Software defined ultra-low latency Video-over-IP system with compression

Siegfried Foessel and Thomas Richter, Fraunhofer IIS, Germany

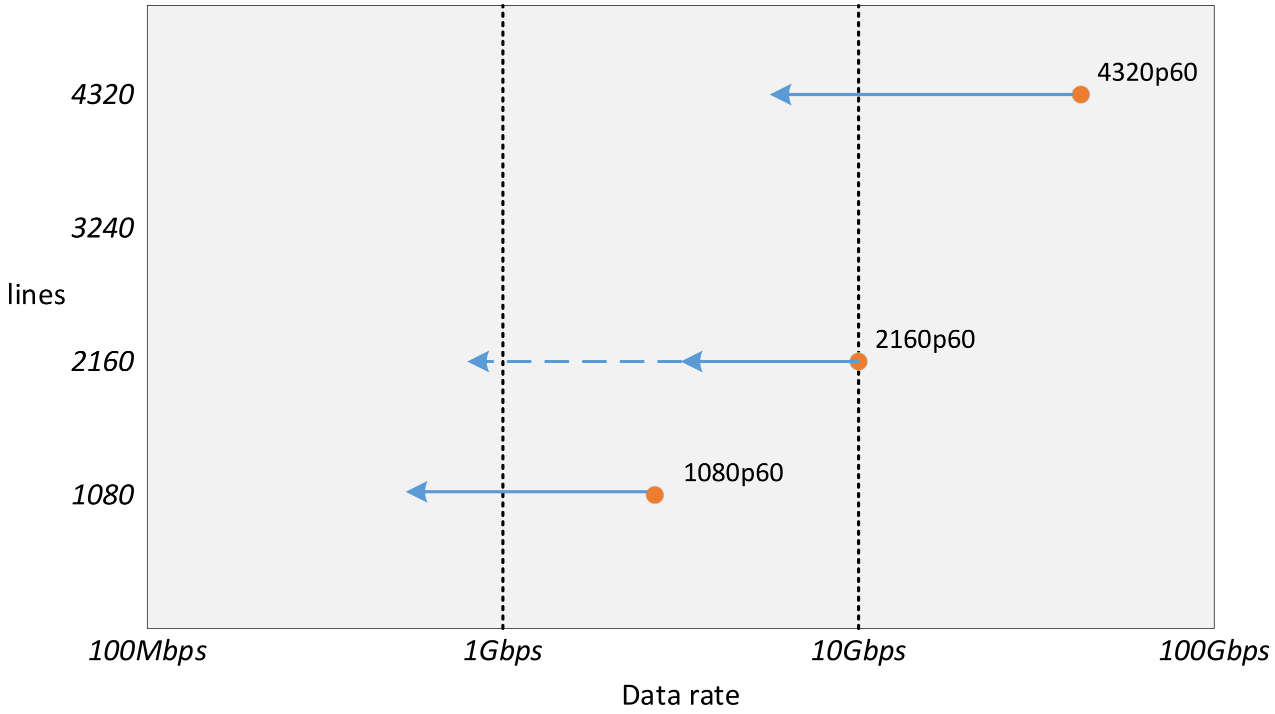
Fraunhofer
IIS

Focus of This Presentation

- Video over IP transmission for production (typically ST 2110)
- Use of low-latency image compression codecs to reduce necessary bandwith
- JPEG XS as a mezzanine codec
- Design of an ultra-low-latency video transmission system in software only (in the range of 10ms)

