A First Look at Disconnection-Centric TCP Performance on High-Speed Railways

Chenren Xu, *Senior Member, IEEE*, Jing Wang, Zhiyao Ma, Yihua Cheng, Yunzhe Ni, Wangyang Li, Feng Qian, and Yuanjie Li, *Member, IEEE*

***Abstract*— High-speed rail (HSR) systems potentially provide a more efficient way of door-to-door transportation than airplane. However, they also pose unprecedented challenges in delivering seamless Internet service for on-board passengers. In this paper, we conduct the first large-scale disconnection-centric measurement study of TCP performance over LTE on HSR. Our measurement targets the main HSR route in China operating at 300/350 km/h. We performed extensive data collection obtaining 378.3 GB data collected over 56639 km of trips. Leveraging such a unique dataset, we measure important performance metrics such as TCP goodput, latency and loss rate across different congestion control algorithm, mobile carrier, and different train speed. We further develop the LTE disconnection taxonomy, and conduct a in-depth correlation study between TCP stall and LTE disconnection. Our findings reveal the networking performance on today's HSR environment "in the wild", as well as identify several root causes of performance inefficiencies, which together highlight the need to develop dedicated protocol mechanisms that are friendly to extreme mobility.**

***Index Terms*— Measurement, high-speed railway, high mobility, TCP stall, CUBIC, BBR, LTE disconnection.**

I. Introduction

RECENTLY, the rapid development of high-speed rails (HSRs) has dramatically changed the way people commute for medium-to-long distance travel. For instance, a train traveling above 300 km/h potentially provides a more efficient way of door-to-door transportation than airplane. To date, 20 countries have developed HSR to connect major cities. In China, the HSR network exceeds 29,000 km in length; in Europe, HSR even travels across international borders [1]; in USA, the HSR projects in Texas and California are under construction and expected to finish in the near future [2]. While such high mobility brings great transportation efficiency, it also poses unprecedented challenge in delivering seamless mobile Internet service for on-board passengers from the trackside broadband radio connectivity in a bottom-up fashion, *i.e.,* from disrupted cellular connectivity to misguided TCP congestion control and abrupt user experience. Existing experimental studies on HSR networking [3]–[5] focus on TCP performance measurement without digging into low-level cellular disconnection events, and thus lack in-depth cross-layer insights.

To bridge such a gap, in this paper, we conduct a disconnection-centric measurement study of TCP performance on HSR. Our measurement targets the most popular HSR route in China operating at above 300 km/h. More than 180 million passengers travel on these routes annually. Through a period of more than one year, we performed extensive data collection through controlled experiments. To our best knowledge, this is the largest HSR TCP-LTE network trace dataset – 378.3 GB data collected over 56639 km of trips.

Leveraging such a unique dataset, we begin with measuring important performance metrics for two TCP variants: CUBIC and BBR, which are state-of-the-art transport layer solutions that have registered real-world deployment. We found that the extreme mobility of HSR effectively degrades the performance of these protocols, across all metrics. For instance, when the train speed increases from 300 km/h to 350 km/h, the average goodput of CUBIC and BBR decreases by 47.5% and 40.1%, respectively. Meanwhile, BBR still holds its native property of low(er) RTT, loss rate and bytes-in-flight albeit longer flow stall percentage (when in comparison to CUBIC) in such extreme mobility environment.

Given the extremely high mobility on HSR train, we identify that not only handovers happen frequently, whose average interval is around 10 seconds, incurring disconnection to LTE cells from time to time, but also in more than 10% of the cases the handover is not successfully performed. To better understand the unstable nature of the LTE connectivity on HSR, we develop taxonomy (with four types) of disconnection to cellular network which extends from (successful) handover to the handover failure and radio link failure associated scenarios. We define *disconnection time* as the time needed to reestablish connection to LTE basestations. To further quantify the impact of LTE disconnection to upper layer data plane transmission stall in LTE networks, we define *disruption time* as the time duration without any user traffic transmission on PDCP layer. Our key findings are that disruption time is

Manuscript received December 11, 2019; revised March 15, 2020; accepted April 28, 2020. Date of publication July 2, 2020; date of current version November 20, 2020. This work was supported in part by the National Key Research and Development Plan, China under Grant 2016YFB1001200, in part by the National Natural Science Foundation of China under Grant 61802007, in part by the 111 Project under Grant B14025, and in part by Microsoft Research Asia. *(Corresponding author: Chenren Xu.)*

Chenren Xu, Jing Wang, Zhiyao Ma, Yihua Cheng, and Yunzhe Ni are with the Department of Computer Science and Technology, Peking University, Beijing 100871, China (e-mail: chenren@pku.edu.cn).

Wangyang Li is with the Department of Computer Science, The University of Texas at Austin, Austin, TX 78712 USA.

Feng Qian is with Computer Science and Engineering Department, University of Minnesota—Twin Cities, Minneapolis, MN 55455 USA.

Yuanjie Li is with Hewlett Packard Laboratories, Palo Alto, CA 94304 USA.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2020.3005486

7.5x on average larger than disconnection time. Specifically, disruption time is about 150 ms for successful handover, and can last several seconds for other types of disconnection.

We then conduct a disconnection-centric TCP-LTE analysis to understand the root cause of TCP stall over HSR networking environment. Our key observations are as follows: First, 41.2% of TCP stalls are associated with LTE disconnection event, and a successful handover introduces less than half of the stall likelihood and stall time than other types of long disconnection; Second, stall time is at least hundreds of milliseconds longer than the disconnection itself; Finally, different types of disconnection tend to cause different causes of stalls.

Our key contributions are summarized as follows:

- We perform the first large-scale TCP-LTE HSR networking measurement study covering the comparative TCP variants and disconnection-centric aspects.
- We develop the first LTE disconnection taxonomy, and conduct a in-depth correlation study between TCP stall and LTE disconnection.
- We release the largest HSR TCP-LTE network trace dataset as well as the disconnection-centric TCP stall diagnosis tool MobiStallDiag to the public.

As a remark, we believe our study provides key insights for (cross-layer) protocol design dedicated for high mobility data networking in general, and even future standards such as LTE-railway (LTE-R) [6], a new standard being discussed for the LTE-based next-generation private HSR communication protocol dedicated to improve communication quality especially for mission-critical applications between trains and trackside operators. Specifically, the lessons we learned from our disconnection-centric measurement study can be also applied to 5G network as it shares the similar handover and random access protocol with LTE standard.

II. Background

A. Why LTE Is Not Good Enough for HSR

LTE is a 3GPP standard for broadband wireless communication for mobile devices. While it typically provides seamless mobile networking performance for clients on highways or regional trains (*i.e.,* below 200 km/h), it runs into severe performance issues when the client mobility is raised to a higher level. According to TR 25.913 [7], *"Mobility across the cellular network shall be maintained at speeds up 350 km/h, yet the performance is not guaranteed."* There are two major reasons behind it – poor link quality and frequent handover.

Link quality on HSRs becomes poorer than usual mainly because of the larger Doppler spread, which is proportional to the relative velocity between the train and base station. As the mobility level increases, the varying Doppler spread and channel coherence time will incur higher channel estimation errors because of the carrier frequency offset and intercarrier interference [8], [9]. As a result, it not only causes higher decoding errors, but also makes a cell choose more conservative modulation scheme and coding rate, which together lower the PHY data rate and throttle TCP throughput during the periods even without handover. Another side effect of worse signal quality is that it reduces the actual on-track LTE coverage, increases the packet loss rate, and hence imposes extra challenges for handover to finish within the overlap zone.

Handovers on HSRs become an important cause of TCP disruption – the increasing mobility level can make handovers more likely to fail because of the following reasons. First, as the link quality degrades, the handover control signal might get lost and incur high overhead to recover. Second, the handover procedure is more likely to fail given the shorter time window within the radio overlap zone of two cells due to high mobility. Third, the "tidal effect" can easily saturate the basestation, in both control and data channels. Upon failure, it needs to spend extra time in discovering and reconnecting to a cell, during which TCP is choked. In a nutshell, handover can lead to complicated interactions between mobile and cellular infrastructure as well as disconnection scenario. We will define a disconnection taxonomy (§V-A) based on our observation from the dataset collected over HSR.

