An Active-Passive Measurement Study of TCP Performance over LTE on High-speed Rails

Jing Wang[†], Yufan Zheng[†], Yunzhe Ni[†], Chenren Xu[†✉], Feng Qian[♯], Wangyang Li[†], Wantong Jiang[†]
Yihua Cheng[†], Zhuo Cheng[†], Yuanjie Li[§], Xiufeng Xie[§], Yi Sun[♭], Zhongfeng Wang[♮] [*]

[†]Peking University, China [♯]University of Minnesota – Twin Cities, USA [§]Hewlett Packard Labs, USA
[♭]Institute of Computing Technology, University of Chinese Academy of Sciences [♮]China Academy of Railway Sciences

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 a large-scale active-passive measurement study of TCP performance over LTE on HSR. Our measurement targets the HSR routes in China operating at above 300 km/h. We performed extensive data collection through both controlled setting and passive monitoring, obtaining 1732.9 GB data collected over 135719 km of trips. Leveraging such a unique dataset, we measure important performance metrics such as TCP goodput, latency, loss rate, as well as key characteristics of TCP flows, application breakdown, and users' behaviors. We further quantitatively study the impact of frequent cellular handover on HSR networking performance, and conduct in-depth examination of the performance of two widely deployed transport-layer protocols: TCP CUBIC and TCP BBR. Our findings reveal the performance of today's commercial HSR networks "in the wild", as well as identify several performance inefficiencies, which motivate us to design a simple yet effective congestion control algorithm based on BBR to further boost the throughput by up to 36.5%. They together highlight the need to develop dedicated protocol mechanisms that are friendly to extreme mobility.

CCS CONCEPTS

• **Networks** → **Network measurement**.

KEYWORDS

Measurement; High-speed Rails; TCP; LTE; High Mobility; Handover; CUBIC; BBR

ACM Reference Format:
Jing Wang, Yufan Zheng, Yunzhe Ni, Chenren Xu, Feng Qian, Wangyang Li, Wantong Jiang, Yihua Cheng, Zhuo Cheng, Yuanjie Li, Xiufeng Xie, Yi Sun, Zhongfeng Wang. 2019. An Active-Passive Measurement Study of TCP Performance over LTE on High-speed Rails. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19), October 21–25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3300061.3300123

1 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, 18 countries have developed HSR to connect major cities. In China, the HSR network exceeds 22,000 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 challenges in delivering seamless Internet service for on-board passengers from the trackside broadband radio (*e.g.,* LTE) connectivity in a bottom-up fashion – from error-prone L1/L2 connectivity to misguided TCP. Specifically, as will be shown, the increasing mobility level poses several new challenges: it degrades the link quality as the Doppler spread increases, increase the BER and reduces the PHY data rate, and hence throttles TCP throughput; from the *handover* perspective, handover not only become more frequent, but also are more likely to fail

[*] J.Wang, Y.Zheng and Y.Ni are the co-primary student authors.
✉: chenren@pku.edu.cn

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico

ACM ISBN 978-1-4503-6169-9/19/10…$15.00
https://doi.org/10.1145/3300061.3300123

because of the unreliable handover control signal transmission and the tighter timing budget for handover completion due to the train's ultra-high mobility.

Given HSR's short history, its networking is still a relatively new topic. Existing experimental studies on HSR networking [3, 4] focus on measuring TCP performance in a controlled setting, and thus lack in-depth cross-layer insights and an understanding of HSR networking performance "in the wild". To bridge such a gap, in this paper, we conduct a cross-layer and large-scale measurement study of TCP performance on HSR. Our measurement targets consist of three popular HSR routes in China operating at above 300 km/h. More than 150 million passengers travel on these routes annually. The onboard Internet connectivity is provided by multi-carrier LTE, the main-stream mobile access technology for HSR [5]. Through a period of 10 months, we performed extensive data collection through both passive monitoring (at the LTE gateway whose access was provided by China Academy of Railway Sciences) and controlled experiments. To our best knowledge, this is the largest HSR network trace dataset – 1732.9 GB data collected over 135719 km of trips. Our measurement consists of four parts as detailed below.

• (§4.1) Leveraging such a unique dataset, we begin with measuring important performance metrics for two TCP variants: CUBIC [6] and BBR [7], 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, due to frequent handover as well as lower PHY rate. Meanwhile, BBR still holds its native property of low(er) RTT, loss rate and bytes-in-flight (when in comparison to CUBIC) in such extreme mobility environment.

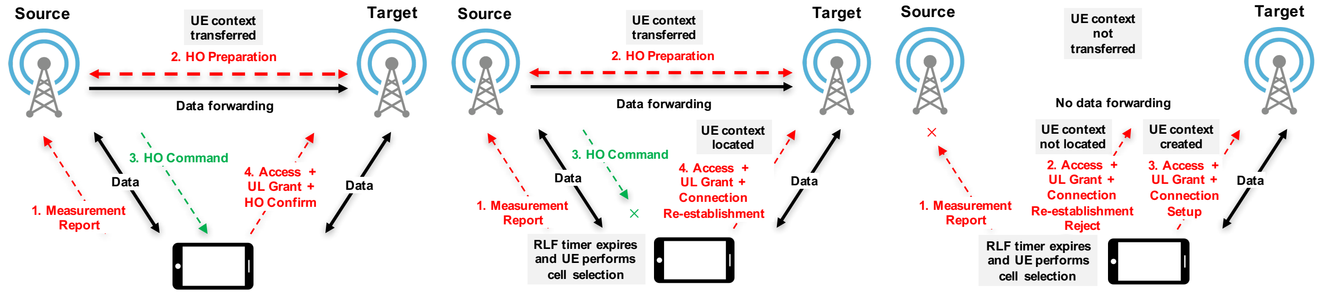
• (§4.2) We then measure key characteristics of TCP flows, application breakdown, and users' behaviors from the on-board passengers' WiFi traffic data. We found that HTTP(S) still dominates the application protocol usage (94.13%). Among them, more than 95% of the flows are composed of text, image and application data (rather than audio and video), and they are slow (less than 1 Mbps), short (less than 100 KB) and unlikely to finish when its size is above 1 MB. Another interesting observation is that in HSR networks, the usage patterns are quite distinct (*e.g.,* weekends do not necessarily generate more data traffic, and the diurnal patterns are less prominent) as attributed to the unique context of passenger traveling by train. The above findings shed light on improving traffic classification, resource allocation, and cellular infrastructure planning for HSR networking.

• (§5) Given the importance of handover in high-mobility cellular access [4], we next conduct a quantitative in-depth study of the handover impact on HSR networking performance. Specifically, we first develop appropriate taxonomy of handover (§2.2), and then correlate the lower-layer LTE messages (*e.g.,* PHY rate, handover) with TCP's behavior. We make four key findings. First, handover occur frequently in HSR – every 13.7/8.6 seconds on average at 300/350 km/h. Second, depending on a handover's type, its performance impact on TCP varies; despite most handovers being successful, as the mobility level increases, more and longer unsuccessful handovers will appear, leading to more negative impact. Third, for successful handover, it is typically too short (*i.e.,* tens of ms) for TCP to react over its normal RTT of hundreds of ms. The fourth finding is what we call the "near effect": an unsuccessful handover (typically more than 1 second) can negatively affect data rate over a much longer window (*e.g.,* more than 8 seconds) after its actual occurrence.

• (§6) We next conduct more in-depth exploration of comparative TCP performance. Our key findings include the following. First, we find that BBR recovers more smoothly and slower than CUBIC after all type of handover because it has a intrinsically less radical strategy in expanding congestion window and thus more handover-agnostic. Specifically, it performs slightly better after Radio Link Failure (RLF) handover, but much worse after Non-Access Stratum (NAS) recovery than CUBIC. Even though, BBR achieves comparable throughput with CUBIC, but with a much shorter RTT and packet loss rate. Second, we find that BBR outperforms CUBIC over the connection with higher random losses and thus more carrier-agnostic in our measurement setting. Third, BBR is still suboptimal in HSR with high variant RTT due to its excessively conservativeness in *RTprop* estimation, a critical parameter controlling its sending window. We managed to achieve 1.36x throughput improvement by simply tuning the strategy of bandwidth probing and adding a stochastic compensation term in *RTprop* estimation of BBR (§7).

