# Mixed-Resolution HESP for More Efficient Fast Channel Switching and Packet-Loss Repair

Glenn Van Wallendael*, Peter Lambert*, Pieter-Jan Speelmans†, and Hannes Mareen*
* *IDLab, Ghent University – imec*, Ghent, Belgium, e-mail: firstname.lastname@ugent.be
† *THEO Technologies*, Leuven, Belgium, e-mail: pieter-jan.speelmans@theoplayer.com

***Abstract*—Low-delay live streaming applications desire fast channel switching and packet-loss repair capabilities. However, existing methods that provide these capabilities have a negative impact on the stream of steady-state users. To minimize this impact, techniques such as the High Efficiency Streaming Protocol (HESP) utilize keyframe injection. Such techniques combine compression-efficient normal streams with corresponding companion streams that are used in case of random access or packet loss. Unfortunately, because a companion stream is needed for every normal stream, the distribution cost and encoding complexity are considerable costs. Additionally, injecting a companion keyframe into a normal stream causes a bitrate spike. Therefore, this paper evaluates the impact of utilizing mixed-resolution keyframe injection in the H.266/VVC standard. By providing a single companion stream for all normal streams of a bitrate ladder, the three mentioned downsides can be mitigated. We found that injecting a lower-resolution keyframe effectively reduces the bitrate spike, at the cost of only a modest quality loss. For dynamic video content, the quality impact reduces over time, and is less perceptible than traditional packet-loss repair using frame copy. In conclusion, mixed-resolution HESP can reduce the bitrate spike and computational/distribution overhead of HESP when enabling fast channel switching or packet-loss repair.**

***Index Terms*—Keyframe Injection, High Efficiency Streaming Protocol (HESP), H.266/VVC, Random Access, Error Recovery**

## I. Introduction

Video providers perform significant efforts to increase the viewer-friendliness of live streaming. For example, the video content is typically encoded at multiple resolutions and quality levels in an adaptive bitrate (ABR) ladder. However, providers may struggle to provide low-delay channel switching (i.e., random access) and packet-loss repair. That is because keyframes that can be decoded without reference to previous frames typically occur infrequently (e.g., every couple of seconds). This frequency is set by the Group of Pictures (GOP) size. On the one hand, we wish to have a large GOP size to enable efficient compression. On the other hand, smaller GOP sizes allow for faster random access and packet-loss repair.

Random-access and packet-loss issues can be resolved using client-based, network-based and content-based methods. For example, client-based methods can provide random access using prefetching [1]–[4] and playback modification [5] techniques. However, these place a large bitrate burden on the client device, or can only slightly acommodate for low latencies. In case of a packet loss, i.e., corrupted Network Abstraction Layer Units (NALUs), client-based error concealment algorithms were developed for Advanced Video Coding (H.264/AVC) [6], [7], yet there is only a limited set of equivalent techniques for High Efficiency Video Coding (H.265/HEVC) [8] , let alone Versatile Video Coding (H.266/VVC) [9].

Within the network, caching or complex processing can be applied to facilitate random access and packet-loss repair. Although caching makes latency for random access solely dependent on buffer size, it puts a large burden on the client network by causing a significant bitrate peak [10], and they cannot be applied to packet-loss scenarios. Allowing complex processing in the network enables concepts such as zapping-accelerator servers. For example, these servers generate time-shifted versions of the video stream until a keyframe is present in the normal stream [11]. However, in this paper, we prefer to keep the complexity in the network minimal.

In content-based methods, additional pictures are generated to enable random access or to solve packet loss. In H.264/AVC, a special picture type was introduced to accommodate random access and packet-loss repair, namely Switching Intra (SI) pictures and Switching Predicted (SP) pictures [12]. However, the overhead introduced in SP-pictures penalizes the stream of steady-state users.

