

Anatomizing Application Performance Differences on Smartphones

Junxian Huang
University of Michigan
hjax@eecs.umich.edu

Z. Morley Mao
University of Michigan
zmao@eecs.umich.edu

Qiang Xu
University of Michigan
qiangxu@eecs.umich.edu

Ming Zhang
Microsoft Research
mzh@microsoft.com

Birjodh Tiwana
University of Michigan
tiwana@eecs.umich.edu

Paramvir Bahl
Microsoft Research
bahl@microsoft.com

ABSTRACT

The use of cellular data networks is increasingly popular due to the widespread deployment of 3G technologies and the rapid adoption of smartphones, such as *iPhone* and *GPhone*. Besides email and web browsing, a variety of network applications are now available, rendering smartphones potentially useful substitutes for their desktop counterparts. Nevertheless, the performance of smartphone applications in the wild is still poorly understood due to a lack of systematic measurement methodology.

We identify and study important factors that impact user-perceived performance of network applications on smartphones. We develop a systematic methodology for comparing this performance along several key dimensions such as carrier networks, device capabilities, and server configurations. To ensure a fair and representative comparison, we conduct controlled experiments, informed by data collected through *3GTest*, a cross-platform measurement tool we designed, executed by more than 30,000 users from all over the world. Our work is an essential step towards understanding the performance of smartphone applications from the perspective of users, application developers, cellular network operators, and smartphone vendors. Our analysis culminates with a set of recommendations that can lead to better application design and infrastructure support for smartphone users.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communication; C.4 [Performance of Systems]: Performance attributes; C.4 [Performance of Systems]: Measurement techniques; D.2.8 [Metrics]: Performance measures

General Terms

Experimentation, Measurement, Performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiSys'10, June 15–18, 2010, San Francisco, California, USA.
Copyright 2010 ACM 978-1-60558-985-5/10/06 ...\$10.00.

Keywords

Smartphone, 3G network, Application performance, Performance comparison, Cellular data network performance

1. INTRODUCTION

As of the third quarter of 2009, global smartphone shipments reached 41.4 million units representing about 15% of the total mobile phone market [3]. It is expected that in the next few years smartphone sales will catch up with the sales of regular phones. Vendors, such as Research in Motion, Samsung, Palm, HTC, and Apple, offer a variety of smartphones equipped with increasingly faster CPUs and larger memory, though still lagging behind desktop or laptop systems. With access to various high-speed 3G networks, such as EVDO and UMTS, they are powerful enough to run modern operating systems and sophisticated network applications such as web browsing, email, and streaming media.

Unlike traditional Internet-based applications, whose performance is mostly constrained by the wired network, network application performance on smartphones with limited physical resources also heavily depends on factors including hardware and software on the phone as well as the quality and load of wireless link. Understanding the application performance on smartphones is important for the purpose of assisting consumers in choosing carriers and phones and guiding application developers in designing intelligent software. Moreover, cellular network operators and smartphone hardware and software vendors can use this knowledge to optimize networks and phones for better end-user experiences. Similarly, content providers can leverage this knowledge to better customize content for mobile users. However, this task is quite challenging since the performance of network applications on smartphones is poorly understood thus far, due to a lack of a systematic approach for controlled experiments and comparative analysis. We believe this work fills this gap.

We focus on developing systematic methodology for measuring and analyzing 3G network performance as well as smartphone application performance. We make it relevant to end users by studying real applications directly on the phone platforms. Our approach differs inherently from most previous work of using laptops equipped with 3G data cards in three ways: (1) We measure the performance of applications rather than that of the low-level protocols. Prior work has shown that application performance often significantly deviates from protocol performance [24]. We target the pervasive web browsing, streaming video, and VoIP applications that most end-users care about; (2) We measure application performance on several common mobile devices. Application performance varies widely across devices due to differences in hardware

and software, necessitating direct experimentation on smartphones instead of on laptops with wireless cards; (3) We study the application performance under real-world scenarios and quantify the performance of web browsing by evaluating commercial websites in addition to locally-constructed ones with replicated, real web content under our control. The latter setup helps dissect and analyze the individual factors that contribute to the overall web browsing performance.

To overcome the limitation of a single vantage point for locally conducted measurements, we design and deploy a cross-platform measurement tool, called *3GTest*, to measure network-level performance, using basic metrics such as throughput, round trip time (RTT), retransmission rate, *etc.* attracting more than 30,000 users all over the world, providing a representative data set on the current 3G network performance. *3GTest* enables us to carry out local experiments informed by realistic 3G network conditions across diverse locations and network carriers. As far as we know, *3GTest* is the first such cross-platform tool available that comprehensively characterizes 3G network performance, and our data set is also unique in that regard.

In addition to shedding light on the overall application and network performance, we perform detailed analysis to identify and isolate factors that impact user-perceived performance to help carriers, phone vendors, content providers, and application developers gain insight. For example, for carriers, we infer various network-level problems, *e.g.*, high latency or high loss rate, which they can directly take action on. For phone vendors, we identify performance bottlenecks on the devices or issues associated with the content. These issues can be resolved either independently or by cooperating with content providers. And for application developers, we evaluate factors such as the overhead of HTML rendering and Javascript execution given a particular software configuration.

We comprehensively study the 3G network and application performance for all four major U.S. wireless carriers including AT&T, Sprint, Verizon, and T-Mobile. We choose popular devices including iPhone, Android G2 from HTC, and Windows Mobile phones from Palm, HTC, and Samsung for carrying out experiments. Our results show that their performance varies significantly across network applications. In fact, even for the same network application such as web browsing, certain types of phones consistently outperform others due to the differences in factors such as downloading behavior, customized contents, and page rendering. The application performance also heavily depends on properties of carriers including DNS lookup, RTT, and loss rate.

We summarize our main observations from extensive experimentation:

1. The four carriers we studied demonstrate distinct characteristics in network performance in terms of throughput, RTT, retransmission rate, and time-of-day effect. For example, compared with T-Mobile and AT&T's median of TCP retransmission rate of 0%, Sprint and Verizon have a higher median value of 0.7%.
2. TCP throughput, RTT, and retransmission rate vary widely even for a single carrier in measurement taken at different times and locations, *e.g.*, downlink throughput ranges from 50 kbps to 4 Mbps for AT&T, with the median value of about 1 Mbps.
3. The wireless delay in the 3G network dominates the whole network path delay, *e.g.*, latency to the first pingable hop is around 200 ms, which is close to the end-to-end Ping latency to landmark servers distributed across the U.S.

4. Besides networks, devices heavily influence application performance. Given the same content and network condition, different devices exhibit vastly different webpage loading time, *e.g.*, the page loading time of Samsung SCHi760 is consistently twice that of iPhone.
5. Mobile devices can benefit from new content optimization techniques like the data URL scheme, *e.g.*, page loading time for GPhone can improve by 20% in our experiments, despite its already good performance compared to other devices.

The paper is organized as follows. §2 covers related work. In §3, we propose our methodology for experiments, followed by details of the experimental setup in §4. In §5, we present 3G network characterization and then focus on web performance in §6. We evaluate the performance of streaming video and voice over IP in §7, and conclude in §8.