Contribution. This work represents the first large scale in-depth HSR networking measurement study covering the following aspects: ① user/traffic behavior in the wild, ② TCP-LTE interactions, ③ comparative TCP variants behavior, and ④ handover-centric. Here are the key findings:

- HTTP(S) flows dominate on-board Internet traffic, while the QoE (*e.g.,* throughput, time to first byte and completion percentage) has huge room for improvement.
- Higher train speed not only causes more (unsuccessful) handovers, but also degrades PHY (and TCP) data rate in a non-linear fashion during periods out of handover.
- While most handover happen successfully (*i.e.,* finish within 100 ms) and cause negligible impact on TCP, the



(a) Successful (Type I) **(b) RLF (Type II)** **(c) NAS Recovery (Type III)**

Figure 1: Different LTE Handover Scenarios.

unsuccessful handover disrupt TCP in a complicated manner which depends on the congestion control algorithm.

- BBR is more handover and carrier agnostic than CUBIC, but recovers much slower after a long disconnection. We demonstrate there is still great potential for further improvement with a simple end-to-end solution.

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) [8], a new standard being discussed for the next-generation private HSR communication system for mission-critical services.

2 BACKGROUND

2.1 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 (of low speed, *i.e.,* below 200 km/h), it runs into severe performance issues when the client mobility is raised to a higher level. According to 3GPP TR 25.913 [9], *"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 large Doppler spread, which is proportional to relative motion 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 [10, 11]. As a result, it not only causes higher decoding errors, but also makes a cell choose lower modulation and coding rate, which together lower the PHY data rate and throttle TCP throughput during the periods even without handover. Another side effect of weak signal 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.

Handover. on HSRs becomes the main root cause for disrupting a TCP flow – the increasing mobility level can make it 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 to finish given the shorter time window within the overlap zone due to high mobility. Third, the "tidal effect" can easily overload the basestation, in both control and data channels. Upon failure, it needs to spend extra time in discovering and reconnecting to the cell, and thus keeps TCP disconnected.

2.2 LTE Handover Primer

Handover [12–14] is a key function for realizing seamless user experience in mobile networking – from a source cell to target cell. In general, a handover can be described as a network-controlled and user equipment (UE)-assisted procedure. According to the 3GPP standard, a handover procedure can be described as follows: UE sends the Measurement Report (*e.g.,* signal strength from all the perceived cells) to the source cell. When the source cell decides to perform a handover, it communicates with the target cell for radio resource preparation, informs the UE of the handover action by sending a Handover Command (RRC Connection Reconfiguration) after forwarding UE's buffered downlink (and optionally uplink) user plane data to the target cell for lossless delivery. Upon receiving this message, UE synchronizes with and gains access to the target cell, and sends a Handover Confirm (RRC Connection Reconfiguration Complete) message to continue the session. In real environment, however, the handover procedure can end up with three different scenarios:

- *Successful handover* (Fig. 1a). It happens when all the controlling signals are received, and the buffered data are losslessly forwarded to target cell and delivered to UE for minimizing the flow disruption.

- *Radio Link Failure (RLF) handover* (Fig. 1b). It happens when radio conditions not good enough for the UE to be able to decode the Handover Command from the source cell. When the UE detects radio link problems, it starts the RLF timer. Upon expiration of the RLF timer, the UE searches for a suitable target cell and re-establishes its connection with it (performing the RRC Connection Reestablishment procedure) if it (the target cell) happens to have been prepared by the source cell. RLF handover incurs additional delay, but no data buffered in cell is lost.

- *Non-Access Stratum (NAS) recovery* (Fig. 1c). It happens when the target cell is not prepared for handover. In such case, the UE attempts to establish a new connection – the UE context needs to be created, and all the buffered data is lost and needs upper-layer (TCP) retransmission.

2.3 TCP Primer

We briefly introduce the necessary background on how CUBIC and BBR deals 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 and flows grow their *cwnd* at the same rate. After CUBIC enters into congestion avoidance 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 will still 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 *RTprop*), to model the end-to-end capacity and determine its congestion control window. Specifically, BBR first uses the slow-start akin to CUBIC's only when the flow is initially launched and then soon reach its bandwidth probing phase after the throughput converges. In this phase, it takes a period-8 cycling *pacing_gain* sequence $(1, 1, 1, 1, 1, 1, 5/4, 3/4, \ldots)$ in turn as a multiplier to *BtlBw* to determine its sending rate for *RTprop* (not RTT) time – while *pacing_gain* $= 1$ at most of the time, a *pacing_gain* > 1 means BBR is in the phase of exploring 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. The take away message is that, BBR is robust to random packet loss, but will take long time (in comparison to CUBIC) to recover after a long disconnection, *e.g.,* after $10 \cdot RTprop$ or longer.

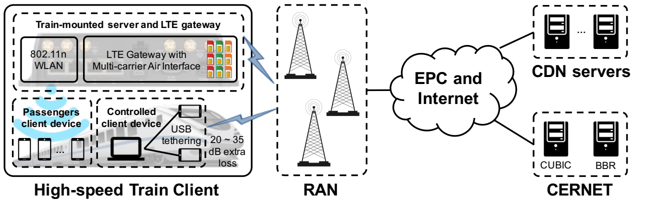


Figure 2: Our experimental testbed collects data from both a dedicated laptop-phone suite and a LTE gateway in active and passive manners respectively.

3 MEASUREMENT METHODOLOGY

3.1 Factors to Examine

In order to demystify the performance issues and optimization opportunities in HSR networking, we use real-world data complemented with on-board controlled experiments to gain insights in the following dimensions: ① What is unique about the interaction between TCP and LTE in high mobility environment? ② How do different TCP congestion control algorithms behave? ③ How do the on-board passengers use Internet? While TCP performance, cross-layer interaction, and user behaviors under stationary or low/moderate mobility have been extensively studied in the literature, there are much fewer studies of them under extreme mobility. We summarize the high-level experimental design in Tab. 1 and experimental setup for data collection in Fig. 2, which will be further explained in the rest of this section.

3.2 Active Measurements in Controlled Setting

We begin with describing controlled experiments conducted on HSR. They allow us to capture and analyze a wide range of information in a cross-layer manner.

3.2.1 Experimental setup

Server. We deploy two powerful co-located servers (Intel NUC6i7KYK with i7-6770HQ, 32 GB DDR4 and Samsung 950 pro 512 GB) in CERNET [15], the nationwide education and research computer network in China.

Client. We tether two Android phones (Xiaomi 5s) to one laptop (Dell XPS 13-9360) via USB 3.0. This tethered setup

Factor to examine	Data collection	Experiments	Distance	Data size
TCP-LTE interaction in high mobility	TCP trace from both ends and LTE trace from client at different speed	Active measurements	51367 km	357.9 GB
TCP variants behavior comparison	TCP trace from servers using different TCP congestion control algorithms	Active measurements	51367 km	323.0 GB
User profile of Internet usage	TCP trace from train-mounted LTE gateway over a few months	Passive measurements	84352 km	1376 GB

Table 1: Dataset description.

allows us to programmatically 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. Note that in China there are three mainstream carriers and we cover two of them; the third carrier uses the same technology (FDD) and exhibits similar performance as Carrier B based on our pilot on-board tests.

3.2.2 Experimental design

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

Flow size. We measure two types of TCP downlink flows, including long flows (*i.e.,* fixed duration of 150 seconds) and fixed size of 64 KB (corresponding to typical web page). Given the smaller protocol overhead, 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 image and JavaScript) to tens of MB such as HD video chunks. These are typical workload of HSR networking.

TCP-LTE interaction in high mobility. We run tshark on both client and server to collect packet-level TCP traces. We also instrument the client phones using MobileInsight [16][1] to collect lower layer information including PHY rate and handover events.

TCP variants comparison. The two co-located servers run Ubuntu 17.04 with kernel 4.10.17 with CUBIC and BBR, arguably the most widely deployed congestion control algorithms, respectively. We compare their performance and their incurred cross-layer interactions.

When comparing CUBIC and BBR, we can either execute them sequentially (back-to-back) or concurrently (side-by-side). For experiments under zero or low mobility, one can typically do back-to-back runs. In HSR, however, the channel condition and link quality may change dramatically over a just few seconds' window, so back-to-back runs may not ensure apple-to-apple comparisons. We therefore run both flows concurrently. However, a concern raised here is that whether this side-by-side setting will cause these two flows to interfere with each together over the LTE network. To study this, we perform the concurrent and sequential experiments in an interleaved manner for 100 times and run each experiment for 1-minute long. We then measure their throughput in Fig. 3. As shown, CUBIC and BBR yield qualitatively similar performance when running with and without another concurrent flow. This is likely attributed to the base stations' proportional fair scheduling [17] as well as their resource capacity capable of serving hundreds of UEs. In short, we believe that on a fully provisioned HSR route, the performance impact due to the inter-device contention is dwarfed by the impact caused by the extreme mobility.

