A Measurement Study on Multi-path TCP with Multiple Cellular Carriers on High Speed Rails

Li Li[1,2], Ke Xu[1,2], Tong Li[3], Kai Zheng[3], Chunyi Peng[4], Dan Wang[5]
Xiangxiang Wang[1,2], Meng Shen[6], Rashid Mijumbi[7]
[1]Beijing National Research Center for Information Science and Technology, China
[2]Tsinghua University, China [3]Huawei Technologies, China
[4]Purdue University, USA [5]The Hong Kong Polytechnic University, China
[6]Beijing Institute of Technology, China [7]Nokia Bell Labs, Ireland
Email: ll-12@tsinghua.org.cn, xuke@tsinghua.edu.cn, li.tong@huawei.com
kai.zheng@huawei.com, chunyi@purdue.edu, csdwang@comp.polyu.edu.hk
wxx15@tsinghua.org.cn, shenmeng@bit.edu.cn and rashid.mijumbi@nokia.com

ABSTRACT

Recent advances in high speed rails (HSRs) are propelling the need for acceptable network service in high speed mobility environments. However, previous studies show that the performance of traditional single-path transmission degrades significantly during high speed mobility due to frequent handoff. Multi-path transmission with multiple carriers is a promising way to enhance the performance, because at any time, there is possibly at least one path not suffering a handoff. In this paper, for the first time, we measure multi-path TCP (MPTCP) with two cellular carriers on HSRs with a peak speed of 310km/h. We find a significant difference in handoff time between the two carriers. Moreover, we observe that MPTCP can provide much better performance than TCP in the poorer of the two paths. This indicates that MPTCP's robustness to handoff is much higher than TCP's. However, the efficiency of MPTCP is far from satisfactory. MPTCP performs worse than TCP in the better path most of the time. We find that the low efficiency can be attributed to poor adaptability to frequent handoff by MPTCP's key operations in sub-flow establishment, congestion control and scheduling. Finally, we discuss possible directions for improving MPTCP for such scenarios.

CCS CONCEPTS

• **Networks** → **Network protocols**;

KEYWORDS

Multi-path TCP, Measurement, Cellular Networks, High Speed Rails

1 INTRODUCTION

In the last few years, there has been a significant worldwide increase in the deployment of high speed rails (HSRs), reaching about 38,000 km at end of 2017. In this regard, China has played a more active role, contributing more than 60% of the world's HSR network in length. The passenger transport volume on HSRs in China was at least 1.7 Billion in 2017 and is growing at an annual rate of over 30%. Therefore, like never before, there is an increasing need for acceptable quality of network services in high speed mobility scenarios.

However, state-of-the-art measurement studies [1–5] indicate that the performance of traditional single-path transmission declines significantly under such extremely high speed mobility. It has been found that frequent handoff is the main cause of performance degradation [4, 5]. It is therefore urgent to find ways to enhance the

performance under frequent handoff. HSR lines in China are covered by cellular networks of various carriers. If handoff of different carriers happens at different times, multi-path transmission with multiple carriers may be able to enhance the performance: if at any time there is at least one path suffering no handoff, the unaffected path(s) can continue data transmission.

Multi-path TCP (MPTCP) [6–8] is a relatively mature solution to support multi-path transmission. The principle of MPTCP is to enhance both efficiency and robustness. For efficiency, MPTCP aims to make good use of available bandwidth of multiple network interfaces. For robustness, MPTCP aims to be more resilient to unpredictable and harmful events (such as packet loss bursts and disconnections) happening in a path. MPTCP has already been implemented in Linux kernel [9] and used in iOS. Previous studies have shown that MPTCP can enhance the performance in data centers [10], and wireless networks in static and low speed mobility cases [11–13]. If handoff of different carriers happens at different times, it is natural to assume that MPTCP can also show an advantage over TCP in high speed mobility cases.

The purpose of our study is to answer two questions:

Q1: Is there a significant difference in handoff time among various carriers covering HSR lines?

Q2: If handoff time is different among various carriers, does MPTCP perform as expected, showing a significant advantage over TCP?

To seek answers to these questions, for the first time, we compare handoff time between two large commercial carriers (referred to in this paper as Carriers M and U) in China, and conduct a comprehensive measurement on MPTCP with the two carriers over HSRs with a peak speed of 310 km/h. There are two main differences between previous measurements in wireless networks and our work: i) They perform measurements in static or low speed mobility cases, while we focus on high speed mobility cases. ii) They mainly measure MPTCP using cellular and WiFi networks at the same time, while we study MPTCP using two different cellular carrier networks simultaneously.

Our measurement work faces several challenges, which previous measurement studies would likely not encounter. In particular:

For Q1, while our focus is on handoff, the impact of other factors, including a diverse terrain of the areas crossed by trains, variation in train speed (parking, acceleration, running at full speed, and deceleration), and varying network type (ranging from 2G to 4G) cannot be ignored. For example, handoff frequency increases with

the increase in train speed. Handoff in the same networks is quite different from handoff between networks with two different types. It is difficult to collect information on so many factors without a dedicated tool.

For Q2, it is necessary to compare the performance of MPTCP and TCP flows. Network conditions vary significantly, and results may be biased by specific time and location during travel. It is challenging to minimize the bias. In addition, there are many external (such as terrain, speed, handoff frequency, and network type) and internal (such as flow size, flow duration, congestion control algorithm) factors which impact performance. These factors are intertwined, making it difficult to evaluate the advantage of MPTCP over TCP.

Finally, the measurement is effort and time intensive. It requires recruitment of many people, who then have to put in considerable time and effort to collect massive data traces on various HSR routes.

To overcome the above challenges, we make significant methodological contributions in both measurement and analysis. During a 22-month period, we accumulated a mileage of 82,266 km, over twice of the equatorial circumference of the earth, collecting nearly 2.8 TB of data. We evaluate the advantage of MPTCP over TCP in robustness and efficiency as follows: If MPTCP outperforms TCP in the poorer of the two paths, MPTCP is considered more robust than TCP. While if MPTCP outperforms TCP in the better path, MPTCP is considered more efficient than TCP.

We find that the probability that both Carriers M and U suffer a handoff at the same time is nearly 0 (as shown in Figure 4). This can be attributed to diversity in base station location and handoff technology between the two carriers. The difference in handoff time makes it possible for MPTCP to enhance the performance. It is natural to expect that MPTCP can show a significant advantage in both efficiency and robustness.

As expected, MPTCP shows a significant advantage in robustness. For both mice and elephant flows, MPTCP can provide a performance much better than TCP in the poorer path. MPTCP's advantage in robustness becomes more significant when the number of handoffs suffered by a flow increases (as shown in Figures 5 and 9).

However, unexpectedly, MPTCP's advantage in efficiency is far from satisfactory. For both mice and elephant flows, MPTCP performs worse than TCP in the better path most of the time. Moreover, MPTCP's efficiency decreases significantly with the increase in the number of handoffs suffered by a flow (as shown in Figures 5 and 9).

We find that mice and elephant MPTCP flows show low efficiency for different reasons. For mice flows, low efficiency is due to poor adaptability of subflow establishment to handoff (as shown in Figure 6), while for elephant flows, it is caused by poor adaptability of congestion control (as shown in Figure 12) and scheduling (as shown in Figure 14) to handoff. Finally, we discuss possible directions for improving MPTCP, or developing new more adaptive multi-path transport protocols for high speed mobility cases.

The rest of this paper is structured as follows: Section 2 discusses the motivation of using MPTCP. Section 3 describes the measurement and analysis methods, before analyzing the performance of mice and elephant flows in Sections 4 and 5 respectively. Section 6 introduces the future work, where possible directions for improving MPTCP are discussed. Section 7 covers related work, and we conclude the paper in Section 8.

2 MOTIVATION OF USING MPTCP

In this Section, we first analyze the negative impact of frequent handoff on single-path transmission, then characterize handoff on HSRs, and finally discuss the potential benefits of using MPTCP on HSRs.

2.1 Negative Impact of Handoff

Handoff can lead to heavy packet loss bursts and sharp increase in delay. It has been found that frequent handoff is the main cause of performance degradation [4, 5]. Compared with fast signal fading due to high speed mobility [14], such as Doppler frequency offset and fast multi-path fading, handoff shows a much more significant impact on single-path transmission.

