

Is two greater than one?: Analyzing Multipath TCP over Dual-LTE in the Wild

Nitinder Mohan
University of Helsinki
Finland

Tanya Shreedhar
IIIT-Delhi
India

Aleksandr Zavodovski
University of Helsinki
Finland

Jussi Kangasharju
University of Helsinki
Finland

Sanjit K. Kaul
IIIT-Delhi
India

ABSTRACT

Multipath TCP (MPTCP) is a standardized TCP extension which allows end-hosts to simultaneously exploit all of their network interfaces. The recent proliferation of dual-SIM mobile phones makes multi-LTE MPTCP setup an attractive option. We perform extensive measurements of MPTCP over two LTE connections in low and high-speed mobility scenarios over five months, both in controlled and in-the-wild environments. Our findings indicate that MPTCP performance decreases at high speeds due to increased frequency of signal strength drops and handovers. Both LTE paths experience frequent changes which result in a sub-optimal subflow utilization. We also find that while path changes are unpredictable, their impact on MPTCP follows a deterministic trend. Finally, we show that both application traffic patterns and congestion control variants impact MPTCP adaptability at high speeds.

1 INTRODUCTION

According to CISCO, global mobile data traffic has grown *17-fold* between 2012-2017 and 71% in a single year [3]. LTE is widely deployed [20] and future technologies, such as 5G, strengthen the need for efficient protocols over cellular links. Studies show that TCP performs poorly over LTE, especially when the user is mobile, primarily due to the large variability in network conditions [10, 18]. This deterioration is further heightened due to presence of large buffers within the ISP network which often results in TCP connection stalls and ineffective link utilization due to excessive queueing [10, 12].

Multipath TCP (MPTCP) is a TCP extension allowing unmodified applications to leverage multiple network interfaces to form parallel TCP connections between end-hosts [22]. Researchers have studied and analyzed its impact on emerging technologies (AR/VR [9], IoT [17], edge computing [16] etc.) while exploiting multiple network types such as WiFi, cellular and ethernet [4]. MPTCP kernel is available as open-source and is in use, e.g., by Apple in their iOS devices [1]. Despite these efforts, MPTCP faces several challenges in mobile networks, such as failing to use heterogeneous network

combinations with large delay differences, e.g. WiFi and LTE [5]. However, we believe that MPTCP's most pragmatic use is in multi-LTE networks for two reasons. *First*, modern smartphones are often equipped with dual-SIM slots and chipsets enabling the use of two LTE connections. [8]. *Second*, vast coverage areas, large combined bandwidth and reliable packet delivery offered by multi-LTE connections can enable effective deployment of emerging technologies discussed above. Also, intuitively, none of MPTCP heterogeneity issues should affect a multi-LTE setup as both paths exhibit similar characteristics with similar delays to the server.

In this paper, we conduct to our knowledge the first comprehensive measurement study of MPTCP over multi-carrier LTE connections in *day-to-day mobility* scenarios. Over a period of *five months*, we performed extensive data collection in controlled environments and in-the-wild to understand the impact of last-mile quality, mobility and application workload on MPTCP performance. Our key findings are:

- (1) MPTCP bandwidth gains over two LTE connections decrease significantly with increasing mobility. Downloads over MPTCP take over twice as long at speeds >60km/h, performing even worse than a single TCP most of the time.
- (2) Frequent signal strength drops and handovers are the cause of MPTCP deterioration at high speeds. Such changes can induce 10-fold RTT spikes on a subflow resulting in severely increased queuing and reordering delays.
- (3) Contrary to our intuitions, we find that MPTCP exhibits a skewed subflow utilization as both LTE paths experiences last-mile changes at varying time and frequency which often overlaps with each other.
- (4) Although LTE last-mile changes are unpredictable at high speeds, their impact on MPTCP follows a deterministic trend. This allows monitoring occurrences of link changes and circumventing their effects to maintain MPTCP's benefit over multiple connections.
- (5) Tweaking application traffic burstiness and congestion control schemes can also assist the protocol in its adaptability and robustness at high speeds. Our analysis shows an improvement of 75% QoE and 18% throughput in MPTCP.

2 BACKGROUND & RELATED WORK

MPTCP is in-kernel extension to TCP that allow multi-homed hosts to use multiple parallel TCP connections over each network interface. Data packets from the application are scheduled to one of the underlying TCP subflows by *scheduler* [26]. The default *minSRTT* scheduler [21] prioritizes the subflow with lowest smoothed round trip time (SRTT) to the receiver. In variable delay networks, the scheduler is known to produce out-of-order packets at the receiver as they experience different delays on parallel paths [27]. Such packets are buffered at the receiver and are only delivered "in-order" to the application. MPTCP minimizes additional reordering delays by employing *coupled congestion* mechanisms at the sender which balance packet congestion over all subflows [24].

