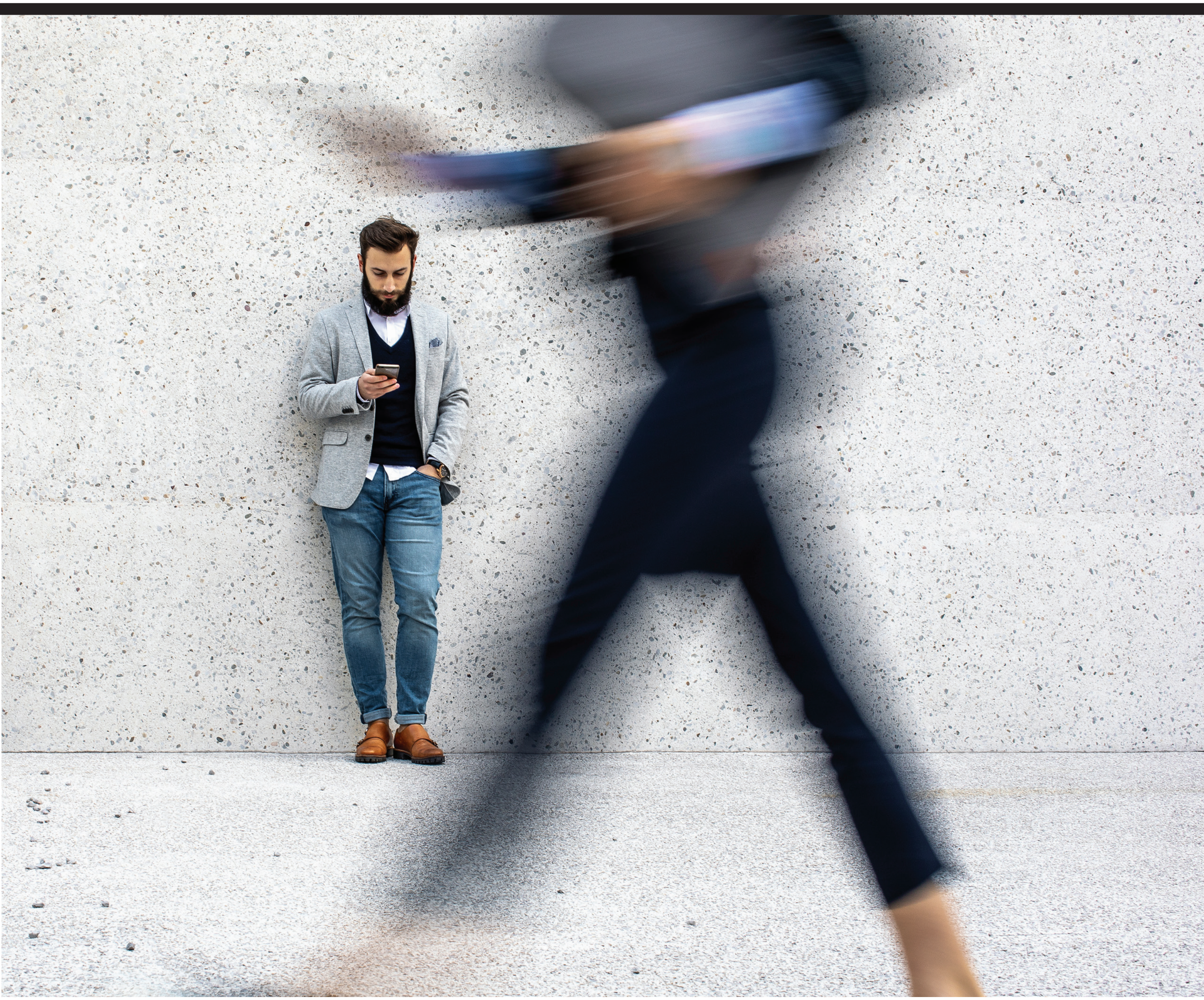
Gerui Lv, Qinghua Wu *Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing*
Yanmei Liu *Alibaba Group*
Zhenyu Li, Qingyue Tan *Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing*
Furong Yang, Wentao Chen, Hongyu Guo, Ying Chen *Alibaba Group*
Gaogang Xie *Chinese Academy of Sciences (CAS), School of Computer Science and Technology, UCAS*



