2024 IEEE/CIC International Conference on Communications in China (ICCC) | 979-8-3503-7841-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCC62479.2024.10681889

Transmission Protocol Customization for On-demand Tile-based 360° VR Video Streaming

Yannan Wei*, Qiang (John) Ye†, Kaige Qu*, Weihua Zhuang* and Xuemin (Sherman) Shen*
* Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada
† Department of Electrical and Software Engineering, University of Calgary, Calgary, AB, Canada
Email: {y272wei@uwaterloo.ca, qiang.ye@ucalgary.ca, k2qu@uwaterloo.ca, wzhuang@uwaterloo.ca, sshen@uwaterloo.ca}

***Abstract*—Tile-based streaming has been proposed to address the challenge of high transmission rate demand in 360° virtual reality (VR) video streaming. However, it suffers from network condition dynamics and viewer field-of-view (FoV) prediction errors. In this paper, we propose a customized transmission protocol based on Quick UDP Internet Connections (QUIC) which operates over a VR video core network slice to support enhanced slice-level VR video data transmission. The QUIC protocol is tailored to accommodate the characteristics of tile-based VR video streaming where explicit mapping relations between requested video tiles and QUIC streams are established. Two customized protocol functionalities including packet filtering and caching-based packet retransmission are proposed, which filter out outdated video data due to FoV prediction errors and achieve efficient packet loss recovery, respectively. A slice-level packet header is designed to support enhanced slice-based VR video packet transmissions with the proposed protocol functionalities. Simulation results are presented to demonstrate the effectiveness of our proposed protocol.**

***Index Terms*—360° VR video streaming, transmission protocol, QUIC, tile-to-stream mappings, packet filtering.**

I. INTRODUCTION

Due to immersive user experience and enormous vertical markets such as in gaming and education, 360° virtual reality (VR) video streaming has attracted significant attention from both academia and industry. It poses technical challenges due to requirements of extremely high transmission rate and low latency. A viewer wears a head-mounted display (HMD) to watch a panoramic video. At any time instant, the viewer watches only part of a VR video due to the limited span of an HMD (e.g., 120°×120°), referred to as field-of-view (FoV). To address the challenge of high transmission rate demand, tile-based VR video streaming has been proposed [1]. At the server side, a VR video is split into multiple non-overlapping video tiles. At the viewer side, head movements are tracked for FoV prediction. Based on the FoV prediction results and the estimated transmission rate, a viewer selectively requests video tiles with different bitrates.

Tile-based VR video streaming suffers from both transmission rate variations and viewing behavior dynamics which are constantly driven by head movements and are vulnerable to FoV prediction errors. In addition, encoded video tiles have various properties in terms of transmission priority, deadline, and reliability requirement. Each video tile can be encoded into one base layer (BL) and multiple enhancement layers (ELs). BL tiles ensure video smoothness and have a higher transmission priority than EL tiles that enhance video quality. Each requested video tile has a strict deadline to be delivered. When FoV is not accurately predicted or viewers suddenly rotate heads, a viewer needs to request additional urgent video tiles to compensate the current viewing experience. Urgent video tiles have a much smaller deadline, called the *motion-to-photon (MTP)* latency requirement (usually less than 20 ms), than that of regular video tile delivery (around 100 ms) [2]. In terms of reliability, BL packets have a higher transmission reliability requirement than EL packets. Deadline-violated EL packets can be directly discarded without retransmissions, while BL packets need to be reliably transmitted. Therefore, in order to stream smooth and high-quality VR videos to users, a supporting transmission protocol is imperative to accommodate various video tile properties and promptly react to network and viewing behavior dynamics (i.e., head movements).

Existing studies on protocol design for VR video streaming conduct prioritized transmissions among requested video tiles by considering properties such as priority, playback deadline, and remaining tile size [3]. The stream multiplexing and prioritized stream scheduling features of the Quick UDP Internet Connections (QUIC) are also leveraged [4]. Most existing works focus on improving protocol operations at end hosts, which may result in slow responsiveness to network dynamics (e.g., increasing congestion level) and viewing behavior dynamics. With the software-defined networking (SDN) and network function virtualization (NFV) technologies, multiple virtual networks, also known as *network slices*, can be created over a shared physical network for supporting diversified services [5]. A network slice with flexible resource orchestration can be created to support the VR video streaming service for finer-grained quality-of-service (QoS) guarantee, where optimal routing path(s) can be established with customized protocol functionalities and operations implemented on virtual nodes [6].

