# Evaluation of MPTCP with BBR Performance on Wi-Fi/Cellular networks for Video Streaming

Masayoshi Kondo*, Dirceu Cavendish**, Daiki Nobayashi**, Takeshi Ikenaga**
*Graduate School of Engineering, **Faculty of Engineering
Kyushu Institute of Technology
Fukuoka, Japan
e-mail: kondo.masayoshi146@mail.kyutech.jp {cavendish@ndrc, nova@ecs, ike@ecs}.kyutech.ac.jp

***Abstract*—Video streaming makes most of Internet traffic nowadays, being transported over Hypertext Transfer Protocol/Transmission Control Protocol (HTTP/TCP). Being the predominant transport protocol, TCP stack performance in transporting video streams has become paramount, specially with regard to multipath transport protocol innovation and multiple client device interfaces currently available. In this paper, we characterize Bottleneck Bandwidth and Round-trip propagation time (BBR) congestion control performance when streaming video over cellular and Wi-Fi access networks, comparing its performance to other available congestion control schemes and path schedulers. We use network performance level, as well as video quality level metrics to characterize multiple path schedulers and resulting network and application layers' interactions.**

***Keywords—Video streaming; TCP congestion control; Multipath TCP; Packet retransmissions; Packet loss.***

## I. Introduction

Data transmission over the Internet relies on transport protocols to control network congestion and avoid uncontrolled data losses. Transmission Control Protocol (TCP) has become the de facto transport protocol of the Internet, supporting reliable data delivery for most applications. Regarding streaming applications, the most dominant type of application over the Internet, stream quality is related to two factors: the amount of data discarded at the client due to excessive transport delay/jitter; data rendering stalls due to lack of timely playout data. It is worth noting that transport delays and data starvation depend heavily on how TCP handles retransmissions during flow and congestion control.

The evolution of portable device hardware, in particular the support of multiple high bandwidth interfaces, has motivated the development of multipath transport protocols, allowing video streaming over multiple IP interfaces and diverse network paths. Multipath video streaming is attractive because it not only increases aggregated device downloading bandwidth capacity, but also improves transport session reliability during transient radio link impairments in handoff situations. An important function in multipath transport is the selection of a path among various active networking paths (sub-flows), which can be done on a packet by packet basis. A path packet scheduler is used for this purpose, and should be designed to prevent head-of-line blocking across various networking paths, potentially with diverse loss and delay characteristics. Head-of-line blocking occurs when data already delivered at the receiver has to wait for additional packets that are blocked at another sub-flow, potentially causing incomplete or late frames to be discarded at the receiver, as well as stream rendering stalling. As the interplay between path schedulers and TCP variants ultimately define streaming quality, we propose to analyze video performance vis-a-vis popular TCP variants, with attention to BBR [1] and various path schedulers.

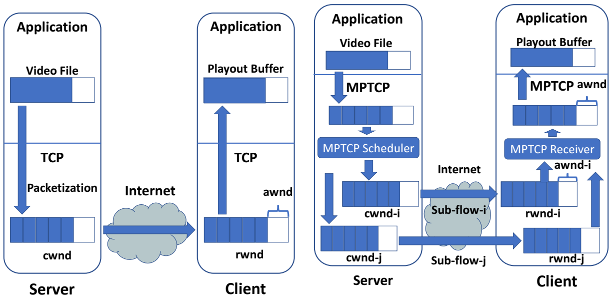
The paper is organized as follows. Related work is included in Section II. Section III describes video streaming transport over Transmission Control Protocol, including different TCP variants, one of which BBR. Section IV describes recently proposed alternative path schedulers. Section V characterizes video streaming performance over Wi-Fi and cellular paths via network emulation. We study the performance of a default (shortest delay) path scheduler, as well as alternative schedulers, working with popular TCP variants. Section VI summarizes our studies and addresses directions we are pursuing as follow up to this work.

## II. Related Work

Several multipath transport studies have appeared in the literature, mostly focusing on throughput performance of data transfers over mobile networks (see [2] and related work). More recently, path scheduler research has been recognized as an important driver of multipath transport sessions performance. For instance, [3] has analyzed loss based congestion control TCP variants interactions with minimum Round Trip Time (RTT) default Multipath TCP (MPTCP) path scheduler, and showed how sender/receiver buffer dimensioning impacts throughput performance via inflation of sub-flow RTTs. Little research work, however, has focused on video streaming performance over multiple paths. Motivated by vehicular communication in assisted driving systems, multipath video streaming on ad-hoc network studies have appeared. For instance, [4] introduces an interference aware multipath video streaming scheme in Vehicular Ad-hoc Networks (VANETs). The authors' goal is reliable transport of high quality video streams on vehicle to vehicle communication over multiple paths. Throughput performance is evaluated, taking into account interference within neighbors, as well as shadowing effects onto Signal to Noise ratio, and data delay. They seek to minimize video freezes and dropped frames, via link layer channel interference control, coupled with efficient routing strategies on ad-hoc vehicular networks. In contrast, we focus on video streaming over regular Internet paths, where link layer channel

