More is Better? Measurement of MPTCP based Cellular Bandwidth Aggregation in the Wild

Zhixiong Niu[1,3], Zhi Wang[2], Hong Xu[1], Chuan Wu[3] and Francis C.M. Lau[3]

[1]NetX Lab, City University of Hong Kong
[2]Graduate School of Shenzhen, Tsinghua University
[3]Department of Computer Science, The University of Hong Kong

Abstract—4G/3G networks have been widely deployed around the world to provide high wireless bandwidth for mobile users. However, the achievable 3G/4G bandwidth is still much lower than their theoretic maximum. Signal strengths and available backhaul capacities may vary significantly at different locations and times, often leading to unsatisfactory performance. Bandwidth aggregation, which uses multiple interfaces concurrently for data transfer, is a readily deployable solution. Specifically, Multi-Path TCP (MPTCP) has been advocated as a promising approach for leveraging multiple source-destination paths simultaneously in the transport layer. In this paper, we investigate the efficiency of an MPTCP-based bandwidth aggregation framework based on extensive measurements. In particular, we evaluate the gain for bandwidth aggregation across up to 4 cellular operators' networks, with respect to factors such as time, user location, data size, aggregation proxy location and congestion control algorithm. Our measurement studies reveal that (1) bandwidth aggregation in general improves the cellular network bandwidth experienced by mobile users, but the performance gain is significant only for bandwidth-intensive delay-tolerant flows; (2) the effectiveness of aggregation depends on many network factors, including QoS of individual cellular interfaces and the location of aggregation proxy; (3) contextual factors, including the time of day and the mobility of a user, also affect the aggregation performance.

I. INTRODUCTION

Recent years have seen a rapid development of 3G/4G cellular network technologies, to enable faster cellular data networks. Although 3G HSPA+ and 4G LTE have been widely deployed in today's world, many performance issues still exist. Cellular data bandwidth is still insufficient for bandwidth-demanding applications such as streaming at 1080p [21]. On one hand, the median downlink speed in 4G network is typically 10.xMbps, far below the theoretical capacity of any 4G technologies [12]. On the other hand, signal coverage is still unsatisfactory: weak signal and poor performance are common in locations far away from the base stations or with surrounding obstacles. In addition, cellular bandwidth is fluctuating even when the device is static, because of the varying number of cellular users nearby [10].

Bandwidth aggregation through multiple network interfaces, *e.g.*, multiple 3G and 4G interfaces, is a promising solution to provide high data bandwidth for applications in cellular networks, on top of the existing physical-layer technologies. By concurrently transmitting traffic across independent cellular networks with different spectrum resources and base station locations, such bandwidth aggregation can increase the throughput, strengthen the channel quality, and reduce performance variation. Many protocols and techniques have been proposed to implement bandwidth aggregation at different layers of the network stack (see Sec. II for more detailed discussion). However, the effectiveness of most bandwidth aggregation approaches has not been empirically established. Most designs are evaluated by simulations under unrealistic settings, such as fixed channel bandwidth [13]. Proof-of-concept prototypes have been implemented in the work of Phatak *et al.* [20] and Chebrolu *et al.* [9], but only two or three paths are used in the prototypes, failing to prove the aggregation performance in real-world applications. To date it remains unclear under what conditions in the wild bandwidth aggregation is beneficial, and how much gain it can provide as compared to conventional single channel transmission.

In this paper, we investigate the performance of bandwidth aggregation in real 3G and 4G networks using multipath TCP (MPTCP), which has been proposed as a promising approach for leveraging multiple source-destination paths simultaneously in the transport layer. We choose MPTCP instead of other possible solutions, because of its maturity and availability [6], [25], [4]. Using MPTCP on a device requires installing MPTCP in the OS kernel, which is not feasible in millions of existing legacy systems. We adopt a proxy-based MPTCP architecture, as commonly used in the literature [24]: (1) an aggregation box with multiple radio interfaces is directly connected with an unmodified user device (*e.g.*, via Wi-Fi), and (2) proxy servers are deployed that connect to the unmodified content servers via conventional TCP, and forward traffic from the original content server to the aggregation box via multiple networks through MPTCP.

In particular, we experiment with four 3G/4G networks from four major cellular providers (3HK, CSL, CMHK, SmarTone) in Hong Kong. Our measurements are conducted in three fixed locations and three moving scenarios with different wireless conditions, over a period of six months from June 2015 to November 2015, consuming over 50GB of 3G/4G data traffic. Our goal is to measure the network under the environment without wired broadband. Hence, we focus only on 3G/4G networks without involving WiFi. This is because WiFi is not designed for large area coverage, and it requires a wired

broadband or 3G/4G network as the backbone. We measure both the average and the instantaneous download speeds experienced by the users, under different network conditions (*e.g.*, number of networks used, 3G/4G, etc.), MPTCP algorithms (including Cubic [15], Lia [18], Bilia [19]), times of the day, and download file sizes. Moreover, our experiments find that proxy location also plays a critical role, which has not been discussed in other work before.

Our key findings based on our extensive measurements are summarized as follows.

First, bandwidth aggregation via multiple cellular interfaces in general improves the bandwidth experienced by end users: more available cellular interfaces can provide more bandwidth. In our measurements, the average aggregate download speeds over 4, 3, and 2 cellular interfaces are 2.39x, 2.38x, and 1.79x larger than the single path baseline, respectively. Besides, aggregation can alleviate the impact of bandwidth fluctuation in individual cellular networks, given our observation that the bandwidth achieved through a single cellular interface can vary 2–6 times of their own smallest bandwidths in different cellular networks.

Second, though cellular aggregation is promising to improve bandwidth and reduce fluctuation, it is affected by many factors, including flow size, location of proxy, scheduling algorithm, condition of individual cellular networks, and mobility of a user. We have made the following observations: (1) The flow size is a critical factor for aggregation performance. The larger a flow is, the better the aggregation performance is, *e.g.*, aggregate bandwidth is 5 times larger when the flow size changes from 8KB to 4MB with the same number of paths used. (2) The location of the proxy server also significantly influences the aggregation performance: both round trip time (RTT) and bandwidth between the proxy server and the aggregation box play a role. (3) The quality of service in each individual cellular network is critical to the aggregation performance, with respect to the signal strength and the backhaul bandwidth allocated.