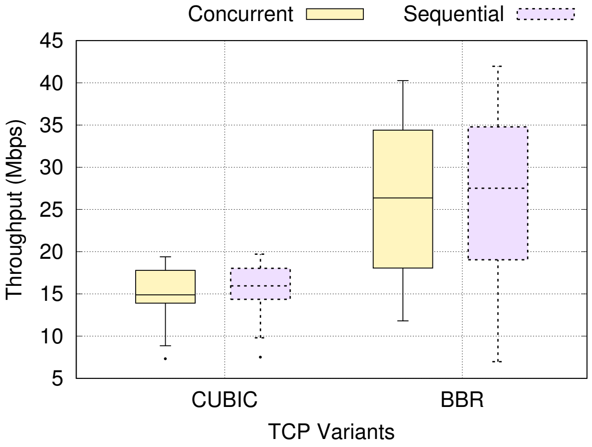


Figure 3: Fairness Validation for Concurrent Tests.

3.2.3 Data Collection and Processing

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. We collected 357.9 GB data by traveling 51367 km on the trains. Since TCP-LTE performance may vary along the route because of the terrain diversity [18] 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 performed numerous types of data processing such as extracting TCP flows and LTE events, calculating various performance metrics, and aligning TCP traces with LTE events. We next describe how we extract two important types of LTE data.

[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.

- *PHY Rate* is number of Transport Block (TB) size (*i.e.,* the number of bytes that can be carried over a subframe) per second. Specifically, the TB size is jointly determined by the number of resource blocks (RB)[2] and the modulation/coding scheme (MCS).

- *Handover* can happen in three different ways (§2.2) on HSR-LTE. In the rest of the paper, we refer to the three handover scenarios as Type I, II, and III as shown in Fig. 1, and denote the start and end time of a handover as HO_{start} and HO_{end}, respectively. HO_{end} can be simply determined by the time when Reconfiguration Connection Reconfiguration Complete message is sent (and logged). However, it is more complicated to determine HO_{start}, especially for unsuccessful handover – in practice we don't have the access to the information of RLF timer, which is triggered by radio link failure and leads to cell selection (and handover). Given the streaming nature of our controlled experiment, we hence set the timestamp of the last LTE downlink packet was perceived before the nearest HO_{end} as HO_{start}.

We release the dataset used for this study in [19].

3.3 Passive Measurements in the Wild

We complement controlled experiments with passive measurements that collect TCP flows from on-board passengers, in order to study the passengers' network usage and their flow characteristics "in the wild". We are unaware of any prior passive measurement of HSR networking.

Since 9/2017, China Railway Corporation launched the new "Fuxing Hao" trains for the Beijing-Shanghai HSR route. They bring two notable features: cruising at the speed up to 350 km/h [20] that is faster than any other HSR route in China, and providing free WiFi service via an on-board LTE gateway [21]. Hence, the LTE gateway becomes an ideal point for our passive data collection in the wild.

The LTE gateway is deployed in a on-train server room by China Academy of Railway Sciences (CARS). It runs OpenWRT 3.9, and is equipped with 9 SIM cards of three major Chinese mobile carriers for data relay between LTE RAN and on-board WiFi users. It also deploys a 2 × 2 MIMO antenna mounted on the top of the carriage. Each new TCP flow is assigned to a SIM card in a round-robin manner.

We obtained permission from CARS to run instrumentation software on the LTE gateway for passive data collection. We deployed tshark to collect packet-level TCP traces (headers only) from the LAN port of the gateway. Note that we cannot run MobileInsight or other PC-based cellular performance monitoring tool such as QXDM [22] because of the OS incompatibility. But we were able to distinguish different passengers' TCP flows from their assigned WLAN IP addresses appearing in the PCAP traces. Overall, we collected 1376 GB data covering 84352 km.

Ethical Considerations. We take measures as much as we can to protect users' privacy. First, all passengers who participated in our study were presented with informed consent statement before connecting to the on-board WiFi service managed by the HSR operator. Second, when collecting the traffic trace, only TCP/IP headers were examined, and the clients' IPs are private addresses from which no personal identifiable information could be inferred. Third, this study has been approved by the Institutional Review Board at the primary authors' institution.

4 BASIC PERFORMANCE STATISTICS

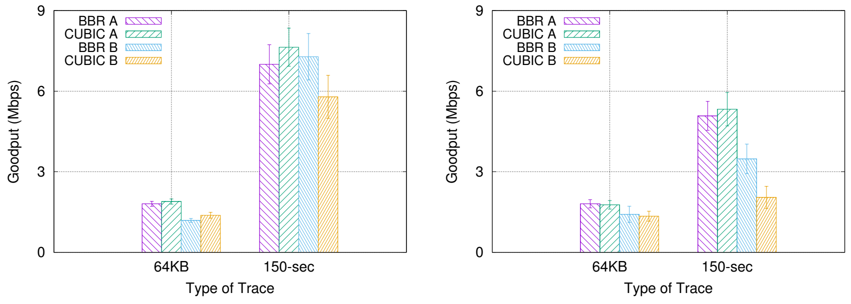
4.1 Performance of TCP Variants

We first utilize the controlled measurement data to study key network-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.

Goodput. Fig. 4 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: a short flow (64 KB) and a long-lived bulk download flow lasting for 150 seconds. As shown, neither the CCA nor the carrier appears to significantly affect the performance of the short flow, which mostly finishes within the slow start stage during which the available bandwidth is under-utilized. For the long flow (150 sec), we make two key observations. First, as the mobility level increases from 300 km/h to 350 km/h, the goodput of CUBIC and BBR both decrease by 47.5% and 40.1%, respectively. This is attributed to the lower PHY rate (caused by the imperfect radio receiver design in high mobility) to be discussed in §5. 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 (to be discussed in §6.2) which is infrastructure-dependent. Such random losses force CUBIC to (more) frequently back off while bringing 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. 2 by the end of this section.

[2] RB is time-frequency resource that occupies 12 subcarriers (12 × 15 kHz) and one slot (0.5 ms). RBs are allocated by the eNB scheduler and this allocation information is sent to the UEs for informing the radio resource and PHY layer rate assignment.



(a) 300 km/h

(b) 350 km/h

Figure 4: Goodput.

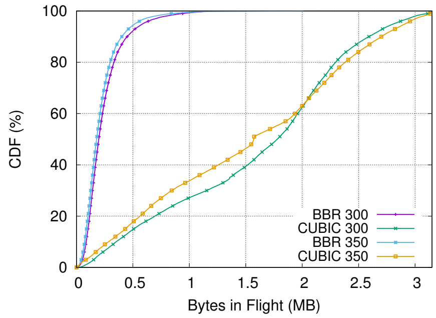


Figure 5: BiF (Carrier A).

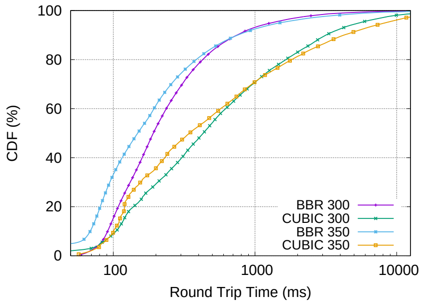


Figure 6: RTT (Carrier A).

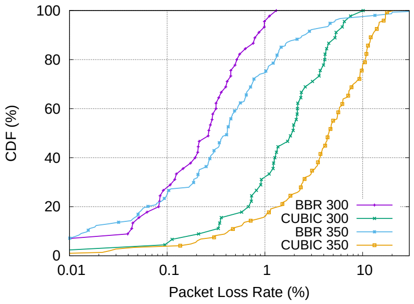


Figure 7: PLR (Carrier A).

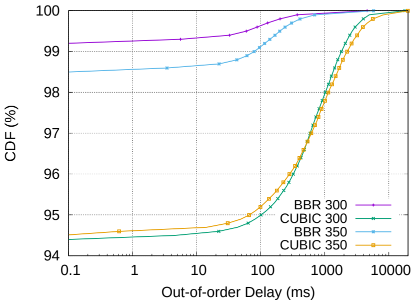


Figure 8: OOD (Carrier A).

Bytes-in-flight (BiF). As shown in Fig. 5, BBR yields an almost a order lower BiF than CUBIC (*e.g.,* 0.20 versus 1.78 MB at 300 km/h, and 0.18 MB versus 1.57 MB at 350 km/h for median value). This cross-validates the RTT difference between BBR and CUBIC shown in Fig. 6, as a large BiF incurs high queuing delay that inflates the RTT [23]. As the mobility level increases, the BiF oftentimes decreases due to reduced throughput. In fact, we found that the higher mobility only causes 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.

Round-trip-time (RTT). As shown in Fig. 6, 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 [24]. The increase of mobility level affects the RTT in two aspects. On one hand, more frequent handover and higher packet loss rate (Fig. 7) 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.

