An End-to-End Measurement Study of Modern Cellular Data Networks

Yin Xu, Zixiao Wang, Wai Kay Leong and Ben Leong

Department of Computer Science, National University of Singapore
{xuyin, zixiao, waikay, benleong}@comp.nus.edu.sg

Abstract. With the significant increase in cellular data usage, it is critical to better understand the characteristics and behavior of cellular data networks. With both laboratory experiments and crowd-sourcing measurements, we investigated the characteristics of the cellular data networks for the three mobile ISPs in Singapore. We found that i) the transmitted packets tend to arrive in bursts; ii) there can be large variations in the instantaneous throughput over a short period of time; iii) large separate downlink buffers are typically deployed, which can cause high latency when the throughput is low; and iv) the networks typically implement some form of fair queuing policy.

1 Introduction

Cellular data networks are carrying an increasing amount of traffic with their ubiquitous deployments and their data rates have increased significantly in recent years [1]. However, networks such as HSPA and LTE have very different link-layer protocols from wired and WiFi networks. It is thus important to have a better understanding of the characteristics and behavior of cellular data networks.

In this paper, we investigate and measure the characteristics of the cellular data networks for the three ISPs in Singapore with experiments in the laboratory as well as with crowd-sourced data from real mobile subscribers. The latter was obtained using our custom Android application that was used by real users over a 5-month period from April to August 2013. From our results, we make the following observations on the cellular data networks investigated: i) transmitted packets tend to arrive in bursts; ii) there can be large variations in the instantaneous throughput over a short period of time, even when the mobile device is stationary; iii) large separate downlink buffers are typically deployed in mobile ISPs, which can cause high latency when the throughput is low; and iv) mobile ISPs typically implement some form of fair queuing policy.

Our findings confirm that cellular data networks behave differently from conventional wired and WiFi networks, and our results suggest that more can be done to optimize protocol performance in existing cellular data networks. For example, the fair scheduling in such networks might effectively eliminate the need for congestion control if the cellular link is the bottleneck link. We also found that different ISPs and devices use different buffer configurations and queuing policies.

2 Related Work

A number of existing works have measured commercial cellular data networks. One common finding is that the throughput and latency in such networks vary significantly [9, 13]. Other works have focused on measuring and characterizing the one-way delay of 3G/HSPA networks [4, 7]. Winstein et al. also mentioned in passing that packet arrivals on LTE links do not follow an observable isochronicity [16]. Jiang et al. measured the buffers of 3G/4G networks for the four largest U.S. carriers as well as the largest ISP in Korea using TCP and examined the bufferbloat problem [6]. Our work extends their work by investigating the buffer sizes and queuing policies of mobile ISPs, and we found some surprising differences among the three local ISPs. Aggarwal et al. discussed the fairness of 3G networks and found that the fairness of TCP is adversely affected by a mismatch between the congestion control algorithm and the network's scheduling mechanism [3]. A recent study also showed various interesting effects of network protocols and application behaviors on the performance of LTE networks [5].

3 Methodology

In this section, we describe our measurement study methodology. Our experiments were conducted on the cellular data networks of the three local ISPs in Singapore, which we anonymize as A, B and C. Some measurements were taken in our laboratory at the National University of Singapore, while the rest were crowd-sourced with the assistance of real users using their personal mobile devices. For the laboratory experiments, we purchased 3G/LTE cellular data plans from each ISP and took measurements with different models of smartphones and USB modems. The LTE data plans were backward-compatible with the older HSPA and HSPA+ networks and allowed us to also access these older networks and use non-LTE-enabled mobile devices.

To obtain crowd-sourced measurements, we developed and published a measurement application, *ISPCheck* [2], on the Android Play Store. To date, it has about 50 installations and the data presented in this paper was obtained over a 5-month period from April to August 2013. During this period, 6,048 sets of experiments from 23 different users were collected, with 2,301 sets for HSPA networks and 3,747 sets for the faster HSPA+ networks. We did not include the data for LTE networks because we had relatively little data for these networks, since the LTE networks in Singapore are relatively new and the majority of subscribers have not yet upgraded to LTE.