We develop a dedicated tool, *MobiNet* [3], to measure handoff of a smartphone. The operational principle of *MobiNet* is introduced in detail in Section 3.1. *MobiNet* can read base station information (including Location Area Code (LAC) and Cell ID (CID)) from Android OS, and record the information in logs. When LAC or CID changes, it is considered that a handoff has occurred. Consistent with previous studies, we also observe that frequent handoff has a negative impact on single-path transmission.

Figure 1 shows an example of a 100 s long TCP flow measured in 4G networks of Carrier U at a speed over 280 km/h on the Beijing-Tianjin HSR line. The flow suffers 13 handoffs in total. Note that, *MobiNet* cannot detect when a handoff begins, but can detect when a handoff ends (i.e. the time when LAC or CID of base station changes). Therefore, we just show the points where handoffs complete by green stems in Figure 1. Figures 1 (a), 1 (b) and 1 (c) show the variation in throughput, RTT and packet loss rate over time respectively. We
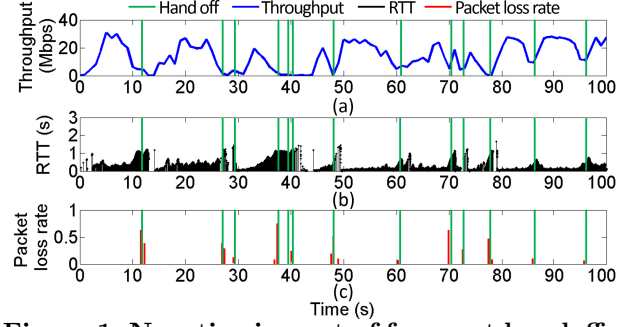

Figure 1: Negative impact of frequent handoff on TCP (Carrier U, Beijing-Tianjin HSR line)

find that the throughput of TCP decreases sharply, and RTT and packet loss rate increase significantly when suffering a handoff. For example, when no handoff occurs, throughput is as high as 30 Mbps, RTT is below 0.3 s, and packet loss rate is 0 at 7 s. However, due to a handoff, throughput drops below 3 Mbps, RTT reaches 1.0 s and packet loss rate surpasses 70% at 12 s. We conclude that frequent handoff hurts the performance significantly.

2.2 Handoff Characteristics

Since handoff is the main cause of performance degradation, it is necessary to characterize handoff on HSRs. After analyzing the logs of *MobiNet*, we make findings in two aspects:

Handoff frequency. We define handoff frequency as the number of handoffs per minute. Figure 2 depicts the CDF of handoff frequency in 4G networks of the two carriers on HSRs. We find that handoff frequency is very high. For example, the frequency exceeds 3 per minute with a probability of 37% and 65% for Carriers M and U respectively.

Difference in handoff time between two carriers. We quantify the difference in handoff time between the two carriers by inter-carrier handoff interval. It is defined as the time span between a handoff happening in one carrier network and a nearest handoff happening in the other carrier network. Figure 3 gives an example. During a 100 s period, 6 handoffs (marked by M-1, M-2, ..., M-6) occur in Carrier M network, and 6 handoffs (marked by U-1, U-2, ..., U-6) occur in Carrier U network. As an example, handoff M-2 is the nearest handoff in Carrier M network to handoff U-2 in Carrier U network. Therefore, inter-carrier handoff interval is the time between M-2 and U-2. In this example, handoff of the two carriers never happens at the same time. Such a case is prevalent. Figure 4 shows the CDF of inter-carrier handoff interval along various HSR lines. It is observed that the probability that the inter-carrier handoff interval equals to 0 is nearly 0. That is to say,
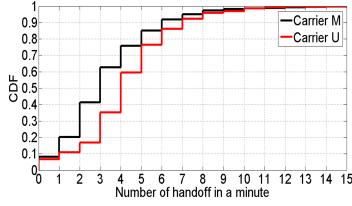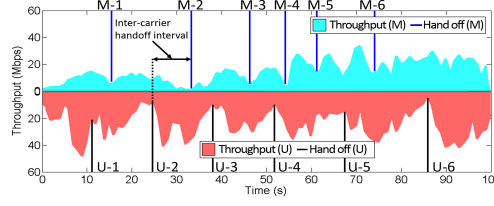
Figure 2: CDF of handoff frequency

Figure 3: An example of difference in handoff time between two carriers
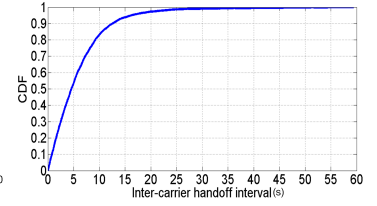

Figure 4: CDF of inter-carrier handoff interval

the handoff of two different carriers rarely happens at the same time. The difference in handoff time can be attributed to differences in base station locations and handoff technologies between the two carriers.

2.3 Potential Benefits of MPTCP

Making use of the difference in handoff time between two carriers, multi-path transmission has the potential to improve performance. Figure 3 shows an example of two single-path TCP flows using different carrier networks measured simultaneously. It is observed that the total throughput of two TCP flows is robust to handoff: when one flow suffers a handoff and throughput drops, the other flow does not suffer a handoff at the same time and throughput is much higher. As a result, the total throughput of the two flows rarely drops to a very low level even when either carrier suffers a handoff. It is worth noting that MPTCP does not simply use a single-path TCP on each path as illustrated in Figure 3. However, the example still implies that MPTCP has the potential to improve the performance. This strongly motivates us to try MPTCP with two carriers on HSRs.

3 MEASUREMENT AND ANALYSIS METHOD

This Section describes the measurement method, presents the dataset, and finally introduces the analysis method.

3.1 Measurement Method

We deploy static servers and mobile clients. The servers are rented from Alibaba's Aliyun Elastic Compute Service (ECS) [15]. Each server is configured as an HTTP server, running Apache 2. We take laptops as the clients on high speed trains to retrieve Web objects of different sizes from the servers using "wget". All of the servers and laptops are running Ubuntu 14.04 with Kernel version 4.1.35 using the stable release of the MPTCP Kernel implementation version v0.91 [9]. Each laptop can access multiple cellular networks of different carriers at the same time using USB cellular modems, or USB WiFi modems accessing WiFi hotspots transferred from cellular networks by smartphones. We explain the reason why we use both these access approaches later.

There are two main tasks:

Task 1: Collecting information on various external factors, including terrain, speed, handoff and network type.

Solution: Laptops use two ways to access cellular networks: 1) USB cellular modems. This is the straightforward way, and is widely used in previous studies [12]. 2) A smartphone hotspot is the alternative approach. Smartphones first transfer cellular networks to WiFi hotspots, and then laptops access the cellular networks through the WiFi hotspots using USB WiFi modems. Table 1 shows all cellular devices used in our experiments which support various radio access technologies (RATs) of different carriers. The reason for also using smartphone hotspots is that we developed an app (for Android smartphones), *MobiNet* [3], to collect information on multiple factors affecting the performance. *MobiNet* is a real-time measurement tool which is able to obtain geographical location and speed of the train via GPS, read the network type, LAC and CID of base station, and record all these information in logs. When LAC or CID changes, it is considered that a handoff has occurred. *MobiNet* is implemented based on multiple Android APIs, such as *GsmCellLocation* [16], *Location* [17], *TelephonyManager* [18], etc. However, we cannot develop a similar app on Linux for the laptops to collect the information of the USB cellular modem, as USB cellular modems do not allow the laptop to access such information. Therefore, we also use the approach of smartphone hotspots.

Task 2: Measuring flows with diverse size and duration, and using diverse congestion control algorithms.

Solution: To compare the performance of MPTCP and TCP, we measure two-path MPTCP flows using Carriers M and U, single-path TCP flows using Carrier M, and single-path TCP flows using Carrier U. As shown in Table 2, we measure three congestion control algorithms for MPTCP: Reno (i.e. using TCP New Reno [19] over each of the sub-flows), Linked Increases Algorithm (LIA) [20] and Opportunistic Linked-Increases Congestion Control Algorithm (OLIA) [21]. Irrespective of algorithm, each MPTCP sub-flow behaves as a legacy

Table 1: Mobile devices

Carrier	Device	Version (model)	RAT
M	Smartphone	Samsung Galaxy S6	TD-LTE (4G), TD-SCDMA (3G), GSM (2G)
U	Smartphone	Samsung Galaxy S6	FDD-LTE (4G), WCDMA (3G), GSM (2G)
M	USB Cellular Modem	ZTE MF832U	TD-LTE (4G), TD-SCDMA (3G), GSM (2G)
U	USB Cellular Modem	ZTE MF832S	FDD-LTE (4G), WCDMA (3G), GSM (2G)

New Reno TCP flow during the slow start stage. After the slow start, sub-flows enter the congestion avoidance stage, where the congestion window (CWND) is updated in an additive increase and multiplicative decrease (AIMD) way. The difference between various congestion control algorithms only lies in the additive increase rate. We measure one congestion control algorithm for TCP: Reno (i.e. New Reno). Although Cubic [22] is more popular than New Reno, we choose to measure the latter. This is because all the three congestion control algorithms used by MPTCP are designed based on TCP New Reno [6, 23, 24]. It would not be fair to compare MPTCP with TCP using Cubic.