Packet Loss Rate (PLR). As shown in Fig. 7, BBR has about an order 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 the more (unsuccessful) handovers and decoding errors.

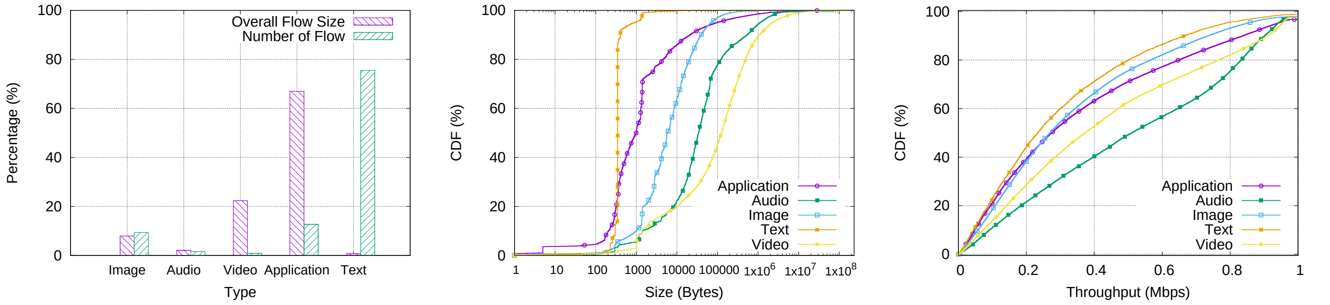
Metrics	Mean	Median	95% percentile	99% percentile
Goodput (Mbps)	5.12/5.40	4.97/4.78	11.85/13.06	14.60/16.29
BiF (MB)	0.22/1.53	0.18/1.57	0.52/2.91	0.84/3.14
RTT (ms)	457.42/1707.2	148.63/345.02	1491.7/8272.1	6022.9/18760.7
PLR (%)	1.38/5.92	0.41/ 4.53	4.64/14.79	20.18/17.77
OOD (ms)	4.44/77.75	0/0	0/66.62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

Table 2: Critical statistics of BBR/CUBIC performance metrics over different carriers at the speed of 350 km/h.



(a) Content-type

(b) Object Size

(c) Object Data Rate for Objects ≥ 16 KB

Figure 9: Web Content Profile.

Out-of-Order Delay (OOD).[3] As shown in Fig. 8, 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 sec respectively, which can significantly affect the QoE. From the mobility level perspective, it only incurs marginal impact in our measurements.

Summary of Key Findings. Our study show 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 bytes-in-flight to potentially mitigate the bufferbloat issue, and thus more efficient in network utilization.

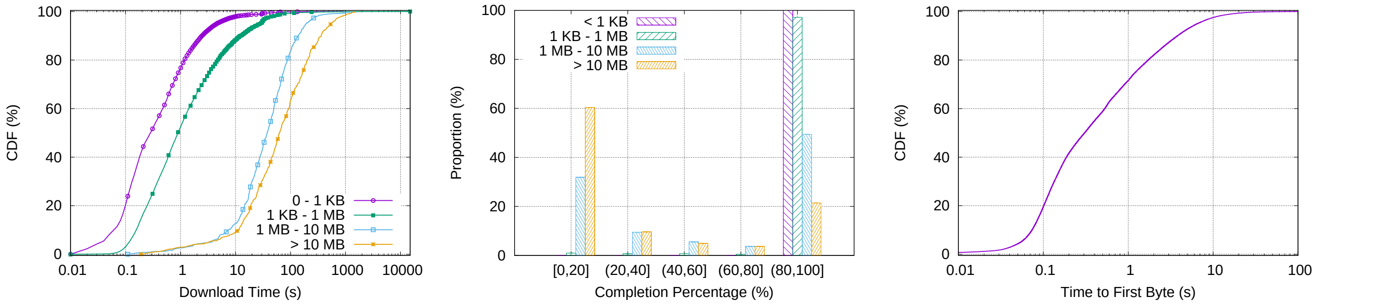
4.2 User, Flow and Traffic Characteristics

Understanding the Internet usage pattern of the on-board passengers is vital to network optimization in terms of bandwidth provision and traffic engineering. Given that the train-mounted LTE gateway is providing free WiFi service to all the passengers, it becomes an ideal spot to collect data in the wild. Inspired by the previous measurement studies of wired [26], WiFi [27] and LTE [28] networks, in this HSR context, we are particularly interested in characterizing the application content, data flow, and the aggregated traffic pattern. For high-level statistics, TCP is still the main transport protocol carrying the Internet traffic from the on-board passengers – 96.95% of the data traffic uses TCP, 2.81% of them uses UDP. Among the TCP flows, 52.94% and 44.15% of them are used by HTTP and HTTPS respectively, and the rest are used by other protocols such as SMTP and FTP.

Web Content Profile. Given that HTTP(S) data dominates the Internet traffic (94.13%), we first dive into the high-level statistics based on the content type, regardless of the protocol version. We classify the object type into image, audio, video, application (*e.g.,* octet-stream, json, Javascript) and text based on the HTTP header. Note that we did not include the HTTPS traffic due to its encryption nature. From Fig. 9a, we observe that application data (66.99%) and video (23.35%) consume most of the traffic in terms of the data size. Specifically, based on the remote IP address analysis, we discover that the majority of the application data are composed of video, gzip file, WeChat text[4], *etc.* . In terms of the number of objects, text (75.51%) dominates the web usage. Regarding the object size (Fig. 9b), the majority (90%) of the text objects are smaller than 500 bytes; image objects have a larger size range, *i.e.,* from 1 KB (10-th percentile) to 40 KB (90-th percentile). Audio and video objects have median sizes of 34 KB and 125 KB respectively. Despite being the largest, they (in particular, the video objects) are still much smaller than those consumed over typical LTE networks when the smartphone is stationary, based on our controlled experiments. This is likely attributed to the video servers' rate adaptation algorithms or the high abandonment rate due to poor network conditions on HSR to be described shortly. Fig. 9c calculates the per-object data rate, defined as the size divided by the object transfer time (including the HTTP request delay). We only calculate data rate for objects larger than 16 KB to ensure meaningful results. Overall, the data rates are low – we barely observe those higher than 1 Mbps. We observe that text has the lowest per-object throughput (median of 233 Kbps), followed by image and application. In addition, we see very low data rate for video objects with a median of only 372 Kbps, which likely discourages user for

[3] The OOD of a packet is measured as the time difference between when a packet arrives at the receive buffer and when its previous packets have arrived [25]. It normally does not affect throughput but goodput because most applications require in-order data delivery.

[4] WeChat is a popular Chinese multi-purpose messaging, social media and mobile payment app.

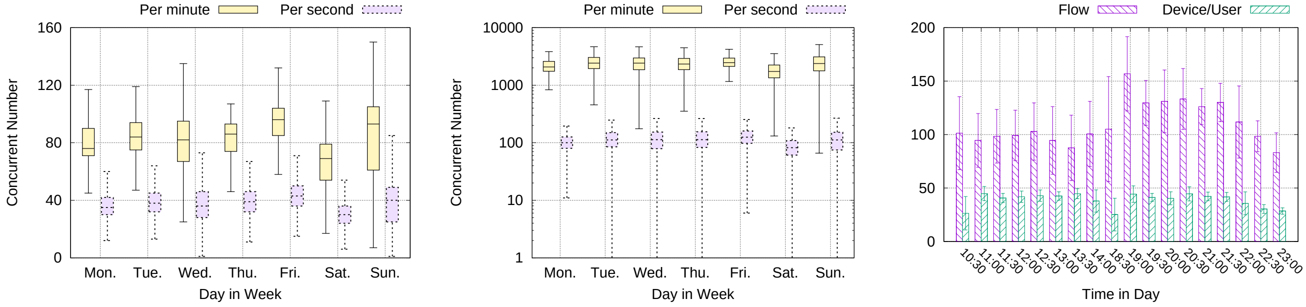


(a) Size vs. Download Time

(b) Completion Percentage

(c) Time to first byte

Figure 10: HTTP QoE Characterization.



(a) Device/User Concurrency

(b) Flow Concurrency

(c) Diurnal Pattern

Figure 11: User Traffic Pattern.

watching videos. The results again indicates the challenges of streaming multimedia contents on HSR.