In our experiments, the measured UDP throughput was never lower than the measured TCP throughput. This suggests that the local ISPs do not throttle UDP flows, unlike the ISPs for other countries [15]. As such, we decided to use UDP flows in all our experiments because UDP provides us with full control over the packet size and sending rate. Also, unless otherwise stated, the packet size for our experiments was 1,420 bytes (including IP headers), since we found that this was the default MTU negotiated by TCP connections in the local networks. For the experiments conducted in the laboratory, we synchronized the clock of the mobile phones to that of our server by pinging the phone over a USB connection with our server. By using pings with RTTs that are less than 2 ms, we were able to synchronize the clocks to within 1 ms accuracy. This allows us to count the packets in flight and determine the exact one-way delay in our measurements precisely. While `tcpdump` was used to log the packets in our laboratory experiments, we could not use it in ISPCheck because it requires root access to the device. So ISPCheck simply logs packet traces at the application layer. All of our results are available online[1].

[1] Our data set is currently available at `http://www.opennat.com/ispcheck`

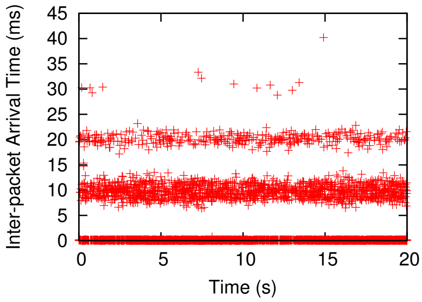


Fig. 1: Trace of the inter-packet arrival time of a downstream UDP flow for ISP C's HSPA network.

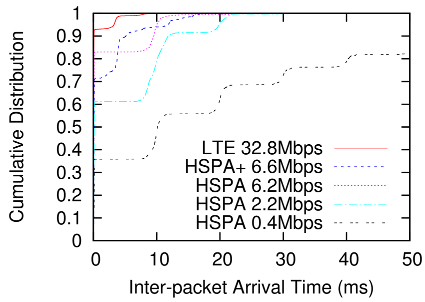


Fig. 2: Cumulative distribution of the inter-packet arrival times for ISP C.

4 Packet Flow Measurement

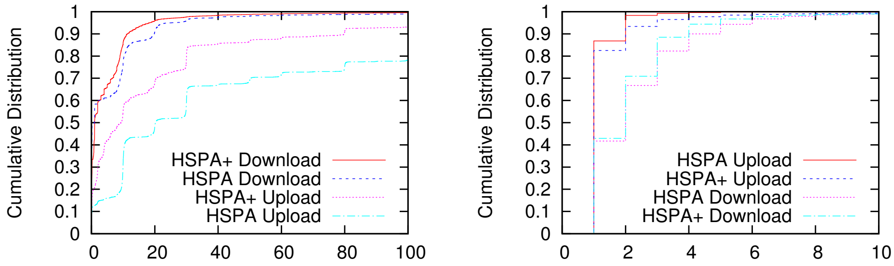
In this section, we investigate the packet flow characteristics of cellular data networks. In particular, we demonstrate that the arrival pattern of cellular data packets is bursty, and it is thus necessary to take this pattern into account when we try to estimate the instantaneous throughput for cellular data networks. Finally, we investigate how the instantaneous throughput of cellular data networks varies over time and find that it can vary by as much as two orders of magnitude within a 10-min interval.

4.1 Burstiness of Packet Arrival

In cellular data networks, packets are typically segmented and transmitted over several frames in the network link and then reconstructed at the receiver. Such networks also incorporate an ARQ mechanism that automatically retransmits erroneous frames, and this can cause packets to be delayed or reordered. To investigate the effect of the link layer protocols on the reception pattern of IP packets, we saturated the mobile link by sending UDP packets from our server to a mobile device at a rate that is higher than the receiving rate. A HTC Desire (HSPA-only) phone was used to measure existing HSPA networks and a Samsung Galaxy S4 phone was used to measure existing HSPA+ and LTE networks. We cannot use the Galaxy S4 to measure HSPA networks because it would always connect to existing HSPA+ networks by default.

