Song CX, Han B, Su JS. 4D-MAP: Multipath adaptive packet scheduling for live streaming over QUIC. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(1): 159-176 Jan. 2024. DOI: 10.1007/s11390-023-3204-z

# 4D-MAP: Multipath Adaptive Packet Scheduling for Live Streaming over QUIC

Cong-Xi Song (宋丛溪), Biao Han* (韩　彪), *Member, CCF*, and Jin-Shu Su* (苏金树), *Fellow, CCF*

*College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China*

E-mail: songcongxi17@nudt.edu.cn; nudtbill@nudt.edu.cn; sjs@nudt.edu.cn

Received March 6, 2023; accepted December 1, 2023.

**Abstract** In recent years, live streaming has become a popular application, which uses TCP as its primary transport protocol. Quick UDP Internet Connections (QUIC) protocol opens up new opportunities for live streaming. However, how to leverage QUIC to transmit live videos has not been studied yet. This paper first investigates the achievable quality of experience (QoE) of streaming live videos over TCP, QUIC, and their multipath extensions Multipath TCP (MPTCP) and Multipath QUIC (MPQUIC). We observe that MPQUIC achieves the best performance with bandwidth aggregation and transmission reliability. However, network fluctuations may cause heterogeneous paths, high path loss, and bandwidth degradation, resulting in significant QoE deterioration. Motivated by the above observations, we investigate the multipath packet scheduling problem in live streaming and design 4D-MAP, a multipath adaptive packet scheduling scheme over QUIC. Specifically, a linear upper confidence bound (LinUCB)-based online learning algorithm, along with four novel scheduling mechanisms, i.e., Dispatch, Duplicate, Discard, and Decompensate, is proposed to conquer the above problems. 4D-MAP has been evaluated in both controlled emulation and real-world networks to make comparison with the state-of-the-art multipath transmission schemes. Experimental results reveal that 4D-MAP outperforms others in terms of improving the QoE of live streaming.

**Keywords** multipath transmission, live streaming, Quick UDP Internet Connections (QUIC), quality of experience (QoE), packet scheduling

## 1 Introduction

With the rapid development of applications such as TikTok and YouTube, live streaming has become a major part of many people's daily life. According to Cisco's Annual Internet Report[①], the amount of traffic is increasing faster as a result of the expansion of video applications. As of 2021, 76.6% of respondents use Real-Time Messaging Protocol (RTMP) for live streaming[②], which typically builds upon TCP as the transport protocol. However, TCP-based live streaming not only requires fine-tuning of TCP in the operating system (OS) kernel but also exhibits a high handshake latency. Recently, Google's Quick UDP Internet Connections (QUIC) protocol[③] has provided a handshake latency of less than one round-trip time (RTT). As a cross-layer protocol, QUIC overcomes the OS-level support obstacles in tuning TCP and is able to combine transport the protocol and video applications to achieve high performance and cost-efficiency at the same time. The emergence of the QUIC

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant No. 62102430, the Hunan Young Talents under Grant No. 2020RC3027, the Natural Science Foundation of Hunan Province of China under Grant No. 2021JJ40688, the Training Program for Excellent Young Innovators of Changsha under Grant No. kq2206001, and the Science Research Plan Program by National University of Defense Technology under Grant No. ZK22-50.

*Corresponding Author (Biao Han guided the research, especially the solution design, and paper writing. Jin-Shu Su initiated and coordinated the work.)

① https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, Nov. 2022.

② https://www.wowza.com/blog/2021-video-streaming-latency-report, Jun. 2022.

③ https://www.rfc-editor.org/info/rfc9000, Nov. 2022.

protocol brings new opportunities for high quality live streaming applications.

Many broadcasters are pursuing high-definition live streaming with 4K and higher resolutions. For example, YouTube Live offers 4K live streaming at 60 frames per second (fps), which suggests a bitrate of 20 Mbps–51 Mbps④. Moreover, more and more applications are requiring bitrates greater than 100 Mbps, such as augment reality (AR) and virtual reality (VR). The ultra-high-definition live videos will even exceed the capabilities of 5G networks[1]. The trend motivates academia and industry to explore new transmission solutions for better quality of experience (QoE) and higher reliability. Multipath transmission technologies enable multi-homed devices to utilize multiple network interfaces such as WiFi, Long Term Evolution (LTE), and 5G for aggregating bandwidth resources and providing path backup. Inspired by Multipath TCP (MPTCP)⑤, Multipath QUIC (MPQUIC)[2] was proposed as an extension of QUIC for multipath transmission. Currently, MPQUIC is used to improve the performance and reliability of video transmission by aggregating the bandwidth of multiple paths[3], while its application in live streaming is still in its infancy.

In order to investigate the performance of applying multipath transmission for live streaming, we first implement two live streaming frameworks over RTMP protocols based on MPTCP and MPQUIC. Based on them, we conduct preliminary experiments to explore the existing challenges when applying them to live streaming. We observe that: 1) MPTCP-based live streaming has a higher start-up latency than MPQUIC; 2) due to network fluctuations, the quality of multiple paths may become heterogeneous, which results in degrading the QoE of live streaming; 3) high path loss deteriorates the QoE of live streaming significantly; 4) insufficient bandwidth cannot accommodate the burst video bitrate. However, the current multipath packet scheduling schemes cannot well address the above challenges by adopting appropriate scheduling under network fluctuations.

The above observations motivate us to develop 4D-MAP, a multipath adaptive packet scheduling scheme for live streaming based on MPQUIC[2]. In 4D-MAP, we design a linear upper confidence bound (LinUCB)[4] based online learning algorithm to make multipath adaptive packet scheduling decisions among our proposed four novel scheduling mechanisms, i.e., Dispatch, Duplicate, Discard, and Decompensate. These 4D-scheduling mechanisms work in an adaptive and cooperative manner. Specifically, the Dispatch mechanism pre-allocates the corresponding size of data into paths, which aims at mitigating out-of-order packet receptions resulting from path heterogeneity. The Duplicate, Discard, and Decompensate mechanisms are enabled when the online learning detects high path loss and bandwidth degradation. In our large-scale experiments, we evaluate the live streaming performance of 4D-MAP in both controlled emulation and real-world networks. The results show that compared with current multipath transmission schemes like MPTCP, MPQUIC, Peekaboo, and DQN*, where Peekaboo and DQN* are two learning-based multipath packet scheduling schemes for handling dynamic network conditions[5, 6], 4D-MAP enhances the QoE of live streaming by reducing re-buffering time, stream delay, and start-up latency while maintaining good visual fidelity. The main contributions of this paper are summarized as follows.

1) *Key Observations.* We conduct preliminary experiments to reveal the limitations of the existing multipath transmission solutions. We observe the benefits of live streaming based on MPQUIC and the QoE reductions from out-of-order packet receptions, high path loss, and insufficient bandwidth, which result from network fluctuations.

2) *4D-MAP Design.* We propose 4D-MAP, a multipath adaptive packet scheduling scheme. It uses a LinUCB-based online learning algorithm to make adaptive scheduling decisions for our four proposed mechanisms.

3) *4D-MAP Implementation.* We implement our proposed 4D-MAP based on the MPQUIC prototype[2]. The source code of our 4D-MAP implementation is available online⑥.

4) *Performance Evaluation.* Experimental results in controlled emulation and real-world networks reveal that 4D-MAP reduces re-buffering time, stream delay, and start-up latency than four state-of-the-art multipath transmission solutions, i.e., MPTCP, MPQUIC, Peekaboo, and DQN*.

④https://restream.io/blog/what-is-a-good-upload-speed-for-streaming/, Jun. 2022.
⑤https://www.rfc-editor.org/info/rfc8684, Nov. 2022.
⑥https://github.com/cxht/4D-MAP, Jan. 2023.

## 2 Background

### 2.1 Live Streaming Process

Live streaming can be transmitted through various streaming protocols. Since over 76.6% of respondents in 2021 use Real-Time Messaging Protocol (RTMP) to perform live streaming, we use RTMP as our application-layer streaming protocol in this work. A QUIC connection is established between the broadcaster and the subscriber. Once the connection has been established, the end-to-end live streaming process can be conducted using RTMP over MPQUIC. Fig.1 depicts this procedure, which consists of the following steps. 1) The broadcaster captures live streaming data using a camera on a mobile device. 2) The data is compressed and encoded into I-frames and P-frames⑦. Group of pictures (GOP) refers to a sequence of consecutive video frames between two I-frames. 3) The video frames are packed into RTMP messages. 4) The RTMP packets are written to QUIC streams at the transport layer. 5) The MPQUIC protocol reads the data from the QUIC streams and packs it into MPQUIC packets' payload. 6) The MPQUIC packets are delivered through multiple paths. 7) The packets travel over the Internet via multiple paths to cloud servers for processing. 8) Subscribers can watch the live stream by using a player to pull the stream.

The performance of the first-mile transmission from the broadcaster's mobile device to the cloud server is crucial especially when broadcasting large events and performing real-time search and rescue. Therefore, our main objective in this work is to propose an enhanced multipath transmission scheme that guarantees fluidity and visual fidelity of live streaming during the first-mile transmission.

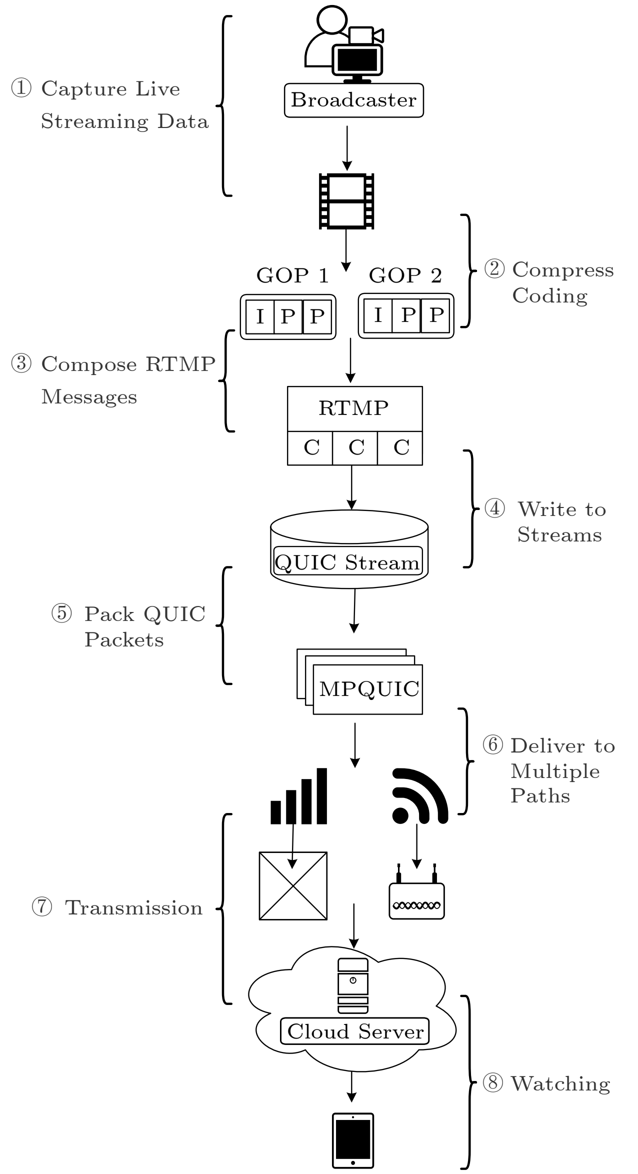


Fig.1. Live streaming transmission procedure via RTMP over MPQUIC. I: I-frame; P: P-frame; C: chunk in an RTMP message.