HTTP QoE Characterization. From our passive data, we are able to infer the QoE of HTTP to some extent, including download time, completion percentage (*i.e.,* proportion of received bytes to the object size indicated in the HTTP header), and time-to-first byte (TTFB [29]). Here our analysis focuses on text, image, and application objects given that they dominate the web browsing content (97.53% in terms of the size). We first show the CDF of the object size versus download time of the objects with 100% accomplish rate (*i.e.,* those fully downloaded) in Fig. 10a. While the data rate is fairly low, 80% of the objects smaller than 1 KB can be finished within 1 second, providing reasonable QoE such as receiving a simple notification message. On the other hand, although most of the large text/images do not exceed 1 MB, 20% and 10% of the download time can be longer than 5 and 10 seconds respectively, significantly hurting the client's user experience. Such high latency is more likely to occur during Type II and III handover. Second, when the network condition is poor, we often experience an incomplete object transfer due to user abandonment or browser timeout, in particular for large objects. In Fig. 10b, we observe that when the object size is small (sub-KB or several KB), the probability that they were fully downloaded are 99.99% and 96.59% respectively. However, when the object size is larger than 10 MB, completion percentage of 100% drops sharply to a surprisingly low value of 16.96%. Finally, we plot the results of TTFB in Fig. 10c. We observe while the median value is 300 ms which is generally considered as acceptable, the top 25% and 10% percentiles can reach higher than 1 and 10 seconds respectively. Overall, the above measurements imply that the QoE on HSR-LTE is far from being satisfactory.

User Traffic Pattern. It is important to know how many active WiFi users are on the train and how many flows they generate simultaneously, for the purpose of future network infrastructure provisioning and traffic scheduling. We compute the statistics across different days and hours. From Fig. 11a, we observe that Saturday turns out to be day with the least network usage. Our experiences as frequent HSR travelers are that more passengers on that day are traveling with family, and therefore spend more time in talking with others rather than surfing the Internet. On weekdays or Sunday there tend to be more business travelers. For the number of concurrent users per second, the median value across different days range from 30 to 43, and the maximum value can reach to up to 98 (out of a typical number of 556 passengers on a fully boarded HSR train). This value is about half of the number when we count per minute. In terms of number of flows (Fig. 11b), the median value is 106 and 2248 per second and per minute respectively. Finally, we report the passengers' diurnal pattern (*i.e.,* number of flows and device per second) in Fig. 11c at the granularity of half an hour. Compared to typical diurnal patterns for residential Internet usage [30], the diurnal patterns here are less prominent as attributed to the unique context of passenger traveling by train. We observe though users are slightly more active in the evening, especially between 19:00 and 21:30 when travelers tend to relax by using Internet for entertainment. Overall, the current on-board WiFi service is only serving less than ten percent of the passengers.

4.3 Remarks on Active-Passive Measurements

While we take both active and passive measurements from the same series of high-speed trains, it is difficult to perform the direct comparison between them. For instance, we observe much lower throughput in passive measurements than active ones (300 Kbps vs. 5 Mbps), which can be attributed to the large (HTTP/TCP) protocol overhead for short flows and potential performance bottleneck on either WiFi links, or the inherently deficient design of the multi-tenant antenna system (shared by 9 SIM cards) and the flow-level round-robin flow scheduling mechanism. The high-level statistical results from the active measurements will shed light on the choice of TCP variants and even multi-path transmission mechanism development atop for application-specific server deployment for high mobility data networking in general.

5 TCP-LTE ANALYSIS IN HIGH MOBILITY

From this section, we start to examine the interaction of TCP-LTE performance from a handover-centric view under different mobility level in a finer-grained manner. Specifically, we use CUBIC as a case study in this section, and defer its comparative performance study with BBR in §6.

5.1 A Mobility-level View

In our study, mobility level is categorized as stationary, low mobility (200 km/h) and high mobility (300 and 350 km/h). Regarding the dataset, it is worthwhile to note that: ① The speed of choice is at which train maintains in most of the data collection time during the whole journey; ② The low mobility (*i.e.,* 200 km/h) data was not common and it was collected over the same route (*i.e.,* Beijing-Shanghai) on the date when train happens to travel at that speed after snowing for safety reasons; ③ We collect the data in stationary case because its signal has experienced the same path loss (including the carriage penetration loss) and thus it serves as a more appropriate baseline. From Tab. 3, we make two key observations *as the mobility level increases:* First, during the period without handover, TCP throughput (Mbps) drops from 13.25 to 8.44 (36.3%), 6.95 (47.6%) and 2.22 (83.2%) as the mobility increases from static to low (200 km/h), high (300 km/h) and even higher (350 km/h) respectively *in a nonlinear fashion*, which is also reflected in the PHY rate. In fact, they are logically correlated – the increasing mobility brings severer Doppler spread and channel estimation error, and thus cause higher BER and makes the basestation more conservative in assigning MCS, reducing the number of TB and thus TCP throughput. We note that UDP will be a better choice because it excludes the impact of congestion control behavior. However, in our experiments we found the carrier will limit the UDP traffic rate (to 1 Mbps) from time to time. Second, there are more and longer periods of (unsuccessful) handover in the 150-sec traces. Their impact on throughput will be further analyzed later in this section.

Speed	Throughput		PHY rate	HO Duration (s)			HO count per 150s-trace		
(km/h)	All	w/o HO	w/o HO	I	II	III	I	II	III
0	13.25	13.25	14.68	–	–	–	–	–	–
200	8.44	8.63	10.19	0.35	3.13	4.03	4.41	0.18	0.18
300	6.95	7.21	8.89	0.16	2.09	3.98	9.36	0.96	0.84
350	2.22	2.35	3.00	0.21	1.91	5.49	14.70	2.00	1.49

Table 3: TCP-LTE performance and handover statistics under different mobility level regarding to no (w/o), successful (I), RLF (II) and NAS (III) handover.

In the rest of the paper, we focus on analyzing the 350 km/h traces as they represent the most challenging HSR networking scenario for today.

5.2 A Handover-centric View

Handover can cause different level of TCP disruptions, depending on how long it takes and whether it is successful. As shown in Fig. 12, 85% of the type I handover finishes within 100 ms, which has a high chance to be hidden from the TCP as their RTT are often more than that period (Fig. 6). On the other side, more than half of the type II/III handover last more than 1 second, and the top 25% of type II and III handover are even longer than 2 and 5 seconds respectively. To quantify their negative impact, we first study how does the data rate change after the handover. We denote *Normalized Rate* as the PHY rate during the time interval (*i.e.,* 200 ms) divided by the average PHY rate among all traces to even out the different networking performance across the traces. Specifically, handover is shown as a single point in the origin and its interval is regarded as HO_{start} to HO_{end} when in calculation. Note that the reason we choose PHY rate instead of TCP throughput is that there exists tens of ms delay between the on-chip time of LTE protocol message (reported by MobileInsight) and the system time of TCP pcap trace. Hence, the handover and PHY information both reported from MobileInsight should be better aligned in timing.

Instantaneous Impact. One way to quantify the impact of handover is to simply observe the instantaneous data rate

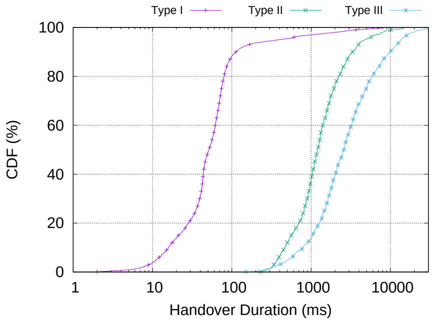
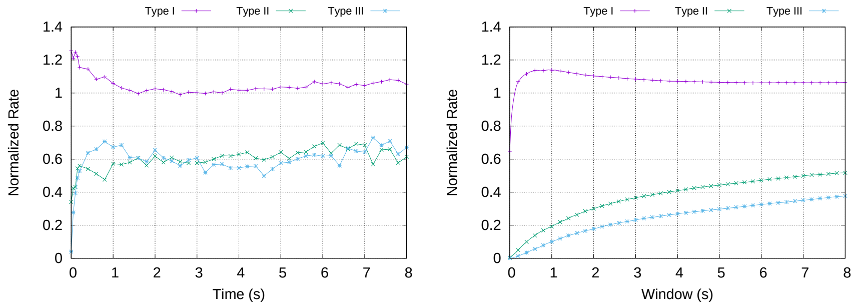


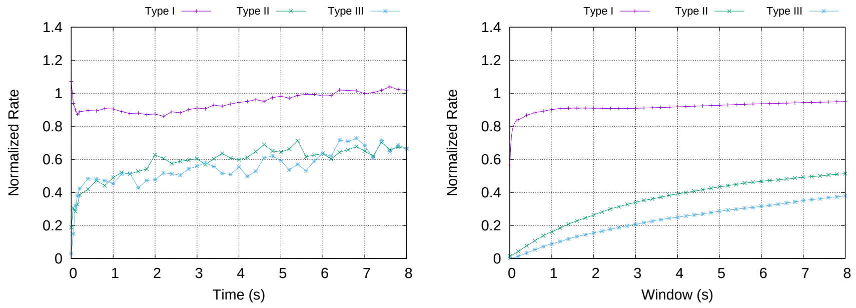
Figure 12: Handover Duration.



