MP-VR: An MPTCP-Based Adaptive Streaming Framework for 360-degree Virtual Reality Videos

Wenjia Wei*, Jiangping Han*, Yitao Xing†, Kaiping Xue*†§, Jianqing Liu‡, Rui Zhuang†
*The Department of EEIS, University of Science and Technology of China, Hefei 230027, China
†The School of Cybersecurity, University of Science and Technology of China, Hefei 230027, China
‡The Department of ECE, University of Alabama in Huntsville, Huntsville, AL 35899, USA
‡Corresponding Author, kpxue@ustc.edu.cn

***Abstract*—360-degree virtual reality videos greatly improve the video experience by providing users with a more immersive and interactive environment than standard streaming video. However, 360-degree videos suffer from bandwidth limits. Existing bandwidth-efficient solutions mainly focus on spatially cutting 360-degree video into tiles, and only provide video content in the Field-of-View (FoV) of users with high quality to reduce bandwidth consumption. Although existing tile-based schemes can reduce the bandwidth consumption, the bandwidth and transmission delay provided by a single-path TCP may still not meet the high requirements of 360-degree videos. Multipath TCP (MPTCP) allows a TCP connection to operate across multiple paths simultaneously and becomes highly attractive to support the mobile devices with various radio interfaces to aggregate multipath bandwidth and improve the throughput. In this paper, by taking the advantage of MPTCP, we propose an MPTCP-based adaptive streaming framework for 360-degree Virtual Reality videos, named MP-VR. MP-VR dynamically selects the appropriate tile bitrate according to the bandwidth and transmission delay of different subflows. Then it schedules the video segments to subflows to improve QoE of users. We conduct experiments on a testbed in our lab and simulations on NS-3. Evaluation results show that MP-VR outperforms existing tile-based strategies when network fluctuations or errors in FoV predictions occur.**

***Index Terms*—Multipath TCP, 360-degree videos, Tiles, Scheduling.**

I. Introduction

With the rapid development of virtual reality (VR) technology, 360-degree videos are becoming more and more popular. 360-degree videos are recorded through a panoramic camera, providing the audience with an immersive viewing experience that cannot be achieved with existing ordinary videos. More and more video providers, such as Netflix and YouTube are starting to provide 360-degree videos to users.

However, high-quality 360-degree videos still face several challenges. Among them, ensuring reliable transmission of 360-degree videos is a key issue. Compared with traditional videos, 360-degree videos usually require higher transmission bandwidth and lower delay since 360-degree videos encode omnidirectional scenes and head-mounted display (HMD) that plays 360-degree videos closer to the viewer's eyes. Therefore, 360-degree videos are usually encoded with a resolution of 4K or higher to obtain a satisfactory viewing experience. According to a Netflix recommendation [1], streaming 4K video requires at least 25Mbps, while according to another report from Akamai [2], the average broadband connection speed in the USA is only 15.3Mbps. This dilemma in terms of limited bandwidth significantly affects the quality of experience (QoE) of online 360-degree video users.

Actually, the viewer's Field-of-View (FoV) only covers a part of the entire 360-degree scene when watching a 360-degree video. Therefore, encoding and delivering the invisible area of the user at a lower bitrate level will hardly affect the visual quality of the 360-degree video on the user side. In order to reduce the use of bandwidth and ensure the user's QoE, the current 360-degree video streaming system can project spherical scenes on a 2D plane to be compatible with traditional codecs, and cut the generated frames into tiles in space and then only encode and deliver the tiles in FoV at a high bitrate.

However, the dynamic characteristic of the network make the tile-based optimal bitrate allocation scheme complicated and challenging. The reason is that the bandwidth provided by the current single-path video transmission method is limited, and it is not robust enough under complex network conditions. Therefore, in order to effectively improve the QoE of users when watching 360-degree videos, it is necessary to simultaneously increase the available transmission bandwidth and deal with high variability and unpredictable network fluctuations.