### 2.2 Related Work

MPTCP⑧ is a standardized multipath extension of traditional TCP implemented by the Internet Engineering Task Force (IETF), which enables multi-home devices to establish multiple paths for simultaneous data transmission. MPQUIC[2, 8] can leverage and establish multiple paths based on the QUIC protocol, which provides multi-stream support, stream-level multiplexing, and encryption to improve security and prevent middlebox interference. Recently, multipath extensions of QUIC have received extensive attentions of IETF QUIC Working Group⑨.

A number of studies have tried to design novel multipath packet schedulers at the transport layer. BLEST[9], ECF[10], and STMS[11] are implemented in MPTCP, and estimate subflow's RTT and band-

⑦I-frames refer to intra-coded pictures which can be decoded as complete images and P-frames refer to predicted pictures based on previous I- or P-frames with video codecs such as H.264[7].

⑧https://multipath-tcp.org/, Jun. 2022.

⑨https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/02/, Nov. 2022.

width to fully utilize faster paths and reduce out-of-order packets delivery. OLS[12] transmits redundant packets to reduce out-of-order packets. Both ReLes[6] and Peekaboo[5] utilize learning algorithms to detect fluctuations, allowing the transmission to adapt to dynamic network conditions. ReLes adopts the Deep Q-Network (DQN) to schedule packets in MPTCP while Peekaboo utilizes a LinUCB-based online learning algorithm to adapt to the current network condition. Although the above studies are capable of handling dynamic network fluctuations, they are not oriented to video transmission, which is content-agnostic.

In MP-DASH[13], video streaming is emphasized with an awareness of the user's preferred network interface. XLINK[3] proposes QoE-driven scheduling and path management mechanisms based on the MPQUIC protocol for short videos. [14] formulates video transmission as an optimization problem in which QoE objectives are obtained by solving the problem. VOXEL[15] incorporates application feedback information with partially reliable transmission[16] to optimize video streaming. VR and AR challenges are addressed in [17] by scheduling higher priority data on the lowest-latency path.

The primary objective of our work is to enhance streaming performance during the first-mile of the transmission process. Additionally, several studies use the Adaptive Bitrate Scheme (ABR) to improve QoE during the last-mile of the transmission between the Content Delivery Network (CDN) and subscribers. There are ABRs specifically designed for live streaming. TCLiVi[18] uses a DRL-based algorithm to adjust the streaming parameters to improve the QoE of live streaming. [19] develops model predictive control and DRL-based frameworks to maximize live streaming QoE by adapting the bitrate. MFVP[20] improves the accuracy and reduces the overhead of viewport prediction in 360-degree video streaming. MultiLive[21] is designed for multi-party live streaming which models the many-to-many ABR selection problem. Our work is also orthogonal to these studies.

### 2.3 Challenges of Live Streaming

Live streaming transmission has specific challenges as follows. 1) Live streaming is a delay-sensitive application. Overdue frames will result in re-buffering of live streaming. 2) Live streaming has inter-frame dependency. P-frames are created based on I-frames or other P-frames, which results in inter-dependency between frames. 3) Live streaming has specific QoE requirements. Because live videos are produced and streamed in real time, the start-up latency and stream delay should be considered.

## 3 Observations and Motivations

To reveal the problems of streaming live videos through RTMP over (MP)TCP and (MP)QUIC, we perform a series of preliminary experiments. We conduct all preliminary experiments in a controlled emulated network environment implemented by Mininet[22]. To mimic LTE and WiFi links, we conduct measurements of link characteristics within a college laboratory by Iperf3⑩ and the Ping tool to determine the one-way delay (OWD) and bandwidth values for emulation. We employ the Linux Traffic Control (TC) tool to set up two paths with measured bandwidths of 14.8 Mbps and 18 Mbps, and OWDs of 25 ms and 39 ms. This configuration is used as the baseline configuration for our preliminary experiments. We use the default scheduler in both MPTCP and MPQUIC[2, 23], namely minRTT, as our multi-path packet scheduling algorithm, which delivers packets on the path with minimum round-trip time (minRTT) until the path's congestion window (CWND) is filled. OLIA[24] is used as the congestion control algorithm. We emulate the live streaming process by streaming a video from a broadcaster to a subscriber through LTE and WiFi paths as depicted in Fig.1. The video video-hy (details are in Table 1) is selected for streaming with an average bitrate of 30.55 Mbps and 60 s playback duration. To ensure fairness, each experiment is repeated 10 times with the same path configuration and video. Preliminary experiments are conducted in a laptop with MPTCP version 0.95 and MPQUIC⑪ based on `quic-go`⑫.

### 3.1 Limitations of RTMP over (MP)TCP

Although TCP's re-transmission, congestion control mechanisms ensure its reliability, its high hand-

⑩http://dast.nlanr.net/Projects/Iperf/, Oct. 2022.
⑪https://multipath-quic.org/, Oct. 2023.
⑫https://github.com/lucas-clemente/quic-go, Oct. 2023.

**Table 1.** Description of Video Set

| Video Set | AvgBR. (Mbps) | MaxBR. (Mbps) | Dur. (s) |
|---|---|---|---|
| Video-bbb[13] | 2.78 | 6.54 | 60 |
| Video-sintel[14] | 9.42 | 20.71 | 60 |
| Video-ss[15] | 22.33 | 34.41 | 60 |
| Video-hy[15] | 30.55 | 73.34 | 60 |
| Video-sintel-10min | 10.20 | 26.72 | 600 |
| Video-ss-10min | 20.46 | 56.04 | 600 |

Note: AvgBR., MaxBR., and Dur. are average bitrate, maximum bitrate, and video duration, respectively.

shake latency cannot be ignored. As the multipath extension of TCP, MPTCP inherits the limitation although it can provide more bandwidth for live streaming. In order to reveal the negative effects resulting from the high handshake latency of (MP)TCP, we compare the start-up latency between (MP)TCP and (MP)QUIC. Start-up latency can be calculated by measuring the duration of the first video frame sent from the broadcaster to the subscriber, which is a crucial QoE metric used to determine whether the subscriber will continue to view this streaming.

*Observation* 1. Live streaming through RTMP over (MP)TCP exhibits longer start-up latency and is cumbrous to be customized with different QoE requirements.

*Motivation* 1. Exploiting QUIC and its multipath extension (MPQUIC) to deliver live streaming can achieve lower start-up latency and more customized QoE preferences.

Fig.2(a) demonstrates the average start-up latency of four transport protocols. When we switch from TCP to QUIC, the average start-up latency decreases from 0.19 s to 0.11 s. As a result of a reduction in average start-up latency, there is an impact on QoE.

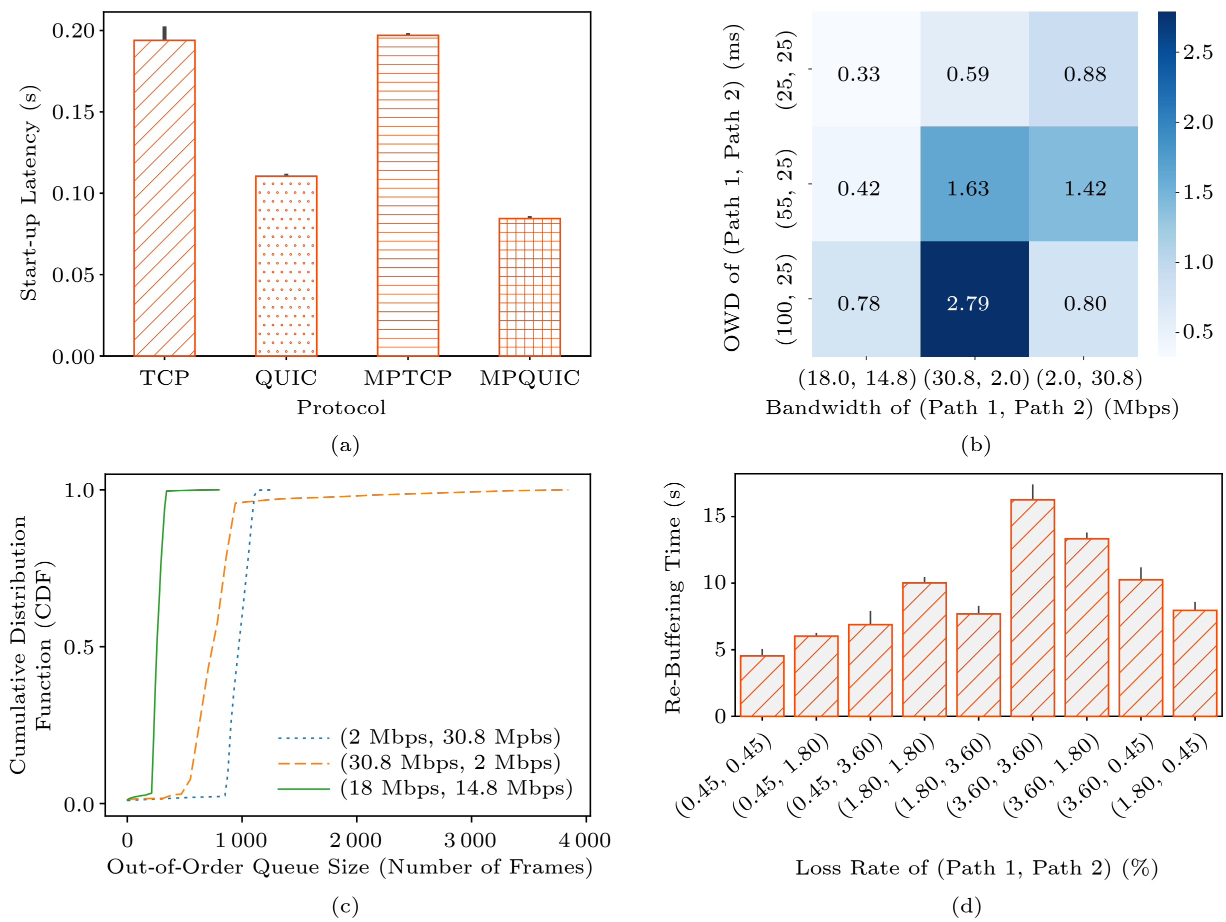


Fig.2. Preliminary experimental results. (a) Start-up latency. (b) Re-buffering time with heterogeneous paths. (c) Out-of-order queue size. (d) Re-buffering time with path loss.

