Research Statement

Zili Meng, Tsinghua University

The metaverse is coming – now we are closer than any time to embrace those real-time interactive multimedia streaming applications (e.g., cloud gaming, remote surgery, and virtual reality). However, while some applications (e.g., videoconferencing) have entered our lives, the others are still far from *a part of our life*. For example, users cannot wear a VR headset all day as normal glasses. People suffer from frequent stalls[1], feel sick due to motion lags[2], or have fear about the use of life-critical interactive applications such as remote surgery[3]. Thus, there is still a gap between *usable* and *usable as a part of life*.

One main reason for the gap is that the network is insufficient to support those applications. As a network researcher, my goal is to build and design networked systems to bring those real-time interactive multimedia applications as a part of our lives. This goal, however, is challenging because:

i **Extreme latency requirements of applications.** Although the existing network provides very high performance in the median[4], the *tail performance* still sucks. However, tail performance is very critical to those real-time interactive multimedia applications – they require an extremely *consistent* low latency at the 99.9th or 99.99th percentile: In a period of a hundred seconds, even the stuttering of one hundred milliseconds will degrade user's experience and potentially have drastic consequences on critical applications such as remote surgery or remote-driving cars.

ii **Complicated causes of latency degradation.** The propagation delay of the network used to be the dominant component, which previous researchers often focused on. When we focus on the 99.99th percentile and further, many additional factors will contribute to the increase in latency. Causes can be competition from other small web pages (§2.3), the transient interference from other devices such as microwaves (§1.1), or even fluctuation in the video codec (§2.1). Some of them are even dependent on each other. The scenario will be even more complicated as the video quality (resolution and frame-rate) goes up and more components are overloaded.

To this end, I investigate the root causes of latency spikes for real-time interactive multimedia applications in different scenarios. I profile and decouple each potential component in the end-to-end transport and quantify their contributions to the increase of latency at the tail. I challenge several long-standing assumptions in real-time multimedia transport.

- Previous efforts have been devoted to optimizing the latency on *data path* (*i.e.,* the time that a packet travels from the sender to the receiver). I show that the latency on *control path* (*i.e.,* the time from a change in the network happens to the sender reacts to the change) is also critical (§1).
- Existing solutions independently optimize the video codec and network transport, since they are in different communities. I find that the lack of coordination can lead to a huge transient increase of the delay (§2.1).
- Current designs react to network changes (*e.g.,* packet losses or flow competitions) in a backpressure way: Senders follow the signals from the network and adapt its policy. I show that such a reactive way degrades the transient performance during the adjustment (§2.2, §2.3).

Figure 1: Now we are closer to having virtual reality and other real-time multimedia applications in our daily life than at any other time in history. Their performance greatly depends on the underlying network support for interaction and streaming (photo credit: great.gov.uk).

[1] Future of cloud gaming | Deloitte Insights

[2] Virtual reality sickness - Wikipedia

[3] Fear of innovation: public's perception of robotic surgery

[4] For example, the maximum throughput for state-of-the-art WiFi is 9.6 Gbps (What Is Wi-Fi 6? - Intel).

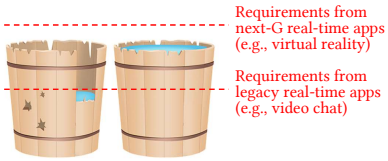

Figure 2: How much water the barrel can contain is decided by the shortest board. The network used to be the shortest board in latency optimization. However, many more components can be the shortest board when optimizing the tail latency.

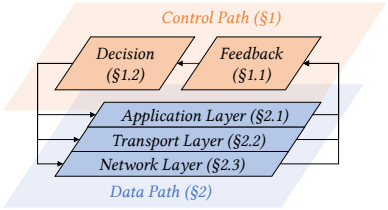

Figure 3: Network researchers usually split different functionality into different layers. My prior (§1, §2) research works on all layers from the network layer and above to optimize end-to-end latency. I have also recently started to collaborate with sister communities (§3) to optimize the latency.