Photo, istockphoto.com

CHORUS: Coordinating Mobile Multipath Scheduling and Adaptive Video Streaming

Excerpted from "Chorus: Coordinating Mobile Multipath Scheduling and Adaptive Video Streaming," *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24)*. https://dl.acm.org/doi/10.1145/3636534.3649359 © ACM 2024

Increasing bandwidth demands of mobile video streaming pose a challenge in optimizing the Quality of Experience (QoE) for better user engagement. Multipath transmission promises to extend network capacity by utilizing multiple wireless links simultaneously. Previous studies mainly tune the packet scheduler in multipath transmission, expecting higher QoE by accelerating transmission. However, because application-layer Adaptive BitRate (ABR) algorithms are inherently uncoordinated with multipath scheduling in the transport layer, multipath adaptive streaming can even experience lower QoE than single-path. This paper proposes *Chorus*, a cross-layer framework that coordinates multipath scheduling with adaptive streaming to optimize QoE jointly. Chorus establishes bidirectional feedback control loops between the server and the client. Furthermore, Chorus introduces Coarse-grained Decisions, which assist appropriate bitrate selection by considering the scheduling decision in throughput prediction, and Fine-grained Corrections, which meet the predicted throughput by QoE-oriented multipath scheduling. Extensive emulation and real-world mobile Internet evaluations show that Chorus outperforms the state-of-the-art MPQUIC scheduler, improving average QoE by 23.5% and 65.7%, respectively.

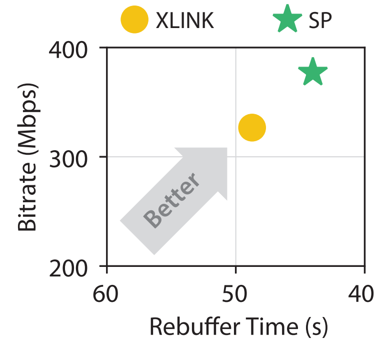
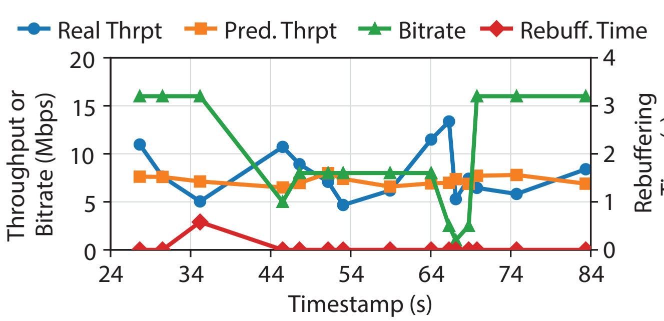


FIGURE 1. XLINK performs worse than SP in highly dynamic mobile networks.

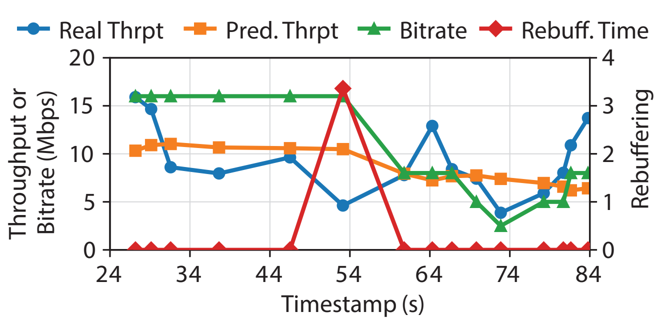
Recent years have witnessed the popular trend of HTTP-based mobile video applications [1], from traditional video-on-demand (VoD) and live video to 360° panoramic video and volumetric video that are applied in Augmented/Virtual/Mixed Reality (AR/VR/MR). To adapt the video quality to fluctuating bandwidth, HTTP-based adaptive streaming (HAS) is widely deployed in commercial video services. HAS encodes video content into multiple quality (bitrate) representations, each divided into chunks of equal playback duration (usually 2–5 seconds). Adaptive BitRate (ABR) algorithms in the client player, which dynamically select the bitrate for each chunk based on throughput prediction, play a crucial role in optimizing the Quality of Experience (QoE), including maximizing video bitrate and minimizing rebuffering time.

