Poster: Adaptive Video Encoder for Network Bandwidth Drops in Real-Time Communication[*]

Hua Meng
hmengab@connect.ust.hk

Xiangjie Huang
xhuangdd@connect.ust.hk

Zili Meng
zilim@ust.hk

Hong Kong University of Science and Technology

Abstract

Real-time communication (RTC) is crucial in digital life, with latency significantly affecting user experience. Latency spikes often occur due to mismatches between video codec bitrates and network capacity, especially during sudden bandwidth drops. Current video encoders adjust bitrates too slowly, increasing latency. This poster suggests that encoders should adapt more quickly to network changes by dynamically adjusting codec parameters, maintaining compression efficiency. Preliminary tests with the x264 codec show these strategies can reduce latency by 28.66% to 78.87% while slightly improving video quality by 0.8% to 3%.

CCS Concepts

• **Networks → Cross-layer protocols**; • **Information systems → Multimedia streaming**.

Keywords

adaptive encoder, low latency, video streaming

ACM Reference Format:
Hua Meng, Xiangjie Huang, and Zili Meng. 2025. Poster: Adaptive Video Encoder for Network Bandwidth Drops in Real-Time Communication. In *ACM SIGCOMM 2025 Conference (SIGCOMM Posters and Demos '25), September 8–11, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3744969.3748452

1 Introduction

Real-time communication (RTC) has become an essential part of modern digital life, such as videoconferencing, cloud gaming and virtual reality. A critical metric that influences user experience is latency. When latency exceeds the human perceptual interval, users may experience a sensation of stuttering or freezing, which will affect the overall experience. For instance, video conferencing requires latency to remain below 150ms [3], while cloud gaming requires latency under 96ms [4].

One of the root causes of latency spike is the mismatch between the bitrate of video codecs and the capacity of networks. Network conditions can be highly variable and may experience sudden drops in bandwidth. When the network capacity drops, if the video codec cannot quickly adapt to the network fluctuations, the video frames will be queued after encoding and before sending to the network.

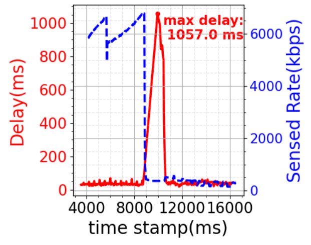

Figure 1: The delay and the bitrate sensed by the RTC during the bandwidth drop from 10 Mbps to 1 Mbps over time.

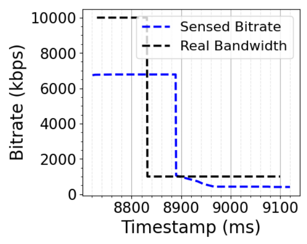

Figure 2: Real bandwidth and bitrate sensed by the RTC during the bandwidth drop from 10Mbps to 1Mbps.

This will result in latency spikes from the end-to-end perspective. For example, in video calls, participants may appear frozen or experience delays. Moreover, such latency fluctuations is increasingly violent and common with the increase of the range of network bandwidth. For example, even with most advanced wireless access networks such as WiFi-6 and 5G, network capacity can drop by 50× in 200 ms with a non-trivial frequency of 1% [5].

Unfortunately, the speed of bitrate adaptation of current video encoders is far slower than the speed of congestion controllers. This is due to the variable bitrate (VBR) mechanism in the existing commercial video codecs. To maximize the compression efficiency of video streams under dynamic contents and ensure the consistent quality at the same time, video encoders allow the encoded frame size to fluctuate around the target bitrate. This non-strict property of the encoded bitrate will result in significant bitrate overshooting when the network capacity drops. We demonstrate this via a motivating experiment of using libx264. As illustrated in Figure 1, WebRTC experiences a significant delay spike when there is a drop in bandwidth. At t=9000ms, when the congestion controller reduces the sending rate, the encoder takes around 200ms to gradually reduce its encoded bitrate, which result in a drastic delay spike of more than one second.

Some existing efforts have tried to mitigate the gap between the encoder's bitrate and network capacity. For example, Salsify [1] will skip the oversized frames. Concerto [8] tries to use deep learning to better predict the frame sizes. ACE [2] allows those overshot frames during bandwidth drops to overwhelm the network for a transient period of time. Unfortunately, they can only mitigate the issue but not solve it – the issue is still in the mechanisms of video codec. Several studies have focused on the adaptive configuration of encoders. For example, ARREMIS [7] dynamically selects different bitrate ladders. However, the bitrate ladder approach offers a predefined set of bitrates and resolutions, which can not fully accommodate the variability of changing network conditions.