2. RELATED WORK

Our work is inspired by the Netdiff system [17], which established a benchmark for comparing performance of different ISPs. In our research, we attempt to establish an equivalent benchmark for comparing network application performance on smartphones. Although some online comparisons are available, such as Speedtest.net [7] and FCC's broadband test [4], which measure throughput and latency in 3G networks, *3GTest* covers a more comprehensive set of metrics, including DNS lookup, ping to the first hop, TCP handshake, and HTTP request to landmark servers. Tan *et al.* carried out a similar measurement study on multiple commercial 3G networks [20]. However, their study is limited to one location (Hong Kong) and a few devices. Compared with their study, our work covers significantly more users from many different locations.

There have been several studies focusing on mobile users from the perspective of applications, such as [21] which characterized the relationship between users' application interests and mobility, and [9] which examined the possibility of geolocating IP address in 3G networks. Other related measurement works of cellular data networks include a study of the interaction between the wireless channels and applications [16], an investigation of application-aware acceleration to improve application performance [24], a performance study of multimedia streaming [12], and performance analysis of TCP/IP over 3G network with rate and delay variation [11]. Our work complements these works with different focus and methodology.

From the perspective of measurement methodology, our *3GTest* tool is among the earliest cross-platform 3G network performance characterization tools for smartphones covering a diverse set of performance metrics, even though the idea of taking measurements from voluntary users is not novel. For example, Netalyzr [5] is one such tool focusing on desktop environment. Unlike previous studies, *e.g.*, [16, 10, 14], which perform measurements on desktop or laptop systems, relying on cellular network data cards or phones tethered through USB as a modem, we evaluate application performance directly using phones as the measurement platform, thus more accurately reflecting the actual user experience.

Compared to many previous works, we do not rely on proprietary data from carriers and mostly take a black-box measurement approach by examining performance at the application and network layers without accessing detailed information of wireless channels (*e.g.*, [16]) or the internal state of cellular data networks (*e.g.*, [23]). This therefore presents an interesting challenge of inferring the bottleneck of observed application performance. We argue that the

limitation of reduced visibility into the inner working of the 3G network does not prevent us from achieving the goal of effectively comparing network and application performance across different 3G carriers and understanding the effect of major factors.

Our work is built upon numerous previous TCP studies for cellular data networks which aim to understand the behavior of TCP using cross-layer measurement techniques [18], to model multi-rate and multi-user behavior [13], and to improve transport layer for wireless wide-area networks [22]. These studies expose the limitations of existing TCP designs, some of which are also confirmed by our work.

3. METHODOLOGY

In this section, we present our methodology for measuring network and application performance over 3G networks. Inspired by previous work in the Internet, *e.g.*, Netylizr [5], which collects measurement data from volunteers, we develop a cross-platform measurement tool used by tens of thousands of users on their smartphones to build a comprehensive data set for cellular networks. By analyzing the performance of web, video, and VoIP applications, we examine the effects of various factors on the overall application performance.

Unlike most previous works, we directly measure application performance on devices that consumers really use with 3G service provided by four major cellular carriers in the U.S., this helps us understand the client side factors and their impact on application performance. The novelty of our measurement methodology stems from our approach of approximately replicating the 3G network condition for controlled experiments using WiFi to enable reproducibility, and isolating the impact of each factor. These techniques are non-trivial given the complexity of mobile devices and network environment, and essential for eliminating interaction across factors.

3.1 Measuring network performance

We first describe the metrics we use for evaluating network performance and how we compute them.

3.1.1 Metrics

To characterize network performance, we use TCP throughput, downlink RTT, retransmission rate, local DNS lookup time, TCP handshake time, and Ping latency to the first responsive IP hop as our metrics. TCP is of particular interest, since most network applications use TCP. An application session usually requires DNS lookup, and every TCP connection begins with a TCP handshake. Hence these two factors contribute heavily to user-perceived performance of many network applications. Ping latency to the first responsive hop provides an estimate of the latency of the wireless hop.

3.1.2 3GTest

We developed a cross-platform tool named *3GTest* for measuring cellular network performance. It runs several experiments to collect measurements such as TCP throughput, DNS lookup delay, TCP handshake time, and Ping latency. In the TCP throughput experiment, data is transferred between the phone and a server connected to the Internet for a time duration. Packet traces are collected at the server side to calculate TCP downlink and uplink throughput, RTT, and retransmission rate.

For the DNS experiment, *3GTest* sends DNS requests to resolve a list of selected popular domain names. By tuning the size of the list and looking up each DNS name twice, we ensure the names are highly likely cached at the local DNS (LDNS) server in carrier

networks but not on the phone based on observed latencies. This is possible since compared to the phone the LDNS server typically has a larger DNS cache. To measure TCP handshake, *3GTest* sends TCP *connect* requests to several landmark servers sparsely distributed across the U.S. To characterize ping latency, our tool pings `www.google.com` with increasing TTL values starting from 1 and records the IP address and corresponding RTT. *3GTest* also pings the landmark servers to obtain the delay distribution to diverse Internet locations.

We have made *3GTest* (<http://www.eecs.umich.edu/3GTest/>) publicly available, which allows us to characterize 3G network performance in multiple cellular carriers at diverse locations over an extended duration.

3.1.3 Analysis methodology

We calculate RTT and TCP retransmission rate from server collected packet trace. We follow the standard method to infer RTTs from traces at the sender. At any given time, we pick one data packet and its corresponding ACK packet to compute one RTT sample. We take one sample per TCP window so that the average RTT will not be skewed by the window size. To prevent retransmission from inflating RTT estimation, we discard an RTT sample if there is any retransmission in the same window. We then compute the average RTT of all the samples of a TCP connection.

3.2 Measuring web browsing performance

Web browsing is one of the most popular smartphone applications. The process of visiting a webpage can be quite complex given the dynamic nature of the content often generated from Javascript, resulting in multiple concurrent TCP connections. Content can also be customized based on mobile device and carrier network.

Web browsing performance depends on various factors, *e.g.*, DNS lookup time, TCP handshake time, TCP transfer time, Javascript execution time, and content size. To study the effect of these factors, we carefully design controlled experiments to manipulate a single factor at a time while keeping others the same. We first describe the metrics used to evaluate web browsing performance, followed by the controlled experiments to measure these metrics.

3.2.1 Metrics

Page loading time: The time between the first DNS packet and the last data packet containing payload from the server during a page loading. It reflects the overall performance perceived by a user. Note that a browser needs to further parse and render a webpage after it is loaded. This additional processing may not be fully included in page loading time due to a lack of visibility of the browser internals. Nonetheless, this is still a key indicator of user-perceived performance when loading a webpage.

Javascript execution speed: Many webpages contain Javascripts, and hence Javascript execution speed has significant impact on page rendering time.

Page size: The total number of unique bytes downloaded. It can be used to compute *average throughput* and to detect content variation and customization. We found that in real web browsing, even the same URL can have different page sizes when accessed from different platforms. We cope with this effect by taking snapshots of URLs and replicate their content on our local web server.

Browser concurrency: Most modern browsers support concurrent TCP connections to a single web domain. The maximum number of concurrent TCP connections to a domain varies across dif-

ferent browsers. Usually, higher concurrency enables better bandwidth utilization which in turn leads to shorter page loading time.