In cellular networks, most flows are small. On the other hand, a very small fraction of large flows, which are known as "heavy-hitter" flows, contribute to the majority of the traffic volume [25, 26]. Therefore, it is of significance to study the performance of both mice and elephant flows. As shown in Table 2, we measure mice flows with three sizes: 64 KB, 256 KB, and 1 MB. Besides, we also measure elephant flows lasting 100 s.

Table 2: Test flows

Protocol	Congestion controller	Size or duration
TCP	Reno	64 KB, 256 KB, 1 MB, 100s
MPTCP	Reno, LIA, OLIA	64 KB, 256 KB, 1 MB, 100s

Table 3: HSR lines.

Route	B-F	B-Y	B-H	B-T	B-K	H-K
Mileage (km)	19,234	6,048	23,724	7,820	11,928	13,512

To compare MPTCP and TCP under similar conditions, MPTCP flows using Carriers M and U, TCP flows using Carrier M, and TCP flows using Carrier U are measured simultaneously on the same train. However, this way, the competition between sub-flows of MPTCP and TCP flows using the same carrier network may have an impact on the performance. Therefore, we also measure MPTCP and TCP at different times to avoid competition between flows. Measuring MPTCP and TCP flows in both of the two different ways can reduce possible bias brought by only using one way.

Before each test, the congestion control algorithm, and flow size/duration are randomly selected from Table 2. This way, all types of flows are measured in random locations and time. Then we can make a fairer comparison between flows with diverse size or duration, and using different congestion control algorithms, because the bias brought by specific location and time is minimized.

3.2 Dataset

Data is collected in two ways: 1) We capture all the packets at both the client and server using tcpdump [27]. 2) *MobiNet* records the location and speed of the train, and various network parameters in logs.

From September 2016 to June 2018, we covered six popular HSR routes in China, i.e. Beijing-Futian (B-F), Beijing-Shenyang (B-Y), Beijing-Shanghai (B-H), Beijing-Tianjin (B-T), Beijing-Kunming (B-K), and Shanghai-Kunming (H-K). Table 3 shows the HSR lines. Taking the B-T line as an example, the length of each trip is 115 km, we accumulate a mileage of 7,820 km in 68 one-way trips.

Note that to perform static measurements, experiments are started dozens of minutes before the train leaves the originating station, and are maintained for a while after the train arrives at the terminal. Besides, we also perform static measurements outdoors in open areas in Beijing, not in high speed trains.

Table 4 shows details of the dataset on HSRs. We divide the dataset into three sub-datasets: MU-MPTCP, M-TCP and U-TCP. The total size of the three sub-datasets is nearly 2.8 TB. The three sub-datasets contain data of MPTCP flows using both Carriers M and U, TCP flows using Carrier M, and TCP flows using Carrier U respectively. As an example, in sub-dataset MU-MPTCP, the number of 64 KB-sized, 256 KB-sized, 1 MB-sized, and 100s long flows collected on B-F line is 1887, 1918, 1895, and 722 respectively. In addition to these data collected on HSRs, we also collect 50 GB of data when performing static measurements outdoors in open areas in Beijing.

3.3 Analysis Method

Many influence factors are highly varying and intertwined together, making it difficult to evaluate the advantage of MPTCP over TCP. To overcome this challenge, we design the analysis method, as follows:

We first filter the data. We make use of the data collected by *MobiNet*: latitude and longitude (we input the latitude and longitude into Google Earth to learn about the terrain), speed, handoff and network type. Terrain of areas along HSR lines is diverse in China, including hills, valleys, tunnels and plains, leading to significant variations in signal quality. The train speed is over 280 km/h most of the time, but is between 0 and 280 km/h during acceleration (after leaving a station) and deceleration (before arriving a station). Handoff frequency

Table 4: Dataset

Route	Dataset: MU-MPTCP				Dataset: M-TCP				Dataset: U-TCP			
	# of flows (64 KB)	# of flows (256 KB)	# of flows (1 MB)	# of flows (100 s)	# of flows (64 KB)	# of flows (256 KB)	# of flows (1 MB)	# of flows (100s)	# of flows (64 KB)	# of flows (256 KB)	# of flows (1 MB)	# of flows (100s)
B-F	1887	1918	1895	722	1933	1922	2015	879	1879	1971	1897	728
B-Y	1336	1350	1323	456	1371	1356	1332	449	1332	1333	1236	466
B-H	3453	3407	3393	988	4038	4019	4002	1381	3671	3978	3661	1012
B-T	3025	3109	3073	1008	3020	3025	3124	1012	3148	2945	3020	997
B-K	1841	1935	1868	611	1801	1847	1863	606	1740	1811	1807	622
H-K	2128	2088	2113	708	2322	2015	2018	713	2027	2104	2235	721

Table 5: Classification of mice flows

Conditions	Group:MS-US			Group:M0-U0			Group:M0-U1			Group:M1-U0		
	MPTCP (M+U)	TCP (M)	TCP (U)	MPTCP (M+U)	TCP (M)	TCP (U)	MPTCP (M+U)	TCP (M)	TCP (U)	MPTCP (M+U)	TCP (M)	TCP (U)
Speed(km/h)	0	0	0	280-310	280-310	280-310	280-310	280-310	280-310	280-310	280-310	280-310
# of handoff(M)	0	0	-	0	0	-	0	0	-	1	1	-
# of handoff(U)	0	-	0	0	-	0	1	-	1	0	-	0

changes with the variation in base station distribution density along HSR lines. Besides, it is observed that 4G LTE is the dominant type of networks covering HSR lines, but passengers on HSRs also connect to 2/3G networks sometimes. Since the four factors are all changing, the analysis is nearly infeasible. To make the analysis feasible, we only focus on two key factors, speed and handoff, and fix the other factors. We only study data collected in 4G LTE networks in areas of open plains. For speed, we only consider two cases: static and high speed (280-310 km/h). The former case is used as a baseline, so that we can understand the difference brought by high speed mobility. Therefore, flows measured when speed is between 0 and 280 km/h are discarded.

After filtering, we evaluate the advantage of MPTCP over TCP. It makes no sense to compare MPTCP and TCP under different conditions. For example, it is not fair to compare a MPTCP flow measured in static cases, with a TCP flow suffering multiple handoffs in high speed mobility cases. Therefore, we compare MPTCP and TCP flows with the same size/duration, measured at the same train speed, and suffering the same number of handoffs in the same carrier network.

We evaluate the advantage of MPTCP over TCP in robustness and efficiency as follows: If MPTCP outperforms TCP in the poorer of the two paths, MPTCP is considered more robust than TCP. While if MPTCP outperforms TCP in the better path, MPTCP is considered more efficient than TCP.

4   FINDINGS ON MICE FLOWS

In this Section, we study the performance of mice flows, including 64 KB-sized, 256 KB-sized, and 1 MB-sized flows.