The rest of the paper is organized as follows. In Sec. II, we discuss background and related work. In Sec. III, we present the the measurement methodologies, including our experiment setup and metrics used. The measurement results and analysis on the impact of different factors are given in Sec. IV. Sec. V includes our discussions on important aspects to improve, for more efficient bandwidth aggregation in cellular networks. We conclude the paper in Sec. VI.

II. BACKGROUND AND RELATED WORK

A. Bandwidth Aggregation

Using multiple wireless interfaces to enable multiple communication paths and improve quality of service for mobile users has been investigated extensively. Based on the layer of implementation, the existing solutions can be divided into the following categories.

Application layer aggregation is implemented inside the Internet applications [5], typically as a middleware between the application and the transport layer. The middleware is responsible for handling multiple interfaces and distributing application data across them. For example, a client can establish multiple TCP connections with the server to make use of multiple available interfaces on the device to fetch videos from different Youtube servers [11]. There are also application-layer frameworks to support bandwidth aggregation (*e.g.*, [24]), which provide APIs to help users implement their aggregation algorithms, revealing information such as the Signal-to-noise ratio (SNR), Bit Error Rate (BER) and the average throughput of each interface.

Transport layer aggregation typically improves TCP to support transmission over multiple network interfaces, including MPTCP which we focus on in this paper. These solutions address issues that hinder TCP from utilizing the available interfaces in parallel, such as (1) packet scheduling across multiple interfaces, (2) reordering of packets belonging to different subflows, and (3) congestion control in the context of multiple communication paths [16], [25].

Network layer aggregation mostly assumes TCP is the target transport layer protocol, and modifies the network layer protocols to address the following issues that prevent TCP from achieving high performance on multiple interfaces [20], [9], [13]: (1) breaking TCP connection semantics, (2) congestion mis-prediction, and (3) RTT estimation. Some solutions [13], [9] aim to make the bandwidth aggregation friendly to the existing TCP.

MAC layer aggregation solutions work only when devices are directly connected through multiple links. In cellular systems, protocols have been designed to aggregate the bandwidth of two or more radio interfaces tuned to different channels [23].

In this paper, we focus on transport layer aggregation using MPTCP. The performance of MPTCP without proxy is measured in [10], where both the client and the server are installed with MPTCP. Different from previous studies, we use a proxy-based solution that provides transparent connectivity to both end users and original servers using conventional TCP.

B. MPTCP

MPTCP [4] is a TCP extension that enables simultaneous communication over multiple interfaces. It has been implemented in Linux kernel as an option [25]. Instead of establishing only one connection between two hosts, after the first *subflow* is initiated in MPTCP, the client creates other subflows via network interfaces with the "MPTCP-JOIN" option. After handshake and authentication, both endpoints associate new subflows to the existing connection [4]. Each subflow created inside an MPTCP connection is similar to an individual TCP connection, *i.e.*, conventional congestion control algorithms (*e.g.*, cubic [15]) can be used to determine the congestion window size (CWND). Several new congestion control algorithms have also been proposed specifically for MPTCP to better utilize multiple paths [22], [19].

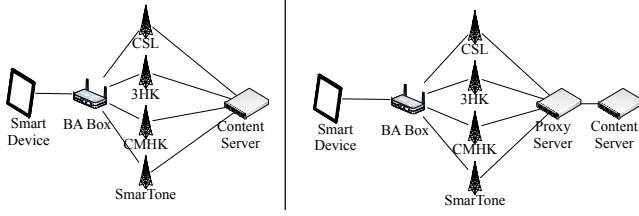## III. MEASUREMENT METHODOLOGIES

A. Experiment Setup

Fig. 1: Left: (1) Bandwidth Aggregation (BA) system without proxy; Right: (2) Bandwidth Aggregation system with proxy

To enable MPTCP-based bandwidth aggregation, we introduce a client-side aggregation box and a server-side proxy. A mobile device directly connects to the aggregation box. The aggregation box is equipped with multiple radio interfaces and serves as an MPTCP endpoint, so that unmodified devices can utilize MPTCP for bandwidth aggregation. The proxy server is the other MPTCP endpoint connected to a content server. Conventional single-path TCP is used between the aggregation box and the user device, as well as between the proxy and the content server. We study two architectures as in Fig. 1: (1) Non-proxy aggregation, which requires MPTCP to be installed by both the content server; (2) Proxy-assisted aggregation, which uses the aggregation box and the proxy server to avoid installation of MPTCP on both content servers and users.

1) Aggregation Box: In our experiments, the aggregation box is implemented as a VMWare Fusion virtual machine in an Apple Macbook Air, running Ubuntu Linux 14.04 with MPTCP v0.89 and v0.90 [7] (MPTCP was upgraded in September, 2015). Four 4G LTE USB dongles are installed on a Belkin USB hub connected to the Macbook as shown in Fig. 2.

Fig. 2: Four 4G-LTE USB dongles on a USB hub

2) Cellular Interfaces: We use Huawei E3272s-153 USB 4G LTE dongles as our 3G/4G devices, with a maximum download speed of 150Mbps and a maximum upload speed of 50Mbps. We use SIM cards from 4 major cellular service providers in Hong Kong: CSL, 3HK, CMHK and SmarTone. They provide both 3G and 4G services. The dongles provide a function to force the 4G SIM cards to run in 3G HSPA+ mode, which we use to experiment with mixed 3G and 4G networks. Notice that the bandwidth achieved in reality is affected by the signal strength, the number of users sharing the base stations serving the area, and the backhaul bandwidth allocated by the operator.

3) Proxy Server: The proxy servers in our experiments are running on virtual machines (VMs) rent from two cloud providers, QingCloud [2] and Amazon EC2, in three locations including Hong Kong, Tokyo, and California. Detailed configurations of these VMs are given in Table I.

We use two types of proxy protocols between the aggregation box and the proxy: (1) Sock5 proxy over SSH Tunnel [1], and (2) HTTP Proxy by Squid3 without cache [3]. At the same time, we also measure the performance of direct MPTCP (*i.e.*, non-proxy aggregation as given in the left of Fig. 1) for comparison.