B. TCP Primer

We choose CUBIC and BBR as the two TCP variants for our measurement study, and briefly introduce the necessary background on how they deal with the network dynamics.

CUBIC modifies the linear window growth function of existing TCP standards to be a cubic function. When a loss event happens, CUBIC registers the current congestion window ($cwnd$) as W_{max} and performs a multiplicative decrease of $cwnd$ by a scaling factor. The cubic function is set to have its plateau at W_{max} and its increasing is based on elapsed time instead of reception of ACK – thus the window growth is independent of RTT. After CUBIC enters congestion avoidance phase from fast recovery, it starts to increase the window using the concave profile of the cubic function until $cwnd$ becomes W_{max}. After that, the cubic function turns into a convex profile to ensure that the window increases very slowly at the beginning and gradually increases its growth rate to probe aggressively for additional capacity. This style of window adjustment (*i.e.,* concave and then convex) makes the $cwnd$ remain almost constant around W_{max}, improves network utilization and scalability of TCP over fast and long distance (*i.e.,* large bandwidth-delay product) networks, and meanwhile treats other TCP connections fairly. However, the fact that it treats packet loss over a lossy wireless link as the signal of network congestion can throttle its $cwnd$ by mistake thus leads to low bandwidth utilization.

BBR employs two parameters, namely *RTprop* (*i.e.,* round trip propagation time estimated by taking the minimum RTT over the last 10 seconds) and *BtlBw* (*i.e.,* bottleneck bandwidth estimated by taking the maximum throughput over the last $10{\cdot}RTprop$), to model the end-to-end network capacity and determine its $cwnd$, *e.g.,* $2\ BtlBw \cdot RTprop$ at most of the time. Specifically, BBR first uses the slow-start akin to CUBIC's only when the flow is initially launched and then soon reaches its bandwidth probing phase after the throughput converges. In this phase, it takes a period-8 cycling $pacing\_gain$ sequence $(1, 1, 1, 1, 1, 1, 5/4, 3/4, \dots)$ in turn as a multiplier to *BtlBw* to determine its sending rate for *RTprop* time – while $pacing\_gain = 1$ at most of the time, a $pacing\_gain > 1$ means BBR is in the phase of exploring

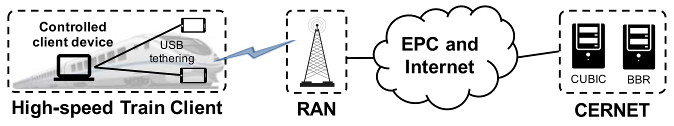


Fig. 1. Our experimental testbed collects data from a dedicated laptop-phone suite.

more bandwidth, after which a $pacing\_gain < 1$ is necessary to guarantee that the queue at the bottleneck will be drained in case there is no more bandwidth to utilize. If $RTprop$ is not updated for 10 seconds (*i.e.,* RTT is not get smaller ever), BBR will enter ProbeRTT phase to drain the queue by setting $cwnd$ to 4, and use their minimum RTT as new $RTprop$. The takeaway message is that, BBR will self-throttle bandwidth immediately upon an RTT increasing trend, and is more conservative in bandwidth growth and robust to random packet loss than CUBIC.

We note there are many alternative TCP variants in the wild, which can be categorized into loss-based [10], [11] and delay-based [12]–[14] congestion control algorithm in general. We choose CUBIC and BBR in our study because they both not only have large-scale real world deployment, but also represent the state-of-art solution in each category – CUBIC provides the best goodput over high-BDP networks [15], and BBR in a sense can be regarded as a delay-based approach as it also aims to keep the delay short and even outperforms CUBIC by 2 to 25x in WAN environments [16]. Meanwhile, we are aware that there are recent designs dedicated for cellular access [17]–[23]. We leave a comprehensive study for future work.

III. Measurement Methodology

A. Experimental Setup

In order to demystify the performance issues and optimization opportunities in mobile networking on HSR, the experimental setup should facilitate collecting data from on-board controlled experiments to gain insights in the following two dimensions: 1) How do different TCP congestion control algorithms behave (on different mobile carriers)? 2) What is unique about the interaction between TCP and LTE in high mobility environment?

1) Server: We deploy two co-located servers (Intel NUC6i7KYK with i7-6770HQ, 32 GB DDR4 and Samsung 950 pro 512 GB) in CERNET [24], the nationwide education and research computer network in China.

2) Client: We tether two Android phones (Xiaomi 5s) to one laptop (Dell XPS 13-9360) via USB. This tethered setup allows us to run two experiments simultaneously on the two phones, which appear as network interfaces on the laptop and function as link-layer devices. We modified the tethering code in Android OS to provide such multihoming support. The phones are equipped with SIM cards of two mobile carriers in China, denoted as Carrier A and Carrier B.

B. Experimental Design

Our high-level experimental methodology is to perform bulk data download over TCP by iPerf. We next detail several important design aspects in terms of what and how to measure.

1) Flow Size: We measure two types of TCP downlink flows, including long flows (*i.e.,* fixed duration of 150 seconds) and fix-sized flows of 64 KB (corresponding to typical web page). We believe the TCP behavior in long flows presents a wide variety of Internet contents, ranging from several MB (*i.e.,* web contents such as images) to tens of MB such as HD video chunks. These are typical workloads of HSR networking.

2) TCP Variants Comparison: The two co-located servers run Ubuntu 17.04 with kernel 4.10.17 with CUBIC and BBR respectively. We compare their performance and their incurred TCP-LTE interactions.

3) TCP-LTE Interaction: We run tshark on both client and server to collect packet-level TCP traces. We also instrument the client phones using MobileInsight [25][1] to collect and parse LTE handover and disconnection events.

C. Data Collection and Preprocessing

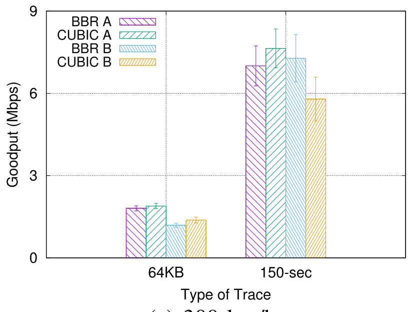
We carried out experiments on the Beijing-Shanghai (300/350 km/h) HSR route as it represents the state-of-art HSR networking environments in terms of train speed and track-side cellular infrastructure. Note that we choose these two speed because the trains travel at speed of either 300 or 350 km/h as stable state most (*i.e.,* 95%) of the time. In other words, the speed below that is both transient and short, thus not representative of the mobile networking condition for on-board passengers. We collected 378.3 GB data by traveling 56639 km on the trains. Since TCP-LTE performance may vary along the route because of the terrain diversity [26] and LTE cell density, we collected the data over the whole route without temporal or spatial sampling. We note that one straightforward way to eliminate the impact of this factor is to log the GPS reading to perform location-aware analysis. However, in our experiments the phone failed to report GPS data at most of the time due to magnetic-shielding from the sealed carriages. After obtaining this unique dataset, we developed a HSR TCP-LTE performance diagnosis software tool called MobiStallDiag to perform numerous types of data processing such as extracting TCP flows and LTE events, calculating various performance metrics, and aligning TCP traces with LTE events for cross-layer TCP stall analysis. Specifically, based on the handover and MAC random access control message and user plane PDCP header, we extract two important types of LTE event, namely LTE disconnection and LTE disruption, which will be introduced in §V-B. We release the dataset and MobiStallDiag used for this study in [27].