4.1   Flow Completion Time

It is observed that in the short duration of a mice flow, a carrier at most suffers one handoff when the speed is over 280 km/h most of the time. Moreover, the probability that the two carriers both suffer a handoff in the same short duration is low. According to speed and number of handoffs suffered by each carrier, as shown
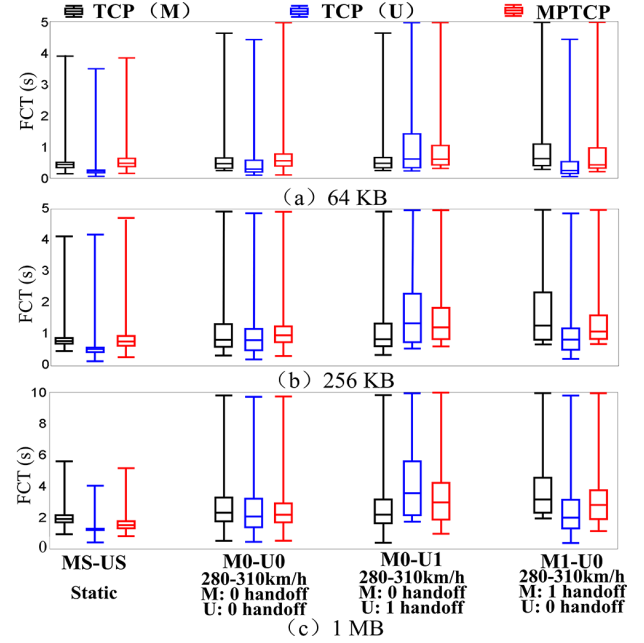

Figure 5: FCT of mice flows

in Table 5, we select four groups of flows: MS-US, M0-U0, M0-U1 and M1-U0. We only compare MPTCP and TCP flows in the same group. As an example, flows in group M0-U1 are measured at a speed over 280 km/h. M0-U1 includes MPTCP flows with one path using Carrier M suffering no handoff, and the other path using Carrier U suffering a handoff, TCP flows with Carrier M suffering no handoff, and TCP flows with Carrier U suffering a handoff.

Figures 5 (a), 5 (b) and 5 (c) show flow completion time (FCT) when downloading 64 KB-sized, 256 KB-sized and 1 MB-sized files respectively. We find that for both TCP and MPTCP, FCT is longer and varies in a wider range in high speed mobility cases, compared with static cases. *We evaluate the advantage of MPTCP as follows:*

a) Robustness. Compared with the case that neither path suffers a handoff (i.e. Groups MS-US and M0-U0),
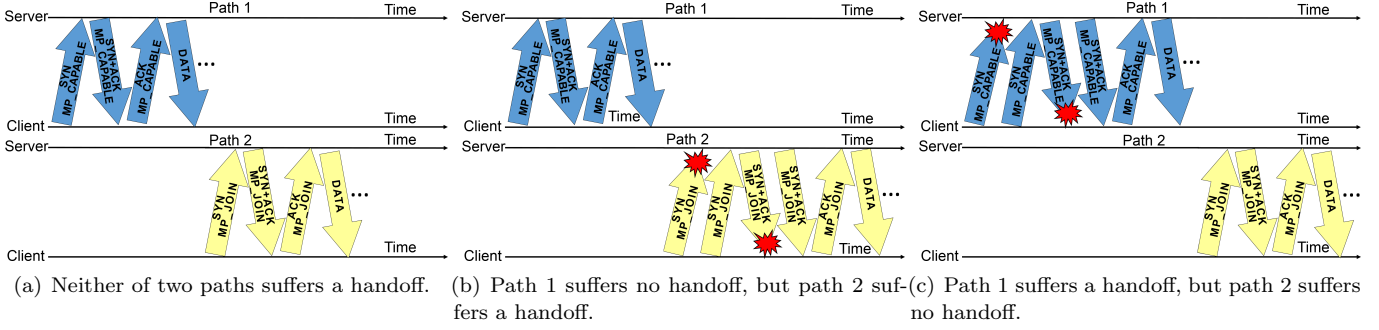
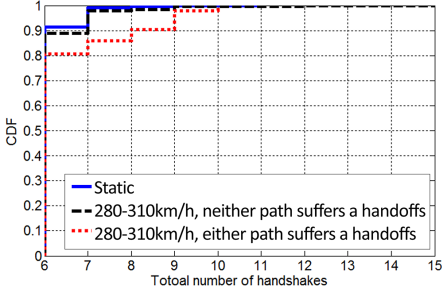(a) Neither of two paths suffers a handoff. (b) Path 1 suffers no handoff, but path 2 suffers a handoff. (c) Path 1 suffers a handoff, but path 2 suffers no handoff.

Figure 6: Sub-flow establishment

Figure 7: CDF of total number of handshakes

Figure 8: CDF of total sub-flow establishment time

MPTCP can show a more significant advantage in robustness in the case that either path suffers a handoff (i.e. Groups M0-U1 and M1-U0). When either path suffers a handoff, FCT of MPTCP is much shorter than that of TCP in the poorer path (i.e. the path suffering a handoff), indicating that MPTCP is more resistant to handoff than TCP.

b) Efficiency. MPTCP cannot show a significant advantage over TCP in efficiency in all the four groups. FCT of MPTCP, is longer than that of TCP in the better path most of the time. Worse still, handoff leads to a significant reduction in efficiency. When the train speed is over 280km/h, compared with the case that neither path suffers a handoff, the gap between FCT of MPTCP to that of TCP in the better path increases significantly in the case that either path suffers a handoff.

Besides, we find that MPTCP Reno, OLIA and LIA show comparable performance. This is because slow start time takes a big portion of FCT. Irrespective of algorithm, each MPTCP sub-flow behaves in the same way during slow start. As a result, the three algorithms show comparable performance when downloading small files.

To summarize, MPTCP shows an advantage in robustness, but is not efficient enough when downloading small files on HSRs. We find that the low efficiency of MPTCP is related with the inefficient sub-flow establishment. We make an analysis in Section 4.2.
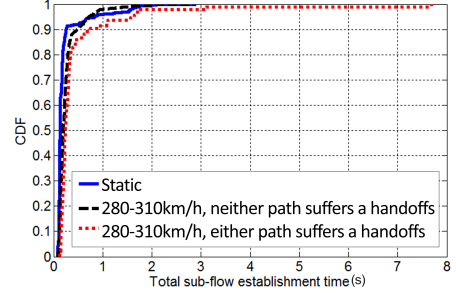
4.2 Sub-flow Establishment

It takes a short time to download a file with size below 1 MB. Sub-flow establishment time takes a considerable portion of the FCT. Therefore, for mice flows, the efficiency of sub-flow establishment has a significant impact on the performance. We find that handoff poses big challenges to sub-flow establishment. We compare the behavior of sub-flow establishment in two cases:
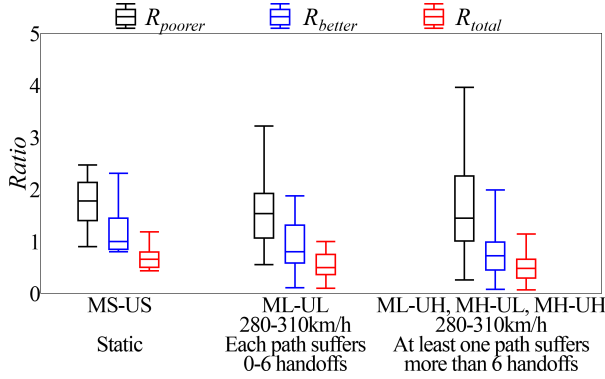
1) Neither path suffers a handoff. Figure 6(a) depicts the process of sub-flow establishment in normal cases. As shown in the figure, MPTCP does not establish two sub-flows in parallel, but serially [28]. It first establishes a sub-flow by three handshakes with a flag of "MP_CAPA-BLE". After the first sub-flow is established successfully, the second sub-flow is also established by three handshakes with a flag of "MP_JOIN". MPTCP uses six handshakes in total, and the total sub-flow establishment time equals to the sum of the time taken to establish each sub-flow.

2) Either path suffers a handoff. Handoff poses big challenges to sub-flow establishment. Packet loss rate of the path suffering a handoff is much higher than that of the other path. Handshakes in the lossy path may be lost and retransmitted. **i)** As shown in Figure 6(b), if MPTCP luckily establishes the first sub-flow in the better path, the effect of handoff is not significant. Although it takes much longer time to establish the second sub-flow in the lossy path due to losses of handshakes,

Table 6: Classification of elephant flows

Conditions	Group:MS-US			Group:ML-UL			Group:ML-UH		
	MPTCP (M+U)	TCP (M)	TCP (U)	MPTCP (M+U)	TCP (M)	TCP (U)	MPTCP (M+U)	TCP (M)	TCP (U)
Speed(km/h)	0	0	0	280-310	280-310	280-310	280-310	280-310	280-310
# of handoff(M)	0	0	-	0-6	0-6	-	0-6	0-6	-
# of handoff(U)	0	-	0	0-6	-	0-6	>6	-	>6

Conditions	Group:MH-UL			Group:MH-UH		
	MPTCP (M+U)	TCP (M)	TCP (U)	MPTCP (M+U)	TCP (M)	TCP (U)
Speed(km/h)	280-310	280-310	280-310	280-310	280-310	280-310
# of handoff(M)	>6	>6	-	>6	>6	-
# of handoff(U)	0-6	-	0-6	>6	-	>6

Figure 9: R_{poorer}, R_{better} **and** R_{total}

the first sub-flow in the better path can begin data transmission timely. **ii)** However, as shown in Figure 6(c), if MPTCP unluckily establishes the first sub-flow in the lossy path, the effect of handoff is more significant. In such a case, it takes longer time to establish the first sub-flow due to handshake losses. Although the network condition of the other path is much better, the second sub-flow cannot be established and data transmission cannot start until the first sub-flow is established successfully. In fact, MPTCP cannot choose the initial path itself. MPTCP relies on routing information to determine the destination for the initialization path regardless of its quality. When MPTCP unluckily selects a path suffering a handoff as the initialization path, establishment time increases significantly due to losses and retransmissions of handshakes.