As a content-based alternative, techniques such as the High Efficiency Streaming Protocol (HESP) [13], [14] utilize keyframe injection or keyframe insertion, which combine the advantages of both a long GOP and a short GOP [15], [16]. That is, a compression-efficient Normal Stream (NS) that uses a (very) long GOP size is accompanied by a Companion Stream (CS) that only consists of keyframes (i.e., it has a very short GOP size). During streaming, clients receive the NS by default. Only when a channel change occurs or a packet loss is encountered, the next occurring keyframe from the CS is transmitted. This keyframe substitutes the corresponding frame in the NS. Although HESP is mainly used for low-latency HTTP adaptive streaming, it has already been tested in low-latency multicast streaming as well [17]. In both scenarios, the NS and CS-frames are generated on the origin server and distributed over the Content Distribution Network (CDN).

Keyframe injection is used in the following two scenarios. First, it is used when starting to watch a live video stream (e.g., due to a channel switch). In that case, client devices request one keyframe of the companion stream from the bitrate ladder,

This work was funded in part by the Research Foundation – Flanders (FWO) under Grant 1S55218N, IDLab (Ghent University – imec), Flanders Innovation & Entrepreneurship (VLAIO) project LIVE-G (Low-latency video over five-G - HBC.2020.2375), and the European Union.

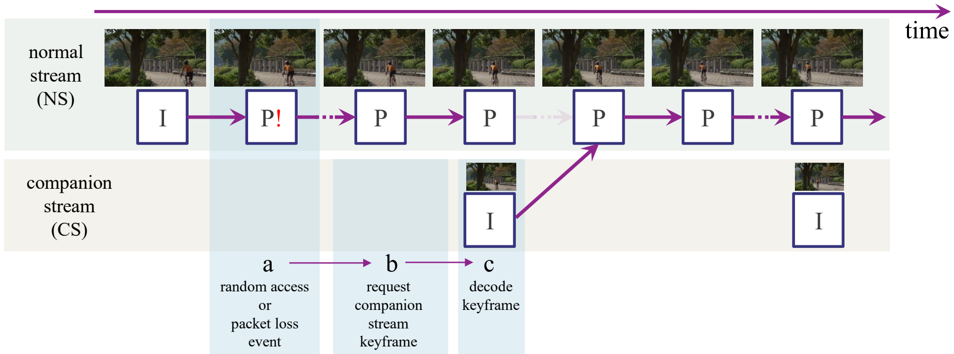



Fig. 1. Example of mixed-resolution keyframe injection where (a) random access or packet loss occurs, (b) a CS keyframe (possibly with different resolution) is requested, and (c) the keyframe is decoded instead of the predicted frame at an equal timestamp.

followed by the inter-predicted pictures from the associated normal stream. As such, with respect to the client device and its incoming bitrate, keyframes from the CS are only injected/received when required at startup. Second, packets or frames get lost due to streaming in a multicast environment or using an unreliable transmission. In the low latency case where there is no time for packet retransmits, a keyframe from the CS is requested only when needed by those client devices that need it to recover from losses.

As a downside of HESP, encoding and distributing a separate CS of keyframes for every NS in the bitrate ladder is computationally complex, as well as expensive for the CDN. Additionally, injecting a keyframe equal in quality to the NS causes the bitrate to spike. Even when lowering the quality, the injected CS keyframe may have a higher bitrate than the NS frame that it replaces.

To reduce the bitrate overhead, Girod *et al.* proposed to utilize the Reference Picture Resampling (RPR) mode in the H.263+ standard that enables spatial resolution changes [18]. However, this coding mode was removed from subsequent standard generations, until it was re-introduced in H.266/VVC and AV1 [19]. Now that this mode is available again, this paper explores the usage of RPR in H.266/VVC.

In short, this paper investigates the impact of injecting CS-keyframes with a different resolution than the NS-frame that it replaces. As such, one can significantly reduce the bitrate of the injected keyframe. Moreover, this enables the use of only a single CS to provide fast random access and packet-loss repair for an entire bitrate ladder. Using a single CS minimizes both the computational and distribution overhead. Our experimental results can aid video distributors that utilize HESP in making a trade-off between the advantages listed above and the corresponding impact on the visual quality.

