Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 23, 2014

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

September 19, 2013

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

This document is still very rough, but is submitted in the hopes of making future discussion productive.

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

This section contains the concepts that can be involved in taking a sequence of physical world stimulus (sound waves, photons, key-strokes) at a sender side and transport them 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 are of some type.

o A Stream has one source transformation and one or more sink transformation (with the exception of Physical Stimulus (Section 2.1.1) that can have no 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.

- 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 conecptual 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 introduces 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 Packet Stream
                Source Packet 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 2.1.13.2.

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 potentialy 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 Packet Streams.

```
        +----------------------+   +----------------------+
        |  Media Transport   |   |  Media Transport   |
        +----------------------+   +----------------------+
                  |                         |
        Received Packet Stream   Received Redundancy PS
                  |                         |
                  |    +--------------------+
                  V    V
        +----------------------+
        |    Media Repair    |
        +----------------------+
                  |
        Repaired Packet 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 captured and provided as media to a receiver. This include soundwaves 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

The process of transforming the Physical Stimulus (Section 2.1.1) into captured media. The Media Capture performs a digitial 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.

2.1.2.1. Alternate Usages

The CLUE WG uses the term "Capture Device" to identify a physical capture device.

WebRTC WG uses the term "Recording Device" to refer to the locally available capture devices in an end-system.

2.1.2.2. Characteristics

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

2.1.3. Raw Stream

The time progressing stream of digitialy sampled information, usually periodically sampled, provided by a Media Capture (Section 2.1.2).

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

2.1.4.1. Alternate Usages

The CLUE WG uses the term "Media Capture" for this purpose. A CLUE Media Capture is identified via indexed notation. The terms Audio Capture and Video Capture are used to identify Audio Sources and Video Sources respectively. Concepts such as "Capture Scene", "Capture Scene Entry" and "Capture" provide a flexible framework to represent media captured spanning spatial regions.

The WebRTC WG defines the term "RtcMediaStreamTrack" to refer to a Media Source. An "RtcMediaStreamTrack" is identified by the ID attribute.

Typically a Media Source is mapped to a single m=line via the Session Description Protocol (SDP) [RFC4566] unless mechanisms such as Source-Specific attributes are in place [RFC5576]. In the latter cases, an m=line can represent either multiple Media Sources, multiple Packet Streams (Section 2.1.10), or both.

2.1.4.2. 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 3.3.2.

```
                         Source Stream
                               |
                               V
                  +--------------------------+
                  |  Scalable Media Encoder  |
```