Figure 7 shows the CDF of the total number of handshakes used to establish two sub-flows successfully. It is observed that the probability with which MPTCP sends more than 7 handshakes is 3%, when no handoff occurs, but rises to as high as 12% when either path suffers a handoff. Therefore, as shown in Figure 8, it takes much longer time to establish two sub-flows, when either path suffers a handoff.

To summarize, efficiency of sub-flow establishment is low. Using a serial approach, the time taken to establish both sub-flows is a big portion of the duration for a mice flow. Worse still, handoff leads to a significant increase in sub-flow establishment time. We discuss possible directions to fix this problem in Section 6.1.

5 FINDINGS ON ELEPHANT FLOWS

In this Section, we study the performance in a long duration, i.e. for flows lasting 100 s.

5.1 Average Rate

It is observed that in a 100 s duration, a carrier can suffer various number of handoffs, when the speed is over 280 km/h. We define two levels of the number of handoffs suffered by a carrier in 100 s: low level (0 to 6 handoffs) and high level (> 6 handoffs). According to speed and level of the number of handoffs, as shown in Table 6, we select 5 groups of elephant flows: MS-US, ML-UL, ML-UH, MH-UL and MH-UH. MPTCP and TCP flows in the same group are measured on the same train. In each experiment, all the three types of flows start at the same time, and end simultaneously 100 s later.

We evaluate the advantage of MPTCP over TCP based on average rate. Average rate is defined as the ratio of the total number of bytes downloaded to the duration, 100 s. The average rate of MPTCP, TCP with Carrier M, and TCP with Carrier U are denoted by $AR_{MPTCP-MU}$, AR_{TCP-M} and AR_{TCP-U} respectively. We propose three ratios to evaluate the advantage of MPTCP:

$$R_{poorer} = \frac{AR_{MPTCP-MU}}{Min(AR_{TCP-M}, AR_{TCP-U})}$$
$$R_{better} = \frac{AR_{MPTCP-MU}}{Max(AR_{TCP-M}, AR_{TCP-U})}$$
$$R_{total} = \frac{AR_{MPTCP-MU}}{AR_{TCP-M} + AR_{TCP-U}}$$

We use MPTCP and TCP flows measured in the same 100 s duration to calculate the three ratios. R_{poorer} is the ratio of $AR_{MPTCP-MU}$ to the minimum value of AR_{TCP-M} and AR_{TCP-U}. If R_{poorer} is greater than 1, MPTCP shows an advantage in robustness. R_{better} is the ratio of $AR_{MPTCP-MU}$ to the maximum value of AR_{TCP-M} and AR_{TCP-U}. If R_{better} is over 1, it is considered that MPTCP shows an advantage in efficiency. R_{total} is the ratio of $AR_{MPTCP-MU}$ to the sum of AR_{TCP-M} and AR_{TCP-U}. R_{total} shows the gap between the average rate of MPTCP with the total average rate when using single-path TCP on each path.

Figure 9 shows R_{poorer}, R_{better} and R_{total} in various cases. **We make following findings:**

1) Robustness. R_{poorer} is over 1 most of the time in both static and high speed mobility cases, indicating an advantage in robustness of MPTCP over TCP. The probability that R_{poorer} reaches a high level increases with the increase in train speed and the number of handoffs suffered. For example, the probability that R_{poorer} surpasses 3 is 0 in static cases (i.e. Group MS-US). However, at a speed over 280km/h, the probability is 5% when each path suffers at most 6 handoffs (i.e. Group ML-UL), and reaches 13% when at least one path suffers more than 6 handoffs (i.e. Groups ML-UH, MH-UL and MH-UH).

2) Efficiency. R_{better} decreases significantly with the increase in train speed and the number of handoffs suffered. The probability that R_{better} surpasses 1 is 51% in static cases. At a speed over 280km/h, the probability is 37% when each path suffers at most 6 handoffs, and drops to 27% when at least one path suffers more than 6 handoffs. This indicates a significant reduction in efficiency. MPTCP cannot show an advantage in efficiency most of the time in high speed mobility cases.

3) Gap to the total average rate of TCP in two paths. R_{total} decreases significantly with the increase in train speed and the number of handoffs suffered. The probability that R_{total} surpasses 0.7 is 46% in static cases. At a speed over 280km/h, the probability drops to 37% when each path suffers at most 6 handoffs, and is as low as 24% when at least one path suffers more than 6 handoffs. It indicates that MPTCP makes poorer use of network resources of two paths in high speed mobility cases than in static cases.

Besides, the difference between MPTCP Reno, OLIA and LIA is bigger for elephant flows than for mice flows. When train speed is high, and handoff happens frequently, Reno provides the highest average rate, OLIA follows and LIA provides the lowest. Figure 10 shows the CDF of the average rate of the three congestion algorithms, when speed is over 280 km/h, and at least one path suffers more than 6 handoffs. It is observed that average rate surpasses 10 Mbps with a probability of 50%, when using Reno. However, the probability drops to 36% and 19% when using OLIA and LIA respectively.

We conclude that based on the average rate, MPTCP is more robust, but less efficient than TCP on HSRs. The low efficiency can be attributed to the poor adaptability of congestion control and scheduling to frequent handoffs, and we make an analysis in Sections 5.2 and 5.3 respectively.
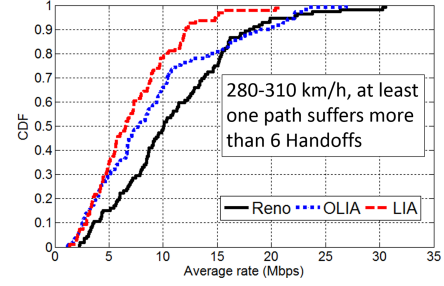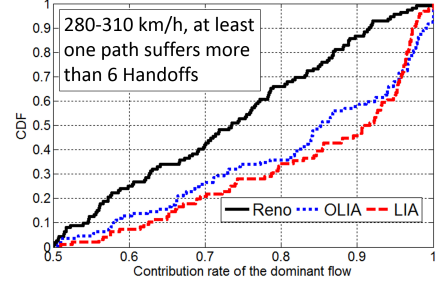

Figure 10: CDF of average rate

Figure 11: CDF of the dominant sub-flow's contribution rate

5.2 Congestion Control

To seek the root cause for low efficiency of MPTCP in high speed mobility cases, we study the traffic distribution balance between sub-flows. We define a sub-flow's contribution rate as the ratio of the number of bytes downloaded by the sub-flow to the total number of bytes downloaded by two sub-flows. If a sub-flow has the higher contribution rate of two sub-flows, we call it the *dominant sub-flow*. We use the contribution rate of the *dominant sub-flow* to quantify the degree of traffic distribution balance. If the *dominant sub-flow*'s contribution rate is close to 1, it implies that traffic distribution is extremely imbalanced. Figure 11 shows the CDF of the dominant sub-flow's contribution rate for 100 s MPTCP flows when the speed is over 280 km/h, and at least one path suffers more than 6 handoffs. It is observed that the traffic distribution between the two paths for Reno, OLIA and LIA are quite imbalanced most of the time. Since one of the two paths makes only a small contribution, it is expected that for elephant flows, MPTCP cannot show a significant advantage over TCP. The traffic distribution imbalance can be attributed to significant CWND distribution imbalance between the two paths.