In summary, the main novelties of this work are:

- We discuss the requirements to enable mixed-resolution keyframe injection in H.266/VVC.
- We extensively evaluate mixed-resolution keyframe injection to minimize the computational, distribution, and bitrate overhead of keyframe injection.
- We compare our method with a traditional packet-loss repair solution that utilizes a frame-copy strategy.

## II. Mixed-Resolution Keyframe Injection

Traditional keyframe injection for low-latency video delivery works by distributing a companion stream, with frequent keyframes, additional to the normal video stream. As illustrated in Figure 1, whenever a random access or a packet loss occurs, the next keyframe from the CS is requested from the CDN via unicast. At the timestamp of that keyframe, it is decoded instead of the predicted frame from the NS. This paper evaluates the impact and requirements of utilizing lower or higher-resolution CS keyframes instead of having an equal-resolution CS. As such, we tackle the three main downsides of current keyframe-injection methods.

More specifically, instead of requiring a CS corresponding to every resolution present in the bitrate ladder of the streaming solution, a single CS can enable random access and packet-loss recovery in the full bitrate ladder. Because the streaming solution can go as low as a single CS for all NSs in the bitrate ladder, both the distribution bitrate cost over the CDN and the encoding complexity can be drastically reduced. Additionally, when having to stay at a fixed resolution, the CS bitrate can only be reduced in a limited way. Therefore, injecting a CS keyframe into an NS can cause a significant bitrate spike. By allowing lower resolutions, lower bitrates can be achieved, which allows us to use CSs with keyframes of approximately-equal size as their NS counterparts. This results in random-access capabilities and packet-loss repair with no bitrate spike, yet at a certain quality cost (quantified in Section III-C).

Enabling mixed-resolution keyframe injection in H.266/VVC has the following requirements.

*a) Reference Picture Resampling (RPR):* Allowing mixed resolutions in a single videostream was enabled by RPR technology in the H.263+ standard, after which it was disabled in subsequent standards. Recently, it was re-introduced in H.266/VVC. That is, when enabling resolution changes inside the Sequence Parameter Set (SPS) of a video stream (mentioned as sps_res_change_in_clvs_allowed_flag in the standard), mixing different resolutions is allowed. With this flag enabled, every picture and its associated Picture Parameter Set (PPS) can specify a scaled width and height of the encoded frame (mentioned as pps_pic_width_in_luma_samples and pps_pic_height_in_luma_samples in the standard).

*b) Parameter Sets:* When performing keyframe injection with mixed resolutions, it is important to cautiously specify parameter set information and retransmit this information after the keyframe-injection operation. New in H.266/VVC is the concept of Adaptation Parameter Sets (APS) [20]. APS packets can specify three types of information (aps_params_type), Adaptive Loop Filter (ALF) parameters, Luma Mapping with Chroma Scaling (LMCS) parameters, and scaling-list parameters. It is crucial to keep a dictionary of APS packets, having unique identification (ID) numbers as keys (aps_adaptation_parameter_set_id). After keyframe injection, the APS dictionary standard-compliant decoders are cleaned up. Therefore, all unique and most recent APS packets must be reinjected in the video stream after keyframe injection. As such, predicted frames succeeding the injected keyframe can again make use of preceding APS information as intended by the NS encoder.

Additional to APS management, the NS and CS encoders should guarantee uniqueness of PPS identifiers. Standard compliance would be broken if PPS sets with different resolution information would have the same PPS ID, because the standard assumes unique and unchanged PPS information coupled to a certain PPS identifier.

*c) Picture Order Count (POC):* To guarantee standard-compliant decoding, it is important to have an identical POC for every frame that gets replaced by a CS keyframe. This can be accomplished by matching the POCs of either Instantaneous Decoder Refresh (IDR) keyframes or Clean Random Access (CRA) keyframes in the CS.