To better support the VR video streaming service, some research issues regarding the transport protocol design should be investigated. First, various video tile properties, not originally supported in QUIC, need to be considered and reflected in protocol operations. Correspondingly, the mapping relations between requested video tiles and QUIC streams need to be determined, and how to conduct stream multiplexing and QUIC packet assembly needs to be revisited. Second, differentiated packet loss recovery mechanisms are required for BL and EL packets, respectively, and a slice-level transmission scheme is needed to support VR video data transmission with fine-grained QoS guarantee. Third, when viewers unexpectedly rotate heads and FoV prediction errors occur, some requested video tiles for the outdated (predicted) FoV are no longer needed. En-route VR video packets may contain outdated data which needs to be

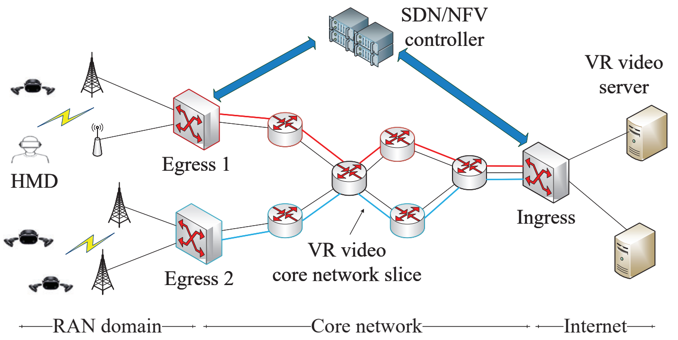


Fig. 1. The SDN/NFV-based E2E network scenario.

dropped to save transmission resources. Thus, a packet filtering functionality is desired to filter out outdated video data from packets.

To deal with the above issues, in this paper, we present an SDN/NFV-based transmission protocol based on QUIC to support enhanced on-demand VR video streaming, which operates over a VR video core network slice. The QUIC protocol is first tailored to accommodate the characteristics of tile-based VR video streaming. Customized protocol functionalities including packet filtering and caching-based packet retransmission are proposed which filter out outdated video data due to FoV prediction errors and achieve efficient packet retransmission with disparate reliability requirements, respectively. A slice-level packet header is designed to support enhanced slice-based VR video data transmission with the proposed protocol functionalities.

II. System Model

A. Network Model

An SDN/NFV-based end-to-end (E2E) transmission network is considered to support VR video streaming service delivered from remote VR video servers to video clients, as shown in Fig. 1. A VR video slice is deployed between each pair of ingress and egress edge switches (referred to as nodes) in the core network for supporting aggregated VR video traffic. Specifically, with the SDN/NFV controller, a dedicated virtual network topology is configured for each slice (the topology is assumed linear for simplicity), and virtual resources including link transmission capacity and processing resources are reserved for data transmission and processing. In addition, customized protocol functionalities are designed and enabled at certain nodes within each slice to achieve finer-grained VR video service provisioning (to be elaborated in Subsection II-C).

E2E VR video transmission under the SDN/NFV architecture traverses three network segments: (i) from a VR video server on the Internet to an ingress node, (ii) over a VR video slice, and (iii) from an egress node to a video client. Here, we focus on the core network and aim to develop a slice-level transmission protocol for supporting enhanced VR video data transmission. The proposed protocol operates over a VR video core network slice, while we adopt the QUIC as our protocol design base for E2E video data transmission [7].

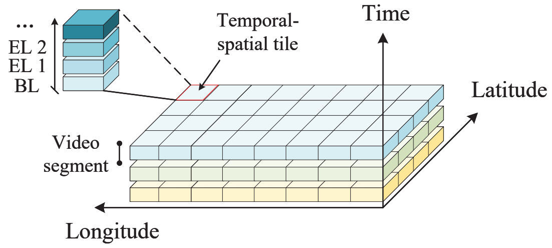


Fig. 2. Tile-based VR video encoding.