```
                       +--------------------------+
                          |         |    ...    |
                          V         V           V
                       Encoded  Dependent  Dependent
                       Stream    Stream     Stream
```

Figure 4: Scalable Media Encoder Input and Outputs

2.1.6.1. Alternate Usages

Within the SDP usage, an SDP media description (m=line) may describe part of the necessary configuration required for encoding purposes.

CLUE's "Capture Encoding" provides specific encoding configuration for this purpose.

2.1.6.2. 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.

2.1.7.1. 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 fragements that are dependent on one or more Encoded Streams (Section 2.1.7) and zero or more Dependent Streams to be possible to decode.

2.1.8.1. Characteristics

o 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 Packet 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 Packet Stream. One such example is the packetization when using SVC, as in Single Stream Transport (SST) usage of the payload format both an Encoded Stream as well as Dependent Streams are packetized in a single Source Packet Stream using a single SSRC.

The Media Packetizer can also produce multiple Packet Streams, for example when Encoded and/or Dependent Streams are distributed over multiple Packet Streams, possibly in different RTP sessions.

2.1.9.1. Alternate Usages

An RTP sender is part of the Media Packetizer.

2.1.9.2. Characteristics

- o The Media Packetizer will select which Synchronization source(s) (SSRC) [RFC3550] in which RTP sessions that are used.
- o Media Packetizer can combine multiple Encoded or Dependent Streams into one or more Packet Streams.

2.1.10. Packet Stream

A stream of RTP packets containing media data, source or redundant. The Packet 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 Packet Stream is a packet 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 Packet Stream (Section 2.1.12).

2.1.10.1. Alternate Usages

The term "Stream" is used by the CLUE WG to define an encoded Media Source sent via RTP. "Capture Encoding", "Encoding Groups" are defined to capture specific details of the encoding scheme.

RFC3550 [RFC3550] uses the terms media stream, audio stream, video stream and streams of (RTP) packets interchangably. It defines the SSRC as the "The source of a stream of RTP packets, ..."

The equivalent mapping of a Packet Stream in SDP [RFC4566] is defined per usage. For example, each Media Description (m=line) can describe one Packet Stream OR properties for multiple Packet Streams OR for an RTP session (via [RFC5576] mechanisms for example).

2.1.10.2. Characteristics

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

o At any given point in time, a Packet Stream can have one and only one SSRC.

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

o Several Packet Streams may map to a single Media Source via the source transformations (see Section 4.1).

o Several Packet 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 Packet Stream to mitigate network transport impairments, like packet loss and delay.

The Media Redundancy exists in many flavors; they may be generating indepdent 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 Packet Stream

A Packet Stream (Section 2.1.10) that contains no original source data, only redundant data that may be combined with one or more Received Packet Stream (Section 2.1.14) to produce Repaired Packet Streams (Section 2.1.17).

2.1.13. Media Transport

A Media Transport defines the transformation that the Packet 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].

2.1.13.1. Characteristics

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

2.1.13.2. Media Stream Decomposition

The Media Transport concept sometimes needs to be decomposited 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).

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

Figure 5: Decomposition of Media Transport

2.1.13.2.1. 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 Packet Stream and emits the packets onto the network using the transport association established for this Media Transport thus creating a Sent Packet Stream (Section 2.1.13.2.2). In this process it transforms the Packet Stream in several ways. First, it gains the necessary protocol headers for the transport assocaition, 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 Packet Stream.

2.1.13.2.2. Sent Packet Stream

The Sent Packet Stream is the Packet Stream as entering the first hop of the network path to its destination. The Sent Packet 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)).

2.1.13.2.3. Network Transport

Network Transport is the transformation that the Sent Packet Stream (Section 2.1.13.2.2) 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 Packet Stream (Section 2.1.13.2.4) at the exit of the network path.

2.1.13.2.4. Transported Packet Stream

The Packet Stream that is emitted out of the network path at the destination, subjected to the Network Transport's transformation (Section 2.1.13.2.3).

2.1.13.2.5. Media Transport Receiver

The receiver End-Point's (Section 2.2.1) transformation of the Transported Packet Stream (Section 2.1.13.2.4) by its reception process that result in the Received Packet Stream (Section 2.1.14). 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.

2.1.14. Received Packet Stream

The Packet 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.

2.1.15. Received Redundandy Packet Stream

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

2.1.16. Media Repair

A Transformation that takes as input one or more Source Packet Streams (Section 2.1.10) as well as Redundancy Packet 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 2.1.21) after Media Decoder (Section 2.1.20). The output is a Repaired Packet Stream (Section 2.1.17).

2.1.17. Repaired Packet Stream

A Received Packet Stream (Section 2.1.14) for which Received Redundancy Packet Stream (Section 2.1.15) information has been used to try to re-create the Packet Stream (Section 2.1.10) as it was before Media Transport (Section 2.1.13).

2.1.18. Media Depacketizer

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

2.1.19. Received Encoded Stream

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

2.1.20. 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).

2.1.20.1. Alternate Usages