## III. Experimental Results

### A. Experimental Setup

We utilized the VVC Test Model (VTM) version 10.2 as a video encoder, modified according to the requirements mentioned in Section II. The NS videos were encoded using a low-delay configuration in which the first frame is a keyframe and all other frames are predicted frames that each take only the preceding frame as reference (i.e., an IPPP structure). The CS was encoded with the same configuration as the NS, albeit only encoding keyframes.

In both the NS and CS configuration, RPR was used to set a scaling ratio of 1, 1.5, and 2, further denoted as $\text{Res}_{\text{NS}}$ for the NS, and $\text{Res}_{\text{CS}}$ for the CS. Note that a higher number signifies a smaller resolution, and using a scaling ratio of 1 for both the NS and CS corresponds to the equal-resolution keyframe injection that exists in the state of the art [16]. Additionally, the results from the scaling ratio 1.5 are omitted from this paper due to space limitations. Each sequence is compressed using four different Quantization Parameters (QPs), namely 22, 27, 32, and 37, which is further denoted as $\text{QP}_{\text{NS}}$ for the NS, and $\text{QP}_{\text{CS}}$ for the CS. Note that we created mutliple CSs for the evaluation, but that applications can opt to only use a single CS (compressed with a certain chosen QP).

The keyframe of the CS is injected in the NS at frame $f = 16$, which is the 17th frame. We further denote this combined stream as Keyframe-Injected stream (KI). The experiments were performed on 22 sequences with resolutions between 416×240 and 2560×1600, that contain between 150 and 600 frames and have a frame rate between 20 and 60 frames per second (fps) [21].

To compare our results to more related work than previous keyframe-injection research [16], we additionally analyzed the impact of using traditional packet-loss repair that utilizes a frame-copy strategy. That is, when a frame that is lost is used as a reference by another frame, the preceding frame is used as reference instead. It should be stressed, though, that this is only a solution for packet-loss repair, and not for channel switching or random access.

TABLE I
Median factor of frame size increase results.

| | $\text{QP}_{\text{CS}}$ | $\text{Res}_{\text{NS}} = 1$ $\text{QP}_{\text{NS}}$ | | | | $\text{Res}_{\text{NS}} = 2$ $\text{QP}_{\text{NS}}$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 22 | 27 | 32 | 37 | 22 | 27 | 32 | 37 |
| $\text{Res}_{\text{CS}} = 1$ | 22 | 8.7 | 28.5 | 63.8 | 142.6 | 33.7 | 81.0 | 201.3 | 520.8 |
| | 27 | 4.9 | 16.8 | 39.6 | 88.6 | 16.8 | 48.3 | 119.9 | 254.4 |
| | 32 | 2.7 | 9.3 | 23.4 | 51.3 | 9.1 | 27.5 | 63.5 | 153.6 |
| | 37 | 1.5 | 4.9 | 12.6 | 27.5 | 5.1 | 14.8 | 35.6 | 87.6 |
| $\text{Res}_{\text{CS}} = 2.0$ | 22 | 2.8 | 8.0 | 20.1 | 44.8 | 9.9 | 25.5 | 63.2 | 135.6 |
| | 27 | 1.7 | 5.1 | 13.3 | 29.8 | 6.0 | 16.2 | 42.1 | 92.1 |
| | 32 | 1.0 | 3.1 | 8.2 | 18.4 | 3.3 | 9.7 | 25.5 | 52.9 |
| | 37 | 0.6 | 1.7 | 4.7 | 10.6 | 1.8 | 5.3 | 13.6 | 30.5 |

### B. Impact on Frame Size

Table I shows the factor of frame size increase from the CS keyframe to the P-frame in the NS that it replaces. The reported values are median values calculated over all tested sequences, for each configuration of QP and resolution, to reduce the influence of outliers. The top-leftmost part of Table I shows the frame size increases when injecting a full-resolution keyframe in a full-resolution NS, which has already been done in the state of the art study [16]. It can be observed that the median frame size increase is between 8.7 and 27.5 when utilizing the same QP for the NS and CS, which means keyframe injection causes a small bitrate spike in those cases.

