2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom) | 979-8-3503-3333-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/MetaCom57706.2023.00038

VICTOR: Video Content-aware Partially Reliable Transmission over Multipath QUIC

Yahui Li*, Biao Han*, Xueqiang Han*, Xiaolan Ji*, Congxi Song*, Guo Chen†
*School of Computer, National University of Defense Technology, Changsha, China.
†College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.
liyahui20@nudt.edu.cn, nudtbill@nudt.edu.cn, hanxueqiangnudt@nudt.edu.cn,
jixiaolan16@nudt.edu.cn, songcongxi17@nudt.edu.cn, guochen@hnu.edu.cn

***Abstract*—Over the past few years, video streaming has been dominating the internet traffic. The Quick UDP Internet Connections (QUIC) protocol and its support for both reliable and unreliable transmission may exhibit attractive advantages for video streaming. Multipath transport techniques like Multi-Path QUIC (MPQUIC) are introduced to aggregate the bandwidth of multiple links and to provide reliable transmission in poor network conditions. However, how to leverage unreliable transmission for video streaming over MPQUIC has not been explored yet. Motivated by extensive experimental observations, we propose VICTOR, video content-aware partially reliable transmission over multipath QUIC. Specifically, VICTOR exploits both reliable and unreliable transmission to improve the quality of experience (QoE) perceived by users. Besides, the scheduling mechanism of MPQUIC is examined and VICTOR split video frames over multiple paths with the perception of QoE. We evaluate VICTOR experimentally in both simulations and a lab testbed. The results reveal that VICTOR can achieve better performance compared with the latest MPTCP and MPQUIC protocols.**

***Index Terms*—partially reliable transmission, multipath QUIC, video streaming, QoE, scheduling**

I. Introduction

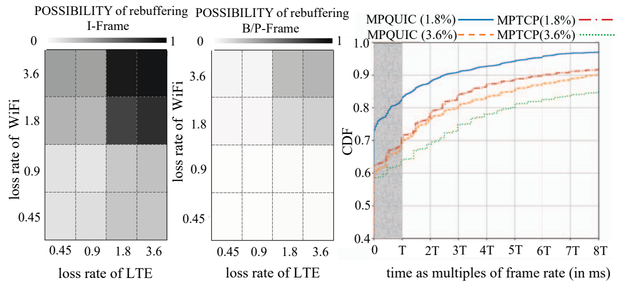
Nowadays, people are spending much more time in online shopping and communication, almost eight hours per week [1], which dominated the mobile Internet traffic by about 65% [2]. Currently, video streaming is commonly delivered by Dynamic Adaptive Streaming over HTTP (DASH) [3] or HTTP Live Streaming (HLS) [4] protocols. One of the drawbacks of DASH and HLS is that they inherit the problems of TCP, which focuses on either *fine-tuning* TCP in the Operating System (OS) kernel or struggling to find a better Adaptive Bit-Rate (ABR) scheme to estimate the bandwidth available to applications in real time. Moreover, the video data has to be transmitted in a reliable manner because of the dependency of HTTP on TCP regardless of whether reliable transmission is really necessary for video streaming.

Over the past few years, Google's Quick UDP Internet Connections (QUIC) has been widely deployed to convey the traffic of Youtube and Chrome browser and accounts for over 30% of Google's total egress traffic [5]. HTTP/3, which runs over QUIC instead of TCP, has been recently standardized by IETF QUIC Working Group (IQWG) as RFC 9114 [6]. As a cross-layer protocol, QUIC overcomes the OS-level support obstacles in tuning TCP and is able to combine transport protocol and video applications to achieve high performance and cost-efficiency at the same time.