DNS lookup time: A browser sometimes needs to look up the IP address of a domain name before establishing a TCP connection with the web server. Since the content of a webpage can be hosted in multiple domains, a browser may have to perform a DNS lookup for each domain.

TCP handshake time: Each TCP connection starts with a three-way handshake during which no data is transferred. More TCP handshakes for a single page loading often lead to larger page loading time.

TCP idle time & transfer time: Given a TCP connection, an *idle period* is defined to be a period of at least T second with no network activity. The remaining time periods within the connection are *transfer periods*. An idle period usually corresponds to the local processing delay or server processing delay. Given the limited CPU power and memory on smartphones, the TCP idle time is likely to be dominated by local processing delay, *e.g.*, between the receipt of a response and transmission of the next request, often caused by HTML rendering and JavaScript execution

3.2.2 Controlled experiments

We create a list of popular URLs from [1]. These websites are visited using smartphones via 3G networks. From the collected packet trace, we infer various metrics such as page loading time. To study the effect of each factor influencing the web browsing performance, we host static copies of these popular URLs on our local web server. The content is replicated to ensure that all the phones download the same content and all HTTP requests are sent to the local server. To control the network conditions, we uniformly use WiFi across all phones while varying one factor at a time. The WiFi link is lightly loaded and has stable throughput and RTT. To produce network conditions comparable to 3G, we artificially introduce delay and packet loss at our server. We study the impact of the following factors on web browsing performance:

Impact of network: To study the effect of network conditions on page loading time, we vary the RTT and loss rate on our server.

Impact of concurrency: To study the effect of concurrency, we control the maximum number of concurrent TCP connections to a web domain on the server side. Because a phone also limits the maximum number of concurrent connections per domain, we create a special webpage in which each web object is hosted in a unique domain on the same web server. This effectively allows us to bypass the concurrency limit imposed by the phones. Note that this is necessary as we do not have the permission to directly modify the concurrency limit on the phone.

Impact of compression: To study the tradeoff between network overhead and computation overhead, we configure our web server into two modes, one uses compression, while the other does not. We compare the page loading time under these two modes.

Impact of Javascript execution speed: To evaluate Javascript execution speed on different phones, we use a benchmark [8] consisting of 26 different Javascripts. The benchmark is hosted on our web server and accessed by phones via WiFi so that the downloading time is negligibly small. We measure the total execution time of these Javascripts.

Impact of data URL scheme: We also study the effect of the data URL scheme [15], a recently-proposed mobile webpage design techniques. We compare the time to load a webpage constructed using and without using the data URL scheme.

3.2.3 Analysis methodology

Now we describe how to analyze the traces collected from controlled experiments to compute the desired metrics. We calculate the page loading time of each URL as defined in §3.2.1 and the average page loading time of all the selected URLs. To measure Javascript execution time, we modify the Javascripts to display their execution time when their execution finishes. We use the *average concurrency* as a measure of browser concurrency. The average concurrency of a page loading is calculated by dividing the total duration of all the TCP connections by the page loading time.

For each TCP connection, TCP handshake time is calculated as the time between the first *SYN* and *SYN-ACK* packets. TCP idle time is measured by scanning the connection for durations of more than T seconds of no network activity. T should be larger than the maximum RTT values and we will discuss the choice of T in §4. TCP transfer time is the rest for the connection. We also calculate the response time of all the DNS lookups T_{dns} .

Since each web browsing session often consists of multiple concurrent TCP connections, to estimate the contribution of each factor to the overall performance, we logically serialize all DNS lookups and TCP connections. This is possible for mobile web browsing since no HTTP pipelining is observed on any phones. After serialization, we get a total time T_{total} which is the sum of each connection's duration. Assuming the actual page loading time is T_{total}^* , the normalized DNS lookup time T_{dns}^* is calculated as $T_{total}^* T_{dns} / T_{total}$. This metric shows the overall weight of DNS lookup in the actual page loading time. The normalized TCP handshake time, TCP idle time and TCP transfer time are calculated in a similar way.

4. EXPERIMENTAL SETUP

In this section, we introduce the platforms used for our experiments, the cross-platform measurement tool (*3GTest*) that we developed and widely deployed, and the actual conditions and parameters chosen for the experiments outlined in §3.

4.1 Platforms

Table 1 lists the devices used and carriers studied in this work. We studied four major carriers in the U.S., AT&T, Sprint, Verizon, and T-Mobile. They split between UMTS/HSPA (AT&T and T-Mobile) and EVDO (Sprint and Verizon). AT&T has the highest advertised downlink and uplink data rates. The actual data rates that a user can attain depend on many factors, such as signal strength, location, and background traffic. One of our goals is to understand how the actual data rates match the advertised ones and which factors have the biggest impact on actual data rates.

To measure user-perceived performance on smartphones, we conducted controlled experiments on five popular devices listed in Table 1. We also used a few desktop computers as web servers. These desktops are connected with high-speed ethernet so that they are unlikely to become the bottleneck. They have Intel Core2 Duo 2.26 GHz processors and 2 GB memory, running Ubuntu 9.10 and Firefox 3.5.

4.2 3GTest

To make our study representative across multiple locations and carriers, we developed a cross-platform measurement tool *3GTest* [6] for three mobile platforms: iPhone, Android and Windows Mobile. *3GTest* consists of several types of experiments essential to characterize network performance, including TCP throughput, DNS lookup, TCP handshake to landmark servers, ping to the first responsive hop, ping to landmark servers, *etc.* For

Referred to as	iPhone	Palm	Samsung	G2	HTC
Carrier	AT&T	Sprint	Verizon	T-Mobile	AT&T
Network	UMTS	EVDO	EVDO	UMTS	UMTS
Advertised Downlink(Mbps)	0.7-1.7	0.6-1.4	0.6-1.4	0.6-1.0	0.7-1.7
Advertised Uplink(Mbps)	0.5-1.2	0.35-0.5	0.5-0.8	0.3-0.7	0.5-1.2
Vendor	Apple	Palm	Samsung	HTC	HTC
Device	iPhone	Treo800w	SCHi760	Android G2	TyTnII
Memory (MB)	128	128	64	192	128
Processor (ARM)	1176	1136	920T	1136EJS	1136EJS
CPU frequency (MHz)	620	333	400	528	400
OS	iPhone OS 2.1	Windows Mobile 6.1	Windows Mobile 6.1	Android 1.6	Windows Mobile 6.1
Browser	Safari	IE	IE	Browser App	IE

Table 1: Device specifications and 3G network carriers

§	Figure	Description	Category
5.1	Figure 1(a)(b)(c)(d)	3GTest downlink, uplink performance	TCP (3GTest)
5.1	Figure 1(e)(f)(g)(h)	3GTest Ping, DNS, TCP handshake	Ping, DNS lookup, TCP handshake (3GTest)
5.2	Figure 2	3GTest user time pattern	User time pattern (3GTest)
5.2	Figure 3(a)(b)(c)	TCP downlink performance and time of day effect	TCP and user time pattern (3GTest-Local)
6.1	Figure 4(a)(b)	Page loading time vs. RTT/retransmission rate	TCP performance vs. web performance
6.2	Figure 4(c)	Impact of parallelism	Parallelism in concurrent TCP connections
6.3	Figure 4(d)	Impact of compression	Content compression
6.4	Figure 5	JavaScript Execution	Client capability
6.5	Figure 6	Data URL scheme	Content optimization
6.6	Figure 7	Time breakdown for simple vs. content-rich URLs	3G web browsing
7.1	Figure 8	Content size and video timeline	Streaming video
7.2	Figure 9	VoIP timeline	VoIP