[13]https://peach.blender.org/, Jun. 2022.
[14]https://durian.blender.org/, Jun. 2022.
[15]www.bilibili.com, Dec. 2022.

The reason is that TCP connections require at least one RTT to establish before transmitting any application data, while TLS adds two additional RTTs. When multiple paths are introduced, MPTCP has a longer average start-up latency compared with TCP, as each path establishment in MPTCP requires a three-way handshake. On the other hand, MPQUIC provides a lower average start-up latency than MPTCP. The reason is that MPQUIC utilizes a new path directly and conveys data in the first packet of the new path. Therefore, for reducing start-up latency and accelerating the loading of the first image of live streaming, RTMP over MPQUIC is a better choice than (MP)TCP.

In addition, another limitation in that (MP)TCP stack is implemented within OS kernels. Any change to this stack requires upgrading the entire kernel[25]. The inflexibility limits the protocol stack to meet various QoE requirements for live streaming. Since (MP)QUIC is implemented in the user space, the transport layer stack can be more flexible to be upgraded with novel mechanisms, thus meeting varied requirements for live streaming.

### 3.2 Path Heterogeneity Degrades Live Streaming QoE

In addition to the characteristics of asymmetry wireless access, such as LTE and WiFi, network fluctuations caused by user movement, frequent handoffs, and changes to topology may result in path heterogeneity. Although QUIC supports multiple streams to prevent severe Head-of-Line (HoL) blocking, MPQUIC that introduces multiple paths will result in out-of-order packet receptions, which may further deteriorate live streaming performance. We perform experiments and analyze the relationship among path heterogeneity, out-of-order receptions, and QoE of live streaming.

In order to mimic delay and bandwidth heterogeneity, OWD and bandwidth are adjusted according to the baseline path configuration. To perform delay heterogeneity, path 1 acts as a "slow path" with an OWD increasing from 25 ms to 100 ms, while path 2 acts as a "fast path" with a constant OWD of 25 ms. In the next step, we configure the bandwidth of two paths, which are set to (18 Mbps, 14.8 Mbps), (30.8 Mbps, 2 Mbps), and (2 Mbps, 30.8 Mbps), respectively. We select these bandwidth settings to provide three types of heterogeneity. A homogeneous bandwidth configuration (18 Mbps, 14.8 Mbps) is the first. In the second configuration (30.8 Mbps, 2 Mbps), the slow path has a higher bandwidth than the fast path. The third configuration (2 Mbps, 30.8 Mbps) provides the fast path with higher bandwidth. Therefore, the fast path outperforms the slow path in terms of path quality. Re-buffering time is used for evaluating the fluidity of live streaming which refers to the time spent in re-buffering in the total video playback duration.

*Observation* 2. In cases of path heterogeneity, MPQUIC's default packet scheduler results in out-of-order packet receptions, further degrading the QoE of live streaming.

*Motivation* 2. A novel multipath packet scheduler is required to mitigate the out-of-order packet reception problem in multipath live streaming.

As shown in Fig.2(b), when the bandwidth of both paths is nearly homogeneous, the average re-buffering time increases due to delay heterogeneity as the delay of the slow path increases from 25 ms to 100 ms. Although the total bandwidth is identical, average re-buffering time increases due to the bandwidth heterogeneity. When bandwidths of two paths are (2 Mbps, 30.8 Mbps), and the OWDs of the paths are (100 ms, 25 ms), the average re-buffering time is 0.8 s. The minRTT scheduler always choose to utilize only one path (path 2) rather than multiple paths because path 1 presents a complete disadvantage in this situation. In the worst case, two paths have OWDs of (100 ms, 25 ms) and bandwidths of (30.8 Mbps, 2 Mbps). Multipath transmission begins with path 2 being selected based on minRTT, resulting in frequent congestion due to its lower bandwidth. In this case, path 1 is used instead of path 2. As a result of congestion, RTTs of both paths are increased, which further prolongs the transmission time.

To identify issues with out-of-order packet receptions during multipath transmission, we measure the out-of-order queue size by calculating the amount of QUIC stream frames that are buffered by the receiver and not deliverable to the application within 0.1 s. We select the three most heterogeneous path configurations (the bottom line configuration of Fig.2(b)) where OWDs are consistent with (100 ms, 25 ms), but bandwidths differ for comparison. As shown in Fig.2(c), when bandwidths are homogeneous at (18 Mbps, 14.8 Mbps), the out-of-order queue size remains at 250 frames at the 60th percentile value. However, when bandwidths become heterogeneous at (2 Mbps, 30.8 Mbps), the out-of-order queue size increases to 1 000 frames at the 60th percentile value. Additionally, when bandwidths are (30.8 Mbps, 2 Mbps), the out-of-order queue size reaches approximately 4 000 frames. This large number of out-of-order frames blocked in the receiving buffer results in prolonged re-buffering time. Consequently, MPQUIC's default packet scheduler, minRTT performs poorly with heterogeneous paths, leading to out-of-order packet receptions and degrading live streaming quality. Thus, a novel multipath packet scheduler is necessary.

### 3.3 High Path Loss Deteriorates Live Streaming QoE

As reported in [26], the loss rate has a higher effect on generating re-buffering events than OWD in LTE and WiFi environments. However, considering poor network conditions with high path loss, the performance of multipath live streaming is still under investigation. Therefore, we compare the average re-buffering time in different path loss rate environments.

*Observation* 3. High path loss leads to severe re-buffering time deterioration due to reliable transmission of MPQUIC.

*Motivation* 3. The multipath live streaming scheme should conduct adaptive packet scheduling in poor network conditions with high path loss.

Based on the baseline path configuration, we adjust the loss rate of both paths ranging from 0.45% to 3.6% to calculate the average re-buffering time. Fig.2(d) depicts the average re-buffering time in different loss rate combinations. When one of the paths experiences a low loss rate of 0.45%, the average re-buffering time stays lower than 10 s. It benefits from multipath transmission reliability which provides alternative paths for transmission. However, when the path loss rate becomes higher, the average re-buffering time increases in a cliff-like manner. The average re-buffering time reaches 16.26 s in the worst case, i.e., 3.6% for each path. The deterioration of re-buffering time is due to the fact that MPQUIC inherits the reliable transmission characteristic of QUIC. A reliable transmission involves re-transmitting every lost packet, resulting in a long transmission time. Therefore, if the network condition becomes poor with a high path loss rate, multipath packet scheduling of MPQUIC should make adaptive reactions to the network condition.

### 3.4 Burst Bitrate Challenges Multipath Bandwidth Aggregation

The real-time bitrate of a video will fluctuate when some drastic images occur. These drastic images refer to video frames that contain sudden changes in visual content, such as scenes with a lot of details or fast-moving actions. These types of images generate more data to be transmitted than other frames in the video, which may cause the real-time bitrate to exceed the aggregated bandwidth, thus degrading video playback quality as the network struggles to keep up with the demands of transmitting the necessary data.

*Observation* 4. Insufficient bandwidth cannot accommodate burst video bitrate.

*Motivation* 4. The multipath live streaming scheme should take into account the case when the bandwidth is insufficient to support video bitrate surges.

A set of videos with their real-time bitrate is shown in Fig.3. WiFi and LTE's measured average bandwidths are marked as two dashed straight lines, and the expected aggregation bandwidth is the solid straight line. For example, a few points of the real-time bitrate of video-ss (average bitrate of 22.33 Mbps, maximum bitrate of 34.41 Mbps) slightly surpass the aggregated bandwidth resources after 45 s, which results in re-buffering. In this case, traditional ABR will tune down the bitrate to provide low-quality video, which results in the significant clarity loss in a video segment. While the multipath technology can provide bandwidth resources by establishing a new path, timely establishment is difficult and not worthy when burst video bitrate occurs temporarily. Therefore, the multipath live streaming scheme should be

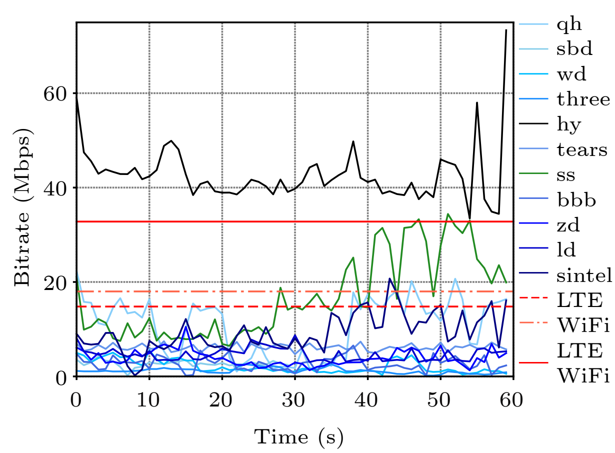


Fig.3. Real-time bitrate of selected videos.

able to adapt to temporary insufficient bandwidth scenarios in real time.

## 4 Proposed 4D-MAP

### 4.1 System Overview

In order to handle heterogeneous paths resulted from network fluctuations for improving user-perceived QoE of live streaming, we propose a multipath adaptive packet scheduling framework called 4D-MAP. As shown in Fig.4, 4D-MAP is based on the architecture of RTMP over MPQUIC, which can establish multiple paths via various interfaces, e.g., WiFi, LTE, and 5G. A lightweight online learning algorithm based on LinUCB learns current network conditions and video characteristics with the information from cross-layer sensing (Subsection 4.2), and then guides our proposed 4D-scheduling on how to adjust scheduling mechanisms (Section 5).

