Performance Evaluation of Multipath TCP Video Streaming on LEO Satellite/Cellular Networks

Yosuke Komatsu*, Dirceu Cavendish**, Daiki Nobayashi**, Takeshi Ikenaga**
*Graduate School of Engineering, **Faculty of Engineering
Kyushu Institute of Technology
Fukuoka, Japan
e-mail: komatsu.yosuke620@mail.kyutech.jp {nova@ecs, ike@ecs, cavendish@net.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 Control Protocol (MPTCP) innovation and multiple client device interfaces currently available. Recently, Low Orbit Satellite networks have become available as a way to cover remote locations where cellular coverage is spotty at best with Internet access. In this paper, we evaluate video streaming performance via cellular and LEO links. Such scenario is commonplace in geographical areas where cellular communication is unreliable, such as disaster and conflict torn situations. We provide an extensive analysis of Bottleneck Bandwidth and Round-trip propagation time (BBR) TCP variant, as well as CUBIC when transporting video streams over terrestrial cellular network (LTE) and LEO (Starlink) access networks. We use network performance level, as well video quality level metrics to characterize quality of multipath video streaming over TCP variants.**

Keywords—Video streaming; TCP congestion control; Multipath TCP; TCP BBR; LEO Satellite.

I. INTRODUCTION

Despite widespread perception that cellular network technology has become ubiquitous, large areas around the globe are still uncovered, as they are not deemed cost-effective given low population density. For such areas, global broadband coverage may be achieved via Low Earth Orbit (LEO) satellite communication. With advances in small satellite technology, several companies are deploying thousand of satellites (e.g., Starlink, OneWeb) and providing early rural broadband services in areas of spotty cellular coverage.

One aspect of satellite communication is its coexistence with cellular infrastructure. From an application standpoint, it is important to study Internet widespread applications, such as video streaming, over satellite and cellular mixed environments. In this article, we study the performance of video streaming application over satellite and cellular access links. In that context, 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 satellite/cellular handoff situations. Regarding streaming applications, video stream quality is related to two factors: the amount of data discarded at the client end point due to excessive transport delay/jitter; data rendering stalls due to lack of timely playout data. Transport delays and data starvation depend heavily on how Transport Control Protocol (TCP) handles retransmissions upon packet losses during flow and congestion control. Moreover, in multipath transport scenarios, it is important to manage head-of-line blocking across various networking paths, potentially with diverse loss and delay characteristics such as ones using cellular and satellite access links. Head-of-line blocking occurs when data already delivered at the receiver has to wait for additional packets that are blocked at another path, potentially causing incomplete or late frames to be discarded at the receiver, as well as stream rendering stalls. Transport delays and data starvation depend heavily on how TCP handles retransmissions upon packet losses during flow and congestion control. Two TCP variants are currently widely deployed: CUBIC [1], and BBR [2]. As TCP variants greatly impact streaming quality, we propose to analyze video performance vis-a-vis these widely deployed TCP variants.

The paper is organized as follows. Related work is included in Section II. Section III describes video streaming transport over TCP, with focus to BBR and CUBIC TCP variants. Section IV introduces these variants. Section V characterizes video streaming performance over Starlink and Long Term Evolution (LTE) paths via network emulation. We compare the application and network performance of BBR against CUBIC, using a default (Estimated shortest transmission time) path scheduler. Our goal is to uncover unfavorable network scenarios that may lead to the design of path schedulers appropriate to satellite/cellular multipath scenarios. 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 [3] and related work).