Although built upon UDP, QUIC inherits the characteristics of TCP such as reliable transmission [5], [7]. The authors in [9] pointed out that fully reliable transmission of QUIC may not be suitable for video streaming when network conditions are worse than ideal. Meanwhile, QUIC also performs poorly when it encounters packet losses that are not due to congestion (e.g., wireless packets loss). Therefore, RFC 9221 [8] has been released by IQWG as an unreliable datagram extension of QUIC, which motivates the academia and industry to develop live video streaming applications over QUIC. In the coming era of Metaverse, cellular technology even the fifth-generation (5G), although promises to provide high throughput and low latency [10], remains challenging to meet high QoE requirements of ultra high-definition (UHD) video streaming and AR/VR applications [11], [12]. Multipath transmission technologies enable multi-homed devices to establish multiple paths for simultaneous data transmission, which are potential to improve throughput and reliability for video transmission under poor network conditions. Inspired by MPTCP [14], Multi-Path QUIC (MPQUIC) was proposed as an extension of QUIC to enable the ability of transmitting data over multiple paths simultaneously [15], [16]. Recently, multipath extensions of QUIC have received extensive attentions by IQWG [13].

Although unreliable transmission has been introduced into QUIC, how to jointly leverage reliable and unreliable transmissions to support multipath video streaming over MPQUIC has not been investigated yet. One of the challenges is to seamless multiplex reliable and unreliable streams over multipath QUIC connections. In this paper, we first conduct extensive preliminary experiments to evaluate the limitations of reliable transmission in streaming videos over MPQUIC protocol proposed by [15]. Based on the observations from preliminary experiments, we propose the idea of partially reliable transmission, in which reliable and unreliable transmissions are utilized in streaming different video frames through a multipath manner. We call this work VICTOR, **VI**deo **C**ontent-aware par**T**ially reliable transmissi**O**n ove**R** multipath QUIC. In our proposed VICTOR, the video frames are processed differently, i.e., I-Frames are processed with reliable streams while P-Frames and B-Frames are delivered via unreliable datagrams. For multipath transmissions, the default scheduler in [15] sends packets on the path with the minimum Round-Trip Time (minRTT) until this path's congestion window (CWND) is filled, which requires a more fine-grained scheduling policy with perception of video frame priorities. Therefore, we propose an efficient priority-aware multipath scheduler to fully utilize the parallel processing

979-8-3503-3333-6/23/$31.00 ©2023 IEEE
DOI 10.1109/MetaCom57706.2023.00038



(a) Different loss rate (in percentage) combinations corresponding to different rebuffering possibilities of different frame types

(b) CDF of frames arrival times

Fig. 1. (a) The possibility of I-Frame that arrives after its deadline is far in excess of that of B/P-Frame with different loss rate combinations. (b) At a loss rate of 1.8%, just 83% (70%) of frames transmitted over MPQUIC (MPTCP) arrive before the deadline (region shaded in gray).

capabilities of multiple paths. Finally, the effectiveness and efficiency of VICTOR are comprehensively evaluated in both simulations and the wild. The main contributions of this paper are summarized as follows:

- **Key Observations.** We conduct extensive preliminary experiments to reveal the efficiency of multipath transmission and the limitations of fully reliable video streaming with MPQUIC.
- **VICTOR Design.** We exploit both reliable and unreliable transmission for video streaming and propose the idea of partially reliable transmission. As far as we know, we are the first to introduce partially reliable transmission into MPQUIC to optimise video streaming.
- **VICTOR Evaluation.** Extensive experiments are conducted to evaluate BufRatio, RateBuf and aSSIM while deploying VICTOR to stream videos. Results reveal that VICTOR significantly reduce video rebuffering time by 70.3% and 83% in controlled environments when packet loss rate of each path reaches 3.6%, along with 59% and 79% reduction in real networks, compared with the latest MPQUIC and MPTCP protocols, respectively.

II. Background and Motivation

A. The Status Quo

1) Video streaming: For live streaming, the video is transmitted in a frame granularity. If a video frame is not fully received before its deadline, the video will encounter unexpected stall. Meanwhile, video streaming always needs to meet user's demand for QoE. The performance metrics commonly used include buffering ratio (*BufRatio*) and rate of buffering events (*RateBuf*), which stand for the time spent in rebuffering to the total playback duration and the rebuffering events to the total playback duration, respectively [17].