When raising the resolution of the injected CS keyframe (i.e., the upper-right part of the diagonal), the frame size increases. In contrast, when lowering the resolution of the injected CS keyframe (i.e., the lower-left part of the diagonal), the frame size decreases. As such, it is possible to inject keyframes that do not cause a bitrate spike (i.e., with a frame size increase factor equal or less than 1.0). For example, when injecting a keyframe with $\text{Res}_{\text{CS}} = 2.0$ and $\text{QP}_{\text{CS}} = 32$ in an NS with $\text{Res}_{\text{NS}} = 1$ and $\text{QP}_{\text{NS}} = 22$, the median frame size increase factor is 1.0.

### C. Impact on Quality

*1) Median PSNR Decrease:* Table II shows the median Peak Signal-to-Noise Ratio (PSNR) decreases. That is, for each test sequence, we average the PSNR decease over all frames. Then, the median is calculated over all test sequences.

TABLE II
MEDIAN PSNR DECREASE RESULTS.

| | $QP_{CS}$ | $\mathrm{Res}_{NS} = 1$ $QP_{NS}$ | | | | $\mathrm{Res}_{NS} = 2$ $QP_{NS}$ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 22 | 27 | 32 | 37 | 22 | 27 | 32 | 37 |
| $\mathrm{Res}_{CS} = 1$ | 22 | 0.45 | 0.37 | 0.21 | 0.21 | 0.09 | 0.18 | 0.24 | 0.05 |
| | 27 | 0.67 | 0.47 | 0.24 | 0.22 | 0.12 | 0.17 | 0.19 | 0.04 |
| | 32 | 1.15 | 0.81 | 0.30 | 0.25 | 0.23 | 0.19 | 0.17 | 0.07 |
| | 37 | 2.10 | 1.00 | 0.74 | 0.27 | 0.32 | 0.32 | 0.21 | 0.09 |
| $\mathrm{Res}_{CS} = 2$ | 22 | 3.09 | 1.48 | 0.79 | 0.50 | 0.16 | 0.18 | 0.23 | 0.08 |
| | 27 | 3.41 | 1.66 | 0.89 | 0.51 | 0.27 | 0.20 | 0.20 | 0.11 |
| | 32 | 3.91 | 2.38 | 1.18 | 0.65 | 0.40 | 0.29 | 0.23 | 0.12 |
| | 37 | 4.61 | 3.07 | 1.63 | 0.98 | 0.63 | 0.46 | 0.38 | 0.23 |
| Trad. Packet-Loss Repair | | 4.37 | 3.42 | 1.70 | 0.89 | 0.94 | 0.93 | 0.52 | 0.28 |

The top-leftmost part of Table II shows the PSNR decreases when injecting a full-resolution keyframe in a full-resolution NS. Again, this scenario has already been studied [16]. The median PSNR decrease in this scenario is relatively low; it is maximum 2.1 dB. When the resolution of the injected keyframe is lowered (i.e., the lower-left part of the diagonal), the median PSNR decrease values increase. However, the quality impact is relatively low for most tested configurations. In contrast, when the resolution of the injected keyframe is increased (i.e., the upper-right part of the diagonal), the impact on the quality decreases, and becomes negligible.

Combining the information of Table I and Table II, one can compare the impact on the quality for the same frame size increase factor. That is, one can make an informed decision of choosing to either change the resolution or change the QP of the injected keyframe. For example, for $\mathrm{Res}_{NS} = 1$ and $\mathrm{QP}_{NS} = 27$, we can inject a keyframe that is approximately 8 to 9 times larger than the P-frame that it replaces in two ways: either with $\mathrm{Res}_{CS} = 1$ and $\mathrm{QP}_{CS} = 32$ or with $\mathrm{Res}_{CS} = 2$ and $\mathrm{QP}_{CS} = 22$. The corresponding median PSNR decreases are 0.81, and 1.48, respectively. Although the difference between the two options is small, it suggests that remaining a higher resolution for the injected keyframe is better in this scenario.