B. Video Traffic Model

Tile-based video encoding based on scalable high-efficiency video coding (SHVC) is adopted [8]. As shown in Fig. 2, a VR video is temporally divided into a sequence of video segments, each with a playback time of typically 2-10 seconds. Each video segment is spatially partitioned into multiple non-overlapping tiles. Each video tile is further encoded into multiple interdependent layers, including one BL and several ELs, corresponding to different video qualities (or bitrates) for adaptive streaming.

Suppose encoded VR videos are stored on remote content servers on the Internet. Video clients progressively download from remote servers on a video segment basis. For each video segment downloading, an HMD selectively requests a set of video tiles with different numbers of layers, based on factors such as the estimated throughput and tracked head movement traces. Video tile requests for a video segment are concurrently sent via different streams in a QUIC connection. An HMD may send additional (urgent) requests when unexpected head movements or FoV prediction errors occur. In addition, we consider that each server streams full-view BL for video robustness and only the EL tiles covering a client's FoV.

C. Main Protocol Functionalities

In order to support enhanced VR video packet transmission, two customized protocol functionalities including packet filtering and caching-based packet retransmission are proposed. Specifically, the packet filtering functionality is enabled at both the ingress and egress nodes of a VR video slice to filter out outdated (EL) video data that is no longer needed by clients due to unexpected FoV switching or FoV prediction errors. The ingress and egress nodes are assumed to be equipped with higher-layer protocol header parsing to identify whether a VR video packet passing through contains outdated data.

In addition, the ingress node has an additional caching buffer for aggregated VR video traffic of each VR video slice. The ingress node temporarily stores video packets sent but not acknowledged in the caching buffer for possible retransmissions. The egress node detects any random BL/EL packet loss occurring in the core network and triggers the retransmission from the ingress node using the cached copy in the caching buffer. Besides, due to the disparate transmission reliability requirements, deadline-violated EL packets do not improve video quality and should be directly discarded without retransmissions. Therefore, we propose to design a dummy packet that is generated by intermediate switches to indicate dropped EL packets due to deadline violation, such that the egress node can differentiate EL packet losses due to deadline violation from those due to random link failures. The detailed designs are discussed in the next section.

III. PROTOCOL CUSTOMIZATION FOR ENHANCING VR VIDEO DATA TRANSMISSION

In this section, we elaborate on our main designs in developing a customized slice-level transmission protocol for supporting enhanced VR video data transmission.

A. Tailored QUIC

To accommodate various VR video packet properties and adapt to viewing behavior dynamics, we revisit the QUIC protocol and tailor it from the following three aspects:

- Mapping relations between streams in a QUIC connection and requested video tiles by a client (or an HMD) are established. As shown in Fig. 3, a one-to-one mapping between each EL video tile and each stream is established, and separate streams are used to transmit full-view BL. This is achieved by using one stream to carry only one EL video tile request and using different streams to carry BL video tile requests. Assume the mapping relations between streams and video tiles are known and kept at both the HMD and the server. With the established tile-to-stream mappings, operations to a specific video tile are realized by controlling the data transmission over its corresponding stream, which is helpful for adapting to viewing behavior dynamics caused by head movements.
- Multiple stream groups, each of which consists of the streams corresponding to the requested video tiles of the same (BL/EL) layer, are formed for QUIC packet assembly. One stream group is selected each time to assemble a QUIC packet where round-robin is conducted among the streams within the same group for filling each STREAM frame in a QUIC packet. The approach proposed in [9] can be applied to decide which stream group is selected to assemble a QUIC packet, where a *block* in [9] represents a stream group in our case. By doing so, STREAM frames (or video data) assembled in a QUIC packet come from the requested video tiles with the same properties. Video data of different encoding layers is assembled into different types of video packets, including regular BL/EL packets and urgent EL packets, and thus the same set of operations can be enforced on each type of packets.
- Three optional fields, *Timestamp (TS)*, *Deadline (DDL)*, and *Priority*, are added to the QUIC packet header to indicate each packet's transmission priority and deadline.