Within the context of SDP, an m=line describes the necessary configuration and identification (RTP Payload Types) required to decode either one or more incoming Media Streams.

2.1.20.2. Characteristics

o 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.

2.1.21. Received Source Stream

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

2.1.22. 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 2.1.24). 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.

2.1.22.1. Characteristics

o 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.

2.1.23. Received Raw Stream

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

2.1.24. 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.

2.1.24.1. 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.

2.2.1. End Point

A single addressabke 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".

2.2.1.1. Alternate Usages

The CLUE Working Group (WG) uses the terms "Media Provider" and "Media Consumer" to describes aspects of End Point pertaining to sending and receiving functionalities.

2.2.1.2. 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 3.1.1). Thus one End Point can have multiple CNAMEs. 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.

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 Packet Streams. Within an RTP session, every participant can find meta-data and control information (over RTCP) about all the Packet Streams in the RTP session. The bandwidth of the RTCP control channel is shared between all participants within an RTP Session.

2.2.2.1. Alternate Usages

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.

2.2.2.2. Characteristics

o Typically, an RTP Session can carry one ore more Packet Streams.

o 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.

o 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].

o Multiple RTP Sessions can be related via mechanisms defined in Section 4.

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.

2.2.3.1. Characteristics

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

o A participant can have several Multimedia Sessions (Section 2.2.4).

- A participant can have several associated transport flows, including several separate local transport addresses for those transport flows.

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.

2.2.4.1. Alternate Usages

RFC4566 [RFC4566] defines a multimedia session as a set of multimedia senders and receivers and the data streams flowing from senders to receivers.

RFC3550 [RFC3550] defines it as set of concurrent RTP sessions among a common group of participants. For example, a videoconference (which is a multimedia session) may contain an audio RTP session and a video RTP session.

2.2.4.2. Characteristics

- Participants and their End Points in RTP Multimedia Sessions are identified via mechanisms such as RTCP CNAME or other application level identifiers, as appropriate.

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

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

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.

2.2.5.1. Alternate Usages

The Session Description Protocol (SDP) [RFC4566] defines a multimedia session as a set of multimedia senders and receivers and the data streams flowing from senders to receivers. In that definition it is however not clear if a multimedia session includes both the sender's and the receiver's view of the same RTP Packet Stream.

2.2.5.2. Characteristics

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

- A Communication Session is composed of at least one Multimedia Session per participant, involving one or more parallel RTP Sessions with potentially multiple Packet 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 3.3.1) needs to determine relations at Packet Stream level, however the reason to relate Packet Streams is that multiple Media Encoders use the same Media Source, i.e. to be able to identify a common Media Source.

3.1. Media Source Relations

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 application logic as well as media handling purposes. These cases are further discussed below.

3.1.1. 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.

A single Multimedia Session can contain media from one or more Synchronization Contexts. 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.

3.1.1.1. 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.

3.1.1.2. 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.

3.1.1.3. CLUE Scenes

In CLUE "Capture Scene", "Capture Scene Entry" and "Captures" define an implied Synchronization Context.

3.1.1.4. 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.

3.1.1.5. 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.

3.1.2. 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 Packet Streams.

In RTP, this identification has been overloded with the Synchronization Context through the usage of the source description CNAME 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.

3.1.3. 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 assoicated 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.

3.1.4. WebRTC MediaStream

An RtcMediaStream, in addition to requiring a single Synchronization Context as discussed above, is also an explicit grouping of a set of Media Sources, as identifed by RtcMediaStreamTracks, within the RtcMediaStream.

3.2. Packetization Time Relations

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 Packet Streams (Section 2.1.10). These are caused by grouping together or distributing these different types of streams into Packet Streams. This section will look at such relationships.

3.2.1. Single Stream Transport of SVC

Scalable Video Coding [RFC6190] has a mode of operation where Encoded Streams and Dependent Streams from the SVC Media Encoder is grouped

together in a single Source Packet Stream using the SVC RTP Payload format.

3.2.2. 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 then independently encoded by a Media Encoder and the different Encoded Streams are then packetized together in a time synchronizated way into a single Source Packet Stream using the used codec's RTP Payload format. Example of such codecs are, PCMA and PCMU [RFC3551], AMR [RFC4867], and G.719 [RFC5404].

3.2.3. 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 the same Media Source as the primary, as depicted below in Figure 6.

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