Recenty, some research work has focused on video streaming performance over multiple paths. In Matsufuji et al. [4], we evaluate the performance of several TCP variants and path schedulers in transporting video streams over multipaths, quantifying frame discards and play stalls. Morawski et al. [5] conduct Linux based experiments of multipath video streaming over Digital Subscriber Line (DSL) path scenarios using Linked Increase Algorithm (LIA), and Opportunistic Linked

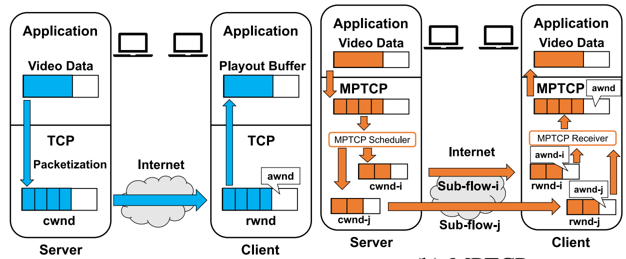
 ISBN: 978-1-68558-133-6

Increase Algorithm (OLIA), as well as Reno, CUBIC, and BBR TCP variants. They show head of line blocking as a major concern. Unfortunately, they do not provide application level performance measures, to evaluate video quality impact. Similarly, Amend et al. [6] evaluate throughput of multipath video streaming DSL multipath scenarios, without providing video level performance measures. Although they also propose a cost optimized scheduler, the lack of video quality performance measures limits conclusions about impact of such scheduler to video quality. Along the same lines, Imaduddin et al. [7] provide a performance evaluation of Multipath TCP (MPTCP) using CUBIC and Vegas TCP variants, as well as minimum Round Trip Time (RTT), round-robin and coupled Balia schedulers. Focusing on throughput performance, they conclude CUBIC to deliver best performance, regardless of the scheduler. Finally, Xing et al. [8] propose a new MPTCP scheduler which they show via network experiments to lower the number of out-of-order packets. The scheduler estimates receiver arrival times, and send redundant packets to cope with estimation errors. Video streaming is simulated via iperf3, and no application layer performance measures are used.

Regarding LEO Satellite communication, few experimental research works are available, due to recent availability of LEO Starlink beta service in some countries. B-Garcia et al. [9] presents an experimental evaluation of Starlink downlink signal acquisition over Germany. The work focuses on spectral analysis of the signal only, hence data transmission being out of scope. [10], on the other hand, presents a data transmission performance characterization of LEO Starlink UpLoad(UL) and DownLoad(DL) links, comparing them with 5G cellular UL and DL performance, when subjected to file transfer (iPerf) type of application. Although the characterization may depend on the Geo location of the satellite antennas and terminal location, the experiments show an average satellite DL throughput around 200Mb/s, as compared with 130Mb/s cellular DL speeds. They further show latency improvements when transmission is performed on both 5G and satellite link simultaneously. The same authors extended their experiments to a mobile ground terminal use case in [11], tracking cellular and satellite link availability across a rural route. Our line of research focuses on application level performance measures in addition to data/network layer performance indicators such as throughput. We focus on video streaming performance both at application as well as transport layer over cellular and LEO satellite access links. We believe that in remote areas where cellular coverage is spotty, multipath transport may provide application level reliability between cellular and satellite networks. We believe that multipath video streaming characterization over satellite/cellular networks is novel in the literature.

III. Video Streaming Over MPTCP

Video streaming over Hypertext Transfer Protocol/Transmission Control Protocol (HTTP/TCP) originates at 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). Upon an HTTP video request, a TCP sender is instantiated to transmit packetized data to the client machine, connected to the application via 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 at the network path, as well as 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 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 receiver available space, so that $cwnd \leq awnd$ condition is enforced by the sender at all times to prevent receiver buffer overflow. At the client application layer, a video player extracts data from a playout buffer, which draws packets delivered by the TCP receiver from receiver TCP socket buffer. The playout buffer hence serves to smooth out variable network throughput. Multiple path transport brings communication reliability enhancements, as well as bandwidth increase. The challenge for real time applications such as video is video rendering degradation due to increase frame discards and buffer underflows originated from head of line blocking.



(a) TCP (b) MPTCP

Figure 1. Video Streaming over TCP/MPTCP.