B. Slice-level Video Data Transmission

A slice-level packet header is designed to support aggregated VR video traffic transmitted over a VR video slice with customized protocol functionalities in the core network. To be compatible with the tailored QUIC implemented at end hosts, header conversion and reversion are performed at the ingress and egress nodes, respectively. Specifically, a slice-level packet header is appended to each video packet arriving at the ingress node, and the slice-level packet header is removed by the egress node before the packet is transmitted to a client.

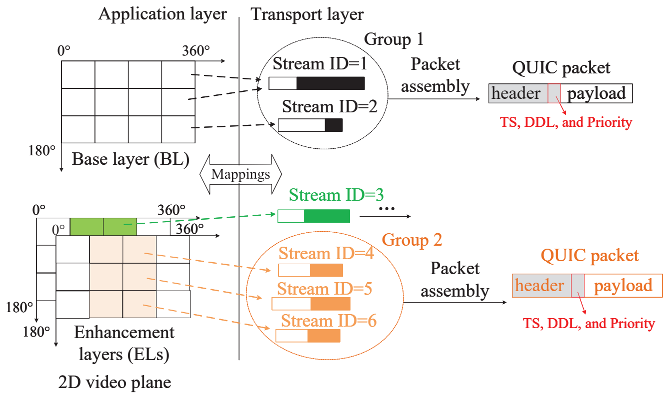
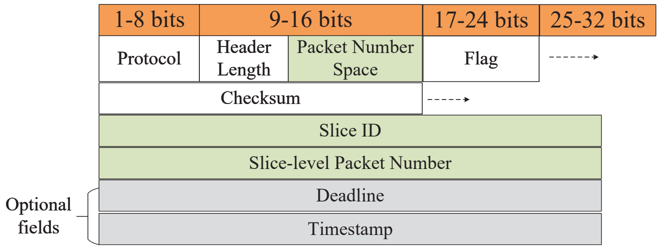


Fig. 3. Tailored QUIC with tile-to-stream mappings.



1-8 bits	9-16 bits		17-24 bits	25-32 bits
Protocol	Header Length	Packet Number Space	Flag	------→
Checksum			------→	
Slice ID				
Slice-level Packet Number				
Deadline				
Timestamp				

Fig. 4. The slice-level packet header format.

A slice-level packet header format is given in Fig. 4 which includes three important fields, i.e., *Slice ID*, *Packet Number Space*, and *Slice-level Packet Number*. Specifically, the Slice ID filed is used for slice identification and packet forwarding, upon which VR video packets are transmitted along the preconfigured routing path of the indicated VR video slice. Besides, two separate packet number spaces are used to differentiate between BL and EL packets with disparate transmission reliability requirements. The Packet Number Space field achieves logical isolation and thus allows for respective operations such as packet retransmission to BL/EL packets. BL/EL packets sent by the ingress node are sequentially numbered through the Slice-level Packet Number field and sent in their respective packet number space to maintain ordered slice-level packet transmissions.

As shown in Fig. 5, when VR video packets arrive at the ingress node, header conversion takes place as described above. During slice-level packet transmissions, intermediate switches transmit packets according to the first-in-first-out (FIFO) principle and directly discard any EL packet that violates its transmission deadline. Once VR video packets reach the egress node, header reversion is performed. Due to possible deadline-violated EL packet dropping, outgoing video packets from the egress node may be out-of-order (OFO). To address this, each reverted VR video packet is renumbered by modifying the *Packet Number* field in the QUIC packet header. Specifically, the egress node records for each E2E connection the packet number of the most recently sent video packet, i.e., the video packet sent with the largest packet number. Then, for each reverted video packet to be transmitted, the modified packet number is the largest packet number recorded plus one.

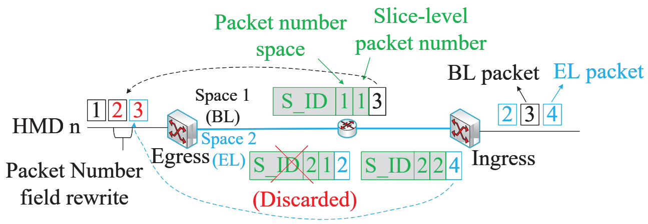


Fig. 5. Video packet transmissions over a VR video slice.