4) Content Server: We deploy three content servers, which store files of different sizes, in the same data centers where the three proxies are deployed. When we experiment with the proxy-assisted system, the content server runs conventional TCP and communicates with users via the proxy. When we experiment without a proxy, the content servers enable MPTCP and connect with the aggregation box directly.

5) User Device: The bandwidth between the bandwidth aggregation box (BA box) and a user device is unlikely to be the bottleneck, as compared to the cellular bandwidth. So we directly use the Macbook that hosts the BA box as the user device to request files from content servers.

6) MPTCP Parameters: We set up MPTCP connections using parameters that are representative in real-world deployment: (1) *Receive buffer.* We set the size of send and receive buffers in each TCP/MPTCP connection to be 8MB, because small receive buffer may lead to inferior performance due to sharing of buffers among the subflows of an MPTCP connection [25]. (2) *Initial slow start threshold.* We use an `ssthresh` of 50 rather than infinity to avoid RTT inflation, which causes poor performance in wireless links [8], [17]. At the same time, we adopt Linux's default initial window size of 10 packets and TCP SACK [14].

B. Experiments and Metrics

We select three measurement locations in Hong Kong, including rooms at Floor G and Floor 8 near the City University of Hong Kong and an apartment room at Floor 21 near the University of Hong Kong, which are listed in Table II in detail. On the other hand, we design three moving scenarios - walking, riding on a bus, and riding on a subway train, where the signal varies from extreme weak to extreme strong.

In our experiments, we use file sizes of 8KB, 128KB, 512KB, 1024KB, 2MB, 4MB, 8MB, 16MB, 32MB, and 64MB. Small files (<=512KB) emulate short flows for web browsing, and larger files (>512KB) emulate video streaming or file downloading traffic. Note that 8KB files are common in websites. Meanwhile, the default MPTCP algorithm is Lia, unless stated otherwise.

We study how much benefit MPTCP-based bandwidth aggregation provides using the following performance metrics: (1) average download bandwidth experienced by users, *i.e.*, the average download speed; (2) download speed improvement, *i.e.*, the improvement in the average bandwidth achieved with aggregation over that of the conventional TCP. (3) The

TABLE I: Detailed configuration of proxies deployed on public clouds

Geo-location	Provider	VM Instance	Network Bandwidth	Avg. RTTs (CSL, 3HK, CMHK, SmarTone)
Hong Kong, China	QingCloud	Performance (2 Cores and 1G RAM)	150Mbps	36.90±7.24, 28.19±2.40, 30.15±2.91, 72.49±4.90
Tokyo, Japan	Amazon AWS	m4.large (2 Cores and 8G RAM)	≥100Mbps	84.31±2.40, 90.34±1.48, 81.10±3.9, 153.41±88.13
California, USA	Amazon AWS	m4.large (2 Cores and 8G RAM)	≥100Mbps	439.19±180.10, 196.43±11.50, 209.44±2.84, 355.13±181.11

TABLE II: Detailed information of experiment locations

Location	Operator	Signal Level (dBm)
Floor 21, Near HKU	CSL	-90
Floor 21, Near HKU	3HK	-92
Floor 21, Near HKU	CMHK	-88
Floor 21, Near HKU	Smartone	-78
Floor 8, Near CityU	CSL	-86
Floor 8, Near CityU	3HK	-87
Floor 8, Near CityU	CMHK	-79
Floor 8, Near CityU	Smartone	-74
Floor G, Near CityU	CSL	-108
Floor G, Near CityU	3HK	-99
Floor G, Near CityU	CMHK	-103
Floor G, Near CityU	Smartone	-92
Walking	All	-70 to -120
Riding on a bus	All	-70 to -120
Riding on a subway train	All	-70 to -120

Fig. 3: Bandwidth of different operators in one hour

contribution ratio, *i.e.*, the contribution of one operator in the experiments. (4) RTT, *i.e.*, the Round Trip Time between the client and the server. We do not use flow completion time (FCT) as a metric, since download bandwidth is more intuitive to demonstrate the improvements of bandwidth aggregation.

We write Python scripts to perform tests in batches: each experiment are repeated 15-30 times. In our scripts, we collect the following information: (1) time of the experiment, (2) downloaded data size which is recorded periodically during a download session, (3) packets in the download sessions, captured by `tcpdump`. In order to know the whole downloading process and key timing (e.g. when a MPTCP subflow is established), we analyze the captured packets by `Wireshark` and `tcptrace`. In all experiments, CPU and the USB interface are unlikely to be the bottleneck, since the CPU utilization is less than 80% and USB 2.0 is used, which is 480Mbps.

IV. IS MORE BETTER? MEASUREMENT RESULTS

We next present our measurement studies to answer the question posted earlier: how useful is bandwidth aggregation in the wild? We first confirm the benefits brought by bandwidth aggregation. Then we seek to identify important factors that affect bandwidth aggregation performance in today's cellular networks.

