Uncovering Cellular Network Characteristics: Performance, Infrastructure, and Policies

Junxian Huang§ Feng Qian§ Qiang Xu§
Zhiyun Qian§ Z. Morley Mao§ Ammar Rayes†
§ *University of Michigan* † *Cisco*

ABSTRACT

Mobile Smart Devices (Smartphones and tablets) have become increasingly popular especially in IT based service management companies. According to IDC, more than 70% of executives and sale managers are replacing their PCs with tablets. This is in part due to the agility flexibility and the availability of diverse network-based and support applications. Thus network characteristics directly affect user-perceived performance and a deep understanding of the properties of contemporary cellular networks for commonly used platforms is important for smartphone application and platform optimization.

In this work, we carry out the largest study to date of cellular networks in terms of users, time duration, location, and networks to understand the performance, infrastructure, and policy characteristics. With the data set collected from around 100K users across the world over 18 months, **`MobiPerf`**, a smartphone network measurement tool we developed and publicly deployed, enables us to analyze network performance along several new dimensions, previously not examined. Our results indicate that with better infrastructure support, large cities appear to have better performance than rural areas. Our case study on packet size's effect on RTT uncovers a surprising pattern for AT&T's uplink RTT. We also show that Internet-based CDN service provides very limited latency improvement in today's cellular networks. We further examine how local DNS servers are assigned to mobile users. In addition, we scrutinize the carriers' policy towards different types of traffic and successfully identify some middlebox behavior of today's cellular carriers.

1. INTRODUCTION

Given the wide adoption of smartphone platforms, such as iOS and Android, there is a growing number of popular mobile applications designed for these platforms. For many of these applications, including web browser, email, VoIP, social networks, network access is required. Even for games that are often run locally, ranking systems and online peer matching systems are widely adopted which also requires network access, *e.g.,* Game Center for iOS. As a result, mobile data traffic volume is sky-rocketing. For example, AT&T's mobile data volumes surged by a staggering 8,000% from 2007 to 2010 [1]. Given the limited network resources available, it would not be surprising if a carrier enforces different policies depending on the traffic types or users. Hence, it is critical to understand performance, infrastructure, and policy in cellular networks.

Our previous study [11] comparing cellular network performance among different carriers has already shown indication that network can be the bottleneck accounting for poor application performance. In this work, with a data set collected for a much longer period (18 months) and a larger user set (about 10,000 unique users across three major smartphone platforms), we study the correlation between performance and several important new dimensions, including network types, location, time, *etc.* to delve deeper into cellular network behavior. In addition, we have conducted local experiments to understand how packet size affects end-to-end latency. Our study uncovers important characteristics for cellular network performance.

We are motivated by our previous study on cellular network infrastructure [19]. In this study, with a comprehensive latency measurement data set, we perform more in-depth analysis and quantify the effectiveness of CDN servers for cellular networks. We also provide fine-grained analysis of the geographical coverage of each individual IP address of local DNS (LDNS) servers and discuss the implications.

Traffic differentiation has long been studied in the Internet [20, 9] given the controversy surrounding the idea of network neutrality. For mobile networks, most policies in cellular networks remain unknown. Our effort shares the same goal as the WindRider [5] project for monitoring mobile network neutrality, but we are the first to report some conclusive results. In this work, we make one of the first attempts to uncover these policies for cellular carriers that affect application performance.

The key contributions and results in this work include:

- We compare the performance across different mobile network technologies (WiFi, 3G, EDGE, GPRS) and within 3G technologies (1xRTT, EVDO, UMTS, HSDPA), providing the largest scale and most comprehensive performance comparison to date.
- We analyze the correlation between RTT and packet size, and find that uplink latency for AT&T's 3G network is a step function of packet size, in contrast to packet size independent RTT behavior for T-Mobile.
- We show little correlation between network latency and physical distance, and rather limited effectiveness of CDN service in today's cellular networks.

- We study traffic policy of cellular carriers and successfully detect some middlebox behavior for T-Mobile. With data collected from global users, we also make one of the first studies of middlebox behavior across locations and carriers.

The remainder of this paper is organized as follows: In §2, we discuss the methodology for local experiments. Then in §3, results related with cellular network performance are discussed, followed by our study on network infrastructure in §4. We discuss policy in cellular networks in §5, before summarizing related work in §6 and concluding in §7.

2. METHODOLOGY

In this study, we use measurement data collected from a publicly deployed tool **`MobiPerf`**[1] as well as from local experiments. **`MobiPerf`** is designed to collect anonymized network measurement information directly from end users, including network type, carrier, GPS, latency to landmark server, as well as TCP and DNS performance. The key methodology for designing **`MobiPerf`** has been discussed in our previous work [11, 19]. In this section, we discuss the improvements and the setup for local experiments.

We have conducted two major sets of local experiments. These tests are implemented as Android applications running on our Android smartphones locally. For both experiments, at every second, the device sends and receives a large packet (MTU) to ensure the radio interface is at high power state to occupy the high speed data transmission channel [15].

RTT vs. Packet Size We study the correlation between RTT and packet size for different carriers using TCP and UDP (for both uplink and downlink) to explore possible effect of packet size on end-to-end delay. The packet size increases from 100 bytes (including headers) with an increment of 25 bytes. The response packet has 1 byte payload to focus on a single direction at a time.