Nowadays, it is common that one communication device has more than one network interface (e.g., a mobile phone with 3G/4G and WiFi interfaces). As the available capacity in single network can not satisfy the goodput and delay demands of high-quality application, new research trends have moved towards integrating the multiple access technology to gain bandwidth aggregation, such as the work in [3], [4]. As a new bandwidth aggregation technology at the transport layer, Multipath TCP (MPTCP) [5] has obtained much attention due to the support for concurrent use of multiple interfaces and its compatibility with various applications. MPTCP aims to provide a reliable and bandwidth aggregation service while keeping fairness similar to regular TCP. MPTCP uses multiple subflows for parallel transmissions to improve throughput and robustness [6]–[8]. MPTCP has also received growing interests from industrial communities. For example, with the potential of guaranteeing reliable data transmission, Apple.inc announced to support MPTCP starting from IOS version 11 [9]. Meanwhile, the Samsung Galaxy cell phones are equipped with multiple radio interfaces and enabled simultaneous access of LTE and WiFi to boost download rate [10].

The existing MPTCP technology can well meet the requirements for the bandwidth and robustness of 360-degree videos transmission, but it does not consider the requirements of upper-layer applications when performing data scheduling, and blindly performs data scheduling for different paths. This will cause the received packets to be out of order, which will affect the end 360-degree video quality. Therefore, it is necessary to improve the 360-degree video transmission performance through a reasonable selection of the video encoding and a scheduling strategy for each video segment.

In this paper, we propose a new 360-degree videos transmission scheme based on MPTCP. Specifically, it can dynamically select the bitrate of tiles in FoV according to the transmission performance of different subflows, and at the same time, finely schedule video segments to different subflows to minimize the distortion of 360-degree videos.

The main contributions are summarized as follows:

- We first propose a new scheme to transmit 360-degree videos over MPTCP, achieving the goals of improving throughput and reducing latency. The design further aggregates the transmission performance of different links and dynamically selects the video bitrate of FoV.
- We employ a new packet scheduling scheme that is more suitable for the 360-degree videos so that the video segments can reach the receiver's play-buffer on time in unstable network environment.
- We conduct experiments on a testbed and the NS-3 simulator. The experiments show that MP-VR can effectively reduce distortion in FoV compared with existing tile-based strategies.

The remainder of this paper is structured as follows. Section II reviews the related work to this paper. In Section III, we present the system design and analytical framework. The performance evaluation and result analysis are provided in Section IV. Section V gives the concluding remarks of this work.

II. Related Work and Background

With the development of Internet technology, the traditional way of transmission becomes insufficient for bandwidth- and delay-demanding applications such as streaming at VR videos [11] [12]. Bandwidth aggregation through multiple network interfaces is a promising solution to satisfy the bandwidth and delay expectations. As an extension to TCP, MPTCP has been designed with the aim to take into account the multi-homing properties of the end-hosts. However, the legacy MPTCP only spread the data to all available paths to aggregate the bandwidth. In order to enhance the performance of MPTCP in terms of throughput and end-to-end delay, many studies have been conducted to deal with these challenges by means of tile-based streaming and MPTCP.

A. Tile-based Streaming

A 360-degree video is presented as a sphere surrounding the viewer's head, and the FoV can be described as a rectangle warped on the surface of the sphere and this part can be seen by the user. Although a user cannot view the area out of FoV, traditional schemes still transmit the entire picture when transmitting 360 videos, which obviously wastes the transmission bandwidth.

In order to reduce the bandwidth required for the transmission of 360-degree videos, a few tile-based solutions were proposed in recent years, in which increased video quality and streaming priority are assigned to tiles within FoV, while in non-FoV regions, reduced quality or priority are assigned. In tile-based solutions, a video is both temporally segmented and spatially tiled, so that each temporal segment of the video is composed of several video tiles. Wang *et al.* [13] propose some tile-based encoding and streaming solutions, including scalable coding scheme and simulcast coding scheme. In these solutions, video tiles are coded in multiple rates based on FoV. Tiles within or close to the predicted FoV are encoded and delivered at a high bitrate. Qian *et al.* [14] propose a framework based on FoV prediction, which only download the video part required by the end user to reduce the bandwidth consumption. In [15], a dynamic video tiles adaptation scheme is proposed to select tile coverage based on the FoV prediction accuracy. This scheme reduces bandwidth waste by approximately 80%. ClusTile-360 [11] can finely select the appropriate bitrate of tiles in FoV according to the storage space of the server. However, these tile-based solutions are all based on single-path transmission.