The majority of the 4D-MAP components are deployed at the broadcaster-side during live streaming. When a video frame is stored in the send buffer, the video analyzer extracts the video characteristics from the application layer. Meanwhile, the network monitor runs in the background to collect current network conditions. With video characteristics and network conditions, the LinUCB based online learning algorithm determines and instructs 4D-scheduling to select mechanisms. Video frames are packed and delivered to corresponding paths in the path pool according to the mechanism. At the server side, the live videos are streamed to the subscribers. Moreover, the feedback for online learning's reward computing is provided to the broadcaster. The packets lost by Discard and Decompensate will also be compensated on the server.

### 4.2 Cross-Layer Sensing

Cross-layer sensing is implemented at the transport layer. It collects information from the network layer by network monitoring. Additionally, its video analyzer analyzes video characteristics from the application layer. In accordance with the current network conditions and video characteristics, an online learning model can be trained and then adopt appropriate scheduling decisions.

#### *4.2.1 Network Monitor*

MPQUIC protocol stack's native mechanisms

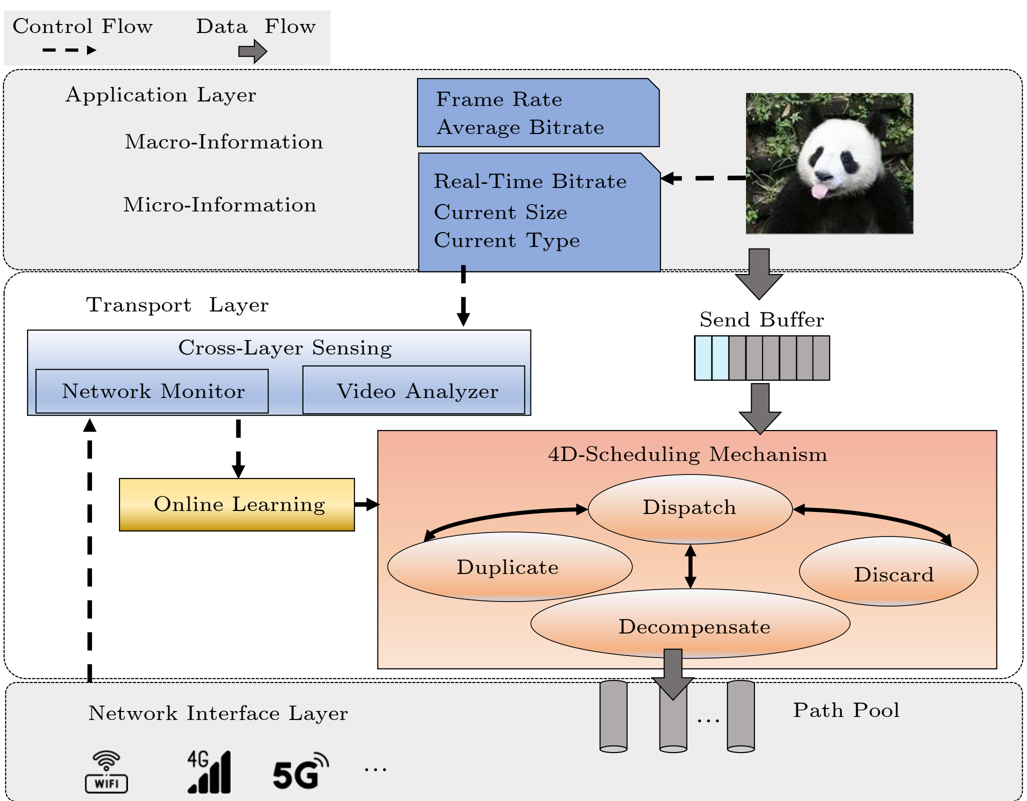


Fig.4. Overview of 4D-MAP.

maintain the statuses of each established path, through which the network monitor continuously collects network conditions. Path statuses are shown below.

1) *Round-Trip Time* ($RTT$). RTT is determined by the time gap between packets sending and receiving acknowledgment (ACK).

2) *Congestion Window* ($CWND$) *and Inflight Bytes* ($IB$). The congestion control mechanism maintains these two statuses.

3) *Packet Loss Rate* ($L$). $L$ is estimated by the total number of packets divided by the number of lost packets in a given period.

### 4.2.2 Video Analyzer

The video analyzer is the interface between the application layer and the transport layer to obtain video characteristics. Two types of characteristics can be utilized, namely macro-information and micro-information. Macro-information is determined at the beginning of live streaming. Micro-information is related to the current frame. The details of how to obtain them are shown as follows.

1) *Frame Rate* ($FR$) *and Average Bitrate* ($AvgBR$). They are macro-information and derived from FFMpeg[16].

2) *Frame Size* ($FS_i$) *and Frame Type* ($FT_i$) *of the Current Video Frame* $i$. They are micro-information and can be obtained from the video frame's header.

3) *Real-Time Bitrate* ($RBR_t$). It is micro-information and is the total size of recent $FR$ video frames at time $t$. It can be calculated as $\sum_{i=0}^{FR} FS_{i^*-i}$, where $i^*$ is the current video frame and $FS_{i^*-i}$ is the size of frame $i^* - i$.

Based on the video characteristics frame type ($FT$), we can prioritize the importance of different video frames. When network fluctuation occurs, more significant data should be transmitted on time. An I-frame is regarded as a high priority frame as it can be decoded as a complete figure without referring to the other frames. P-frames, originating from I-frames, are regarded as of low priority.

## 4.3 Online Learning with the LinUCB-Based Algorithm

We use a LinUCB-based online learning algorithm[4] to learn scheduling decisions that are adaptive to current conditions. The mechanism selection can be modeled as a Multi-Armed Bandit (MAB) problem with context information[27]. According to current vectors of states which represent network conditions and video characteristics (i.e., the context), the agent selects an action from the available action set. The agent's objective is to maximize a reward utility function. Thus, we begin by defining the state, action, and reward of our problem.

*State*. Cross-layer sensing provides items in the state vector. The agent makes the decision when the data of a new frame is written into the send buffer at time $t$. The state is represented by $\boldsymbol{S}_t = (\boldsymbol{s}_{t,1}, \ldots, \boldsymbol{s}_{t,i}, \ldots, \boldsymbol{s}_{t,N}, \boldsymbol{vf}_t)$, where $i$ refers to path $i$, and $N$ is the total number of paths. $\boldsymbol{s}_{t,i} = (RTT_{t,i}, CWND_{t,i}, IB_{t,i}, L_{t,i})$, $\boldsymbol{vf}_t = (FR, RBR_t, FS_t, FT_t)$. $FS_t$ and $FT_t$ are the video frame size and frame type of the newest generated video frame at time $t$ respectively. Different from each path $i$ having its own network status metrics $\boldsymbol{s}_{t,i}$, only one group of video characteristics metrics $\boldsymbol{vf}_t$ is in the state at time $t$.

*Action*. 4D-scheduling maintains a fundamental scheduling mechanism Dispatch and three additional mechanisms Duplicate, Discard, and Decompensate. Hence, online learning should detect the appropriate time to enable these mechanisms to adapt to the current network condition. The action set includes: normal, duplicate, discard, and decompensate, which refer to stay in the normal Dispatch mechanism, enable the Duplicate, Discard, or Decompensate mechanism, respectively.

*Reward*. Since 4D-MAP aims at improving the QoE of live streaming under dynamic network conditions, we compute the reward with the consideration of both fluidity and visual fidelity. Since the agent makes decisions when a video frame is written to the send buffer, we should monitor the transmission performance of this video frame to identify the influence of our selected action. We evaluate the influence of this action by the transmission duration $FD_i$, i.e., the duration of transmission from the broadcaster to the server. As transmission duration is related to frame size, we divide $FS_i$ by $FD_i$ when computing rewards as $R = FS_i/FD_i$.

According to the overdue setting in [28], we assume that a video frame is overdue after 200 ms, which can be regarded as a deadline. All packets that

[16]http://ffmpeg.org/, Jun. 2022.

convey the same video are assumed with the same deadline. To penalize the visual fidelity loss from the Discard and Decompensate mechanisms' dropping packets (details are in Section 5), we add a penalty item $P_{vf}$ to fix the transmission duration for a video frame $vf$ that contains dropped packets by (1).

$$P_{vf} = \left(1 - \frac{FS_{vf}^{\text{recvBytes}}}{FS_{vf}}\right) \times deadline, \quad (1)$$

where $FS_{vf}^{\text{recvBytes}}$ is the actual received bytes of video frame $vf$. $P_{vf}$ is 0 when $FS_{vf}^{\text{recvBytes}} = FS_{vf}$, which means frame $vf$ is a complete frame and no penalty is required. In order to take into account the long-term impact of actions, we calculate the reward by considering not only the current video frame $vf^*$, but also the subsequent three frames from $vf^*+1$ to $vf^*+3$. The reward function for the current video frame $vf^*$ is computed as (2):

$$R^{'} = \sum_{vf=vf^*}^{vf^*+3} \frac{FS_{vf}}{FD_{vf}+P_{vf}} \times w(FT_{vf}), \quad (2)$$

where $w(FT_{vf})$ is the weight of different frame types. According to preliminary analysis, an I-frame has a higher weight of 1, while a P-frame has a lower weight of 0.7. After receiving a video frame $vf$, the server can calculate $FD_{vf}$, which feedbacks the broadcaster through ACKs. As soon as the broadcaster receives feedback information, it calculates the reward.

To solve this contextual MAB problem, we use the LinUCB algorithm[4] to balance exploration and exploitation. Additionally, its disjoint linear model reduces the high computational overhead associated with ridge regression. A detailed description of the LinUCB algorithm can be found in Appendix⑰.

We first pre-learn a model offline based on historical states, actions, and rewards. The model warms up the online learning process at the beginning of transmission by pre-learning. The amount of training data for pre-learning is 10 network traces whose network configurations are generated within the range shown in Table 2. Using the derived model, action selections are performed during video transmission. The decision is made when the data of a new video frame is written into the send buffer. Meanwhile, the online learning process continuously reads states and computes rewards based on feedback. The model is periodically updated in order to accommodate current conditions.

