RTP Payload Format for HTTP Adaptive Streaming

draft-wei-payload-has-over-rtp-01

ietf97@Seoul

Background

- Paid video service is becoming a revenue driver for operators.
- The draft aims at providing some convergence of OTT & IPTV, to help deliver better video service.

OTT Video

HTTP Adaptive Streaming (e.g. DASH/HLS) are widely adopted.

Managed IPTV Network

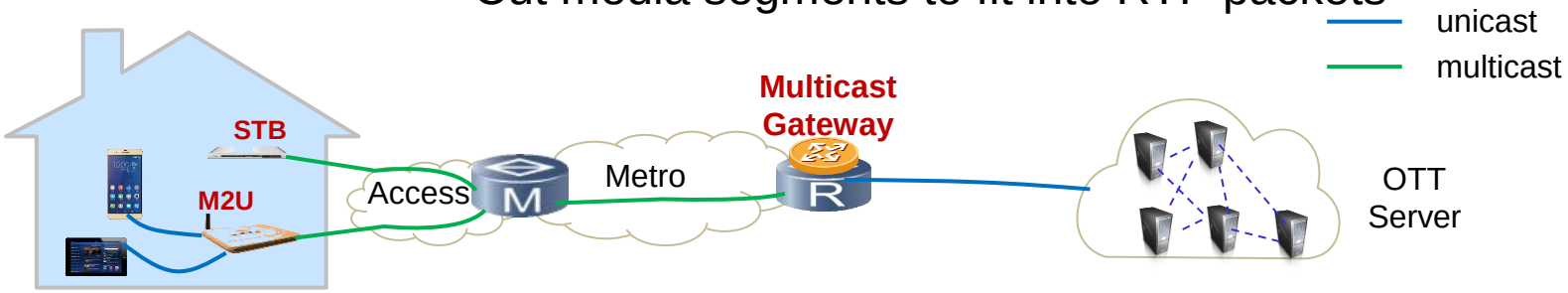
Already uses RTP over IP Multicast, with features (e.g. FCC, FEC, RET) to ensure Quality of Experience.

Progress since IETF96

- Architectural work has been initiated in the Broadband Form.
- Updates to the draft
  - Revised the architecture and use scenarios
  - Updated some fields in the payload format

Proposed Architecture

- Pull adaptive streams from OTT Server
- Put each stream to a multicast group
- Cut media segments to fit into RTP packets



- Multicast to STB (multicast receiver)
- Multicast to M2U (RTP translator), then convert multicast to RTP unicast to end devices (unicast receivers).

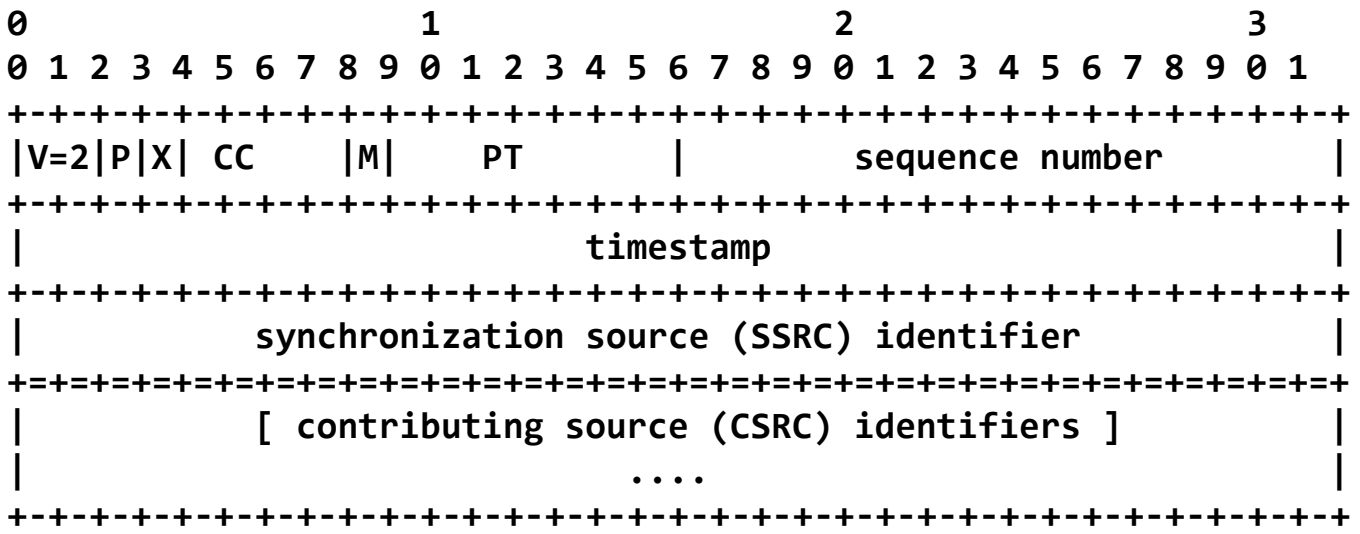
Trade-offs are made for live video of high quality (e.g. 4K UHD).