*2) Static vs. Dynamic Content:* Fig. 2a shows the PSNR decreases for all frames of all test sequences, when injecting a keyframe encoded at half resolution ($\mathrm{Res}_{CS} = 2$) with $\mathrm{QP}_{CS} = 32$) in a full-resolution NS ($\mathrm{Res}_{NS} = 1$) with $\mathrm{QP}_{NS} = 22$). This configuration was chosen to be highlighted because it results in a median frame size increase factor of 1.0. Note that it is an extreme example that results in a relatively high median PSNR decrease; much better results were obtained for higher resolutions or lower QPs of the CS.

In Fig. 2a, we can observe that *SlideEditing*, *SlideShow* and *ChinaSpeed* behave significantly worse than the others sequences. These are sequences containing purely static content; they contain screen recordings or a video game's heads-up display (HUD). Because the frames after keyframe injection copy the static regions in the reference frames, these regions remain of a low resolution.

Additionally, the sequence *BQSquare* has a noticeable quality impact. Whereas the quality loss of other sequences decreases or remains approximately constant over time, *BQSquare*'s quality impact becomes more perceptible. This sequence contains zebra-pattern artifacts, which were also observed and discussed in related work [16]. Therefore, we do not discuss this rare worst case in detail here.

In contrast, sequences with dynamic content (such as *Race-Horses* and *BasketballDrive*) have similar PSNR decrease values as the static-content sequences in the first frames , but quickly recover afterwards. This is because the dynamic content that is introduced in the next P-frames is of a high resolution again. Video examples of the impact in dynamic and static content can be found on our website[1].

In general, we can state that the quality decrease of keyframe injection is approximately capped to the quality of the injected keyframe. When the keyframe is of low resolution and low quality, the next P-frames will retain the static low-resolution regions, but may increase when dynamic high-resolution content is introduced.

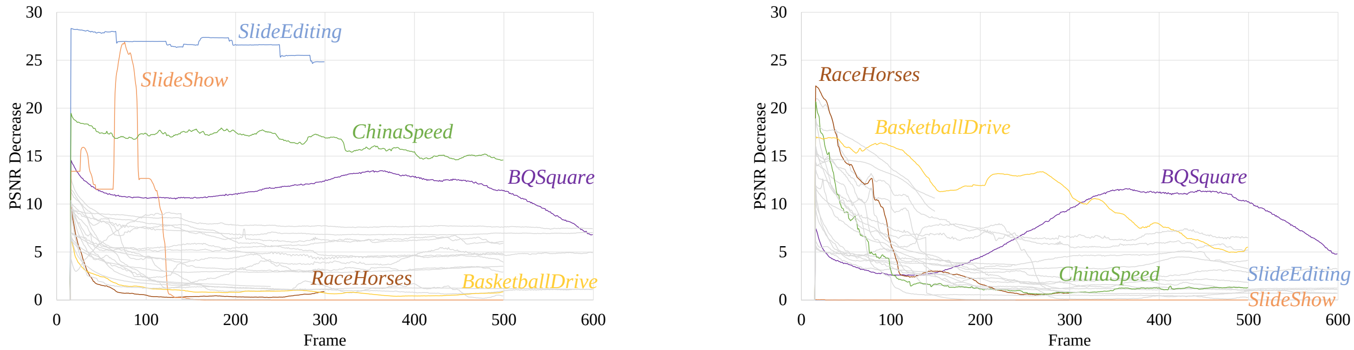
*3) Comparison with Traditional Packet-Loss Repair:* As explained in Section III-A, frame copy is used as a traditional packet-loss repair solution, which replaces the lost frame by the preceding frame. As such, this preceding frame is used as a reference by subsequent frames. The last row of Table II shows the median PSNR decrease results of traditional packet-loss repair. For example, the median PSNR decrease values for $\mathrm{Res}_{NS} = 1$ are between 0.9 and 4.4, which is comparable to the worst results of mixed-resolution keyframe injection, i.e. when using a lower-resolution and lower-quality CS.