To explain this issue clearly, we show three examples: a MPTCP flow using Reno in static cases (as shown in Figure 12(a)), two MPTCP flows using Reno and LIA respectively in high speed mobility cases (as shown in Figures 12(b) and 12(c) respectively). In each figure, we show the total throughput of two sub-flows, and the flight size (i.e. the number of bytes in flight) of each

sub-flow. We also mark moments when handoffs occur. We use flight size of a sub-flow to estimate the size of the sub-flow's sliding window (SWND), which is the minimum of the sub-flow's CWND and advertised window (AWND) of the receiver. It is observed that AWND is bigger than CWND most of the time. Therefore, flight size of a sub-flow roughly equals to the CWND. **We make the following findings :**

As shown in Figure 12(a), in static cases, only a few multiplicative decreases occur due to light packet losses in two paths, and the distribution of CWND between two paths is quite balanced. However, as shown in Figures 12(b) and 12(c), due to frequent packet loss bursts, CWND of sub-flows suffers multiplicative decreases repeatedly and CWND distribution is extremely imbalanced in high speed mobility cases. It is observed that packet loss bursts often happen around moments when handoffs occur. Since handoff in two paths happens at different times, and harm brought by handoff (such as increases in delay and packet loss rate) is quantitatively different between Carriers M and U, network conditions of two paths are quite different. This is the main cause of CWND distribution imbalance.

It is observed that the CWND distribution imbalance is higher for LIA and OLIA than Reno. We explain this as follows: CWND of two paths for LIA and OLIA are coupled. LIA and OLIA are designed to transfer traffic from a congested path to a less congested one. For OLIA, sub-flow with maximum $\frac{l_r^2}{rt_r}$ is treated as the "best" sub-flow, where l_r and rt_r are packet loss rate and RTT of path r separately. It tries to increase increment factor [23] for CWND of the "best" sub-flow, and decrease the factor of other sub-flows. Therefore, if the network conditions of a path with smaller CWND improve, and the path is considered as the "best" sub-flow by OLIA, CWND of the path can rise faster. This can reduce the imbalance of CWND distribution. However, all sub-flows of LIA have the same increment factor [6] for CWND, so a path with smaller CWND does not have a chance to increase its CWND faster, and the imbalance of CWND distribution cannot be reduced. Compared with OLIA and LIA, Reno does not transfer traffic from a congested path to a less congested path, because the two paths are not coupled, but independent. **Therefore, LIA shows the most significant CWND distribution imbalance, OLIA follows, and Reno shows the lowest.** The cases shown in Figures 12(b) and 12(c) are prevalent. This explains why traffic distribution imbalance of LIA is more significant than OLIA and Reno in high speed mobility cases, as shown in Figure 11. Note
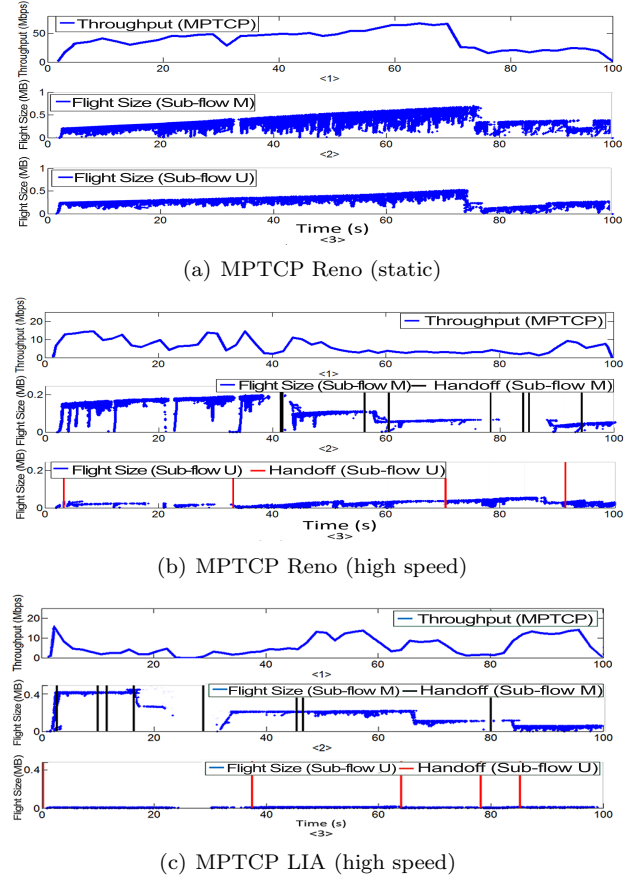

(a) MPTCP Reno (static)

(b) MPTCP Reno (high speed)

(c) MPTCP LIA (high speed)

Figure 12: Variation in throughput and flight size of MPTCP over time

that, since MPTCP cannot distinguish between congestion and handoff, coupled congestion control algorithms, including LIA and OLIA, cannot transfer traffic among sub-flows timely and accurately on HSRs. Coupled congestion control algorithms may lead to unnecessary traffic convergency to one of the two paths.

CWND distribution imbalance reduces the resistance of MPTCP to handoff. Handoff has a significant impact on CWND and SWND. The additive increase and multiplicative decrease approach used by MPTCP cannot adapt well to frequent handoffs. MPTCP conducts very aggressive CWND reduction when heavy packet loss bursts occur due to handoff, and CWND cannot increase quickly after packet loss bursts. As a result, CWND probably drops to a very low level after suffering multiple handoffs, hurting the performance significantly. Worse still, SWND is sometimes frozen for a certain period when suffering handoffs. During the freeze period, SWND cannot slide, and no or only a few retransmitted packets can be sent out. For example, the SWND of sub-flow with Carrier M is frozen from 80 s to 87 s in Figure

12(b)-2 due to a handoff at 77 s. SWND freezes can be explained by the exponential back-offs triggered by re-transmitting a packet for multiple times due to the high packet loss rate. Making use of the difference in hand-off time between two carriers, MPTCP is expected to have better resistance to handoff than single-path TCP: even if one sub-flow suffers a handoff and throughput is reduced, the other sub-flow does not suffer a hand-off at the same time and throughput is much higher. Therefore, the total throughput of two sub-flows would be robust to handoff. However, the CWND distribution imbalance reduces the resistance of MPTCP to handoff. For example, the SWND of a sub-flow with Carrier M is frozen from 22 s to 32 s in Figure 12(c)-2. Unfortunately, the CWND of the other sub-flow is nearly zero in the same period in Figure 12(c)-3, so it cannot help MPTCP enhance the resistance to handoff. As a result, through-put drops significantly during that period. Compared with LIA, the resistance of Reno and OLIA to handoff is a little better. For example, although the SWND of a sub-flow with Carrier M is frozen from 80 s to 87 s in Fig-ure 12(b)-2, the CWND of the other sub-flow is about 0.06 MB in the same period in Figure 12(b)-3. With the contribution of the sub-flow with Carrier U, throughput does not drop to zero, but to around 3.5 Mbps during the period. A similar observation for OLIA can also be made.

To summarize, due to the big difference in network conditions between two carrier networks, CWND dis-tribution is less balanced in high speed mobility cases than in static cases. Moreover, coupled congestion con-trol algorithms, which are designed to transfer traffic from a congested path to a less congested path, lead to more significant CWND distribution imbalance. Ex-treme CWND distribution imbalance leads to significant traffic distribution imbalance, preventing MPTCP from showing enough advantage. Since MPTCP cannot differ-entiate between congestion and handoff, its congestion control cannot adapt well to frequent handoffs, hurting the performance significantly. We discuss possible direc-tions to fix this problem in Section 6.2.

5.3 Out of Order Problem

In addition to the inefficient congestion control, a signif-icant out-of-order problem is another important cause for MPTCP's low efficiency.

MPTCP uses a two-layered sequence number: sub-flow sequence number and data sequence number. Each sub-flow has an independent sub-flow sequence num-ber. All sub-flows share the same data sequence num-ber space. The receiver reorders packets received from
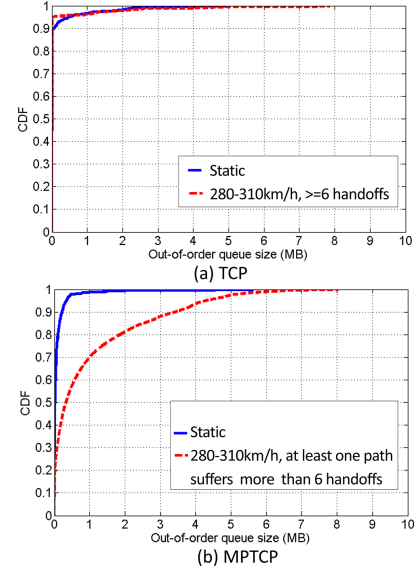

Figure 13: Out-of-order queue size

different sub-flows according to the data sequence num-ber. Only in-order packets can be submitted to the ap-plication layer. This means that out-of-order packets have to wait in an out-of-order queue, until the packets are ranked in order after all the expected packets are received. We define the size of the out-of-order queue as the number of bytes waiting in the queue. Previous studies show that out-of-order problem hurts the perfor-mance of MPTCP significantly [8, 12].