The rapid increase in bandwidth requirements (tens to hundreds of Mbps) of emerging video applications poses significant challenges to maintaining consistently high-quality delivery in mobile networks. As a promising solution, multipath transmission, including Multipath TCP (MPTCP) [2] and Multipath QUIC (MPQUIC) [3], extends network capacity by aggregating the bandwidth of different wireless links (e.g., WiFi and 5G). The packet scheduler, the core component of multipath transmission, determines when, how, and in what order to assign packets to all paths, to optimize transport performance, such as achieving higher throughput or reducing transmission time, thereby improving Quality of Service (QoS). MinRTT [4] is the typical default scheduler, selecting a path with the smallest RTT and available congestion window (CWND) for each packet. Reinjection is widely employed to improve MinRTT, which retransmits packets of slower paths on faster paths at the cost of redundancy.

While multipath transmission is expected to provide performance no worse than the best single-path, our comprehensive evaluations in mobile adaptive streaming show that: Even with higher chunk throughput, the ABR algorithm can still notably experience much lower QoE



(a) A case of SP-based ABR.



(b) A case of MinRTT+RI-based ABR.

FIGURE 2. A comparison case of multipath and single-path adaptive streaming performance in mobile networks.

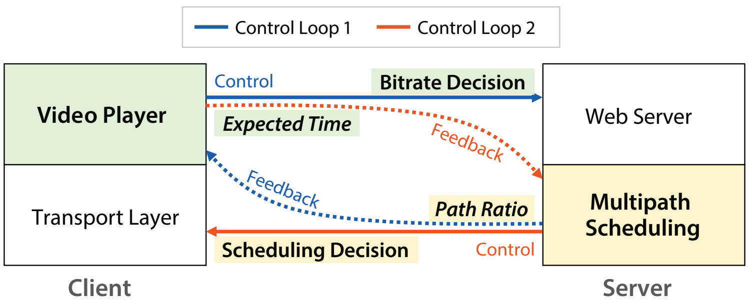


FIGURE 3. Chorus overview: bidirectional feedback control loops.

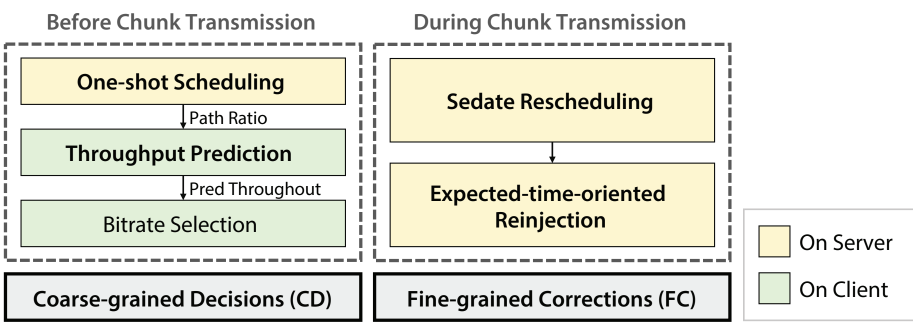


FIGURE 4. The submodules of Chorus's CD&FC.

on multiple paths than on a single path, especially in highly dynamic networks. The root cause of this issue is that *adaptive streaming is uncoordinated with multipath scheduling in dynamic mobile networks.* Specifically, ABR algorithms overlook the impact of multipath scheduling on perceived throughput, resulting in inaccurate throughput predictions. Previous studies have widely reported the significant impact of multipath scheduling on application throughput. Yet, since scheduling typically operates after the bitrate selection for each video chunk, it is overlooked by ABR algorithms. Consequently, multipath scheduling introduces additional uncertainty into ABR throughput predictions, leading to incorrect bitrate decisions and ultimately catastrophic QoE degradation.