IV. Performance of TCP Variants

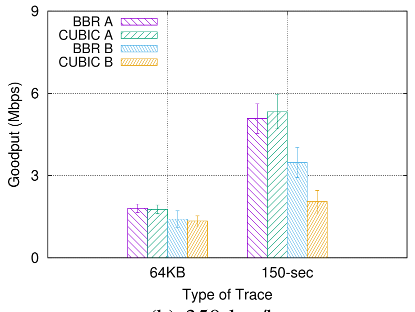
A. Basic TCP Performance Metrics

We first utilize the measurement data to study key connection-level performance metrics including goodput, bytes-in-flight (BiF), round trip time, packet loss rate, and out-of-order delay. In particular, we investigate how TCP congestion control algorithm (CCA) affects the above metrics.

[1]Android-based in-device software tool that collects runtime network information and exposes protocol messages on both control plane and (below IP) data plane from the 3G/4G chipset from operational cellular networks.



(a) 300 km/h



(b) 350 km/h

Fig. 2. Goodput.

TABLE I
CRITICAL STATISTICS OF BBR/CUBIC PERFORMANCE METRICS OVER DIFFERENT CARRIERS AT THE SPEED OF 350 KM/H

Metrics	Mean	Median	95% percentile	99% percentile
Goodput (Mbps)	5.12/5.40	4.97/4.78	11.85/13.06	14.60/16.29
BiF (MB)	0.22/1.53	0.18/1.57	0.52/2.91	0.84/3.14
RTT (ms)	457.42/1707.2	148.63/345.02	1491.7/8272.1	6022.9/18760.7
PLR (%)	1.38/5.92	0.41/ 4.53	4.64/14.79	20.18/17.77
OOD (ms)	4.44/77.75	0/0	0/66.62	79.79/2244.2

(a) Carrier A

Metrics	Mean	Median	95% percentile	99% percentile
Goodput (Mbps)	3.54/2.06	2.65/1.02	11.20/7.58	12.29/10.24
BiF (MB)	0.28/1.08	0.19/1.06	0.82/2.62	1.66/3.01
RTT (ms)	269.91/2193.3	141.42/1067.86	798.41/7454.55	2095.70/18934.9
PLR (%)	2.13/3.99	0.64/2.25	8.93/11.47	16.04/24.57
OOD (ms)	42.47/95.62	0/0	2.56/170.08	740.82/2565.08

(b) Carrier B

1) Goodput: Fig. 2 plots the goodput of downloading different files under two speeds (300 km/h and 350 km/h) for Carrier A and Carrier B. We consider two workloads: short flows (64 KB) and long-lived bulk download flows lasting for 150 seconds. As shown, neither the CCA nor the carrier appears to significantly affect the performance of short flows, which mostly finishes within the slow start stage during which the available bandwidth is under-utilized. For the long flow (150 seconds), we make two key observations. First, as the speed increases from 300 km/h to 350 km/h, the goodput of CUBIC and BBR decrease by 47.5% and 40.1%, respectively. This is attributed to the lower PHY rate caused by the imperfect radio receiver design encountering Doppler Spread in high mobility. Second, when compared to CUBIC, BBR yields marginally lower goodput over Carrier A, as CUBIC is known to expand its congestion windows (and bytes-in-flight) aggressively. Over Carrier B, however, BBR yields higher goodput (25.79% higher at 300 km/h and 70.19% higher at 350 km/h) compared to CUBIC. This is because Carrier B has higher random loss rate in our another independent experiment,[2] which is infrastructure-dependent. Such random losses force CUBIC to (more) frequently back off while bring much smaller impact on BBR, which does not rely on random packet losses for modeling the network capacity.

For the sake of space, we will focus on Carrier A for the rest of the metrics not only because both carriers exhibit similar pattern in terms of comparative performance across CCA and mobility level, but also Carrier A is the most popular local carrier. We summarize the critical statistics for both carriers in Tab. I by the end of this section.

[2]We carried the same controlled experiment setup, let each TCP packet carrying 1 byte of data and sends at a stable rate of 20 packets per second to avoid self-inflicted congestion. The random loss rate of CA and CB are 0.21% and 1.35% respectively.

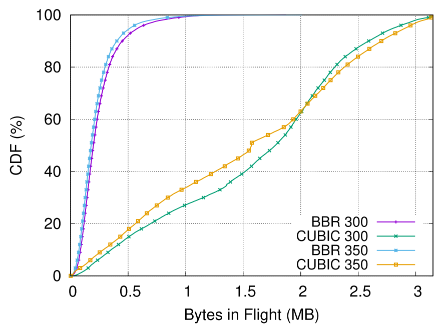


Fig. 3. BiF (Carrier A).

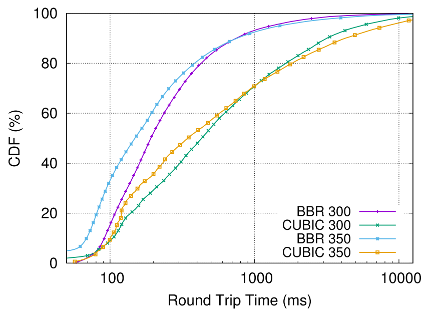


Fig. 4. RTT (Carrier A).

2) Bytes-in-Flight (BiF): As shown in Fig. 3, BBR yields almost an order of magnitude lower BiF than CUBIC (*e.g.,* 0.18 MB versus 1.57 MB for median value). This cross-validates the RTT difference between BBR and CUBIC shown in Fig. 5, as a large BiF incurs high queuing delay

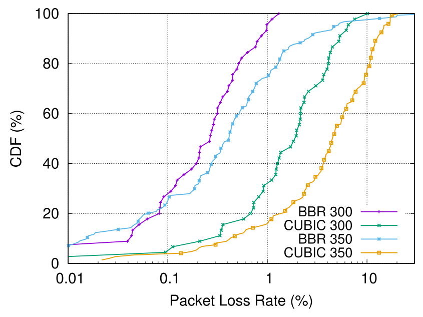


Fig. 5. PLR (Carrier A).

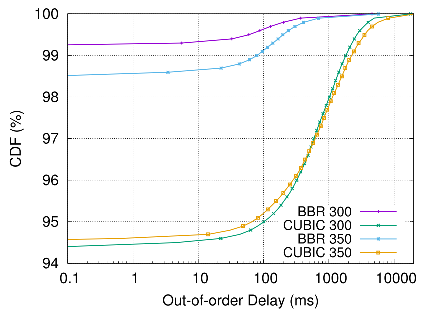


Fig. 6. OOD (Carrier A).

that inflates the RTT [17]. As the mobility level increases, the BiF oftentimes decreases due to reduced throughput. In fact, we found that higher mobility causes only marginal impact on both CUBIC and BBR. However, we observe that for CUBIC, the BiF can sometimes increase to 3 MB at 350 km/h. This is explained by the higher likelihood of an uplink ACK packet being delayed or lost, causing a "spuriously inflated" BiF.

3) Round-Trip-Time (RTT): As shown in Fig. 5, BBR has more than twice lower RTTs than CUBIC (*e.g.,* 191.53 ms versus 431.35 ms at 300 km/h, and 148.63 ms versus 345.02 ms at 350 km/h for median value) due to their different CCA design rationales: BBR intends to suppress the RTT to overcome the bufferbloat problem [28]. The increase of mobility level affects the RTT in two aspects. On one hand, more frequent handover and higher packet loss rate (Fig. 5) lengthen the RTT, especially contribute longer tails; on the other hand, when traveling faster, the CCA dictates the server to send data slower, which oftentimes leads to reduced the in-network buffer occupancy level (as well as queuing delay) and henceforth the RTT. Note that low RTT (*e.g.,* less than 200 ms) is critical to meet the QoE requirement for popular network applications such as teleconferencing and gaming for on-board passengers.

4) Packet Loss Rate (PLR): As shown in Fig. 5, BBR has about an order of magnitude lower PLR than CUBIC (*e.g.,* 0.27% versus 1.95% at 300 km/h, and 0.41% versus 4.53% at 350 km/h for median value). This is because BBR is designed to keep RTT or queuing delay low to avoid tail-drop in the buffer inside the network. As the mobility level increases, PLR increases because of more (unsuccessful) handovers and decoding errors.