(a) Instantaneous impact

(b) Near effect

Figure 13: Handover-centric Analysis of CUBIC.



(a) Instantaneous impact

(b) Near effect

Figure 14: Handover-centric Analysis of BBR.

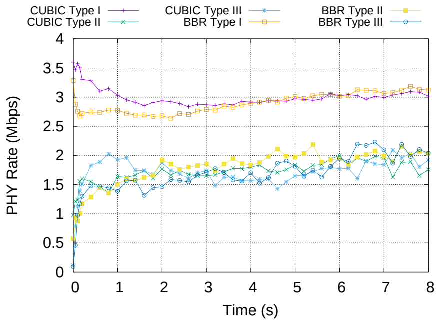


Figure 15: Comparative Performance.

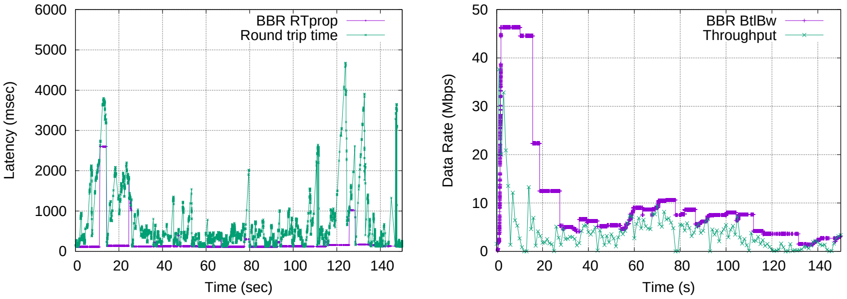
after it happens, or HO_{end}. Taking CUBIC as an example, in Fig. 13a, we make three key observations: First, for type I handover, the normalized rate is almost unaffected by handover and reaches the average rate immediately after the handover, indicating only a few packets are delayed but connection is marginally affected. This is because most type I handover completes within 100 ms, which will unlikely trigger RTOs. In fact, we even observe tens of ms data burst right after the handover ends. This is because the lossless nature of type I handover will ensure the data will not get lost during the handover procedure and can be delivered to the UE from the target cell instead of the server. Note that the normalized rate can reach above 1 since it is normalized by the average rate over the trace, which can certainly be smaller than some instantaneous rate out of the handover period. Second, type II/III handover has a more significant impact on the throughput – it often turns the connection down for a longer time (*i.e.,* longer than 2 seconds in more than quarter of the time) and is more likely to trigger RTO and slow start. Third, although both type II and type III handover are triggered by radio link failure, the data rate at time 0 of the former one is typically higher. This is because type II handover (by definition) is able to transfer the UE context as well as the buffered data to the target cell, while type III handover fails to do and needs upper-layer retransmission.

Near Effect. In practice, the (negative) impact of handover is beyond the instantaneous rate, *i.e.,* user experience is affected not only after the handover ends, but also during its period. We define such phenomenon as *Near Effect*, and denote *window* of length x where x means $[HO_{\text{start}}, HO_{\text{end}} + x]$. We quantify the near effect by computing the ratio of the average PHY rate of the *handover incorporating that window* x to the average PHY rate among all traces. From Fig. 13b, the key observation we make is that while type I handover shows a similar pattern as it shows in the instantaneous impact because of its short duration, type II/III handover has a much lower normalized rate in terms of near effect, primarily because they themselves have a longer (handover) duration, together with the higher probability of multiplicative decrease and slow start due to packet loss and RTO respectively. Specifically, it does not reach half of the average rate after 10 seconds for type III handover.

6 COMPARING BBR WITH CUBIC

6.1 BBR is More Handover-Agnostic

We follow the same analysis procedure (in §5.2) and study BBR in a handover-centric manner. From the instantaneous impact perspective (Fig. 14a), the key observation we make is that, for all types of handover, BBR maintains a smoother and slower data rate change than CUBIC. This is because BBR has a intrinsically less radical strategy than CUBIC in



(a) Round-trip propagation time **(b) Bottleneck bandwidth**

Figure 16: Handover-centric Analysis of BBR.

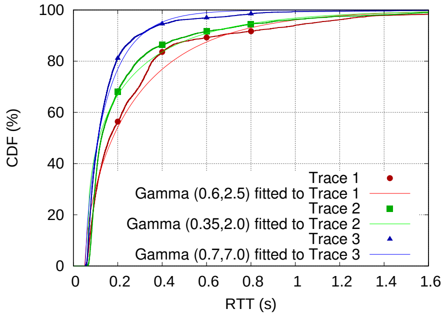


Figure 17: RTT Distribution.

expanding its *cwnd* – approximately at most 25% increase for *BtlBw* at per 8·*RTprop*, which means the same amount of increase in *cwnd* (and the sent data) is not as much as CUBIC's over the same time window. Note that the lossless nature of type I/II handover is observed in the BBR case as well. In terms of near effect (Fig. 14b), we observe similar pattern as shown in the instantaneous impact. Finally, by putting CUBIC and BBR in parallel and showing their PHY rate (Fig. 15), we observe that CUBIC slightly outperforms BBR for all type of handover in the early stage. However, BBR starts to surpass CUBIC after 3 seconds in all cases because it paces smoothly regardless of (random) packet loss.

Remarks. We would like to point out that type III handover for BBR could potentially represent the worst situation in our study: after the long network disconnection (*i.e.,* exceeds 10 ·*RTprop* with certain probability), BBR has to recover from the *BtlBw* of nearly 0, which is a much smaller basis than type I/II in expanding its *cwnd*. We observe such phenomenon (*e.g.,* the normalized instantaneous rate after type III handover cannot even recover to 0.25 after 8 seconds) in a few collected traces, which is however not reflected in statistics (Fig. 14a) due to their relatively small proportion.

6.2 BBR is More Carrier-Agnostic

Recall that in 150-*sec* trace of 350 km/h (Fig. 4b), BBR has comparable goodput (Mbps) with CUBIC over Carrier A (5.12 vs. 5.40), but outperforms around 70% than CUBIC (3.54 vs. 2.06) over Carrier B. Intuitively, CUBIC could suffer over a lossy connection, while BBR is insensitive to packet loss. To verify the fact that BBR outperforms CUBIC in Carrier B is because the cellular infrastructure causes higher random loss, we carry a concurrent test in both static and 350 km/h using the same setup as the controlled experiment described in §3.2, except that each TCP packet only carries 1 byte of data and sends at a stable rate of 20 packets per second to avoid self-inflicted congestion. The results summarized in Tab. 4 show that while Carrier B does cause higher packet loss in non-congestion conditions, potentially due to its poorer mobility support, less network coverage, *etc.* , which explains the relatively poor CUBIC performance over Carrier B.

Flow setup	Static A	HSR A	Static B	HSR B
Loss rate	0.008%	0.21%	0.008%	1.35%

Table 4: Random loss rate for concurrent flows.

6.3 BBR is Suboptimal on High-speed Rails

Our measurements reveal that BBR is able to maintain its desired property of low RTT on HSR, but shows only comparable throughput with CUBIC, not as good as it performs in large-scale WAN – throughput gain over CUBIC by 2-20x [31]. Given its model-based nature, it determines its congestion control window size based on its estimation of *RTprop* and *BtlBw*. Hence, to understand how well the model works, we randomly pick a flow over Carrier A and plot the time series of these two parameters along with the measured RTT and throughput in Fig. 16. From Fig. 16a, we can see that BBR's estimated *RTprop* significantly deviates (0.35x on average) from the RTTs, which can potentially in turn self-throttles its throughput in such networking environment with high RTT variation. On the other hand, even though the estimated *BtlBw* is generally larger than instantaneous throughput (1.6x on average) (Fig. 16b), the resulting estimated congestion control window (*i.e.,* the product of *RTprop* and *BtlBw*) can still be potentially increased, maybe at a acceptable cost of RTT. Specifically, we argue that the estimation of *RTprop*, *i.e.,* the minimum RTT over the last 10 seconds, leads to the underutilization of the connection capacity. We extend this discussion and present a renovated BBR design with experimental evaluation in §7.

7 IMPROVING DATA TRANSFER ON HSR

Although our study reveals that TCP performance on HSR is greatly constrained by the imperfect lower-layer coordination, it is always worthwhile revisiting the TCP (rather than cross-layer) design when considering deployment cost. In comparison to CUBIC, BBR achieves similar throughput but much lower RTT (§4.1), and thus sets a better basis for low-latency networking to provide better QoE for most today's applications. Our goal here is to *renovate BBR to further improve the throughput at an reasonable cost of latency.*

7.1 BBR+ Design

The design of BBR is intrinsically suboptimal in networking environment where both bandwidth and RTT change rapidly, *e.g.,* HSR in our study, because both its bandwidth probing strategy and round-trip propagation time estimation do not adapt to the network dynamics in a agile way. Herein, we improve the BBR design in the following two aspects.