To solve the above issue, we propose Chorus, a cross-layer framework that coordinates multipath scheduling with mobile adaptive streaming to jointly optimize QoE. Chorus aims to meet two necessary conditions for improving QoE in mobile adaptive streaming: (i) ensuring appropriate bitrate selection and (ii) providing transport performance that satisfies QoE requirements. Therefore, Chorus introduces a novel design named **Coarse-grained Decisions and Fine-grained Corrections (CD&FC)**. The CD phase achieves better multipath chunk throughput prediction by predetermining chunk-level packet scheduling decisions, and the FC phase meets the predicted throughput while balancing transport efficiency and cost considerations. To accomplish this, Chorus incorporates **bidirectional feedback control loops** between the server transport layer and the client application.

We have implemented Chorus using a user-space QUIC library [5] and integrated Chorus into a real-world mobile video system with minimal modifications. Extensive evaluations in both emulated and wild mobile networks confirm the consistent superiority of Chorus in optimizing QoE, attributed to its better multipath throughput prediction and adequate transport performance with minimal costs. In real-world scenarios, Chorus improves the average overall QoE by 65.7%–114.4% over XLINK [6], a state-of-the-art QoE-driven multipath scheduler, and single path QUIC. Furthermore, Chorus relies on only a few assumptions and computational resources, making it robust and practical to deploy in mobile video applications.

CHORUS: A CROSS-LAYER COORDINATION FRAMEWORK FOR MULTIPATH SCHEDULING AND ABR ALGORITHMS TO OPTIMIZE QOE JOINTLY

MOTIVATION

Multipath transmission seeks to optimize transport performance (QoS metrics), i.e. higher throughput or shorter completion time, through packet scheduling [6, 7, 8, 9].

However, enhancing multipath scheduling independently does not necessarily induce QoE improvements for adaptive video streaming, which has its own control module (i.e., ABR algorithms) and optimization target (i.e., QoE metrics).

We have evaluated multipath adaptive streaming using XLINK [6], a state-of-the-art packet scheduler. Notably, our results revealed that XLINK-based adaptive streaming underperforms compared to the best single-path (SP) in scenarios with highly fluctuating bandwidths. It showed an 11.7% lower bitrate and a 6.2% increase in rebuffering time on average, as depicted in Figure 1.

To determine whether this issue arises from insufficient transport performance, we next investigate MinRTT+RI, which allows unlimited packet reinjection (RI) when the CWND is available, representing the upper-bound performance for both XLINK and MinRTT [6]. Figure 2 illustrates a 60-second case from our controlled experiments. Compared to SP, MinRTT+RI indeed improves transport performance, achieving a 17.5% higher average chunk throughput. However, this improvement brings a 4.4% increase in total bitrates, but also a 4.7x increase in total rebuffering time, eventually resulting in a 20.6% decrease in total QoE.

Further analysis indicates that the multipath-based ABR algorithms are more susceptible to incorrect decisions due to inaccurate throughput predictions.

Specifically, MinRTT+RI is more prone to overestimating throughput than SP, with an 18.2% greater overestimation error rate at the 95th percentile. This consistent overestimation in multipath transmission mainly stems from its unique component: packet scheduling. Multipath packet schedulers determine their decisions for each chunk after the bitrate selection and thus are overlooked by ABR algorithms. In essence, *adaptive streaming operates as a prediction-based optimal control system* [10, 11], *whereas multipath scheduling is a black box to its core control module (ABR algorithms)*. This opacity increases prediction uncertainties, leading to control failures—inappropriate bitrate decisions that significantly degrade sessions' QoE.

In summary, our analyses reveal that **superior transport performance does not necessarily translate into better QoE for ABR algorithms**. The root cause is that ABR algorithms are blind to the impact of multipath scheduling on perceived throughput, thereby leading to more prediction errors and further severe QoE degradation due to incorrect bitrate decisions. At a high level, adaptive streaming is uncoordinated with multipath scheduling in dynamic mobile networks. These observations prompt us to consider cross-layer coordination between multipath scheduling and ABR algorithms.