B. DASH with MPTCP

Taking advantage of multipath characteristics in DASH (Dynamic Adaptive Streaming over HTTP) has been explored in [16], [17]. Han *et al.* [16] strategically schedule video chunks' delivery to reduce the cellular data usage and radio energy consumption. Nikravesh *et al.* [17] strategically splits the file into byte range requests sent over multipath, and dynamically balances the workload across all paths. They can improve video streaming performance by utilizing multiple TCP connections simultaneously. However, these studies mainly rely on the client's bandwidth estimation or buffer status, without considering the characteristics of video content.

Our work aims at building upon these previous efforts in the area of multipath transport to improve VR video streaming. The difference is that we focuses on how to jointly consider the data content in the application layer and link parameters in the MPTCP transport layer to improve users' QoE.

III. System Design and Analytical Framework

Given the research problems in current 360-degree video transmissions, a solution using MPTCP to improve the performance of the 360-degree video transmission is proposed. With this idea, we take advantage of the bandwidth aggregation capabilities and connection robustness brought by MPTCP.

Based on the characteristics of each subflow in the MPTCP connection and the tile characteristics of the 360-degree video, the video bit rate is dynamically selected. Our scheme contains the following two contents: Bitrate selection and Packet Scheduling Mechanism.

A. Bitrate selection

Based on the previous tile-based solution, we divide the video area into multiple tiles, and select the appropriate bitrate for each tile according to the available link bandwidth and delay performance to minimize distortion in video transmission.

Fig. 1. Cut the entire area into 4*6 tiles

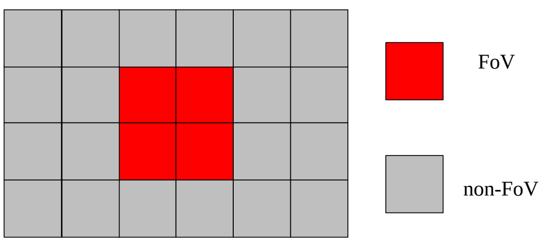


Fig. 2. Divide the area of the video into two parts, FoV and non-FoV

As shown in Fig. 1, in the process of video bitrate selection, we first cut the entire area into 4*6 tiles. Then, we divide the area of the video into two parts, FoV and non-FoV as shown in Fig. 2. The user's FoV area is predicted according to an existing prediction algorithm (i.e., linear regression based prediction, LP scheme). The bitrate of tiles in FoV area is variable, while the lowest bitrate for non-FoV in transmission. Our goal is to select an appropriate bitrate for the tiles in FoV based on the multiple link status to minimize the end-to-end distortion of the entire area.

$$\text{minimize}: \sum_{i \in FoV} D_i, \tag{1}$$

where D_i represents the end-to-end distortion of tile i, consists of encoding distortion and transmission distortion:

$$D_i = \alpha + \frac{1}{B_i - \beta} + \kappa P_{loss}, \tag{2}$$

where B_i denotes the video bitrate of tile i. $\alpha + \frac{1}{B_i - \beta}$ is the encoding distortion, which is determined by the code rate B_i selected by tile i, α β are parameters depending on specific video codec and sequence. These parameters can be estimated online by using trial encodings at the sender [18] [19]. κP_{loss} is the transmission distortion. At the same time, the sum of the code rates selected by all the tiles must be less than or equal to the sum of the maximum transmission rates T_r that can be provided by each subflow r, that is: $\sum\limits_{i \in I} B_i \leq \sum\limits_{r \in R} T_r$.