Previous studies have focused on analyzing MPTCP over heterogeneous networks such as WiFi and cellular. The key takeaway is that MPTCP performance is limited due to consistent delay difference between both paths resulting in ineffective utilization [4, 19]. Little attention has been paid to MPTCP in multi-LTE networks. LTE connections are resilient to packet losses and offer high bandwidth aggregation opportunity for MPTCP [10]. However, studies analyzing TCP over LTE have shown that with increasing mobility, TCP performance degrades due to delays in connection establishment, timeouts and interruptions [18, 25]. At high speeds, TCP packets experience excessive on-device and in-network queuing along with packet losses [15, 28]. Therefore, it is pertinent to understand whether MPTCP provides any benefits to mobile clients while utilizing multiple LTE paths.

Study by Li et al. [14] is closest to ours. While the authors analyze MPTCP using dual-LTE in very high-speed rails ($>250\text{km/h}$), our work focuses on understanding MPTCP's behavior in day-to-day mobility using generic transportation modes ($<100\text{km/h}$), such as walking, driving, etc. Furthermore, the focal point of authors work in [14] was to study LTE handover's impact on MPTCP performance. On the other hand, we investigate the correlation between any/all last-mile link changes (signal drops, handovers) and other network parameters (app traffic, congestion control) on MPTCP.

3 MEASUREMENT & ANALYSIS METHOD

Our aim is to *analyze MPTCP performance over two distinct LTE connections in different mobility*. We elected to use a device equipped with two separate LTE antennas. Current dual-SIM enabled smartphones are fitted with a single antenna which is time-shared by both connections [8]. Our aim in using two antennas is to establish the upper bound on MPTCP performance and not be limited, e.g., by NIC queuing due to resource contention. We also wanted to use multiple ISPs since the internal network configuration, such

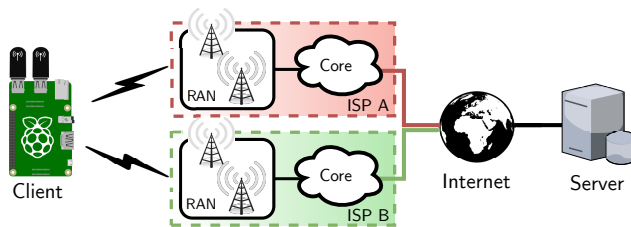


Figure 1: Data collection setup. Client is MPTCP-capable RPi equipped with two LTE connections from different ISPs. Server is an AWS ec2 instance.

as scheduling, routing, priority queue etc., can differ for each ISP depending on its QoS promises.

3.1 Setup Configuration

We designed our measurement setup as shown in Figure 1. We conducted our experiments in the capital region of a European country from *September 2018 to February 2019*. Our test device is a Raspberry Pi 2 (RPi) equipped with two Telewell CAT4 LTE USB modems. The RPi runs Raspbian OS over latest MPTCP v0.94 [22] and is powered by an external battery to enable mobility. We equip USB modems with LTE connections from *two* major cellular providers, *ISP A* and *ISP B*; capped to 150 Mbps downlink and 50 Mbps uplink. Both ISP's offer extensive network coverage in the region with *almost equal* LTE basestation (BS) deployment density. As the server, we set up AWS ec2 instances running MPTCP v0.94 on 32 GB RAM, 1 Gbps ethernet, 16-core 2.4GHz CPU and Ubuntu 18.04. Both cellular providers have *non-intersecting* network path to AWS, which we verify by periodically running traceroute over both connections. By default, both client and server use *minSRTT* scheduler and CUBIC congestion control [7]. In stable conditions, we observe ≈ 54 ms RTT to the server over both ISP connections.

3.2 Data Collection

We built a BASH-based **data collector** for RPi which performs the following tasks: *network measurements and mobility detection, data transfer and upload to measurement server*.

Network Measurements. Every second (defined as "period") the data collector queries attached LTE modems with AT commands and logs their current signal strength (dBm) and associated basestation ID (*BSID*). As we wish to analyze the impact of mobility on MPTCP, it is necessary to detect when the RPi is mobile. However, unlike smartphones, the RPi cannot infer mobility using conventional techniques due to lack of a built-in accelerometer or GPS. We overcame this limitation by constantly monitoring changes in signal strength (δ_{signal}) over both modems. The collector flags $\delta_{signal} \geq 6\text{dBm}$ on either modem as "mobility start" and initiates data transfer. We found 6dBm to be the best fit

	MPTCP Conns.	TCP Conns.	File DLs	Video plays
Uncontrolled	10863	7966	7902	1793
Controlled	1345	937	825	192

Table 1: Measurement statistics.