Impact: Owing to my intrigue for *real-world* systems, my research has received attention from both the academia and industry in the network community. Particularly, several of my research has been used in production by Tencent, Alibaba, Kuaishou, and Baidu. Based on this research, I'm honored to receive the Microsoft Research PhD Fellowship in Asia[5], and the ByteDance PhD Fellowship[6]. My research also received widespread media coverage and paper awards[7] from the community.

[5] Fellowship at Microsoft Research Asia (2020)

[6] ByteDance Scholar Program (2022)

[7] SIGCOMM 2018 SRC Gold Medal, IEEE/ACM IWQoS 2021 Best Paper Award, IEEE ICC 2020 Best Paper Award

1 *Control Path Latency*

Since the network condition is fluctuating all the time by nature, how to support those applications with *consistent* latency is not trivial. Rate adaptation is a critical mechanism to adapt to network changes, but existing solutions are designed mainly to optimize the median latency in the steady state. In this case, the latency will still increase transiently during rate adaptation and fail to meet the latency requirements at the extreme percentiles. In this case, the control path latency (the time from a change in the network happens to the sender reacts to the change) is very critical. It contains two parts:

- Latency of measurement feedback: the time from a change happening in the network to the sender knowing it happened (§1.1).
- Latency of decision making: the time from the sender first knowing that the network change occurred, to the sender finally reacting to the change and adjusting the sending rate (§1.2).

1.1 *Measurement Feedback*

Feedback delay is important for the transient behavior of real-time applications when the network capacity is reduced. Ideally, senders would react quickly when bandwidth reduction occurs, e.g., by reducing their bitrate to prevent from high latency and loss. Unfortunately, senders are fundamentally limited in how quickly they can react to congestion because congestion signals are carried along the same congested path as data packets. Put simply, to observe that the bottleneck queue is filling, a sender must first receive feedback from a packet that has actually waited in that queue.

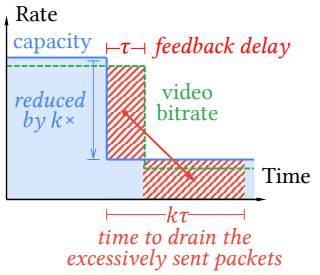

Figure 4: The feedback loop plays a dominant role when the network capacity drops. The solid blue line denotes the drop in link capacity in the bottleneck, and the dashed green line denotes how the sender will react and reconverge to the new sending rate. Users will suffer from interaction lags during the process of draining those excessively sent packets.

I showed that the inflation of feedback delay in the control path can drastically compromise the latency at the tail. I then proposed Zhuge[8], which reduced the inflated feedback delay without modifying the sender or receiver. In this case, Zhuge can effectively reduce stuttering events for wireless real-time streaming users when deployed on commercial wireless routers. Zhuge also attracted attention from the media[9].

[8] **Zili Meng**, Y. Guo, C. Sun, B. Wang, J. Sherry, et al. Achieving Consistent Low Latency for Wireless Real-time Communications with the Shortest Control Loop. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '22)*, 2022

[9] APNIC Blog

1.2 *Decision Making*

Meanwhile, the decision-making in current rate adaptation algorithms is increasingly time-consuming and unreliable at tail. This is critical since decisions in rate adaptation are made at a very high frequency of O(1 ms). At the 99.99th percentile, delayed or incorrect decisions will also degrade the latency of real-time interactive multimedia applications. The underlying reason is that algorithms recently proposed in rate adaptation are becoming increasingly complex. Researchers introduce integer linear programming (ILP) or deep neural networks (DNNs) to optimize the rate adaptation algorithms.

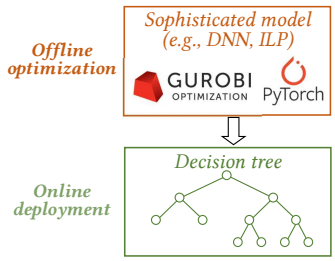

Figure 5: By decoupling the offline optimization from online deployment, my solutions can reduce the uncertainty in the decision-making in rate adaptation of real-time streaming applications.