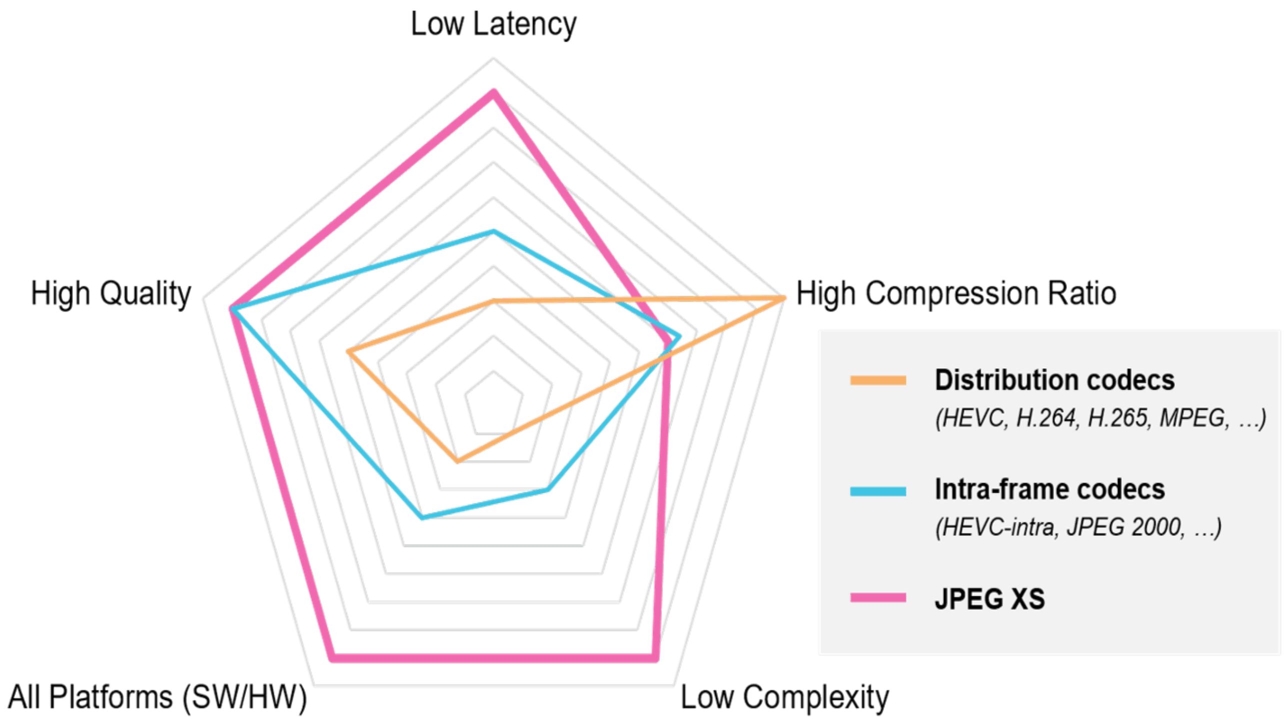
Video Over IP for Production

- Trend to go from SDI lines to IP networks
- Uncompressed video data transmission requires high bandwidth on transmission channel
- Mezzanine compression allows use of lower cost equipment and lower bandwidth transmission channels



Video Over IP for Production

- However, compression causes latencies
- Latency depends on
  - Spatial-Frequency Transform
  - Intra-/interframe decorrelation
  - Complexity/Processing Time
  - Area for Rate/Distortion Calculation



JPEG XS (ISO/IEC 21122)

- JPEG XS is a new standardised codec for low latency video transmission
  - Specifically **designed and tailored** for IP transmission with SMPTE ST2110
  - To achieve low complexity, the codec was only designed for compression ratios from lossless to 10:1 (mezzanine compression)
  - Allows reducing the necessary bandwidth for IP transmission, but still ensures visual lossless quality
- JPEG XS is highly parallizable on different granularities

Key Features of JPEG XS

- Slice based processing (typ. 16 lines)
- Max. 32 lines latency for encoder + decoder (algorithmicwise)

Low Latency

- Predictive
- Constant bit rate
- No frame drops

Rate Control

- Visually lossless or mathematically lossless
- Multi-generation robust

High Quality

- 4:4:4, 4:2:2 and 4:2:0
- CFA (Raw) Compression
- Up to 16 bit per component

Wide Parameter Set

- ISO/IEC 21122 (JPEG XS)
- RFC 9136 (RTP for JPEG XS)
- ST 2124 (MXF for JPEG XS)
- MPEG-TS, MP4, HEIF, JXS

Interoperable, Standardized

- 4k 60p realtime on i7 processor
- 8k 60p realtime on Epyc 2 processor

Low Complexity

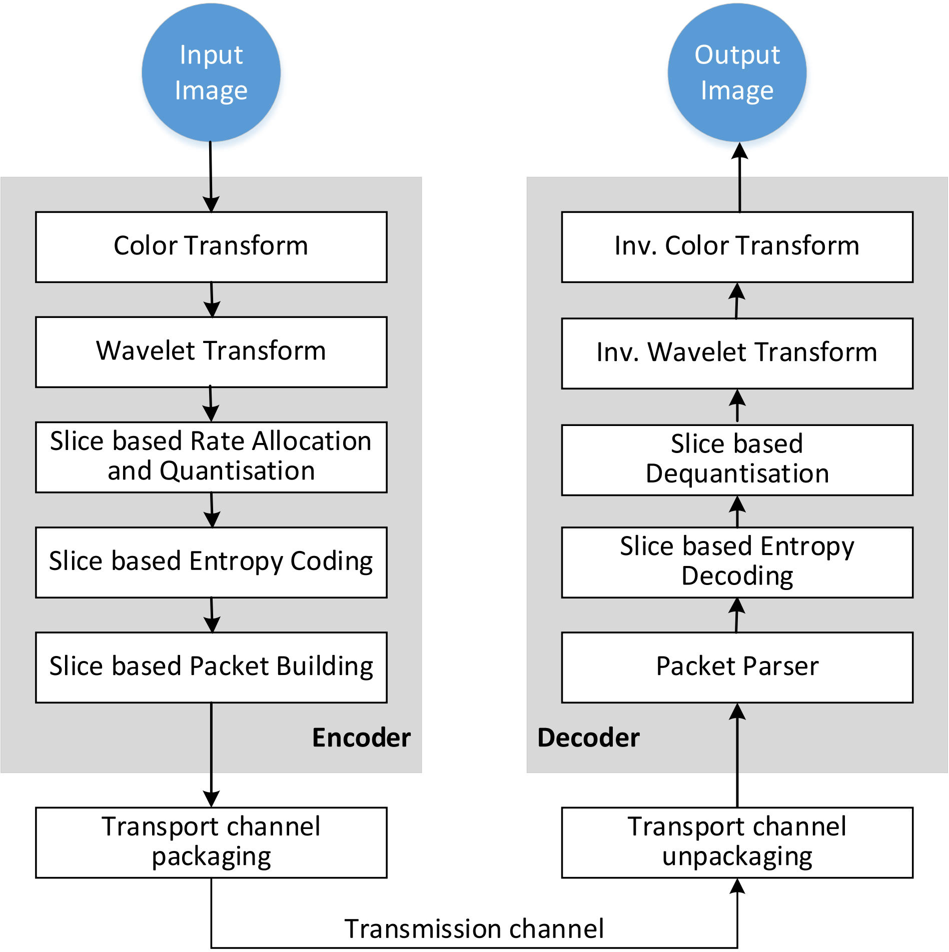
- For all platforms (FPGA, CPU,GPU, ASIC)
- Fine grained (GPU)
- Coarse grained (CPU, FPGA)