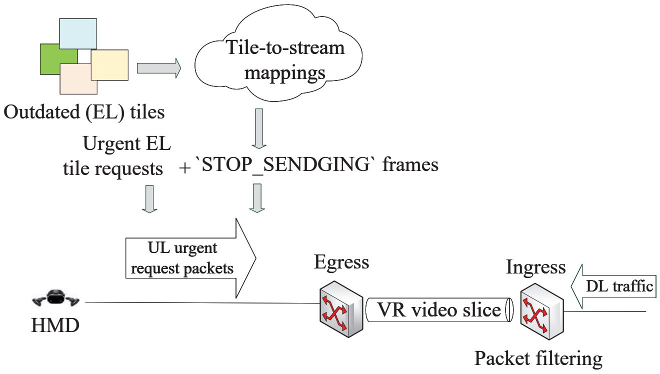


Fig. 6. The workflow of video packet filtering.

Type (i) = 0x05	Stream ID (i)	Application Protocol Error Code (i)

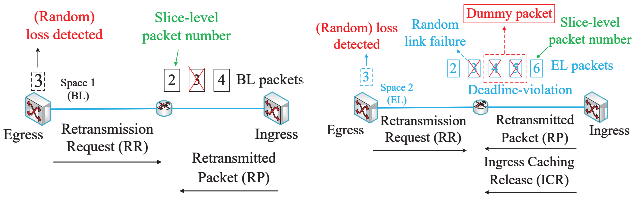
Fig. 7. The STOP_SENDING frame format.

C. Video Packet Filtering

As shown in Fig. 6, the main idea for achieving the packet filtering functionality is to leverage the *Stream ID* field in STOP_SENDING frame (see Fig. 7) defined in the QUIC to tag the set of outdated video tiles, based on the established tile-to-stream mappings in Subsection III-A. The Stream IDs in STOP_SENDING frames sent by an HMD, which can be multiplexed with urgent EL tile requests to form uplink urgent request packets, are extracted by the ingress and egress nodes to identify and filter out outdated (EL) video data. Specifically, STREAM frames with the same IDs as those extracted from the STOP_SENDING frames are ruled out from packets upon reception while others stay.

D. Packet Loss Recovery

The Packet Number Space field in the designed slice-level packet header opens us the opportunities to design differentiated packet loss recovery schemes for BL and EL packets due to their disparate transmission reliability requirements.



(a) Lost BL packet retransmission (b) Lost EL packet retransmission

Fig. 8. Packet loss recovery for BL and EL packets.

Since BL/EL packets are numbered and sent in order in their respective packet number space, OFO packet arrivals at the egress node indicates a random packet loss. Therefore, for achieving the caching-based packet retransmission functionality, our basic idea is to leverage packet OFO for random packet loss detection. As Fig. 8 shows, when a random BL/EL packet loss is detected, the egress node sends a **Retransmission Request (RR)** packet to the ingress node which retransmits the lost packet from its caching buffer. Besides, since deadline-violated EL packets require no retransmissions, a dummy packet is generated by an intermediate switch to indicate consecutive EL packet losses due to deadline violation and thus differ from EL packet losses due to random link failure. Particularly, if a requested EL packet is already discarded from its caching buffer due to deadline violation, the ingress node responds with an **Ingress Caching Release (ICR)** packet to avoid meaningless retransmission requests from the egress node.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the effectiveness of our proposed transmission protocol based on real data traces[1] [10].

A. Data Preprocessing

The selected data traces contain viewing orientations of 50 subjects watching ten 360° VR videos from YouTube with Oculus Rift DK2 being the HMD. The equirectangular projection (ERP) is adopted for 360° video encoding. For each 1-min 360° video, we extract the first 30s for experiments. The viewing trajectories are given in radians in terms of yaw (from $-\pi$ to π) and pitch (from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$). In the ERP-formatted 2D plane, we first conduct sphere-to-plane coordinate transformation based on Eq. (1) and Eq. (2), which maps the yaw (θ_i) and pitch (φ_i) of spherical videos to the horizontal ($w_i \in [0°, 360°]$) and vertical ($h_i \in [0°, 180°]$) coordinates in the 2D video plane.

$$\theta_i = w_i \cdot \frac{2\pi}{W} - \pi, W = 360° \tag{1}$$

$$\varphi_i = h_i \cdot \frac{\pi}{H} - \frac{\pi}{2}, H = 180° \tag{2}$$

In addition, we consider a 4×8 tiling layout, as shown in Fig. 9. The panoramic scene is partitioned into 32 video tiles, each of which covers a $45° \times 45°$ view span and is indexed in raster-scan order. We consider an FoV of $100° \times 100°$ and map a specific FoV to the video tile IDs it covers. Finally, we add Gaussian noises to the selected data traces to generate noised traces with FoV prediction errors for simulation purposes [11].