Mobility	Rate (Mbps)		RTT (ms)		DL Time (s)	
	MPTCP	TCP	MPTCP	TCP	MPTCP	TCP
Stationary	69.2	42.1	56	57	36.6	71.4
Walking	42.5	34.8	65	68	61.2	78.1
Driving	31.9	31.7	74	75	81.5	82.3

Table 2: Overview of controlled measurement results.

for mobility threshold as it achieves the maximum accuracy (98.3%) with zero false negatives in contrast to other values in our controlled experiments (§4.1). Also, we denote change in BSID between consecutive periods as successful handover.

Data Transfer & Measurement Upload. We developed a client-side application which downloads a 300MB¹ file from the server over HTTP.² We also conducted DASH video streaming measurements over MPTCP to provide further granularity to our analysis (§5.1). During data transfer, the collector passively records packet traces via `tcpdump` which we use to analyze several MPTCP/TCP metrics such as RTT, bytes-in-flight, out-of-order queue size etc. The collected data in the RPi is uploaded to our measurement server every night if δ_{signal} remains zero for two consecutive hours.

We gave our test devices to three volunteers to carry along their daily commute, encompassing walking, driving and public transport such as trams, buses etc. (see §4.2). We did not record any real-world information such as distance traveled, locations, mode of transport etc. We also conducted multiple measurements in controlled environments where we chart out a planned test route for different mobility speeds (see §4.1). Controlled experiments serve us two purposes; (1) developing classification models over network changes for grouping our in-the-wild traces into different mobility categories; and, (2) closer inspection of MPTCP behavior. Table 1 shows our measurement statistics.

4 IMPACT OF LTE MOBILITY ON MPTCP

We first analyze MPTCP’s performance in controlled mobility tests and provide further explanation to trends in its behavior by scrutinizing the data collected in-the-wild.

4.1 Controlled Mobility Measurements

We conducted multiple measurements for two common mobility types, *walking* and *driving*, using a fixed route for both modes. Our driving route was 13km long inter-city highway

¹We opt for a longer flow as MPTCP is known to perform badly for short-lived flows due to slow congestion window growth [19].

²We leave measurement of uplink performance as future work.

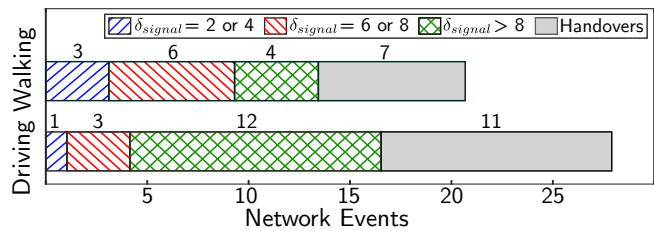


Figure 2: Network event frequency every five minutes in dual-LTE in controlled experiments.

while our walking path was 4.7km encircling the city center. We maintain an average speed of $\approx 6\text{km/h}$ and $\approx 80\text{km/h}$ while walking and driving, respectively. The RPi was placed near the vehicle’s windshield to avoid any signal shielding by the chassis. While we regulate the speed in our experiments, we have no control over the underlying network environment. We eliminated any outlier bias by performing multiple iterations of each experiment at different times of the day. We also performed *baseline experiments* where the RPi was kept stationary throughout the data transfer.

Table 2 compares the average throughput, RTT and file download time obtained by MPTCP and TCP over LTE from each ISP. Both MPTCP and TCP show a decline in performance with increasing speeds. Although the RTTs for both protocols show a similar degrading trend, the impact of high speed on throughput and file downloads is more significant in MPTCP. While TCP exhibits 10% reduction in throughput and download time, MPTCP throughput decreases by 37% and downloads take *twice* as long while driving. Interestingly, MPTCP’s bandwidth gains over two LTE connections lessen at higher speeds as it achieves similar throughput as TCP. We investigate the root cause of this degradation by dissecting our collected traces.

LTE link changes with mobility. As the mobility only changes the last-mile link (USB modem \leftrightarrow LTE basestation), the first question we answer is, *what is the impact of client mobility on last-mile LTE?* An LTE connection is likely to experience both changes in signal strength and handovers while the client moves closer/away from the BS. Our analysis shows that only *drops in signal strength* (δ_{signal}) and *handovers*³ have an impact on MPTCP throughput. We quantized occurrences of all network events on both LTE connections while driving and walking. Figure 2 shows the frequency of different network events every five minutes on both connections. We observed a 35% increase in overall events along with a 57% increase in handovers while driving compared to walking. Furthermore, while drops $\leq 8\text{dBm}$ are more frequent for a walking client (69%), signal drops $>8\text{dBm}$ predominate network events observed while driving (75%). We

³Collectively referred as *network events* throughout the paper.