**Table 2.** Range of Path Heterogeneity in Terms of Bandwidth, One-Way Delay (OWD), and Loss Rate

| Path | Bandwidth (Mbps) | OWD (ms) | Loss Rate (%) |
|---|---|---|---|
| 1 | 2–25 | 20–100 | 0.01–0.70 |
| 2 | 2–30 | 30–100 | 0.05–1.00 |

## 5 Proposed 4D-Scheduling Mechanisms

We design 4D-scheduling which contains four novel scheduling mechanisms aiming at improving QoE under dynamic heterogeneous network conditions. The Dispatch mechanism mitigates out-of-order packet receptions caused by heterogeneous paths, which is the default scheduling mechanism in 4D-MAP. Other three mechanisms Duplicate, Discard, and Decompensate are enabled as enhancements for Dispatch by the online learning model.

### 5.1 Dispatch Mechanism

In order to mitigate out-of-order packets receptions, we need to determine the appropriate segments of data to be sent along specific paths so that these segments arrive in order. We assume that $S$ bytes of video frames generated by the application layer need to be scheduled at time $t$. $N$ paths are established to deliver data and their RTT, congestion window, and inflight bytes are $(RTT_1, \ldots, RTT_N)$, $(CWND_1, \ldots, CWND_N)$, and $(IB_1, \ldots, IB_N)$, respectively, which are collected by network monitoring. Initially, we sort established paths according to RTT of all paths. Then, we need to calculate in advance how much data $n_i$ needs to be pre-allocated to path $i$, with statuses of path $i$ and path $i+1$ as (3)–(5):

$$n_{\text{gap}} = \frac{RTT_{i+1}}{RTT_i} \times CWND_i, \quad (3)$$

$$n_i = \begin{cases} \min(S, n_{\text{gap}} - IB_i), & \text{if } \min(S, n_{\text{gap}} - IB_i) > 0, \\ 0, & \text{otherwise}, \end{cases} \quad (4)$$

$$S = S - n_i, \quad (5)$$

where $n_{\text{gap}}$ refers to the gap size which resides for the faster path. By reserving data of size $n_{\text{gap}}$ for the faster path, the slower path can transmit data directly that follows behind. The faster path can also trans-

⑰https://github.com/cxht/4D-MAP/Appendix.pdf, Jun. 2023.

mit consecutive data after a few congestion events, ensuring that both paths can transmit data in order as much as possible. Our mechanism takes account of the inflight data, and more accurate RTT estimation from QUIC's ACK feedback. We build a path buffer for each path $i$, which stores the corresponding size of pre-allocated data as (4). Since pre-allocation data is stored in the path buffer for each path, each path will be iterated to pack packets from the buffer continuously, ensuring that all paths can be utilized as soon as they are available.

### 5.2 Duplicate Mechanism

The loss rate of path $i$ is represented by $L_i$. When we duplicate a packet over multiple paths, its chance of reaching the receiver is higher than over only one path with $1-\prod_{i=0}^{n} L_i$. Therefore, we can mitigate the negative effects of high loss rate network conditions by generating redundancy. Packets that convey high priority data will be duplicated and distributed to alternate paths when using Duplicate.

When Duplicate is enabled, the path with the lowest loss rate is chosen as the primary path, while all the other paths with available CWND are set up as alternate paths, which carry both the normal and redundant packets. Due to the reliable transmission in the QUIC protocol, redundant packets will aggravate the overhead of re-transmission. If the original and redundant packets are both lost, double re-transmissions will waste bandwidth resources. To conserve bandwidth resources, we only duplicate those packets which convey high priority data. We set those redundant packets as "semi-reliable", which do not have to be re-transmitted when they are lost again. With Duplicate, it is possible to ensure the timely delivery of important video frames and maintain high priority data arrival as much as possible.

### 5.3 Discard Mechanism

Videos containing fast-moving actions or high levels of details may result in a real-time bitrate that surpasses the available bandwidth. This can cause rebuffering issues. In order to improve the live streaming fluidity without compromising QoE in visual fidelity too much, we design the Discard mechanism. The Discard mechanism can reduce the volume of data that is not crucial, and bandwidth resources can be conserved for more crucial data.

According to the priority of each video frame, when online learning enables Discard, just the header of P-frames is transmitted, and the rest are dropped. A header is resident because it contains the size, timestamp, and other crucial information that is needed by application layers to divide RTMP messages into chunks. Data with high priority, on the other hand, is transmitted without being dropped.

### 5.4 Decompensate Mechanism

When the current path loss rate is high enough, and the aggregated bandwidth is not sufficient to convey current video content (high real-time bitrate), the Duplicate mechanism will consume additional bandwidth for generating redundancy which aggravates the bandwidth shortage. To address this case, we develop another mechanism Decompensate.

Decompensate modifies the re-transmission mechanism with the consideration of data priority. When packing video data into packets, we mark all QUIC packets with high priority payloads and RTMP message headers with the reliable tag, and all the others are marked with the unreliable tag. A lost unreliable QUIC packet will not trigger retransmission.

When packets are lost or arrive out-of-order in a QUIC stream, gaps in the receive buffer prevent the application layer from reading data. The stream delay is prolonged by waiting for these gaps, which deteriorates the QoE of time-sensitive live streaming applications. There are some gaps created by the Discard and Decompensate mechanisms as they drop or do not re-transmit lost packets. We implement a gap padding mechanism on the server. Gap padding is activated after a deadline, determined by the longest OWD of all paths. If the waiting time exceeds the deadline, we zero pad those gaps. Since these gaps are low priority video frame data, the frame will only have a few mosaic points during playback. Using this mechanism, a trade-off between visual fidelity and fluidity is made in order to reduce re-buffering by sacrificing a little visual fidelity.

## 6 Implementation

The 4D-MAP system is implemented based on `quic-go` and its multipath extension[2] and an RTMP library[18]. Our live streaming framework is built, and the underlay transport protocol is modified to

[18] https://github.com/q191201771/lal, Apr. 2022.

MPQUIC. Thus, RTMP connections can be established by using MPQUIC connections.

We implement interfaces which provide access to information at both the application and network layers. With the interface `GetChunkInfo()`, broadcasters can get the timestamp, size, and frame type of video frames. We can derive current network conditions via interfaces `GetRTT()`, `GetInflight()`, `GetCWND()`, and `GetLossrate()` in the send packet handler. For reward computing, we add a 64-bit field in the QUIC's ACK frame that consists of the last received video frame timestamp and frame duration.

## 7 Performance Evaluation

We evaluate the performance of 4D-MAP in both controlled emulation and real-world networks. Apart from the experiments for evaluating the entire 4D-MAP framework, separate experiments are adopted for measuring the validity of each proposed mechanism. Each experiment is repeated 20 times with the same path configuration and video for fairness. A detailed overhead analysis can be found in Appendix[19].

### 7.1 Experimental Setup

#### 7.1.1 *Video Set*

Two widely used videos from prior work, namely Big Buck Bunny (video-bbb)[20] and Sintel (video-sintel)[21], are applied. We also download two videos for evaluation from a video website Bilibili[22] for live streaming. We call them video-ss and video-hy, respectively. Table 1 illustrates the details of these videos. Apart from 60 s duration, we use a 600 s version video-sintel-10min and video-ss-10min to perform adaptability experiments.

#### 7.1.2 *Network Testbed*

We set up two testbeds in both controlled environment and real-world network. As shown in Fig.5, in both testbeds, MPTCP and MPQUIC are installed to be compared with 4D-MAP. We establish two paths in both testbeds.

*Controlled Emulation*. The baseline experimental setting is the same as in Section 3. Bandwidths for

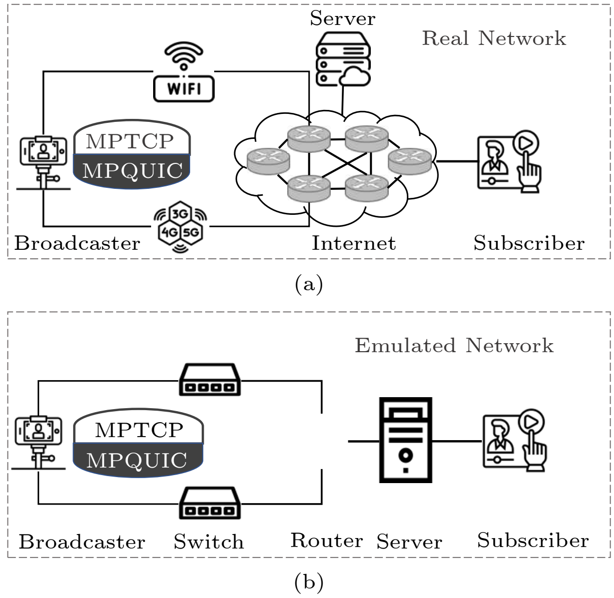


Fig.5. Network topologies. (a) Real-world network. (b) Controlled emulation network.

the two paths are 14.8 Mbps and 18 Mbps, respectively, and OWDs for them are 25 ms and 39 ms, respectively. Depending on different evaluation objectives, we adjust the configuration accordingly. All controlled emulations are implemented in Mininet[22].

*Real Network*. In the real-world experiments, we deploy our broadcaster in a college laboratory environment. We use the same Linux laptop with WiFi and LTE network interfaces. We measure their link characteristics by the Iperf3 and Ping tool. The LTE and WiFi interfaces are connected to the Internet service providers (ISP) of China Telecom and China Mobile, respectively. An Alibaba cloud server is used to receive and forward the stream pushed from the broadcaster. The subscriber is another laptop which pulls the live streaming video from the server.

#### 7.1.3 *QoE Metrics*

We evaluate the following QoE metrics in the performance evaluation. 1) Re-buffering time refers to time spent in re-buffering in the total video playback duration. 2) Start-up latency can be calculated by measuring the duration of the first video frame sent from the broadcaster to the subscriber. 3) Stream delay approximates the average duration between the broadcaster sending a video frame and the frame be-

[19]https://github.com/cxht/4D-MAP/Appendix.pdf, Jun. 2023.
[20]https://peach.blender.org/, Jun. 2022.
[21]https://durian.blender.org/, Jun. 2022.
[22]www.bilibili.com, Dec. 2022.

ing received by the subscriber. 4) Adjusted SSIM (aSSIM) is an extension of structural similarity index (SSIM)[29]. Based on SSIM, [16] computes aSSIM to take both fluidity and visual fidelity into consideration by scoring wherein each frame period (i.e., frame rate $FR$) over the duration of the stall is assigned an SSIM index of zero. The method of computing aSSIM is shown as:

$$aSSIM = \frac{1}{(N + FR \times Dur_{\text{rb}})} \times \sum_{i=1}^{N} SSIM_i,$$

where $Dur_{\text{rb}}$ means re-buffering time, and $N$ means the number of frames in the video.