Port Scanning We use three Android smartphones with AT&T, T-Mobile, and Verizon 3G service enabled respectively. The ports scanned are either popular Internet ports or special ports for mobile platforms, *e.g.,* port 5228 is used by various Android services including Android Market, and port 5223 is used by Apple's push notification service. For each port, there is a TCP server and a UDP server running on a local host to simply echo back any message received. At the client side, an Android app first connects to the TCP server by sending a short (100 bytes including headers) unique message. The client then sends another short unique message to the UDP server. The client measures the time to establish a TCP connection, to get the response back in TCP or UDP. Besides UDP port 161 (SNMP), no other port is blocked by the firewall at the server side. We do not measure TCP data transfer time for TCP ports 22, 80 and 443 for simplicity. All ports are scanned sequentially with the entire scanning process repeated for more than 48 hours.

[1] **`MobiPerf`** is the newer version the measurement tool 3GTest we developed, with various functionality and UI improvements

3. MOBILE NETWORK PERFORMANCE MEASUREMENT AND ANALYSIS

We first describe the public deployment of our mobile network measurement tool, and then present the performance analysis along dimensions including technology type, time, and location. In addition, we study in more depth the correlation between RTT and packet size in cellular networks.

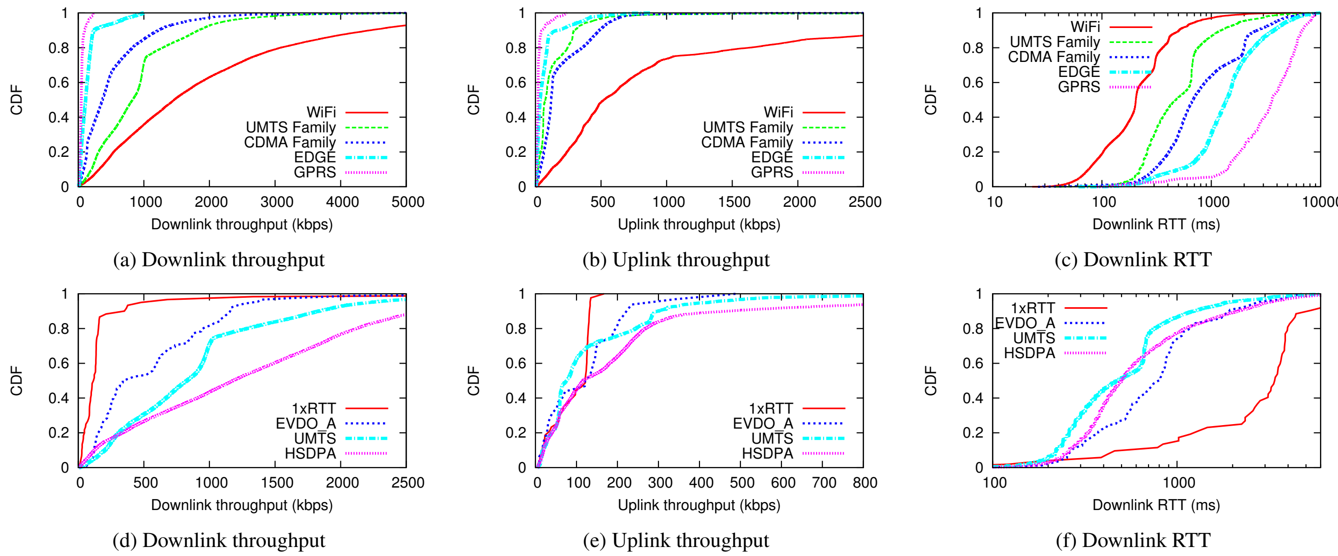
3.1 MobiPerf Deployment and User Statistics

We publicly deployed the **`MobiPerf`** application in August, 2009, distributed via Apple's App Store, Google's Android Market and Microsoft's Windows Marketplace for Mobile. Ever since the initial deployment, we have been continuously improving and releasing updates for iOS and Android version of our app. Till April, 2011, 99.1K users from across the world have run our app for 439.5K times. The number of users and runs for three different platforms, including iOS, Android, and Windows Mobile, is listed in Table 1. The average number of runs for each Android user is larger than the other two platforms, because for the Android version of our app, we give an option to the users to periodically run the tests. We observe users from 179 countries or regions according to the collected GPS information. Among all 93.3K users, 63.7K (68.27%) have GPS readings and 52.24% of them are from the U.S., and among these 63.7K users, about 1.0K (1.57%) users have run our app in more than one countries or regions. We also observe more than 800 carrier names. However, carriers may adopt different names in different countries, making it difficult to accurately estimate the actual number of carriers. Figure 2 shows the user coverage of **`MobiPerf`**, with one dot representing one run of **`MobiPerf`**. Given the wide coverage of regions, we believe our data set is fairly representative of the entire smartphone population, especially for North America with denser user distribution. In this study, our analysis mostly focuses on U.S. users.

3.2 Performance Comparison among Mobile Network Technologies

In our previous study [11], we have compared network performance among different major U.S. cellular carriers. In this study, we focus on comparing performance among technology types. We first break down technology types into WiFi, UMTS family, CDMA family, EDGE and GPRS. Within 3G family, we select 4 major network types, including HSDPA, UMTS (without HSDPA), 1xRTT and EVDO_A, since these network types cover most 3G users.