ISBN: 978-1-61208-975-1

and route optimization opportunities are limited. Integrated optimization of video and transport layer is proposed by [5], where they introduce a path-and-content-aware path selection approach coupling MPEG Media Transport (MMT) with multipath transport. They estimate path quality conditions of each subflow, and avoid sending I-frames on paths of low transport quality. A similar approach, where different sub-flows are utilized for segregating high priority packets of Augmented Reality/Virtual Reality streams has been introduced by Silva et al. [6]. In contrast, our current work does not couple applications with multipath transport, rather focusing on generic path schedulers and TCP variants to deliver high quality video streaming. About new path schedulers, Ferlin et al. [7] have introduced a path scheduler based on a path head-of-line blocking predictor. They carry out emulation experiments of their proposed scheduler against minimum RTT default scheduler, in transporting bulk data traffic, Web transactions and Constant Bit Rate (CBR) applications. They use goodput, data transport completion time and packet delays as performance evaluation metrics. Still on new schedulers, Kimura et al. [8] have shown throughput performance improvements using schedulers driven by path sending rate and TCP window space, on bulk data transfers. Also, Xue et al. [9] have introduced a path scheduler based on estimation of the amount of data each path is able to transmit. They evaluate the scheduler's throughput performance on simulated network scenarios. In contrast, in our previous works, we have introduced multipath path scheduling generic principles, which can be applied in the design of various path schedulers to specifically improve video stream quality. Using these principles, we have introduced in [10] Multipath TCP path schedulers based on dynamically varying path characteristics, such as congestion window space and estimated path throughput. In addition, in [11], we have also proposed to enhance path schedulers with TCP state information, such as whether a path is in fast retransmit and fast recovery states. Finally, in [12], we have introduced a novel concept of sticky scheduling, where once a path switch is executed, the scheduler stays with the new path until the path bandwidth resources become exhausted. In this work, we evaluate several path schedulers, some of our proposal, in combination with popular BBR TCP [1] over realistic Wi-Fi/Cellular multipath scenarios, focusing on video quality at application layer. We seek to determine whether BBR delivers high performance in combination with path schedulers over multiple network scenarios. BBR performance in MPTCP transport is a novelty. Reference [13] has recently introduced one such study, where BBR is evatualed in combination with a new adaptive packet scheduling scheme (adaptive redundant + predictive). Multiple copies of a packet are injected in paths of low quality, for reliability improvement. The scheme also predicts packet delivery on paths in order to keep in order delivery, mitigating head of line blocking. Their evaluation, however, is limited to throughput and download time performance metrics of files. In contrast, our current study of BBR and schedulers focuses on video streaming quality evaluation.



(a) TCP (b) MPTCP

Figure 1. Video Streaming over TCP/MPTCP

## III. Video Streaming Over TCP

Video streaming over HTTP/TCP involves a HTTP server storing video content, where video files can be streamed upon HTTP requests over the Internet to video clients. At the transport layer, a TCP variant provides reliable transport of video data over IP packets between the server and client end points. Figure 1 illustrates these video streaming components. As mentioned, HTTP server stores encoded video files. Triggered by a HTTP video request, a TCP sender is instantiated to transmit packetized data to the client machine, connected to the application by a TCP socket. At the TCP transport layer, a congestion window is used at the sender to control the amount of data injected into the network. The size of the congestion window ($cwnd$) is adjusted dynamically, according to the level of congestion experienced through the network path, as well as the space available for data storage ($awnd$) at the TCP client receiver buffer. Congestion window space at the sender is freed only when data packets are acknowledged by the receiver. Lost unacknowledged packets are retransmitted by the TCP layer to ensure reliable data delivery. At the client, in addition to acknowledging arriving packets, the TCP receiver informs the TCP sender about its current available space, so that $cwnd \leq awnd$ condition is enforced by the sender at all times. At the client application layer, a video player extracts data from a playout buffer, which draws packets delivered by the TCP receiver from its socket buffer. The playout buffer hence serves to smooth out variable network throughput.