Cycling *pacing_gain*. Recall that in §2.3, default cycling *pacing_gain* sequence makes BBR pace the sending rate according to the drain rate (*i.e., BtlBw*) to keep the bottleneck buffer nearly empty in most of the time. This mechanism is designed for a connection with relatively stable *BtlBw*, not the case in our scenario. Therefore, we adjust the sequence to be more radical to adapt to the HSR environment:

$$3/2, 1/2, 3/2, 1/2, \ldots$$

***RTprop*.** As discussed in §6.3, the *RTprop* BBR estimates is too conservative in HSR scenario when it *has to* be regarded as a constant over the last 10 seconds since the theoretical foundation of BBR is the local stability of *RTprop* and *BtlBw*. Our intuition is that *RTprop* needs a compensation term accounting for the network dynamics and we observe that the RTTs in each trace approximately follow a *shifted gamma distribution* (shifted by min RTT) with a fat tail. In Fig. 17, we show the randomly chosen three BBR traces (with disparate mean and variance of RTTs) along with their corresponding fitted shifted gamma distributions with different parameters, denoted by Gamma(α, β). For a random variable following a shifted gamma distribution, we have:

$$\mathrm{E}(\boldsymbol{X}) = \min X + \sqrt{\alpha \mathrm{Var}(\boldsymbol{X})},$$

where α is the shape parameter of gamma distribution. Therefore, $\sqrt{\alpha \mathrm{Var}(RTT)}$ becomes the natural compensation term to let us get closer to the expectation of actual *RTprop*. Hence, we have:

$$RTprop = \min RTT + \lambda \sqrt{\mathrm{Var}(RTT)}$$

Here, $\mathrm{Var}(\boldsymbol{X}) = E(\boldsymbol{X}^2) - E(\boldsymbol{X})^2$, and $\lambda = \sqrt{\alpha}$ is a tunable parameter which allows us to trade off between bandwidth and RTT. Specifically, we estimate $E(\boldsymbol{X})$ by using EWMA (*i.e.,* Exponentially Weighted Moving Average) with *time-based weight decay* instead of classical EWMA so that the assigned exponentially decayed weights are irrelevant to the sending rate but determined by the time elapsed:

$$E(\boldsymbol{X}) \approx \frac{1}{c} \int_{-\infty}^{t_0} \mathrm{e}^{-c(t_0-t)} X(t)\mathrm{d}t$$

Congestion Control	BBR	BBR+$_{(\lambda=1/8)}$	BBR+$_{(\lambda=1/2)}$
Goodput (Mean ± Std in Mbps)	4.44 ± 3.09	5.49 ± 3.26	6.06 ± 3.25
RTT (25/50/75th percentile in ms)	76/149/311	80/242/694	98/333/1090

Table 5: Performance Comparison of BBR and BBR+.

Note that the two terms in the new *RTprop* estimation are complementary to some extent: the former term monitors the long-term bottleneck buffer occupancy, while the compensation term captures the short-term network dynamics.

7.2 Evaluation

We evaluate BBR+ in the same experimental setting in §3.2, except that the servers run BBR, BBR+$_{(\lambda=1/8)}$ and BBR+$_{(\lambda=1/2)}$ respectively, where λ is the constant used for calculation of *RTprop*. The experimental results summarized in Tab. 5 show that λ is effectively controlling the amount of packets to be filled into the (bottleneck) buffers for throughput gain at the cost of RTT – as λ increase from 1/8 to 1/2, its throughput gain over BBR also increases from 24% to 36%, with increased median RTT of 93 ms and 184 ms respectively, which is still much less than CUBIC.

Remarks. The efforts in this pilot study mean to be inspirational rather than comprehensive. We demonstrate that BBR+ can potentially achieve throughput gain in the applications with a tolerable latency bound. Although the choice of *pacing_gain* and λ in our experiments shows its efficacy, we note that they do not prove to be optimal either theoretically or experimentally. However, we believe there is great potential in the design space of HSR networking.

8 DISCUSSION

TCP Variants. We note there are many alternative (single path) end-to-end TCP variants in the wild, which can be categorized into loss-based [32, 33] and delay-based [34–36] congestion control algorithm in general. We choose CUBIC and BBR in our study because they both not only have large-scale real world deployments, but also represent the state-of-art solution in each category – CUBIC provides the best goodput over high-BDP networks [37], 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 [31]. Meanwhile, we are aware that there are recent designs dedicated for cellular access [23, 38–43]. We leave a comprehensive study for future work.

Railway Route. HSR networking performance may be dramatically different on different routes. Our analysis so far is based on the data collected from Beijing-Shanghai HSR route, which has the best (LTE) coverage among all the routes in China. For other routes with poorer LTE coverage in terms of weaker signal strength and higher packet loss rate, we expect BBR will outperform CUBIC in such cases.

Beyond 350 *km/h*. Recent studies [44, 45] have started to look into the networking performance on airplanes (*i.e.,* 800+ km/h). We believe these two extreme mobility use cases together will call for attention on making best use of cellular and satellite links for improving efficiency and robustness.

9 RELATED WORK

TCP Measurement Study on HSRs (300+ km/h). Most prior measurement work on HSR only focused on the TCP level. The study in [46] 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 [47], 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 for investigating the root cause of TCP (abnormal) 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.

Cross-layer Measurement Study on Mobile Networks. This type of work typically requires access to the low level (L1/2) information. As studied in [48], 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 [49] 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 [50] 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 (number of streams). The authors in [28] 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 [51] revealed burstiness 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 handover-centric study and quantifying its impact at different mobility level up to 350 km/h on high-speed rails.

Large-scale Mobile Network Usage and Performance Characterization. The study [52] characterized application usage and network traffic based on user demographics from 255 users. The work [53] revealed that the application performance difference can be attributed to device type, operating system and web software (compression modes, concurrency). The authors in [54] collected one-week's data from a tier-1 network's UMTS core network to characterize their genre, geographic, popularity, co-occurrence, diurnal and mobility patterns. It was discovered [27] that the mobile application performance can be enhanced by CDN-optimized initial congestion window, but may also be degraded due to the suboptimal design of application-level protocol. The first large real-world LTE packet trace was collected in [28] via a monitor point between eNB and EPC to analyze the flow profile (*e.g.,* size, duration, rate and concurrency) as well as the segmented network latency. The analysis [55] demonstrated there is significant variance in key performance both within and across carrier at different location and time-of-day.

10 CONCLUSION

We perform an in-depth measurement study of HSR networking performance by examining a wide range of factors including TCP performance metrics, flow characteristics, application breakdown, and network usage patterns. In particular, we quantitatively investigate the impact of handovers on HSR networking performance, and compare two representative TCP variants: CUBIC and BBR. Our identified performance issues are often times attributed to the upper-layer protocol design (*e.g.,* BBR's underestimation of $RTprop$), and how they interact with lower-layer characteristics (*e.g.,* TCP's unawareness of high-frequency handovers and their "near effect"). Our insights gained from the study guides us to design a simple yet effective BBR-based congestion control solution to improve the data transfer over HSR. In our future work, we plan to utilize our measurement findings to design transport protocol mechanisms that are more friendly to extreme mobility. We will also develop mechanisms that leverage the path diversity of multiple carriers to boost the robustness of HSR connectivity.

ACKNOWLEDGMENTS

We are grateful to the reviewers and our shepherd, Dr. Matt Welsh in particular, for their constructive critique and comments, all of which have helped us greatly improve this paper. This work is supported in part by National Key Research and Development Plan, China (Grant No. 2016YFB1001200), National Natural Science Foundation of China (Grant No. 61802007 and 61672499) and Science and Technology Innovation Project of Foshan City, China (Grant No. 2015IT100095).

REFERENCES

[1] High-speed rail. https://en.wikipedia.org/wiki/High-speed_rail.

[2] High-speed rail in the united states. https://en.wikipedia.org/wiki/High-speed_rail_in_the_United_States.

[3] Qingyang Xiao, Ke Xu, Dan Wang, Li Li, and Yifeng Zhong. Tcp performance over mobile networks in high-speed mobility scenarios. In *IEEE ICNP*, 2014.

[4] Li Li, Ke Xu, Dan Wang, Chunyi Peng, Kai Zheng, 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), 2017.

[5] China launches upgraded high-speed trains, with wi-fi. https://gbtimes.com/china-launches-upgraded-high-speed-trains-wi-fi.