Highly parallelizable

- Tolerant against bit flip errors
- Many resync points

Error robust

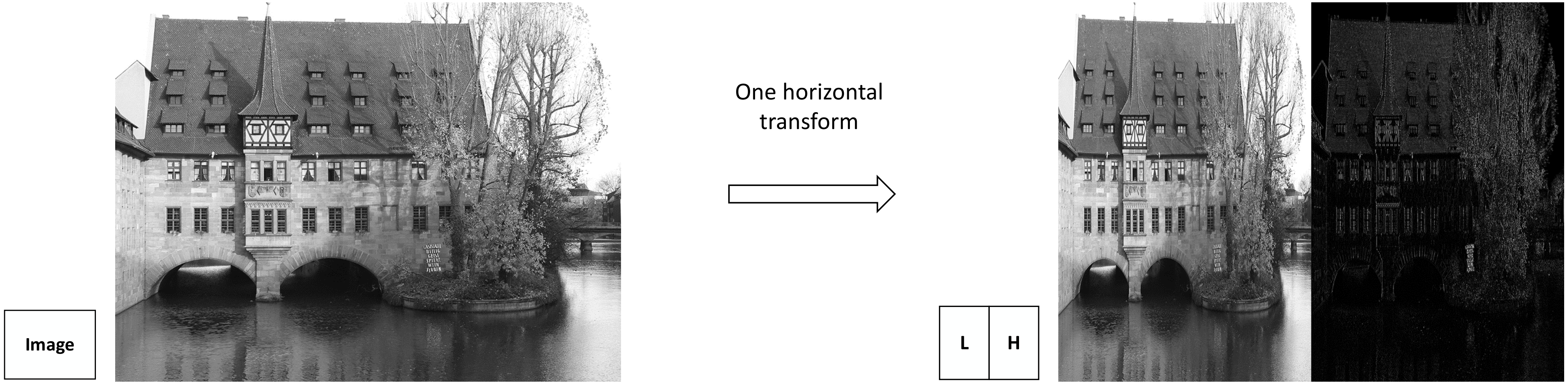
JPEG XS Codec Pipeline



Input Image
Color Transform
Wavelet Transform
Slice based Rate Allocation and Quantisation
Slice based Entropy Coding
Slice based Packet Building
Encoder
Transport channel packaging
Transmission channel
Transport channel unpackaging
Decoder
Packet Parser
Slice based Entropy Decoding
Slice based Dequantisation
Inv. Wavelet Transform
Inv. Color Transform
Output Image

JPEG XS Wavelet Transform

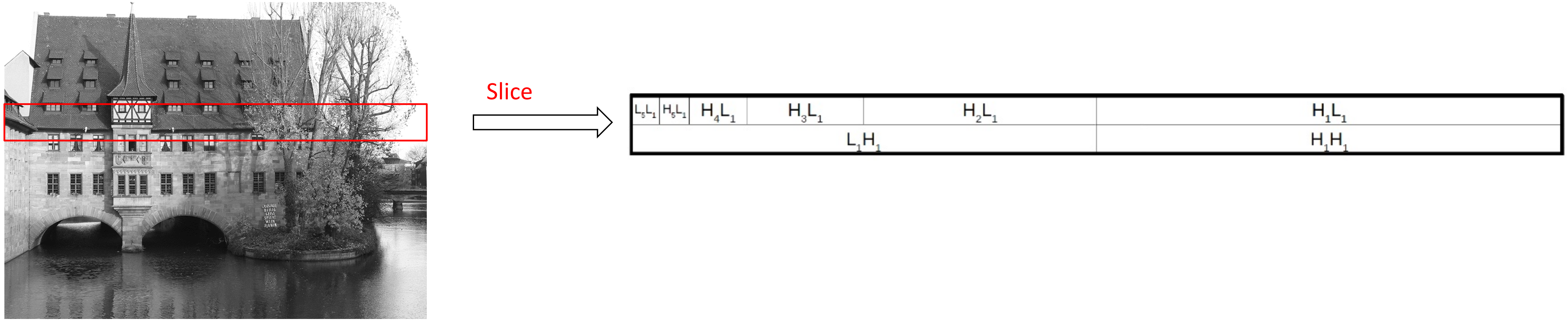
- Wavelet transform decomposes images in low and high frequency components with the goal of an energy compaction



- This will be done multiple times, with JPEG XS up to 5x in horizontal direction and up to 2x in vertical direction

JPEG XS Wavelet Transform

- After wavelet transform image regions/slices (typically 16 lines high) are represented as coefficents in multiple frequency subbands



- **Important: Slices can be coded independently!**

JPEG XS Performance Data

- Actual x86 CPUs:

Performance (Proc. factor) per Core on 3GHz	HD 422	UHD-1 (4k) 422	UHD-2 (8k) 422
Encoding	30-35 fps (2.0 … 1.7)*	8.5-10.5 fps (7.1 … 5.7)*	2.5-3.0 fps (24.0 … 20.0)*
Decoding	45-60 fps (1.3 … 1.0)*	11.5-16fps (5.2 … 3.7)*	3.0-4.0 fps (20.0 … 15.0)*

Hyperthreading improves performance per core between 1.2..1.5

*Processing factor is the ratio between real-time transmission at 60 fps and processing time,
a processing factor of 2 means the processing for encoding or decoding needs twice the time as the data comes in