Figures 13(a) and 13(b) show the CDF of the out-of-order queue size for 100 s long TCP and MPTCP flows respectively. It can be observed that, only using a sin-gle path, the out-of-order queue size of TCP is small in both static and high speed mobility cases. However, the out-of-order queue size rises significantly for two-path MPTCP in high speed mobility cases, especially when suffering frequent handoffs. To explain the issue clearly, we give two examples. Figure 14(a) shows an example of a 100 s long MPTCP flow measured in static cases. Fig-ures 14(a)-1, 14(a)-2, 14(a)-3, 14(a)-4 and 14(a)-5 show the variation in RTT of sub-flow using Carrier M, RTT of sub-flow using Carrier U, out-of-order queue size, data rate, and throughput over time respectively. Through-put is defined as the number of all bytes received by the client per second. Data rate is defined as the num-ber of in-order bytes received by the client per second. Although out-of-order packets can be used to calculate the throughput, they cannot be used to calculate the data rate until they are ranked in order after waiting in an out-of-order queue. Similarly, Figure 14(b) shows corresponding metrics of a 100 s long MPTCP flow mea-sured on B-H line at a speed over 280 km/h. We also
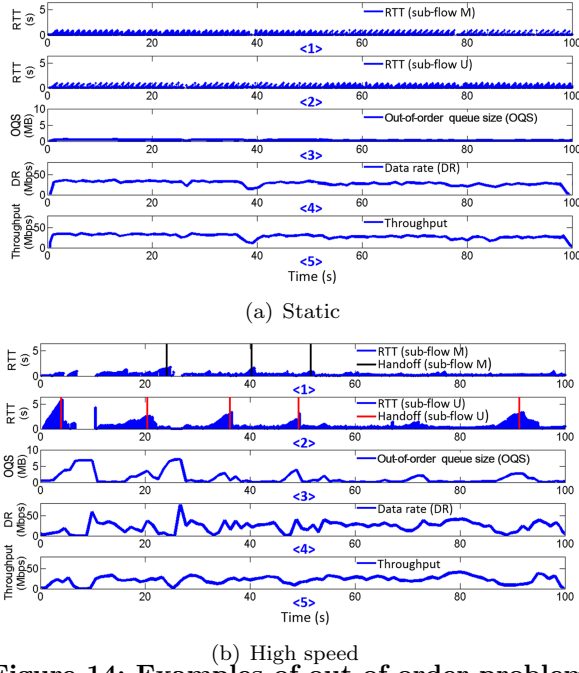
(a) Static

(b) High speed

Figure 14: Examples of out-of-order problem.

mark moments when handoffs occur. We make following findings:

i) Static cases. RTT is stable and low, and the size of the out-of-order queue is small. Data rate is roughly equal to the throughput. Both data rate and throughput are high and stable, rarely dropping to a low level.

ii) High speed mobility cases. Two paths experience rapid and wide-ranged RTT variation. In particular, significant RTT spikes appear when suffering a handoff. As shown in Figure 14(b), the out-of-order queue size rises with an increase in the RTT of sub-flows, especially around moments of handoffs. Since only in-order packets can be used to calculate the data rate, the gap between throughput and data rate is enlarged. For example, as shown in Figures 14(b)-4 and 14(b)-5, throughput is as high as 13 Mbps at 33 s, but the data rate is nearly zero. This reveals that the significant reduction of data rate is caused by out-of-order packets. Besides, the data rate surpasses 40 Mbps with a probability of 5%, much higher than the probability with which throughput surpasses 40 Mbps (nearly zero). This can be explained by a large number of accumulated packets ranked in order again after a period of waiting.

The significant out-of-order problem can be attributed to poor adaptability of the scheduler to rapid and wide-ranged varying RTT. We discuss possible directions to fix this problem in Section 6.3.

6 FUTURE WORK

In the future, we will improve MPTCP in three aspects: sub-flow establishment, congestion control and scheduling. We summarize previous works that are of significant reference value. We also propose new suggestions for high speed mobility cases.

6.1 Sub-flow Establishment

MPTCP cannot choose the initial path itself. MPTCP relies on routing information to determine the destination for the initialization path regardless of its quality. This has a big impact on its performance. To solve this problem, Nguyen et al. propose to duplicate the first SYN packet [29]. The first received SYN/ACK is determined the best network (i.e., the one with the smallest RTT) to initialize the MPTCP connection.

MPTCP can reduce the delay of sub-flow establishment by supporting TCP Fast Open (TFO) [30, 31]. Szilágyi [32] changes MPTCP's sequential connection establishment procedure to simultaneous sub-flow creation. Joshi et al. [33] enable the quick negotiation of sub-flows between MPTCP hosts during its session establishment that reduces the time required for sub-flow negotiation down to one RTT.

Handshakes may be lost and retransmitted multiple times due to handoffs, consuming considerable amount of energy. Therefore, it is necessary to detect a handoff, and open sub-flows only after a handoff to preserve battery lifetime of mobile devices.

6.2 Congestion Control

MPTCP always attributes packet loss to congestion, hence conducting very aggressive congestion control when heavy packet drops occur due to handoff. MPTCP should differentiate between congestion and handoff. For example, Sinky et al. detect handoff with cross-layer assistance [34]. Besides, the additive increase and multiplicative decrease approach used by MPTCP Reno, OLIA and LIA cannot adapt fast enough to the rapid network capability variation on HSRs. We suggest more adaptive solution to increase/decrease the CWND or sending rate, for example, the approach used by BBR [35].

We suggest a coupled congestion control that is able to make good use of the difference in handoff time between multiple carriers. For example, MPTCP can pause the timeout timer and freeze data transmission of a path suffering a handoff. At the same time, MPTCP allows CWND of other paths not suffering a handoff to rise much faster than usual. Using such a coupled congestion control, traffic can be transferred timely and accurately from paths suffering a handoff to paths not suffering a handoff. This would ensure that total throughput of all sub-flows can keep relatively stable even when any

path suffers a handoff, and the resistance of MPTCP to frequent handoff is enhanced.

6.3 Scheduling

MPTCP's default scheduler (minRTT) sends packets through the available path with the smallest estimated RTT. However, when RTTs in separate paths differ, the default scheduler will cause an out-of-order arrival of packets at the receiver side. Shi et al. propose and implement a scheduler that pre-allocates packets to send over the fast path for in-order arrival [36]. It dynamically adapts the MPTCP-level send window based on the packets acknowledged. Guo et al. [37] proposed a balanced scheduler to move some traffic from slow sub-flows to fast sub-flows to achieve synchronous completion on receiver side.

Since packet loss rate is high during a handoff, we also suggest retransmitting packets lost in paths suffering a handoff timely, from other paths not suffering a handoff to reduce the out-of-order queue size.

Referring to previous studies, we will improve MPTCP or design a new more adaptive protocol, and measure its performance in the future. Other interesting topics, such as localization and detection with wireless signals [38–40] will also be studied in high speed mobility cases.

7 RELATED WORK

We categorize the most relevant literature into two groups: TCP in high speed mobility cases and MPTCP in static and low speed mobility cases.

Single-path TCP in High speed Mobility Cases. With regard to scenarios with speeds of 200 km/h or more, Merz *et al.* [1] test an LTE system in a train with velocities up to 200 km/h. Xiao *et al.* [3] carry out a measurement study in mobile data networks under high speed mobility (300 km/h), and Li *et al.* [5] investigate TCP behavior on HSR with a peak speed of 310 km/h. All these measurements obtain a conclusion that TCP's performance declines greatly in high speed mobility cases.

