving one is the reverse direction of the same one as used for sending. An RTP Session may use many Media Transports and these define the session's network interconnection topology. A single Media Transport can normally not transport more than one RTP Session, unless a solution for multiplexing multiple RTP sessions over a single Media Transport is used. One example of such a scheme is Multiple RTP Sessions on a Single Lower-Layer Transport [I-D.westerlund-avtcore-transport-multiplexing].

- Multiple RTP Sessions can be related.

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 have several Multimedia Sessions (Section 2.2.4).

- A Participant can have several associated End Points (Section 2.2.1).

2.2.4. Multimedia Session

A multimedia session is an association among a group of participants 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:

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

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

- o 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 group of participants communicating with each other via a set of Multimedia Sessions.

Characteristics:

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

- o A Communication Session is composed of at least one Multimedia Session per participant, 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. Relations at Different Levels

This section uses the concepts from previous section and look at different types of relationships among them. These relationships occur at different levels and for different purposes. The section is organized such as to look at the level where a relation is required. The reason for the relationship may exist at another step in the media handling chain. For example, using Simulcast (discussed in Section 3.7) needs to determine relations at RTP Stream level, however the reason to relate RTP Streams is that multiple Media Encoders use the same Media Source, i.e. to be able to identify a common Media Source.

Media Sources (Section 2.1.4) are commonly grouped and related to an End Point (Section 2.2.1) or a Participant (Section 2.2.3). This occurs for several reasons; both due to application logic as well as for media handling purposes.

At RTP Packetization time, there exists a possibility for a number of different types of relationships between Encoded Streams (Section 2.1.7), Dependent Streams (Section 2.1.8) and RTP Streams (Section 2.1.10). These are caused by grouping together or distributing these different types of streams into RTP Streams.

The resulting RTP Streams will thus also have relations. This is a common relation to handle in RTP due to that RTP Streams are separate and have their own SSRC, implying independent sequence numbers and timestamp spaces. The underlying reasons for the RTP Stream relationships are different, as can be seen in the sub-sections below.

RTP Streams may be protected by Redundancy RTP Streams during transport. Several approaches listed below can be used to create Redundancy RTP Streams;

- Duplication of the original RTP Stream

- Duplication of the original RTP Stream with a time offset,

- Forward Error Correction (FEC) techniques, and

- Retransmission of lost packets (either globally or selectively).

The different RTP Streams can be transported within the same RTP Session or in different RTP Sessions to accomplish different transport goals. This explicit separation of RTP Streams is further discussed in Section 3.13.

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 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 [I-D.ietf-avtcore-clksrc], despite using NTP format timestamps, it is not required that the clock be synchronized to an NTP source.

3.1.2. Clock Source Signaling

[I-D.ietf-avtcore-clksrc] provides a mechanism to signal the clock source in SDP 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

The WebRTC WG defines "RtcMediaStream" with one or more "RtcMediaStreamTracks". All tracks in a "RtcMediaStream" are intended to be possible to synchronize when rendered.

3.1.4. Explicitly via SDP Mechanisms

RFC5888 [RFC5888] defines m=line grouping mechanism called "Lip Synchronization (LS)" for establishing the synchronization requirement across m=lines when they map to individual sources.

RFC5576 [RFC5576] extends the above mechanism when multiple media sources are described by a single m=line.

3.2. End Point

Some applications requires knowledge of what Media Sources originate from a particular End Point (Section 2.2.1). This can include such decisions as packet routing between parts of the topology, knowing the End Point 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) item. This works for some usages, but sometimes it breaks down. For example, if an End Point 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. Thus, an End Point may have multiple CNAMEs. The CNAMEs or the Media Sources themselves can be related to the End Point.

3.3. Participant

In communication scenarios, it is commonly needed to know which Media Sources that originate from which Participant (Section 2.2.3). Thus enabling the application to for example display Participant Identity information correctly associated with the Media Sources. This association is currently 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 End Point.

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 Contributing Sources (CSRC) that carries such 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. Single- and Multi-Session Transmission of SVC

Scalable Video Coding [RFC6190] has a mode of operation called Single Session Transmission (SST), where Encoded Streams and Dependent Streams from the SVC Media Encoder are sent in a single RTP Session (Section 2.2.2) using the SVC RTP Payload format. There is another mode of operation where Encoded Streams and Dependent Streams are distributed across multiple RTP Sessions, called Multi-Session Transmission (MST). SST denotes one or more RTP Streams (SSRC) per Media Source in a single RTP Session. MST denotes one or more RTP Streams (SSRC) per Media Source in each of multiple RTP Sessions. This is not always clear from the SVC payload format text [RFC6190], but is what existing deployments of that RFC have implemented.