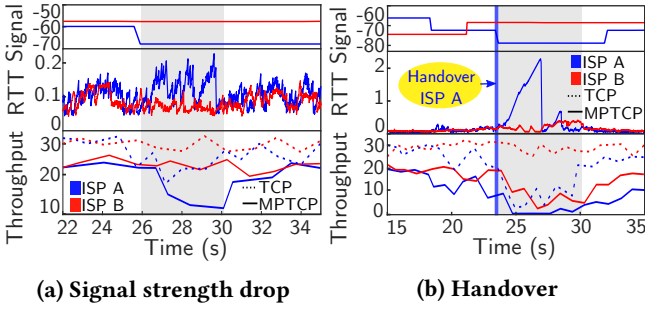


Figure 3: Effect of network events on throughput (Mbps), RTT (s) of MPTCP and parallel TCP. Shaded region denotes RTT recovery time (t_R).

also find that 43% and 52% handovers follow a signal drop while walking and driving, respectively.

Takeaway1: *Both LTE connections experience increased frequency of network events at the last-mile with increasing mobility. While low-intensity signal drops are more probable for slow-moving clients, the rate of handovers and large signal drops increases for clients moving at higher speeds.*

Impact of network events. We now examine the effect of network events on MPTCP performance. Figure 3 shows a snippet of throughput over MPTCP and simultaneous TCP over each ISP. *ISP A* connection observes 8dBm signal drop (Figure 3a) and handover (Figure 3b). As RTT and signal strength of TCP flows follow a similar behavior as MPTCP, we exclude them from our graph to maintain legibility.

We observe from Figure 3a that the drop in signal strength on *ISP A* is immediately followed by $2\times$ RTT and results in 22% decrease in overall throughput. This result agrees with previous studies which attributes the spike in RTT to increased buffering at the BS as it switches to a lower link rate to accommodate for the drop in client’s signal strength [12]. However, increased RTT impacts MPTCP performance more severely compared to regular TCP (38%↓ vs. 22%↓). This is due to the behavior of minSRTT scheduler. As *ISP A* connection observes elevated RTTs, the scheduler opts to send subsequent packets on *ISP B* to avoid out-of-order deliveries at the receiver. This continues until the RTT of *ISP A* improves as the BS flushes out queued packets to the client. We denote time taken by RTT of affected subflow to recover as t_R , shown as the shaded portion of the graph. As LTE can experience multiple signal drops while the client moves away from the BS, we find t_R to be as large as 7s and 12s while walking and driving, respectively.

On the other hand, RTT spikes induced by handovers on the subflow surpass those caused by signal drops by a large margin and result in $8\times$ RTT difference between subflows (Figure 3b). Furthermore, while signal drops cause throughput decline only on the affected subflow, handovers impact the performance of both paths and result in 74% throughput

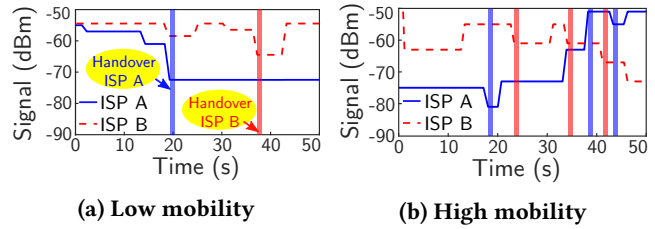


Figure 4: Network quality signatures for classifying data captured in in-the-wild into mobility categories.

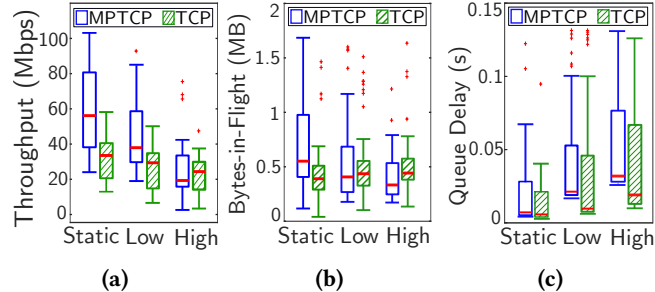


Figure 5: Comparison between MPTCP and TCP in (a) throughput achieved, (b) bytes-in-flight and (c) queuing delays for different mobility classes.

decrease. This trend is absent for TCP flows. While driving we find that consecutive handovers and signal drops on both links can result in $t_R \geq 40s!$ Presence of such large delay differences between subflows closely resembles MPTCP behavior over heterogeneous networks [19]. We explain MPTCP’s response to handovers in our in-the-wild analysis.