B. Packet Scheduling Mechanism

In heterogeneous scenarios, due to the uneven performance between different links, the data allocated to different subflows will not reach the receiver at the same time, causing out-of-order problems. The performance of MPTCP will be seriously degraded as the difference in subflow performance increases, especially for 360-degree videos which have high latency requirements. Therefore, it is necessary to carefully design the scheduling algorithm, reasonably allocate data to different subflows, adjust the allocation factors of different subflows, and ensure that the data at each subflow of different link characteristics reaches the receiving end at the same time. In so doing, we aim to avoid the occurrence of out-of-order problem, and reduce 360-degree videos transmission distortion.

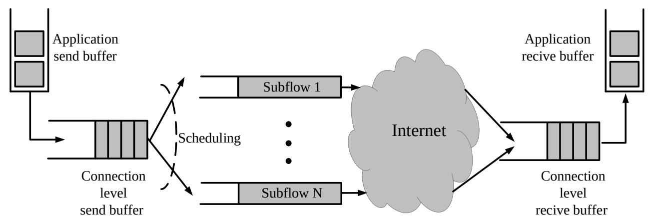


Fig. 3. Forward-prediction based scheduling to reduce latency

When transmitting video segments, packet loss on the transmission link and the timeout of the video arrival time will cause distortion of the transmitted video. Therefore, when the MPTCP layer performs data scheduling for each subflow, it is necessary to consider the packet loss rate and delay of different paths and the throughput of the subflow to minimize transmission distortion. The amount of data scheduled on the subflow r is $M\theta_r$, the transmission delay is d_r, the packet loss rate on the subflow is p_r and the delay constraint is t_d. The goodput represents the amount of data successfully received by the client within the imposed deadline t_d. Therefore, the expected delivered data after experiencing transmission losses P_{loss} through path subflow r needs to consider both link loss and timeout loss:

$$P_{loss} = 1 - \sum_{r \in R} \theta_r (1 - p_r)\{1 - p(d_r > t_d)\} \tag{3}$$

The overdue packets are considered to be dropped since they are no longer useful for the real-time applications. The probability for the dispatched packets on subflow r to be overdue can be approximated by an exponential distribution [18], i.e.,

$$p\{d_r > t_d\} \approx \frac{1}{\sqrt{2\pi}} \exp(-\frac{t_d}{d_r}), \tag{4}$$

$$d_r = \frac{B_i d_{seg} \theta_r}{T_r} + t_r, \tag{5}$$

where d_r is the estimated transmission delay, consisting of link delay and queuing delay. t_r is the link delay, representing the time it takes for the data to arrive at the receiver in the network transmission with the packet loss rate p_r. When the sender detects a loss event, it updates the loss rate p_r, calculates the average loss rate, and forecasts the time arriving at the destination. $\frac{B_i d_{seg} \theta_r}{T_r}$ is related to the packets that have not been scheduled. In order to make the data on different subflows reach the receiving buffer at the same time, the fast subflow will be scheduled for much more data, so the waiting time on the fast subflow needs to be estimated. The estimation is performed based on a subflow's T_r, and the number of assigned packets is $B_i d_{seg} \theta_r$.

We estimate throughput T_r of each subflow r based on the changing law of the congestion control window w_r under coupled congestion control [20]. Here we need to consider the current subflow window and its congestion avoidance and fast retransmission phases, as well as timeout and slow start phase. First, in the congestion avoidance and fast retransmission stages, before the next packet loss occurs, the number of rounds of successful data transmission is $E\left[cnt_r\right] = \frac{2(1-p_r)}{p_r {w_r}^2}$, then the amount of data transmitted at this stage is:

$$E\left[S^{CA}\right] = \left(\frac{3 * E[w_r]^2 + 2E\left[w_r\right]}{4}\right) \frac{E\left[cnt_r\right]}{2},$$

where $E\left[w_r\right]$ is the expected value of window size and the during time of congestion avoidance phase is:

$$E\left[X^{CA}\right] = \{\left(w_r^j - \frac{w_r^{j-1}}{2}\right) E\left[cnt_r\right] + \frac{E\left[cnt_r\right]+1}{2}\} RTT_r.$$