Chorus Design

We propose Chorus, a cross-layer framework that coordinates multipath scheduling with ABR algorithms to optimize QoE jointly. To this end, Chorus incorporates **bidirectional feedback control loops**, as illustrated in Figure 3. By utilizing the QOE_CONTROL_SIGNAL frame (QoE frame for short) in MPQUIC [6, 12], Chorus facilitates a bidirectional exchange of information between the server and client across different layers. As a result, both entities can collaboratively work towards a unified goal: optimizing QoE. Specifically, the client-side ABR algorithm predicts chunk throughput based on the server's pre-determined scheduling decision (*Path Ratio* in Figure 3). In turn, the server packet scheduler adjusts its decision during chunk transmission in dynamic conditions, to meet the expected transmission time of the ABR algorithm (*Expected Time* in Figure 3). Note that the two control loops operate on different time grains.

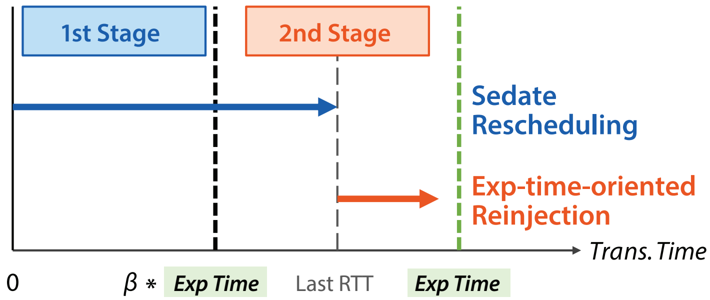


FIGURE 5. Timeline of Chorus during the Fine-grained Corrections (FC) phase.

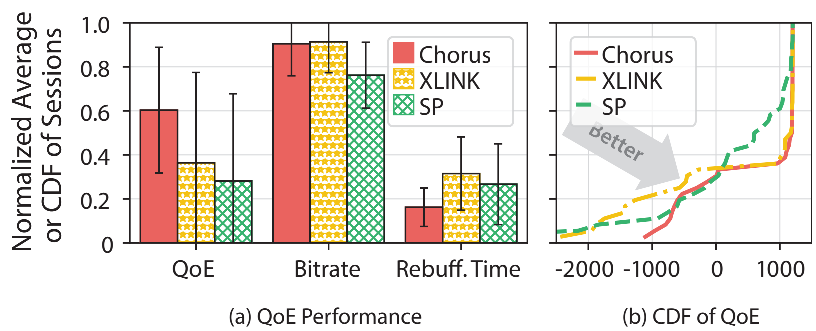


FIGURE 6. Overall performance of Chorus vs. baselines in the real world.

Chorus takes two necessary conditions for ABR algorithms to optimize QoE as its goals: (i) ensuring appropriate bitrate selection and (ii) providing transport performance that meets the needs of ABR algorithms. To achieve these goals, Chorus introduces a novel design called **Coarse-grained Decisions and Fine-grained Corrections (CD&FC)**, which is depicted in Figure 4. The CD phase takes place before transmitting a chunk, while the FC phase operates during the chunk transmission. In the CD phase, Chorus first predetermines the scheduling decision of all packets in the chunk, then predicts throughput for this chunk based on that decision and selects the bitrate. Once the transmission of the chunk starts, Chorus enters the FC phase and corrects the initial one-shot scheduling if it is non-optimal by Sedate Rescheduling and Expected-time-oriented Reinjection.

In this paper, Chorus is illustrated with two paths for convenience, but not confined to any specific number of paths, which can be easily extended to more paths.

Coarse-grained Decisions (CD)

In the CD phase, Chorus coordinates the *SERVER* with the *CLIENT* to predetermine the chunk-level scheduling decision for bitrate selection. This process operates as follows:

1. Informing Scheduling Decision: Periodically (every 200ms), the *SERVER* sends a QoE frame to the *CLIENT*, encapsulating its latest scheduling decision, namely the path assignment ratio.
2. Throughput Prediction: Using the last received scheduling information, the *CLIENT* predicts the next chunk's multipath throughput, calculates the corresponding expected transmission time given the selected bitrate, and sends it to the *SERVER* via a QoE frame. This expected time is utilized in the subsequent FC phase.
3. Bitrate Selection: The *CLIENT* feeds the ABR algorithm with the predicted throughput to select the next chunk's bitrate and makes an HTTP request for the corresponding chunk to the *SERVER*.
4. One-shot Scheduling: Upon receiving the *CLIENT*'s chunk request, the *SERVER* schedules packets of the entire HTTP response (chunk), assigning them across all paths at one time and then starting the chunk transmission.