### A. Video Application and TCP Transport Interaction

When tracing data path, at the server side, the HTTP server transfers data into the TCP sender socket according to TCP $cwnd$ space availability. Hence, the injection rate of video data into the TCP socket is constrained by the congestion condition of the network path, reflected at the $cwnd$ size, and thus does not follow the video variable encoding rate. On its turn, TCP throughput performance is affected by the RTT of the TCP session over a specific path, since only up to a $cwnd$ worth of data can be delivered without acknowledgements. Hence, for a given $cwnd$ size, from the moment a first packet is sent until the first acknowledgement arrives back, the TCP session throughput is capped at $cwnd/RTT$. As there are various possible TCP congestion avoidance schemes regulating $cwnd$, according to the TCP variant, the size of the congestion window size is computed by a specific algorithm at the time of packet acknowledgement reception by the TCP source. Regardless of the variant, however, the size of the congestion window computed is capped by the available TCP receiver space $awnd$, communicated back from the TCP client, in order to ensure that the receiver buffer never overflows. At the client side, video data is retrieved from the TCP client socket by the video player into a playout buffer, from which data is delivered to the video renderer. Even though client playout buffer may underflow, if TCP receiver window empties out, the playout buffer never overflows, since the player will not pull more data into the playout buffer if no space is available.

### B. Multipath TCP

Multipath TCP is an Internet Engineering Task Force (IETF) extension of TCP transport layer protocol supporting data transport over multiple concurrent legacy TCP sessions [14]. The network multipath transmission of the transport session is hidden from application layer by a legacy TCP socket exposed per application session. Under the hood, at the transport layer, however, MPTCP coordinates concurrent TCP variant sub-flows, each of which in itself unaware of the multipath nature of the application session. In order to accomplish multipath transport, a path scheduler connects the application socket with transport sub-flows, extracting packets from the application facing MPTCP socket, selecting a sub-flow for transmission, and injecting packets into the selected sub-flow. MPTCP transport architecture is depicted in Figure 3(b).

The first and most used path scheduler, called default scheduler, selects the path with shortest Round Trip Time (RTT) among paths with currently available congestion window space for new packets. Other path schedulers have appeared recently. These path schedulers can operate in two different modes: uncoupled, and coupled. In uncoupled mode, each sub-flow congestion window $cwnd$ is adjusted independently of other sub-flows. On the other hand, in coupled mode, MPTCP scheduler couples the congestion control of the sub-flows, by adjusting the congestion window $cwnd_k$ of a sub-flow $k$ according with current state and parameters of all available sub-flows. Although many coupling mechanisms exist, we focus on performance study of Bottleneck Bandwidth and round trip [1], Linked Increase Algorithm (LIA) [15], Opportunistic Linked Increase Algorithm (OLIA) [16], and Balanced Linked Adaptation algorithm (BALIA) [17]. We include also evaluation of uncoupled schedulers recently proposed.

Regardless of path scheduler used, IETF MPTCP protocol supports the advertisement of multiple IP interfaces available between two endpoints via specific TCP option signalling. IP interfaces may be of diverse nature (e.g., Wi-Fi, cellular). A common signalling issue is caused by intermediate IP boxes, such as firewalls, blocking IP options. Paths that cross service providers with such boxes may require Virtual Private Network (VPN) protection so as to preserve IP interface advertising between endpoints. In addition, multipath transport requires MPTCP stack at both endpoints for the establishment and usage of multiple paths.

### C. TCP variants

TCP protocol nowadays have branched into different variants, implementing different congestion window adjustment schemes. TCP protocol variants can be classified into delay and loss based congestion control schemes. Loss based TCP variants use packet loss as primary congestion indication signal, typically performing congestion window regulation as $cwnd_k = f(cwnd_{k-1})$, which is ack reception paced. Most $f$ functions follow an Additive Increase Multiplicative Decrease (AIMD) window adjustment scheme, with various increase and decrease parameters. AIMD strategy relies on a cautious window increase (additive) when no congestion is detected, and fast window decrease (multiplicative) as soon as congestion is detected. TCP NewReno [18] and Cubic [19] are examples of AIMD strategies. In contrast, delay based TCP variants use queue delay information as the congestion indication signal, increasing/decreasing the window if the delay is small/large, respectively. Compound [20] and Capacity and Congestion Probing (CCP) [21] are examples of delay based congestion control variants. Delay based congestion control does not suffer from packet loss undue window reduction due to random, not congestion, packet losses, as experienced in wireless links. Regardless of the congestion control scheme, TCP variants follow a phase framework, with an initial slow start, followed by congestion avoidance, with occasional fast retransmit, and fast recovery phases. BBR congestion control may be considered delay based, since BBR measures the bandwidth and RTT of the bottleneck which a flow goes through [1]. Based on such measurements, BBR adjusts the sending rate to make the best use of the bottleneck bandwidth without dropping its rate during wireless link random losses.