A. Baseline: Bandwidth of a Single Path

We first study the bandwidth achieved through a single network interface without aggregation. This serves as the baseline. In this set of experiments, we use `iperf` to measure the 4G bandwidth between a cellular interface and a content server in Hong Kong every three minutes during a one-hour period of an afternoon in June 2015. Each `iperf` session lasts for 10 seconds. We repeat the same measurement in all three testing locations. Fig. 3 presents the result: the curves are the download speeds recorded against time in different operators' networks. The main finding is that the bandwidth fluctuates wildly all the time. The largest bandwidth
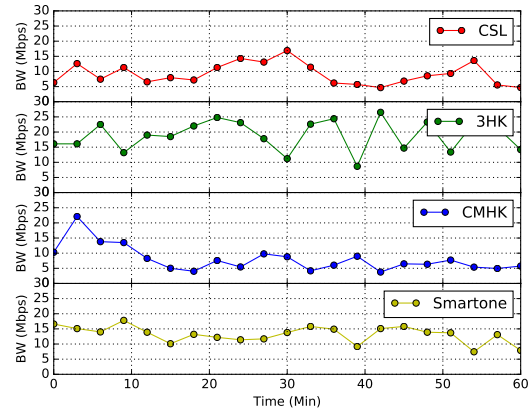
of CSL, 3HK, CMHK, and SmarTone is 16.9Mbps, 26.5Mbps, 22.1Mbps, and 17.8Mbps, respectively, and the smallest is 4.67Mbps, 8.68Mbps, 3.77Mbps, and 7.44Mbps, respectively. The largest bandwidths are 2x–6x of the smallest bandwidths. These results indicate that cellular data bandwidth experienced by users can be quite unstable even at small time scales. Further, the maximum bandwidth of different providers occurs at different times. This implies that aggregation across multiple networks potentially provides more stable and better overall bandwidth.
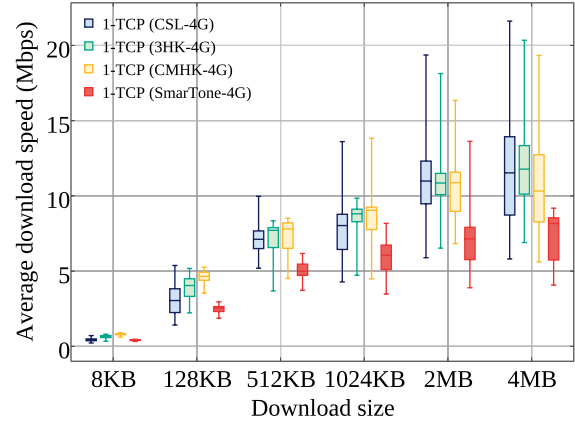

Fig. 4: Average download speed of different operators using 4G

Fig. 4 further shows average 4G LTE bandwidth of different operators with varying file sizes through one TCP connection. The results are collected from June 2015 to November 2015 in three different locations, which is stated in section III-B. We use box charts to demonstrate the results here. The line inside each box indicates the median, the top border and the bottom border of each box correspond to 25%ile and 75%ile of the values, and the ends of the whiskers are the minimum
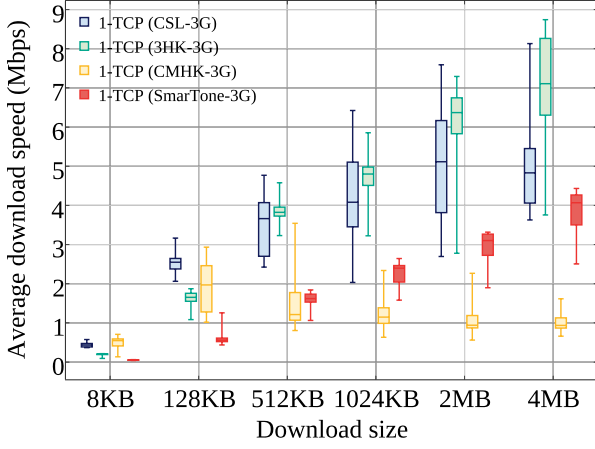
Fig. 5: Average download speed of different operators using 3G

TABLE III: Combinations of operators and modes

# of inter-faces	ISP 1	ISP 2	ISP 3	ISP 4
1	CSL (4G)	N/A	N/A	N/A
1	3HK (4G)	N/A	N/A	N/A
1	CMHK (4G)	N/A	N/A	N/A
1	SmarTone (4G)	N/A	N/A	N/A
1	CSL (3G)	N/A	N/A	N/A
1	CMHK (3G)	N/A	N/A	N/A
2	CSL (4G)	3HK (4G)	N/A	N/A
2	CSL (3G)	3HK (4G)	N/A	N/A
2	CMHK (4G)	SmarTone (4G)	N/A	N/A
2	CMHK (3G)	SmarTone (4G)	N/A	N/A
3	CSL (4G)	3HK (4G)	CMHK (4G)	N/A
4	CSL (4G)	3HK (4G)	CMHK (4G)	SmarTone (4G)
4	CSL (3G)	3HK (4G)	CMHK (3G)	SmarTone (4G)

and maximum values. We find the download performance varies significantly at different runs of the experiments due to the variation of mobile networks. For instance, the maximum and the minimum average bandwidths of CSL 4G when downloading a 4MB file are 21.6Mbps and 5.8Mbps, respectively, even when we use the exactly same configuration in each experiment. We also observe that the average bandwidth depends on the file size: the larger a file is, the higher the bandwidth achieved is. We will discuss the impact of file sizes in following experiments.

Fig. 5 shows average 3G download bandwidth. When downloading a 4MB file, 4.83Mbps, 7.10Mbps, 0.93Mbps, and 4.06Mbps are achieved by the four operators, respectively. The results show similar trends as the 4G results. Among the four operators, CMHK does not have the 3G license. It has to lease the spectrum resources from other providers, leading to inferior performance as shown in the figure. This observation indicates that the bandwidth of the cellular backhaul network is critical for the download speed experienced by users.

In the following subsections, we use CSL's 3G and 4G bandwidths as the single path baselines for comparison, since its bandwidths are moderate among the four and represent the common rather than the extreme. When presenting results of bandwidth aggregation using mixed 3G/4G networks, we use 3HK and SmarTone for 4G and CSL and CMHK for 3G. The reason is that the lowest single path bandwidth is from SmarTone for 4G and CMHK for 3G, and bandwidth aggregation is more useful when users are using these networks.

B. Effectiveness of Cellular Bandwidth Aggregation

We first evaluate data download performance when aggregating bandwidth from different cellular networks. We list in Table III the combinations of networks used in our experiments. In order to collect data at about the same time and avoid the influence of bandwidth variation over time in each cellular network, the minimal unit of experiments is a set and we repeat experiments set by set. For example, a set of experiments may consist of 1-TCP, 2-MPTCP (MPTCP via 2 operators), 3-MPTCP, 4-MPTCP. We carry out one set of experiments at one

time, rather than starting 2-MPTCP experiments after finishing all 1-TCP experiments. Additionally, the default proxy server and content server are both in Hong Kong unless stated otherwise.
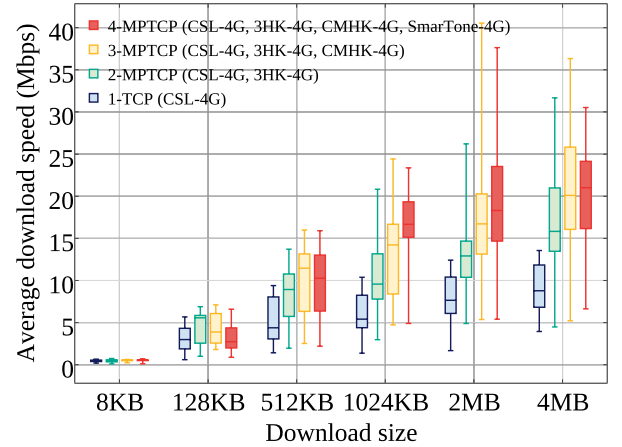