5) Out-of-Order Delay (OOD)[3]*:* As shown in Fig. 6, we found that BBR has much fewer packets with OOD than CUBIC (*i.e.,* 0.80% versus 5.68% at 300 km/h, and 1.53% versus 5.48% at 350 km/h), primarily because of its lower RTT and PLR. Regarding long tail aspect, CUBIC has a much more serious issue: 95% and 98% percentile can reach 100 ms and 1 second respectively, which can significantly affect the QoE. From the mobility level perspective, it only incurs marginal impact in our measurements.

[3]The OOD of a packet is measured as the difference of the time between the arrival of a packet at the receive buffer and that of its previous packet [29]. It normally does not affect throughput but goodput because most applications require in-order data delivery.

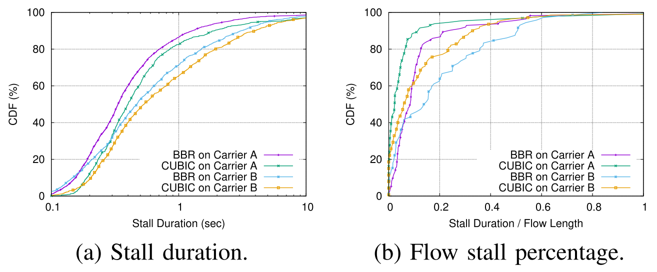


Fig. 7. TCP stall profiling.

B. TCP Stall Measurement

On HSR, TCP flow can encounter frequent disruptions, which in turn significantly degrade user experience. To quantify TCP disruption, we leverage the concept of *TCP Stall* [30] defined as an event where the duration between two consecutive packets received or sent by the sender is larger than min(τ·SRTT, RTO)[4]. We used the TAPO toolkit developed in [30] to analyze TCP stall events based on our 150-sec flows data collected from train traveling at 350 km/h.

1) Stall Duration: As shown in Fig. 7a, we make three main observations. First, flows over carrier A have shorter TCP stall time than carrier B. This is because carrier A has a smaller random loss rate as mentioned before. Second, BBR leads to statistically shorter stall than CUBIC. This is because stalls are identified based on SRTT – BBR has shorter RTT than CUBIC and thus it is easier to meet the stall definition when streaming packets under such highly dynamic networking environment. Therefore, BBR will have more frequent but shorter stall than CUBIC. Third, the discrepancy of carriers has a larger impact on stall time than CCA – there are 13.5% and 17.4% of the stalls that are longer than 1 second for carrier A with BBR and CUBIC respectively, while the number of that for carrier B is 28.6% and 34.5%. Overall, we observe an average of 21.0, 6.5, 19.7 and 11.1 stalls in each flow from BBR and CUBIC of carrier A, and BBR and CUBIC of carrier B, respectively.

2) Flow Stall Percentage: We further show the flow stall percentage (*i.e.,* total stall duration divided by flow duration) in Fig. 7b. Interestingly, we find that CUBIC has shorter one than BBR. This is because CUBIC experiences less stalls, *i.e.,*

[4]Here the Smoothed RTT (SRTT) and RTO are calculated according to RFC 6298 [31] as implemented in the Linux kernel. Similar to [32], we set τ to 2, as under normal circumstances, a TCP sender should be able to receive or send at least one packet during 2 RTTs.

69% and 44% of BBR's on carrier A and carrier B respectively. We also observe that there are 10% and 20% of the flows over CUBIC and BBR of carrier B experiencing TCP stall in 40% of the time, which can significantly user experience. However, we would like to point out that BBR flow with longer stall time might still have better goodput than CUBIC (with shorter stall) because of less packet loss, shorter RTT, and thus more efficient bandwidth utilization.

C. Summary of Key Findings

Our study shows that increasing the speed from 300 km/h to 350 km/h reduces the TCP goodput by above 40% and increases the loss rate by up to 92.97%, while does not significantly affect the RTT; in a high mobility environment, BBR performs reasonably well by preserving its key advantages (compared to CUBIC) such as being robust to random losses and incurring a smaller amount of BiF to potentially mitigate the bufferbloat issue, leading to less packet loss and shorter RTT, and thus more efficient in network utilization despite the fact that it has longer TCP flow stall percentage.

V. From Handover to Disconnection

When mobile clients are in (extreme) high mobility, they encounter more frequent handover and link disconnection than usual, ultimately leading to TCP stall and abrupt user experience. Our goal is to provide a comprehensive disconnection taxonomy based on all the cases we observe from our dataset.

A. Disconnection Taxonomy

In cellular networks, handover is a common control plane procedure used to support client mobility. As specified in 3GPP TR 36.331 [33], a mobile client or user equipment (UE) periodically measures the signal strength of its serving cell (sCell) and neighboring cells, and report to the sCell if the signal strength exceeds any threshold configured by the sCell, *e.g.,* one of the neighboring cell has stronger signal than the sCell. A successful handover is defined as a procedure that the UE receives the Handover Command message from sCell, delivers a Handover Complete message to the instructed target or new cell (nCell) and stays connected after a successful MAC random access procedure completes. However, such handover control message transmission can be unreliable, leading to handover failure or radio link failure cases if Handover Complete or Handover Command message gets lost respectively. In these two failure cases, the UE context is successfully transferred from sCell to nCell, and UE can still successfully perform an RRC connection re-establishment. In another case, UE may suddenly lose the signal and connection, leaves the sCell unprepared for handover, and hence leads to RRC re-establishment failure and a consequent new RRC connection establishment, *i.e.,* a *non-access stratum and recovery* procedure. Note that for both RRC connection establishment and re-establishment, MAC random access procedure and RRC Connection Reconfiguration used for data bearer setup are needed before the data service.

We formally define LTE disconnection taxonomy as the following four types: *successful handover (HO)*, *handover failure and recovery (HOF Recovery)*, *radio link failure and recovery (RLF Recovery)*, and *non-access stratum recovery (NAS Recovery)*, as illustrated in Fig. 8. At a high level, HOF failure catches the case where UE receives a handover command, but the handover procedure does not complete successfully. In the last two cases, UE fails to receive any handover command upon radio link failure and starts reconnection immediately – they differ in that UE succeeds for the first time in the former case. In both HOF recovery and RLF recovery, UE happens to connect with the cell holding its context, which is not the case for NAS recovery. Finally, we observe that in the cases other than HO, the UE can recover connection to either the previous sCell or a nCell.

TABLE II
Disconnection Breakdown

Carrier	HO	HOF Recovery		RLF Recovery		NAS Recovery	
	nCell	sCell	nCell	sCell	nCell	sCell	nCell
A	88.2%	0.46%	0.23%	5.3%	0.80%	0.45%	4.7%
B	83.0%	0.45%	0.32%	11.4%	0.75%	0.31%	3.7%

B. Disconnection and Disruption Analysis

We start to examine the collected statistics about LTE disconnection under the mobility of 350 km/h on two carriers. As shown in Tab. II, HO represent 88.2% and 84% of the disconnection cases on carrier A and B, which means most of the disconnections are successful handovers. On the other hand, RLF and NAS recoveries happen much more frequently than HOF recovery. Specifically, we observe that when RLF recovery happens, the UE is more likely to connect back its sCell rather than a nCell. This happens when the signal strength provided by the sCell is weak in its coverage duration. In contrast, after experiencing a NAS recovery (mostly due to a sudden blackout or no coverage), UE often connects to a nCell after traveling out of the coverage by its previous serving cell.

Next, we study how long does disconnection interrupt the data service in LTE networks. We focus on two metrics:

• *Disconnection time* is determined by computing the timestamp between the first and last control message (of solid line) according to the disconnection taxonomy (Fig. 8).

• *Disruption time* is counted as the interval between the time receiving the last PDCP[5] packet containing user traffic before disconnection and the time of receiving the first user traffic PDCP packet after disconnection.