## IV. MPTCP Path Schedulers

A MPTCP scheduler selects a sub-flow to inject packets into the network on a packet by packet basis. The default strategy is to select the path with shortest average round trip packet delay, hereafter called LRF. If a short and non-congested path exists between the end points, it becomes the preferred path for data transport. Non-congested path is defined as a path for which its congestion window ($cwnd$) has available space among packets yet to be acknowledged by the receiver. Hence, a congested path will have no space for more unacknowledged packets to be injected. Other path schedulers are possible, and the following are evaluated in this paper:

- **Low RTT First (LRF):** In low RTT first, the scheduler first rules out any congested path. Among the non-congested paths, the scheduler selects the path with small smooth RTT ($sRTT$). Smooth RTT is computed as an average RTT of recently transmitted packets on that sub-flow.
- **Largest Packet Credits (LPC):** In largest packet credits scheduler, this scheduler selects the sub-flow with largest available space among the non-congested paths.
- **Largest Estimated Throughput (LET):** In largest estimated throughput scheduler, the scheduler estimates

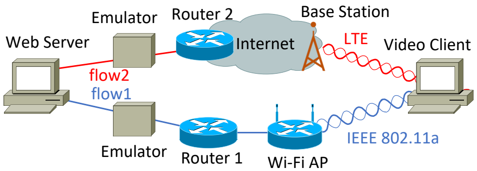


Figure 2. Video Streaming Emulation Network

TABLE I. EXPERIMENTAL NETWORK SETTINGS

| Element | Value |
|---|---|
| Video size | 113 MBytes |
| Video rate | 5.24 Mb/s |
| Playout time | 3 mins |
| Video Codec | H264 MPEG-4 AVC |
| MPTCP variants | BBR, Cubic, Compound, LIA, OLIA, BALIA |
| MPTCP schedulers | LRF, LET, LPC, GR-STY, TP-STY, TR-STY |

TABLE II. EXPERIMENTAL NETWORK SCENARIOS

| Scenario | Emulator | Path properties (RTT) |
|---|---|---|
| A1- Baseline (LTE/Wi-Fi)<br>Scenario A packet loss | LTE) delay 0 ms<br>Wi-Fi) delay 20 ms | RTT 80 ms<br>RTT 40 ms |
| A2- Large delay (LTE/Wi-Fi)<br>Scenario A packet loss | LTE) delay 0 ms<br>Wi-Fi) delay 30 ms | RTT 80 ms<br>RTT 60 ms |
| B1- Baseline (LTE/Wi-Fi)<br>Scenario B packet loss | LTE) delay 0 ms<br>Wi-Fi) delay 20 ms | RTT 80 ms<br>RTT 40 ms |
| B2- Large delay (LTE/Wi-Fi)<br>Scenario B packet loss | LTE) delay 0 ms<br>Wi-Fi) delay 30 ms | RTT 80 ms<br>RTT 60 ms |

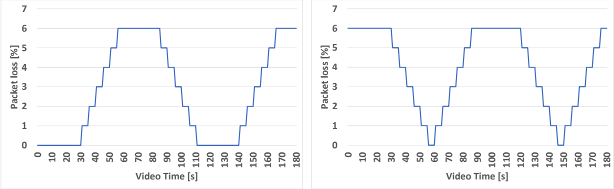
the throughput of each non-congested sub-flow, as $cwnd/sRTT$, selecting the one with largest throughput.

- **Greedy Sticky (GR-STY):** As it is the case of default scheduler (LRF), on the onset of a video streaming session, greedy sticky scheduler selects the path with smallest RTT. However, once a new path is selected, the scheduler stays on this new path for as long as there is available congestion window space, until the new path experiences congestion.
- **Throughput Sticky (TP-STY):** Similar to default scheduler (LRF), throughput sticky scheduler selects the path of lowest RTT. However, a new path is selected only if the throughput of the new path is larger than the throughput of the currently selected path.
- **Throughput RTT Sticky (TR-STY):** As with default scheduler (LRF), the path of lowest RTT is first chosen. However, in addition to requiring a larger throughput of a new candidate path as per TP-STY, an extra condition for path switch requires that a new path has smaller RTT than the currently used one.