[6] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5), 2008.

[7] Bbr congestion control algorithm. https://github.com/google/bbr.

[8] Ruisi He, Bo Ai, Gongpu Wang, Ke Guan, Zhangdui Zhong, Andreas F Molisch, Cesar Briso-Rodriguez, and Claude P Oestges. High-speed railway communications: From gsm-r to lte-r. *IEEE Vehicular Technology Magazine*, 11(3), 2016.

[9] Universal mobile telecommunications system (umts); lte; requirements for evolved utra (e-utra) and evolved utran (e-utran). http://www.3gpp.org/DynaReport/25913.htm. 3GPP TR 25.913 version 8.0.0 Release 8 (2009-01-02).

[10] Mark Russell and Gordon L Stuber. Interchannel interference analysis of ofdm in a mobile environment. In *IEEE VTC*, 1995.

[11] Yaoqing Yang, Pingyi Fan, and Yongming Huang. Doppler frequency offsets estimation and diversity reception scheme of high speed railway with multiple antennas on separated carriages. In *IEEE WCSP*, 2012.

[12] Sudeep Palat and Ph Godin. The lte network architecture: a comprehensive tutorial. *The UMTS Long Term Evolution: From Theory to Practice. John Wiley & Sons*, 2009.

[13] Konstantinos Dimou, Min Wang, Yu Yang, Muhammmad Kazmi, Anna Larmo, Jonas Pettersson, Walter Muller, and Ylva Timner. Handover within 3gpp lte: design principles and performance. In *IEEE VTC Fall*, 2009.

[14] Lte; evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran); overall description; stage 2. http://www.3gpp.org/dynareport/36300.htm. 3GPP TS 36.300 version 8.12.0 Release 8 (2010-04-28).

[15] Xing Li, Congxiao Bao, Maoke Chen, Hong Zhang, and Jianping Wu. The china education and research network (cernet) ivi translation design and deployment for the ipv4/ipv6 coexistence and transition. Technical report, 2011.

[16] Yuanjie Li, Chunyi Peng, Zengwen Yuan, Jiayao Li, Haotian Deng, and Tao Wang. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *ACM MobiCom*, 2016.

[17] Raymond Kwan, Cyril Leung, and Jie Zhang. Proportional fair multiuser scheduling in lte. *IEEE Signal Processing Letters*, 16(6), 2009.

[18] 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.

[19] http://soar.group/projects/hsrnet.

[20] Meet china's newest high-speed train – the fuxing hao. http://www.atimes.com/article/meet-chinas-newest-high-speed-train-fuxing-hao/.

[21] Speed limit rockets to 350 km/h. http://www.chinadaily.com.cn/business/2017-09/22/content_32327165.htm.

[22] Qxdm professional™ qualcomm extensible diagnostic monitor. https://www.qualcomm.com/documents/qxdm-professional-qualcomm-extensible-diagnostic-monitor.

[23] Haiqing Jiang, Yaogong Wang, Kyunghan Lee, and Injong Rhee. Tackling bufferbloat in 3g/4g networks. In *ACM IMC*, 2012.

[24] Jim Gettys and Kathleen Nichols. Bufferbloat: Dark buffers in the internet. *ACM Queue*, 9(11), 2011.

[25] 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 *ACM IMC*, 2013.

[26] Yin Zhang, Lee Breslau, Vern Paxson, and Scott Shenker. On the characteristics and origins of internet flow rates. In *ACM SIGCOMM*, 2002.

[27] Xian Chen, Ruofan Jin, Kyoungwon Suh, Bing Wang, and Wei Wei. Network performance of smart mobile handhelds in a university campus wifi network. In *ACM IMC*, 2012.

[28] 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. *ACM SIGCOMM Computer Communication Review*, 43(4):363–374, 2013.

[29] Emir Halepovic, Jeffrey Pang, and Oliver Spatscheck. Can you get me now? estimating the time-to-first-byte of http transactions with passive measurements. In *ACM IMC*, 2012.

[30] Wolfgang John, Sven Tafvelin, and Tomas Olovsson. Trends and differences in connection-behavior within classes of internet backbone traffic. In *PAM*. Springer, 2008.

[31] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, et al. Bbr: congestion-based congestion control. *Communications of the ACM*, 60(2), 2017.

[32] Van Jacobson. Congestion avoidance and control. In *ACM SIGCOMM*, 1988.

[33] Lisong Xu, Khaled Harfoush, and Injong Rhee. Binary increase congestion control (bic) for fast long-distance networks. In *IEEE INFOCOM*, 2004.

[34] Lawrence S. Brakmo and Larry L. Peterson. Tcp vegas: End to end congestion avoidance on a global internet. *IEEE Journal on selected Areas in communications*, 13(8), 1995.

[35] Saverio Mascolo, Claudio Casetti, Mario Gerla, Medy Y Sanadidi, and Ren Wang. Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In *ACM MobiCom*, 2001.

[36] Kun Tan, Jingmin Song, Qian Zhang, and Murad Sridharan. A compound tcp approach for high-speed and long distance networks. In *IEEE INFOCOM*, 2006.

[37] Mohamed A Alrshah, Mohamed Othman, Borhanuddin Ali, and Zurina Mohd Hanapi. Comparative study of high-speed linux tcp variants over high-bdp networks. *Journal of Network and Computer Applications*, 43, 2014.

[38] Keith Winstein, Anirudh Sivaraman, Hari Balakrishnan, et al. Stochastic forecasts achieve high throughput and low delay over cellular networks. In *USENIX NSDI*, 2013.

[39] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. Adaptive congestion control for unpredictable cellular networks. In *ACM SIGCOMM*, 2015.

[40] Feng Lu, Hao Du, Ankur Jain, Geoffrey M Voelker, Alex C Snoeren, and Andreas Terzis. Cqic: Revisiting cross-layer congestion control for cellular networks. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 45–50, 2015.

[41] Xiufeng Xie, Xinyu Zhang, and Shilin Zhu. Accelerating mobile web loading using cellular link information. In *ACM MobiSys*, 2017.

[42] Wai Kay Leong, Zixiao Wang, and Ben Leong. Tcp congestion control beyond bandwidth-delay product for mobile cellular networks. In *ACM CoNEXT*, 2017.

[43] Shinik Park, Jinsung Lee, Junseon Kim, Jihoon Lee, Sangtae Ha, and Kyunghan Lee. Exll: an extremely low-latency congestion control for mobile cellular networks. In *ACM CoNEXT*, 2018.

[44] John P Rula, Fabián E Bustamante, and David R Choffnes. When ips fly: A case for redefining airline communication. In *ACM HotMobile*, 2016.

[45] John P Rula, James Newman, Fabián E Bustamante, Arash Molavi Kakhki, and David Choffnes. Mile high wifi: A first look at in-flight internet connectivity. In *WWW*, 2018.

[46] Keon Jang, Mongnam Han, Soohyun Cho, Hyung-Keun Ryu, Jaehwa Lee, Yeongseok Lee, and Sue B Moon. 3g and 3.5 g wireless network performance measured from moving cars and high-speed trains. In *ACM MICNET*, 2009.

[47] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. A measurement study on multi-path tcp with multiple cellular carriers on high speed rails. In *ACM SIGCOMM*, 2018.

[48] Xin Liu, Ashwin Sridharan, Sridhar Machiraju, Mukund Seshadri, and Hui Zang. Experiences in a 3g network: interplay between the wireless channel and applications. In *ACM MobiCom*, 2008.

[49] Fung Po Tso, Jin Teng, Weijia Jia, and Dong Xuan. Mobility: A double-edged sword for hspa networks: A large-scale test on hong kong mobile hspa networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10), 2012.

[50] Ruben Merz, Daniel Wenger, Damiano Scanferla, and Stefan Mauron. Performance of lte in a high-velocity environment: A measurement study. In *Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges*, pages 47–52, 2014.

[51] Yin Xu, Zixiao Wang, Wai Kay Leong, and Ben Leong. An end-to-end measurement study of modern cellular data networks. In *PAM*. Springer, 2014.

[52] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, Dimitrios Lymberopoulos, Ramesh Govindan, and Deborah Estrin. Diversity in smartphone usage. In *ACM MobiSys*, 2010.

[53] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z Morley Mao, Ming Zhang, and Paramvir Bahl. Anatomizing application performance differences on smartphones. In *ACM MobiSys*, 2010.

[54] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In *ACM IMC*, 2011.

[55] Ashkan Nikravesh, David R Choffnes, Ethan Katz-Bassett, Zhuoqing Morley Mao, and Matt Welsh. Mobile network performance from user devices: A longitudinal, multidimensional analysis. In *PAM*. Springer, 2014.