The key difference is that disconnection time is detected from the signaling between UEs and cells, while the disruption time is the perceived stall of receiving packet at PDCP layer. We show the statistics of both metrics in Fig. 9. As expected, HO incurs the smallest impact to upper layer data service, *i.e.,* 40 ms and 150 ms for disconnection time and disruption time respectively. Note that the disruption time is 2.75× larger than disconnection because it still takes a while after the UE

[5]Packet Data Convergence Protocol (PDCP) layer sits at the topmost part of the radio stack that adds the PDCP header to the incoming data and forwards to Radio Link Control (RLC) layer in downlink, or removes the PDCP header from the incoming packet and forwards it to IP layer in case of uplink.

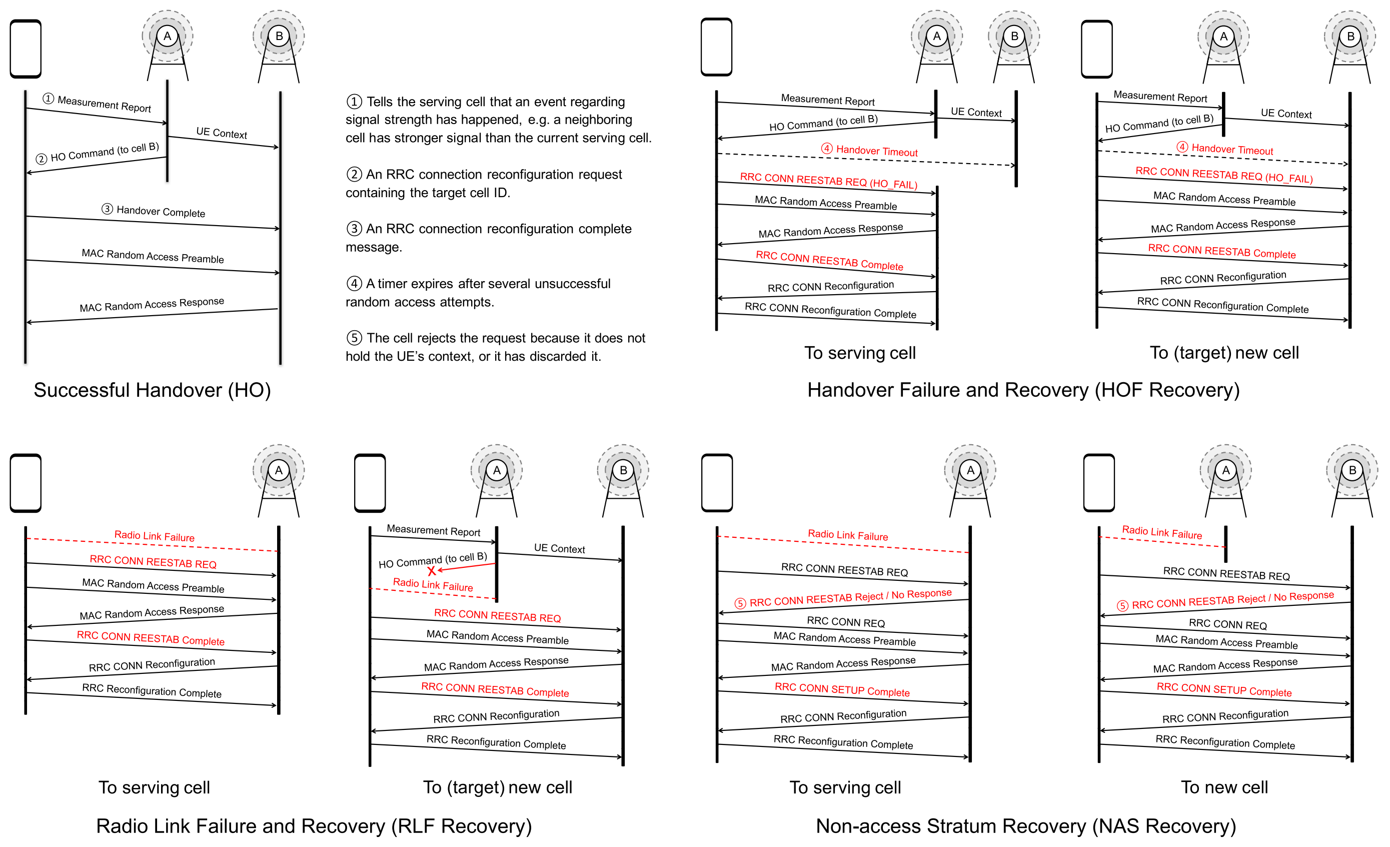
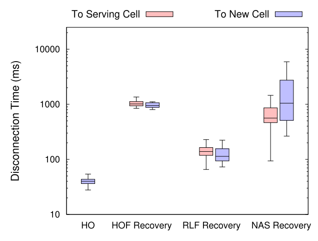
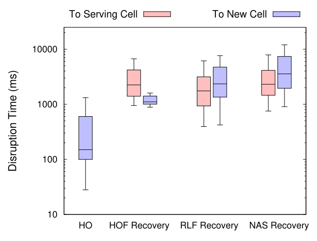

① Measurement Report
UE Context
② HO Command (to cell B)
③ Handover Complete
MAC Random Access Preamble
MAC Random Access Response
Successful Handover (HO)
① Tells the serving cell that an event regarding signal strength has happened, e.g. a neighboring cell has stronger signal than the current serving cell.
② An RRC connection reconfiguration request containing the target cell ID.
③ An RRC connection reconfiguration complete message.
④ A timer expires after several unsuccessful random access attempts.
⑤ The cell rejects the request because it does not hold the UE's context, or it has discarded it.
Measurement Report
UE Context
HO Command (to cell B)
④ Handover Timeout
RRC CONN REESTAB REQ (HO_FAIL)
MAC Random Access Preamble
MAC Random Access Response
RRC CONN REESTAB Complete
RRC CONN Reconfiguration
RRC CONN Reconfiguration Complete
To serving cell
To (target) new cell
Handover Failure and Recovery (HOF Recovery)
Radio Link Failure
RRC CONN REESTAB REQ
MAC Random Access Preamble
MAC Random Access Response
RRC CONN REESTAB Complete
RRC CONN Reconfiguration
RRC Reconfiguration Complete
To serving cell
To (target) new cell
Radio Link Failure and Recovery (RLF Recovery)
⑤ RRC CONN REESTAB Reject / No Response
RRC CONN REQ
RRC CONN SETUP Complete
To serving cell
To new cell
Non-access Stratum Recovery (NAS Recovery)

Fig. 8. LTE disconnection taxonomy illustration.



(a) Disconnection time.



(b) Disruption time.

Fig. 9. Disconnection/disruption statistics from carrier A.

reestablishes connection to the target cell before resuming data transmission. In contrast, a HOF recovery experiences much longer time, *i.e.,* more than 1 second as median value because of extra time spent on MAC random access attempts before handover timeout and RRC connection reestablishment to recover the connection to the cell. Interestingly, we observe that it takes slightly longer disconnection and disruption time for connection to sCell than nCell. This is because the connecting back (to sCell) behavior is not expected (after UE context is transferred to nCell), which might introduce extra control plane overhead. For RLF recovery, it introduces a much smaller disconnection time than HOF recovery (*i.e.,* 138 ms as median) because it does not experience the handover timeout. However, its disruption time is similar to the HOF recovery case because the data transmission already stops for a while before radio link failure happens. For NAS recovery, the median disconnection time is 689 ms, which is about 5x longer than RLF recovery due to the extra RRC connection setup overhead. Note that the NAS recovery has a long tail because the nCell does not always send reestablishment rejection back immediately and can keep UE waiting until timeout for sending RRC connection request.

Summary of Key Findings. Our study shows that HO has the lowest disconnection and disruption time (*i.e.,* 40 and 150 ms as median value) and the highest probability of occurrence (*i.e.,* nearly 90%), as expected. While RLF recovery only incurs about 100 ms of disconnection, it causes the same actual cellular data disruption time as the HOF recovery and NAS recovery cases, with more than 1 second as median value. In general, disruption time is much longer than disconnection time, *i.e.,* 3.8×, 2.1×, 13.2× and 4.2× for HO, HOF recovery, RLF recovery and NAS recovery respectively.