Table 2: Summary of experimental results

TCP throughput experiment, *3GTest* conducts a length of 20 seconds of data transfer between the phone and our server. Duration of 20 seconds is chosen so that enough packets are transferred between phone and server and users will not suffer from long waiting time. We chose 80 domain names for DNS experiment so that DNS resolution results are not locally cached on the phones but are cached on LDNS server. For TCP handshake and Ping latency experiments, we chose 20 landmark servers from PlanetLab sparsely distributed across the U.S. Results for these experiments are sent back to our server before *3GTest* terminates.

We have been using *3GTest* to collect data for several months, during which tens of thousands of users from various countries have installed and run *3GTest*. The data set used in this paper was collected between August 27, 2009 to December 1, 2009, containing 68,908 runs of *3GTest* with 30,105 unique users in total. In this paper, we only analyzed the 3G network data collected inside the U.S. (~50% of the entire data set).

4.3 Network performance experimental setup

To measure the network performance over a long term, we created an internal version of *3GTest-Local* and installed it on the smartphones listed in Table 1. *3GTest-Local* is modified to record the signal strength on the Samsung and Palm phones. *3GTest-Local* continually conducts measurements every 10 minutes to collect one week’s data (excluding weekends) in Ann Arbor, MI. We make sure that the phones are placed at the same location with excellent signal strength during the entire measurement study. Since the data is

collected continually for a long period of time, it can be used for characterizing the time-of-day effect.

4.4 Web browsing experimental setup

For web browsing experiments, we picked a list of 20 popular and representative URLs including search engines, emails, online maps, social networking websites, *etc.* For most of the URLs, we used their mobile version. To facilitate repeated experiments, we wrote a program to invoke browser to visit each URL in turn with an interval of 120 seconds. Such interval is expected to be large enough to complete the page download. We used *Apache* 2.0 HTTP server for hosting the replicated websites. We collected packet traces on iPhone and GPhone using *tcpdump* and on Windows Mobile phones using *netlog*. We verified that the CPU utilization caused by trace collection is under 5%. All the experiments were repeated 10 times.

To introduce artificial delay and loss, we ran a user-level program on the server. This program intercepts all packets destined to a particular IP address and injects delay and random packet loss. We controlled loss rate values from 0% to 10% and RTT values from 0 ms to 800 ms. These values cover the entire range of loss rate and RTT values observed in 3G networks from our *3GTest* data set.

We controlled the maximum concurrent connections by configuring the *Apache* server with the help of the *mpm_prefork_module*. We constructed a webpage with 30 embedded objects in the main HTML page to infer the maximum number of concurrent connections allowed by a phone to one web domain. For concurrency

experiments, we used RTT of 400 ms and loss rate of 0%. They are the median values from the *3GTest* data set.

For the compression experiments, we used *SetOutputFilter* and *BrowserMatch* directives to specify whether compression is enabled for a specific type of browser. We fixed loss rate at 0% and varied RTT from 0 ms to 800 ms. Our goal is to understand whether compression is beneficial under different network conditions.

For the data URL experiment, we constructed a webpage with 20 images; 10 of them are 18KB and the rest 10 are 0.241KB. We created two versions of this webpage, one with links to download the images and the other with the images embedded in the webpage itself. We could not carry out this experiment for Windows Mobile phones since IE currently does not support the data URL scheme.

When analyzing the web browsing traces, we chose 1 second as the threshold to identify TCP idle time. In the *3GTest* data set, RTT is smaller than 1 second in 99% of the cases. Thus, if there are no network activities for 1 second or more, the phone should be busy with some local processing, e.g., rendering HTML pages or executing Javascripts.

4.5 Video streaming experimental setup

Streaming video is another popular application on smartphones. We measure streaming video performance by playing a 37:40-minute video on the phones using a Youtube application. From the collected packet trace, we calculate the downloading size of the video by adding up the payloads for all the packets from the server to phone while excluding the retransmitted packets.

4.6 VoIP experimental setup

We used Skype to study VoIP performance on smartphones given its popularity. The current version of Skype cannot be installed on iPhone OS 2.1 or G2. So we only ran Skype on the Samsung and Palm phones. In our experiments, we played a 3-minute music file from both the phone side and the server side. We collect packet trace at the phone side to calculate throughput. Given that different volume may lead to different data size, throughout the experiment, we keep the volume to be the same.

5. 3G NETWORK CHARACTERIZATION

To fully understand the underlying factors accounting for the observed performance of network applications, we first focus on characterizing the performance of current commercial 3G networks. Two data sets are used for this purpose, one from *3GTest* (Figure 1) and the other from *3GTest-Local* (Figure 3). To verify the representativeness of the results from *3GTest-Local*, we compare the data ranges of the metrics studied in both data sets. As expected, the data ranges from *3GTest-Local* are within those from *3GTest*.

5.1 Comparison across carriers

Figures 1(a) illustrates measured TCP downlink throughput. Given stable TCP throughput is roughly inversely proportional to RTT and to the square root of packet loss rate [19], we also analyze RTT and retransmission rate. In Figure 1(b), all carriers show comparable RTT distributions, with T-Mobile showing slightly larger RTT values and correspondingly lower downlink TCP throughput. Various reasons contribute to large RTT in 3G networks, e.g., queueing delays at the base station or other internal nodes, such as RNC, SGSN, and GGSN in UMTS networks. We study this in more details in §5.2. Large RTTs may also be due to packet loss recovered through link layer retransmission, which we do not have direct information about.

Figure 1(c) plots measured TCP uplink throughput. Unlike downlink throughput, AT&T and T-Mobile have lower uplink

throughput compared with Sprint and Verizon. One of the reasons could be the lack of support for UMTS/HSUPA on the phones used for AT&T and T-Mobile. Even the latest version of iPhone 3GS does not claim to support HSUPA. The median uplink throughput for AT&T and T-Mobile ranges from 200 kbps to 300 kbps, while that for Sprint and Verizon is around 400 kbps.

Figure 1(d) shows that Verizon and Sprint exhibit slightly higher TCP retransmission rate, matching observations from our local experiments. On average, AT&T's downlink throughput outperforms that of the other carriers due to its relatively lower RTT and loss rate. The median of TCP downlink throughput for all carriers ranges from 500 kbps to 1 Mbps. Median RTT varies from 300 ms to 500 ms, suggesting 400 ms is a representative delay value to emulate 3G networks. AT&T and T-Mobile have a median retransmission rate of 0%, while that for Sprint and Verizon is 0.7%.

