

AHG Report: High-Level Syntax and Error Resilience

Ye-Kui Wang (Nokia)

Stephane Pateux (Orange -France Telecom Division R&D)

Peter Amon (Siemens Corporate Technology)

Thomas Schierl (Heinrich Hertz Institute (FhG))

Mandates

- To study ROI information signaling, particularly for interactive ROI
- To study enhancements to scalability information SEI message
- To study decoded picture buffer management taking to account key pictures and AR-FGS
- To study coded picture buffer modeling
- To consider SVC restrictions
- To refine the error resilience test conditions if needed
- To study error resilience in scalable video applications
- To build error resilient simulation environment
- To generate JSVM error/loss resilience anchor bitstreams
- Consider high-level syntax design
- Consider 'simple_priority_id' in the NAL unit header, especially related to the possibility of a mapping between DTQ and 'simple_priority_id' and the possibility of further restrictions

E-Mail discussion on 'simple_priority_id'

- Several solutions were discussed
 - A) Not specifying any constraints for 'simple_priority_id'
 - B) Constraint on the validity of the bitstream
 - C) Specifying a mapping between DTQ and 'simple_priority_id'
 - D) Supporting several use cases: B) and C)

A) Not specifying any constraints

- Leaves the definition of the use of ‘simple_priority_id’ open to other systems layers (e.g. file format, transport format)
- Interaction between application areas could be harmed, if the semantics of ‘simple_priority_id’ is not specified uniquely.

B) Constraint on the validity of the bitstream

- Constraint: "Priority_ids MUST be assigned in such a way that, given a valid stream with N being the highest priority_id, any subset of that stream which includes all NALUs with priority_id $\leq M$, where $M \leq N$, must also be a valid stream."
- One-dimensional adaptation path, if only 'simple_priority_id' is used for adaptation
- Leads to a “‘naïve’ thin” of the bit-stream.

C) Specifying a mapping between DTQ and 'simple_priority_id'

- There are two sub-cases
 - One-to-many mapping: $PRID = f(DTQ)$
 - One-to-one mapping:
 $PRID = f(DTQ); DTQ = f^{-1}(PRID)$
- One-to-many mapping is preferable, since it naturally define layers and can be combined with case B.
- Pre-defined spatio-temporal (and also quality_id) layers can be created e.g. to address different service classes (maybe only two).
- Concept of such mapping is not compatible with the concept of 'quality layers'.

D) Supporting several use cases

- Especially supports the two use cases B and C:
 - Case C: 1D mapping, where a pre-computed layered version of the stream can be easily created, e.g. with only a few layers
 - Case B: No mapping with a few constraints ensuring the validity of the stream: this supports multidimensional adaptation and also quality layers.
- Reassigning the 'reserved_zero_bit' e.g. as a 'fixed_mapping_flag' in order to switch between the two constraints (more precisely: extent the constraint B to constraint C)
- In principle the solution in JSVM 5, avoiding the disadvantage of the short headers, where the mapping of 'simple_priority_id' to DTQ is stored in a parameter set.
- If case B is seen as a minimum constraint, only valid streams are extractable by adaptations only based on 'simple_priority_id'.
- Specifying the behavior and constraints of the SVC stream on high-level syntax assures that a SVC stream is compatible between different application areas. Even a 'translation/transcoding' between the two 'use cases' is possible, since the signaling, which use case applies, is explicitly done in the NAL unit header

Recommendations

- To review all the contributions during the meeting
- To call for studies and/or implementations in the following aspects:
 - coded picture buffer modeling
 - coding of multiple slices per picture
 - slice coding mode of fixed number of bytes
- To specify a solution for stream thinning for layered scalability (priority based or any other solution for stream thinning) and combined scalability using the NAL unit header