VI. Disconnection-Centric Cross-Layer Analysis

Our analysis in §V evidently shows that the impact of low-layer LTE link disconnection can be amplified when in upper layer (PDCP) data service disruption, let alone TCP layer, *e.g.,* TCP stall event (§IV-B). This naturally calls for request to correlate these two events at different layer and provide a in-depth analysis. In this section, we first introduce how we align LTE events with TCP stalls. Based on that, we then present our cross-layer analysis.

A. TCP Stall and LTE Disconnection Alignment

To precisely correlate the LTE disconnection and TCP stall events, the first step is to synchronize these two traces. This

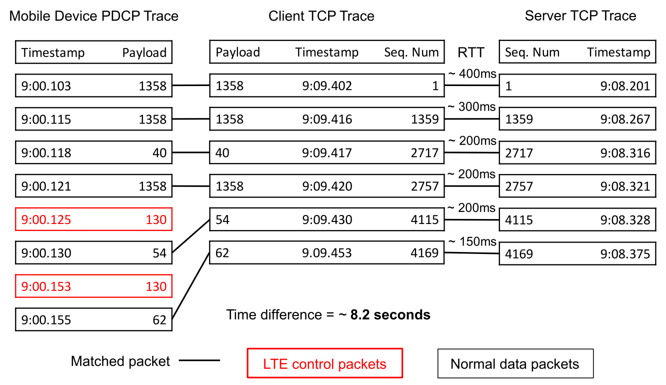


Fig. 10. Example of end-to-end alignment.

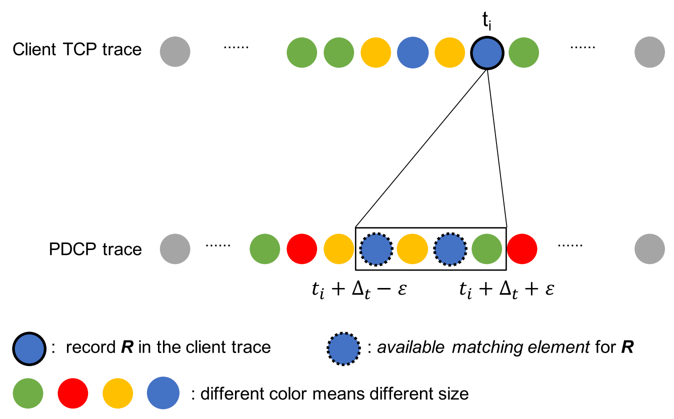

Client TCP trace
t_i
PDCP trace
$t_i + \Delta_t - \varepsilon$
$t_i + \Delta_t + \varepsilon$
: record *R* in the client trace
: *available matching element* for *R*
: different color means different size

Fig. 11. TCP and PDCP trace alignment.

procedure is conceptually simple, but non-trivial to achieve in practice. The first challenge is that the timestamps of TCP packets and LTE events are reported from two different systems, *i.e.,* Linux system on the server side and cellular chipset on the client, and the difference between them can be as high as 10 seconds observed in our data.

To address this challenge, we follow two steps to align timestamps from server and mobile device. An example of this procedure is shown in Fig. 10. We first synchronize mobile devices with our (laptop) client, and then align client and server. Our solution is based on the assumption that the clock misalignment between those devices remain constant during the testing period, *e.g.,* several minutes. The second step is straightforward because we can simply use TCP sequence number and timestamp to obtain packet-level alignment. Note that error is no more than 1 RTT (*i.e.,* hundreds of milliseconds), which will not result in a mismatch between two irrelevant TCP stall and LTE disconnection event that usually happens every more than 5 seconds on average.

To further synchronize LTE event and TCP packet at the client side, we leverage the size and timestamp of PDCP packets as the key information for time alignment because PDCP data packet carries the same payload as in TCP/IP packet[6]. Let S_{pdcp} and S_{tcp} be the time series of the packet size from PDCP and TCP traces. Our goal is to search for a Δ_t that every element $(t_i, Size_i)$ in S_{tcp} has a matched element $(t_i + \Delta_t, Size_i)$ found in S_{pdcp}. A naive approach is to enumerate all the time difference between any pair of timestamp in the two traces into a set S and find it in a brute-force manner. However, in practice, the timestamps of both PDCP and TCP traces are not perfectly accurate (as illustrated in Fig. 10) – MobileInsight incurs an internal delay of up to 100 ms to report LTE events. As a result, Δ_t might not be a constant cross all the synchronized PDCP-TCP pair. Therefore, we add the error term ϵ to relax the matching condition that the valid matching element now includes all the records in S_{pdcp} within $[t_i + \Delta_t - \epsilon, t_i + \Delta_t + \epsilon]$ that has size of $Size_i$ (as shown in Fig. 11). We then apply a simple DFS algorithm to find the smallest ϵ that ensures solution has only one Δ_t.

In real dataset, both PDCP and TCP traces have millions of records, and thus the size of S can be more than a billion. To improve efficiency, we applied two heuristics to reduce the searching space. First, we remove all Δ_t in S if $|\Delta_t|$ is larger than a threshold T, *e.g.,* 10 minutes, the maximum possible clock shift between the devices already have time synchronization. Second, we prioritize the record with size other than MTU (*e.g.,* SYN and SYNACK), which is much less common given the sending behavior of iPerf. After applying these optimizations, it only takes 30 seconds on average to finish the alignment task for a 150-second trace.

B. Disconnection-Centric TCP Stall Analysis

Now we are ready to revisit TCP stall analysis with a disconnection-centric manner. Specifically, we examine the relationship between LTE disconnection and TCP stall in terms of cooccurrence and duration, and further conduct a comparative cause analysis based on whether the stall is associated with disconnection or not.

1) Cooccurrence Analysis: From the top-down manner, we find that 41.2% of the stalls happen with a concurrent disconnection event – HO, HOF recovery, RLF recovery and NAS recovery contribute 29.4%, 0.8%, 6.74% and 4.21% of total stall events respectively. As another perspective, different type of LTE disconnection causes stall with different probability. As shown in Fig. 12, only 41.2% of the successful handovers (HO) will lead to stalls. This is because the downlink packets will be buffered at the target cell and delivered to the mobile client during and after handover respectively, which consequently minimizes the LTE data service disruption and the probability of introducing TCP packet loss or RTO. In contrast, all the other types of disconnection lead to much longer disruption, and end up with 80.0%, 81.9% and 92.1% of the chances to result in stall.

2) Duration Analysis: We further look at the duration of TCP stall caused by LTE disconnection. As shown in Fig. 13, we make three key observations. First, the stall duration is much longer than their associated LTE disconnection events, *i.e.,* duration of stalls is on average 15.2×, 1.85×, 18.96×, 1.14× the disconnection duration of HO, RLF recovery, HOF recovery, NAS recovery respectively. Second, the stall duration introduced by HOs has a median of 345 ms, which is much

[6]PDCP trace contains both data plane traffic and control plane messages.

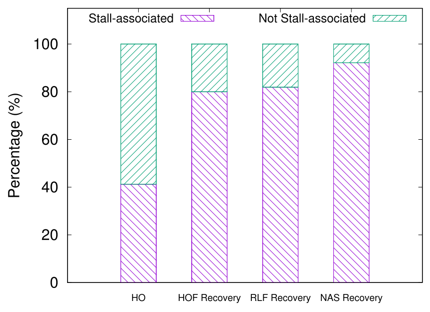


Fig. 12. Cooccurrence analysis.

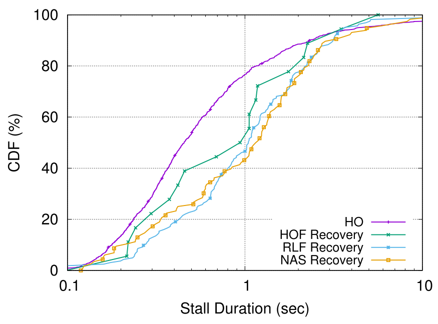


Fig. 13. Duration analysis.

smaller than that of other disconnection types. Third, RLF recovery, HOF recovery and NAS recovery exhibit statistically similar in stall duration, with a median of about 1 second.