2) Video codecs: Video streams are usually encoded by H.264 codec to reduce the data size [18], in which the compressed video contains a series of GOP [19] and each GOP consists of three types of frames: Intra-coded Frame (I-Frame), Predicted Frame (P-Frame), and Bi-directional predicted Frame (B-Frame). I-Frame is a complete figure and exploits the spatial correlation to reduce the amount of data it contains, which is called intra-prediction. Besides, P/B-Frames are compressed by exploiting temporal correlation between frames as well as intra-prediction.

B. Motivation

1) Priority-aware video streaming: To evaluate the importance of different video frames, we compare the rebuffering possibilities between I-Frame and B/P-Frame over multipath by conducting a preliminary experiment. We stream the famous *Big Buck Bunny* video with MPQUIC over LTE and WiFi links in an emulated network, in which the packets loss rate of LTE and WLAN is varied from 0.45% to 3.6%, respectively. Then we repeat the experiments 10 times and average the results. As Fig. 1 (a) shows, I-Frames experience much later delivery than that of B/P-Frames at almost all loss rate combinations over multipath. For I-Frames, the rebuffering possibility is near to one when the loss rate of LTE and WiFi is high, e.g., each is 3.6%. Therefore, the delivery priority of different frames should be considered. As an independent frame, I-Frames are much larger in data size than that of B/P-Frames. In other words, I-Frame is most likely to encounter packet loss which may block subsequent frames' decoding and playback though the other frames are successfully received. To reduce the possibility of interrupted playback caused by I-Frames, they should be processed with the highest priority while scheduling among multiple paths.

2) Unreliable transmission over MPQUIC: Unreliable streams which deliver packets in a best-effort manner has become a candidate mechanism for QUIC only currently. In order to investigate how fully reliable transport performs for video streaming in multipath tranport protocols, i.e., MPTCP and MPQUIC, we record the arrival time of each video frame and compare it with the frame's deadline. The gray area in Fig. 1 (b) depicts the frames that arrive on time. As shown in Fig 1 (b), although the results show MPQUIC performs better than MPTCP with a loss rate of both 1.8% and 3.6%, there are about 18% and 31% video frames failing to deliver on time. In other words, these frames miss their deadlines when they are required, thereby causing stalls. Motivated by recent works proposed by [9], [22], which introduced unreliable datagrams into QUIC, we are very expect for exploiting unreliable streams to transmit video frames over multipath QUIC, in addition to reliable streams.

Meanwhile, the works in [19]–[21] show that the loss of I-Frame has significant impact on the video streaming QoE. The reason is that P- and B-Frame are encoded differently compared with I-Frame. The loss of I-Frame leading to the failure when decoding the difference of subsequent P- and B-Frames in a GOP. Hence, it is better to deliver I-Frames in a reliable way. On the other hand, a lost packet containing P/B-Frame can be dropped because the recovery process may delay the playback time of the corresponding frame. It leads to traffic overhead and obstructs the subsequent frames' arrival. Consequently, the QoE degenerates during the recovery process. These observations motivate us to process different video frames in distinct manners and we will express the implementation details in next section.

III. Implementation of VICTOR

A. Backward Compatible Unreliable Transmission

Stream is an abstraction in QUIC, which can be regarded as an in-order and reliable data pipe between the client and the server. Initially, we are eager whether there is a simple and backward compatible way to extend another "stream" in QUIC, which provides an ordered but unreliable delivery of videos. To this end, we design unreliable "stream" named *datagram* based on QUIC's characteristic, which is different from RFC 9221 [8]. Firstly, the congestion control is separated from datagram and urgent data over datagram is not limited though it will endure some packets loss. Besides, streams and *datagrams* are individually setup through two different interfaces. The reliable data sent on streams will be written in STREAM Frames and unreliable data sent on *datagrams* will be written in DATAGRAM Frames. Afterwards, DATAGRAM Frames and STREAM Frames will be encapsulated into QUIC packets and sent to the receiver. Besides, all the DATAGRAM frames in a lost QUIC packet will not be re-transmitted at the sender. Hence, if some QUIC packets that contain DATAGRAM frames are lost, there will be gaps when the corresponding *datagram*'s data is reordered at the receiver and the gaps will be filled with zero. On the other hand, multipath QUIC is an extension to QUIC , the design of *datagram* is compatible with streams in MPQUIC. In Sec. VI, the experimental results show that our design of *datagram* works well in MPQUIC.