Downlink throughput is compared in Figure 1 (a). WiFi has the best performance with median throughput of 1.46 Mbps. For 3G network, UMTS family appears to outperform CDMA family, with median downlink throughput of 964 kbps compared to 368 kbps. EDGE lags with median



(a) Downlink throughput (b) Uplink throughput (c) Downlink RTT

(d) Downlink throughput (e) Uplink throughput (f) Downlink RTT

Figure 1: Downlink/Uplink performance comparison among different types of networks

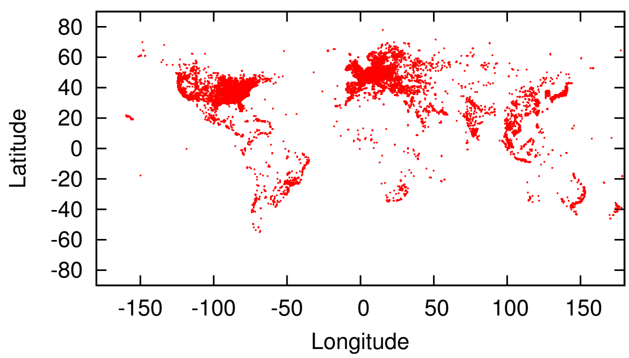


Figure 2: User coverage of `MobiPerf`

	Android	iOS	Win Mobile	All
User	39.3K 39.7%	47.0K 47.4%	12.8K 12.9%	99.1K
Run	273.8K 62.3%	127.0K 28.9%	38.7K 8.8%	439.5K

Table 1: User and run breakdown of different platforms

downlink throughput 112 kbps and GPRS is the slowest at 45 kbps. The ranking of downlink retransmission rate is consistent with that of downlink throughput, except that UMTS and CDMA have similar retransmission rate distribution. In Figure 1 (c), UMTS's median RTT is 495 ms, smaller than CDMA's 680 ms. This helps explain the throughput gap between UMTS and CDMA, since TCP throughput is lower with higher RTT and loss rate. Among 3G networks shown in Figure 1 (d), HSDPA has the highest median downlink throughput of 1.18 Mbps and 1xRTT has the smallest throughput of 115 kbps, since it is one of the earliest CDMA 3G technologies. Similarly, we observe high RTT in Figure 1 (f) and high retransmission rate for 1xRTT correlated with its low throughput.

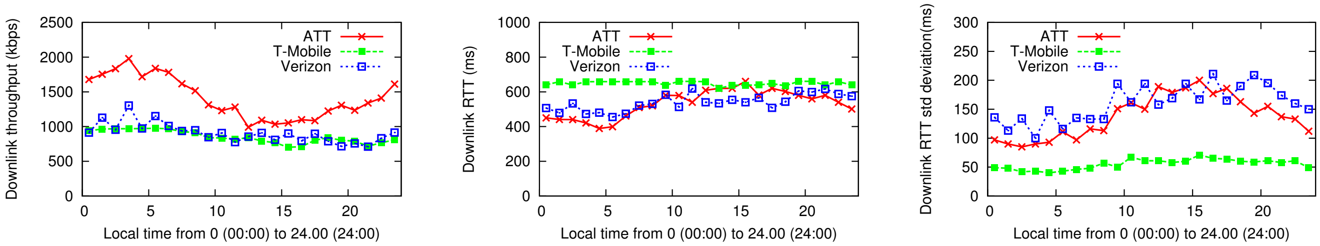
We closely study the variation of TCP downlink RTT, often known as jitter. Compared with WiFi, whose median of RTT standard deviation is 41 ms, UMTS has a higher value of 93 ms and 233 ms for CDMA. If applications running on smartphones fail to tolerate this high variation in RTT, user experience would be degraded.

In Figure 1 (b) , the uplink throughput difference between UMTS and CDMA is less obvious compared with downlink. For example, at 50^{th} percentile, UMTS's uplink throughput is 110 kbps and CDMA's is 120 kbps. Within the 3G family, as shown in Figure 1 (e), all network types experience less than 150 kbps median uplink throughput.

3.3 Long-term Trend and Diurnal Pattern

Given the relatively long duration of the data set, we analyze the trend of cellular network performance over time. We group users into each individual month and study the performance changes for the major U.S. cellular carriers, including AT&T, T-Mobile, Verizon, and Sprint. However, for all performance related metrics, including downlink/uplink RTT, RTT standard deviation, throughput and local DNS lookup time, we do not observe any clear trend of changes for 3G networks, suggesting that the performance is relatively stable. For example, the median DNS lookup time to local DNS servers across all carriers changes by no more than 10%. We do observe a few LTE, WiMAX users, a topic which we plan to explore in the near future.

Our previous work [11] showed using local experiments that performance of some carriers is correlated with time of day. In this study, we further analyze this pattern using the data set collected across locations. Figure 3 shows the aggregate time of day analysis of TCP downlink throughput, RTT, and jitter for all AT&T, T-Mobile, and Verizon 3G users in



(a) Downlink throughput (b) Downlink RTT (c) Downlink RTT standard deviation

Figure 3: Time of day pattern of downlink performance for major carriers in the U.S. (large packets)

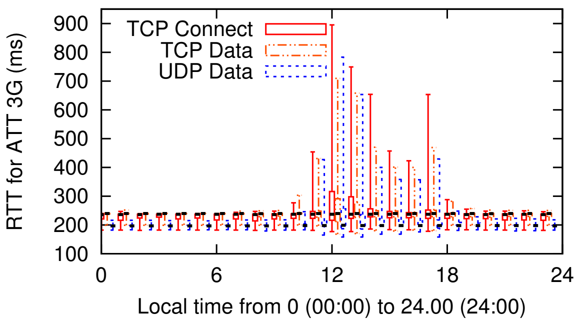


Figure 4: Delay in AT&T 3G network (small packets)

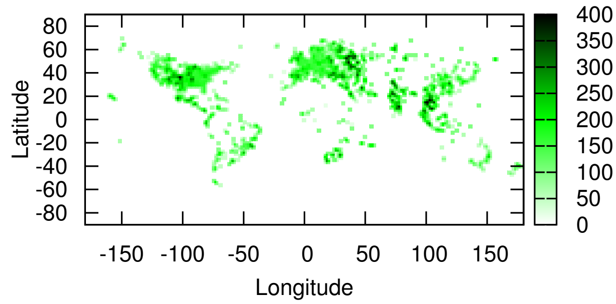


Figure 5: DNS lookup time in 3G networks (ms)

the U.S. Each data point is median value of all data samples across locations in the U.S. within an hour based on user's local time. This analysis technique avoid potential bias by a specific small group of users or any particular locations.