Then, we calculate the probability of a timeout during the timeout phase $Q_r \approx \min\left(1, \frac{3}{E[w_r]}\right)$, so the the timeout time is

$$E\left[X^{TO}\right] = RTO_r.$$

The amount of data transferred during the slow start phase is:

$$E\left[S^{SS}\right] = \sum_{i=1}^{\log_2^{ssth}+1} 2^{i-1},$$

and the during time of slow start phase is:

$$E\left[X^{SS}\right] = \left(\log_2^{ssth}+1\right) RTT_r.$$

Therefore, throughput T_r of each subflow r is:

$$T_r = \frac{E\left[S^{CA}\right] + Q_r\left(E\left[S^{SS}\right]+1\right)}{E\left[X^{CA}\right] + Q_r\left(E\left[X^{SS}\right] + E\left[X^{TO}\right]\right)} \tag{6}$$

Thus, our optimization goal is to minimize the distortion of tiles in FoV.

$$minimize: D = \sum_{i \in FoV} \left(\alpha + \frac{1}{B_i - \beta} + \kappa P_{loss}\right),$$
$$s.t. \left\{ \begin{array}{l} \sum_{r \in R} \theta_r = 1 \\ M = \sum_{i \in FoV} B_i d_{seg} \\ \sum_{i \in I} B_i \leq \sum_{r \in R} T_r \\ \max\{d_r\} \leq t_d \\ p\{d_r > t_d\} = \frac{1}{\sqrt{2\pi}} \exp(-\frac{t_d}{d_r}) \end{array} \right\}. \tag{7}$$

This is a linearly constrained non-linear optimization problem. The optimal solution derived in polynomial time may not be available. Since the code rates that can be selected in 360-degree video server is limited, we consider the fixed minimum bit rate and the highest FoV bit rate to solve the problem. At this time, the problem becomes a convex problem, that is, for an M-size video segment, how to adjust the distribution factor θ_r of each subflow so that the end-to-end distortion of the video segment is the lowest. We solve the optimal distribution factor θ_r with the water-filling algorithm.

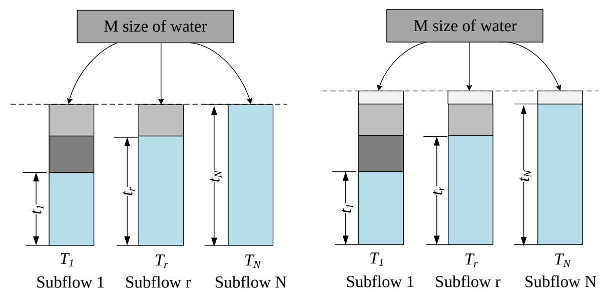


Fig. 4. Water filling solution for segment assignment

In the water-filling algorithm, M units of water will be filled into the N buckets, as shown in Fig. 4. According to the throughput and delay constraints of each subflow, the proportion of water is calculated to ensure that all subflows complete the task in equal time. Note that the physical characteristics of each path differ from each other, especially in heterogeneous wireless networks. The proposed water-filling algorithm starts by assigning the flow from the path with the lowest end-to-end delay in a progressive manner to respect the delay constraint as ruled in optimization problem. With the water-filling algorithm, we get θ_r as:

$$\theta_r = \frac{T_r}{\sum_{r' \in R} T_{r'}} \{1 - \frac{\sum_{r' \in R} T_{r'}(t_r - t'_r)}{M}\}. \tag{8}$$

IV. Performance Analysis

We evaluate MP-VR by comparing its performance with tile-based streaming methods BAS-360 [12] and MPTCP (on-demand). BAS-360 can finely select the appropriate bitrate of tiles in FoV and transmit through a single-path. MPTCP (on-demand) refers to the use of MPTCP to transmit video segments in the FoV area on demand. Both BAS-360 and MPTCP (on-demand) will choose the same bitrate for all involved tiles without exceeding the predicted bandwidth budget. In the evaluation, we actively control the network conditions to observe how different schemes react to the network fluctuations.