Takeaway2: *Last-mile changes have a direct correlation to MPTCP performance degradation. Following Takeaway1, both subflows observe spikes in RTT as response to network events, resembling a heterogenous network-like behavior. However, existing MPTCP solutions are not applicable in dual-LTE due to the lack of a consistently "better" path.*

4.2 In-The-Wild Measurements

Before we analyze MPTCP behavior in-the-wild, we first need to accurately classify the collected data traces into different mobility categories based on volunteer’s speed at collection time. Using our observations in §4.1, we design a **classification model** which labels data traces as *low-speed* and *high-speed* depending on the frequency of observed network events. The low-speed category represents traces collected while walking and high-speed constitutes all motorized transport modes. Figure 4 shows a representation from both categories. Our controlled tests revealed that this model achieves 98.7% classification accuracy.

How "Multi-Path" is MPTCP? We analyze data traces for trends in network parameters with increasing mobility. Figure 5a shows throughput achieved by MPTCP and TCP over

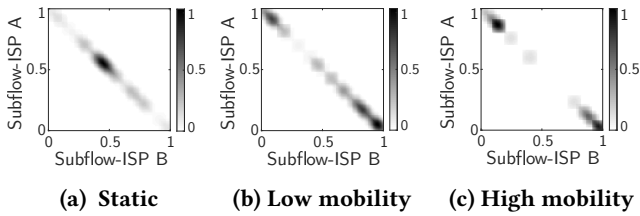


Figure 6: MPTCP subflow utilization with increasing mobility. Heat near corners denote skewed usage.

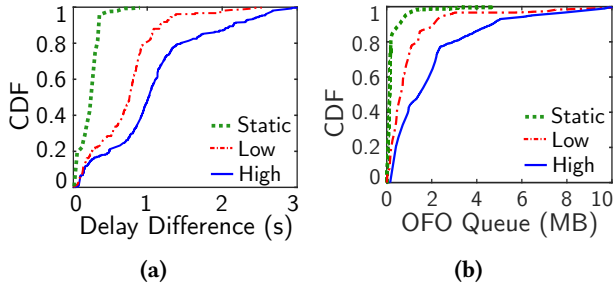


Figure 7: Variation of (a) RTT difference between subflows and (a) out-of-order queue size at receiver.

each ISP as a bar plot. The results for static and low mobility fall in line with our observations in controlled measurements (Table 2). For high mobility, we find that MPTCP over two LTE connections achieves even lower throughput than TCP over single LTE. The result is intriguing as it directly contradicts the basic design goal of MPTCP to at least perform as-good-as TCP over better path [6]. Figure 6 provides us with more insight. For a static client, MPTCP uses both subflows almost equally (heat concentrated in the center of the diagonal) resulting in $1.6\times$ throughput compared to TCP. However, with increasing speeds, the utilization skews towards one of the subflows which triggers 65% drop in throughput for low-speed mobility. In high-speed transportation modes, such as metro, trains etc., MPTCP limits its use to single LTE connection; that too inefficiently as evident by the declining trend in the average number of in-flight packets which seems to be absent for TCP (see Figure 5b). As discussed in §4.1, the degradation is caused due to minSRTT scheduler’s response to RTT spikes induced by network events. To find the root cause of such spikes, we plot the distribution of *queuing delay* endured by MPTCP packets in Figure 5c (calculated as instantaneous delay in excess of minimum RTT throughout transfer). Our suspicion of bufferbloat at the basestation is proven correct as both MPTCP and TCP experience increasing delays with changes on last-mile. 75% MPTCP packets at high speeds experience queuing delays as large as $1.7\times$ end-to-end RTT compared to a stationary receiver.

Dissecting inner-workings of MPTCP. We now investigate the consequences of network events on MPTCP decision-making and provide reasons for its behavior to handovers. To

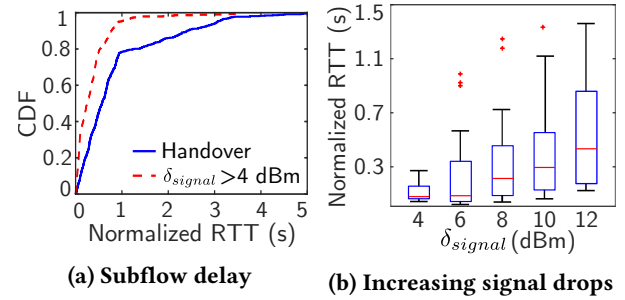


Figure 8: Impact of network events on subflow RTT.

this end, we first examine the impact of induced RTT spikes. Figure 7a shows the distribution of RTT difference between subflows for different mobility. 75% MPTCP connections observe $>1\text{s}$ delay gap between both subflows at higher speeds. The primary reason for increasing delays is growing occurrences of network events on alternate LTE links which keeps the underlying network continually unstable. Considering Figure 7b, its impact on MPTCP becomes apparent with increased out-of-order buffer occupancy at the receiver. Larger the occupancy, longer a packet waits in the buffer before being delivered to the application in-ordered sequence. At high speeds, the out-of-order queue size increases to accommodate packets experiencing considerable delay differences on both paths, until reaching its maximum capacity. At this stage, the receiver cannot allow for any more packets due to *buffer stalling*. This, along with reordering, is the cause for throughput drop on all subflows witnessed in §4.1.