Figure 3 (a) shows that AT&T has the most clear time of day pattern for downlink throughput. Verizon has less obvious yet still observable diurnal pattern for downlink throughput and even less obvious for T-Mobile. TCP retransmission rate for these carriers stays low consistently at different hours, hence the diurnal pattern of TCP downlink RTT in Figure 3 (b) explains the throughput fluctuation, especially for AT&T. The standard deviation for TCP downlink RTT also demonstrates a clear diurnal pattern for AT&T and Verizon, while less obvious for T-Mobile shown in Figure 3 (c).

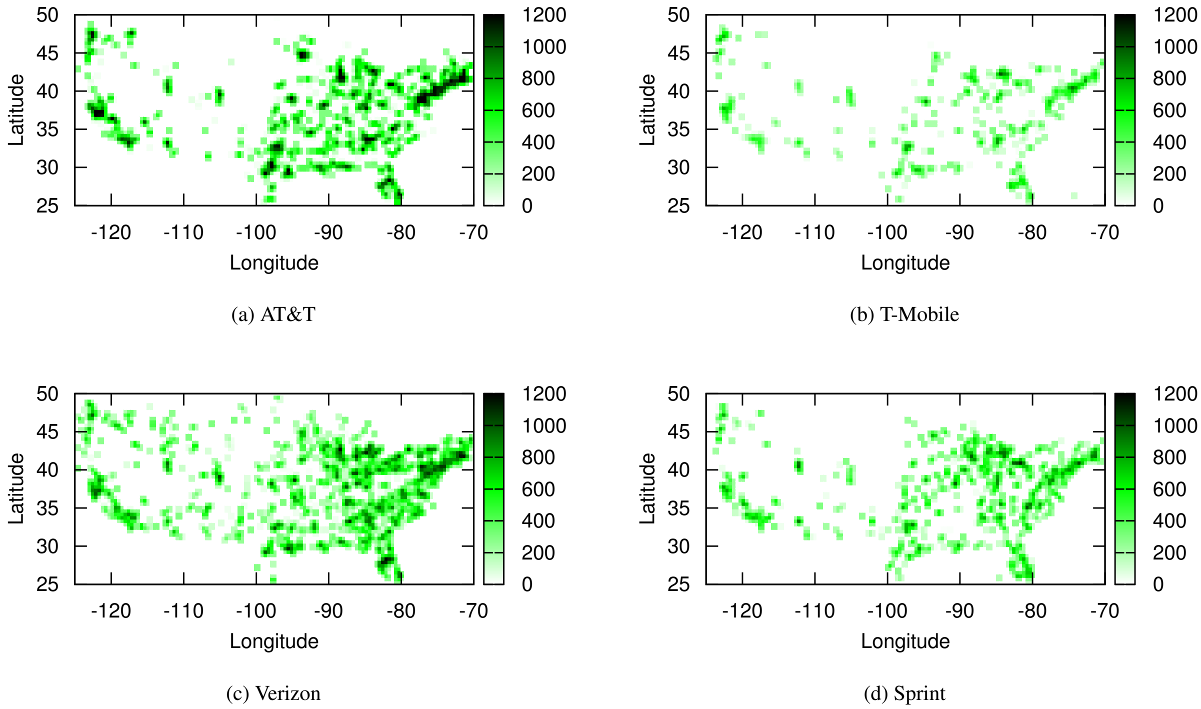
These observations suggest that applications with intensive network download requirement and little tolerance on RTT jitter may experience worse performance during peak hours. Uplink and LDNS performance are relatively consistent across different hours of day, indicating that the infrastructure support is capable of handling peak hours for such operations. However, for downlink, network resource at peak hours becomes a bottleneck.

Since the downlink RTT in Figure 3 (b) is for a large packet, mostly with a size of MTU, we use the latency data collected in the local *"Port Scanning"* experiment (§2) to understand the diurnal pattern of small packets. For T-Mobile, similar to Figure 3 (b), we do not observe any time of day effect on RTT. For AT&T, in Figure 4 showing RTT of small packets (100 bytes) with boxplot, at the median level, there is no diurnal pattern for RTT. At the 75^{th} percentile, 12:00PM and 1:00PM have larger RTT values, and at the 95^{th} percentile, hours between 10:00 AM and 6:00 PM clearly have much larger RTTs. This indicates that during peak hours, most small packets do not experience longer delay, but some (at least 5%) experience much longer RTTs. By comparing with large packets, our local experiment suggests that small packets have less obvious, yet still observable diurnal pattern for AT&T.

3.4 Location and Performance Correlation

For **MobiPerf**, our server is located inside the U.S., hence users outside the U.S. experience longer Internet delay and the performance measurement might be biased. So for the worldwide measurement, we only look at local DNS lookup time, which is not affected by the location of our central server. In Figure 5, for each cell with a size of 50 kilometers$\times$50 kilometers, the median DNS lookup time of all 3G users within this region is selected. We can see that for most parts of the world, the DNS lookup time is around 150 ms to 250 ms. Regions including South Asia, Middle East, Eastern Europe and some regions in the middle of the U.S. experience higher DNS delays. For those regions with large LDNS latency, websites which often use DNS based load-balancing using services such as Akamai may suffer longer delays due to relatively higher DNS overhead.