A. Performance Evaluation in Testbed

We first build a multipath VR video test platform as shown in Fig. 5. It consists of a multipath VR video server and a multipath VR video client. The VR video test platform is based on the DASH architecture. There are two paths for transmitting VR video data between the video server and the video client. Based on this platform, we compare our MP-VR algorithm with BAS-360.

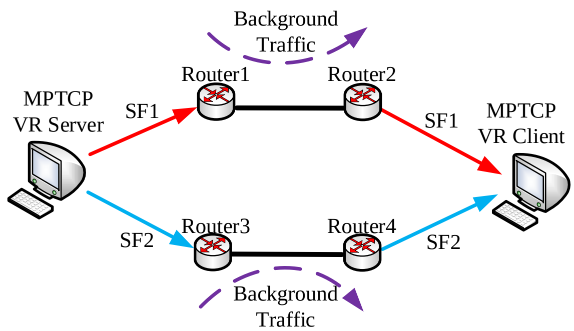


Fig. 5. The multipath VR video test platform

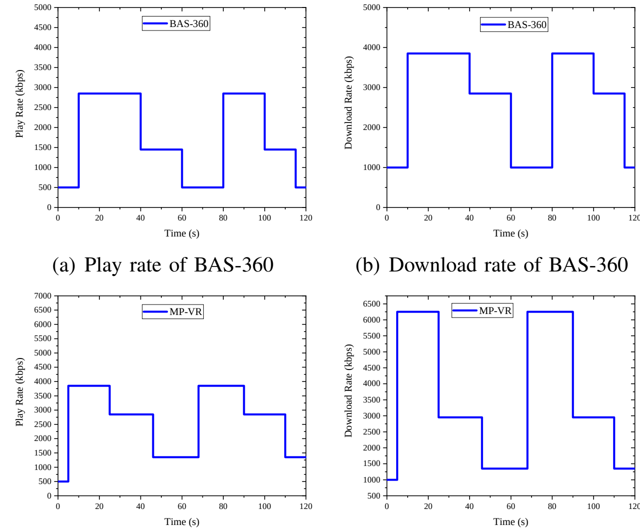
The video file is first split spatially into tiles (4*6) by ffmpeg, and the tiles are further segmented temporally by the MP4Box tool. We use public datasets to give FoV data (http://dash.ipv6.enstb.fr/headMovements/) when users watch the video Rollercoaster. We use tc as the network shaping tool that imposes shaping rules on IP packets that have 8080 as either dst port or src port. Such a setting is to eliminate the impact of other uncertainties so that we can get a more accurate assessment of our scheduling method. The link parameters and the bitrates of tiles in the experiment are as follows:

- The bandwidth of link changes according to 6150 kbps, 2850 kbps, 1450 kbps, 6150 kbps, 2850 kbps, 1450 kbps.
- The entire area is cut into 4*6 tiles, the bitrates of each tile are encoded into four code rates: 400 kbps, 200 kbps, 100 kbps, 40 kbps.

Fig. 6 shows the played video quality and the downloaded video quality of BAS-360 and MP-VR. The x-axis is the playback time in seconds and the y-axis is quality summation of the played tiles and the downloaded tiles, indicating the proportion of bandwidth allocated to deliver the actually viewed content. We can observe that our scheme can fully exploit the bandwidth resources to improve visual quality, whose average downloaded quality summation is 3000 kbps and the average played quality summation is 2545 kbps. Both of the metrics are higher than BAS-360.