One key observation is that packets tend to arrive in bursts. In Fig. 1, we plot the inter-packet arrival times of a representative trace from one of our experiments. We can clearly see that packets tend to arrive in clusters at 10 ms intervals, and that within each cluster, most packets tend to arrive within 1 ms of one another. In Fig. 2, we plot the cumulative distribution of the inter-packet arrival times for 5 traces for networks with different data rates. From these results, we can see that packet arrival is bursty at 10 ms intervals in HSPA networks and at 4 ms in the faster HSPA+ and LTE networks.

In Fig. 3(a), we plot the cumulative distribution of the inter-packet arrival times for the crowd-sourced data collected with ISPCheck. In total, the data set consisted of more than 1 million downstream packets and over 400,000 upstream packets. Again, we can see that the packets arrive in distinct bands even when the packet traces are recorded at the application layer. We consider packets that arrive within 1 ms of each other to constitute a burst, and plot the cumulative distribution of burst sizes in Fig. 3(b). We can see that the majority of downstream packets arrive in bursts. This is likely because the



(a) Inter-packet arrival time (ms) **(b)** Number of packets in one burst

Fig. 3: Inter-packet arrival times and number of packets in one burst for *ISPCheck*.

downlink of cellular data networks allows for the parallel transmission of frames which could result in multiple packets being reconstructed at the same time at the receiver.

The arrival of packets at distinct intervals of either 10 ms or 4 ms is likely due to the polling duty cycle of the radio driver in the mobile devices, but we were not able to verify this from the available hardware specifications. We noticed that older (and slower) phones like the HTC Desire had a longer interval of 10 ms, while the newer Galaxy S4 has an interval of only 4 ms. To ascertain that this was independent of the kernel tick interval, we performed the same experiments over a 802.11g WiFi network, and confirmed that there was no distinct banding of packets.

4.2 Measuring Instantaneous Throughput

Our observation of bursty packet arrivals suggests that traditional bandwidth measurement techniques using packet pairs [11] or packet trains [12] will not work well for cellular data networks. In order to obtain a reasonably good estimate of the instantaneous throughput, we would likely have to observe at least two bursts worth of packets, but even that might not be sufficient because of the coarse granularity of the clock.

To investigate the effect of bursty packet arrival on instantaneous throughput estimation, we initiated a large number of saturating downstream UDP flows (each 30 s long) over a period of time, until we found a trace where the flow seemed to be stable. Since this flow achieved an average throughput of 6.9 Mbps over the entire period, and the maximum speed of our data plan was 7.2 Mbps, we assumed that there was very little interference from other users and network traffic for this trace. Hence, any variations could be attributed to the burstiness of the packet arrivals and the transmission medium.

The packet arrivals in the trace were segmented into bursts of packets all arriving within 1 ms of each other. Next, we estimated the instantaneous throughput by using a consecutive number of n bursts. That is, we ignored the first burst and divided the data in the last $n - 1$ bursts over the total time elapsed between the n bursts. We computed all possible windows of n-bursts in the flow and plot the standard deviation and error between the estimates and the long-term average throughput of 6.9 Mbps (normalized against 6.9 Mbps) in Fig. 4 for the estimates obtained as n varies from 2 to 1,000.

As expected, the accuracy and the standard deviation of our estimates will improve if we use a larger number of bursts. However, it is not feasible to use too much data because doing so is not only costly, it might cause the measurement to take too long and the resulting instantaneous measurement might not be too meaningful. Our results in Fig. 4 suggest that using 50 bursts of packets achieves a reasonable trade-off between accuracy and data required. This translates to about 100 KB and 300 KB of data respectively, or at

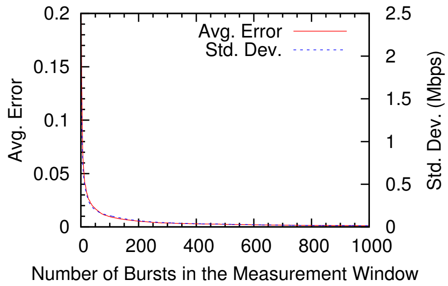


Fig. 4: The accuracy of throughput estimation with different window.