A. MPTCP

MPTCP is an Internet Engineering Task Force (IETF) extension of TCP transport layer protocol to support data transport over multiple concurrent TCP sessions [12]. The network multipath transmission of the transport session is hidden from application layer by a legacy TCP socket exposed per application session. At the transport layer, however, MPTCP coordinates concurrent TCP sessions on various 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 1 (b).

The first and most used path scheduler, called default scheduler, selects the path with shortest 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 BBR [2] TCP variant over uncoupled shortest RTT scheduler.

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, LTE). 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.

IV. CUBIC AND BBR TCP VARIANTS

TCP protocol nowadays has 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 [13] and CUBIC [1] 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 [14] and Capacity and Congestion Probing (CCP) [15] 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 [2]. 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.

CUBIC TCP Congestion Avoidance: TCP CUBIC is a Loss-based TCP that has achieved widespread usage as the default TCP of the Linux operating system. During congestion avoidance, its congestion window is adjusted as follows (1):

$$\begin{aligned} AckRec: \quad cwnd_{k+1} &= C(t-K)^3 + Wmax \\ K &= (Wmax\frac{\beta}{C})^{1/3} \\ PktLoss: \quad cwnd_{k+1} &= \beta cwnd_k \\ Wmax &= cwnd_k \end{aligned} \tag{1}$$

where C is a scaling factor, Wmax is the cwnd value at time of packet loss detection, and t is the elapsed time since the last packet loss detection. K parameter drives the CUBIC increase away from Wmax, whereas β tunes how quickly cwnd is reduced on packet loss. This adjustment strategy ensures that its $cwnd$ quickly recovers after a loss event.

BBR TCP Congestion Avoidance: BBR is a bandwidth delay product based TCP that has achieved widespread usage as one of available TCP variants in the Linux operating system. BBR uses measurements of a connection delivery rate and RTT to build a model that controls how fast data may be sent and the maximum amount of unacknowledged data in the pipe. Delivery rate is measured by keeping track of the number of acknowledged packets within a defined time frame. In addition, BBR uses a probing mechanism to determine the maximum delivery rate within multiple intervals.

More specifically, BBR regulates the number of inflight packets to match the bandwidth delay product of the connection, or $BDP = BtlBw \times RTprop$, where $BtlBw$ is the bottleneck bandwidth of the connection, and RTprop its propagation time, estimated as half of the connection RTT. These quantities are tracked during the lifetime of the connection, as per equations below (2):

$$\begin{aligned} RTT_t &= RTprop_t + \eta_t \\ R\hat{T}prop &= RTprop + min(\eta_t) \\ &= min(RTT_t) \forall t \in [T - W_R, T] \\ Btl\hat{B}w &= max(deliveryRate_t) \forall t \in [t - W_B, T] \end{aligned} \tag{2}$$

where η_t represents the noise of the queues along the path, W_R a running time window, of tens of seconds, and W_B a larger time window, of tens of RTTs. This adjustment strategy seeks to tune its $cwnd$ to a number of packets equivalent to the connection bandwidth delay product.

V. VIDEO STREAMING PERFORMANCE OVER STARLINK/LTE

Figure 2 describes the network testbed used for emulating network paths with Starlink and LTE wireless access links. An HTTP Nginx video server is connected to two L3 switches. In order to support multiple network scenarios, L3 switches can be directly connected to another router, at which a client is