For cellular networks, given that the infrastructure support differs across locations for different carriers, we intend to



(a) AT&T

(b) T-Mobile

(c) Verizon

(d) Sprint

Figure 6: Downlink throughput of major carriers in the U.S. (kbps)

study how performance correlates with location. We compare TCP downlink performance of major U.S. carriers in Figure 6. The plotted cell size is 50 kilometers $\times$ 50 kilometers, excluding those without enough data points to show statistically meaningful results. The coverage of each carrier shown in this figure is clearly affected by the popularity of our app among its customers. Since we focus on the median performance at different locations, where we do have data, our measurements can provide a fairly representative sampled view of the cellular network performance across different locations. Comparing across carriers shows large variation in performance across locations, with a few locations clearly having better downlink performance. Second, these locations of different carriers are typically large cities, such as New York, San Francisco, and Los Angeles. Third, for each carrier, the eastern part of the U.S. has higher user coverage than the western part, except for the state of California and Washington. These observations suggest that for regions with larger user base, carriers may have provided better infrastructure support and newer technology, compared with rural areas with fewer users.

Given the large scale of our dataset across locations, as future work, we will provide a "performance near me" service to help users understand network problems from nearby users in the same or different networks.

3.5 Case Study on RTT vs. Packet Size

With the local "RTT vs. Packet Size" experiment (§2), we study the correlation between RTT and packet size using both TCP and UDP for AT&T and T-Mobile. For T-Mobile, RTT for TCP and UDP remains stable with packet size. Figure 7 shows for AT&T how the packet size including IP and TCP/UDP headers affects the median RTT from more than 1000 measurements. We make the following observations.

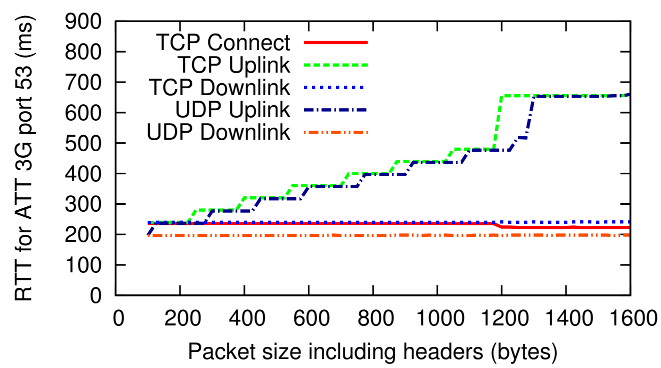


Figure 7: RTT vs. packet size for AT&T 3G

First, similar to T-Mobile, UDP/TCP downlink RTT remains stable as packet size changes. Second, TCP downlink RTT for AT&T is consistently about 40 ms larger than that of UDP downlink, indicating a slightly higher overhead for TCP traffic than UDP, unlike T-Mobile with all very similar RTT values. Third, AT&T's uplink RTT is a step function of packet size for both TCP and UDP, *i.e.,* RTT increases by about 40 ms with every packet size increase by roughly 150 bytes until packet size exceeds 1200 bytes triggering a jump in RTT of 180 ms. Laner *et al.* [13] shows that for HSUPA, RTT increases by 10 ms for every packet size increase of 1800 bytes, due to lower layer fragmentation. For our experiments, the delay is longer since HSUPA is not available on our device. A similar step-based increase has been observed in [6] and our result differs from theirs in that we do not observe increased delay for packets smaller than 200 bytes, and we observe a bigger jump for packet size exceeding 1200 bytes. Another interesting observation for AT&T is that UDP trails behind TCP's jumps by about 50 bytes for uplink. One possible explanation of this offset is the difference in header compression between TCP and UDP.

Given that T-Mobile does not exhibit such behavior based on the same experimental setup, this implies that neither the device nor the server is accountable for it. Experiments on another device using the same SIM card but only AT&T's EDGE network show observations consistent with those in 3G, though with larger variations. So this phenomenon should be related with AT&T network's infrastructure and configuration. If this observation is consistent across locations, users would experience much longer delay when uploading contents using AT&T's 3G service.

4. CELLULAR NETWORK INFRASTRUCTURE AND IMPLICATIONS

In this section, we study LDNS assignment and quantify the effectiveness of CDN service in cellular networks.

4.1 LDNS Servers

How LDNS servers are assigned to customers is an interesting and important design decision made by carriers. It reflects how loading balancing is done and affects LDNS lookup performance. It also impacts the effectiveness of DNS-based server selection. With a representative data set for the major American carriers, we study how they assign LDNS servers to their customers. We observe 12 different LDNS IPs for AT&T in 4 different /24 address blocks, each consisting of 3 IPs. For T-Mobile, 11 LDNS IPs are observed in 5 /24 blocks, and for Verizon, 12 LDNS IPs in 3 /24 blocks are detected. In Figure 8, we present the correlation between users' location and the assigned LDNS IPs. For both AT&T and Verizon, we can observe that clients for each LDNS address block tend to cluster geographically. This suggests that both carriers assign LDNS servers to clients based on geographic regions. However, for T-Mobile, all identified LDNS IPs are used across the country. These results confirm the unsupervised clustering results of our previous work [19].

To further understand how different individual IPs among a /24 address block are used, we show in Figure 8 the geographic distributions of four representative LDNS IP addresses. We can observe that even at the individual IP level, there exists clear correlation between location and LDNS assignment, despite some overlapping regions. Through local experiments for AT&T, we confirm that the assigned LDNS IP remain constant for over 48 hours. Having one LDNS IP for a region allows the flexibility to customize DNS caches according to the users' interests within the region. The comparison of the LDNS latency distribution for AT&T and T-Mobile shows no clear performance difference when querying LDNS servers at different locations. As we discuss further below in §4.2, we conjecture that the bottleneck of network latency is along the wireless hop, making the server location less important. Note that DNS-based server selection cannot effectively choose nearby servers if the LDNS server assignment is not based on geographic regions.