LPC focuses on scenarios of a large RTT path has plenty of bandwidth, as compared to a shorter RTT path. LET addresses another scenario, in which a short path has plenty of bandwidth. Although the default scheduler may select this path due to its short RTT, if the short RTT has a smaller cwnd, LET will divert traffic away from this path prior to path congestion, whereas default scheduler will continue to inject traffic through it. Finally, the last three sticky schedulers attempt to reduce the number of path switches during transport session, in order to reduce head of line blocking during the streaming session.

## V. PATH SCHEDULERS OVER WI-FI & CELLULAR PATHS

Figure 2 describes the network testbed used for emulating network paths with Wi-Fi and Cellular (LTE) wireless access links. An HTTP Apache video server is connected to two L3 switches, one of which directly connected to an 802.11a router, and the other connected to an LTE base station via a cellular network card via emulator boxes. Since the bandwidth of IEEE 802.11a is sufficiently large for the bit rate of video, we have adopted 802.11a as the wireless LAN interface. In this paper, the emulator boxes are used to vary each path RTT, as well as inject controlled packet losses. The simple topology and isolated traffic allow us to better understand the impact of differential delays, packet loss, TCP variants, and path schedulers on streaming performance.



(a) Scenario A (b) Scenario B

Figure 3. Wi-Fi Packet Loss Dynamic Change Scenario

Network settings and scenarios under study are described in Tables I and II, respectively. Video settings are typical of a video stream, with video playout rate of 5.24 Mb/s, and size short enough to run multiple streaming trials within a short amount of time. Two Wi-Fi packet loss pattern scenarios are used: A and B. Scenario A represents streaming sessions camped at a reliable Wi-Fi network, with occasional packet loss ramp-ups. Scenario B represents an unreliable Wi-Fi network. Emulator boxes are tuned to generate various multiple path network conditions, and have been selected as per Table II to represent commonplace LTE/Wi-Fi streaming situations at home. TCP variants used are: Cubic, Compound, LIA, OLIA and BALIA. Performance measures are:

- **Picture discards:** number of frames discarded by the video decoder.
- **Buffer underflow:** number of buffer underflow events at video client buffer.
- **Sub-flow throughput:** the value of TCP throughput on each sub-flow.

We organize our video streaming experimental results in network scenarios summarized in Table II): A1- A Wi-Fi-Cellular (LTE) baseline with scenarioA packet loss, where Wi-Fi path of low RTT is predominantly used; A2- A Wi-Fi-Cellular scenarioA, where a slightly larger Wi-Fi path delay causes cellular path to be used; B1- A Wi-Fi-Cellular with scenarioB packet loss, where a Wi-Fi link with low delay faces a heavier loss scenario representing user situation at which device is at the end of Wi-Fi range; B2- A Wi-Fi-Cellular scenarioB packet loss, with a Wi-Fi path delay large enough to have cellular path predominantly being used;

### A. A1: Baseline Scenario A

Scenario A packet loss emulates a transition between good to poor Wi-Fi coverage, typical of handoffs between Wi-Fi and cellular when user is leaving home Wi-Fi network.

Figures 4(a) and (b) report on video streaming buffer underflow and picture discard performance when Wi-Fi delay is 20 ms. Video performance is excellent for BBR, Cubic and Compound TCP variants over all path schedulers. LIA variants perform poorly. Figures 5(a) and (b) report of Cellular and Wi-Fi total packets delivered. We can see that Wi-Fi path is most used for all TCP variants and path schedulers. Moreover, Figures 5(c) and (d) show a larger number of BBR packet retransmissions as compared with other TCP variants across all schedulers in both Wi-Fi and cellular paths.

### *B. A2: Large Wi-Fi delay on Scenario A*

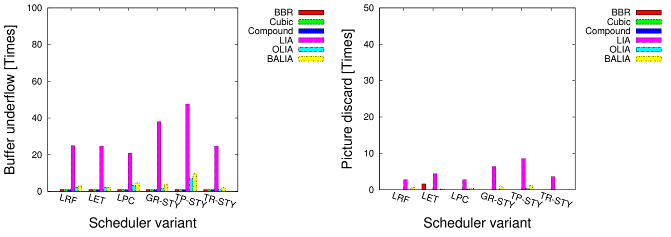
Figures 6(a) and (b) report on video streaming performance of Wi-Fi - Cellular network scenario with a large 30 ms Wi-Fi path delay. This time, BBR, and Cubic perform well with all schedulers, Compound TCP has a large buffer underflow and picture discard performance using TP-STY scheduler, similar to LIA variants. Throughput performance in Figures 7 (a) and (b) shows that TCP variants of poor video performance under TP-STY prefers Wi-Fi path to LTE path, even under large Wi-Fi path delay and packet loss. In comparison with previous delay case, we see that slowly reactive packet loss TCP variants in combination with throughput driven path selection scheduler delivers bad video streaming experience. Cellular and Wi-Fi retransmissions (c) and (d) are larger for BBR than other variants across most schedulers except TP-STY on Wi-Fi path.