3) Stall Cause Characterization: Beyond stall detection, TAPO can also identify the cause of stall into the following common categories, namely, delayed or lost ACK, double retransmissions, too many out-of-order packets, delayed packets, and resource constraint (*i.e.,* server does not provide new data to TCP stack when sending window is non-zero). We use TAPO to analyze the cause of all the TCP stall events associated to LTE disconnections or not respectively. As we can see in Fig. 14, disconnection-associated stalls are more likely to be caused by ACK delay or loss, double retransmission, out-of-order delay and resource constraint reasons in comparison to other stalls. In addition, HO is more likely to cause delayed or lost ACK, accounting for 91.3% of them. This is because successful handover will not drop TCP packets, but delay ACKs. Over 30% of out-of-order packets and double retransmissions are caused by the other three types. This is because prolonged disconnections are more likely to cause packet loss and out of order delivery instead of ACK delay. As a side note, the analysis aforementioned does not include the 40% stall cases that TAPO fails to report cause. With MobiStallDiag, we can identify that 60% of them are associated with an LTE disconnection event.

4) Summary of Key Findings: Our disconnection-centric stall analysis shows that: 1) HO has less than half in both probability in causing stall and caused stall duration in comparison to the other three types of long disconnection; 2) Stall duration is at least hundreds of milliseconds longer than the disconnection itself; 3) HO is more likely to create delay or

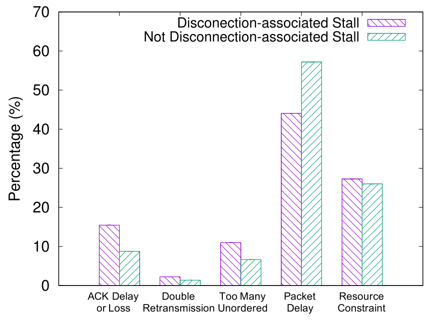


Fig. 14. Stall cause characterization.

lost ACK, while the other types of long disconnection tend to cause more packet loss and out-of-order delivery events.

VII. Discussion

A. Cross-Layer Congestion Control

Our study shows that LTE disconnections are highly related to TCP stalls, and different types of disconnections have different impacts on stalls. This opens up the door of improving network performance by using cross-layer knowledge, especially for mobile networking on HSR. For instance, once a mobile client detects the measurement report event (before a successful handover), it may inform the server that the next few ACKs will be delayed to potentially avoid unnecessary TCP retransmissions. Upon predicting an upcoming long disconnection, the server may temporarily decrease its sending rate or send duplicated packets to avoid bursty packet loss.

B. Reliable Coded Transmission

The erasure codes [34], [35] can potentially eliminate loss-associated TCP stalls by providing coded redundancy. However, such coding mechanism still have limitation in the HSR scenario because the packet loss pattern can be bursty and unpredictable, rather than random. One of the possible solution is to take advantage of the trackside carrier diversity and incorporate erasure codes with multi-carrier multipath transmission to address this problem.

VIII. Related Work

A. TCP Measurement Study on HSRs

Most prior measurement work on HSR only focused on the TCP level. The study in [36] showed that ACK compression is common and that spurious retransmission represent more than 50% retransmission. The work [3] presented the first public large-scale empirical study on TCP performance in HSR scenarios. The main observation is that the TCP throughput is much worse (3x and 2x) than static and driving scenarios, primarily because of the larger RTT jitter and variance, induced by channel loss and handover. Most recently, Li *et al.* [4] quantified TCP's poor adaptation to high mobility environments, such as high spurious RTO rate, aggressive congestion window reduction, a long delay of connection establishment and closure, and transmission interruption. In [5], they further discovered that MPTCP with coupled congestion control over multiple cellular carrier setup provides better performance than TCP in the poorer of the two paths, while performs worse than TCP in the better path most of the time. Our work differs from them in that we not only look into the LTE protocol message including L1/2 to investigate the root cause of TCP stall behavior, but also extend it to a comparative study on TCP variants (*i.e.,* CUBIC and BBR) to shed light on rethinking the protocol design for data networking in such challenging environments.

B. Cross-Layer Measurement Study on Mobile Networks

This type of work typically requires access to the low level (L1/2) information. As studied in [37], TCP performance is not significantly influenced by wireless channel data rate but rather the queuing effect primarily due to the presence of large buffers in 3G networks. The work [38] presented the first public report on a large-scale empirical study on the performance of commercial mobile HSPA (3.5G) networks. The key relevant finding is that the throughput performance does not monotonically decrease with increased mobility level when below 100 km/h. The study [39] showed that the performance of LTE remains robust up to 200 km/h and the SNR is the most important factor to ensure reliable operation in terms of higher order of modulation and coding schemes (MCS) and rank (*i.e.,* number of streams). The authors in [40] found that the high queuing delay (and its variance) in LTE networks often cause TCP congestion window to collapse upon a single packet loss, or fail to adapt fast enough and thus under-utilize the bandwidth. The work [41] revealed bursty pattern of packets arrival due to the polling duty cycle of the radio driver in mobile devices. Our work extends these findings by conducting a in-depth disconnection-centric study and quantifying its impact at different mobility level up to 350 km/h on high-speed rails.

IX. Conclusion

We perform an in-depth measurement study of HSR networking performance by examining a wide range of factors including TCP performance metrics, congestion control algorithm and LTE disconnection event. Our extensive measurement results highlight that: 1) BBR performs reasonably well by preserving its key advantages (compared to CUBIC) such as being robust to random losses and incurring a smaller amount of bytes-in-flight, and thus is more friendly to carrier and train speed diversity; 2) LTE disconnection has a strong impact on TCP stall in terms of both occurrence and duration, and it is highly desirable to avoid disconnections other than successful handover. We believe these key findings will shed light on designing dedicated link-aware congestion control strategy to optimize the performance of high mobility data networking.

References

[1] *High-Speed Rail*. Accessed: Mar. 2020. [Online]. Available: https://en.wikipedia.org/wiki/High-speed_rail

[2] *High-Speed Rail in the United States*. Accessed: Mar. 2020. [Online]. Available: https://en.wikipedia.org/wiki/High-speed_rail_in_the_United_States

[3] Q. Xiao, K. Xu, D. Wang, L. Li, and Y. Zhong, "TCP performance over mobile networks in high-speed mobility scenarios," in *Proc. IEEE 22nd Int. Conf. Netw. Protocols*, Oct. 2014, pp. 281–286.

[4] L. Li *et al.*, "A longitudinal measurement study of TCP performance and behavior in 3G/4G networks over high speed rails," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2195–2208, Aug. 2017.

[5] L. Li *et al.*, "A measurement study on multi-path TCP with multiple cellular carriers on high speed rails," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2018, pp. 161–175.

[6] R. He *et al.*, "High-speed railway communications: From GSM-R to LTE-R," *IEEE Veh. Technol. Mag.*, vol. 11, no. 3, pp. 49–58, Sep. 2016.

[7] *Universal Mobile Telecommunications System (UMTS); LTE; Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (e-utran)*. Standard 3GPP TR 25.913 Version 8.0.0 Release 8, Jan. 2009. [Online]. Available: http://www.3gpp.org/DynaReport/25913.htm

[8] M. Russell and G. L. Stuber, "Interchannel interference analysis of OFDM in a mobile environment," in *Proc. 21th IEEE 45th Veh. Technol. Conf. Countdown Wireless Century*, Jul. 1995, pp. 820–824.

[9] Y. Yang, P. Fan, and Y. Huang, "Doppler frequency offsets estimation and diversity reception scheme of high speed railway with multiple antennas on separated carriages," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2012, pp. 227–233.

[10] V. Jacobson, "Congestion avoidance and control," in *Proc. Symp. Commun. Archit. Protocols SIGCOMM*, 1988, pp. 1–25.

[11] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2514–2524.

[12] L. S. Brakmo and L. L. Peterson, "TCP vegas: End to end congestion avoidance on a global Internet," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[13] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. MobiCom*, 2001, pp. 287–297.