Fig. 6: Average aggregate download speed: different numbers of 4G interfaces with Sock5 proxy

In Fig. 6, we show the bandwidth aggregation performance using different numbers of cellular networks. We observe that "more is better" is in general true in proxy-assisted bandwidth aggregation. When the file size is not very small (e.g., ≥128KB), the more networks are aggregated, the larger the bandwidth is. When the file size is 4MB, the median average aggregate download speeds over 4, 3, and 2 interfaces are 2.39x, 2.38x and 1.79x of that of the single path baseline, respectively. Meanwhile, we observe that the improvement is not proportional to the number of networks used, since the difference between 3-MPTCP and 2-MPTCP is small.

Fig. 7 illustrates the average aggregated download speed with mixed 3G and 4G networks. When the file size is 4MB, the median average download speed are 15.80Mbps, 13.49Mbps, 13.65Mbps and 4.66Mbps, respectively. We observe that adding CMHK-3G does not bring any benefit, given
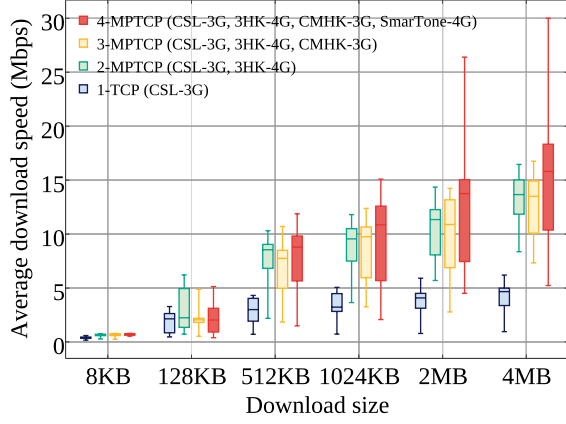
Fig. 7: Average aggregate download speed: different numbers of mixed 3G and 4G interfaces with Sock5 proxy

the very similar average aggregate bandwidth of 2-MPTCP and 3-MPTCP. The is due to the low bandwidth (about 1Mbps), high RTT (about 100ms), and high packet loss rate (0.1% - 1%) of CMHK-3G, which are enormously weaker than other operators.
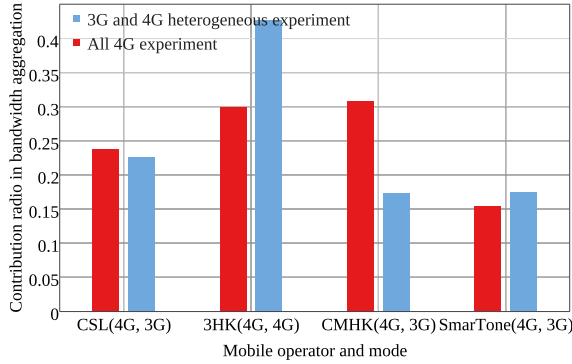

Fig. 8: Contribution in bandwidth aggregation

We also look at the contributions of different cellular operators in the aggregated bandwidth by analyzing the `tcpdump` data from the previous two experiments. Fig. 8 shows the average fraction of download rate contributed by each operator. The left red bar is the contribution radio in the 4G measurement and the right blue bar indicates the mixed 3G/4G measurement. As we observe, the usage of different paths is roughly balanced. Hence, we conclude that such bandwidth aggregation can leverage all available paths in a balanced fashion rather than rely on certain paths.

C. Impact of File Sizes

We have observed in Fig. 6 that the aggregate bandwidth is larger when the file to download has a larger size. For example, the average bandwidth achieved by 4-MPTCP when downloading a 4MB file is 44 times of that when downloading an 8KB file. The rationale is as follows. MPTCP as a TCP extension also involves the slow start process when the connection is established initially. When the file size is small, the data transfer may be completed well before the

end of slow start. In addition, the three-way handshake before data transmission also introduces relative more overhead for small file downloading, reducing the average bandwidth. Chen *et al.* [10] have also discovered that file size is a critical factor in bandwidth aggregation, in the case of direct MPTCP connection without server-side proxy. The second and the following subflows of MPTCP are established only after the first subflow has been set up successfully, which means that the download may already be done using the first subflow before or near the establishment time of other subflows. We observe by analyzing the packets by `Wireshark` that the average time of establishing a subflow is about 75 ms, and the finishing time of 8KB downloading is about 100 ms, which are very close.
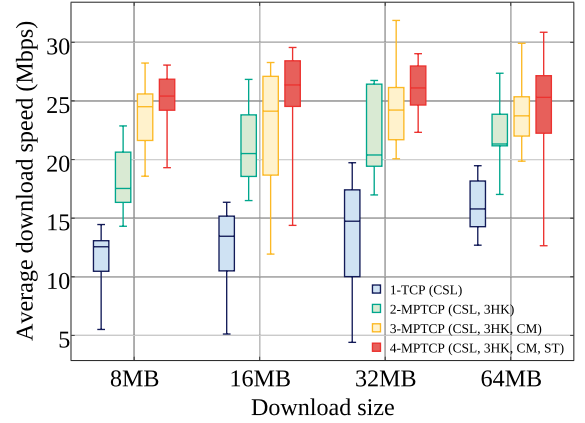

Fig. 9: Average aggregate download speed: different numbers of 4G interfaces with Sock5 proxy for downloading large files

An interesting question is whether the aggregate bandwidth is stable in different file sizes. To answer this question, we show in Fig. 9 the aggregate bandwidth achieved when downloading very large files. We observe that when the file size is larger than or equal to 8MB, the average bandwidth achieved by 2-MPTCP, 3-MPTCP and 4-MPTCP, respectively, all becomes consistent in different file sizes. This is because the MPTCP slow start process has been finished and all subflows are successfully established, *i.e.*, the connection is in a relative stable status during majority of the download time, when downloading size is more than 8MB. In video streaming scenario, the file sizes are large and the average download speed is stable while making use of all available bandwidth.