Takeaway3: *Frequent LTE link changes induce large delay differences between subflows which results in unequal subflow utilization, re-ordering delays and buffer stalling, to the extent that single TCP outperforms MPTCP at high speeds.*

Investigating subflow behavior. We now explore solutions for improving MPTCP adaptability at high speeds in multi-LTE networks. We begin with investigating the impact of network events on MPTCP subflow, as any trends in subflow performance can be leveraged by MPTCP scheduling policy. Frequent network events at high speeds, often overlapping, makes this analysis challenging. We carefully separated data traces which had sufficient gaps between consecutive network events. We calculated *normalized RTT* (instantaneous RTT recorded more than initial RTT at connection establishment) of the affected subflow for associated t_R . This allows us to identify any spikes in RTT on a subflow post a network event. Figure 8a shows the distribution of normalized RTT in effect of handover and signal drop on MPTCP subflow. This distribution validates our results in controlled measurements (§4.1) as it shows $3\times$ and $10\times$ RTT spikes (compared to average RTT) on subflow experiencing signal drop and handover respectively. We further dissect the signal drop distribution to analyze the effect of different

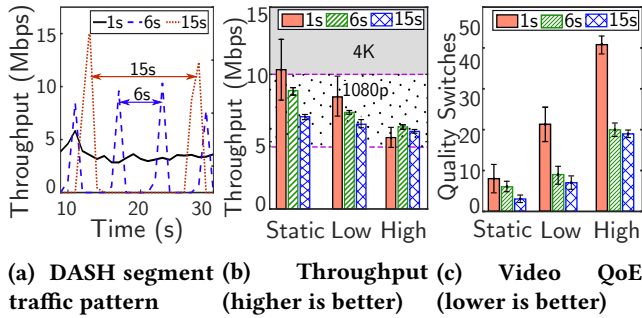


Figure 9: Video streaming over dual-LTE MPTCP.

drop levels on subflow’s RTT. Interestingly, we see a linearly increasing trend emerge indicating that larger signal drops result in higher and longer RTT spikes. On closer analysis, we find that the trend is deterministic, i.e. *for every 2dB increase in the signal drop, affected subflow observes 1.7-fold RTT spike*. The result is quite encouraging and suggests that although link changes on last-mile are unpredictable in LTE, their impact on MPTCP can be accurately predicted.

Takeaway4: *A well-designed, cross-layer MPTCP scheduler, one which actively monitors occurrences of network events on last-mile can assist in providing robustness and adaptability over multiple LTE connections.*

5 DISCUSSION

5.1 Impact of Application Traffic

We also investigated the impact of mobility on an application’s QoE by analyzing performance of video streaming over MPTCP for two reasons. *First*, video streaming accounts for the largest share of mobile traffic in the network [3]. *Second*, different DASH segment sizes allow us to simulate varying application traffic. We set up a DASH server in our AWS instance and host a 10 minute long *Big Buck Bunny* video on it [2]. The video was encoded in resolutions ranging from 240p to 4K for bitrates from 50Kbps to 15Mbps. We manually throttled both LTE connections to 8Mbps each (total 16Mbps) to remove excess bandwidth. We re-encoded each resolution into three segments ordered by *increasing burstiness*; 1, 6 and 15 seconds (Figure 9a). The VLC video player in RPi downloads segment sizes which can be best supported by available network capacity. Overall, we analyzed ≈ 2000 traces categorized into three mobility groups.

Figure 9b shows achieved throughput. Shaded regions denote required throughput for maintaining 4K and 1080p quality. Interestingly, we find that MPTCP performance in mobility differs for different traffic patterns. While only constant traffic (1s segment) can support 4K in the static category, it is also affected the worst by high speed and barely achieves 1080p (49%↓ throughput). The impact of mobility on its QoE is also substantial as the number of video quality

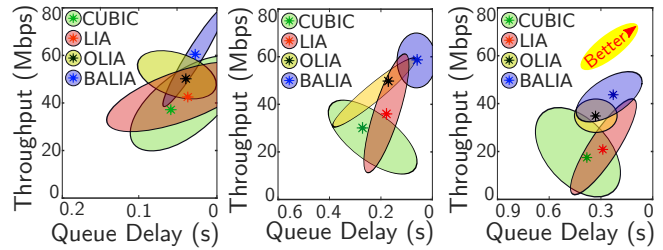


Figure 10: Performance of MPTCP congestion control algorithms in (a) static, (b) low and (c) high mobility.