## 7.2 Emulation Results

### 7.2.1 *Efficiency of 4D-Scheduling*

We first evaluate the performance of Dispatch. We select minRTT, ECF[10], and STMS[11] as the representative algorithms of the schedulers. We utilize video-hy for streaming. As shown in Fig.6(a), when two paths' bandwidths are 18 Mbps and 14.8 Mbps, respectively, ECF and STMS perform better than minRTT. However, ECF and STMS exhibit a higher re-buffering time when the bandwidth becomes more heterogeneous than minRTT. In contrast, Dispatch reduces the average re-buffering time in three bandwidth heterogeneity scenarios compared with other schedulers. In the most heterogeneous scenario, where OWDs of path 1 and path 2 are (100 ms, 25 ms) and bandwidths are (30.8 Mbps, 2 Mbps), Dispatch outperforms minRTT, ECF, and STMS by 5.7%, 19.4%, and 26.0% respectively. The reason is that our proposed Dispatch mechanism pre-allocates data based on collected path statuses to mitigate out-of-order packet receptions and fully utilizes two paths, therefore reducing the average re-buffering time.

Next, we evaluate the improvement from Duplicate. In addition to 4D-MAP without the Duplicate mechanism (4D-MAP w/o Duplicate), we also implement another scheduling mechanism that redundantly sends packets over all paths (RDDT)[30] in MPQUIC for comparison. The loss rates of both paths are configured as (1.8%, 1.8%), (1.8%, 3.6%), and (3.6%, 3.6%). We utilize video-bbb for streaming. As shown in Fig.6(b), both RDDT and the Duplicate mechanism can benefit from the reliability by redundancy generation in high loss rate network conditions. Especially, when the loss rates of two paths are at both 3.6%, 4D-MAP with Duplicate (4D-MAP/Duplicate) reduces the average re-buffering time by almost 5.5% more than that without Duplicate. As shown in Fig.7(a), compared with RDDT, the Duplicate mechanism produces less redundant data due to its semi-

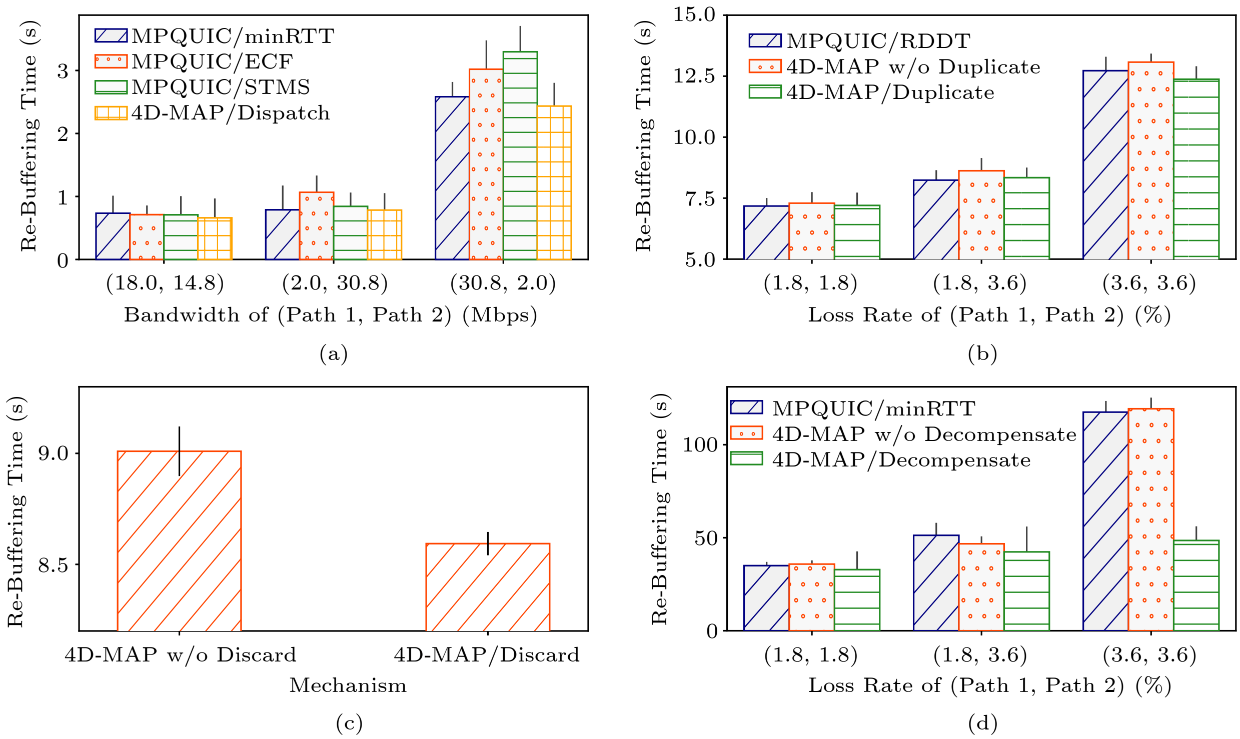


Fig.6. Experimental results of re-buffering time. (a) Dispatch. (b) Duplicate. (c) Discard. (d) Decompensate.

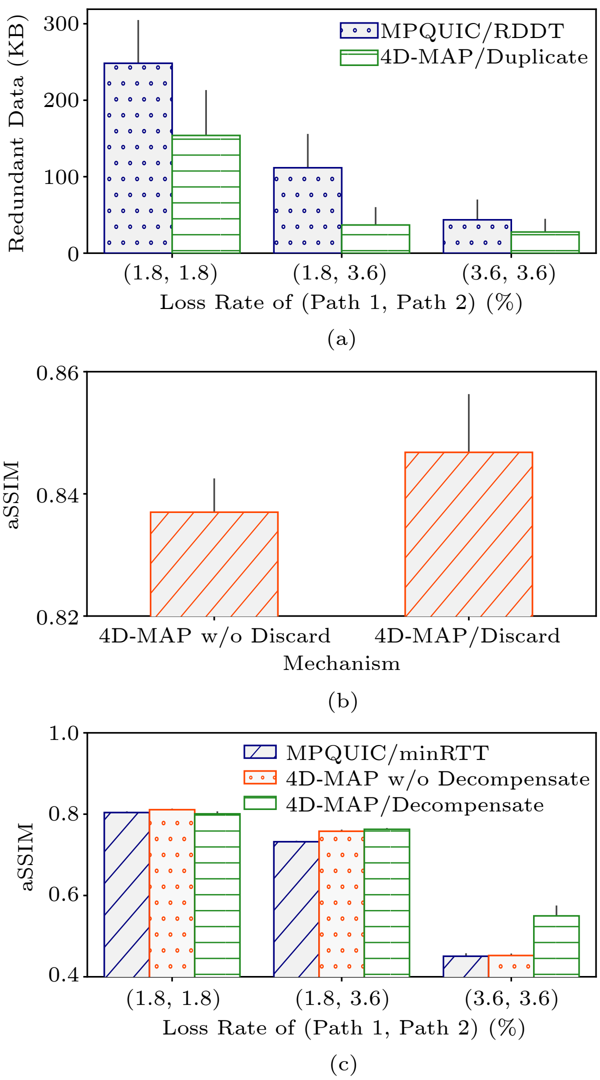


Fig.7. (a) Redundant bytes of RDDT and Duplicate. (b) aSSIM of Discard or without Discard. (c) aSSIM of Decompensate, without Decompensate or MPQUIC.

reliable policy.

We compare 4D-MAP with and without the Discard mechanism. We select video-ss as the video. As shown in Fig.6(c), 4D-MAP with the Discard mechanism brings 4.8% reduction in the average re-buffering time. Moreover, the achievable average aSSIM is even higher than that without Discard as illustrated in Fig.7(b). The results prove that it is worth dropping some low priority packets to exchange fluidity.

Finally, we compare the fluidity improvement from the Decompensate mechanism. We select video-sintel as the video. As shown in Fig.6(d), 4D-MAP with the Decompensate mechanism performs better than the others. Especially when loss rates of two paths are both at 3.6%, 4D-MAP with Decompensate reduces the average re-buffering time by up to 58.7% and 59.3% than MPQUIC with minRTT and Decompensate respectively. Fig.7(c) depicts that Decompensate offers superior or similar performance in the average aSSIM compared with the other two scheduling mechanisms. It has been proven that Decompensate's unreliable transmission policy improves QoE by reducing unnecessary re-transmissions.

#### 7.2.2 *Adaptability Experiments*

4D-MAP is evaluated for its adaptability to dynamic and heterogeneous networks. Based on the baseline configurations of both paths, we change the bandwidth, OWD, and loss rate of each path every 30 s in 600 s. Within the scope of Table 2, all configurations are selected based on uniform random sampling using the WSP algorithm[31]. In the adaptability experiments, we evaluate two videos. One is video-sintel-10min with an average bitrate of 10.20 Mbps, and the other is video-ss-10min with an average bitrate of 20.46 Mbps. For comparison, MPTCP and MPQUIC are both used. Because minRTT performs robustly in most scenarios, we use it as the scheduler in both MPTCP and MPQUIC. In addition, we compare our 4D-MAP with an intelligent online learning based scheduler Peekaboo[5] which was proposed to handle dynamic network environments. In addition, we include another offline training-based scheduler ReLes for comparison[6] which uses DQN to generate policy. However, ReLes was designed for MPTCP and its source code is not open. We utilize another DQN-based multipath scheduling framework[32], combined with the approach from [6] to reproduce ReLes in MPQUIC. We call this implementation as DQN*. We select OLIA as the congestion control algorithm in all experiments.