I challenge a fundamental assumption: rate adaptation algorithm designers increase the complexity of models not to make them *more expressive*, but to

make them *easier to optimize*. Therefore, I decouple offline optimization and online execution for bitrate adaptation in the streaming[10], which can reduce decision-making latency for client devices by up to two orders of magnitude. I further demonstrate the performance improvements by reducing the delayed and incorrect decisions in rate adaptation[11], which has been received well in the community[12]. The outcome of this work has also been adopted by Tencent and Kuaishou in their video platforms in production.

2 Data Path Latency

To simplify network engineering, network researchers split the communications between systems into different layers. As the focus of applications moves from median percentiles to extreme tails, all the application layer (§2.1), transport layer (§2.2), and network layer (§2.3) will contribute to the increase of tail latency. My research revisits the causes of latency in different layers in the Internet architecture and rethinks if the current design in the respective layer is suitable for real-time interactive multimedia applications.

2.1 Application Layer

Compared to legacy real-time applications (video chat), the video quality of next-generation interactive multimedia applications (cloud gaming, VR) also increases significantly. This will impose a new challenge on the application layer: Streams with higher frame-rate and resolution will take longer time to be processed, which might make the application layer to be the bottleneck. In this case, data will be buffered between the network stack and the application. However, the buffer between the application and the network is not designed for low latency and lacks active management. The queue between the network and the application (streaming decoder) will be overloaded.

Therefore, I propose an adaptive frame rate (AFR)[13] controller that helps to achieve ultra-low latency by actively managing the queue between the network stack and application, and adaptively coordinating the frame rate with fluctuating network conditions and application capacity. AFR tackles a series of challenges and introduces new control signals from the arrival and service stochastic processes to reduce the queueing delay. AFR has been deployed in the cloud gaming service of Tencent for more than one year.

2.2 Transport Layer

I also rethink how the sender should recover the packet loss in the transport layer. With the recent development of new infrastructure (e.g., 5G and WiFi 6) and new framework (e.g., edge computing), the median network latency has been reduced from 100-200ms to 10-20ms. I collaborate with researchers in the human-computer interaction community and find that such a latency has now gone across a significant boundary – the median network latency is lower than the human's perception ability.

Therefore, I challenge a common belief in real-time streaming applications – existing work will aggressively add redundant packets to avoid retransmissions due to previous high network latency. With reduced network latency, it is not necessary to always add redundant packets to avoid retransmissions. In response, I reinvestigate the relationship between retransmission and redundancy, and design a new packet loss recovery mechanism, Hairpin[14], to balance between the tail latency and bandwidth cost. Notably, with the new

[10] **Zili Meng**, J. Chen, Y. Guo, C. Sun, H. Hu, et al. PiTree: Practical Implementation of ABR Algorithms Using Decision Trees. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, 2019

[11] **Zili Meng**, M. Wang, J. Bai, M. Xu, H. Mao, et al. Interpreting Deep Learning-based Networking Systems. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '20)*, 2020

[12] SIGCOMM '20 (Top 10 by citations).

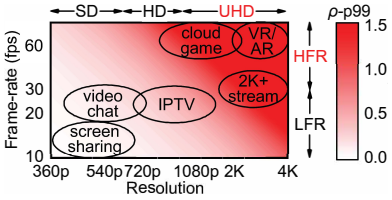

Figure 6: For next-generation real-time streaming applications, with the simultaneous increase of resolution and frame-rate, the queue between network and application (video decoder) at the client will be overloaded. ρ is the load of that queue.

[13] **Zili Meng**, T. Wang, Y. Shen, B. Wang, M. Xu, et al. Enabling High Quality Real-Time Communications with Adaptive Frame-Rate. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI '23)*, 2023c

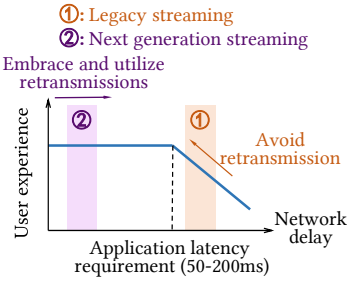

Figure 7: Legacy real-time streaming applications struggle to avoid retransmission to reduce the end-to-end latency. With a better infrastructure deployment and reduced network propagation delay, users can tolerate the limited times of retransmissions due to the perception ability of humans.