D. Impact of Proxy Locations

We next deploy the content server in AWS US West (N. California) data center. We further set up the proxy server in three different locations, QingCloud (Hong Hong), AWS Japan (Tokyo) and AWS US West (N. California). We investigate the impact of the proxy location on the achieved aggregate bandwidth. In Fig. 10, we observe the following: 1) the aggregate bandwidth achieved by 4-MPTCP with proxy assistance always outperforms that of a single TCP connection without bandwidth aggregation; 2) the aggregate bandwidth is the largest when the proxy in Japan is used, which is not
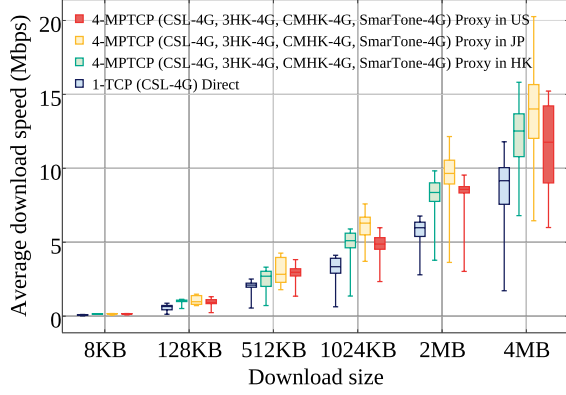
Fig. 10: Average aggregate download speed: different proxy locations - all 4G interfaces

Fig. 11: Average aggregate download speed: different times of day - all 4G interfaces

TABLE IV: Average single-path bandwidth for 4M file downloading and RTT between aggregation box and proxy via different cellular interfaces

Operator	Proxy Location	Bandwidth (Mbps)	RTT (ms)
CSL HK	QingCloud HK	11.66±3.80	36.90±7.24
3HK	QingCloud HK	11.72±2.66	28.19±2.40
CMHK	QingCloud HK	10.58±3.06	30.15±2.91
SmarTone HK	QingCloud HK	7.23±1.86	72.49±4.90
CSL HK	AWS Japan	21.44+1.25	84.31±2.40
3HK	AWS Japan	27.92±1.64	90.34±1.48
CMHK	AWS Japan	19.36±1.95	81.10±3.9
SmarTone HK	AWS Japan	14.24±1.73	153.41±88.13
CSL HK	AWS US West	7.2±1.5	439.19±180.10
3HK	AWS US West	14.48±2.32	196.43±11.50
CMHK	AWS US West	12.16±1.99	209.44±2.84
SmarTone HK	AWS US West	11.68±4.31	355.13±181.11

the nearest to the aggregation box in Hong Kong nor the closest to the content server in US West, while the performance achieved by proxies in US and HK is similar. We observe similar phenomenons when the number of interfaces to use is two or three as well.

The reason is mainly that both RTT and bandwidth between the proxy server and the aggregation box affect the aggregation performance. In our experiments, the bandwidth from the proxy in Japan to the aggregation box is the largest, rather than from the proxy located in QingCloud in Hong Kong, as given in Table IV, although the RTT to the proxy in Japan is slightly larger. On the other hand, although the bandwidth between the aggregation box and the proxy server in US is larger than that when the proxy is in Hong Kong, the corresponding RTT is larger as well, leading to similar performance when using proxies in US and Hong Kong, respectively. We hence conclude that both RTT and bandwidth between the proxy server and the aggregation box affect the aggregation performance, and RTT is usually less important for proxy location selection.

E. Impact of Time of Day

We divide a day into four time intervals: (0:00-5:59), (6:00-11:59), (12:00-17:59) and (18:00-23:59), and evaluate the aggregate bandwidth in each time interval, averaged over 20 experiments in each of them. The 20 experiments are measured
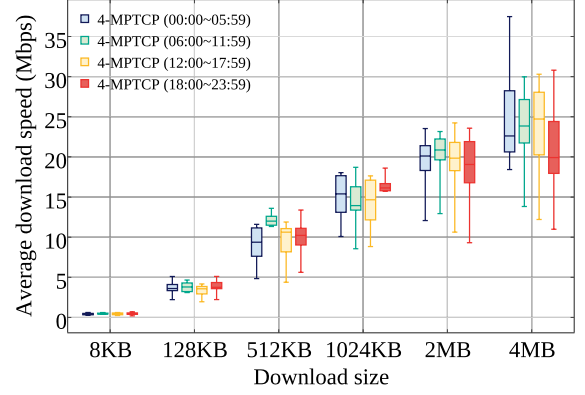
uniformly in each 6 hours. In Fig. 11, we do not observe big differences among aggregate bandwidths measured in the four time intervals. In different time intervals, the background traffic both in network and in cell can be different, and the experiment results indicate that such aggregation can work under different levels of background network loads.

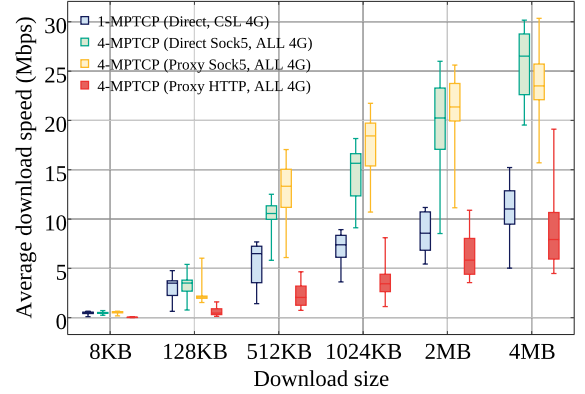### F. Impact of Different Proxy Technologies

Fig. 12: Average aggregate download speed: no proxy, Sock5 proxy, HTTP proxy - all 4G interfaces