 ISBN: 978-1-68558-133-6

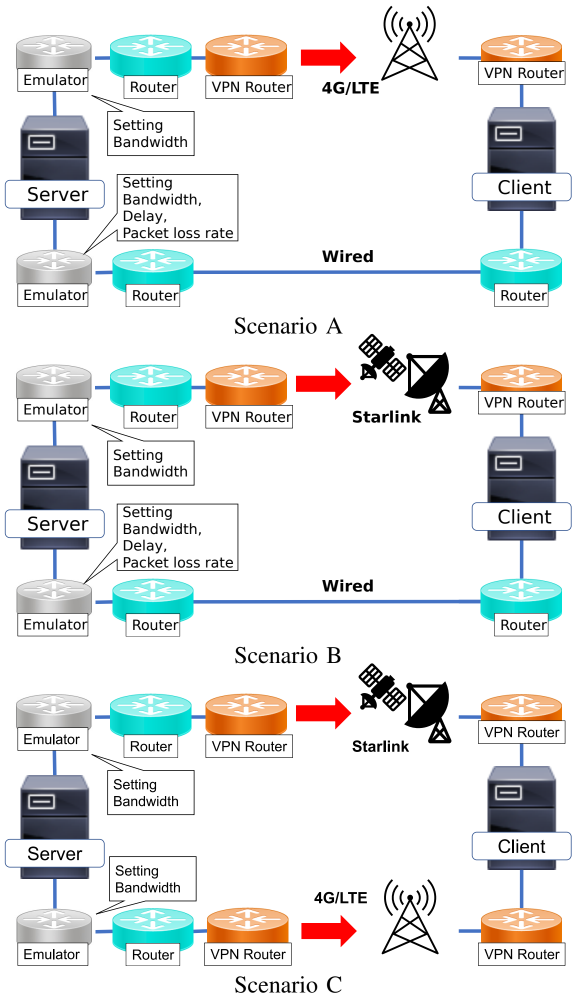


Figure 2. Experimental environment scenarios.

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
MPTCP schedulers	Default (Estimated shortest transmission time)

TABLE II. EXPERIMENTAL NETWORK SCENARIOS

Scenario	Emulator (BW, Packets Loss, Delay)
A - LTE and Wired Initial flow: LTE	Starlink : BW 3Mbps Wired : BW 3Mbps, Loss 0.5% Delay 60, 90ms
B - Starlink and Wired Initial flow: Starlink	LTE : BW 3Mbps Wired : BW 3Mbps, Loss 0.5% Delay 60, 90ms
C - Starlink and LTE Initial flow: Starlink or LTE	Starlink : BW 3Mbps LTE : BW 3Mbps

connected, or connected to an LTE base station, or connected to satellite access link. In this paper, the emulator boxes are used to vary each path RTT. The simple topology and isolated traffic allow us to better understand the impact of differential delays on TCP variant's performance.

Application and network 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. Three network scenarios are used (Figure 2). Scenario A represents dual path video streaming over wired and LTE access links. Scenario B supports dual path video streaming over wired and Starlink access links. Finally, Scenario C represents dua path video streaming over LTE and satellite access links. Emulator boxes are tuned to generate various multiple path network conditions. 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 retransmission:** TCP retransmission on each sub-flow.
- **Sub-flow cwnd:** TCP cwnd value on each sub-flow.

We organize our video streaming experimental results in network scenarios summarized in Table II): A- A LTE/wired scenario A; B- A Starlink/wired scenario B; C- A Starlink/LTE scenario C.

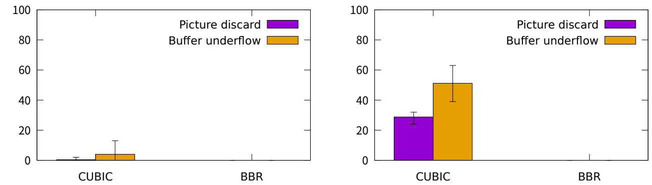
A. Cellular/Wired Scenarios

Scenario A is an experimental environment using LTE and Wired, with 3Mbit bandwidth for both paths, 0.5% packet loss rate on the wired side, and 60ms or 90ms RTT delays. Figures 3 (a) and (b) show five average video streaming frame discards / buffer underflows, and the number of packet retransmissions. Picture discard and buffer underflow were detected only when using CUBIC, with large values for delay case of 90 ms. Video streaming over BBR suffers not degradation regardless the large delays. The number of retransmissions of CUBIC seems to be much less than BBR, and for both TCP variants seem to bear little correlation with the delay values. Figures 4 (a) and (b) show CWND dynamics of a single streaming experiment using CUBIC and BBR, respectively. CUBIC seems to have a smaller CWND on wired path than in LTE path across the entire streaming, whereas BBR seems to maintain a more equitable CWND on both paths.