[14] **Zili Meng**, X. Kong, J. Chen, B. Wang, M. Xu, et al. Hairpin: Rethinking Packet Loss Recovery in Edge-based Interactive Video Streaming. In *Submission*, 2023b

design space of co-optimizing the redundancy and retransmission, Hairpin is able to reduce the latency jitters by one order of magnitude.

2.3 Network Layer

A final thing that raises my attention is the latency in the network layer. In the current Internet architecture, it is always end hosts that take care of the behaviors of the capricious network. When the network changes, rate adaptation algorithms on the end hosts cover for the network to provide a consistent experience for the application. However, when we are looking at the 99.99th percentile, the covering-up process will lead to the latency increase. In this case, I believe that the network itself should be more responsible and be considerate on end hosts' reactions when reallocate network resources.

To that end, I investigate how existing scheduling algorithms in the network layer, e.g., fair queueing, FIFO, behave in network fluctuations. The existing queue scheduling algorithms cannot provide consistent performance during the transient period where the network condition changes. Confucius[15], a new queue management mechanism, is consequently designed to retroactively make the network change to follow the sender's rate adaptation algorithm, which significantly improves the transient performance.

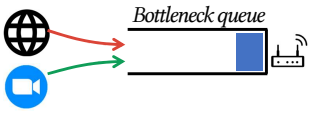

Figure 8: When a real-time streaming application (e.g., Zoom) competes with a Web application, in the time when the Web application joins the competition, the real-time application will suffer a transient latency spike.

[15] **Zili Meng**, N. Atre, M. Xu, J. Sherry, and M. Apostolaki. Confucius Queue Management: Be Fair But Not Too Fast. In *Submission*, 2023a

3 Future Work

I intend to leverage my knowledge to optimize real-time systems toward two main research directions: facilitating cross-community knowledge (§3.1), and generalizing my research to more real-time applications (§3.2).

3.1 Cross-community Optimization

Real-time interactive multimedia applications involve not only the network community but also many sibling communities in both the systematic and the algorithmic aspects. In the future, I would like to work on leveraging the information and knowledge from other communities to further optimize the performance of those applications.

Systematic – Multi-component coordination. Although the network takes most of the end-to-end latency of real-time streaming applications, it does not mean that the network is the only component contributing to latency. For example, video encoding and decoding, CPU scheduling, and graphics rendering can also have latency jitters during runtime. I have some preliminary attempts to co-design with the video codec to improve latency in the application layer[16]. I am also investigating how to achieve real-time performance on non-real-time operating systems (e.g., Windows or MacOS), especially when more and more content providers are customizing their own network stacks. Finally, I am eager to collaborate with the wireless community to see if coordination with the underlying wireless protocols can bring additional benefits to real-time interactive multimedia applications.

Algorithmic – Optimization with advanced algorithms. With more and more components that need to be considered when making decisions, the state and the action space of decision-making also increase. In this case, we do need some tools to automate the process of designing new algorithms. I made some preliminary attempts to apply deep learning-based algorithms in scheduling[17] and congestion control[18]. With efficient and optimal algorithms and solvers popping up, I am also working on how to better utilize these tools

[16] T. Wang, **Zili Meng**, M. Xu, R. Han, and H. Liu. Enabling High Frame-rate UHD Real-Time Communication with Frame-skipping. In *Proceedings of the 3rd ACM Workshop on Hot Topics in Video Analytics and Intelligent Edges (HotEdgeVdeo '21), co-located with MobiCom '21*, 2021

[17] H. Mao, M. Schwarzkopf, S. Venkatakrishnan, **Zili Meng**, and M. Alizadeh. Learning Scheduling Algorithms for Data Processing Clusters. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '19)*, 2019

[18] J. Zhang, E. Dong, **Zili Meng**, Y. Yang, M. Xu, et al. WiseTrans: Adaptive Transport Protocol Selection for Mobile Web Service. In *Proceedings of the Web Conference (WWW '19)*, 2021

to solve problems in real-time streaming applications.