MPTCP in Static and Low speed Mobility Cases. Paasch *et al.* [11] enable smooth handovers for WiFi and Cellular and prove MPTCP's effectiveness in the current Internet. Chen *et al.* [12] compare the latency measurements under MPTCP and the single-path TCP, indicating that MPTCP reduces the variability in download latencies. Williams *et al.* [13] conduct experiments to augment cellular 3G connections and find that MPTCP provides benefits in the vehicle-based field test (40 km/h). Above studies are all conducted in the static or low

speed environments. In this paper, we fill a void by making a detailed measurement study on MPTCP via multiple carrier networks in high speed mobility scenarios.

8 CONCLUSION

For the first time, we perform a comprehensive large scale measurement on MPTCP with two cellular carriers on HSRs with a peak speed of 310 km/h. During a 22-month period, we accumulated a mileage of 82,266 km, collecting nearly 2.8 TB of data. To overcome the challenge that multiple external and internal influence factors are highly varying and intertwined together, we make significant methodological contributions in both measurement and analysis. We find that, MPTCP can provide a performance much better than TCP in the poorer path, but performs worse than TCP in the better path most of the time. This indicates that MPTCP can show a significant advantage in robustness, but its efficiency is far from satisfactory. We find that the low efficiency can be attributed to poor adaptability to frequent handoff by MPTCP's key operations in sub-flow establishment, congestion control and scheduling. Finally, we discuss possible directions on improving MPTCP for high speed mobility scenarios.

REFERENCES

[1] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. Performance of LTE in a high-velocity environment: A measurement study. In *Proceedings of ACM AllThingsCellular*, pages 47–52, 2014.

[2] Keon Jang, Mongnam Han, Soohyun Cho, Hyung-Keun Ryu, Jaehwa Lee, Youngseok Lee, and Sue Moon. 3G and 3.5G Wireless Network Performance Measured from Moving Cars

and High-Speed Trains. In *Proceedings of ACM MICNET*, pages 19–24, 2009.

[3] Qingyang Xiao, Ke Xu, Dan Wang, Li Li, and Yifeng Zhong. TCP Performance over Mobile Networks in High-speed Mobility Scenarios. In *Proceedings of IEEE ICNP*, pages 281–286, 2014.

[4] Li Li, Ke Xu, Dan Wang, Chunyi Peng, Qingyang Xiao, and Rashid Mijumbi. A Measurement Study on TCP Behaviors in HSPA+ Networks on High-speed Rails. In *Proceedings of IEEE INFOCOM*, pages 2731–2739, 2015.

[5] Li Li, Ke Xu, Dan Wang, Kai Zheng, Chunyi Peng, Rashid Mijumbi, and Qingyang Xiao. A Longitudinal Measurement Study of TCP Performance and Behavior in 3G/4G Networks Over High Speed Rails. *IEEE/ACM Transactions on Networking*, 25(4):2195–2208, 2017.

[6] Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley. Design, implementation and evaluation of congestion control for multipath TCP. In *Proceedings of Usenix NSDI*, pages 1–14, 2011.

[7] Alan Ford, Costin Raiciu, Mark Handley, Sebastien Barre, and Janardhan Iyengar. Architectural guidelines for multipath TCP development. RFC 6182, 2011.

[8] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. How hard can it be? Designing and implementing a deployable multipath TCP. In *Proceedings of Usenix NSDI*, pages 29–29, 2012.

[9] Multipath TCP in the Linux Kernel. http://www.multipath-tcp.org.

[10] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. Improving datacenter performance and robustness with multipath TCP. In *Proceedings of ACM SIGCOMM*, pages 266–277. ACM, 2011.

[11] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. Exploring mobile/WiFi handover with multipath TCP. In *Proceedings of ACM CellNet*, pages 31–36, 2012.

[12] Yung-Chih Chen, Yeon-sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili, and Don Towsley. A measurement-based study of multipath tcp performance over wireless networks. In *Proceedings of ACM IMC*, pages 455–468, 2013.

[13] Nigel Williams, Prashan Abeysekera, Nathan Dyer, Hai Vu, and Grenville Armitage. Multipath TCP in Vehicular to Infrastructure Communications. CAIA Technical Report 140828A, 2014.

[14] Fengyu Luan, Yan Zhang, Limin Xiao, Chunhui Zhou, and Shidong Zhou. Fading Characteristics of Wireless Channel on High-Speed Railway in Hilly Terrain Scenario. *International Journal of Antennas and Propagation*, 2013:1–9, 2013.

[15] AliCloud ECS. https://www.aliyun.com/.

[16] GsmCellLocation. https://developer.android.com/reference/android/telephony/gsm/GsmCellLocation.html.

[17] Location. https://developer.android.com/reference/android/location/Location.html.

[18] TelephonyManager. https://developer.android.com/reference/android/telephony/TelephonyManager.html.

[19] S Floyd and T Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582, 1999.

[20] Costin Raiciu, Mark Handley, and Damon Wischik. Coupled congestion control for multipath transport protocols. RFC

6356, 2011.

[21] Ramin Khalili, Nicolas Gast, Miroslav Popovic, and Jean yves Le Boudec. Opportunistic linked-increases congestion control algorithm for mptcp. Internet-Draft draft-khalili-mptcp-congestion-control-02, Internet Engineering Task Force, 2013.

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

[23] Ramin Khalili, Nicolas Gast, Miroslav Popovic, Utkarsh Upadhyay, and Jean-Yves Le Boudec. MPTCP is not pareto-optimal: performance issues and a possible solution. In *Proceedings of ACM CoNEXT*, pages 1–12, 2012.

[24] Qiuyu Peng, Anwar Walid, Jaehyun Hwang, and Steven H. Low. Multipath TCP: Analysis, design, and implementation. *IEEE/ACM Transactions on Networking*, 24(1):596–609, 2016.

[25] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. In *Proceedings of ACM SIGCOMM*, pages 363–374, 2013.

[26] Feng Qian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, Oliver Spatscheck, and Walter Willinger. TCP Revisited: A Fresh Look at TCP in the Wild. In *Proceedings of ACM IMC*, pages 76–89, 2009.

[27] TCPDUMP. http://www.tcpdump.org/.

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

[29] Kien Nguyen, Kentaro Ishizu, Mirza Golam Kibria, and Fumihide Kojima. An Improvement of MPTCP Initialization. Internet-Draft draft-kien-mptcp-init-00, Internet Engineering Task Force, 2016.

[30] Sebastien Barre, Gregory Detal, Olivier Bonaventure, and Christoph Paasch. TFO support for Multipath TCP. Internet-Draft draft-barre-mptcp-tfo-03, Internet Engineering Task Force, 2018.

[31] Yuchung Cheng, Jerry Chu, Sivasankar Radhakrishnan, and Arvind Jain. TCP Fast Open. RFC 7413, December 2014.

[32] Péter Szilágyi. Faster MPTCP sub-flow establishment for interactive applications. In *Proceedings of IEEE PIMRC*, pages 1–7, 2017.

[33] Kalpana D Joshi and Kotaro Kataoka. SFO: SubFlow Optimizer for MPTCP in SDN. In *Proceedings of IEEE ITNAC*, pages 1–6, 2016.

[34] Hassan Sinky, Bechir Hamdaoui, and Mohsen Guizani. Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks. *IEEE Transactions on Wireless Communications*, 15(7):4754–4764, 2016.

[35] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. BBR: Congestion-based congestion control. *ACM Queue*, 14(5):50, 2016.

[36] Hang Shi, Yong Cui, Xin Wang, Yuming Hu, Minglong Dai, Fanzhao Wang, and Kai Zheng. STMS: Improving MPTCP Throughput Under Heterogeneous Networks. In *Proceedings of Usenix ATC*, 2018.

[37] Yihua Ethan Guo, Ashkan Nikravesh, Z. Morley Mao, Feng Qian, and Subhabrata Sen. Accelerating Multipath Transport Through Balanced Subflow Completion. In *Proceedings of ACM MobiCom*, pages 141–153, 2017.

[38] Fu Xiao, Zhongqin Wang, Ning Ye, Ruchuan Wang, and Xiang-Yang Li. One more tag enables fine-grained RFID localization and tracking. *IEEE/ACM Transactions on Networking*, 26(1):161–174, 2018.

[39] Hai Zhu, Fu Xiao, Lijuan Sun, Ruchuan Wang, and Panlong Yang. R-TTWD: robust device-free through-the-wall detection of moving human with WiFi. *IEEE Journal on Selected Areas in Communications*, 35(5):1090–1103, 2017.

[40] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-Level Localization with a Single WiFi Access Point. In *Proceedings of Usenix NSDI*, pages 165–178, 2016.