B. A Priority-aware Multipath Scheduler

VICTOR simultaneously provides reliable and unreliable transmission through two interfaces so that *datagram* and stream are handled independently. However, there are many challenges for implementation. By extracting the video frame's header, we can identify the video types of I-Frames and P/B-Frames. Therefore, all the videos frames are parsed so that I-Frames will be delivered via streams, while the remaining frames of the GOP will be delivered via *datagrams*. As it is required to reorder the arrived video frames of a GOP at the receiver, we set up another stream to transmit the arrangement information. For example, the order of the video frames and their playback timestamp. The gaps in the P/B-frames' buffer will be filled with zero[1] once the corresponding GOP's deadline comes. In VICTOR, a video frame will be submitted to the video application if it is required, though some data in P/B-Frames is lost. However, in default MPTCP or MPQUIC, the frames will be blocked. Note that video stall still occurs in VICTOR if I-Frame experiences packet loss though its deadline arrives. The coexist of datagrams and streams in a QUIC connection and the cross-layer design bring further improvements for MPQUIC. For the packet scheduler, I-Frames are treated with higher priority than P/B-Frames. As Fig. 2 shows, QUIC frames containing I-Frames will be allocated on the fast path firstly unless the path's congestion

[1]Many video players (e.g, VLC) are capable of decoding frames with zero filled zones.

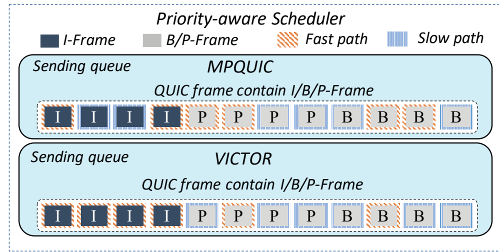


Fig. 2. VICTOR prioritizes sending QUIC frames with I-Frame on the fast path.

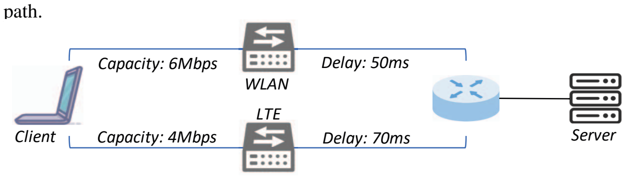


Fig. 3. The simulation topology.

window is filled. The other QUIC frames containing P/B-Frames will be allocated on the rest of the fast path and the slow path afterwards. In other words, I-Frames will be delivered to the receiver before other B/P-Frames. As a result, B/P-Frames' decoding will not be blocked because I-Frames always arrives earlier. Currently, the fast path is still indicated by the RTT of paths in consideration of timely arrival. When all the paths' congestion window is filled and there is not existing QUIC frames consist of I-Frame, the remaining QUIC frame consists of P/B-Frame will be delivered in a best-effort though they may endure packets loss. They are transmitted on the first path with the smallest RTT until the first path's RTT gets larger than the second path. If there is I-Frame waiting for sending and all the paths' congestion window is filled, I-Frame's sending is blocked and B/P-Frame will not be transmitted because I-Frame should be delivered before them. Besides, the arrangement information is also written in the stream frame and shared the same priority with I-Frame, because it should be delivered earlier to arrange all the video frames of a GOP. The arrangement information only accounts a small amount of data in the video file.

IV. Performance Evaluation

A. Experiment Setup

1) Video Set: *Big Buck Bunny* is a typical video widely used in the related works [22], [23]. Therefore, we deliver *Big Buck Bunny* with different definitions (as shown in Tab. I) in controlled and real environment, respectively.

TABLE I
Video Set Used: Resloution (Res); Average Bitrate (Abr) and Maximum Bitrate (Mbr), in Mbps; Duration (Dur), in second; and Size, in MB.