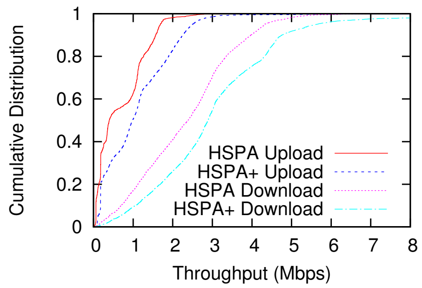


Fig. 5: Plot of cumulative distribution of the throughput for data from *ISPCheck*.

least 400 ms and 325 ms respectively in terms of time, for measuring the upstream and downstream throughputs of 2 Mbps upstream/7.2 Mbps downstream HSPA networks.

4.3 Variations in Mobile Data Network Throughput

We now present our findings on the variations in the networks that we investigated. In Fig. 5, we plot the cumulative distribution of the crowd-sourced data obtained from ISPCheck. As expected, HSPA+ networks are generally faster than HSPA networks. While HSPA+ can in principle achieve speeds higher than 7.2 Mbps, we rarely found speeds higher than that because most of the local data plans have a maximum rate limit of 7.2 Mbps. Overall, we see significant asymmetry in the upstream and downstream data rates and also that the actual throughput achieved by the local subscribers can vary significantly from a few Kbps to several Mbps.

To understand temporal variation, we initiated a 10-min long UDP flow in the HSPA+ network of ISP C and maintained a constant number of packets in flight to keep the buffer filled and ensure that the cellular link is always busy. We estimated the instantaneous throughput over the entire period using windows of 50 bursts of packets, as discussed in Section 4.1. We plot the estimated instantaneous throughput for both an upstream flow and a downstream flow in Fig. 6. We can see that not only does the throughput change fairly quickly, it also varies by as much as over two orders of magnitude several times within a 10-min interval. This corroborates the claims of previous work [13, 16] and may lead to significant degradation in TCP and HTTP performance [5].

5 Buffer and Queuing Policy

This section highlights our measurements of the buffer configurations on both ends of the cellular data networks and our investigation into the queuing policies.

Downlink Buffer Size. We estimate the buffer size by sending UDP packets at a rate higher than the receiving rate, which causes the buffer to fill over time with packets and eventually overflow. We can accurately determine the number of outstanding packets in the network, or packets in flight, by synchronizing the clock of our mobile phones to that of the server. Finally, we can estimate the buffer size by subtracting the measured bandwidth-delay product from the total packets in flight. Interestingly, we found that instead of being conventionally sized in bytes, the downstream buffers at the ISPs are sized in packets. In these experiments, we vary the size of the packets from 200 to 1,420 bytes. We could not use packets smaller than 200 bytes because our receiving devices and `tcpdump` are not able to process such small packets fast enough when we try to saturate the networks to measure the buffer size.

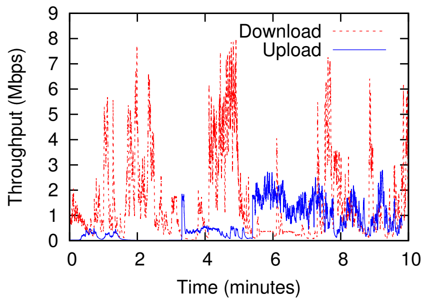


Fig. 6: The huge variation of the download and upload throughput for ISP C's HSPA+ network.

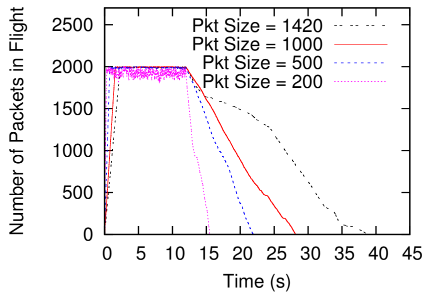


Fig. 7: The number of packets in flight for downloads with different packet size for ISP C's HSPA network.