```
                 Packet Stream
```

Figure 6: 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 6) 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.

3.3. Packet Stream Relations

This section discusses various cases of relationships among Packet Streams. This is a common relation to handle in RTP due to that Packet Streams are separate and have their own SSRC, implying independent sequence numbers and timestamp spaces. The underlying reasons for the Packet Stream relationships are different, as can be seen in the cases below. The different Packet Streams can be handled within the same RTP Session or different RTP Sessions to accomplish different transport goals. This separation of Packet Streams is further discussed in Section 3.3.4.

3.3.1. 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 Packet 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 3.3.4.

```
                          +----------------+
                          |  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
            | Packet                 | Packet                  | Packet
            | Stream                 | Stream                  | Stream
            +-----------------+      |    +-----------------+
                              |      |    |
                              V      V    V
                        +--------------------+
                        |  Media Transport  |
                        +--------------------+
```

Figure 7: Example of Media Source Simulcast

The simulcast relation between the Packet Streams is the common Media Source. In addition, to be able to identify the common Media Source, a receiver of the Packet 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.

3.3.2. Layered Multi-Stream Transmission

Multi-stream transmission (MST) is a mechanism by which different portions of a layered encoding of a Source Stream are sent using separate Packet Streams (sometimes in separate RTP sessions). MSTs 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 dependency. 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 Packet 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|
       +----------------+   +----------------+   +----------------+
               |                    |                     |
         Packet Stream        Packet Stream         Packet Stream
               |                    |                     |
               +------+      +------+                     |
                      |      |                            |
                      V      V                            V
                +-----------------+              +-----------------+
                | Media Transport |              | Media Transport |
                +-----------------+              +-----------------+
```

           Figure 8: Example of Media Source Layered Dependency

The SVC MST 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 Packet 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 Packet Streams are sent from one Media Encoder in the same RTP Session, 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 Packet Streams.

3.3.3. Robustness and Repair

Packet Streams may be protected by Redundancy Packet Streams during transport. Several approaches listed below can achieve the same result;

- Duplication of the original Packet Stream

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

- Forward Error Correction (FEC) techniques, and

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

3.3.3.1. RTP Retransmission

The figure below (Figure 9) represents an example where a Media Source's Source Packet Stream is protected by a retransmission (RTX) flow [RFC4588]. In this example the Source Packet Stream and the Redundancy Packet Stream share the same Media Transport.

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

Figure 9: Example of Media Source Retransmission Flows

The RTP Retransmission example (Figure 9) helps illustrate that this mechanism works purely on the Source Packet Stream. The RTP Retransmission transform buffers the sent Source Packet Stream and upon requests emits a retransmitted packet with some extra payload header as a Redundancy Packet Stream. The RTP Retransmission mechanism [RFC4588] is specified so that there is a one to one relation between the Source Packet Stream and the Redundancy Packet Stream. Thus a Redundancy Packet Stream needs to be associated with its Source Packet Stream upon being received. This is done based on CNAME selectors and heuristics to match requested packets for a given Source Packet Stream with the original sequence number in the payload of any new Redundancy Packet Stream using the RTX payload format. In cases where the Redundancy Packet Stream is sent in a separate RTP Session from the Source Packet Stream, these sessions are related, e.g. using the SDP Media Grouping's [RFC5888] FID semantics.

3.3.3.2. Forward Error Correction

The figure below (Figure 10) represents an example where two Media Sources' Source Packet Streams are protected by FEC. Source Packet Stream A has a Media Redundnacy transformation in FEC Encoder 1. This produces a Redundancy Packet Stream 1, that is only related to Source Packet Stream A. The FEC Encoder 2, however takes two Source Packet Streams (A and B) and produces a Redundancy Packet Stream 2 that protects them together, i.e. Redundancy Packet Stream 2 relate to two Source Packet Streams (a FEC group). FEC decoding, when needed due to packet loss or packet corruption at the receiver, requires knowledge about which Source Packet Streams that the FEC encoding was based on.