As shown in Fig.8(a), when streaming video-sintel-10min, the average re-buffering time of 4D-MAP is 97.0%, 18.1%, 37.6%, and 34.5% lower than that of MPTCP, MPQUIC, Peekaboo, and DQN*, respectively. Furthermore, the average stream delay of 4D-MAP in Fig.8(b) is 88.5%, 1.2%, 12.7%, and 11.9% lower than that of MPTCP, MPQUIC, Peekaboo, and DQN*, respectively, when streaming video-ss-10min. 4D-MAP also outperforms the other frameworks in these two metrics when streaming video-ss-10min. From Fig.8(c), we find 4D-MAP performs lower average start-up latency in video-sintel-10min than other four frameworks. The aSSIM of selected frameworks is compared by evaluating visual fidelity. Fig.8(d) de-

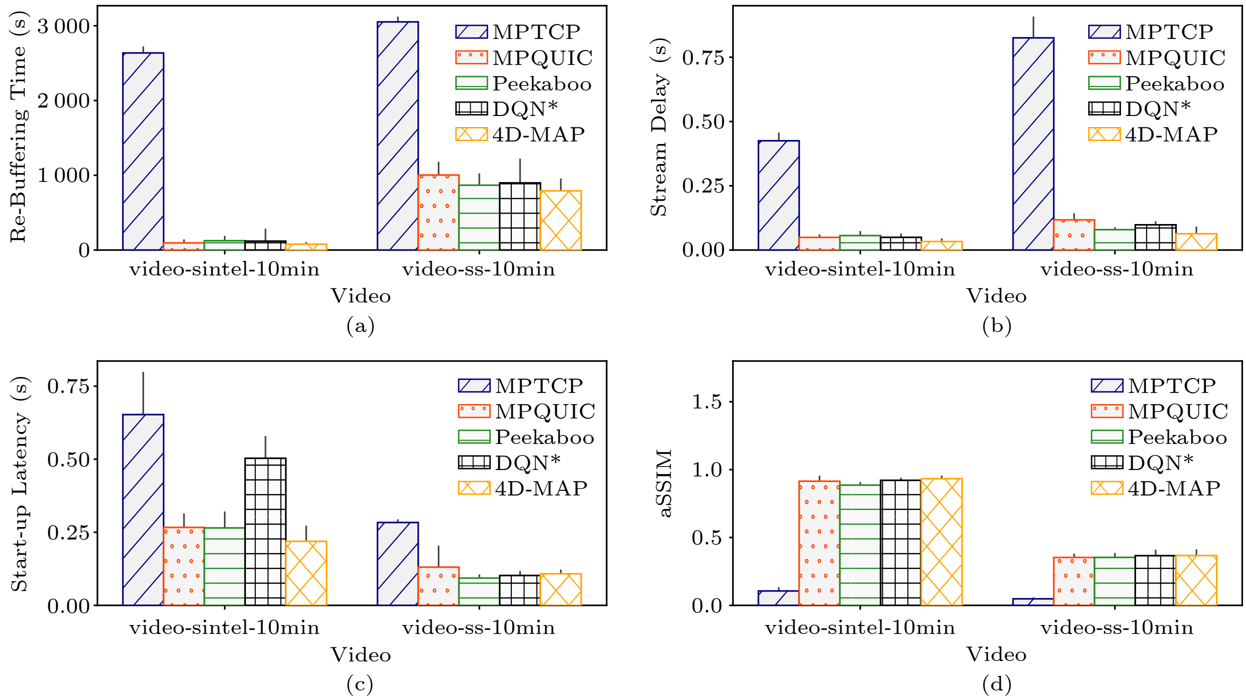


Fig.8. Adaptability experimental results. (a) Re-buffering time. (b) Stream delay. (c) Start-up latency. (d) aSSIM.

picts that MPTCP's average aSSIM lags far behind the others. 4D-MAP achieves superior or similar to others when streaming both video slices.

According to the results, 4D-MAP outperforms MPTCP and MPQUIC in fluidity QoE metrics without a large loss in visual fidelity, validating 4D-MAP's capability to provide smoother live streaming in dynamic network conditions. Furthermore, 4D-MAP performs better than Peekaboo and DQN*. The reason is that their policies do not take into account the details of video content, and thus their strategies cannot well suit to video transmission scenarios.

### 7.3 Real-World Experimental Results

According to the real network setup, we establish LTE and WiFi paths, respectively. The Ping tool is used to measure RTT for LTE and WiFi paths in real network experiments, with average values of 50 ms and 78 ms, respectively. Based on Iperf3 measurements, their bandwidths are 14.8 Mbps and 18 Mbps. video-sintel-10min and video-ss-10min are also selected for comparison. Our evaluations include re-buffering time, stream delay, start-up latency, and aSSIM.

As shown in Fig.9(a), in comparison with MPTCP, MPQUIC, Peekaboo, and DQN*, 4D-MAP results in lower average re-buffering time by 97.6%, 74.1%, 95.0%, and 73.8% when streaming video-sintel-10min, and 89.0%, 26.8%, 62.5%, and 85.0% when streaming video-ss-10min. Moreover, it reaches lower average stream delay than MPTCP, MPQUIC, Peekaboo and DQN* up to 48.1%, 40.3%, 29.1%, and 18.6% when streaming video-sintel-10min, respectively, and 29.9%, 4.4%, 49.0%, and 44.5% when streaming video-ss-10min, respectively, which is illustrated in Fig.9(b). In addition, as shown in Fig.9(c), 4D-MAP achieves lower average start-up latency than the other frameworks by 47.7%, 41.1%, 18.5%, and 27.1%, respectively, when streaming video-sintel-10min, and 66.1%, 44.7%, 37.3%, and 47.0%, respectively, when streaming video-ss-10min. In Fig.9(d), 4D-MAP achieves higher average aSSIM than the others when streaming both video slices.

Experimental results prove that the 4D-MAP framework can adapt to real network environments, which provides smoother live streaming without sacrificing too much visual fidelity.

### 7.4 Limitations and Future Work

*Prompt Monitoring of Network Status.* In real-world network environments, network status fluctuates drastically, which may result in untimely network monitoring. At some extreme scenarios, only relying on RTT and loss estimation by ACK frames is not enough. It is therefore necessary to design or find an algorithm that is agile and accurate for network monitoring.

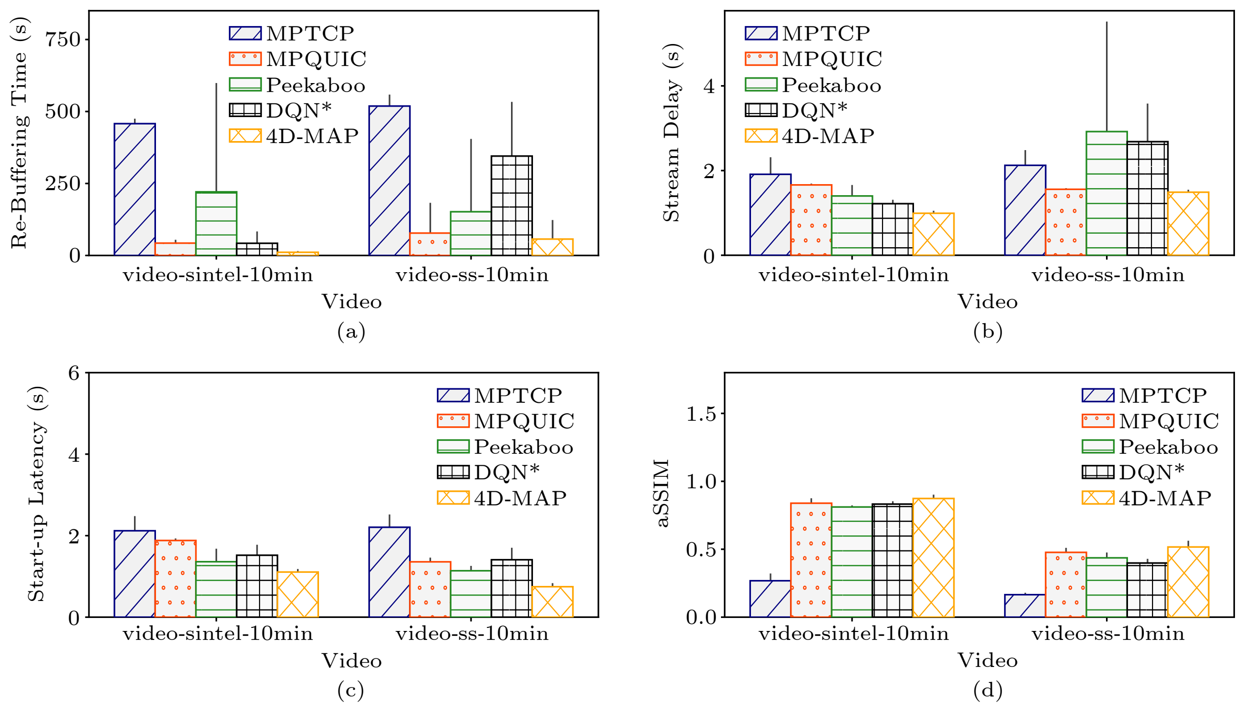


Fig.9. Real-world experimental results. (a) Re-buffering time. (b) Stream delay. (c) Start-up latency. (d) aSSIM.

*Understanding the Importance of Video Content.* While our 4D mechanisms improve video fluidity, visual fidelity is unavoidably reduced. The reductions are generated primarily by the Discard and the Decompensate mechanisms. As long as we know which frame will generate a relative lower reduction, we can adopt an unreliable policy on packets that convey those video frames.

*One-to-Many Live Streaming.* When it comes to one-to-many live streaming, efficiently delivering streams to multiple subscribers is crucial. Our 4D-MAP can be deployed between the server and the subscriber to facilitate real-time streaming. In addition, by combining our strategy with SVC (Support Vector Machine), we can schedule the base layer or enhancement layer to different paths, ensuring the high-quality video with multipath resources.

## 8 Conclusions

In order to improve the QoE for live streaming under dynamic heterogeneous network conditions, we presented 4D-MAP, a multipath adaptive packet scheduling solution which is built upon RTMP over MPQUIC with online learning to indicate 4D-scheduling. The results of experiments in both controlled emulation and real-world networks demonstrated that 4D-MAP can adapt to dynamic and heterogeneous network conditions. It provides lower re-buffering time, stream delay, and start-up latency without compromising visual fidelity than traditional multipath transmission solutions. Furthermore, this work presents a new architecture that divides transmission into different modules, i.e., sensing and decision-making. After decoupling the transmission, the sensed network condition and application characteristics can be used to match algorithm and parameter selection in decision-making, making it easier to provide specific QoS for applications. In the future, flexible sensing and multipath resource allocation can be customized based on the characteristics of various applications. Our next step is to investigate multipath challenges in other applications such as VR and AR through multipath.