- Multicast Gateway: reduced complexity to ensure low delay and high TPT
- Receivers: may require function update

Why HAS Over RTP?

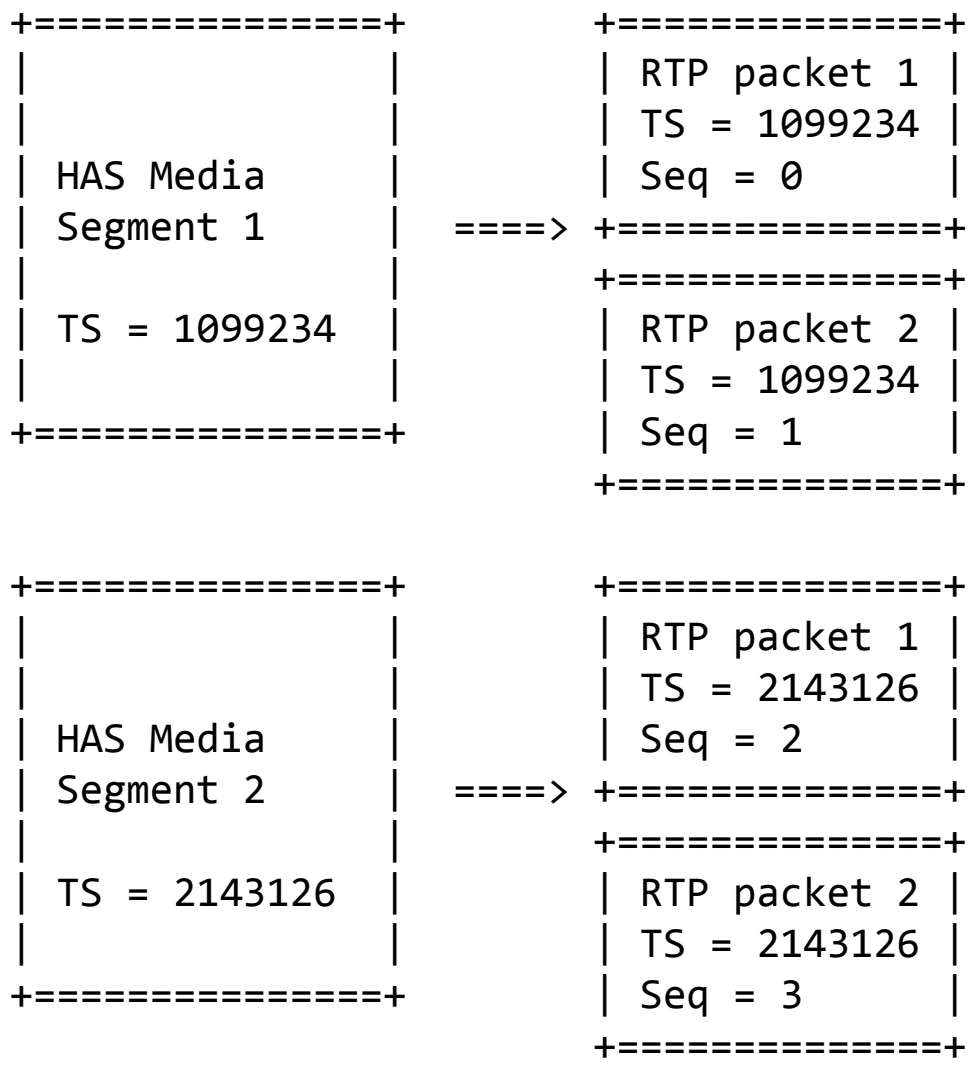
- We want to simplify the processing of the multicast gateway:
  - Only needs to parse manifest files (e.g. DASH .mpd , HLS .m3u8) and packetize media segments.
  - Don’t need to concern too much about the specific packaging formats of the media segments.
- Using existing RTP payload formats:
  - Maybe compatible with existing clients.
  - Added complexity because the multicast gateway may have to parse the specific media segments.