It is important to analyze the worst cases of traditional packet-loss repair as well. Fig. 2b shows the course of quality decrease over time for all tested sequences. It is especially interesting to compare these graphs with those of keyframe injection in Fig. 2a. We can observe that the frame-copy solution can introduce a significant quality loss, of up to approximately 23 dB in this test configuration. This is comparable to the worst cases that we observed in the shown challenging test configuration of mixed-resolution keyframe injection. However, it is important to stress that the frame-copy solution does not provide fast random access.

In contrast to keyframe injection, the sequences in which the traditional packet-loss-repair artifacts are most perceptible are not static (such as *ChinaSpeed*, *SlideEditing* and *SlideShow*), but are rather dynamic ones (such as *RaceHorses* and *BasketballDrive*). That is, this traditional packet-loss repair solution works very well for static content. If the content does not change, the preceding frame is a good replacement reference frame. In that case, keyframe injection may not be particularly beneficial and only provide a bitrate overhead. Most importantly, when the preceding frame is not a good match in dynamic content such as *RaceHorses* and *BasketballDrive*, perceptible artifacts may be created. A video example of these perceptible artifacts is provided on our website[1].

Future work will investigate how to predict whether or not traditional packet-loss repair would create perceptible artifacts. As such, it could be used to decide on the fly to either perform keyframe injection or traditional packet-loss repair.

[1] https://media.idlab.ugent.be/hesp-mixed-res




(a) PSNR decrease keyframe injection.

(b) PSNR decrease traditional packet-loss repair.

Fig. 2. The quality decrease over time in terms of PSNR, for mixed-resolution keyframe-injection method (a) as well as traditional packet-loss repair using a frame-copy strategy (b). The shown keyframe-injection configuration is a full-resolution NS with $\mathrm{QP_{NS}} = 22$ with a half-resolution CS with $\mathrm{QP_{CS}} = 32$, resulting in a median frame size increase of 1.0 (i.e., bitrate unchanged).

## IV. Conclusion

This paper evaluated mixed-resolution keyframe injection to reduce the number of encoded companion streams, as well as to potentially reduce or completely resolve the bitrate spike. Thorough experiments were performed to analyze the impact on the frame sizes and quality.

We demonstrated that the impact of keyframe injection on the quality is typically not disrupting. In general, the quality is capped by the quality of the injected keyframe. That is, if the resolution and quality of the injected keyframe is low, the static content in the subsequent P-frames of the NS remains of similar resolution and quality. In contrast, the quality and resolution of dynamic content increases again over time. This is not the case in the compared traditional packet-loss repair solution, which can create very perceptible artifacts in dynamic content. Thus, mixed-resolution keyframe injection outperforms traditional packet-loss repair.

In conclusion, mixed-resolution keyframe injection can be used by video content providers for ultra-low-latency video streaming. As such, we reduced the required encoding overhead and bitrate spike to provide fast channel switching, random access and packet-loss repair.

## References

[1] M. Mignon, K. Bouckhout, J. Gahm, and A. C. Begen, “Scaling server-based channel-change acceleration to millions of iptv subscribers,” in *Int. Packet Video Workshop (PV)*, 2012, pp. 107–112.

[2] A. C. Begen, N. Glazebrook, and W. Ver Steeg, “Reducing channel-change times with the real-time transport protocol,” *IEEE Internet Computing*, vol. 13, no. 3, pp. 40–47, 2009.