One-shot Packet Scheduling. This one-shot scheduling aims to achieve simultaneous completion on all paths, thereby minimizing

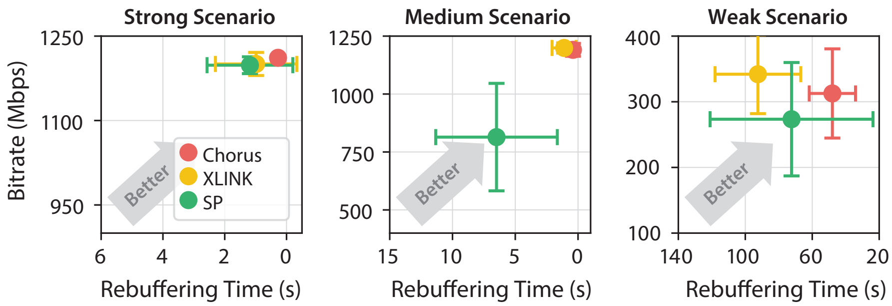


FIGURE 7. QoE performance of Chorus vs. baselines under various scenarios.

the total transmission time for each video chunk. In detail, Chorus assigns α of the packets at the beginning and $1 - \alpha$ of the packets at the ending of a chunk to the fast (with higher bandwidth) and slow paths, respectively. This assignment is performed only once per chunk and is not limited by CWND. The value of α is determined by the ratio of the fast path's bandwidth to the combined bandwidth of both paths. Note that Chorus does not expect this one-shot scheduling to be perfect during the transmission of the entire chunk due to possible network changes and invokes the subsequent FC phase to correct it if necessary.

Throughput Prediction for ABR Algorithms. Based on the one-shot scheduling decision information, Chorus aims to provide better multipath throughput prediction for existing ABR algorithms on the client side. Specifically, Chorus adopts a simple yet effective prediction method, which incorporates the server-side assignment ratio α and the client-side path receiving rate. This approach ensures accuracy by promptly detecting and adapting to changes in network conditions or scheduling on individual paths. Additionally, it is relatively conservative to avoid severe rebuffering events that significantly degrade QoE under dynamic networks.

Fine-grained Corrections (FC)

During the transmission of each chunk, Chorus enters the FC phase to cope with the possibility that the one-shot decision made in the CD phase is non-optimal due to prediction error. In the FC phase, Chorus needs to provide adequate transport performance by limited but effective corrections.

To this end, Chorus performs *two-stage corrections* in the FC phase, as depicted in Figure 5. The 1st stage begins with the chunk transmission. In this stage, Chorus aims to fully utilize the bandwidth of all paths under dynamic conditions through **Sedate Rescheduling**. If the transmission time exceeds a deadline based on the ABR logic's expected time, Chorus enters the 2nd stage. During the last RTT of the chunk transmission (when all packets have been sent), Chorus conducts **Expected-time-oriented Reinjection** for inflight packets to accelerate the transmission on the premise of minimizing traffic redundancy.

1st Stage. Rescheduling is activated by the following event: when one path has extra CWND available to send more packets (i.e., no assigned but unsent packets left), while another path still has unsent packets beyond its current CWND, indicating that the total bandwidth is underutilized. In this case, Chorus retrieves all unsent packets from the respective paths and reschedules them based on the latest bandwidth.

2nd Stage. Chorus's server sets a deadline (*Exp Time* in Figure 5) based on the expected time. During transmitting each chunk, Chorus continually checks if the transmission time exceeds the expected time. If so, Chorus reinjects inflight packets in the last RTT of the transmission, by checking if all packets are sent and at least one path has available CWND.

IMPLEMENTATION AND DEPLOYMENT

Chorus is implemented based on the multipath version of XQUIC [5], a user-space QUIC library. Chorus uses decoupled CCA (specifically, Cubic [13]) for each path, following [6, 8, 14]. A detailed technical report is available at https://greenlv.github.io/files/Chorus_MobiCom24_tech_report.pdf.