### *C. B1: Baseline Scenario B*

Scenario B packet loss emulates a transition between poor to good Wi-Fi coverage when user enters the edge of a Wi-Fi network. Figures 8(a) and (b) report on video streaming performance of Wi-Fi - Cellular network scenario with a medium Wi-Fi path delay. We notice a wide variety of performances vis a vis path scheduler/TCP variant combinations. Impressive is the consistent good performance of BBR TCP variant, even across all schedulers. Throughput performance in Figures 9 (a) and (b) reveals a preference of Wi-Fi path for all TCP variants, consistent with Scenario A results presented earlier. Notice, however, that TCP variants that utilizes the most of Wi-Fi path deliver worst buffer underflow video performance, perhaps due to heavy packet losses incurred in the Wi-Fi path. Notice also that BBR maintains a better Wi-Fi utilization, striking a balance between LTE and cellular paths across all packet schedulers. Retransmission data charts, ommited for space sake, show BBR with a significantly larger number of retransmissions across all schedulers than the other TCP variants.

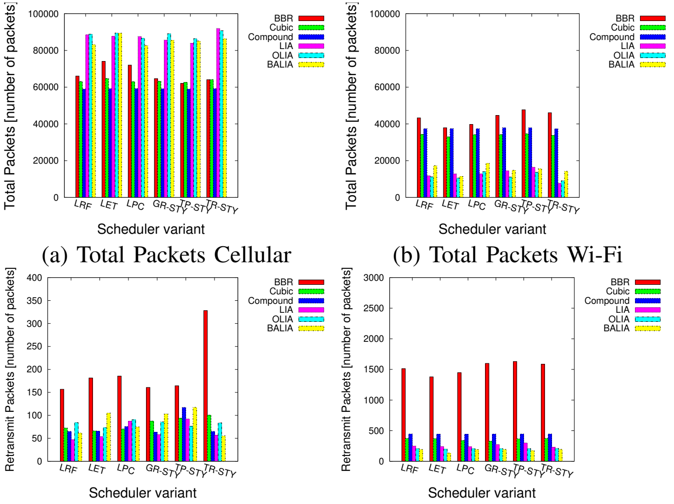
### *D. B2: Large Wi-Fi delay on Scenario B*

Figures 10(a) and (b) report on video streaming performance of Wi-Fi - Cellular network scenario with a large 30 ms Wi-Fi path delay. As in previous packet loss ScenarioA, the only TCP variants able to deliver good performance across all schedulers is BBR and Cubic. Compound TCP again presents a large buffer underflow performance using TP-STY scheduler. LIA variants all deliver large buffer underflow due to their lack of aggressiveness. Throughput performance in Figures 11 show large LTE path utilization, due to Wi-Fi large delay and heavy packet losses across all TCP variants. In comparison with previous delay case, all TCP variants back-off from Wi-Fi path, with BBR still able to use more of the Wi-Fi path across all TCP variants and on most packet scheduler cases. Packet retransmission data, again omitted for space sake, shows a larger number of retransmissions for BBR.



(a) Buffer Underflow (b) Picture Discard

Figure 4. A1- 20ms Wi-Fi delay - Video Performance



(a) Total Packets Cellular (b) Total Packets Wi-Fi

(c) Retransmit Packets Cellular (d) Retransmit Packets Wi-Fi

Figure 5. A1- 20ms Wi-Fi delay - Transmission Performance

## VI. Conclusion

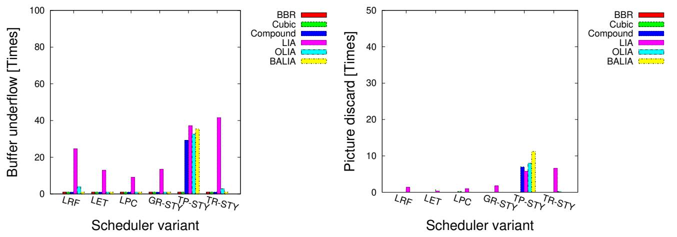
We have provided extensive performance data about path schedulers and TCP variants impact on video streaming performance over multiple paths. We have shown that some combinations of path schedulers with TCP variants negatively impact video streaming performance under certain network scenarios. Noticeable is the impressive performance of BBR across multiple network scenarios and schedulers. Moreover, we have shown video performance degradation for popular LIA and OLIA TCP variants, which can be traced to their congestion adjustment coupling slowing down their packet loss recovery. We are currently investigating the reasons for BBR consistently good streaming performance, despite its large number of retransmissions.