Fig. 7 shows the plot of packets in flight against time for one of our experiments using different packet sizes over ISP C's HSPA network. We can see that the number of packets in flight plateaus at the same value for different packet sizes. In this instance, the bandwidth delay product was small ($\approx$ 50 packets), and so we deduced that the buffer size was fixed at about 2,000 packets. We observed similar behavior in the downstream buffers for all the networks studied, with the exception of ISP A's LTE network.

The downstream buffer for ISP A's LTE network behaved quite differently from the rest. As shown in Fig. 8, the buffer size seems to be a linear function of the throughput (c.f. $y = 0.8x$). In other words, the size of the buffer appears to vary proportionally to the throughput in a way that keeps the maximum queuing delay constant at 800 ms. We suspect that ISP A might have implemented a Codel-like [10] AQM mechanism in their network, i.e., packets are timestamped when they arrive, and checked at the head of the queue. Packets that spent more than 800 ms in the buffer would be dropped. While there is certainly an absolute limit of the buffer in terms of physical memory space, we were not able to exceed that even when we sent packets at the maximum supported data rate. A summary of the estimated buffer sizes for all three local ISPs is shown in Table 1.

Overall, we observed that the downstream buffers for most of the ISP networks are fairly large. Because the variation in the throughput can be very large, it is possible on occasion for the latency to become very high when throughput is too low to drain the buffer fast enough [6]. By controlling the maximum time that a packet can spend in the buffer (like in ISP A's LTE network), the maximum latency can however be kept at a stable value (about 800 ms for ISP A's LTE network) independent of the throughput.

Drop Policy. We also investigated the drop policy of the various ISPs by studying the traces of the packet losses and found that a drop-tail policy was implemented in all the networks except for ISP B's HSPA(+) and ISP A's LTE network. We repeated our experiments several times with different parameter settings and at different physical locations, and consistently obtained the results summarized in Table 1.

We explain how we inferred the drop policies with the following examples: in Fig. 9(a), we plot the number of packets sent, packets lost and packets in flight over time for ISP C's HSPA(+) network, and in Fig. 9(b), we plot a corresponding trace for ISP B's HSPA(+) network. Because the traces are analyzed offline, we could determine the lost packets by observing that they were sent but never received. However, we cannot determine precisely when the packet losses happened. Hence, the "Lost" line in our graphs refers to the time when the lost packets were sent and not when they were actually dropped. We see

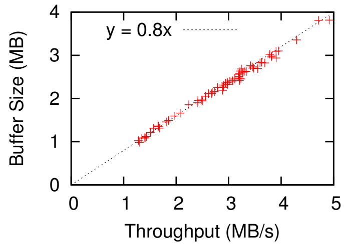
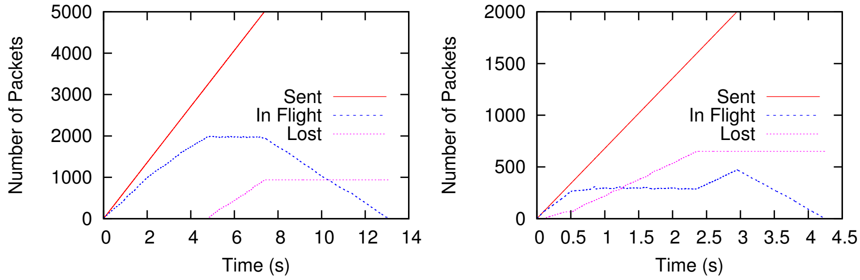


Fig. 8: In ISP A's LTE network, the effective buffer size seems to be proportional to the throughput.

Table 1: Downlink buffer characteristics for local ISPs

ISP	Network	Buffer Size	Drop Policy
ISP A	HSPA(+)	4,000 pkts	Drop-tail
	LTE	($\leq$ 800 ms)	AQM
ISP B	HSPA(+)	400 pkts	Drop-head
	LTE	600 pkts	Drop-tail
ISP C	HSPA(+)	2,000 pkts	Drop-tail
	LTE	2,000 pkts	Drop-tail



(a) ISP C HSPA(+)

(b) ISP B HSPA(+)

Fig. 9: Trace of the packets sent, lost and in flight in a UDP downstream flow.

in Fig. 9(a), that for ISP C's network, packet losses only occur to packets sent after time $t = 5$. This also coincides with the start of a plateau in the number of packets in flight because we exclude known lost packets when plotting the number of packets in flight. Thus, we can infer that Fig. 9(a) suggests a drop-tail queue, where the buffer is fully saturated around time $t = 5$ and newly sent packets are dropped until no more packets are sent at time $t = 7.2$ and the buffer starts to empty.