JPEG XS Performance Data

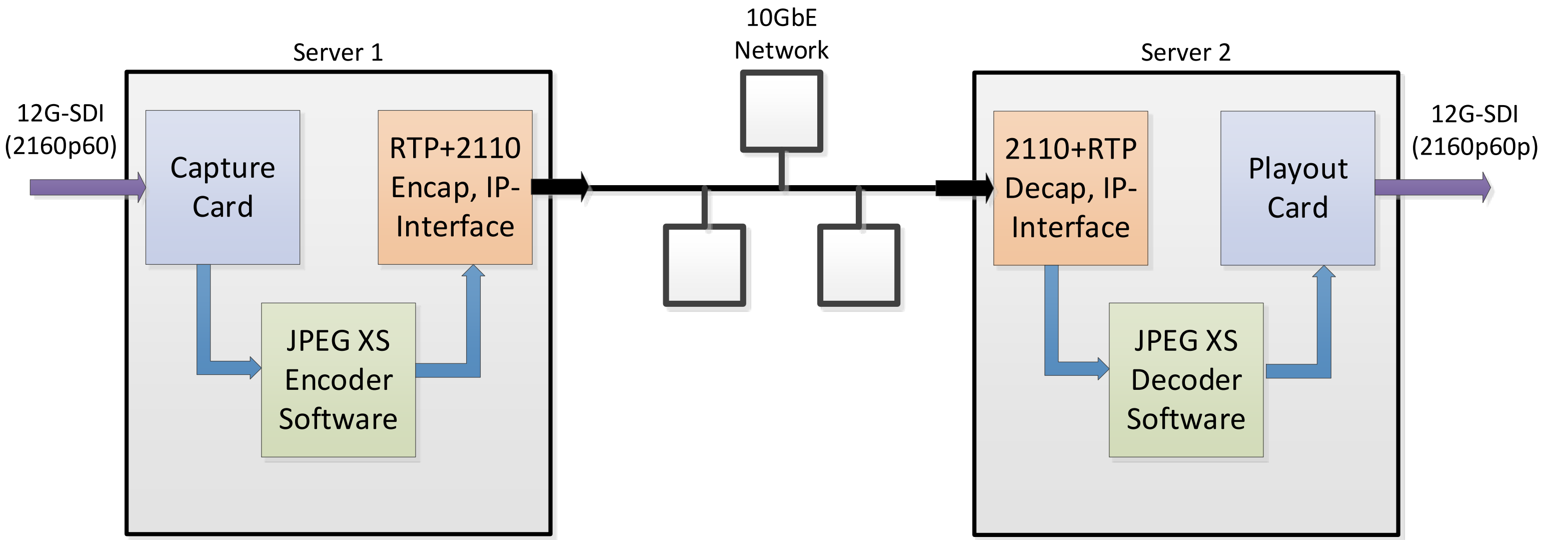
- Dependent on image size,
  the number of slices per image changes
- Slices can be processed individually by threads
- To reduce thread management overhead, multiple consecutive slices should be grouped together to a **slicegroup**
- For comparison
  frame-to-frame delta @ 60fps = 16.66ms

Image size	No. of slices
1920x1080	68 (67,5)
3840x2160	135
7680x4320	270

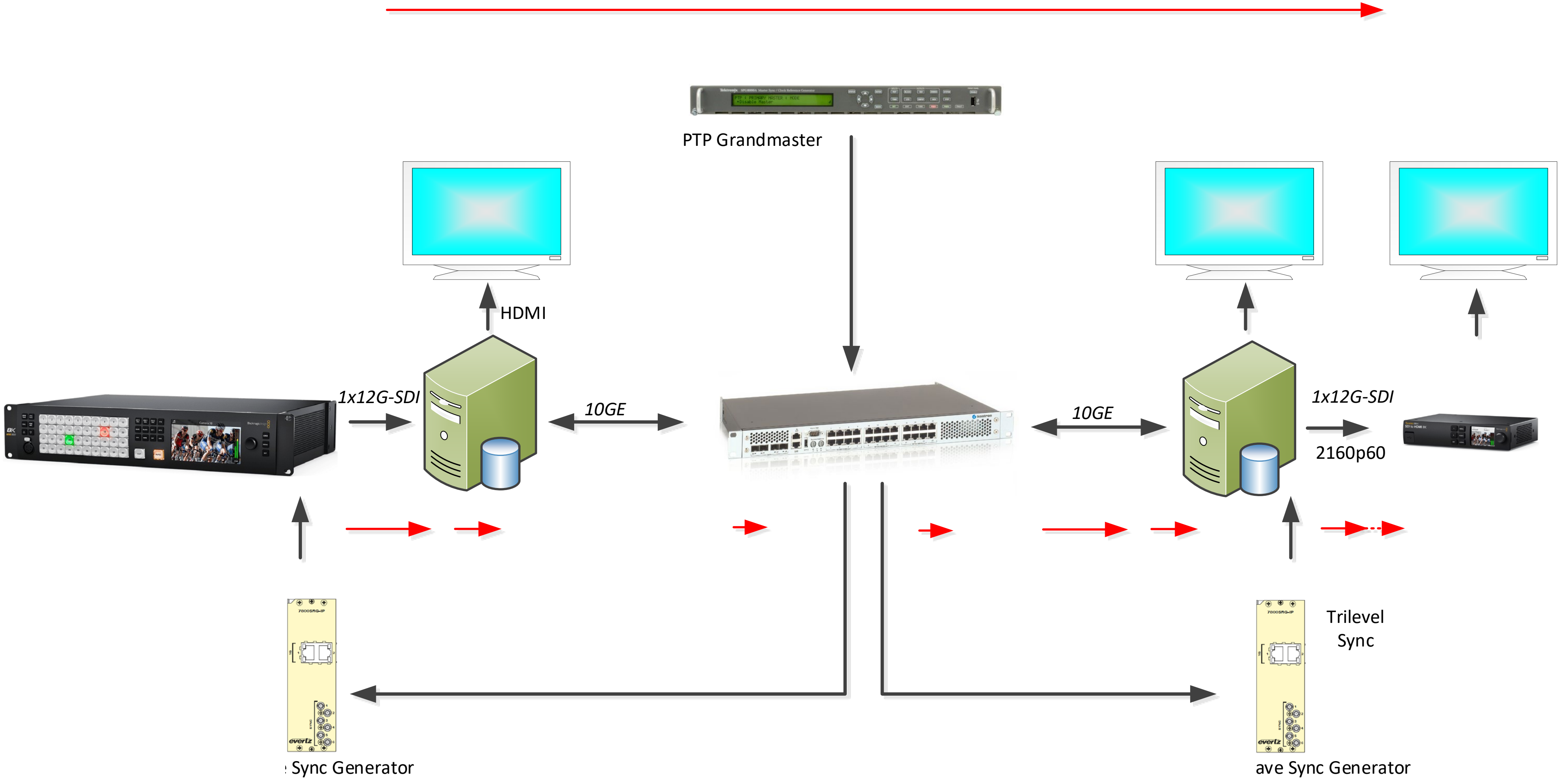
Slice type	Processing time on 3.7GHz CPU core (2bpp encoding)	Uncompr. Transmission time for 60 fps
1920 slice	0.342 ms	0.237ms (3G-SDI)
3840 slice	0.574 ms	0.119ms (12G-SDI)
7680 slice	1.018 ms	0.059ms (4x12G)