## Acknowledgments

Work supported by JSPS KAKENHI Grant #21H03430, and National Institute of Information and Communication Technology (NICT).

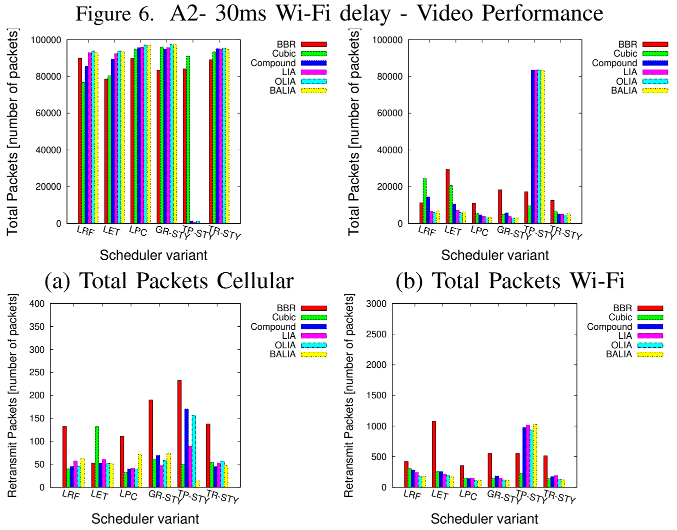
## References

[1] N. Cardwell, Y. Cheng, I. Swett, and V. Jacobson, “BBR Congetion Control,” *IETF draft-cardwell-iccrg-bbr-congestion-control-01*, November 2021.



(a) Buffer Underflow (b) Picture Discard

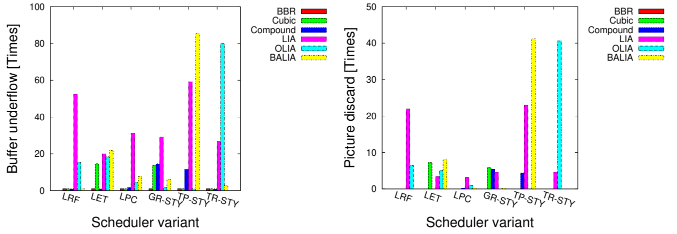
Figure 6. A2- 30ms Wi-Fi delay - Video Performance



(a) Total Packets Cellular (b) Total Packets Wi-Fi

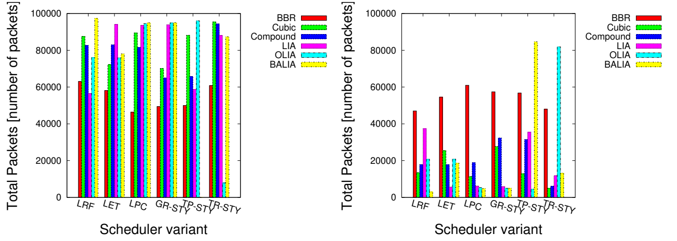
(c) Retransmit Packets Cellular (d) Retransmit Packets Wi-Fi

Figure 7. A2- 30ms Wi-Fi delay - Transmission Performance



(a) Buffer Underflow (b) Picture Discard

Figure 8. B1- 20ms Wi-Fi delay - Video Performance



(a) Total Packets Cellular (b) Total Packets Wi-Fi

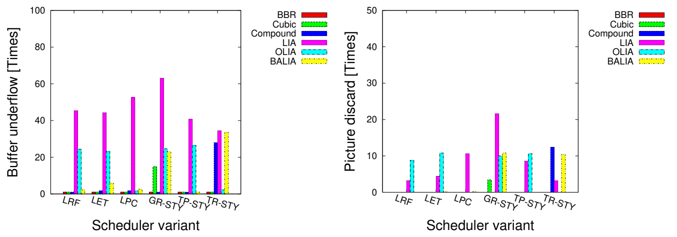
Figure 9. B1- 20ms Wi-Fi delay - Transmission Performance

[2] M. R. Palash et al., “MPWiFi: Synergizing MPTCP Based Simultaneous Multipath Access and WiFi Network Performance,” IEEE Transactions on Mobile Computing, Vol. 19, No. 1, pp. 142-158, Jan. 2020.

[3] R. Lubben and J. Morgenroth, “An Old Couple: Loss-Based Congestion Control and Minimum RTT Scheduling in MPTCP,” IEEE 44th Conference on Local Computer Networks, pp.300-307, October 2019.