Video Name	Video Information				
	Res	***Abr***	***Mbr***	***Dur***	***Size***
Big Buck Bunny (HD)	1920×1080	4.4	8.34	100	55
Big Buck Bunny (UHD)	3840×2160	12.32	24	100	154

2) Evaluation Metrics: In the experiments, **BufRatio** and **RateBuf** of VICTOR are compared with those of MPQUIC and MPTCP. However, transmission of P/B-Frame over *datagram* will lead to inevitable picture distortion compared with

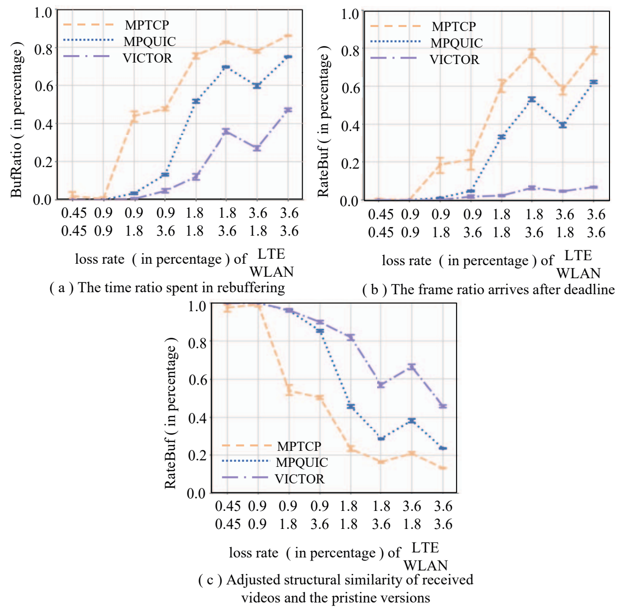


Fig. 4. Video streaming BuffRatio, RateBuf, aSSIM with different loss rate combinations on WLAN and LTE paths in controlled environment.

its pristine version due to lost packets. The structural similarity (SSIM) [24] is introduced to assess the degree of this "distortion" objectively. SSIM ranges from 0 to 100, with higher values representing higher similarity. However, the rebuffering time brings negative influence on the QoE. Here, we apply another metric called **adjusted SSIM (aSSIM)** introduced in [9]. The difference between aSSIM and SSIM is aSSIM will assign each frame over a stall duration an SSIM index of zero.

3) Simulation Setup: We evaluate the efficiency of VICTOR in Mininet [25] and the simulation topology is as shown in Fig. 3. The server streams *Big Buck Bunny (HD)* through two paths with different link capacities and delays. Here, we evaluate VICTOR's capability of sustaining better QoEs under different path losses. Therefore, the loss rate of LTE and WLAN is respectively set as 0.45%, 0.9%, 1.8% and 3.6% in each experiment. The maximum receive window of (MP)QUIC and (MP)TCP are uniformly set to 16 MB. Besides, we employ minRTT [15] as the packet scheduler and OLIA [26] as the congestion control algorithm for both MPTCP and MPQUIC.

4) Real-world Experiment Setup: We deploy a cloud server with 100 Mbps down-link bandwidth, which is sufficient for streaming the UHD version of *Big Buck Bunny*. The software version is consistent with those in the simulation. The performance of VICTOR is evaluated through LTE and WLAN wireless interfaces within a campus laboratory.

B. Simulation Results

As shown in Fig. 4 (a), when the packet loss rate of both LTE and WLAN is lower than 0.9%, the rebuffering time of three proctols spent in playback is less than 2%. In other words, they all succeed in making full use of multipath links, thereby achieving great link aggregation in an ideal

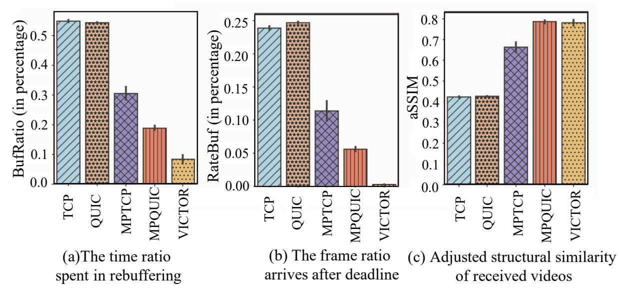


Fig. 5. Video streaming performance over WLAN and LTE wireless interfaces within a campus laboratory.