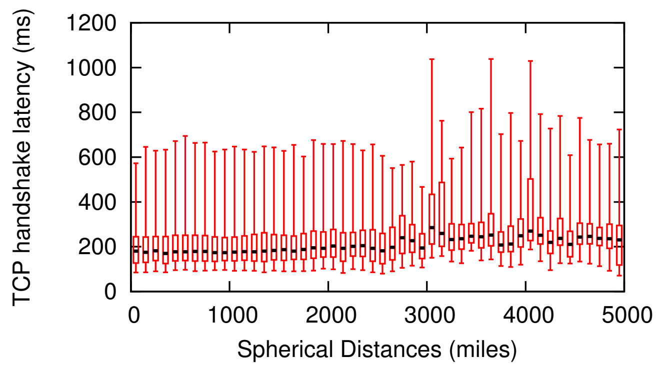


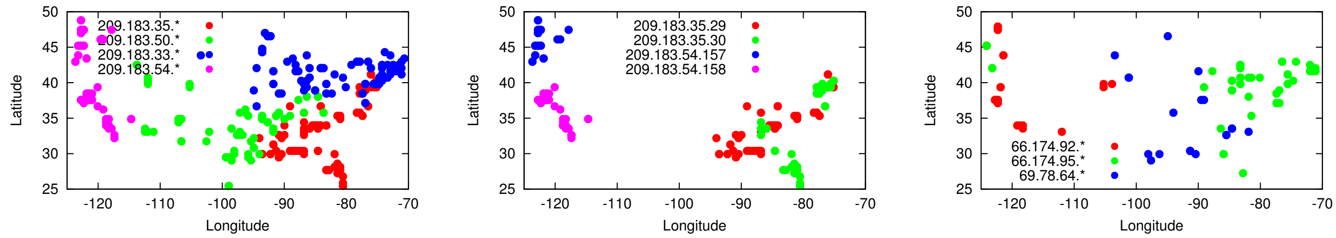
Figure 9: Handshake RTT vs. spherical distances

4.2 CDN Service for Cellular Networks

Using landmark test results, we study the effectiveness of CDN service in cellular networks. First, we select the users with GPS information and calculate the physical spherical distance [2] to the list of 20 landmark servers. The distribution of TCP handshake latency is show in Figure 9. We observe high variation in RTT between users and landmark servers with no clear correlation with physical distance. This suggests that the latency in the 3G wireless hop is dominant in determining the end-to-end latency, rather than the Internet latency.

To quantify the effectiveness of CDN services in cellular networks, we assume the 20 landmark servers to be CDN servers. Then we study 5 different scenarios, assuming 20, 10, 5, 2 and 1 CDN servers are used, respectively. We select the CDN servers to maximize the geographic spread. For each scenario, each user chooses the physically closest CDN server based on its GPS information, and the latency to this server is regarded as the user perceived latency. Figure 10 shows the distribution of the user perceived latency in all studied scenarios. Additional CDN servers have very limited effect on reducing user perceived latency. Comparing the best and worst scenario with 20 and 1 CDN servers each, we found the median RTT difference to be 20 ms, only 10% of the median RTT of 200 ms. However, the equivalent RTT

[2]Spherical is the accurate distance in the great circle of the earth based on latitude and longitude.



(a) Coverage of all AT&T's DNS servers (b) Coverage of 4 sampled DNS IPs in AT&T (c) Coverage of Verizon's DNS servers

Figure 8: Coverage of local DNS servers

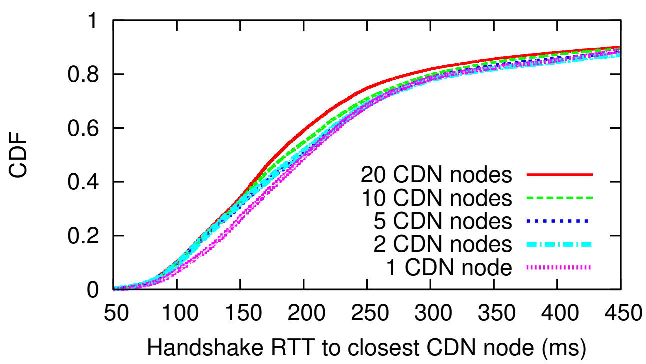


Figure 10: RTT to closest synthetic CDN node

saving can be more significant in LTE 4G networks with a much shorter latency at the wireless hop. Given that today's cellular network routing is heavily restricted due to limited number of gateways to the Internet which are the closest locations to the clients for deploying CDN servers, caches can be pushed closer to the user to reduce latency in LTE 4G network, which uses a flatter architecture [16].