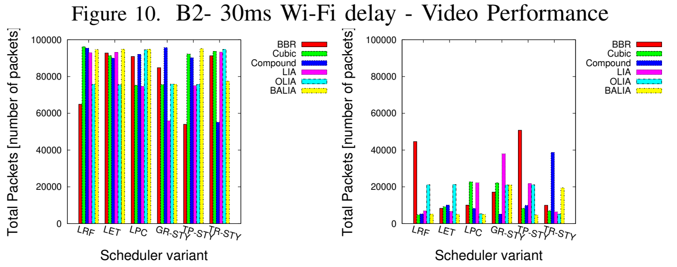
[4] A. Aliyu et al., “Interference-Aware Multipath Video Streaming in Vehicular Environments,” In IEEE Access Special Section on Towards Service-Centric Internet of Things (IoT): From Modeling to Practice, Volume 6, pp. 47610-47626, 2018.

[5] S. Afzal et al., “A Novel Scheduling Strategy for MMT-based Multipath Video Streaming,” In Proc. of IEEE Global Communications Conference - GLOBECOM, pp. 206-212, 2018.



(a) Buffer Underflow (b) Picture Discard

Figure 10. B2- 30ms Wi-Fi delay - Video Performance



(a) Total Packets Cellular (b) Total Packets Wi-Fi

Figure 11. B2- 30ms Wi-Fi delay - Transmission Performance

[6] F. Silva, D. Bogusevschi, and G-M. Muntean, “A MPTCP-based RTT-aware Packet Delivery Prioritization Algorithm in AR/VR Scenarios,” In Proc. of IEEE Intern. Wireless Communications & Mobile Computing Conference - IWCMCC 18, pp. 95-100, June 2018.

[7] S. Ferlin et al., “BLEST: Blocking Estimation-based MPTCP Scheduler for Heterogeneous Networks,” In Proc. of IFIP Networking Conference, pp. 431-439, 2016.

[8] Kimura et al., “Alternative Scheduling Decisions for Multipath TCP,” IEEE Comm. Letters, Vol. 21, No. 11, pp. 2412-2415, Nov. 2017.

[9] Xue et al., “DPSAF: Forward Prediction Based Dynamic Packet Scheduling and Adjusting With Feedback for Multipath TCP in Lossy Heterogeneous Networks,” IEEE/ACM Trans. on Vehicular Technology, Vol. 67, No. 2, pp. 1521-1534, Feb. 2018.

[10] Matsufuji et al., “Multipath TCP Packet Schedulers for Streaming Video,” IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) , August 2017, pp. 1-6.

[11] Nagayama et al., “TCP State Driven MPTCP Packet Scheduling for Streaming Video,” IARIA 10th International Conference on Evolving Internet, pp. 9-14, June 2018.

[12] Nagayama et al., “Path Switching Schedulers for MPTCP Streaming Video,” IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) , August 2019, pp. 1-6.

[13] J. Han et al., “Leveraging Coupled BBR and Adaptive Packet Scheduling to Boost MPTCP,” IEEE Transactions on Wireless Communications, Vol. 20, No. 11, pp. 7555-7567, November 2021.

[14] A. Ford et al., “Architectural Guidelines for Multipath TCP Development,” IETF RFC 6182, 2011.

[15] C. Raiciu, M. Handly, and D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols,” IETF RFC 6356, 2011.

[16] R. Khalili, N. Gast, and J-Y Le Boudec, “MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution,” IEEE/ACM Trans. on Networking, Vol. 21, No. 5, pp. 1651-1665, Aug. 2013.

[17] A. Walid et al., “Balanced Linked Adaptation Congestion Control Algorithm for MPTCP,” IETF draft-walid-mptcp-congestion-control, 2014.

[18] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control,” IETF RFC 2581, April 1999.

[19] I. Rhee, L. Xu, and S. Ha, “CUBIC for Fast Long-Distance Networks,” Internet Draft, draft-rhee-tcpm-ctcp-02, August 2008.

[20] M. Sridharan, K. Tan, D. Bansal, and D. Thaler, “Compound TCP: A New Congestion Control for High-Speed and Long Distance Networks,” Internet Draft, draft-sridharan-tcpm-ctcp-02, November 2008.

[21] D. Cavendish, K. Kumazoe, M. Tsuru, Y. Oie, and M. Gerla, “Capacity and Congestion Probing: TCP Congestion Avoidance via Path Capacity and Storage Estimation,” IEEE Second International Conference on Evolving Internet, pp. 42-48, September 2010.