Emulation testbed. This testbed contains a simple server application, and a virtual client video player based on XQUIC. The ABR logic and the chunk request logic are implemented. We use Mahimahi and its multipath version mpshell to replay the traces collected in mobile networks.

Deployment in the real world. We build a real-world platform, including both the server and the mobile client player. The server runs Tengine Web Server with Chorus deployed, hosting video chunks. For the client, Chorus is integrated into MediaPlayer-Extended, an Android DASH video player. As per [7, 14], the WiFi link is selected as the primary path by default in Android. Finally, one 4G and two 5G Android smartphones are used to run the player APP.

EVALUATION

We have comprehensively evaluated Chorus in both emulated and real-world mobile Internet. Due to space constraints, we focus on presenting results from our real-world platform. The real-world evaluations involve three scenarios – strong, medium, and weak – classified based on the relationship between the highest bitrate and the average bandwidth of the WiFi and cellular links. Half of the tests were conducted while stationary and the other half while walking. For details on the experimental setup and additional results, please refer to [15].

Figure 6 displays the overall performance of all schemes across all scenarios (with 95% confidence). Compared to XLINK and SP, Chorus improves the total QoE by 65.7% and 114.4% on average, respectively. Specifically, Chorus achieves a similar bitrate sum with XLINK (18.8% higher than SP) and the lowest rebuffering time (48.6% and 39.2% reduction over XLINK and SP, respectively). Figure 6b further shows that Chorus performs well in almost all tests.

A closer inspection of the results reveals that Chorus's main improvement in QoE comes from the weak scenario, as illustrated in Figure 7. In both strong and medium scenarios, Chorus parallels XLINK's near-optimal QoE. Specifically, Chorus experiences no rebuffering after the first chunk and has the shortest total rebuffering time. Although SP maintains the highest bitrate selection when stationary in the strong scenario, its QoE decreases during mobility or in the medium scenario. Conversely, multipath schemes can leverage an extra link to maintain the highest video quality. In the weak scenario, Chorus notably outperforms XLINK and SP by reducing the rebuffering time by 48.1% and 33.7% on average, respectively. These gains stem from Chorus's better throughput prediction and expected-time-oriented transport performance while minimizing costs.

CONCLUSION

Most previous studies on mobile multipath transmission aim to optimize transport performance. However, this optimization does not necessarily induce better QoE for adaptive streaming. The root cause is that adaptive streaming is uncoordinated with multipath scheduling. This paper proposes Chorus, a cross-layer coordination framework for multipath scheduling and ABR algorithms to optimize QoE jointly. Chorus incorporates bidirectional feedback control loops and introduces Coarse-grained Decisions and Fine-grained Corrections (CD&FC). In this way, Chorus ensures appropriate bitrate selection and expected-time-oriented transport performance for multipath adaptive streaming. Evaluations in emulation and real-world mobile Internet demonstrate Chorus's consistent superiority in optimizing QoE, especially in the heavy tail. ■

Funding Acknowledgment

We thank the anonymous shepherd and reviewers for their constructive feedback, and the editors for their efforts to improve this paper. This work is partially supported by the National Key R&D Program of China (2022YFB2901800).

Gerui Lv is an assistant researcher at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). His research interests lie in video streaming, network transport protocol, and Internet measurements. He received his PhD in Computer System Architecture from the University of Chinese Academy of Sciences (UCAS) in 2024.

Qinghua Wu is an associate researcher at ICT, CAS. His research interests lie in network transport protocol and Internet measurements. He received his PhD from ICT, CAS in 2015.

Yanmei Liu is currently a Senior Staff Engineer at Alibaba Group. Her research focuses on QUIC, multipath QUIC, and transport protocol design. She received a BE in microelectronic technology from the University of Electronic Science and Technology of China in 2013.

Zhenyu Li is a professor at ICT, CAS. His research interests include Internet measurement and Networked Systems. He received his PhD at the Graduate School of CAS in 2009.

Qingyue Tan is a PhD student at the ICT, CAS. Her research interests lie in adaptive streaming and Internet measurements.

Furong Yang is currently a Senior Engineer at Alibaba Group. His research interests include congestion control, transport protocol, and Internet measurement. He received his PhD from UCAS in 2022.