B. Performance Evaluation in NS-3

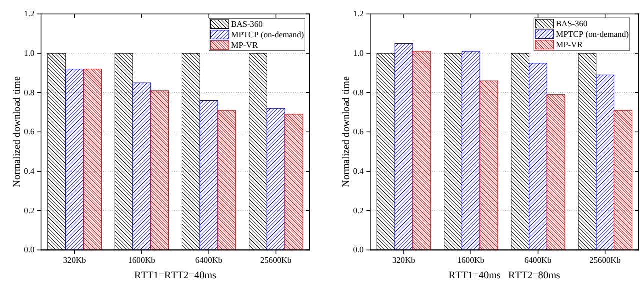
We then evaluate the proposed MP-VR with BAS-360 and MPTCP (on-demand) in a simulation environment that is built



(a) Play rate of BAS-360
(b) Download rate of BAS-360
(c) Play rate of MP-VR
(d) Download rate of MP-VR

Fig. 6. Video quality variations with preset bandwidth information

based on the Kheirkhah et al. [21] NS-3 open source MPTCP implementation of the IETF RFC 6824. The NS3 simulation topology is set to reflect the previous real network testbed. Here we will analyze the impact of delay, bandwidth, and packet loss on video transmission.



(a) The bandwidth of two subflow are all 6150kbps and the RTT of two subflows are all 40 ms
(b) The bandwidth of two subflow are all 6150kbps while the RTT of two subflows are 40 ms and 80 ms

Fig. 7. Normalized download time in symmetric and asymmetric link

We first analyze the completion time of different schemes (set the performance of BAS-360 as baseline) when the size of video segments changes. Fig. 7 shows that when the size of video segments is small, the performance improvement brought by MPTCP (on-demand) and MP-VR is not large compared to BAS-360, but this advantage becomes more and more obvious as the size of video segments increases. At the same time, MP-VR has the highest performance. When the RTTs of the subflows are different, MPTCP (on-demand) has no performance improvement over BAS-360, and the performance becomes worse. This is because the video segmentation is very small, and an inappropriate multipath data scheduling algorithm may deteriorate the overall transmission performance. MP-VR still performs very well, because it can more accurately allocate data according to the subflow's link parameters.

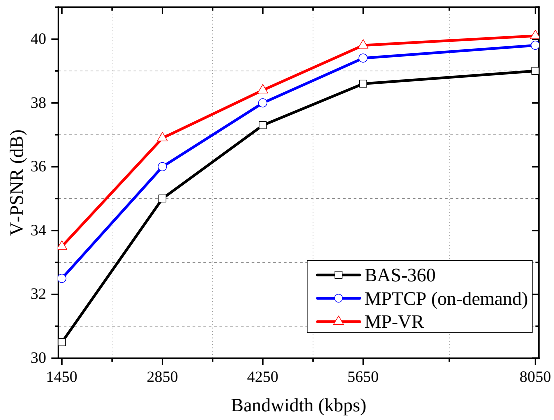


Fig. 8. The PSNR in FoV region with varying bandwith and the RTT of two subflows are all 40 ms

In Fig. 8, the PSNR in FoV area with varying bandwidth of the three algorithms are shown. It can be observed that PSNR in FoV significantly increases as the transmission bandwidth increases. Compared with BAS-360, MPTCP (on-demand) aggregates the bandwidth of multiple paths and improves the video quality. MP-VR can further improve the video quality through bitrate selection and fine data scheduling to the available paths.

In order to further evaluate the performance of our scheduling algorithm in unstable network environment, we add dynamic background traffic to each path, set the on-off time of the network background traffic to match the Pareto mode, and add a random packet loss rate of 2% on each subflow. Then we observe the video segment loss rate in the FoV area. The experimental results show that the video segment loss rate of BAS-360, MPTCP (on-demand) and MP-VR in FoV area are 5.81%, 3.52% and 3.05%, respectively. We can see that the multipath transmission scheme can provide better robustness as MPTCP (on-demand) and MP-VR reduce the segment loss rate. MP-VR can solve the strict low-latency requirement of view switching points through reasonable scheduling, and further improve the performance.