3.2 Generalization to Other Applications

The demand for real-time transport is beyond the scope of multimedia transport. Especially in recent years, many more applications require real-time transport, which I would also like to work on.

Vehicle network. In emerging applications such as autonomous driving and vehicle-to-everything (V2X), reliable transport inside or between vehicles is very critical. A delayed signal from the smart devices to the vehicle can cause the car to miss the exit or even result in a crash. Thus, vehicle networks sometimes have even more extreme requirements for latency consistency. During my internship at a switching ASIC startup, I proposed a queue scheduling algorithm by utilizing the time-sensitive network to provide deterministic latency for data communications inside vehicles[19]. I am also planning to extend my previous research efforts to vehicle networks.

[19] Z. Ruan, **Zili Meng**, and Y. Huang. A Scheduling Method, Device, and Electronic Equipment, 2022. Chinese Patent CN113872887A. Granted: August 16, 2022

Internet of Things. With the development of the Internet of Things (IoT), it is expected that the sensors in daily life will significantly increase. Many experimental playgrounds have been established in the context of smart factories or smart cities. Among them, real-time communication between the sensor and controllers is also necessary. However, an outstanding feature of IoT is that the volume of communication is massive – thousands of sensors are talking to each other. In the future, I intend to address the challenges and build real-time IoT systems by leveraging networking advancements.

References

Click on the title to access the corresponding PDF. All my publications, talks, and CV are also available online at zilimeng.com.

H. Mao, M. Schwarzkopf, S. Venkatakrishnan, **Zili Meng**, and M. Alizadeh. Learning Scheduling Algorithms for Data Processing Clusters. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '19)*, 2019.

Z. Ruan, **Zili Meng**, and Y. Huang. A Scheduling Method, Device, and Electronic Equipment, 2022. Chinese Patent CN113872887A. Granted: August 16, 2022.

Zili Meng, J. Chen, Y. Guo, C. Sun, H. Hu, et al. PiTree: Practical Implementation of ABR Algorithms Using Decision Trees. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, 2019.

Zili Meng, M. Wang, J. Bai, M. Xu, H. Mao, et al. Interpreting Deep Learning-based Networking Systems. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '20)*, 2020.

Zili Meng, Y. Guo, C. Sun, B. Wang, J. Sherry, et al. Achieving Consistent Low Latency for Wireless Real-time Communications with the Shortest Control Loop. In *Proceedings of the ACM SIGCOMM Conference (SIGCOMM '22)*, 2022.

Zili Meng, N. Atre, M. Xu, J. Sherry, and M. Apostolaki. Confucius Queue Management: Be Fair But Not Too Fast. In *Submission*, 2023a.

Zili Meng, X. Kong, J. Chen, B. Wang, M. Xu, et al. Hairpin: Rethinking Packet Loss Recovery in Edge-based Interactive Video Streaming. In *Submission*, 2023b.

Zili Meng, T. Wang, Y. Shen, B. Wang, M. Xu, et al. Enabling High Quality Real-Time Communications with Adaptive Frame-Rate. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI '23)*, 2023c.

T. Wang, **Zili Meng**, M. Xu, R. Han, and H. Liu. Enabling High Frame-rate UHD Real-Time Communication with Frame-skipping. In *Proceedings of the 3rd ACM Workshop on Hot Topics in Video Analytics and Intelligent Edges (HotEdgeVdeo '21), co-located with MobiCom '21*, 2021.

J. Zhang, E. Dong, **Zili Meng**, Y. Yang, M. Xu, et al. WiseTrans: Adaptive Transport Protocol Selection for Mobile Web Service. In *Proceedings of the Web Conference (WWW '19)*, 2021.