switches while streaming exceeds other segments by 75% (see Figure 9c). On the other hand, both 6s and 15s streams perform poorly while the client is static, primarily due to limited growth of congestion window size which restricts full bandwidth utilization by bursty traffic. However, both segments maintain a consistent 1080p at higher speeds along with minimal quality switches (average 1.7/minute). The reason for increased adaptability is a timing mismatch between last-mile changes and application traffic bursts.

5.2 Effect of Congestion Control

We also explored the impact of different congestion control schemes on cellular mobility. We conducted a study where we switch the congestion control algorithm in our server and RPi to MPTCP coupled variants, i.e., LIA [23], OLIA [13] and BALIA [11], during our in-the-wild measurements. We collected ≈ 400 download traces for each scheme and compared it with default uncoupled CUBIC (§4.2). Figure 10 presents our results as throughput-delay graphs with flipped x-axis. The graph shows $1-\sigma$ ellipses for Gaussian distribution of the points. An ellipse’s orientation signifies covariance between the two axis and asterisks denote median values. Protocols on the “top-right” are the best on such plots.

Coupled schemes out-perform CUBIC while remaining throughput-fair (narrower towards y-axis) with increasing mobility. Unlike CUBIC, coupled algorithms balance congestion over all subflows and the difference between each variant only lies in their additive increase phase. At higher speeds BALIA outperforms other available flavors, achieving 18% increase in throughput and 17% decrease in queue delay. OLIA closely follows it and displays similar behavior. While we report our observations in this work, a detailed analysis of this behavior is left as future work.

Takeaway5: *Our results show that both application traffic shape and congestion control flavor impacts MPTCP’s ability to adapt to last-mile changes, and should be further explored.*

6 CONCLUSION

This paper studied MPTCP behavior over multi-carrier LTE networks in day-to-day mobility scenarios. Following our

extensive data collection over five months, we observed that MPTCP throughput is severely affected with increasing speeds, often performing even worse than a single TCP connection. This is primarily due to frequent last-mile changes on both LTE connections, including signal strength drops and handovers, which result in significant delay differences between MPTCP subflows. At high speeds, MPTCP struggled to recover from increased out-of-order transmissions and exhibited a skewed utilization. We found that effective solutions are possible as the impact of link changes follow a deterministic trend. With a better choice of the application traffic pattern and congestion control, MPTCP showed an improvement of 75% QoE & 18% throughput.