Figures 1(e) & (f) show that Ping latency to the first responsive hop is close to that to landmark servers, suggesting that the first responsive hop consisting of 3G wireless link contributes to most of the delay along the end-to-end network path. Note that Ping latency to the first responsive hop actually refers to the first IP hop responding to ICMP probing. For AT&T and T-Mobile, the first IP hop, when TTL is set to 1, does not respond in most cases. Only the second IP hop replies with a private IP address. For Sprint and Verizon, the first IP hop does reply with a public IP address. The median latency to the first responsive hop ranges from 150 ms to 200 ms, while that to landmark servers is between 180 ms and 250 ms. We observe that both the Ping latency and TCP handshake time are smaller than RTT values measured in TCP downlink experiments.

Figure 1(g) shows DNS lookup performance. We design the experiment in a way that all DNS lookups are cached at the LDNS server but not locally on the phone (§4.2). This allows us to more accurately estimate the delay to the LDNS servers. The LDNS servers studied tend not to respond to ICMP packets, making it challenging to directly measure the network delay between the phone and LDNS server. From the results, we found that all carriers exhibit similar trend with median values close to 200 ms. Given that the DNS lookup delay is already close to Ping latency to the first responsive hop, there is limited room for improving DNS lookup performance.

As shown in Figure 1(h), the median of TCP handshake delay ranges from 160 ms to 200 ms, close to the Ping latency to the first responsive hop in Figure 1(f). We also observe that the relative ranking among all carriers is consistent with that in Figure 1(f). Compared with Figure 1(b), large packets with size close to MTU (e.g., 1348 bytes in AT&T) are found to have 2 - 4 times RTT of small packets.

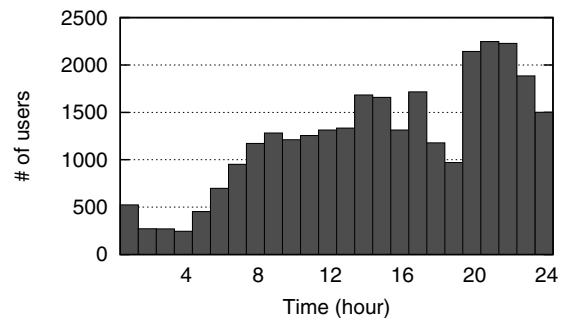


Figure 2: Number of *3GTest* users vs. time of day

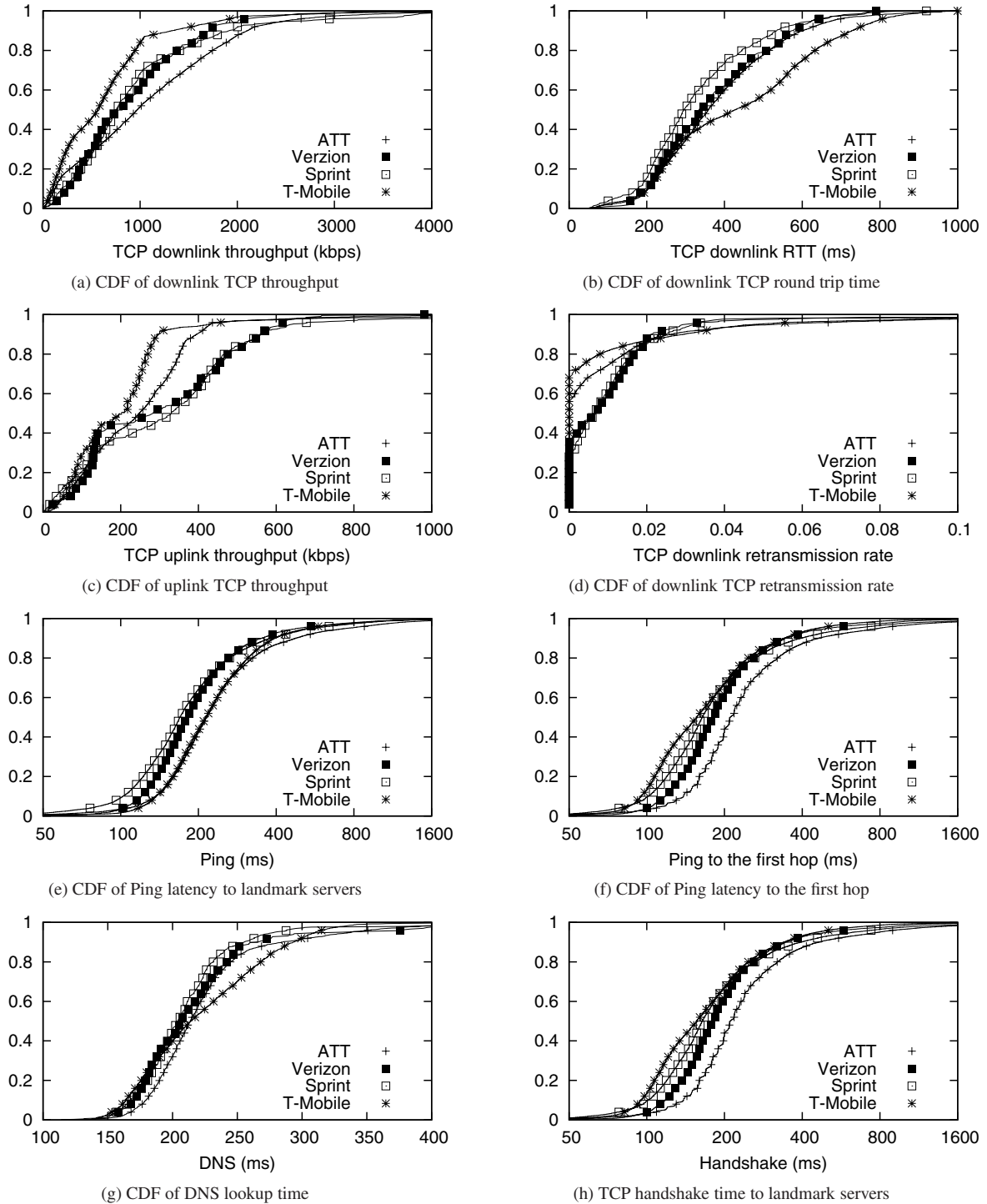


Figure 1: TCP performance comparison among carriers (data from deployed application *3GTest*, only U.S.)

5.2 Time of day correlation

Understanding whether traffic patterns exhibit any time of day behavior is useful for improving the design of applications and mobile network infrastructure. We expect smartphone users to have diurnal patterns in their behavior. For example, we can observe such a pattern in Figure 2. To further understand its impact on performance, we search for such a pattern in the data collected dur-

ing 5 contiguous weekdays as an initial investigation. The data set is from *3GTest-Local* described in §4.3 with results shown in Figure 3.

First, time of day effect is less pronounced for uplink throughput compared to downlink throughput, comparing Figure 3(a) and (d). This is likely due to higher demand for downlink capacity by popular applications such as web browsing and video streaming.

Second, we observe an obvious time pattern for AT&T's downlink throughput. At night and early morning hours, between 2 am and 8 am, the downlink throughput can reach 1 Mbps. However, the downlink throughput of lower than 500 kbps is observed at other times. This phenomenon is possibly due to the large number of iPhone users and the large traffic volume brought by various network applications. For Sprint and Verizon, we observe similar though less prominent trend compared to that for AT&T. For T-Mobile, the TCP downlink throughput is more stable, which we conjecture is due to the fact that its 3G service has only recently become available at our location.