In Figure 10 all Packet Streams are sent on the same Media Transport. This is however not the only possible choice. Numerous combinations exist for spreading these Packet 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 Packet Stream A                Source Packet Stream B
                 |                                     |
           +-----+-------+-------------+       +-------+------+
           |             V             V       V              |
           |    +---------------+  +---------------+          |
           |    | FEC Encoder 1 |  | FEC Encoder 2 |          |
           |    +---------------+  +---------------+          |
           |             |                 |                  |
           |     Redundancy PS 1    Redundancy PS 2           |
           V             V                 V                  V
       +----------------------------------------------------------+
       |                    Media Transport                       |
       +----------------------------------------------------------+
```

Figure 10: Example of FEC Flows

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

3.3.4. Packet Stream Separation

An important aspect of Packet Stream relations is the level of separation between the Packet Streams. This section discusses some alternatives.

3.3.4.1. SSRC-Only Based Separation

When the Packet Streams that have a relationship are all sent in the same RTP Session and their identification and separation from each other are based on the SSRC only, it is SSRC-Only Based Separation.

3.3.4.2. RTP Session Based Separation

Packet Streams that are related, but are sent in the context of different RTP Sessions to achieve separation between the Packet Streams. This is commonly used when the different Packet Streams are intended for different Media Transports.

Several mechanisms that uses RTP Session based separation relies 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 signalling level relations have been established using semantics from Grouping of Media lines framework [RFC5888]. Examples of this are RTP Retransmission [RFC4588], SVC Multi Stream Transmission [RFC6190] and XOR Based FEC [RFC5109].

3.3.4.3. Multimedia Session based Separation

Packet 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.

3.4. 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 Packet 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 identifers need to be considered in both signalling design and when defining terminology.

4. Topologies and Contexts

WARNING: This section has not yet been updated to reflect the changes in the previous sections (1-3). This is planned to be addressed in the next version!

This section provides various relationships that can co-exist between the aforementioned concepts in a given RTP usage. Using Unified Modeling Language (UML) class diagrams [UML], Figure 11 below depicts general relations between a Media Source, its Media Provider(s) and the resulting Media Stream(s).

Note: The RTCP Stream related to the RTP Stream is not shown in the figure.

```
+--------------+  <<uses>>  +-------------------------+
| Media Source |- - - - - ->| Synchronization Context |
+--------------+            +-------------------------+
      < > 1..*
       |
       | 0..*
+-------------+
|             |<>-+ 0..*
|    Media    |   |
|   Encoder   |   |
|             |---+ 0..*
+-------------+
      < > 1
       |
       | 0..*
+---------------+ 0..*     1 +-------------+
| Packet Stream |----------<>| RTP Session |
+---------------+            +-------------+
```

Figure 11: Media Source Relations

Media Sources can have a large variety of relationships among them. These relationships can apply both between sources within a single RTP Session, and between Media Sources that occur in multiple RTP Session. Ways of relating them typically involve groups: a set of Media Sources has some relationship that applies to all those in the group, and no others. (Relationships that involve arbitrary non-grouping associations among Media Sources, such that e.g., A relates to B and B to C, but A and C are unrelated, are uncommon if not nonexistent.) In many cases, the semantics of groups are not simply that the the members form an undifferentiated group, but rather that members of the group have certain roles.

4.1. Equivalence Context

In this relationship, different instances of a concept are treated to be equivalent for the purposes of relating them to the Media Source.

Figure 12 below depicts in UML notation the general relation between a Media Source and its Media Stream(s), including the Packet Stream specializations Source Packet Stream and Redundancy Packet Stream.