[3] E. Lee, J. Y. Ku, and H. Bahn, “An efficient hot channel identification scheme for IPTV channel navigation,” *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 124–129, 2 2014.

[4] C. Yang and Y. Liu, “On Achieving Short Channel Switching Delay and Playback Lag in IP-Based TV Systems,” *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 1096–1106, 7 2015.

[5] M. Kalman, E. Steinbach, and B. Girod, “Adaptive media playout for low-delay video streaming over error-prone channels,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 841–851, 6 2004.

[6] C. Nam, C. Chu, T. Kim, and S. Han, “A novel motion recovery using temporal and spatial correlation for a fast temporal error concealment over H.264 video sequences,” *Multimedia Tools and Appl.*, vol. 79, no. 1, pp. 1221–1240, 2020.

[7] M. Kazemi, M. Ghanbari, and S. Shirmohammadi, “A review of temporal video error concealment techniques and their suitability for hevc and vvc,” *Multimedia Tools and Appl.*, vol. 80, no. 8, pp. 12 685–12 730, 2021.

[8] Y. Peng and P. C. Cosman, “Weighted boundary matching error concealment for HEVC using block partition decisions,” in *2014 48th Asilomar Conference on Signals, Systems and Computers*, 11 2014, pp. 921–925.

[9] M. Benjak, Y. Samayoa, and J. Ostermann, “Neural network-based error concealment for vvc,” in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 2114–2118.

[10] B. S. Ghahfarokhi, N. Moghim, and S. Eftekhari, “Reducing channel zapping time in live TV broadcasting over content centric networks,” *Multimedia Tools and Appl.*, vol. 76, no. 22, pp. 23 239–23 271, 2017.

[11] Y. Bejerano and P. V. Koppol, “Improving Zap Response Time for IPTV,” in *IEEE INFOCOM*, 2009, pp. 1971–1979.

[12] M. Karczewicz and R. Kurceren, “The SP- and SI-frames design for H.264/AVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 637–644, 7 2003.

[13] P.-J. Speelmans, “Low encoding overhead ultra-low latency streaming via hesp through sparse initialization streams,” in *Proc. Mile-High Video Conf. (MHV)*. New York, NY, USA: Association for Computing Machinery, 2022, p. 115–116.

[14] ——, “HESP - High Efficiency Streaming Protocol,” Internet Engineering Task Force, Internet-Draft draft-theo-hesp-01, Nov. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-theo-hesp-01

[15] G. Van Wallendael, H. Mareen, J. Vounckx, and P. Lambert, “Keyframe Insertion: Enabling Low-Latency Random Access and Packet Loss Repair,” *Electronics*, vol. 10, no. 6, 2021.

[16] H. Mareen, M. Courteaux, J. Vounckx, P. Lambert, and G. Van Wallendael, “Keyframe insertion for random access and packet-loss repair in H.264/AVC, H.265/HEVC, and H.266/VVC,” in *Data Compression Conference (DCC)*, 2022.

[17] E. Okerman and J. Vounckx, “Fast startup multicast streaming on operator iptv networks using hesp,” in *IEEE Int. Symposium Multimedia (ISM)*, 2021, pp. 79–86.

[18] B. Girod, N. Farber, and U. Horn, “Scalable codec architectures for internet video-on-demand,” in *Asilomar Conf on Signals, Systems and Computers*, vol. 1, 1997, pp. 357–361 vol.1.

[19] M. Kerdranvat, Y. Chen, R. Jullian, F. Galpin, and E. François, “The video codec landscape in 2020,” *ITU Journal: ICT Discoveries*, vol. 3, no. 1, pp. 73–83, 2020.

[20] B. Bross, Y. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, “Overview of the versatile video coding (VVC) standard and its applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, 2021.

[21] F. Bossen, “Common test conditions and software reference configurations,” ITU-T Joint Collaborative Team on Video Coding (JCT-VC), Tech. Rep. JCTVC-L1100, 2013.