In contrast, Fig. 9(b) paints a very different picture for ISP B's network. We see that packet losses start to occur very early in the trace and stop after time $t = 2.4$, i.e., there were no losses for the final batch of 400 packets sent after time $t = 2.4$. This suggests a drop-head queuing policy. In addition, the packets in flight plateaus at a lower value before increasing to a peak from time $t = 2.4$ to $t = 3$. The explanation for this observation is that the line for packets in flight excludes the lost packets even though for a drop-head queue, they would have occupied space in the buffer before they get dropped at the head of the queue. Thus, our estimate of the packets in flight is an underestimate of the actual value while packets are dropped at the head of the buffer. From time $t = 2.4$ to $t = 3$, the older packets in the buffer are still being dropped but no new packet are lost. Hence, the proportion of packets dropped decreases, which explains why our estimate of the packets in flight gradually increases to the true value at $t = 3$.

Uplink Buffer Size. The uplink buffer is at the radio interface of the mobile device, and for all the mobile phones we tested, the buffer is sized in terms of bytes rather than number of packets like the downlink buffer. In Fig. 10, we plot the bytes in flight over time for the experiments carried out on a HTC Desire phone. We see that the number of bytes in flight remained constant for different packet sizes. On the other hand, the Huawei USB modems we tested had buffers that were sized in terms of number of packets. Our results are summarized in Table 2.

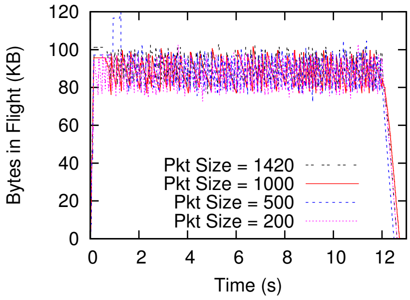


Fig. 10: The bytes in flight for uploads with different packet sizes for HTC Desire.

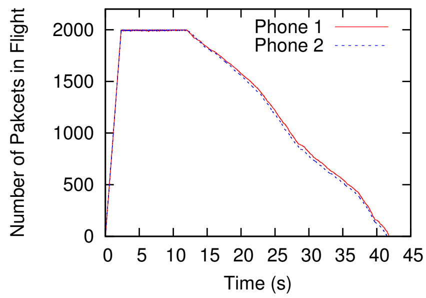


Fig. 11: The number of packets in flight for two concurrent downloads for ISP C's HSPA network.

Table 2: The radio interface buffer size of different devices

Device Type	Model	Network	Buffer Size
Android Phone	HTC Desire	HSPA	64 KB
	Galaxy Nexus	HSPA+	1.5 MB
	Galaxy S3 LTE†	HSPA+	200 KB
		LTE	400 KB
	Galaxy S4†	HSPA+	200 KB
		LTE	400 KB
USB Modem	Huawei E3131	HSPA+	300 pkts
	Huawei E3276	LTE	1,000 pkts

†These devices have additional buffering of 1,000 packets in the kernel.

Another interesting finding is that the newer Samsung Galaxy S3 LTE and Galaxy S4 phones seem to buffer packets in the kernel (which is sized in packets), in addition to the regular buffer in the radio interface (which is sized in bytes). Our measurement application was blocked from sending UDP packets once there were about 200 packets in the kernel buffer. This behavior was unexpected because we do not typically expect UDP packet transmissions to be blocked and indeed, this was not observed in the older Android phones. It is plausible that the phone manufacturers have come to realize that because the uplink bandwidth can sometimes be very low, not blocking UDP transmissions would likely cause packets to be dropped even before the phone can get a chance to transmit them, and thus have modified the kernel to implement blocking even for UDP transmissions. To further investigate this phenomenon, we tethered the phone to a desktop computer via USB and used the desktop as the packet source, instead of an Android application. By running `tcpdump` on the USB and the radio interfaces of the phone, we can directly observe the flow of packets through the phone. In these experiments, we found that the buffering in the kernel was 1,000 packets for both the Galaxy S3 LTE and S4. There was no evidence that packets were buffered in the kernel for the other Android phone models that we investigated.