B. Starlink/Wired Scenarios

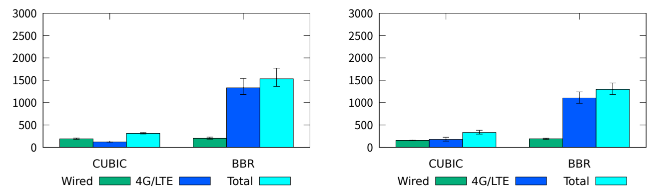
Scenario B is an experimental environment using Starlink and Wired paths, with 3Mbit bandwidth for both paths, 0.5% packet loss rate on the Wired side, and 60ms or 90ms RTT delays. Figures 5 (a) and (b) show five average video streaming frame discards / buffer underflows, and the number of packet retransmissions. Picture discard and buffer underflow were detected when using CUBIC and at large delay for BBR, with large values for CUBIC delay case of 90 ms. The number of retransmissions of CUBIC seems again to be much less than BBR, and for both TCP variants they have little correlation with the delay values. Figures 6 (a) and (b) show CWND dynamics of a single streaming experiment using CUBIC and BBR, respectively. CUBIC present a smaller CWND on wired path than in Starlink path across the entire streaming, whereas BBR seems to maintain a more equitable CWND on both paths.

 ISBN: 978-1-68558-133-6



Delay 60ms Delay 90ms

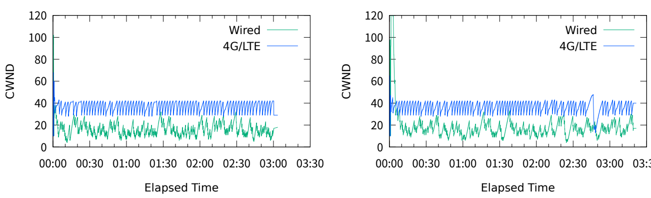
(a) Picture discard and Buffer underflow



Delay 60ms Delay 90ms

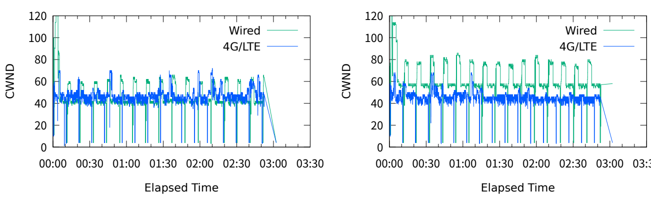
(b) Retransmission

Figure 3. Scenario A - Video Performance.



Delay 60ms Delay 90ms

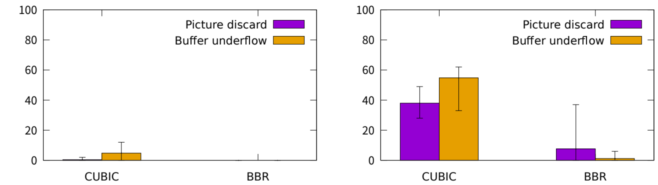
(a) CUBIC



Delay 60ms Delay 90ms

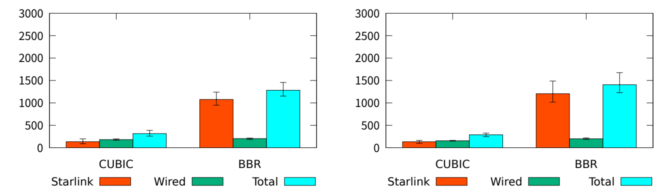
(b) BBR

Figure 4. Scenario A - CWND.