Figures 3(b) & (c) indicate that RTT and retransmission rate exhibit time of day pattern for some carriers. For AT&T, the downlink throughput is found to be mostly affected by RTT values, likely to be caused by queueing delays in AT&T's 3G networks. RTT varies from 300 ms during late nights to as high as 700 ms at peak times. For Verizon and Sprint, the RTT values are more stable, though with varying TCP retransmission rate. One possible explanation is that in Verizon and Sprint's 3G networks, shared queues would drop packets once the queue length exceeds a threshold. This design will restrict the variation of RTT but incur more packet losses.

5.3 Signal strength effects

Signal strength is an important factor that affects 3G network performance, since higher signal-to-noise ratio (SNR) allows higher bit rate. We therefore also carried out experiments to understand this correlation. Since it is not easy for us to control the signal strength, we continuously monitor signal strength and TCP downlink throughput during a week. Due to space limitation, we only highlight our major observations here. When the signal strength is too weak, TCP connections will disconnect. When signal strength is at some middle range, we observe clear correlation between signal strength and TCP downlink throughput. TCP downlink throughput is not affected by the signal strength if the latter is above some threshold. Given these observations, we exclude the data points corresponding to poor signal strength from the *3GTest-Local* data set.

5.4 Smartphone vs. laptop

To understand whether the computation capability of a smartphone limits its 3G performance, we set up a controlled experiment to compare a smartphone (iPhone 3G) with a laptop (ThinkPad T42). The laptop can access AT&T's 3G network via a wireless data card, while the iPhone measurement is conducted at the same location and the same time. We found that the distribution of downlink throughput is similar, implying that the performance bottleneck is within the 3G network instead of on the phone. However, for other computation-intensive applications, the performance difference is more pronounced. We will study this in more details in §6.4.

Summary: The main observations of 3G network performance are:

1. Typical values for 3G throughput range from 500 kbps to 1 Mbps for downlink, and 200 kbps to 400 kbps for uplink, both lower than the advertised rates.
2. Network performance differs across all carriers. For downlink RTT and throughput, the differences among carriers are evident.
3. Sprint and Verizon have higher TCP retransmission rate compared with AT&T and T-Mobile.
4. Large packets can have 2-4 times RTT of small packets.

5. Some carriers show clear time of day pattern on weekdays, especially for AT&T's downlink throughput.
6. For simple TCP downloading/uploading, the performance bottleneck is within the 3G network.
7. 3G wireless delay dominates the end-to-end RTT.

Our observations suggest that the low uplink throughput and large RTT of current 3G networks raise challenges for offloading computation into the cloud. Network application designers should avoid chatty protocols and minimize total bytes to transfer. 3G operators need to examine their queueing and link layer retransmission policies to reduce latency in wireless links.

6. WEB BROWSING PERFORMANCE STUDY

Given the previous discussions on the performance of 3G networks, we now examine one of the most popular applications on smartphone, namely web browsing, in addition to two other popular mobile applications, streaming video and VoIP in §7. Note that many factors jointly determine user perceived performance, as an application may not fully utilize available network bandwidth due to limited processing power or memory on the phone [24].

Our study shows that the available 3G bandwidth is often not fully utilized for web browsing, and several modifications can be applied to current web browsers on smartphones to make better use of available network resources, *e.g.*, increasing the limit on concurrent TCP connections per domain, optimizing Javascript engines *etc.* We also evaluate the effectiveness of a few content optimization techniques, including compression and the recently-proposed data URL scheme [15].

In the following, we study the impact of network condition, browser concurrency, compression, and Javascript execution speed on web performance. We then break down the page loading time into several major components and identify the performance bottleneck for some mobile devices (§6.6).

6.1 Network effects on web browsing

To understand how network condition affects web browsing, we fix the web content, server configurations, browser concurrency and only vary the network condition. We emulate the 3G network condition by injecting packet delay and loss on the WiFi network path as described in §3.

Figure 4(a) shows page downloading time increases linearly with the RTT between smartphone and local web server. The downloading time is computed by averaging across 20 replicated URLs, with each visited 3 times. This is expected as throughput is inversely proportional to RTT. No additional packet loss is introduced since previous section shows packet losses are rare in 3G networks. The base RTT in our WiFi network is between 30 ms and 50 ms. The x-axis in Figure 4(a) shows the actual RTT after injecting extra delay. We can observe that under the same network condition, downloading time varies across phones though the relative ranking remains consistent. Note that web browsing cannot fully utilize the available bandwidth, due to page execution and rendering on the phone.

Figure 4(b) shows the effect of varying downlink packet loss rate for a fixed RTT value of 400 ms. Again the ranking in downloading time across phones is consistent. For small packet loss rate, *e.g.*, 2%, there is little performance degradation. However, with 10% loss rate, the page downloading time increases up to 35 seconds. In summary, smartphone web browsing performance heavily depends on network delay and loss conditions.

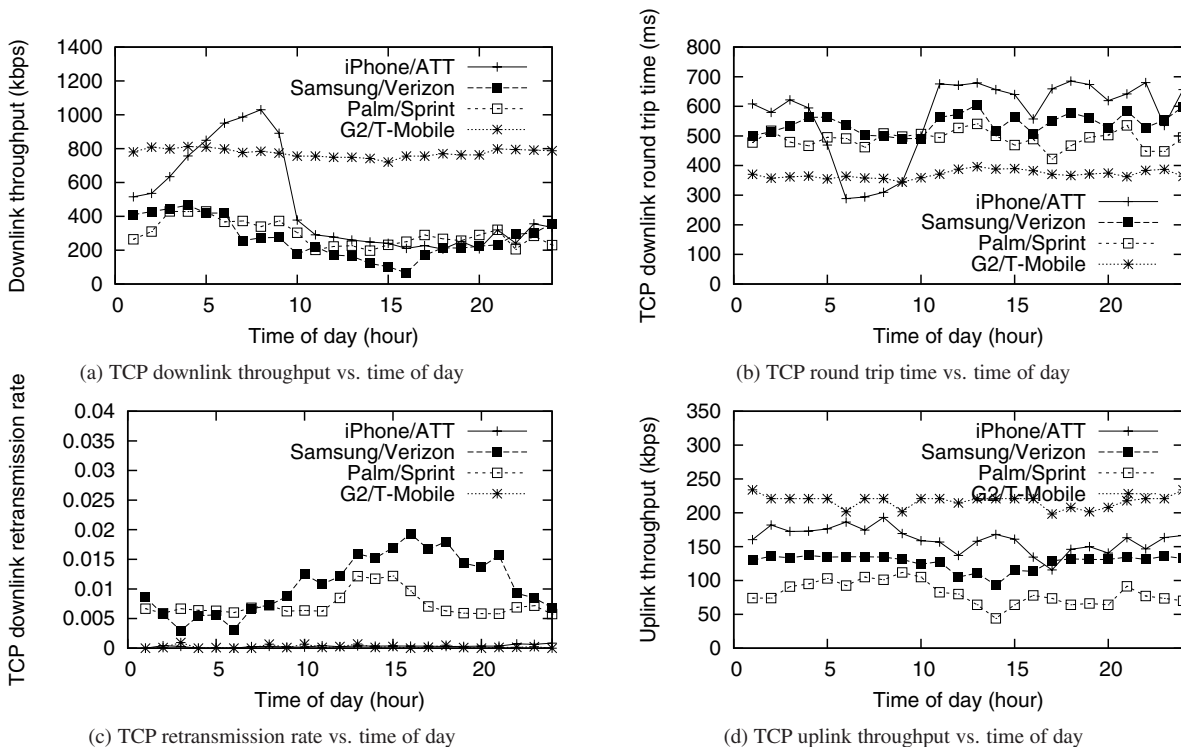


Figure 3: Correlation between TCP performance and time of day (3GTest-Local)