RTP Header

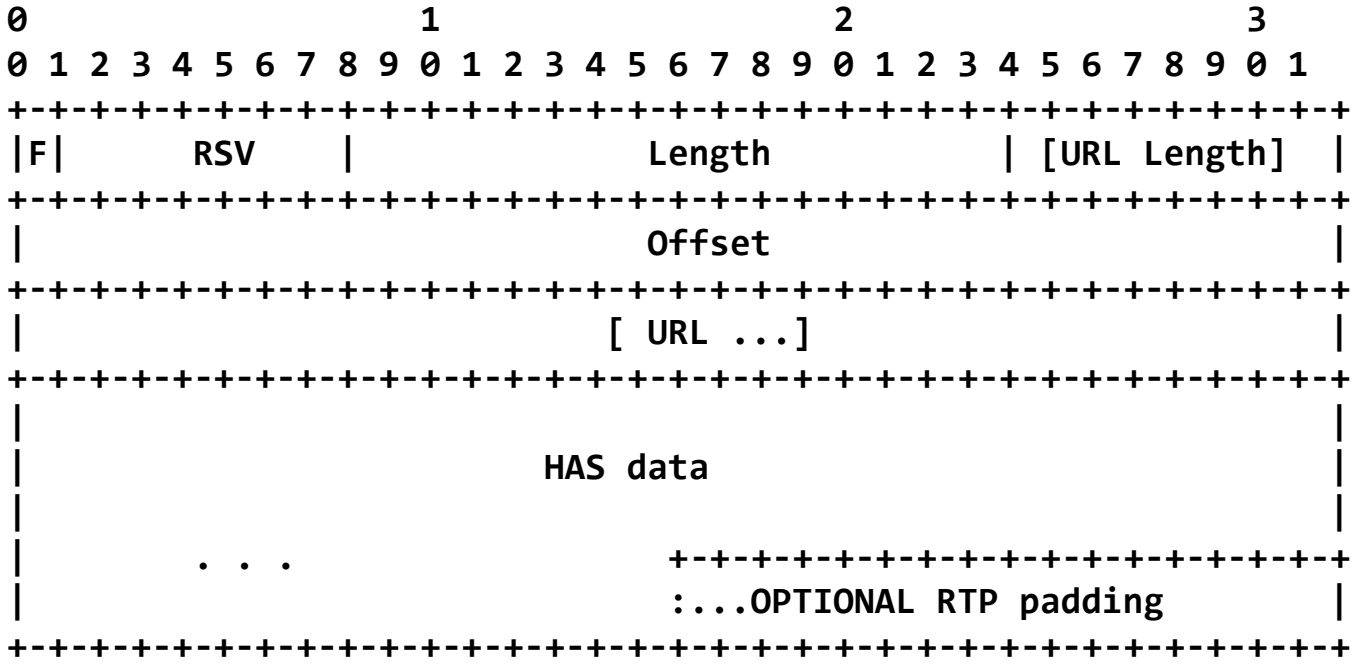


- Marker (M): set to 1 to indicate the last fragment of a media segment
- Payload Type (PT): set accordingly to the type used
- Sequence Number: increment by one per fragment
- Timestamp: same for all fragments of a media segment

Packetization Examples



HAS Payload



- Fragmentation (F): set to 1 to indicate it is a fragment of a media segment.
- URL: to help relate a packet to the URL of a media segment
- (added) Offset: to help locate the fragment in the media segment
- (removed) Type: 0=Manifest, 1=Initial Information, 2=Media Seg.

Fragmentation Considerations

- Straightforward way
  - blindly cut the media segments to fit into the MTU
  - less resilient to packet loss, one lost of fragment can lead to the whole media segment undecidable,
- Intelligent way
  - If hints are provided to the multicast gateway, it can repack media segments into smaller decodable pieces, then fragmentation may be avoided
  - Added complexity, but more resilient to packet loss, the processing can be expensive
- Trade-off should be made to balance between packet loss and complexity in the multicast gateway.

Next Step

- Comments & suggestions?
- WG Adoption?