Delay 60ms Delay 90ms

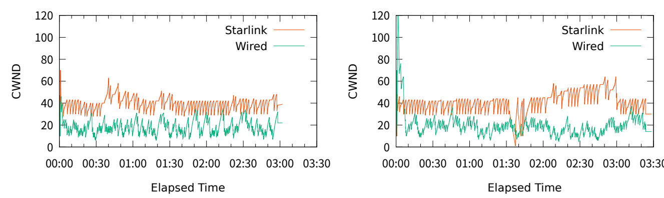
(a) Picture discard and Buffer underflow



Delay 60ms Delay 90ms

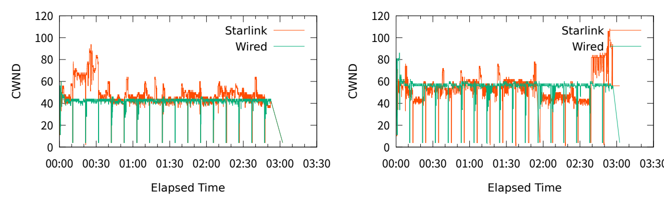
(b) Retransmission

Figure 5. Scenario B - Video Performance.



Delay 60ms Delay 90ms

(a) CUBIC



Delay 60ms Delay 90ms

(b) BBR

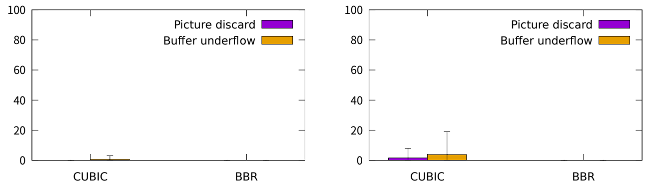
Figure 6. Scenario B - CWND.

C. Starlink/Cellular Scenario

Scenario C is an experimental environment using Starlink and LTE paths, with 3Mbit bandwidth for both paths, no additional packet loss nor delays. Figures 7 (a) and (b) show average video streaming frame discards / buffer underflows over five trials, and the number of packet retransmissions, when initial MPTCP flow is LTE or Starlink, respectively. Picture discard and buffer underflow were detected only when using CUBIC, and in small amounts. The number of retransmissions of CUBIC seems again to be much less than BBR, and for both TCP variants they have little correlation with which initial flow the streaming started with. Figures 8 (a) and (b) show CWND dynamics of a single streaming experiment using CUBIC and BBR, respectively. Both TCP variants, CUBIC and BBR, present the same CWND sizes throughout the entire video streaming session, regardless of the initial flow used. This is an indication that the TCP variants split the video traffic equitably across the two wireless paths. However, when we compare 60 msec vs 90 msec delay results, we see that CUBIC CWND size is insensitive to specific delay value, whereas BBR adjusts CWND to higher levels for higher delays.

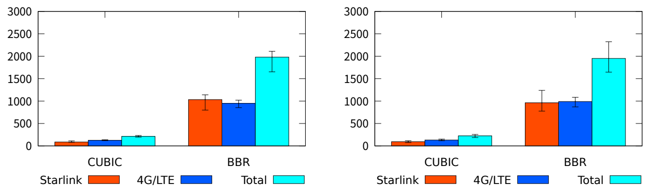
VI. Conclusion and Future Work

We have studied BBR and CUBIC transport performance of video streaming on multipath wired/LTE/Starlink mixed scenarios. From our results, we can infer that video streaming over satellite and LTE mixed environments is viable, with little degradation of streaming performance. We have detected a consistently larger levels of retransmission for BBR TCP variant as compared with CUBIC. We have also detected a bias in using more wireless paths for CUBIC TCP variant, although in LTE/Starlink mixed scenarios there was no perceived bias in path utilization for both TCP variants. All our



Initial flow: LTE Initial flow: Starlink

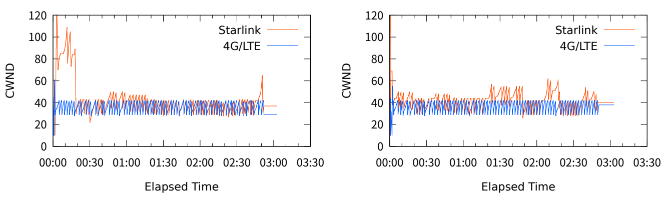
(a) Picture discard and Buffer underflow