6.2 Concurrent TCP connections

3G network’s downlink throughput as measured normally ranges from 500 kbps to 1 Mbps for the carriers we studied (Figure 1(a)). We used the phones to visit the chosen URLs and found that the average throughput is only between 20 kbps and 50 kbps, indicating that more concurrent TCP connections can potentially improve web browsing performance.

Current web browsers on smartphones already allow concurrent connections. In browser’s settings, there is a parameter specifying the maximum number of concurrent TCP connections per domain. On Windows Mobile phones, it is a registry value named *MaxConnectionsPerServer* with a default value of 4. When we set the value to be smaller than 4, we observe decreased concurrency. However, when we increase the value to be larger than 4, the concurrency does not increase accordingly. This implies there exists another setting on maximum allowed concurrency per domain, which we cannot configure. For iPhone and GPhone, we are unable to set this parameter either. We design controlled experiments to measure the default concurrency setting on different platforms and found it to be 4 for all the phones studied.

We also found that no HTTP pipelining support is present on these platforms. Web objects are fetched sequentially within a persistent connection, and browser will not send a new HTTP request before data transfer of the previous request completes. We analyzed the 20 popular URLs and found that there are 10.8 images embedded in each page on average, along with several other types of embedded objects, such as Javascript, CSS files, etc. Those websites which do not have a mobile version, tend to have even more objects.

To understand how concurrency affects web browsing performance, we devised a set of experiments, with results shown in Figure 4(c). We first vary the maximum concurrent connections allowed at the server side from 1 to 4. We observe a significant

performance degradation across all platforms with more restricted concurrency. Under the restriction of a single connection per domain, web browsing is 3.5 to 4.5 times slower compared to that under the default setting. This indicates that today’s mobile browsers already benefit much from concurrency.

To understand whether further increasing concurrency will improve performance, we take a clever approach of DNS aliasing (§3.2.2) to bypass the concurrency limit on the phone since we are unable to change this setting directly. Figure 4(c) shows that the phones can indeed attain a higher level of concurrency. For example, iPhone and G2 can establish up to 9 concurrent connections for some content-rich URLs. The concurrency for other phones are slightly lower (6 to 7), likely due to their slower rendering and execution speed. Generally, an improvement of 30% is observed when concurrency limit on the phone is removed. This means that given the selected popular URLs, and given current network condition (with RTT of 400 ms), the default concurrency setting on mobile browsers appears to be too conservative. Allowing higher concurrency can help phones make better use of available bandwidth and save page downloading time.

6.3 Content compression

Compression can dramatically reduce web content size. For text objects, such as HTML, CSS, Javascript, PHP, etc., the object size can be reduced by around 70%. Usually, a web server does not compress image objects. We calculate the compression ratio for the popular URLs in column *Compress* of Table 3, showing that the content size can be reduced by more than 50% for most of the URLs we studied.

While compression reduces the bytes transferred over the network, decompression will increase computation overhead on the phone. To understand this tradeoff, we vary RTT covering the measured range and compare the web browsing performance in com-

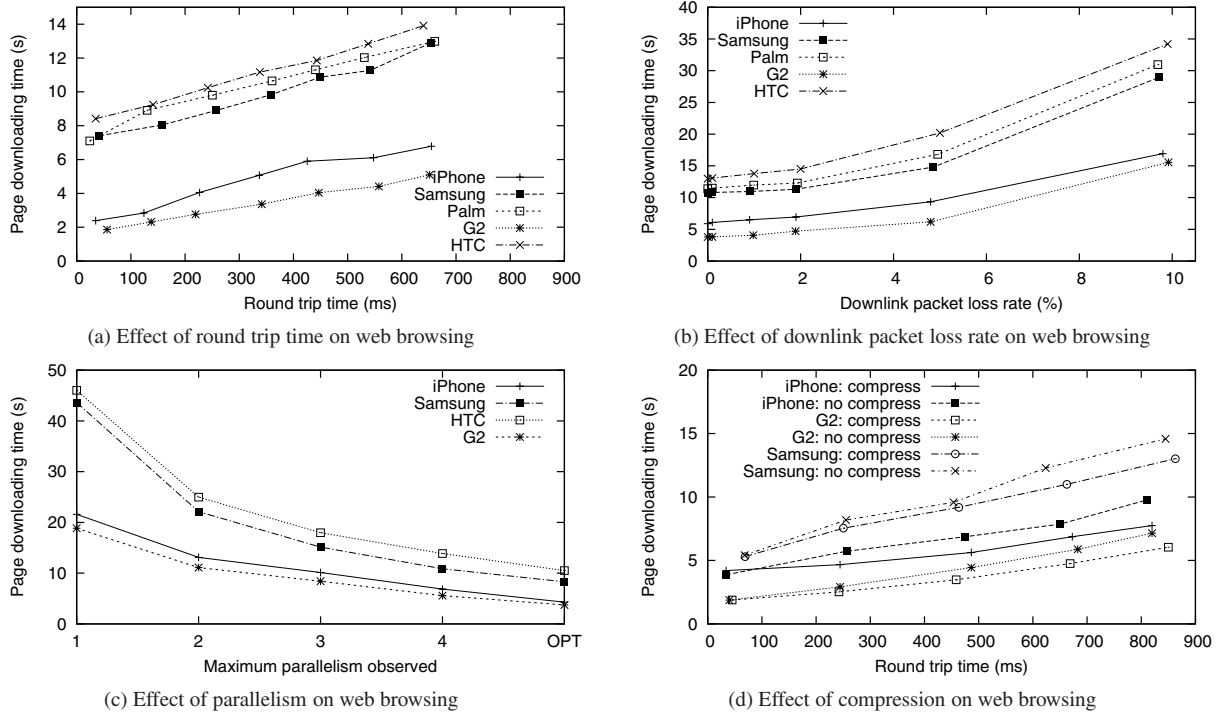


Figure 4: Factors impacting web browsing performance

URL	Text ¹	Image	Size(KB)	Original(KB) ²	Compress ³	GZIP	Lines/index ⁴	Redirect	Server IPs
www.google.com	4	1	79.2	77.6	2.56	✓	14	-	2
m.bing.com	4	3	42.9	218.1	1.46	-	2	-	1
maps.google.com	6	10	479.8	656.0	2.78	✓	8	-	4
mapquest.com	6	13	135.1	1326.35	1.96	✓	752	2	6
xhtml.weather.com	22	9	41.4	977.3	2.53	-	70	4	2
m.youtube.com	5	3	77.6	490.1	2.34	✓	231	-	3
m.ebay.com	4	3	58.6	484.0	2.17	-	1	-	1
m.facebook.com	4	1	19.7	399.1	2.81	✓	7	2	2
m.myspace.com	3	2	14.6	600.2	2.6	✓	98	1	2
m.fox.com	4	26	306.6	2083.0	1.16	✓	297	-	4
mobile.craigslist.org	3	0	113.8	113.8	3.58	✓	652	-	1