```
                      +---------------+
                      |               |<>-+ 0..*
                      |     Media     |   |
                      |    Encoder    |   |
                      |               |---+ 0..*
                      +---------------+
                             < > 1
                              |
                              | 0..*
                      +---------------+ 0..*  1 +-----------------+
                      | Packet Stream |<>-------| Media Transport |
                      +---------------+         +-----------------+
                        /\         /\
                       +--+      +--+
                        |           |
                +-------+           +-------+
                |                           |
        +---------------+           +---------------+ 1
        |     Source    |<>---------|  Redundancy   |<>-+
        | Packet Stream | 1..* 0..* | Packet Stream |---+
        +---------------+           +---------------+ 0..*
```

Figure 12: Media Stream Relations

This relation can in combination with Figure 11 be used to achieve a set of functionalities, described below.

4.1.1. SDP FID Semantics

RFC5888 [RFC5888] defines m=line grouping mechanism called "FID" for establishing the equivalence of Media Streams across the m=lines under grouping.

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

4.2. Session Context

There are different ways to construct a Communication Session. The general relation in UML notation between a Communication Session, Participants, Multimedia Sessions and RTP Sessions is outlined below.

```
                            +---------------+
                            | Communication |
                            |    Session    |
                            +---------------+
                         0..* < >       < > 1..*
                               |         |
                    +----------+         +--------+
               1..* |                             | 1..*
             +-------------+ 1     0..* +--------------------+
             | Participant |<>----------| Multimedia Session |
             +-------------+            +--------------------+
                   < > 1                         < > 1
                    |                             | 0..*
                    |                      +-------------+
                    |                      | RTP Session |
                    |                      +-------------+
                    |                            < > 1
                    | 0..*                        | 0..*
             +-----------------+ 1   0..* +--------------+
             | Media Transport |--------<>| Packet Stream|
             +-----------------+          +--------------+
```

Figure 13: Session Relations

Several different flavors of Session can be possible. A few typical examples are listed in the below sub-sections, but many other are possible to construct.

4.2.1. Point-to-Point Session

In this example, a single Multimedia Session is shared between the two Participants. That Multimedia Session contains a single RTP

Session with two Media Streams from each Participant. Each Participant has only a single Media Transport, carrying those Media Streams, which is the main reason why there is only a single RTP Session.

```
                            +-----------------+
                            | Point-to-Point  |
                            |     Session     |
                            +-----------------+
                              < >   < >   < >
                               |     |     |
           +-------------------+     |     +-------------------+
           |                         |                         |
      +-------------+     +--------------------+      +-------------+
      | Participant |<>---| Multimedia Session |----<>| Participant |
      +-------------+     +--------------------+      +-------------+
         < >                        < >                        < >
          |                          |                          |
          |  +-----------+     +-----------+     +-----------+  |
          |  |  Media    |---<>|    RTP    |<>---|  Media    |  |
          |  |  Stream   |     |  Session  |     |  Stream   |  |
          |  +-----------+     +-----------+     +-----------+  |
          |   < >               < >     < >               < >   |
          |    |                 |       |                 |    |
    +-----------+      +-----------+   +-----------+      +-----------+
    |   Media   |----<>|  Media    |   |  Media    |<>----|   Media   |
    | Transport |      |  Stream   |   |  Stream   |      | Transport |
    +-----------+      +-----------+   +-----------+      +-----------+
```

Figure 14: Example Point-to-Point Session

4.2.2. Full Mesh Session

In this example, the Full Mesh Session has three Participants, each of which has the same characteristics as the example in the previous section; a single Media Transport per peer Participant, resulting in a single RTP session between each pair of Participants.

```
   +-----------+              +-------------+              +-----------+
   |   Media   |------------<>| Participant |<>------------|   Media   |
   | Transport |              +-------------+              | Transport |
   +-----------+                     |                     +-----------+
       |      |         +----------+ | +----------+         |      |
      < >    < >        |Multimedia| | |Multimedia|        < >    < >
   +--------++--------+ |  Session | | | Session  | +--------++--------+
   | Media  || Media  | +----------+ | +----------+ | Media  || Media  |
   | Stream || Stream |  < >    |    |    |    < >  | Stream || Stream |
   +--------++--------+   |     |    |    |     |   +--------++--------+
```

