

RTP Payload Format for High-Efficiency Video Coding

[draft-ietf-payload-rtp-h265-00.txt](#)

Ye-Kui Wang

yekuiw@qti.qualcomm.com

Yago Sanchez

yago.sanchez@hhi.fraunhofer.de

Thomas Schierl

thomas.schierl@hhi.fraunhofer.de

Stephan Wenger

stewe@stewe.org

Miska Hannuksela

miska.hannuksela@nokia.com

High Efficiency Video Coding (H.265/HEVC)

- ITU H.265 published in Apr. 2013, ISO-IEC to follow after procedural delay
- Adoption of RTP-based H.265 into application standards in progress
 - 3GPP SA4 agreed the support of HEVC in 3GP-DASH, support in PSS, MBMS, MTSI, and MMS expected to be decided soon
 - 3GPP's HEVC time plan expects the RTP payload RFC to be available early 2014
 - DVB is also considering the support of RTP-based HEVC in DVB specifications
- draft-ietf-payload-rtp-h265 is a WG draft since July 1st, 2013

Open Issues (Overview)

- **Add max-fps**
 - Indicating the maximum frame rate in units of hundredths of frames per second that can be efficiently received
 - Some discussion in payload reflector – not concluded due to summer vacation season
 - Editor proposal: include in draft
- **Improved parallel processing signaling**
 - Proposals received from from two parties:
 - The two parties have been working with the editors at the Vienna JCT-VC meeting to devise a harmonized solution, to be discussed in this session
 - Editor proposal: include harmonized proposal into the draft
- **New FU type**
 - Suggested by Mo Zanaty, Thomas Davies, and Paul Bright-Thomas (Cisco)
 - Quite some discussions on the payload reflector, little/no support but from the proponents due to unclear/unconvincing use case
 - Editor proposal: Do NOT include proposal into the draft
- **Temporal error resilience (parts of RFC 6190 PACSI concept)**
 - Editor proposal: wait for input

Open Issue 1: max_fps

- max-fps indicates the maximum frame rate in units of hundredths of frames per second that can be efficiently received
- Purpose: dial-down the frame rate allowed based on signaled H.265 profile/level
 - H.265 would allow very high frame rate for a given level if the picture size is small enough—max_fps sets a limit.
 - Similar mechanism present in RFC 3984/RFC6184 for the same reasons.
- Some discussion in payload reflector – no conclusion (perhaps due to summer vacation season)
- Editor proposal: include
- Can we close this now?

Open issue 2: Signaling of parallelization tools

- Two sets of suggestions from two parties: {Tom Kristensen} and {Rickard Sjoberg, Bo Burman, and Magnus Westerlund}
- The two parties have been working together with the editors to get a harmonized solution

Signaling of parallelization tools: Sketch

- Rename segmentation_id to sprop segmentation_id and spatial-segmentation-id to sprop-spatial-segmentation-idc to better characterize their use in the SDP
 - Otherwise unchanged semantics

- Add dec-parallel-cap with ABNF as follows:

```
dec-parallel-cap = "dec-parallel-cap={" cap-point *("," cap-point) "}"
```

```
cap-point = ("w" / "t" spatial-seg-idc) 1*5";" cap-parameter)
```

```
spatial-seg-idc = 1*4DIGIT ; 1-4095
```

```
cap-parameter = tier-flag / level-id / max-ls / max-lps / max-br
```

- Examples

- a=fmtp:98 level-id=93;dec_parallel_cap={t8;level-id=120}

indicates capability of decoding 720p30 without parallelization, but 1080p30 with tiles t8 means that the max tile size must be at most PictureSize/3 samples

- a=fmtp:98 level-id=93;dec_parallel_cap={w;max-ls=2088960;max-lps=62668800}

indicates capability of decoding 720p30 without parallelization, but 1080p30 using max-ls and max-lps, and wavefronts are used in the stream

Redesign of Fragmentation Units

- At present, Aggregation and Fragmentation as in RFC 3984
- Proposed by Mo Zanati by email 7/3/2013
 - Allows for aggregation of NAL units with fragments of other NAL units, though offsets in the fragmentation header
- Lengthily email discussions on payload@ietf.org
- Authors did not sense a need to include Mo's proposal
 - Lack of compelling use case
 - Unknown impact of change to overall design

Inclusion of Temporal Scalability Error Resilience Information

- Topic came up in Vienna (though authors were contemplating something like this before at individual draft stage)
- Include RFC6190 PACSI NAL unit fields S, E, TLOPICIDX, PDRPICID into H.265 payload
- Perhaps find a way to include this info in payload without PACSI NAL unit construct (has certain advantages in conjunction with aggregation and fragmentation)
- This is a heads-up only; a draft is expected shortly.