[14] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," in *Proc. IEEE INFOCOM 25th Int. Conf. Comput. Commun.*, Apr. 2006, pp. 1–12.

[15] M. A. Alrshah, M. Othman, B. Ali, and Z. M. Hanapi, "Comparative study of high-speed linux TCP variants over high-BDP networks," *J. Netw. Comput. Appl.*, vol. 43, pp. 66–75, Aug. 2014.

[16] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Commun. ACM*, vol. 60, no. 2, pp. 58–66, Jan. 2017.

[17] H. Jiang, Y. Wang, K. Lee, and I. Rhee, "Tackling bufferbloat in 3G/4G networks," in *Proc. ACM Conf. Internet Meas. Conf. IMC*, 2012, pp. 329–342.

[18] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *Proc. USENIX NSDI*, 2013, pp. 459–471.

[19] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, "Adaptive congestion control for unpredictable cellular networks," in *Proc. ACM Conf. Special Interest Group Data Commun. SIGCOMM*, 2015, pp. 509–522.

[20] F. Lu, H. Du, A. Jain, G. M. Voelker, A. C. Snoeren, and A. Terzis, "CQIC: Revisiting cross-layer congestion control for cellular networks," in *Proc. 16th Int. Workshop Mobile Comput. Syst. Appl. HotMobile*, 2015, pp. 45–50.

[21] X. Xie, X. Zhang, and S. Zhu, "Accelerating mobile Web loading using cellular link information," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2017, pp. 427–439.

[22] W. K. Leong, Z. Wang, and B. Leong, "TCP congestion control beyond bandwidth-delay product for mobile cellular networks," in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 167–179.

[23] S. Park, J. Lee, J. Kim, J. Lee, S. Ha, and K. Lee, "ExLL: An extremely low-latency congestion control for mobile cellular networks," in *Proc. 14th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2018, pp. 307–319.

[24] X. Li, C. Bao, M. Chen, H. Zhang, and J. Wu, "The China education and research network (CERNET) IVI translation design and deployment for the IPv4/IPv6 coexistence and transition," Tech. Rep., 2011.

[25] Y. Li, C. Peng, Z. Yuan, J. Li, H. Deng, and T. Wang, "Mobileinsight: Extracting and analyzing cellular network information on smartphones," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2016, pp. 202–215.

[26] F. Luan, Y. Zhang, L. Xiao, C. Zhou, and S. Zhou, "Fading characteristics of wireless channel on high-speed railway in hilly terrain scenario," *Int. J. Antennas Propag.*, vol. 2013, pp. 1–9, Jan. 2013.

[27] [Online]. Available: http://soar.group/projects/hsrnet

[28] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the Internet," *Queue*, vol. 9, no. 11, p. 40, Nov. 2011.

[29] Y.-C. Chen, Y.-S. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of MultiPath TCP performance over wireless networks," in *Proc. Conf. Internet Meas. Conf. IMC*, 2013, pp. 455–468.

[30] J. Zhou *et al.*, "Demystifying and mitigating TCP stalls at the server side," in *Proc. 11th ACM Conf. Emerg. Netw. Exp. Technol. CoNEXT*, 2015, pp. 1–13.

[31] V. Paxson, M. Allman, J. Chu, and M. Sargent, "Rfc 6298," *Comput. TCP's Retransmission Timer*, to be published.

[32] T. Flach *et al.*, "Reducing Web latency: The virtue of gentle aggression," in *Proc. ACM SIGCOMM*, 2013, pp. 159–170.

[33] Radio Resource Control (RRC). (Apr. 11, 2019). *3GPP TS 36.331 Version 12.18.0 Release 12*. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36331.htm

[34] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, Nov. 2002, pp. 271–280.

[35] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[36] K. Jang *et al.*, "3G and 3.5G wireless network performance measured from moving cars and high-speed trains," in *Proc. 1st ACM Workshop Mobile Internet Through Cellular Netw. MICNET*, 2009, pp. 19–24.

[37] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang, "Experiences in a 3G network: Interplay between the wireless channel and applications," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. MobiCom*, 2008, pp. 211–222.

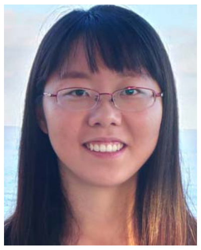
[38] F. P. Tso, J. Teng, W. Jia, and D. Xuan, "Mobility: A double-edged sword for HSPA networks: A large-scale test on hong kong mobile HSPA networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1895–1907, Oct. 2012.

[39] R. Merz, D. Wenger, D. Scanferla, and S. Mauron, "Performance of LTE in a high-velocity environment: A measurement study," in *Proc. 4th Workshop Things Cellular, Oper., Appl., Challenges AllThingsCellular*, 2014, pp. 47–52.

[40] J. Huang *et al.*, "An in-depth study of LTE: Effect of network protocol and application behavior on performance," in *Proc. ACM SIGCOMM*, 2013, pp. 1–12.

[41] Y. Xu, Z. Wang, W. K. Leong, and B. Leong, "An end-to-end measurement study of modern cellular data networks," in *Proc. PAM*. Springer, 2014, pp. 34–45.

Chenren Xu (Senior Member, IEEE) received the Ph.D. degree from WINLAB, Rutgers University. He was a Post-Doctoral Fellow at Carnegie Mellon University and held a visiting scholar position at AT&T Shannon Laboratories and at Microsoft Research. He is currently an Assistant Professor at the Department of Computer Science and Technology and also an Affiliated Member of CECA at Peking University, where he directs Software-hardware Orchestrated ARchitecture (SOAR) Laboratory since 2015. His research interests span wireless, networking, and systems. He was a recipient of the Alibaba DAMO Academy Young Fellow and the CCF-Intel Young Faculty awards.

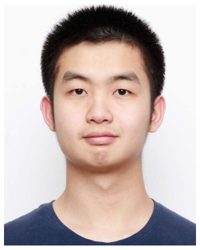


Jing Wang received the bachelor's degree in computer science from Peking University in 2017, where she is currently pursuing the Ph.D. degree in computer science. Her research interests include wireless networking and networking systems.

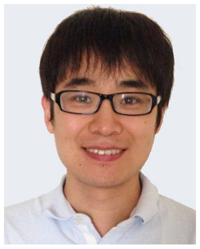
Zhiyao Ma is currently pursuing the degree in computer science with Peking University. His research interests include systems and networks.

Yihua Cheng is currently pursuing the degree at Peking University. His research interests include software-defined networks and high performance systems.

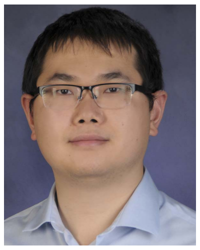
Yunzhe Ni received the bachelor's degree in computer science from Peking University in 2018, where he is currently pursuing the Ph.D. degree in computer science. His research interests include wireless networking and networking systems.



Wangyang Li received the bachelor's degree from Peking University. He is currently pursuing the Ph.D. degree with The University of Texas at Austin. His research interests include computer network and systems, wireless communications, and network security.



Feng Qian received the bachelor's degree from Shanghai Jiao Tong University, and the Ph.D. degree from the University of Michigan. He is currently an Assistant Professor at the Computer Science and Engineering Department, University of Minnesota—Twin Cities. His research interests include the broad areas of mobile systems, VR/AR, computer networking, and system security.



Yuanjie Li (Member, IEEE) received the B.E. degree in electronic engineering from Tsinghua University in 2012, and the Ph.D. degree in computer science from UCLA in 2017. He is currently a Researcher with Hewlett Packard Labs, Palo Alto, CA, USA. From 2017 to 2018, he was the Co-Founder of MobIQ Technologies. His research interests include network systems and security, with a recent focus on mobile networking, intelligent wireless edge, and Internet-of-Things (IoT). He was a recipient of the ACM MobiCom'17 and MobiCom'16 Best Community Paper Awards, and the UCLA Dissertation Year Fellowship in 2016.