REFERENCES

- [1] Apple Inc. 2017. Use Multipath TCP to create backup connections for iOS. "https://support.apple.com/en-us/HT201373,2017".
- [2] Blender. 2019. Big Buck Bunny. peach.blender.org.
- [3] CISCO. 2017. *Cisco Visual Networking Index: Global mobile data traffic forecast update, 2016 to 2021*. Technical Report.
- [4] Shuo Deng, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. 2014. WiFi, LTE, or Both?: Measuring Multi-Homed Wireless Internet Performance. In *Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14)*. ACM, New York, NY, USA, 181–194. <https://doi.org/10.1145/2663716.2663727>
- [5] S. Ferlin, A. Alay, O. Mehani, and R. Boreli. 2016. BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*. 431–439. <https://doi.org/10.1109/IFIPNetworking.2016.7497206>
- [6] Alan Ford, Costin Raiciu, Mark Handley, Sebastien Barre, and Janardhan Iyengar. 2011. Architectural Guidelines for Multipath TCP Development. RFC 6182. <https://rfc-editor.org/rfc/rfc6182.txt>
- [7] Sangtae Ha, Injong Rhee, and Lisong Xu. 2008. CUBIC: A New TCP-friendly High-speed TCP Variant. *SIGOPS Oper. Syst. Rev.* 42, 5 (July 2008), 64–74. <https://doi.org/10.1145/1400097.1400105>
- [8] Ling Hang and Chih-Ping Hsu. 2017. Simultaneous voice and data for dual-sim-dual-standby (DSDS) wireless device. US Patent App. 15/855,652.
- [9] B. Hayes, Y. Chang, and G. Riley. 2018. Controlled Unfair Adaptive 360 VR Video Delivery over an MPTCP/QUIC Architecture. In *2018 IEEE International Conference on Communications (ICC)*. 1–6. <https://doi.org/10.1109/ICC.2018.8422861>
- [10] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2013. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. *SIGCOMM Comput. Commun. Rev.* 43, 4 (Aug. 2013), 363–374. <https://doi.org/10.1145/2534169.2486006>
- [11] Jaehyun Hwang, Steven H. Low, Anwar Walid, and Qiuyu Peng. 2016. Balanced Linked Adaptation Congestion Control Algorithm for MPTCP. *IETF*.
- [12] Haiqing Jiang, Yaogong Wang, Kyunghan Lee, and Injong Rhee. 2012. Tackling Bufferbloat in 3G/4G Networks. In *Proceedings of the 2012 Internet Measurement Conference (IMC '12)*. ACM, New York, NY, USA, 329–342. <https://doi.org/10.1145/2398776.2398810>
- [13] Ramin Khalili, Nicolas Gast, Miroslav Popovic, and Jean-Yves Le Boudec. 2014. *Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP*. Internet-Draft draft-khalili-mptcp-congestion-control-05. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/draft-khalili-mptcp-congestion-control-05>
- [14] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. 2018. A Measurement Study on Multi-path TCP with Multiple Cellular Carriers on High Speed Rails. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. ACM, New York, NY, USA, 161–175. <https://doi.org/10.1145/3230543.3230556>
- [15] L. Li, K. Xu, D. Wang, C. Peng, K. Zheng, R. Mijumbi, and Q. Xiao. 2017. A Longitudinal Measurement Study of TCP Performance and Behavior in 3G/4G Networks Over High Speed Rails. *IEEE/ACM Transactions on Networking* 25, 4 (Aug 2017), 2195–2208. <https://doi.org/10.1109/TNET.2017.2689824>
- [16] Nitinder Mohan, Tanya Shreedhar, Aleksandr Zavodavoski, Otto Waltari, Jussi Kangasharju, and Sanjit K. Kaul. 2018. Poster: Redesigning MPTCP for Edge Clouds. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. ACM, New York, NY, USA, 675–677. <https://doi.org/10.1145/3241539.3267738>
- [17] Michał Morawski and Przemysław Ignaciuk. 2017. On implementation of energy-aware MPTCP scheduler. In *International Conference on Information Systems Architecture and Technology*. Springer, 242–251.
- [18] Binh Nguyen, Arijit Banerjee, Vijay Gopalakrishnan, Sneha Kasera, Seungjoon Lee, Aman Shaikh, and Jacobus Van der Merwe. 2014. Towards Understanding TCP Performance on LTE/EPC Mobile Networks. In *Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges (AllThingsCellular '14)*. ACM, New York, NY, USA, 41–46. <https://doi.org/10.1145/2627585.2627594>
- [19] Ashkan Nikravesh, Yihua Guo, Feng Qian, Z. Morley Mao, and Subhabrata Sen. 2016. An In-depth Understanding of Multipath TCP on Mobile Devices: Measurement and System Design. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 189–201. <https://doi.org/10.1145/2973750.2973769>
- [20] OpenSignal. 2018. State of LTE. "https://www.opensignal.com/reports/2018/02/state-of-lte".
- [21] Christoph Paasch, Simone Ferlin, Ozgu Alay, and Olivier Bonaventure. 2014. Experimental Evaluation of Multipath TCP Schedulers. In *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop (CSWS '14)*. ACM, New York, NY, USA, 27–32. <https://doi.org/10.1145/2630088.2631977>
- [22] Paasch et al. 2019. Multipath TCP Linux. "www.multipath-tcp.org/".
- [23] Costin Raiciu, Mark J. Handley, and Damon Wischik. 2011. Coupled Congestion Control for Multipath Transport Protocols. RFC 6356. <https://doi.org/10.17487/RFC6356>
- [24] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. 2012. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. USENIX Association, San Jose, CA.
- [25] Rui Li, M. Shariat, and M. Nekovee. 2016. Transport protocols behaviour study in evolving mobile networks. In *2016 International Symposium on Wireless Communication Systems (ISWCS)*. 456–460. <https://doi.org/10.1109/ISWCS.2016.7600947>
- [26] T. Shreedhar, N. Mohan, S. K. Kaul, and J. Kangasharju. 2018. QAware: A Cross-Layer Approach to MPTCP Scheduling. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. 1–9. <https://doi.org/10.23919/IFIPNetworking.2018.8696843>
- [27] H. Sinky, B. Hamdaoui, and M. Guizani. 2016. Proactive Multipath TCP for Seamless Handoff in Heterogeneous Wireless Access Networks. *IEEE Transactions on Wireless Communications* 15, 7 (July 2016), 4754–4764. <https://doi.org/10.1109/TWC.2016.2545656>
- [28] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong. 2014. TCP Performance over Mobile Networks in High-Speed Mobility Scenarios. In *2014 IEEE 22nd International Conference on Network Protocols*. 281–286.