V. Conclusion

Since traditional single-path transmission cannot meet the high bandwidth and low latency requirements of current 360-degree videos, we propose an MPTCP-based adaptive streaming framework for 360-degree virtual reality video in this paper. Our design can dynamically select the bitrate of tiles in FoV according to the parameters of each subflow, and further schedule the video segments to different subflows so that they can reach the receiver's play-buffer on time. The simulation results and testbed tests demonstrate that the proposed scheme offers better performance than single-path video transmission schemes.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grants No. 61972371 and No. U19B2033, and Youth Innovation Promotion Association of CAS under Grant No. 2016394.

References

[1] "Internet connection speed recommendations," https://help.netflix.com/en/node/306.

[2] "Akamai's [state of the internet] Q1 2016 report," https://www.akamai.com/uk/en/multimedia/documents/state-of-the-internet/akamai-state-ofthe-internet-report-q1-2016.pdf.

[3] X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1787–1801, 2017.

[4] B. Jin, S. Kim, D. Yun *et al.*, "Aggregating LTE and Wi-Fi: Toward intra-cell fairness and high TCP performance," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6295–6308, 2017.

[5] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," *IETF RFC 6824*, 2013.

[6] B. Han, F. Qian, S. Hao, and L. Ji, "An anatomy of mobile web performance over multipath TCP," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. ACM, 2015, p. 5.

[7] X. Corbillon, R. Aparicio-Pardo, N. Kuhn *et al.*, "Cross-layer scheduler for video streaming over MPTCP," in *Proceedings of the 7th International Conference on Multimedia Systems (MMsys)*. ACM, 2016, p. 7.

[8] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens *et al.*, "A measurement-based study of multipath tcp performance over wireless networks," in *Proceedings of the 2013 conference on Internet Measurement Conference (IMC)*. ACM, 2013, pp. 455–468.

[9] "Use multipath TCP to create backup connections for iOS," https://support.apple.com/en-us/HT201373.

[10] Q. D. Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "Observing real smartphone applications over multipath TCP," *IEEE Communications Magazine*, vol. 54, no. 3, pp. 88–93, 2016.

[11] C. Zhou, M. Xiao, and Y. Liu, "ClusTile: Toward minimizing bandwidth in 360-degree video streaming," in *Proceedings of the 37th IEEE International Conference on Computer Communications (Infocom)*. IEEE, 2018, pp. 962–970.

[12] M. Xiao, C. Zhou, V. Swaminathan *et al.*, "BAS-360: Exploring spatial and temporal adaptability in 360-degree videos over http/2," in *Proceedings of the 37th IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2018, pp. 953–961.

[13] Y. Wang, M. Karczewicz *et al.*, "Tile based VR video encoding and decoding schemes," 2016, jCTVC Doc. X0077.

[14] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges (AllThingsCellular)*. ACM, 2016, pp. 1–6.

[15] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "HEVC-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM International Conference on Multimedia (MM)*. ACM, 2016, pp. 601–605.

[16] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan, "MP-DASH: Adaptive video streaming over preference-aware multipath," in *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies (CoNEXT)*. ACM, 2016, p. 129–143.

[17] A. Nikravesh, Y. Guo, X. Zhu *et al.*, "MP-H2: A client-only multipath solution for HTTP/2," in *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2019, pp. 1–16.

[18] X. Zhu, P. Agrawal, J. P. Singh *et al.*, "Distributed rate allocation policies for multihomed video streaming over heterogeneous access networks," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 752–764, 2009.

[19] J. Wu, B. Cheng, M. Wang, and J. Chen, "Energy-aware concurrent multipath transfer for real-time video streaming over heterogeneous wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 2007–2023, 2018.

[20] Q. Liu, K. Xu, H. Wang, and L. Xu, "Modeling Multi-Path TCP throughput with coupled congestion control and flow control," in *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. ACM, 2015, p. 99–106.

[21] M. Kheirkhah, I. Wakeman, and G. Parisis, "Multipath-TCP in ns-3," in *The 2014 Workshop on ns-3 (WNS3)*, 2014.