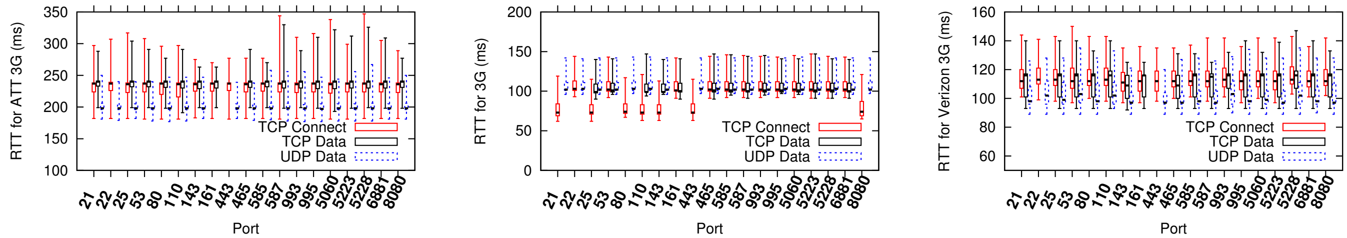
5. CELLULAR NETWORK POLICY

In this section, we study cellular network policies towards different types of traffic. Compared with our previous work [18] on analyzing NAT and firewall policies in cellular networks, this study focuses on performance related policies exposing traffic differentiation behavior. Figure 11 summarizes local "Port Scanning" (§2) experiment for AT&T, T-Mobile, and Verizon 3G networks using boxplot. "TCP Connect" represents the time between the sending of a TCP SYN packet and the receipt of SYN-ACK at the client. "TCP Data" is the time for the client to send a short unique message to the server and receive the echoed response. "UDP Data" is the corresponding data transfer time for UDP. Verizon and AT&T share similar behavior in Figures 11 (a) and (c). There is no obvious difference across all ports. For T-Mobile in Figure 11 (b), first, there are two levels of TCP connect time, one around 70 ms and the other around 100 ms. Second, port 22 (FTP) and port 8080 (HTTP proxy) are blocked. For the other ports, their median TCP and UDP data transfer time are both around 100 ms.

To understand the root cause of these abnormal behaviors in T-Mobile, we collect packet traces at both the client and the server side. All packets sent from the server have an initial TTL value of 64. But at the client side, it receives packets from server's IP with a TTL of 197 for ports 25 (SMTP), 80 (HTTP), 110 (POP), 143 (IMAP), 443 (HTTPS), and 8080 (HTTP proxy). For port 21 (FTP), the TTL in the packets received at the client side is 253, also larger than the server's initial TTL value. These ports are also exactly the ones with smaller TCP connect time of around 70 ms. All these observations support the existence of a middlebox that rewrites packets going through.

We analyze packets traces to understand the detailed blocking behavior of T-Mobile. While nothing abnormal is observed for UDP traffic on port 21, TCP traffic on port 21 is blocked. The client can successfully establish a TCP connection to our server via three-way handshake. However, after the client sends a packet containing data payload to the server, the client receives an ACK packet with the server's IP as the source IP, which however, did not originate from the server, verified by the packet traces collected at the server. The server also did not receive any data packet from the client after the three-way handshake. We further verify that the ACK comes from a middlebox based on TTL. The spoofed ACK has a TTL of 253 (larger than the server's initial TTL value of 64), indicating that the actual sender of this packet is at most 2 hops away. So there must be a middlebox intercepting data packets and sending spoofed ACK to the client without letting data packets through. For comparison, we use a standard FTP client on Android and it works in T-Mobile's 3G network, suggesting that the middlebox does deep packet inspection (DPI) and selectively blocks our measurement traffic which does not contain a valid FTP command, possibly due to security concerns.

The HTTP proxy port 8080 is also blocked in T-Mobile's 3G network. Different from port 21, the server does not receive any packet from the client throughout the flow. But interestingly, the client still receives the SYN-ACK packet. Subsequently, when the client sends data packets to the server, it can also receive spoofed ACK packets, making it believe that the server has received the data, which is actually not the case. Also, packets from the spoofed server's IP



(a) AT&T 3G (b) T-Mobile 3G (c) Verizon 3G

* Among all ports, only UDP port 161 (SNMP) is blocked by the firewall of our server.

Figure 11: RTT for different ports in 3G cellular networks

address has a TTL of 197. To verify whether our traffic is blocked due to invalid payload in port 8080, we use the web browser on an Android device to visit a website hosted on port 8080. Though the website is accessible through WiFi, it is blocked in T-Mobile's 3G network, suggesting that port 8080 is completely blocked.

Using packet traces in **`MobiPerf`** data set collected from global users at the server side, we compare the TTL of packets from the client to the server. Given that for each run, within a short period, the TTL of packets from the same client across different ports should not be significantly different from each other, since each device has a fixed initial TTL for its outgoing packets and the routing paths across different ports are comparable in length. However, we observe that for all T-Mobile users in the U.S., the TTL for port 80 is larger than TTL for other ports by at least 75. The gap of TTL is 117 for majority of T-Mobile users. This agrees with our local experiments, suggesting that T-Mobile uses middlebox for some ports including port 80. We do not observe this phenomenon for most users of AT&T, Verizon or Sprint in the U.S. However, this does not necessarily mean that middlebox is not used by these carriers, because middlebox does not necessarily modify TTL. Besides the U.S., we also observe the big TTL gap phenomenon for users in other regions including Australia and Singapore.

The reasons of using middlebox can be multifold. Carriers may use middlebox to increase end user performance and to block suspicious traffic for security in cellular networks. As a case study, we download a text file hosted on port 80 containing a long string of character "a"s in WiFi and T-Mobile 3G, respectively. Although the file is downloaded as plaintext through WiFi, gzip compression is used in 3G, indicating that the middlebox compresses web content for 3G users. However, some carriers may also use middlebox to intercept, monitor, and modify traffic and differentiate specific types of traffic or groups of users.

6. RELATED WORK

Existing studies have compared 3G and WiFi performance on smartphones [10] and studied the influence of the packet size on the delay in 3G networks [6]. In a previous study [7], the correlation among IP address, location and network latency has been analyzed for smartphones. Our previous study [11] has compared cellular network performance among carriers and shown indications that the network can be the bottleneck accounting for poor application performance. In this work, with data set collected for a period of 18 months and a large set of 10,000 unique users, we study cellular network performance along several important new dimensions, including network types, location, time, *etc.* This is the first work studying cellular networks with a data set covering such a long period.