Thanks

Back up slides

- Differences relative to RFC 6184 (two slides)
- More detailed example of parallelization signaling (one slide)

Compared to RFC 6184 (1/2)

- No multiple packetization modes – just one now
- No support of multi-time aggregation packets (MTAPs)
- Single NAL unit packet still supported
 - With optional inclusion of decoding order number (DON)
- (Single-time) aggregation packets (APs) still supported
 - With optional inclusion of DON
- Fragmentation unit (FU) still supported
 - Only one type
 - With optional inclusion of DON

Compared to RFC 6184 (2/2)

- Multi-session transmission (MST) is now supported as HEVC has a complete design for temporal scalability support
- A single de-packetization process applies regardless of whether MST or single-session transmission (SST) is in use, and regardless of whether interleaved packetization is in use
 - Expected to work also for HEVC scalable and 3D extensions
- Includes the HEVC specific use with feedback messages as specified in AVPF (RFC 4585) and CCM (RFC 5104)
- Included media type parameters "segmentation-id", and "spatial-segmentation-idc" for parallel processing

Example

mandatory base
capability

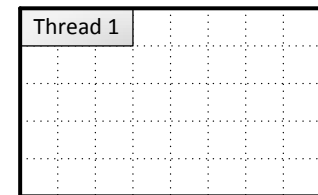
optional parallel
capability #1

optional parallel
capability #2

a=fmtp:98 level-id=93;**dec_parallel_cap**={t8;level-id=120,w;max-ls=2088960;max-lps=62668800}

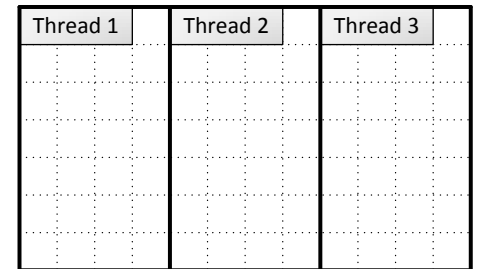
•The transmitter can send a 720p30 stream which is not made parallel since:

- level-id=93 indicates that HEVC level 93/30 = 3.1 is supported
- Level 3.1 supports 720p30



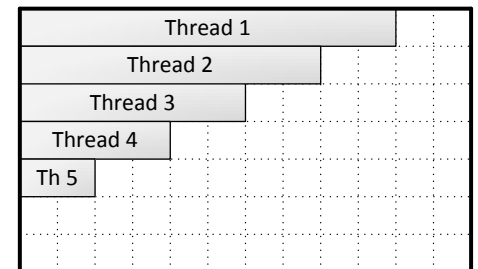
•The transmitter can send a 1080p30 stream with 3 equally sized tiles since:

- t is a condition that slices or tiles must be used
- 8 specifies a largest tile size of $4 * \text{PicSize} / (8+4) = \text{PicSize} / 3$
- Level 120/3 = 4 supports 1080p30



•The transmitter can send a 1080p30 stream with wavefronts since:

- w is a condition that wavefronts must be used
- 1080p30 fits within max-ls=2088960;max-lps=62668800



•The optional parameters are: tier-flag, level-id, max-ls, max-lps, and max-br

spatial_segmentation_idc

Informative note: When `spatial_segmentation_idc` is present in a stream and wavefronts are not used, [HEVC] specifies that there will be either no slice or no tile in the stream with more than $4 * \text{PicSizeInSamplesY} / (\text{min_spatial_segmentation_idc} + 4)$ luma samples.

- For t8 (as in the example) that means:
- the maximum tile size in samples is $4 * 1080 / (8 + 4) == 360$
- $1080 / 3 == 360$