B. Simulation Settings

The considered network scenario in the simulation is shown in Fig. 10. VR video clients download 360° video segments from remote servers on the Internet where QUIC is implemented with *aioquic*[2]. The integrated video traffic traverses a VR video slice between a pair of ingress and egress nodes in the core network. During the network operation, we throttle the link capacity (in packet/s) of intermediate switch s_1 available to the considered VR video slice to indicate the case of network congestion due to cross-traffic from other services.

---

[1] https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour

[2] https://github.com/aiortc/aioquic

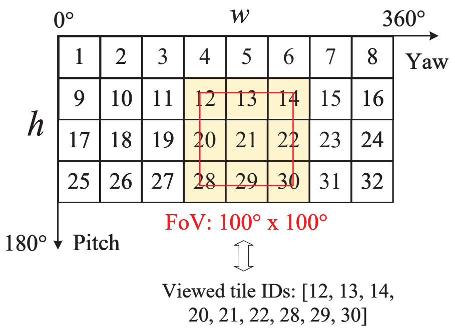


Fig. 9. The considered video tiling layout.

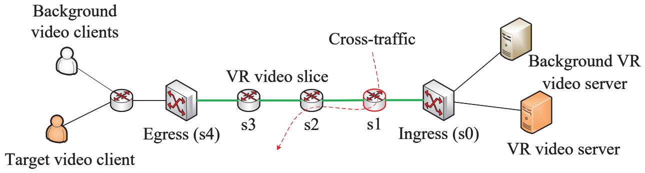


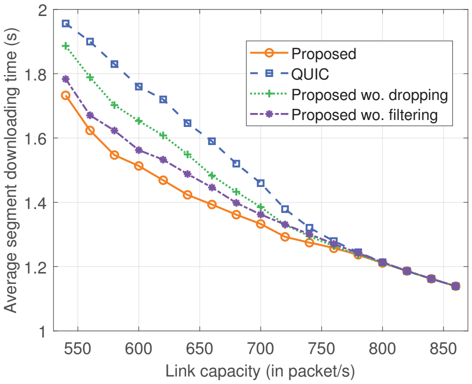
Fig. 10. The considered network scenario in the simulation.

On the other hand, the duration of each video segment is set as 2 seconds [12]. Each video client requests video content on a segment basis, and is considered to progressively prefetch 1 video segment ahead only. Specifically, for each video segment, a video client requests the full-view BL and only the EL tiles covered by the FoV. In view of decoding dependency between BL and EL, we consider that BL packets are sent before EL packets by the VR video servers. For simplicity, we assume each video client unexpectedly rotates head at most once in the middle of watching a video segment. If a video client unexpectedly rotates head, additional EL tiles corresponding to the current FoV are requested for compensating the current viewing experience. Besides, outdated (EL) video tiles corresponding to the previously predicted FoV will be filtered out due to viewpoint prediction errors, and additional EL tiles corresponding to the updated FoV are requested.