To elaborate, what could be called SST-SingleStream (SST-SS) uses a single RTP Stream in a single RTP Session to send all Encoded and Dependent Streams from a single Media Source. Similarly, SST-MultiStream (SST-MS) uses a single RTP Stream per Media Source in a single RTP Session to send the Encoded and Dependent Streams. MST-SS uses a single RTP Stream in each of multiple RTP Sessions, where each RTP Stream can originate from any one of possibly multiple Media Sources. Finally, MST-MS uses multiple RTP Streams in each of the multiple RTP Sessions, where each RTP Stream can originate from any one of possibly multiple Media Sources. This is summarized below:

RTP Streams per Media Source	Single RTP Session	Multiple RTP Sessions
Single	SST-SS	MST-SS
Multiple	SST-MS	MST-MS

Table 1: SST / MST Summary

3.6. Multi-Channel Audio

There exist a number of RTP payload formats that can carry multi-channel audio, despite the codec being a 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 then packetized together in a time synchronized way into a single Source RTP Stream using the used codec's RTP Payload format. Example of such codecs are, PCMA and PCMU [RFC3551], AMR [RFC4867], and G.719 [RFC5404].

3.7. Simulcast

A Media Source represented as multiple independent Encoded Streams constitutes a simulcast of that Media Source. Figure 7 below represents an example of a Media Source that is encoded into three separate and different 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 be any combination of these two. It is other considerations that affect which usage is desirable, as discussed in Section 3.13.

```
                            +----------------+
                            |  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 7: 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 to enable selection of the stream that is most useful in the application at that moment.

3.8. Layered Multi-Stream

Layered Multi-Stream (LMS) is a mechanism by which different portions of a layered 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. The figure below 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, i.e. a different RTP Session.

```
                         +----------------+
                         |  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 8: Example of Media Source Layered Dependency

As an example, the SVC MST (Section 3.5) relation needs to identify the common Media Encoder origin for the Encoded and Dependent Streams. The SVC RTP Payload RFC is not particularly explicit about how this relation is to be implemented. When using different RTP Sessions, thus different Media Transports, and as long as there is only one RTP Stream per Media Encoder and a single Media Source in each RTP Session (MST-SS (Section 3.5)), 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 (SST-MS), 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 need to be applied to create the MST relationship between the RTP Streams.

3.9. 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. It is a specific type of redundancy and 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 is which. This can also be seen as a specific type of Simulcast (Section 3.7) 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 9: Example of RTP Stream Duplication

3.10. Redundancy Format

The RTP Payload for Redundant Audio Data [RFC2198] defines how one can transport 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 below in Figure 10.

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

Figure 10: 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, or in the case depicted above (Figure 10) relate the Received Source Stream fragments coming out of different Media Decoders to be able to combine them together into a less erroneous Source Stream.

3.11. RTP Retransmission

The figure below (Figure 11) represents 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 11: Example of Media Source Retransmission Flows

The RTP Retransmission example (Figure 11) helps illustrate that this mechanism works purely on the Source RTP Stream. The RTP Retransmission transform buffers the sent Source RTP Stream and upon requests emits a retransmitted packet with some extra payload header as a Redundancy RTP Stream. The RTP Retransmission mechanism [RFC4588] is specified so that there is a one to one relation between the Source RTP Stream and the Redundancy RTP Stream. Thus a Redundancy RTP Stream needs to be associated with its Source RTP Stream upon being received. 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 separate RTP Session from the Source RTP Stream, these sessions are related, e.g. using the SDP Media Grouping's [RFC5888] FID semantics.

3.12. Forward Error Correction

The figure below (Figure 12) represents an example where two Media Sources' Source RTP Streams are protected by FEC. Source RTP Stream A has a Media 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 together, i.e. Redundancy RTP Stream 2 relate 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 12 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 12: Example of FEC Flows

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 to either use a separate RTP session or to use 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.

3.13. 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 End Point. [RFC5576]-based approaches, when used, can explicitly relate various such RTP Streams.

On the other hand, when RTP Streams that are related but 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.