Initial flow: LTE Initial flow: Starlink

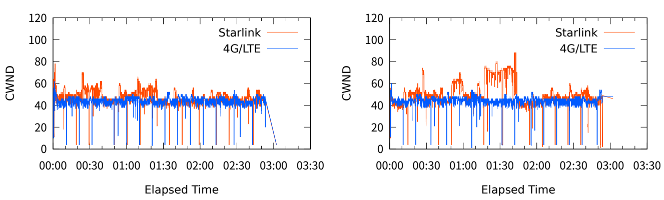
(b) Retransmission

Figure 7. Scenario C - Video Performance.



Initial flow: LTE Initial flow: Starlink

(a) CUBIC



Initial flow: LTE Initial flow: Starlink

(b) BBR

Figure 8. Scenario C - CWND.

experiments were performed with MPTCP path default scheduler (Estimated shortest transmission time). We are currently investigating whether alternate schedulers may deliver better performance at application layer, or less retransmissions at transport layer.

ACKNOWLEDGMENTS

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

REFERENCES

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

[2] N. Cardwell, Y. Cheng, I. Swett, and V. Jacobson, "BBR Congestion Control," *IETF draft-cardwell-iccrg-bbr-congestion-control-01*, November 2021.

[3] M. R. Palash and K. Chen, "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.

[4] R. Matsufuji, D. Cavendish, K. Kumazoe, D. Nobayashi, and T. Ikenaga, "Multipath TCP Packet Schedulers for Streaming Video," IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), August 2017, pp. 1-6.

[5] M. Morawski and P. Ignaciuk, "A Price to Pay for Increased Throughput in MPTCP Transmission of Video Streams," In Proc. of 24th Intern. Conference on System Theory, Control and Computing - ICSTCC, pp. 673-678, October 2020.

[6] M. Amend, V. Rakocevic, and J. Habermann, "Cost optimized multipath scheduling in 5G for Video-on-Demand traffic," In Proc. of IEEE Wireless Communications and Networking Conference - WCNC 21, pp. 1-6, March 2021.

[7] M. F. Imaduddin, A. G. Putrada, and S. A. Karimah, "Multipath TCP Scheduling Performance Analysis and Congestion Control on Video Streaming on the MPTCP Network," In Proc. of Intern. Conference on Software Engineering & Computer Systems and 4th Intern. Conference on Computational Science and Information Management - ICSECS-ICOCSIM, pp. 562-567, August 2021.

[8] Y. Xing et al., "A Low-Latency MPTCP Scheduler for Live Video Streaming in Mobile Networks," IEEE Transactions on Wireless Communications, Vol. 20, No. 11, pp. 7230-7242, Nov. 2021.

[9] R.B-Garcia et al., "Experimental Acquisition of Starlink Satellite Transmission for Passive Radar Applications," *International Conference on RADAR Systems*, Edinburgh, UK, pp. 130-135, Oct. 2022.

[10] M. Lopez, S. B. Damsgaard, I. Rodriguez, and P. Mogensen, "An Empirical Analysis of Multi-Connectivity between 5G Terrestrial and LEO Satellite Networks," *Proceedings of IEEE Globecom Workshop on Cellular UAV and Satellite Commun.*, pp. 1115-1119, Rio De Janeiro, Brazil, pp. 1115-1120, Dec. 2022.

[11] M. Lopez, S. B. Damsgaard, I. Rodriguez, and P. Mogensen, "Connecting Rural Areas: An Empirical Assessment of 5G Terrestrial-LEO Satellite Multi-Connectivity," *IEEE 97h Vehicular Technology Conference*, Florence, Italy, pp. 1-5, Jun. 2023.

[12] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural Guidelines for Multipath TCP Development," IETF RFC 6182, 2011.

[13] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," IETF RFC 2581, April 1999.

[14] 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.

[15] 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.