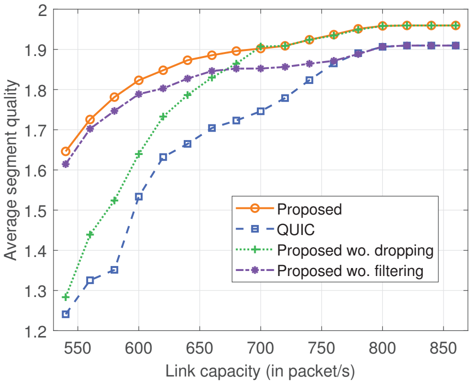
C. Simulation Results

Fig. 11(a) and Fig. 11(b) show the average (video) segment downloading time and quality with link capacity, respectively. The segment downloading time is defined as the time to receive all the BL tiles, indicating the time when a video segment is ready to play smoothly with basic quality. The segment quality is defined by the ratio of the number of expected EL tiles timely delivered to the total number of expected EL tiles requested, given that the corresponding BL tiles have already been received. We can see from Fig. 11 that as link capacity increases, the average segment downloading time reduces, and the average segment quality improves, where our proposed transmission protocol achieves better performance compared to the benchmarks. As link rate increases, it takes shorter time to receive all the BL tiles of a segment, and more EL packets can be timely delivered to the target video client, leading to smaller segment downloading time and improved segment quality.

Originally, QUIC is a reliable transmission protocol which ensures the reliable delivery of packets sent by a server. However, deadline-violated and outdated EL packets do not improve video quality, which wastes link transmission resources and aggravates network congestion instead. In our proposed transmission protocol, differentiation between BL and EL packets in terms of transmission reliability requirement is achieved and reflected in the proposed slice-level packet header. Deadline-violated EL packets are directly discarded. In addition, with the proposed packet filtering functionality, outdated EL packets are filtered out. Therefore, deadline-violated and outdated EL packets will not affect the BL/EL packet transmissions of subsequent segments, while link transmission resources can be saved to transmit those expected packets that may still be able to be timely delivered in a congestion-mild environment. Hence, our proposed protocol achieves better performance than QUIC.



(a) Segment downloading time.



(b) Segment quality.

Fig. 11. Average segment downloading time and quality vs. Link capacity.

In addition, for the proposed protocol without packet filtering functionality (Proposed wo. filtering), deadline-violated EL packets are discarded without affecting the BL tile downloading of the next segment, thus achieving smaller segment downloading time than that when the proposed protocol without (deadline-violated EL) packet dropping is applied (Proposed wo. dropping). In terms of segment quality, as link capacity is small, there will be many deadline-violated EL packets. The proposed protocol without packet filtering thus achieves better segment quality than the proposed protocol without packet dropping. As link capacity increases, the number of deadline-violated EL packets is small. In this case, filtering out outdated EL packets due to unexpected FoV switching brings larger segment quality improvement. Contrarily, if only packet dropping is enabled, part of saved transmission resources is still used to transmit outdated EL packets. Therefore, the proposed protocol

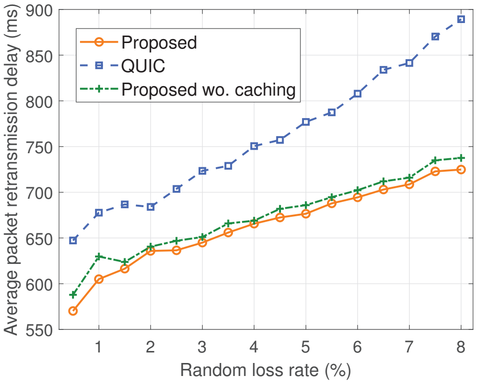
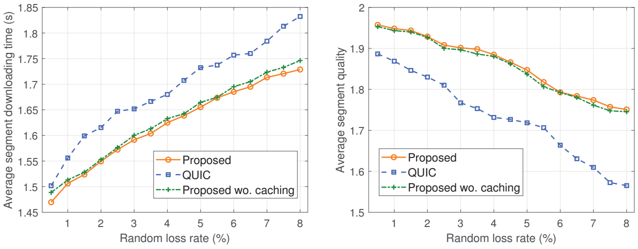


Fig. 12. Average packet retransmission delay vs. Random loss rate.



(a) Segment downloading time (b) Segment quality

Fig. 13. Average segment downloading time and quality vs. Random loss rate.

without packet dropping achieves higher segment quality than the proposed protocol without packet filtering.