There are existing measurement applications such as Speedtest.net [4] and FCC's broadband test [2] measuring throughput on end-user's devices, similar to our study. However, our study is more comprehensive looking at various network performance metrics including DNS lookup time and TCP retransmission rate. Netalyzr [3] carries out network measurement for Internet users, not for smartphone users.

In our previous work [19], we made qualitative observations on the effectiveness of CDN service in cellular networks and the geographical coverage of LDNS servers. In this study, with a comprehensive end-to-end latency measurement data set, we quantify the effectiveness of adopting different CDN servers for cellular networks. We also provide fine-grained analysis of the geographical coverage of each individual IP address of LDNS servers and discuss the implication on performance and reliability.

Existing work has built tools to infer traffic differentiation policies from either local experiments or public deployment [20, 9, 5]. And our work is among the first to uncover previously unknown policies in cellular networks. Our work is among the first attempts to systematically understand the network policy inside cellular networks.

Our work is inspired by numerous network measurement studies [8, 21, 17, 7], *e.g.,* Trestian *et al.* characterized the relationship between users' application interests and mobility [17], Balakrishnan *et al.* examined the dynamics of cellular IP addresses in 3G networks [7], Zhuang *et al.* investigated application-aware acceleration to improve application performance [21], and Liu *et al.* studied the interaction between the wireless channels and applications [14]. Unlike these studies, we analyze the network performance along diverse dimensions, *e.g.,* continents, geographic locations, carriers, platforms, cellular technology, and time.

Previous studies [14, 8, 12] use cellular network data cards or phones tethered through USB to measure network performance on desktop or laptop systems. In this study, the data is collected directly from end-users' devices, thus more accurately reflecting the real perceived performance and allowing us to study location-wise performance differences.

7. CONCLUSION

In this study, we analyze the data collected from our globally deployed network measurement tool **MobiPerf** over 18 months and local experiments. We examine various factors, such as locations, time, network types by investigating how each factor impacts the user-perceived performance. We also characterize the cellular network infrastructure to understand how LDNS servers are assigned and quantify the effectiveness of deploying CDN servers for today's cellular networks. Given the lack of openness in cellular networks, we further explore the traffic policy of cellular carriers and have successfully detected different middlebox behaviors. We believe that our study can help uncover the key characteristics of cellular networks in various dimensions.

8. REFERENCES

[1] AT&T Admits To Network Overload. `http://thedroidguy.com/2011/04/att-admits-to-network-overload/`.

[2] FCC Consumer Broadband Test. `http://broadband.gov/qualitytest/`.

[3] ICSI Netalyzr. `http://netalyzr.icsi.berkeley.edu/`.

[4] Speedtest.net. `http://www.speedtest.net/`.

[5] WindRider: A Mobile Network Neutrality Monitoring System. `http://www.cs.northwestern.edu/~ict992/mobile.htm`.

[6] P. Arlos and M. Fiedler. Influence of the Packet Size on the One-Way Delay in 3G Networks. In *Proceedings of PAM*, 2010.

[7] M. Balakrishnan, I. Mohomed, and V. Ramasubramanian. Where's That Phone?: Geolocating IP Addresses on 3G Networks. In *Proceedings of IMC*, 2009.

[8] R. Chakravorty, S. Banerjee, P. Rodriguez, J. Chesterfield, and I. Pratt. Performance Optimizations for Wireless Wide-Area Networks: Comparative Study and Experimental Evaluation. In *Proceedings of ACM MOBICOM*, 2004.

[9] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu. Glasnost: Enabling End Users to Detect Trafc Differentiation. In *NSDI*, 2010.

[10] R. Gass and C. Diot. An Experimental Performance Comparison of 3G and WiFi. In *Proceedings of PAM*, 2010.

[11] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. Anatomizing Application Performance Differences on Smartphones. In *Proceedings of Mobisys*, 2010.

[12] K. Jang, M. Han, S. Cho, H.-K. Ryu, J. Lee, Y. Lee, and S. B. Moon. 3G and 3.5G Wireless Network Performance Measured from Moving Cars and High-speed Trains. In *Proc. of ACM MICNET*, 2009.

[13] M. Laner, P. Svoboda, E. Hasenleithner, and M. Rupp. Dissecting 3G Uplink Delay by Measuring in an Operational HSPA Network. In *PAM*, 2011.

[14] 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 ACM MOBICOM*, 2008.

[15] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Characterizing Radio Resource Allocation for 3G Networks. In *IMC*, 2010.

[16] S. Sesia, I. Toufik, and M. Baker. LTE: The UMTS Long Term Evolution From Theory to Practice. John Wiley and Sons, Inc., 2009.

[17] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Measuring Serendipity: Connecting People, Locations and Interests in a Mobile 3G Network. In *Proceedings of IMC*, 2009.

[18] Z. Wang, Z. Qian, Q. Xu, Z. M. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *SIGCOMM*, 2011.

[19] Q. Xu, J. Huang, Z. Wang, F. Qian, A. Gerber, and Z. M. Mao. Cellular Data Network Infrastructure Characterization and Implication on Mobile Content Placement. In *SIGMETRICS*, 2011.

[20] Y. Zhang, Z. M. Mao, and M. Zhang. Detecting Trafc Differentiation in Backbone ISPs with NetPolice. In *IMC*, 2009.

[21] Z. Zhuang, T.-Y. Chang, R. Sivakumar, and A. Velayutham. A3: Application-Aware Acceleration for Wireless Data Networks. In *Proc. of ACM MOBICOM*, 2006.