```
       |           |      |     |    |    |     |      |          |
       |          < >     |    < >  < >  < >    |     < >         |
       |         +---------+ +--------------+  +---------+        |
       +-------<>|   RTP   | |  Full Mesh   |  |   RTP   |<>------+
       +-------<>| Session | |   Session    |  | Session |<>------+
       |         +---------+ +--------------+  +---------+        |
       |          < >         < >   < >   < >      < >            |
       |           |           |     |     |        |             |
   +--------++--------+        |     |     |        +--------++--------+
   | Media  || Media  |        |     |     |        | Media  || Media  |
   | Stream || Stream |        |     |     |        | Stream || Stream |
   +--------++--------+        |     |     |        +--------++--------+
      < >    < >               |     |     |               < >    < >
       |      |                |     |     |                |      |
   +-----------+               |     |     |               +-----------+
   |   Media   |               |     |     |               |   Media   |
   | Transport |               |     |     |               | Transport |
   +-----------+ +-------------+     |     +-------------+ +-----------+
                 |                   |                   |
   +-------------+        +--------------------+         +-------------+
   | Participant |<>------| Multimedia Session |-------<>| Participant |
   +-------------+        +--------------------+         +-------------+
     < >                            < >                            < >
      |                              |                              |
      |   +--------+           +---------+          +--------+     |
      |   | Media  |---------<>|   RTP   |<>---------| Media  |     |
      |   | Stream |           | Session |          | Stream |     |
      |   +--------+           +---------+          +--------+     |
      |    < >               < >     < >               < >      |
      |     |                   |       |                   |       |
   +-----------+           +--------+ +--------+           +-----------+
   |   Media   |---------<>| Media  | | Media  |<>---------|   Media   |
   | Transport |           | Stream | | Stream |           | Transport |
   +-----------+           +--------+ +--------+           +-----------+
```

Figure 15: Example Full Mesh Session

4.2.3. Centralized Conference Session

Text to be provided

TBD

Figure 16: Example Centralized Conference Session

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, Colin Perkins, Keith Drage, and Harald Alvestrand.

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. Also the rewriting of the relationships he has contributed.

8. Open Issues

Much of the terminology is still a matter of dispute.

It might be useful to distinguish between a single endpoint's view of a source, or RTP session, or multimedia session, versus the full set of sessions and every endpoint that's communicating in them, with the signaling that established them.

(Sure to be many more...)

9. IANA Considerations

This document makes no request of IANA.

10. References

10.1. Normative References

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

[UML] Object Management Group, "OMG Unified Modeling Language (OMG UML), Superstructure, V2.2", OMG formal/2009-02-02, February 2009.

10.2. Informative References

[I-D.ietf-avtcore-clksrc]
Williams, A., Gross, K., Brandenburg, R., and H. Stokking, "RTP Clock Source Signalling", draft-ietf-avtcore-clksrc-06 (work in progress), September 2013.

[I-D.ietf-clue-framework]
Duckworth, M., Pepperell, A., and S. Wenger, "Framework for Telepresence Multi-Streams", draft-ietf-clue-framework-11 (work in progress), July 2013.

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Multiplexing Negotiation Using Session Description Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-bundle-negotiation-04 (work in progress), June 2013.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-08 (work in progress), September 2013.

[I-D.westerlund-avtcore-transport-multiplexing]
Westerlund, M. and C. Perkins, "Multiple RTP Sessions on a Single Lower-Layer Transport", draft-westerlund-avtcore-transport-multiplexing-06 (work in progress), August 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.

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

[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.

[RFC6222] Begen, A., Perkins, C., and D. Wing, "Guidelines for Choosing RTP Control Protocol (RTCP) Canonical Names (CNAMEs)", RFC 6222, April 2011.

Appendix A. Changes From Earlier Versions

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

A.1. Changes From Draft -00

o Too many to list

- Added new authors

- Updated content organization and presentation

A.2. Changes From Draft -01

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

- Section 3 rewritten to focus on exposing relationships.

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
Farogatan 6
SE-164 80 Kista
Sweden

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