Transmission System Architecture

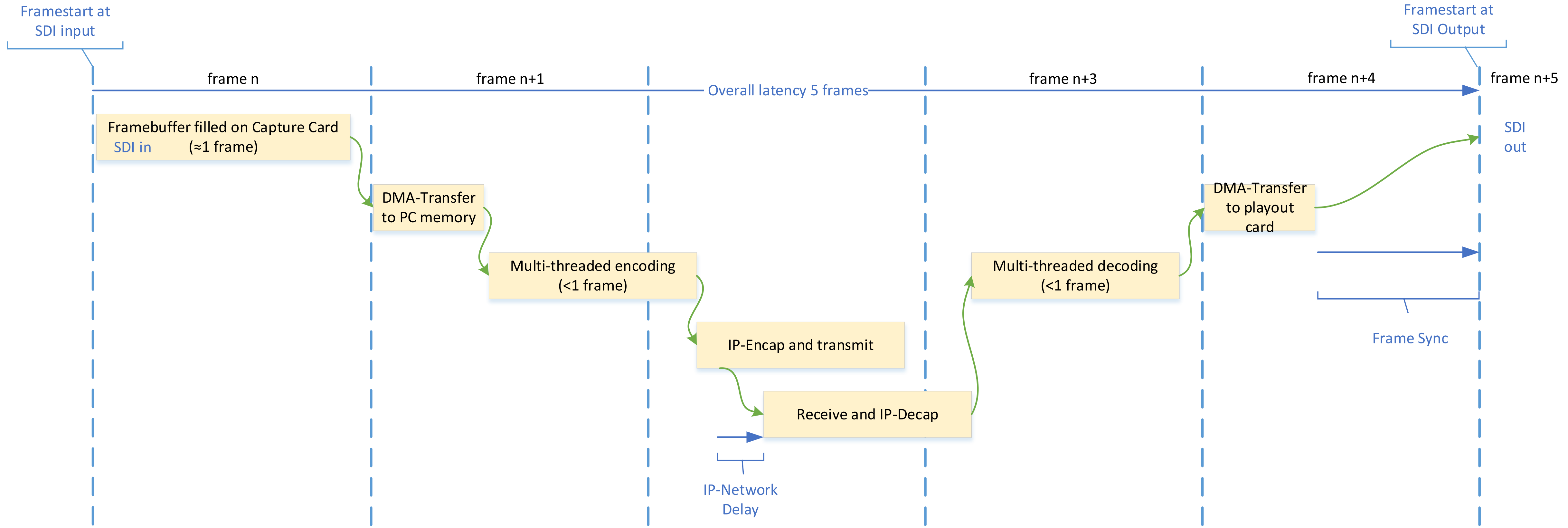
- Design Example
  - 12G SDI in
  - Encoding
  - RTP packaging
  - Transmission over ST2110
  - RTP unpackaging
  - Decoding
  - 12 SDI out