Separate Downlink Buffers. Winstein et al. claimed that ISPs implement a separate downlink buffer for each device in a cellular data network [16]. To verify this claim, we performed an experiment where we started saturating UDP flows to two mobile phones concurrently connected to the same radio cell. If there was a common buffer, we will likely see differences as the packets for the two flows jostle for a place in the common buffer. Instead, in Fig. 11, we can see that the packets in flight reach the same and constant value for both phones, indicating that it is unlikely for the buffer to be shared between the devices. We observed the same behavior for all the three ISPs.

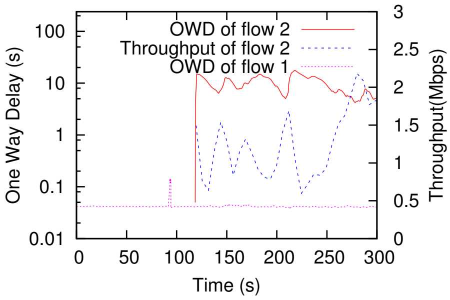


Fig. 12: Comparison of delay-sensitive flow and high-throughput flow for ISP C's HSPA network.

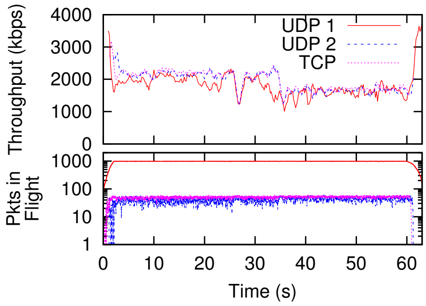


Fig. 13: The throughput and packets in flight of three downlink flows for ISP C's HSPA network.

Queuing Policy and Fairness. To investigate if the ISPs implement a fair scheduling algorithm such as Round Robin, Maximum C/I and Proportional Fair as specified in [14], we ran the following experiment: using two mobile phones connected to the same cell with the same signal strength, we sent a UDP flow to one of the phones at the constant rate of one 50-byte packet every 10 ms. After 2 min, we started a saturating UDP flow to the other phone using 1,420-byte packets and saturated the buffer by maintaining 1,000 packets in flight. The first flow mimics a low-throughput, delay-sensitive application, while the second mimics a high-throughput application. In Fig. 12, we plot the downstream one-way delay (OWD) of both flows together with the throughput of the second saturating UDP flow. If the queuing policy were FIFO, we would expect that since flow 2 saturates the buffer, the one-way delay for flow 1 would greatly increase. Instead, our results show that the delay of flow 1 remains low and stable throughout.

To investigate if the scheduling policy was fair among devices, we designed another experiment using three HTC Desire mobile phones connected to the same cell with similar signal strength. A downstream flow was initiated to each phone: i) a UDP flow that maintains 1,420 KB of data in flight, ii) a UDP flow that maintains 64 KB of data in flight, and iii) a TCP flow whose maximum receiver window was set at 64 KB. In Fig. 13, we plot the throughput of all three flows with the number of packets in flight. It turns out that the throughput is fairly distributed among the three devices, independent of the number of packets in their buffer. We repeated this experiment for the HSPA(+) networks of all three local ISPs and found similar results.