network environment. However, once the loss rate of any path increases, the congestion window of that path shrinks immediately, thereby resulting in lower aggregation efficiency and QoE degeneration. In Fig. 4 (a), we observe that the BufRatio of MPTCP and MPQUIC increases to 48% and 13% when WLAN experiences a higher packet loss rate from 0.9% to 1.8%, even though LTE still maintains 0.9%. Similar results are obtained in Fig.4 (b), more video frames arrives after their deadline when the loss rate increases, which causes frequent playback interruption during video streaming. As for VICTOR, however, the lost packets will not be retransmitted by the sender and the receiver still submits the received frames to the application, thereby leading to significant BufRatio and RateBuf reduction. Moreover, the results of aSSIM in Fig. 4 (c) show that it will not bring too much video distortion. The benefits of unreliable transmission on the fluency of video playback make up for picture distortion to some extent. This is reasonable because rebuffering time is the most dominant metric related to user engagement, as mentioned in [17].

C. Real-world Experiment Results

For real-world experiments, single-path protocols such as TCP and QUIC are far from satisfying UHD video delivery because of the scant link capacity. As shown in Fig. 5 (a)-(c), TCP and QUIC are further exceeded by mulitpath protocols. MPQUIC outperforms MPTCP because a wider SACK (Selective Acknowledgement) range is utilized in QUIC which enables MPQUIC recovers quickly from packets loss. On the basis of MPQUIC, VICTOR further reduces the BufRatio from 18% to 8%. The frames received after their deadline still remain a low level compared with MPTCP and MPQUIC, which is shown in Fig. 5 (a) and Fig. 5 (b). Besides, the results of aSSIM in Fig. 5 (c) indicate VICTOR achieves roughly equivalent video quality with MPQUIC and higher video quality than MPTCP.

V. RELATED WORK

There is lots of previous work about video streaming or partially reliable protocols. They discuss the methods to improve the QoE by estimating the available bandwidth or adjusting the protocol's reliability. Here, we briefly introduce some typical work which is most relivent to our work.

ABR Schemes. ABR scheme aims to estimate the available bandwidth in real time in user space. Spiteri et al. [27]

proposed an algorithm called BOLA by formulating ABR schemes as a utility optimization problem of video rebuffering time and average video bitrate. Xing et al. [28] investigated the DASH technique over multiple links and formulate the video streaming process over multiple links as a finite-state Markov Decision Process (MDP) problem to avoid playback interruption. However, these schemes are at the application layer without awareness of fluctuating network characteristics, leading to inaccurate estimations.

Partially Reliability. Palmer et al. in [9], [22] introduced unreliable transmission into QUIC, but how to jointly leverage reliable and unreliable transmissions to support multi-path video streaming is far from exploration. Xiaohui et al. proposed a partially reliable protocol called APRT in [29] which dynamically re-transmits the lost packets in order to make a tradeoff between instantaneity and video quality. However, APRT is just designed for the lossy and long delay satellite channels and is not adaptive to other scenes. McQuistin et al. proposed a TCP variant called TCP Hollywood in [30] and re-transmitted packets are used to deliver new data. However, it has not discussed how to leverage multiple paths.

VI. CONCLUSION

In this paper, we propose the idea of partially reliable transmission for MPQUIC on the basis of extensive experiments. Afterwards, a priority-aware multipath scheduler is introduced to VICTOR. With the experiments in both simulations and real networks, VICTOR has made obvious improvements on the quality of experiences users perceived. In the future, we will consider cross-layer multipath optimization for other applications to explore VICTOR's performance in depth.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No.62102430), Hunan Young Talents Grant (No. 2020RC3027) and Training Program for Excellent Young Innovators of Changsha (No. kq2206001).

REFERENCES

[1] Limelight, "The state of online video," Aug. 2020. [Online]. Available: https://www.limelight.com/resources/market-research/state-of-online-video-2020

[2] Sandvine, "Mobile internet phenomena report," Feb. 2020. [Online]. Available: https://www.sandvine.com/press-releases/sandvine-releases-2020-mobile-internet-phenomena-report-youtube-is-over-25-of-all-mobile-traffic

[3] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE multimedia*, vol. 18, no. 4, pp. 62–67, 2011.