Our key insight is that *encoders should become more adaptive to network fluctuations*. This adaptability can be achieved in two key ways. First, encoders should dynamically adjust codec parameters to quickly accommodate changes in bandwidth. When the

[*]This work was supported by Guangdong NSF Project (No. 2025A1515010460). Zili Meng is the corresponding author.

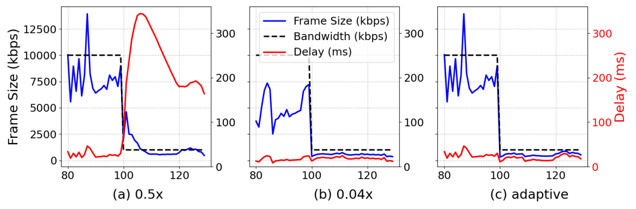

Figure 3: Frame sizes and delay with different vbv_buffer_size settings during the bandwidth drop from 10 Mbps to 1 Mbps. The x-axis represents the frame index. (a) set to half of the current bitrate; (b) set to the size of one frame; (c) adaptively chosen, using a smaller vbv_buffer_size during the drop while keeping the others at half of the current bitrate.

network is stable, the encoder still exploits the fluctuation of bitrate to maximize the compression efficiency. When the bandwidth drops happen, the encoder should strictly follow the bitrate to accelerate the speed of rate adaptation. Since bandwidth drop only accounts for a small portion of all the time, the overall compression efficiency is hardly affected. Moreover, they can utilize additional network information to predict potential bandwidth drops, such as reusing the existing congestion control mechanisms with a loose threshold. By implementing these strategies, we can significantly enhance the responsiveness of encoding processes, ultimately improving the overall user experience in RTC.

We evaluate our preliminary idea using x264. By adaptively adjusting codec parameters and predict potential drops in advance, it reduces the 90th percentile delay by 28.66% to 78.87%, while maintaining video quality or achieving a slight improvement of 0.8% to 3%.

2 Design

We approach the adaptive encoder through two key dimensions.

Adaptive encoder parameter Many encoding parameters will affect the final size of encoded frames, which in turn affect transmission time and latency. The `vbv_buffer_size` parameter in x264 is the dominant parameter among all these parameters. It manages the remain size for encoding the current frame. In practice, this parameter is usually set to half of the current bitrate to allow the fluctuation of frame sizes. As illustrated in Figure 3 (a), consistently using this setting can cause significant delays during bitrate drops. On the other hand, setting the vbv_buffer_size to the size of a single frame ensures that no frame exceeds the expected bitrate, but will lead to quality fluctuations and wasted bandwidth. Therefore, an adaptive vbv_buffer_size strategy can effectively reduce delays during bitrate drops by using smaller parameters while maintaining normal settings under stable conditions to preserve video quality.

Predict potential drops If we only adapt the parameter after the bandwidth drop, the performance will still be much limited. Thus, we want to proactively adjust the parameter before the bandwidth drops. This sounds to be a challenging task since congestion control has already been well studied but still open. However, the opportunity here is that we can allow false positives – if the bandwidth drop does not really happen, the impact on the video quality is limited for a transient period of time. Figure 2 shows that it takes 58 ms for the encoder to adjust its bitrate after a bandwidth drop. Improving the responsiveness of RTC to these changes is challenging due to the frequent spikes and sudden fluctuations in real-world conditions. Quickly lowering the target bitrate based on these random changes can significantly reduce frame quality. Additionally, congestion control mechanisms in RTC do not only set the encoder's bitrate; they also influence the sending

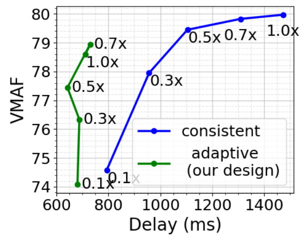

Figure 4: The delay and VMAF values for various vbv_buffer_size settings and our design during the bandwidth drop from 10 Mbps to 1 Mbps.

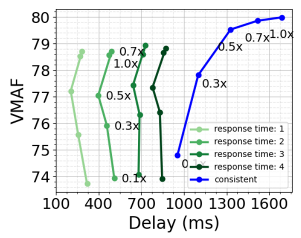

Figure 5: The delay and VMAF values for various response times during the bandwidth drop from 10 Mbps to 1 Mbps.

rate, queuing strategies, and other factors. Therefore, it is crucial to consider the needs of various components when determining the target bitrate. Our adaptive encoder does not directly change the bitrate; instead, it adjusts other parameters to achieve a quicker response.