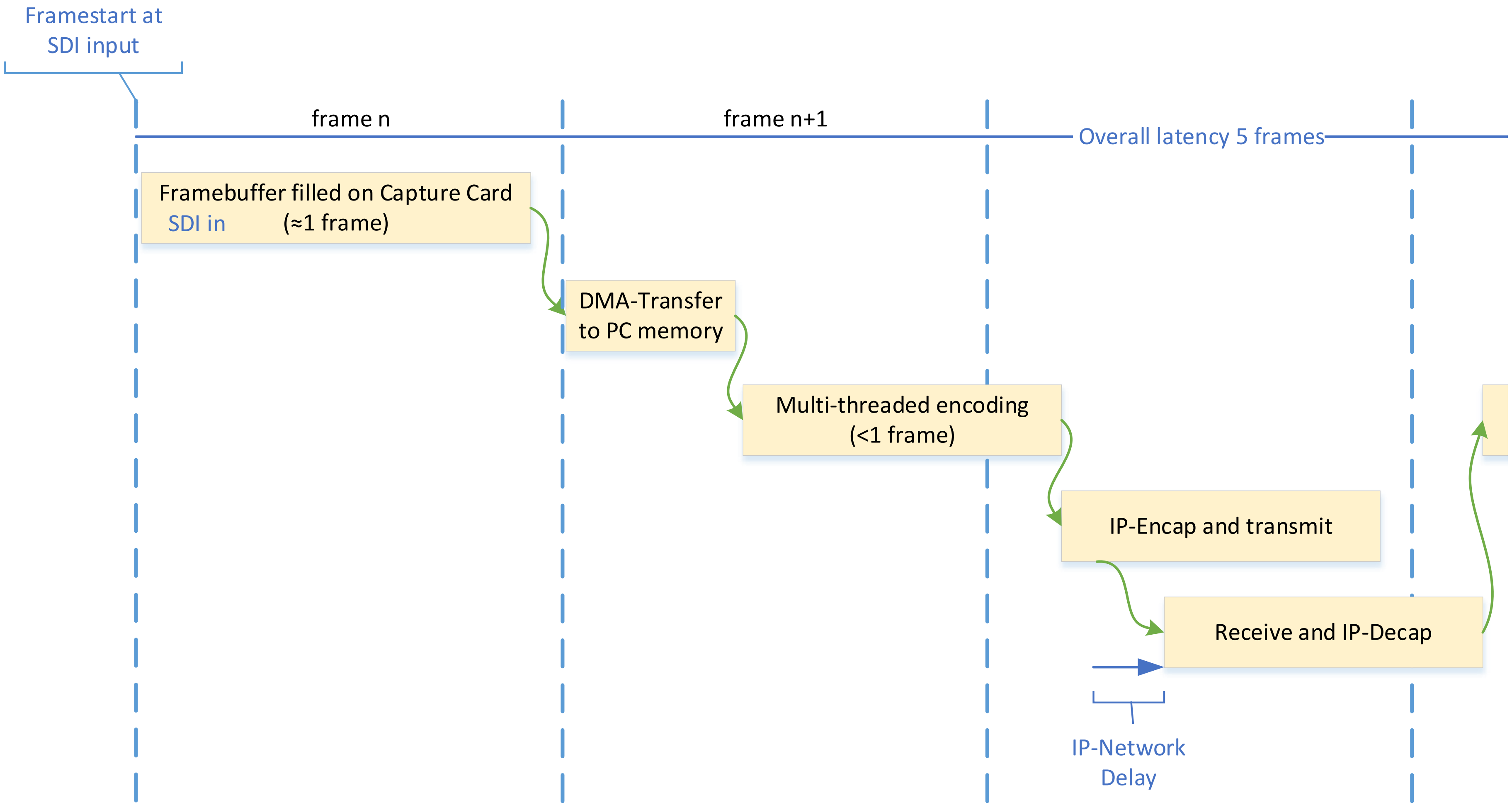
System Architecture – Standard



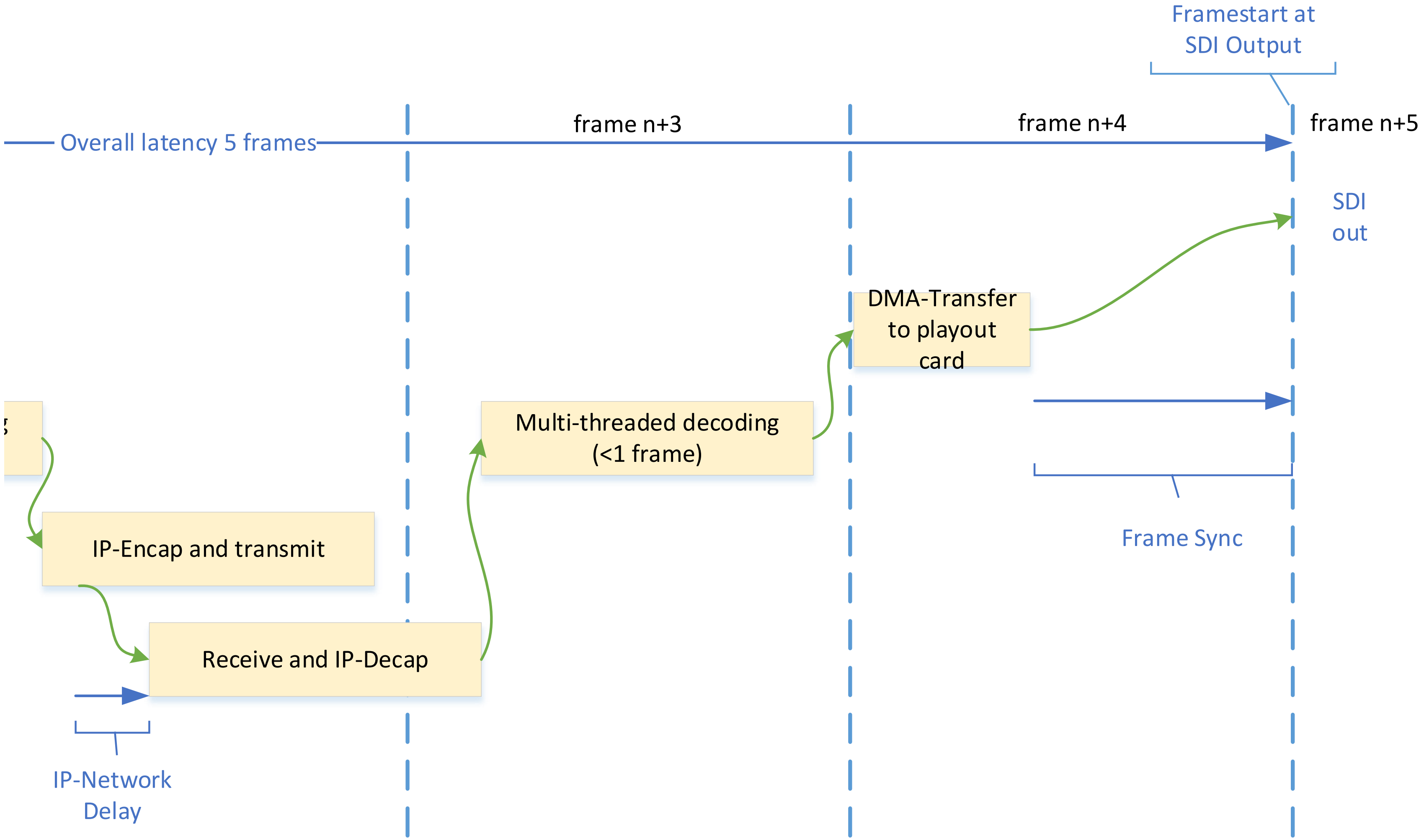
Latencies on a Standard System



Latencies - Sender

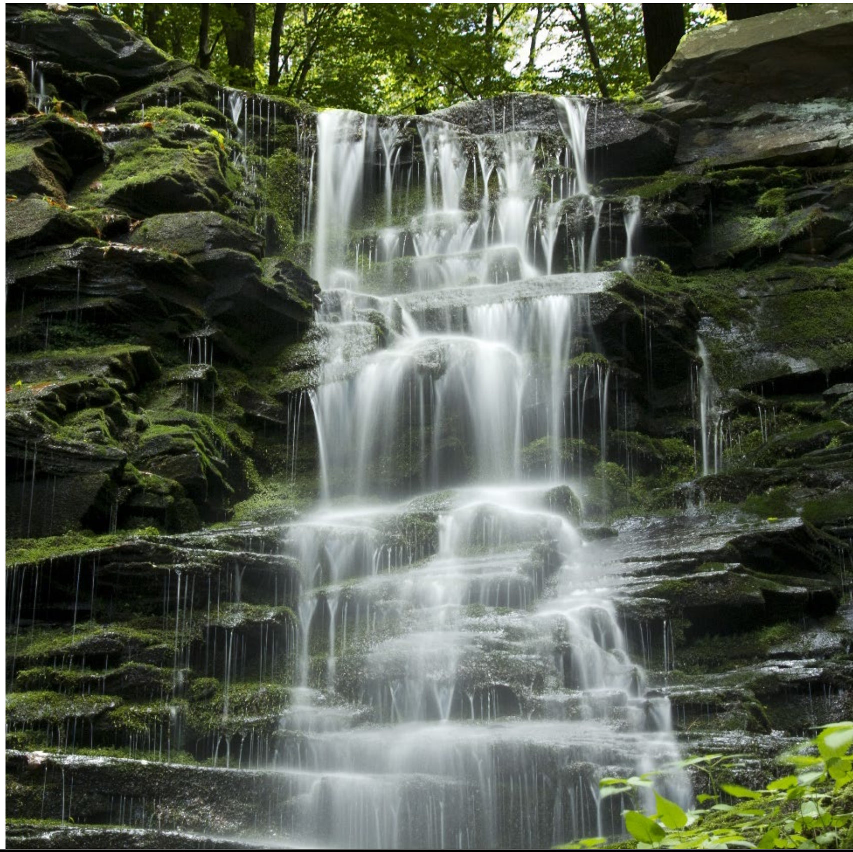


Latencies - Receiver



Latency Optimized System

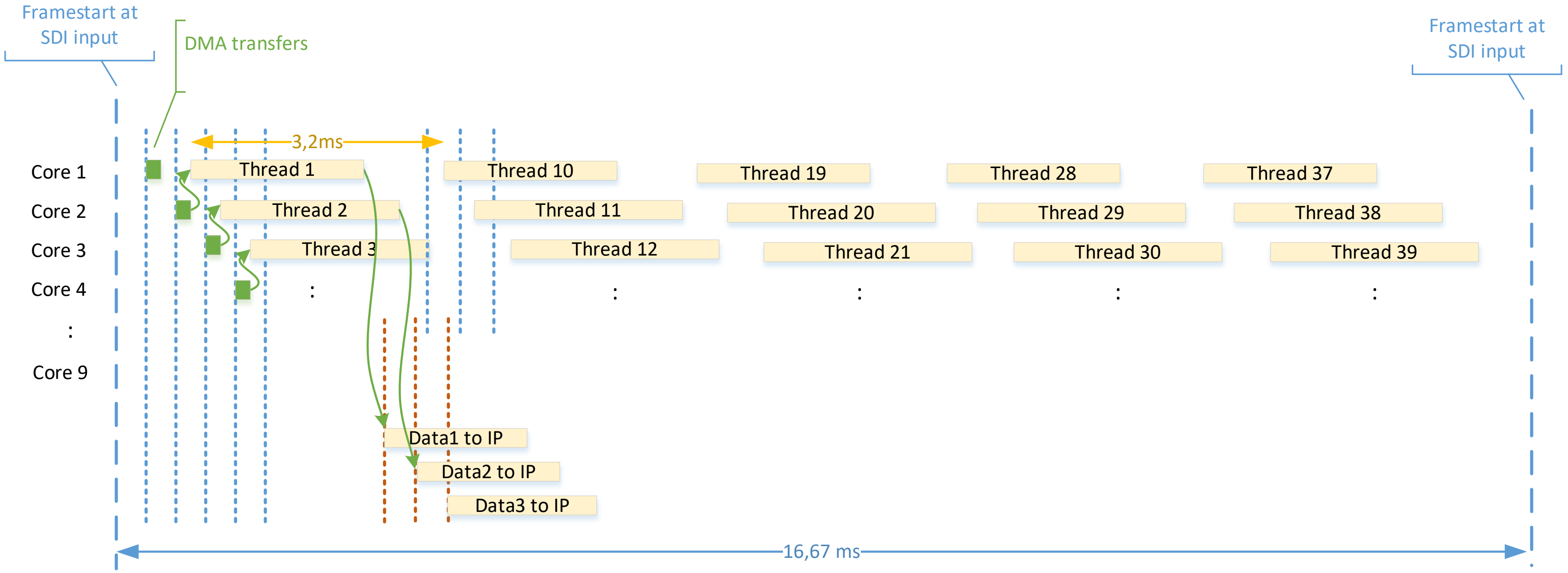
- Usage of SDI frame-grabber and playout card with Subframe-DMA access (in our case Deltacast card DELTA-12G-elp-h 40 and DELTA-12G-elp-h 04)
- All processing tasks parallelized to multiple threads and cascaded – **Waterfall principle!**
- Example:
  135 slices for UHD-1 are processed by 45 threads each processing 3 slices with
  9 CPU cores (each CPU core executes 5 threads per image)