We next investigate the impact of different proxy technologies (Fig. 1 Left), and compare the performance with the case of no proxy (Fig. 1 Right). We make several observations in Fig. 12: (1) HTTP proxy has the lowest aggregated bandwidth, and the reason is the overhead of HTTP that requires additional application handshakes. We observe the delay between HTTP request and respond could be 1s∼2s in some experiments, which could enormously slower the downloading rate; (2) Sock5 works better than HTTP proxy, since its overhead is smaller (3) Sometimes, the Sock5 proxy outperforms direct connection without a proxy. This is because a BA box maintain a continuous SSH-Sock5 TCP connection. For each experiment, the only connection is established between the proxy server and the content server whose RTT is about 10ms. In terms of the direction connection without a proxy, the connection is set up between the BA Box and the content server, whose RTT is higher (30ms).
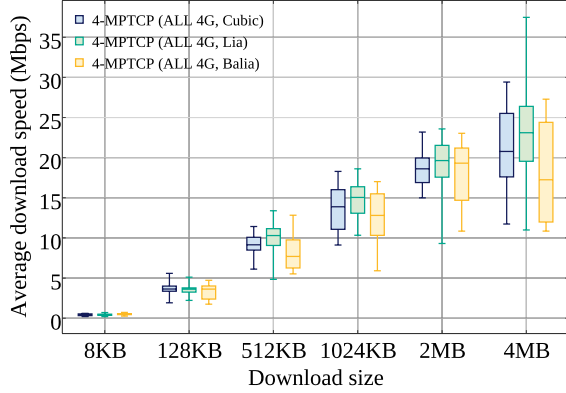
G. MPTCP Algorithm

Fig. 13: Average aggregate download speed: different MPTCP algorithms, no proxy, all 4G interfaces

In this experiment, we investigate the impact of the congestion control algorithms used in MPTCP, including cubic (congestion windows for subflows are controlled by a cubic function), lia (congestion windows are controlled according to a tradeoff between optimal congestion balancing and responsiveness) and balia (a balanced congestion control across multiple paths) [19]. In Fig. 13, we compare the aggregation performance of these algorithms over 4 cellular networks. We observe that these algorithms perform very similarly, which are 20.77Mbps, 23.104Mbps, 17.24Mbps in 4M downloading, respectively, and lia only slightly outperforms the other two. These observations indicate that in cellular bandwidth aggregation, congestion control in individual subflows is less important.

H. Impact of User Mobility

Finally, we evaluate bandwidth aggregation performance when the user device and the aggregation box are on a move. Fig. 14a shows the results when a person carrying the devices are walking near Cornwall Street (22.339606N, 114.169161E) to (22.340053N, 114.178731E). Fig. 14b shows the results collected when the person is taking Bus No. 682 in Kowloon. During the route, the bus passes some tunnels and bridges. Fig. 14c shows the results on a subway train, *i.e.*, Mass Transit Railway (MTR) of East Railway Line, West Railway Line and Ma On Shan Line. The data are collected through repeating each experiment for >15 times. We observe that bandwidth aggregation is alway beneficial in all scenarios. Meanwhile, we can see that the results are not very stable, since the network condition typically varies significantly during movement. From the users' perspective, bandwidth aggregation is beneficial. When users are moving, the signal strength is varying and connectivity provided by one cellular network is easily in trouble. Bandwidth aggregation with 2-4 interferes can largely remedy the problem, and provide a uniform, reliable, and fast network to users.

V. DISCUSSIONS

We study the performance of cellular bandwidth aggregation in the wild using real-world experiments. Our MPTCP aggre-

gation solution based on an aggregation box with multiple cellular interfaces and a cloud-based proxy can *transparently* improve wireless bandwidth experienced by users. Based on our measurement studies, we discuss for what applications such solution can benefit users, what parameters we can control to improve the performance, and what mechanisms are still in demand for its practical use.

A. Aggregation for Delay-Tolerant Large Flows

We have observed that the aggregation may only be beneficial for flows with certain characteristics. First, the size of flows in the aggregation affects the aggregation performance, *i.e.*, large flows tend to benefit more from the aggregation than small flows; second, in real-world implementation, the aggregation requires additional connections to be established between the aggregation proxy and the proxy server, which may lead to latency overhead, especially for small flows. Such latencies may cause degraded quality of experience in some applications. We believe such an aggregation solution is mostly useful for bandwidth-intensive delay-tolerant flows.

B. Proxy Server Deployment

For the solution to work transparently with today's online services, where conventional TCP is still used by default by both users and servers, aggregation boxes and proxy servers are used, to transmit data in a "TCP over MPTCP" manner. Today, the cloud computing paradigm allows us to rent virtual machines at different locations, which can serve as the aggregation servers in our design. As a result, choosing the locations of proxy servers is critical to the aggregation performance, which has been confirmed in our design.

Simple rule-based strategy like "choose an aggregation server that is closest to the original content server" might not always ensure good aggregation performance, since the bandwidth experienced by the user is affected by both the bandwidth between the original content server and the proxy server, and the bandwidth between the proxy server and the BA box. Bandwidth on the two segments usually varies over time. Good mechanisms are in need for bandwidth detection/estimation between candidate proxy servers and BA box/original content servers, for efficient proxy server deployment.

C. ISP Strategies

Performance of cellular networks is determined by the ISPs inherently. Some critical factors are controlled by the ISPs, which are usually not accessible to the aggregation system, *e.g.*, the channel allocation in a cell, the resource sharing between users, and the bandwidth allocation for the base station backhaul. Some parameters are further affected by the charging strategies of the ISPs. It will help to boost the performance of cellular bandwidth aggregation, if more knowledge of the ISPs' strategies can be acquired.