We make several observations from the results of our experiments. First, all the ISPs clearly implement some form of fair queuing and unlike in the core Internet, UDP and TCP traffic seem to be treated equally by our local mobile ISPs. While we could observe this behavior end-to-end, we could not determine if the fairness was enforced at the MAC layer or within the network. Second, having more data in flight may not help increase throughput because flows are effectively separated and do not compete for the same buffer space at a cellular base station. Instead, if the throughput is low, saturating the buffer will only result in increased latency. Third, since the fairness among connected mobile devices is enforced by a scheduling policy, congestion control at the transport layer (i.e. TCP) may not be necessary across a cellular link. This suggests that if the cellular link is the bottleneck link, which is common in the older HSPA networks, an end-to-end approach to congestion control may be possible [16]. Also, it is possible for an end-to-end flow to be split at the gateway of the cellular data network and a more efficient protocol can be used on the cellular link [17, 8].

6 Conclusion

In this paper, we showed that the packet arrivals in cellular data networks are bursty and that this burstiness needs to be taken into account when estimating instantaneous throughput. We verified that the throughput of existing networks can vary by as much as two orders of magnitude within a 10-min interval, and found that mobile ISPs often maintain large and separate downlink buffers for each user. The ISPs also implement some form of fair queuing, but for different networks, the buffer management policies may be quite different. Whether these configurations are optimal and what makes a configuration optimal are candidates for further study. We believe that our observations would be useful for the design and optimization of protocols that work with cellular data networks.

Acknowledgment

This research was carried out at the SeSaMe Centre. It is supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.

References

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20122017 .
2. ISPCheck. https://play.google.com/store/apps/details?id=com.ispcheck.
3. V. Aggarwal, R. Jana, K. Ramakrishnan, J. Pang, and N. K. Shankaranarayanan. Characterizing Fairness for 3G Wireless Networks. In *Proceedings of LANMAN '11*, Oct. 2011.
4. A. Elmokashfi, A. Kvalbein, J. Xiang, and K. R. Evensen. Characterizing Delays in Norwegian 3G Networks. In *Proceedings of PAM '12*, Mar. 2012.
5. J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. In *Proceedings of SIGCOMM '13*, Aug. 2013.
6. H. Jiang, Y. Wang, K. Lee, and I. Rhee. Tackling Bufferbloat in 3G/4G Networks. In *Proceedings of IMC '12*, Nov. 2012.
7. M. Laner, P. Svoboda, E. Hasenleithner, and M. Rupp. Dissecting 3G Uplink Delay by Measuring in an Operational HSPA Network. In *Proceedings of PAM '11*, Mar. 2011.
8. W. K. Leong, Y. Xu, B. Leong, and Z. Wang. Mitigating Egregious ACK Delays in Cellular Data Networks by Eliminating TCP ACK Clocking. In *Proceedings of ICNP '13*, Oct. 2013.
9. X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang. Experiences in a 3G Network: Interplay Between the Wireless Channel and Applications. In *Proceedings of MobiCom '08*, Sep. 2008.
10. K. Nichols and V. Jacobson. Controlling Queue Delay. *Queue*, 10(5):20:20–20:34, May 2012.
11. V. Paxson. End-to-end Internet Packet Dynamics. In *Proceedings of SIGCOMM '97*, Sep. 1997.
12. V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell. pathChirp: Efficient Available Bandwidth Estimation for Network Paths. In *Proceedings of PAM '03*, Apr. 2003.
13. W. L. Tan, F. Lam, and W. C. Lau. An Empirical Study on the Capacity and Performance of 3G Networks. *IEEE Transactions on Mobile Computing*, 7(6):737–750, Jun. 2008.
14. P. Tapia, J. Liu, Y. Karimli, and M. J. Feuerstein. *HSPA Performance and Evolution: A Practical Perspective*. WILEY, 2009.
15. F. P. Tso, J. Teng, W. Jia, and D. Xuan. Mobility: A Double-Edged Sword for HSPA Networks. In *Proceedings of MobiHoc '10*, Sep. 2010.
16. K. Winstein, A. Sivaraman, and H. Balakrishnan. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *Proceedings of NSDI '13*, Oct. 2013.
17. Y. Xu, W. K. Leong, B. Leong, and A. Razeen. Dynamic Regulation of Mobile 3G/HSPA Uplink Buffer with Receiver-Side Flow Control. In *Proceedings of ICNP '12*, Oct. 2012.