3 Evaluation

We select x264 as an example encoder to demonstrate the benefits of our design. The Video Multi-method Assessment Fusion (VMAF) [6] metric is utilized to evaluate the received video quality, while delay is calculated based on the encoded frame size and the current bandwidth. The bandwidth is maintained at 10 Mbps before the 100th frame, after which it drops to 1 Mbps. In the original design, we set the vbv_buffer_size to 0.1×, 0.2×, 0.5×, 0.7×, and 1.0× of the current bitrate. In contrast, for the adaptive design, we set the vbv_buffer_size to one frame's bitrate (current bitrate / fps) for the subsequent 10 frames following the encoder's detection of the bandwidth drop. The remaining frames retain the same settings as the original design. Figure 4 illustrates the improvements achieved with the adaptive vbv_buffer_size. The results indicate that the tail 90 delay is reduced by 17.43% to 63.08%, while maintaining video quality within -0.8% to -3%, which is imperceptible to the human eye.

We also conducted a simulation experiment to demonstrate the benefits of predicting bandwidth drops. As illustrated in Figure 5, the response time indicates the number of frames required to detect a bandwidth drop. The results show that even with the ability to predict just one frame in advance, the tail 90% delay can be significantly reduced by 12.62% - 30.6%,, while maintaining the same level of video quality. By combining these two factors, we can achieve greater improvements, reducing tail 90 delay by 28.66% to 78.87%, while maintaining video quality or achieving a slight improvement of 0.8% to 3%.

4 Conclusion and Future Work

This poster proposes an adaptive encoder to respond quickly to the change in network bandwidth. The simulation result shows that it can reduce tail 90% delay by 28.66% to 78.87%.

For future work, a challenge is developing a comprehensive strategy for predicting bandwidth drops and adaptively adjusting encoder parameters in response. Furthermore, discrepancies between actual bandwidth and sensed bitrate can cause fluctuations in the encoder's performance during bandwidth drops, potentially impacting the effectiveness of the improvements

References

[1] Sadjad Fouladi, John Emmons, Emre Orbay, Catherine Wu, Riad S Wahby, and Keith Winstein. 2018. Salsify:{Low-Latency} network video through tighter integration between a video codec and a transport protocol. In *15th USENIX Symposium on*

Networked Systems Design and Implementation (NSDI 18). 267–282.

[2] Xiangjie Huang, Jiayang Xu, Haiping Wang, Hebin Yu, Sandesh Dhawaskar Sathyanarayana, Shu Shi, and Zili Meng. 2025. ACE: Sending Burstiness Control for High-Quality Real-time Communication. In *To appear at Proceedings of the ACM SIGCOMM 2025 Conference*.

[3] Alan Jones, Peter Sevcik, and Rebecca Wetzel. 2021. Internet Connection Requirements for Effective Video Conferencing to Support Work from Home and eLearning | NetForecast. (2021). https://www.netforecast.com/wp-content/uploads/NFR5137-Videoconferencing_Internet_Requirements.pdf

[4] Teemu Kämäräinen, Matti Siekkinen, Antti Ylä-Jääski, Wenxiao Zhang, and Pan Hui. 2017. A measurement study on achieving imperceptible latency in mobile cloud gaming. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. 88–99.

[5] Zili Meng, Yaning Guo, Chen Sun, Bo Wang, Justine Sherry, Hongqiang Harry Liu, and Mingwei Xu. 2022. Achieving consistent low latency for wireless real-time communications with the shortest control loop. In *Proceedings of the ACM SIGCOMM 2022 Conference*. 193–206.

[6] Marta Orduna, César Díaz, Lara Muñoz, Pablo Pérez, Ignacio Benito, and Narciso García. 2019. Video multimethod assessment fusion (VMAF) on 360VR contents. *IEEE Transactions on Consumer Electronics* 66, 1 (2019), 22–31.

[7] Farzad Tashtarian, Abdelhak Bentaleb, Hadi Amirpour, Sergey Gorinsky, Junchen Jiang, Hermann Hellwagner, and Christian Timmerer. 2024. {ARTEMIS}: adaptive bitrate ladder optimization for live video streaming. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 591–611.

[8] Anfu Zhou, Huanhuan Zhang, Guangyuan Su, Leilei Wu, Ruoxuan Ma, Zhen Meng, Xinyu Zhang, Xiufeng Xie, Huadong Ma, and Xiaojiang Chen. 2019. Learning to coordinate video codec with transport protocol for mobile video telephony. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.