**Conflict of Interest** The authors declare that they have no conflict of interest.

## References

[1] Huang Y K, Zhu Y W, Qiao X Q, Tan Z J, Bai B Y. AiTransfer: Progressive AI-powered transmission for real-time point cloud video streaming. In *Proc. the 29th ACM International Conference on Multimedia*, Oct. 2021, pp.3989–3997. DOI: 10.1145/3474085.3475624.

[2] De Coninck Q, Bonaventure O. Multipath QUIC: Design and evaluation. In *Proc. the 13th International Conference on Emerging Networking Experiments and Technolo-*

*gies*, Nov. 2017, pp.160–166. DOI: 10.1145/3143361.3143370.

[3] Zheng Z L, Ma Y F, Liu Y M, Yang F R, Li Z Y, Zhang Y B, Zhang J H, Shi W, Chen W T, Li D, An Q, Hong H, Liu H H, Zhang M. XLINK: QoE-driven multi-path QUIC transport in large-scale video services. In *Proc. the ACM SIGCOMM 2021 Conference*, Aug. 2021, pp.418–432. DOI: 10.1145/3452296.3472893.

[4] Li L H, Chu W, Langford J, Schapire R E. A contextual-bandit approach to personalized news article recommendation. In *Proc. the 19th International Conference on World Wide Web*, Apr. 2010, pp.661–670. DOI: 10.1145/1772690.1772758.

[5] Wu H J, Alay Ö, Brunstrom A, Ferlin S, Caso G. Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments. *IEEE Journal on Selected Areas in Communications*, 2020, 38(10): 2295–2310. DOI: 10.1109/jsac.2020.3000365.

[6] Zhang H, Li W Z, Gao S H, Wang X L, Ye B L. ReLeS: A neural adaptive multipath scheduler based on deep reinforcement learning. In *Proc. the 2019 IEEE Conference on Computer Communications*, Apr. 29–May 2, 2019, pp.1648–1656. DOI: 10.1109/INFOCOM.2019.8737649.

[7] Sullivan G J, Wiegand T. Video compression—From concepts to the H.264/AVC standard. *Proceedings of the IEEE*, 2005, 93(1): 18–31. DOI: 10.1109/JPROC.2004.839617.

[8] Viernickel T, Frömmgen A, Rizk A, Koldehofe B, Steinmetz R. Multipath QUIC: A deployable multipath transport protocol. In *Proc. the 2018 IEEE International Conference on Communications*, May 2018. DOI: 10.1109/ICC.2018.8422951.

[9] Ferlin S, Alay Ö, Mehani O, Boreli R. BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks. In *Proc. the 2016 IFIP Networking Conference (IFIP Networking) and Workshops*, May 2016, pp.431–439. DOI: 10.1109/IFIPNetworking.2016.7497206.

[10] Lim Y S, Nahum E M, Towsley D, Gibbens R J. ECF: An MPTCP path scheduler to manage heterogeneous paths. In *Proc. the 13th International Conference on Emerging Networking Experiments and Technologies*, Nov. 2017, pp.147–159. DOI: 10.1145/3143361.3143376.

[11] Shi H, Cui Y, Wang X, Hu Y M, Dai M L, Wang F Z, Zheng K. STMS: Improving MPTCP throughput under heterogeneous networks. In *Proc. USENIX Annual Technical Conference* (*USENIX ATC 18*), Jul. 2018, pp.719–730. DOI: 10.5555/3277355.3277425.

[12] Xing Y T, Xue K P, Zhang Y, Han J P, Li J, Liu J Q, Li R D. A low-latency MPTCP scheduler for live video streaming in mobile networks. *IEEE Trans. Wireless Communications*, 2021, 20(11): 7230–7242. DOI: 10.1109/TWC.2021.3081498.

[13] Han B, Qian F, Ji L S, Gopalakrishnan V. MP-DASH: Adaptive video streaming over preference-aware multipath. In *Proc. the 12th International on Conference on Emerging Networking Experiments and Technologies*, Dec. 2016, pp.129–143. DOI: 10.1145/2999572.2999606.

[14] Wu J Y, Yuen C, Cheng B, Yang Y, Wang M, Chen J L. Bandwidth-efficient multipath transport protocol for quality-guaranteed real-time video over heterogeneous wireless networks. *IEEE Trans. Communications*, 2016, 64(6): 2477–2493. DOI: 10.1109/tcomm.2016.2553138.

[15] Palmer M, Appel M, Spiteri K, Chandrasekaran B, Feldmann A, Sitaraman R K. VOXEL: Cross-layer optimization for video streaming with imperfect transmission. In *Proc. the 17th International Conference on Emerging Networking Experiments and Technologies*, Dec. 2021, pp.359–374. DOI: 10.1145/3485983.3494864.

[16] Palmer M, Krüger T, Chandrasekaran B, Feldmann A. The QUIC fix for optimal video streaming. In *Proc. the 2018 Workshop on the Evolution, Performance, and Interoperability of QUIC*, Dec. 2018, pp.43–49. DOI: 10.1145/3284850.3284857.

[17] Silva F, Bogusevschi D, Muntean G M. A MPTCP-based RTT-aware packet delivery prioritisation algorithm in AR/VR scenarios. In *Proc. the 14th International Wireless Communications & Mobile Computing Conference*, Jun. 2018, pp.95–100. DOI: 10.1109/IWCMC.2018.8450524.

[18] Cui L Z, Su D Y, Yang S, Wang Z, Ming Z. TCLiVi: Transmission control in live video streaming based on deep reinforcement learning. *IEEE Trans. Multimedia*, 2021, 23: 651–663. DOI: 10.1109/tmm.2020.2985631.

[19] Sun L Y, Zong T Y, Wang S Q, Liu Y, Wang Y. Towards optimal low-latency live video streaming. *IEEE/ACM Trans. Networking*, 2021, 29(5): 2327–2338. DOI: 10.1109/tnet.2021.3087625.

[20] Zhang L, Xu W Z, Lu D H, Cui L Z, Liu J C. MFVP: Mobile-friendly viewport prediction for live 360-degree video streaming. In *Proc. the 2022 IEEE International Conference on Multimedia and Expo*, Jul. 2022. DOI: 10.1109/icme52920.2022.9859789.

[21] Wang Z Y, Cui Y, Hu X Y, Wang X, Ooi W T, Cao Z, Li Y. MultiLive: Adaptive bitrate control for low-delay multi-party interactive live streaming. *IEEE/ACM Trans. Networking*, 2022, 30(2): 923–938. DOI: 10.1109/tnet.2021.3129481.

[22] Kaur K, Singh J, Ghumman N. Mininet as software defined networking testing platform. In *Proc. the 2014 International Conference on Communication, Computing & Systems*, Aug. 2014, pp.139–142.

[23] Wu H J, Alay O, Brunstrom A, Caso G, Ferlin S. FALCON: Fast and accurate multipath scheduling using offline and online learning. arXiv: 2201.08969, 2022. https://doi.org/10.48550/arXiv.2201.08969, Jan. 2024.

[24] Khalili R, Gast N, Popovic M, Le Boudec J Y. MPTCP is not pareto-optimal: Performance issues and a possible solution. *IEEE/ACM Trans. Networking*, 2013, 21(5): 1651–1665. DOI: 10.1109/TNET.2013.2274462.

[25] Langley A, Riddoch A, Wilk A, Vicente A, Krasic C, Zhang D, Yang F, Kouranov F, Swett I, Iyengar J, Bailey J, Dorfman J, Roskind J, Kulik J, Westin P, Tenneti R, Shade R, Hamilton R, Vasiliev V, Chang W T, Shi Z Y. The QUIC transport protocol: Design and Internet-

scale deployment. In *Proc. the 2017 Conference of the ACM Special Interest Group on Data Communication*, Aug. 2017, pp.183–196. DOI: 10.1145/3098822.3098842.

[26] Aloman A, Ispas A I, Ciotirnae P, Sanchez-Iborra R, Cano M D. Performance evaluation of video streaming using MPEG DASH, RTSP, and RTMP in mobile networks. In *Proc. the 8th IFIP Wireless and Mobile Networking Conference*, Oct. 2015, pp.144–151. DOI: 10.1109/WMNC.2015.12.

[27] Lu T, Pál D, Pál M. Contextual multi-armed bandits. In *Proc. the 13th International Conference on Artificial Intelligence and Statistics*, May 2010, pp.485–492.

[28] Zuo X T, Cui Y, Wang X, Yang J Y. Deadline-aware multipath transmission for streaming blocks. In *Proc. the 2022 IEEE Conference on Computer Communications*, May 2022, pp.2178–2187. DOI: 10.1109/INFOCOM48880.2022.9796942.

[29] Wang Z, Bovik A C, Sheikh H R, Simoncelli E P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 2004, 13(4): 600–612. DOI: 10.1109/tip.2003.819861.

[30] Frommgen A, Erbshäußer T, Buchmann A, Zimmermann T, Wehrle K. ReMP TCP: Low latency multipath TCP. In *Proc. the 2016 IEEE International Conference on Communications*, May 2016. DOI: 10.1109/icc.2016.7510787.

[31] Sergent M, Luu R P T, Elguero J. Statistical analysis of solvent scales. Part 1. *Anales de Química*, 1997, 93: 71–75.

[32] Roselló M M. Multi-path scheduling with deep reinforcement learning. In *Proc. the 2019 European Conference on Networks and Communications*, June 2019, pp.400–405. DOI: 10.1109/EuCNC.2019.8802063.

**Cong-Xi Song** received her B.S. degree in computer science and technology from Beihang University, Beijing, in 2017, and her M.S. degree in computer science and technology from National University of Defense Technology, Changsha, in 2019. She is currently pursuing her Ph.D. degree in cybersecurity in National University of Defense Technology, Changsha. Her research interests include multipath transmission technology and video communication.

**Biao Han** is currently an associate professor in College of Computer Science and Technology, National University of Defense Technology, Changsha. He received his B.S. degree in computer science and technology from National University of Defense Technology, Changsha, in 2007, and his Ph.D. degree in computer science from University of Tsukuba, Tsukuba, in 2013. His research interests are in cyber-physical systems, cybersecurity, and AI for networking.

**Jin-Shu Su** received his B.S. degree in mathematics from Nankai University, Tianjin, in 1985, and his M.S. and Ph.D. degrees in computer science from National University of Defense Technology, Changsha, in 1988 and 2000, respectively. He is a professor in College of Computer Science and Technology, National University of Defense Technology, Changsha. His research interests include Internet architecture, network security, and AI for networking.