Fig. 12 and Fig. 13 show the performance of average packet retransmission delay and average video segment downloading time and quality, respectively, when random packet loss occurs at each hop along the slice-level VR video transmissions between the pair of ingress and egress nodes. It can be seen that as random loss rate increases, average packet retransmission delay and average video segment downloading time increase, and average video segment quality decreases, while our proposed transmission protocol achieves the best performance compared to the benchmarks. As random loss rate increases, more BL/EL packets are lost and need retransmissions. Thus, the packet retransmission delay increases. In the mean time, retransmitted BL packets lead to increased video segment downloading time, while retransmitted EL packets degrade video quality due to deadline violations. In addition, with our proposed caching-based packet retransmission functionality/scheme, lost BL or EL packets are retransmitted by the ingress node using the cached packet copies in the caching buffer, instead of by the remote servers. Thus, a smaller packet retransmission delay is achieved. Furthermore, deadline-violated EL packets and outdated EL packets due to FoV prediction errors are discarded without being further transmitted, including the retransmitted EL packets. On the contrary, in QUIC, all lost BL and EL packets including the deadline-violated ones are retransmitted from the remote servers. Therefore, compared to QUIC, our proposed transmission protocol achieves smaller average packet retransmission delay, smaller average video segment downloading time, and better average video segment quality. Finally, for the proposed transmission protocol without the caching-based packet retransmission scheme enabled (Proposed wo. caching), lost BL/EL packets are retransmitted in a less congested environment compared to QUIC, which, resultingly, achieves better performance.

V. CONCLUSION

In this paper, we have presented a customized slice-level transmission protocol based on QUIC for tile-based VR video streaming. Diverse properties of video tiles are embedded and supported by tailoring the QUIC protocol in terms of optional header fields, stream multiplexing, and packet assembly, where explicit mapping relations between requested video tiles and QUIC streams are established. A packet filtering functionality is designed to filter out outdated (EL) video data due to FoV prediction errors, in prompt response to viewing behavior dynamics. In addition, a slice-level packet header is designed to support enhanced VR video data transmission, where a caching-based packet loss recovery scheme is proposed to achieve efficient packet retransmissions with disparate reliability requirements. Simulation results demonstrate the effectiveness of the proposed protocol. For future work, we will focus on optimizing the required caching buffer size in the caching-based packet loss recovery scheme via analytical modeling.

ACKNOWLEDGEMENT

The authors would like to thank Xu Li and Bill Gage for helpful discussions and suggestions.

REFERENCES

[1] A. Yaqoob, T. Bi, and G.-M. Muntean, “A survey on adaptive 360 video streaming: Solutions, challenges and opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[2] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, “A two-tier system for on-demand streaming of 360 degree video over dynamic networks,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 43–57, 2019.

[3] J. Song, F. Yang, W. Zhang, W. Zou, Y. Fan, and P. Di, “A fast FoV-switching DASH system based on tiling mechanism for practical omnidirectional video services,” *IEEE Transactions on Multimedia*, vol. 22, no. 9, pp. 2366–2381, 2019.

[4] S.-C. Yen, C.-L. Fan, and C.-H. Hsu, “Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol,” in *Proceedings of the 24th ACM Workshop on Packet Video*, 2019, pp. 7–12.

[5] Q. Ye, W. Shi, K. Qu, H. He, W. Zhuang, and X. Shen, “Joint RAN slicing and computation offloading for autonomous vehicular networks: A learning-assisted hierarchical approach,” *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 272–288, 2021.

[6] J. Chen, Q. Ye, W. Quan, S. Yan, P. T. Do, P. Yang, W. Zhuang, X. Shen, X. Li, and J. Rao, “SDATP: An SDN-based traffic-adaptive and service-oriented transmission protocol,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 756–770, 2019.

[7] J. Iyengar and M. Thomson, “RFC 9000 QUIC: A UDP-based multiplexed and secure transport,” *Omtermet Emgomeeromg Task Force*, 2021.

[8] J. Son, D. Jang, and E.-S. Ryu, “Implementing motion-constrained tile and viewport extraction for VR streaming,” in *Proceedings of the 28th ACM SIGMM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2018, pp. 61–66.

[9] C. Zhou, W. Wu, D. Yang, T. Huang, L. Guo, and B. Yu, “Deadline and priority-aware congestion control for delay-sensitive multimedia streaming,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4740–4744.

[10] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, “360 video viewing dataset in head-mounted virtual reality,” in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 211–216.

[11] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, “360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 315–323.

[12] S. García, J. Cabrera, and N. García, “Quality-control algorithm for adaptive streaming services over wireless channels,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 50–59, 2014.