[4] R. Pantos and W. May, "Http live streaming, rfc 8216," *IETF*, 2017.

[5] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

[6] M. Bishop *et al.*, "Hypertext transfer protocol version 3 (http/3)," *Internet Engineering Task Force, RFC 9114*, 2022.

[7] R. Hamilton, J. Iyengar, I. Swett, A. Wilk *et al.*, "Quic: A udp-based secure and reliable transport for http/2," *IETF, draft-tsvwg-quic-protocol-02*, 2016.

[8] T. Pauly, E. Kinnear, and D. Schinazi, "An unreliable datagram extension to quic," *Internet Engineering Task Force, draft-pauly-quic-datagram*, 2022.

[9] M. Palmer, T. Krüger, B. Chandrasekaran, and A. Feldmann, "The quic fix for optimal video streaming," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2018, pp. 43–49.

[10] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, N. Dai, and H.-S. Lee, "Furion: Engineering high-quality immersive virtual reality on today's mobile devices," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1586–1602, 2019.

[11] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z.-L. Zhang, "A first look at commercial 5g performance on smartphones," in *Proceedings of The Web Conference 2020*, 2020, pp. 894–905.

[12] J. Sommers and P. Barford, "Cell vs. wifi: on the performance of metro area mobile connections," in *Proceedings of the 2012 internet measurement conference*, 2012, pp. 301–314.

[13] Y. Liu, Y. Ma, C. Huitema, Q. An, and Z. Li, "Multipath extension for quic," *Proceedings of the IETF Internet Draft, Berlin, Germany*, vol. 17, 2021.

[14] A. Ford, C. Raiciu, M. Handley, O. Bonaventure *et al.*, "Tcp extensions for multipath operation with multiple addresses," 2013.

[15] Q. De Coninck and O. Bonaventure, "Multipath quic: Design and evaluation," in *Proceedings of the 13th international conference on emerging networking experiments and technologies*, 2017, pp. 160–166.

[16] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath quic: A deployable multipath transport protocol," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[17] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM computer communication review*, vol. 41, no. 4, pp. 362–373, 2011.

[18] Bitmovin, "Global media format report 2018," Sep. 2019. [Online]. Available: https://bitmovin.com/bitmovin-2019-video-developer-report-av1-codec-ai-machine-learning-low-latency/

[19] Á. Huszák and S. Imre, "Analysing gop structure and packet loss effects on error propagation in mpeg-4 video streams," in *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*. IEEE, 2010, pp. 1–5.

[20] N. Feamster and H. Balakrishnan, "Packet loss recovery for streaming video," in *12th International Packet Video Workshop*. PA: Pittsburgh, 2002, pp. 9–16.

[21] P. Orosz, T. Skopkó, and P. Varga, "Towards estimating video qoe based on frame loss statistics of the video streams," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1282–1285.

[22] M. Palmer, M. Appel, K. Spiteri, B. Chandrasekaran, A. Feldmann, and R. K. Sitaraman, "Voxel: cross-layer optimization for video streaming with imperfect transmission," in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 2021, pp. 359–374.

[23] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd multimedia systems conference*, 2012, pp. 89–94.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[25] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 253–264.

[26] R. Khalili, N. Gast, M. Popovic *et al.*, "Opportunistic linked-increases congestion control algorithm for mptcp," 2013.

[27] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[28] M. Xing, S. Xiang, and L. Cai, "A real-time adaptive algorithm for video streaming over multiple wireless access networks," *IEEE Journal on Selected Areas in communications*, vol. 32, no. 4, pp. 795–805, 2014.

[29] X. Li and J. Wang, "Elastically reliable video transport protocol over lossy satellite links," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 1097–1108, 2018.

[30] S. McQuistin, C. Perkins, and M. Fayed, "Tcp hollywood: An unordered, time-lined, tcp for networked multimedia applications," in *2016 IFIP networking conference (IFIP networking) and workshops*. IEEE, 2016, pp. 422–430.