D. Knowledge of Routing Paths

MPTCP works by aggregating bandwidth from multiple routing paths between the aggregation box and the proxy
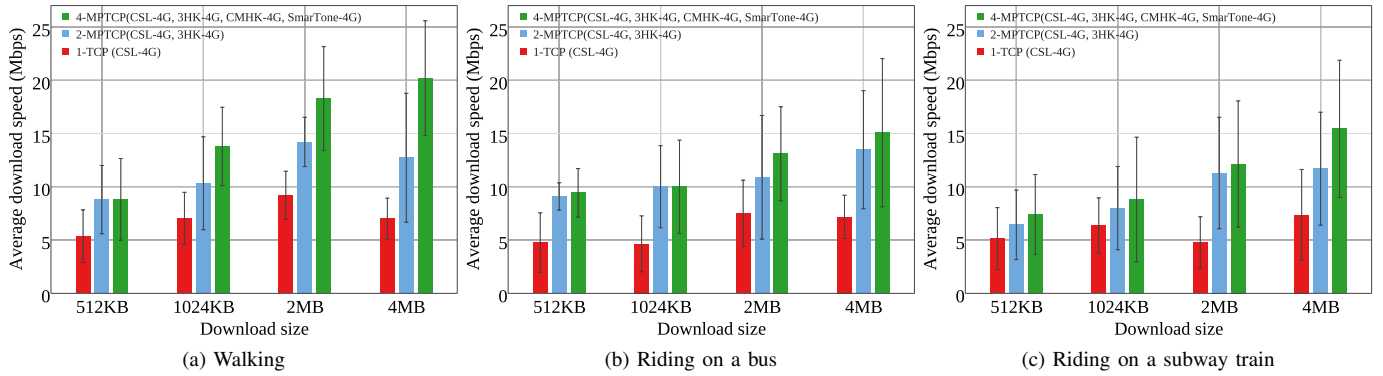
Fig. 14: Average aggregate download speed in three on-the-go scenarios

server. Thus, the routing paths can significantly affect the aggregation performance. For example, the performance for aggregation over paths sharing the same bottleneck can be worse than aggregation over paths that have different bottle-necks in the network. However, since the bottlenecks in routing paths vary over time, it is challenging to maintain updated knowledge about them. This therefore represents an intriguing direction for further research, in order to bring the performance of cellular bandwidth aggregation to the next level.

VI. CONCLUSION

This paper studies the real-world performance of a transparent cellular bandwidth aggregation solution based on MPTCP. We carry out measurements involving 4 cellular network providers for 3G/4G, and 2 cloud providers for proxy server deployment over the world. The bandwidth aggregation via multiple cellular interfaces generally improves the bandwidth in the wild. The cellular networks, the size of the downloading file, the location of the proxy server, the technology of the proxy, the user mobility, and the congestion control algorithms in MPTCP jointly affect the aggregation performance. We also discuss limitations and improvement for this solution.

VII. ACKNOWLEDGEMENTS

REFERENCES

[1] OpenSSH. http://www.openssh.com.
[2] QingCloud. http://www.qingcloud.com.
[3] Squid3. http://www.squid-cache.org.
[4] IETF RFC 6824: TCP Extensions for Multipath Operation with Multiple Addresses. 2013.
[5] G. Ananthanarayanan, V. N. Padmanabhan, C. Thekkath, L. Ravindranath, et al. Collaborative Downloading for Multi-homed Wireless Devices. In *Proc. IEEE HotMobile*, 2007.
[6] M. Becke, H. Adhari, E. P. Rathgeb, F. Fa, X. Yang, and X. Zhou. Comparison of Multipath TCP and CMT-SCTP based on Intercontinental Measurements. In *Proc. IEEE GLOBECOM*, 2013.
[7] C. Paasch, S. Barre, et al. Multipath TCP in the Linux Kernel. http://www.multipath-tcp.org.

[8] M. C. Chan and R. Ramjee. TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation. *Wireless Networks*, 11(1-2):81–97, 2005.
[9] K. Chebrolu, B. Raman, and R. R. Rao. A Network Layer Approach to Enable TCP over Multiple Interfaces. *Wireless Networks*, 11(5):637–650, 2005.
[10] 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. ACM IMC*, 2013.
[11] Y.-C. Chen, D. Towsley, and R. Khalili. MSPlayer: Multi-Source and multi-Path LeverAged YoutubER. In *Proc. ACM CoNEXT*, 2014.
[12] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan. WiFi, LTE, or Both? Measuring Multi-Homed Wireless Internet Performance. In *Proc. ACM IMC*, 2014.
[13] K. Evensen, D. Kaspar, P. Engelstad, A. F. Hansen, C. Griwodz, and P. Halvorsen. A Network-Layer Proxy for Bandwidth Aggregation and Reduction of IP Packet Reordering. In *Proc. IEEE LCN*, 2009.
[14] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky. RFC 2883: An Extension to the Selective Acknowledgement (SACK) Option for TCP. 2000.
[15] S. Ha, I. Rhee, and L. Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, 2008.
[16] J. R. Iyengar, P. D. Amer, and R. Stewart. Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths. *IEEE/ACM Trans. Netw.*, 14(5):951–964, 2006.
[17] H. Jiang, Y. Wang, K. Lee, and I. Rhee. Tackling Bufferbloat in 3G/4G Networks. In *Proc. ACM IMC*, 2012.
[18] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec. MPTCP Is Not Pareto-Optimal: Performance Issues and a Possible Solution. In *Proc. ACM CoNEXT*, 2012.
[19] Q. Peng, A. Walid, J.-S. Hwang, and S. H. Low. Multipath TCP: Analysis, Design, and Implementation. http://arxiv.org/abs/1308.3119, 2014.
[20] D. S. Phatak and T. Goff. A Novel Mechanism for Data Streaming Across Multiple IP Links for Improving Throughput and Reliability in Mobile Environments. In *Proc. IEEE INFOCOM*, 2002.
[21] K. Pires and G. Simon. DASH in Twitch: Adaptive Bitrate Streaming in Live Game Streaming Platforms. In *Proc. ACM VideoNext*, 2014.
[22] C. Raiciu, M. Handley, and D. Wischik. RFC 6356: Coupled Congestion Control for Multipath Transport Protocols. Technical report, 2011.
[23] A. L. Ramaboli, O. E. Falowo, and A. H. Chan. Bandwidth Aggregation in Heterogeneous Wireless Networks: A Survey of Current Approaches and Issues. *Journal of Network and Computer Applications*, 35(6):1674–1690, 2012.
[24] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee. MAR: A Commuter Router Infrastructure for the Mobile Internet. In *Proc. ACM MobiSys*, 2004.
[25] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, Implementation and Evaluation of Congestion Control for Multipath TCP. In *Proc. USENIX NSDI*, 2011.