3.14. Multiple RTP Sessions over one Media Transport

[I-D.westerlund-avtcore-transport-multiplexing] describes a mechanism that allow 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. Thus using a common network path or potentially have different ones. There is reduced need for NAT/FW traversal resources and no need for flow based QoS.

However, Multiple RTP Sessions over one Media Transport makes it clear that a single Media Transport 5-tuple is not sufficient to express which RTP Session context a particular RTP Stream exists in. 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. Audio Capture

Telepresence specifications from CLUE WG uses this term to describe an audio Media Source (Section 2.1.4).

4.2. Capture Device

Telepresence specifications from CLUE WG use this term to identify a physical entity performing a Media Capture (Section 2.1.2) transformation.

4.3. Capture Encoding

Telepresence specifications from CLUE WG uses this term to describe an Encoded Stream (Section 2.1.7) related to CLUE specific semantic information.

4.4. Capture Scene

Telepresence specifications from CLUE WG uses this term to describe a set of spatially related Media Sources (Section 2.1.4).

4.5. Endpoint

Telepresence specifications from CLUE WG use this term to describe exactly one Participant (Section 2.2.3) and one or more End Points (Section 2.2.1).

4.6. Individual Encoding

Telepresence specifications from CLUE WG use this term to describe the configuration information needed to perform a Media Encoder (Section 2.1.6) transformation.

4.7. 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 End Points (Section 2.2.1) (sometimes one or more per conference participant).

4.8. Media Capture

Telepresence specifications from CLUE WG uses this term to describe either a Media Capture (Section 2.1.2) or a Media Source (Section 2.1.4), depending on in which context the term is used.

4.9. Media Consumer

Telepresence specifications from CLUE WG use this term to describe the media receiving part of an End Point (Section 2.2.1).

4.10. Media Description

A single Source 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).

4.11. Media Provider

Telepresence specifications from CLUE WG use this term to describe the media sending part of an End Point (Section 2.2.1).

4.12. Media Stream

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

4.13. 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 End Points and the RTP Streams that flow between them. In this memo, Multimedia Session also assumes those End Points 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 End Points) sharing a set of concurrent RTP Sessions. In this memo, Multimedia Session also defines those RTP Sessions to have some relation and be part of a communication among the Participants.

4.14. Recording Device

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

4.15. RtcMediaStream

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

4.16. RtcMediaStreamTrack

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

4.17. RTP Sender

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

4.18. RTP Session

Within the context of SDP, a singe m=line can map to a single RTP Session 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.

> Editor's note: Consider if the contents of Section 2.2.2 should be moved here, or if this section should be kept and refer to the above.

4.19. 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).

4.20. Stream

Telepresence specifications from CLUE WG use this term to describe an RTP Stream (Section 2.1.10).

4.21. Video Capture

Telepresence specifications from CLUE WG uses this term to describe a video Media Source (Section 2.1.4).

5. Security Considerations

This document simply tries to clarify 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.

Hopefully 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], 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, and Alex Eleftheriadis.

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-clksrc]
Williams, A., Gross, K., Brandenburg, R., and H. Stokking, "RTP Clock Source Signalling", draft-ietf-avtcore-clksrc-11 (work in progress), March 2014.

[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-04 (work in progress), May 2014.

[I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", draft-ietf-avtcore-rtp-topologies-update-02 (work in progress), May 2014.

[I-D.ietf-clue-framework]
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-15 (work in progress), May 2014.

[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-07 (work in progress), April 2014.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Browser-based Applications", draft-ietf-rtcweb-overview-10 (work in progress), June 2014.

[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, September 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.

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

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

[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, April 2007.

[RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.

[RFC5404] Westerlund, M. and I. Johansson, "RTP Payload Format for G.719", RFC 5404, January 2009.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", RFC 6190, May 2011.

[RFC7160] Petit-Huguenin, M. and G. Zorn, "Support for Multiple Clock Rates in an RTP Session", RFC 7160, April 2014.

[RFC7197] Begen, A., Cai, Y., and H. Ou, "Duplication Delay Attribute in the Session Description Protocol", RFC 7197, April 2014.

[RFC7198] Begen, A. and C. Perkins, "Duplicating RTP Streams", RFC 7198, April 2014.

Appendix A. Changes From Earlier Versions

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

A.1. 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.2. 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.3. 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.4. 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.5. 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
Ericsson
Kistavagen 25
SE-164 80 Kista
Sweden

Phone: +46 10 714 13 11
Email: bo.burman@ericsson.com