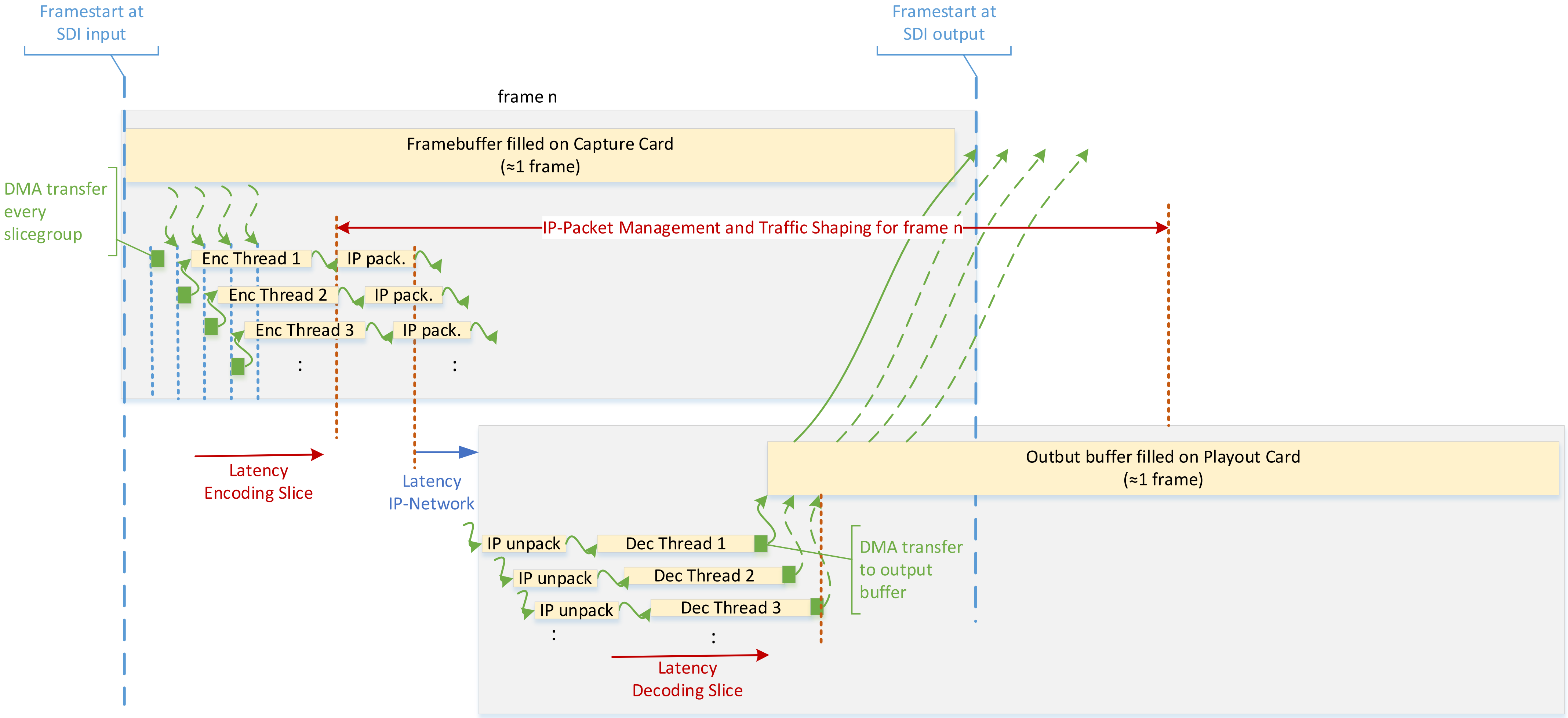
Latency Optimized System

- One thread processes 3 slices in a slicegroup
- One of 9 cores will be activated 3.2ms

Slices per thread (slicegroup)	Processing time @ 3.7GHz CPU core (2bpp encoding)	Uncompr. Transmission time for 60 fps
3 x 3840 slice	1.722 ms	0.357 ms (12G-SDI)



Latency Optimized System



Further Optimizations

- Slice based packaging to IP transport packets
- Out of order transmission of packets:
  - Send packets as soon as slices are encoded
  - Keep in mind encoding times may vary dependent on content
- Slices will be reordered at receiver side

Conclusion

- Due to slice-based processing, system latency can be adapted very flexible
- Variables: No. of available CPU cores, Clock of CPU cores, intended latency
- Tuning parameters: No. of threads, slices per slicegroup
- Latest implementation in lab for UHD-1 transmission: Using 5 cores of an AMD Ryzen 7 5700G for encoding or decoding with one frame delay end-to-end
- Outlook: Field test at live concert in Berlin playing jointly music on different stages on 11.12.2021

Contact Information

Fraunhofer IIS

- If you have any questions, please contact:

Prof. Dr.-Ing. Siegfried Foessel
Head of Department
Moving Picture Technologies

siegfried.foessel@iis.fraunhofer.de