Codec agnostic RTP payload format

Sergio Garcia Murillo

https://www.ietf.org/archive/id/draft-gouaillard-avtcore-codec-agn-rtp-payload-00.txt

RFC7656 Media Chains are modified by SFrame

- Media Packetizer transforms a single Encoded Stream into one or several RTP packets.
- The Encoded Stream is coming straight from the Media Encoder and is expected to follow the format produced by the Media Encoder.
- SFrame End-to-end encryption is implemented by inserting application-specific Media Transformers between Media Encoder and Media Packetizer.
- Current Media Packetizer formats do not support this Transformed stream input and a per-codec “hack” is required: https://datatracker.ietf.org/meeting/109/materials/slides-109-sframe-video-payloads-usage-with-sframe-00
- The generic packetization is intended to solve this problem by defining a new Media Packetizer that works with media content in a codec-agnostic way.

```
       Physical Stimulus
              |
              V
+----------------------+
|     Media Capture    |
+----------------------+
              |
          Raw Stream
              V
+----------------------+
|     Media Source     |<-- Synchronization Timing
+----------------------+
              |
         Source Stream
              V
+----------------------+
|     Media Encoder    |
+----------------------+
              |
        Encoded Stream
              V
+----------------------+
|  Media Transformer   |<-- NEW: application-specific transform
+----------------------+         (e.g. SFrame Encryption)
              |
   Transformed Stream     +------------+
              V           |            V
+----------------------+  | +----------------------+
|  Media Packetizer    |  | | RTP-Based Redundancy |
+----------------------+  | +----------------------+
              |           |            |
              +-----------+  Redundancy RTP Stream
       Source RTP Stream               |
              V                        V
+----------------------+    +----------------------+
|  RTP-Based Security  |    |  RTP-Based Security  |
+----------------------+    +----------------------+
              |                        |
   Secured RTP Stream   Secured Redundancy RTP Stream
              V                        V
+----------------------+    +----------------------+
|   Media Transport    |    |   Media Transport    |
+----------------------+    +----------------------+
```

The codec agnostic RTP payload format

MUST NOT DO

- This generic packetization must not change how the mapping between one or several encoded or dependant streams are mapped to the RTP streams or how the synchronization sources(s) (SSRC) are assigned.
- Modify the behavior of the redundancy or recovery mechanism. That is, RTX, RED, ULPFEC, FLEX FEC must work with the codec agnostic RTP out of the box without any modifications.
- Must not interfere with BWE or CC, RMCAT RTCP FB, transport wide CC or REMB must work without any modifications.

MUST DO

- Simulcast must work.
- Support Single RTP stream on a Single media Transport (SRST) when Scalable Video Coding (SVC) is in use.
- Negotiation is done via SDP.
- Allow SFUs to perform layer selection and packet forwarding.

SDP negotiation

```
a=rtpmap:96 vp9/90000
a=rtpmap:97 vp8/90000
a=rtpmap:98 generic/90000
a=rtpmap:99 rtx/90000
a=fmtp:99 apt=96
a=rtpmap:100 rtx/90000
a=fmtp:100 apt=97
a=rtpmap:101 rtx/90000
a=fmtp:101 apt=98
```

- Generic packetizer negotiation relies on negotiation of the standard packetizer for the each codec.
- Negotiate a generic payload type for all codecs in order to avoid duplication or triplication the number of payload types in use.
- Requires sending actual codec payload type, as a RTP header extension or as a prefix to the payload, which will cause minor network overhead.
- Requires negotiating different payload types for each clock rate for audio.

RTP Packetization

- When the packetizer receives a frame from the application, it MUST fragment the frame content in multiple RTP packets to ensure packets do not exceed the network MTU.
- The content of the frame will be treated as a binary blob by the packetizer, so the decision about the boundaries of each fragment is decided arbitrarily by the packetizer.
- The marker bit of each RTP packet in a frame MUST be set according to the audio and video profiles specified in RFC3551.
- In the case of a video codec supporting spatial scalability, each spatial layer MUST be split in its own frame by the application before passing it to the packetizer.
- The spatial layer frames are sent in ascending order, with the same RTP timestamp, and only the last RTP packet of the last spatial layer frame will have the marker bit set to 1.

Payload Type Multiplexing

- In order to reduce the number of payload type in the SDP exchange, a single payload type is used all negotiated media formats.
- That requires to identify the original payload type code of the negotiated media format, called the associated payload type (APT).
- The APT value is the payload type code of the associated format passed to the generic Media Packetizer before any transformation is applied.
- The APT value is sent in a dedicated header extension:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID   | len=0 |S|     APT     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      ID       |     len=1     |S|     APT     |    0 (pad)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- The APT value is the associated payload type value.
- The S bit indicates if the media stream can be forwarded safely starting from this RTP packet. It will be set to 1 on the first RTP packet of an intra video frame and in all RTP audio packets.
- Receivers MUST be ready to receive RTP packets with different associated payload types in the same way they would receive different payload type codes on the RTP packets.
- The URI for declaring this header extension in an extmap attribute is **`urn:ietf:params:rtp-hdrext:associated-payload-type`**

Frame metadata

Potential metadata of interest

- At which packet SFU can switch (SVC and simulcast).
- Resolution and more generally stream 'quality': frame rate, bit rate...
- Codec specific information like profile/levels.
- Recovery mechanism required in case of loss (none, RTX, LRR/PLI).
- Opus TOC to know frame length (recording scenarios).

Potential solutions

- Use/extend frame marking.
- Use AV1 Dependency Descriptor (current proposed solution).
- Design a new RTP header extension, potentially complemented with either frame marking or AV1 Dependency Descriptor.