Wentao Chen is an Engineer at Alibaba Group. He received his MS from Tongji University in 2020. His research interests include transport protocol and networked systems.

Hongyu Guo is a Senior Staff Engineer/Director at Alibaba Group. His research interests lie in mobile computing and systems.

Ying Chen is a Researcher/Senior Director at Alibaba Group, where Dr. Chen heads the Audiovisual Technology team at Taobao. He earned his PhD from Tampere University of Technology in Finland in 2010, and both his BS and MS degrees from Peking University in 2001 and 2004, respectively.

Gaogang Xie is a professor at the Computer Network Information Center of CAS, and the School of Computer Science and Technology, UCAS. His research interests include computer networking and distributed systems, and Internet measurement.

REFERENCES

[1] SANDVINE. 2023 global internet phenomena report. https://www.sandvine.com/global-internet-phenomena-report-2023, 2023.

[2] Alan Ford, Costin Raiciu, Mark J. Handley, and Olivier Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, January 2013.

[3] Yanmei Liu, Yunfei Ma, Quentin De Coninck, Olivier Bonaventure, Christian Huitema, and Mirja Kühlewind. Multipath Extension for QUIC. Internet-Draft draft-ietf-quic-multipath-12, Internet Engineering Task Force, July 2025. Work in Progress.

[4] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. April 2012. How hard can it be? Designing and implementing a deployable multipath TCP. *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 399–412, San Jose, CA. USENIX Association.

[5] alibaba/xquic. https://github.com/alibaba/xquic, 2025.

[6] Zhilong Zheng, Yunfei Ma, Yanmei Liu, Furong Yang, Zhenyu Li, Yuanbo Zhang, Jiuhai Zhang, Wei Shi, Wentao Chen, Ding Li, Qing An, Hai Hong, Hongqiang Harry Liu, and Ming Zhang. XLINK: QoE-driven multi-path QUIC transport in large-scale video services. *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM '21, 418–432, New York, NY, USA, 2021. ACM.

[7] Yeon-sup Lim, Erich M. Nahum, Don Towsley, and Richard J. Gibbens. ECF: An MPTCP path scheduler to manage heterogeneous paths. *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '17, page 147–159, New York, NY, USA, 2017. Association for Computing Machinery.

[8] Yihua Ethan Guo, Ashkan Nikravesh, Z. Morley Mao, Feng Qian, and Subhabrata Sen. 2017. Accelerating multipath transport through balanced subflow completion. *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, MobiCom '17, 141–153, New York, NY, USA. ACM.

[9] Hang Shi, Yong Cui, Xin Wang, Yuming Hu, Minglong Dai, Fanzhao Wang, and Kai Zheng. July 2018. STMS: Improving MPTCP throughput under heterogeneous networks. *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, 719–730, Boston, MA. USENIX Association.

[10] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A control-theoretic approach for dynamic adaptive video streaming over HTTP. *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, 325–338, New York, NY, USA. ACM.

[11] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with Pensieve. *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, 197–210, New York, NY, USA. ACM.

[12] Yanmei Liu, Yunfei Ma, Christian Huitema, Qing An, and Zhenyu Li. Multipath Extension for QUIC. Internet-Draft draft-liu-multipath-quic-04, Internet Engineering Task Force, September 2021. Work in Progress.

[13] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. Cubic: a new TCP-friendly high-speed TCP variant. ACM SIGOPS *Operating Systems Review*, 42(5):64–74.

[14] Bo Han, Feng Qian, Lusheng Ji, and Vijay Gopalakrishnan. 2016. MP-DASH: Adaptive video streaming over preference-aware multipath. *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, 129–143, New York, NY, USA. ACM.

[15] Gerui Lv, Qinghua Wu, Yanmei Liu, Zhenyu Li, Qingyue Tan, Furong Yang, Wentao Chen, Yunfei Ma, Hongyu Guo, Ying Chen, and Gaogang Xie. 2024. Chorus: Coordinating mobile multipath scheduling and adaptive video streaming. *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, ACM MobiCom '24*, 246–262, New York, NY, USA. ACM.