¹This column shows the number of text objects including HTML, JavaScript and CSS files

²This column shows the total size of the original website for each mobile URL, for example, www.bing.com for the row of m.bing.com

³This column shows the compression ratio for mobile URLs, total size in no compression mode / total size in compression mode

⁴This column shows the total number of lines in the index page indicating whether minification is used

Table 3: Characteristics of today's popular mobile websites

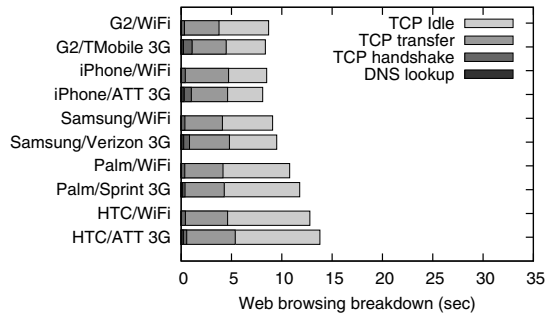
pressed and uncompressed modes. In Figure 4(d), we exclude the results for HTC and Palm phones as they show similar trends. We observe that compression consistently helps to improve web performance, irrespective of the RTT values. It is especially helpful under poor network condition. For example, it reduces iPhone's page downloading time by 30% when RTT is 800ms.

6.4 Javascript execution

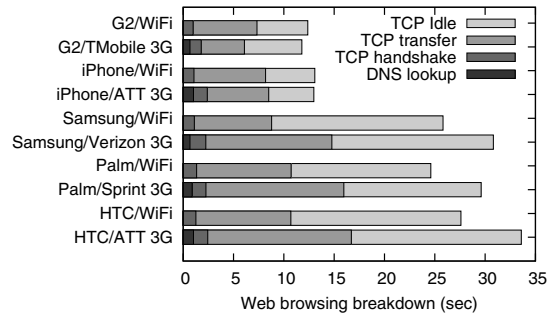
Given the limited processing power on smartphones, HTML rendering and Javascript execution may become the bottleneck for web browsing. Several factors jointly determine the page processing speed, including CPU frequency, memory, OS, and browser. Even

for the same OS, such as Windows Mobile 6.1, phone vendors can have different builds for different models of phones.

We measured Javascript execution time on different phones using a benchmark consisting of 26 different Javascripts [8]. Figure 5(a) shows the total time taken to execute the benchmark on different phones. The results demonstrate that execution time is 20-80 times longer on smartphones than on desktop computers. Among the smartphones, G2 has the best performance followed by iPhone. For example, G2 is 3 times faster than the HTC phone. Such performance gap helps to explain the differences in the page loading time of G2 and iPhone compared to that of the Samsung and Palm phones under the same network conditions in Figure 4.



(a) Web browsing breakdown for m.ebay.com (URL a)



(b) Web browsing breakdown for mapquest.com (URL b)

Figure 7: Web browsing anatomy for two popular mobile websites

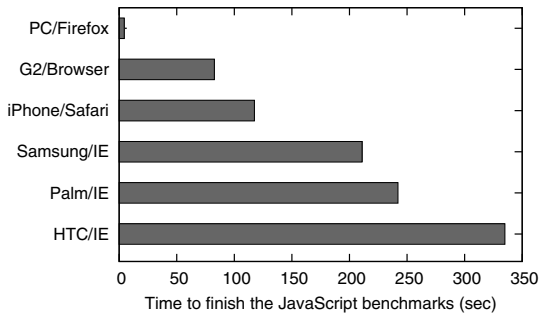


Figure 5: Script execution speed for different platforms

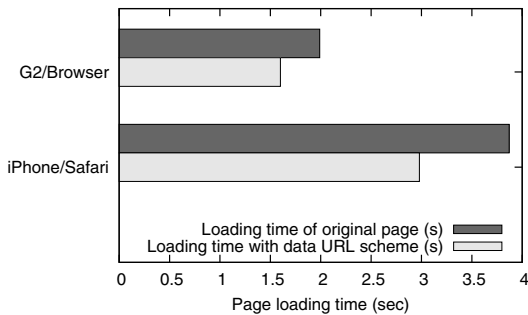


Figure 6: Evaluation for data URL scheme

Large Javascript execution time leads to more TCP idle time and under-utilization of network bandwidth.

This experiment shows that network is not the only bottleneck for web browsing performance. Phone itself also plays a major role, underscoring the necessity of measuring application performance on real smartphones. Web designers should avoid using complex Javascripts when building mobile versions of their websites.

6.5 Server configuration & content optimization

Server configurations and content optimization are important factors for web browsing performance. One type of server configuration is the maximum concurrent connections with a client. In §6.2, we found that mobile browsers set a default concurrency limit of 4 per domain. However, we did not observe any web servers limit the concurrency per client to be smaller than 4, likely because

servers have the incentives to attain good web browsing experience. The compression configuration is similarly important, with the identified setting of the URLs studied shown in the *GZIP* column in Table 3. Despite the fact that compression almost always helps with web browsing performance (§6.3), we found some websites do not enable it by default.

Various content optimization techniques also help to improve web browsing performance on smartphones. Most popular websites already customize their contents for mobile users, with more concise texts and fewer and smaller images, *e.g.*, via code minification [2], image scaling, *etc.* We study in particular code minification which refers to the elimination of redundant characters, such as spaces, tabs, line breaks, *etc.* The size reduction varies from 5% to 25% for the URLs studied. Column *Lines/index* in Table 3 shows the number of lines in the index page of a website, providing a hint for whether minification is used. The number of lines will be small for index pages treated with minification. It seems that half of the URLs use this technique to optimize their contents.

Another type of optimization helps to reduce the number of HTTP requests used to fetch contents, including the data URL scheme [15], CSS3, *etc.* The general idea is to eliminate TCP connections and HTTP requests for small objects, such as the corner image of a page. We set up a controlled experiment to demonstrate the effectiveness of the data URL scheme, under which small images are integrated with the main HTML page rather than linked as separate objects (4.4). In our experiment, we found that the images are actually 1.3-1.5 times of its original size under the data URL scheme. Figure 6 shows that it cuts page loading time by about 20%.

The data URL scheme has not been ubiquitously supported. In fact, only the browser of iPhone and G2 supports it. We also did not observe any URLs we studied adopt this technique, possibly due to the concern of lack of browser support. Without browser support, the image represented by the data URL scheme will be displayed as a default error image.

Redirection (HTTP response code 301 and 302) is another issue which may adversely impact web browsing performance. For mobile web browsing, this issue becomes more pronounced given the large RTTs in 3G networks. In column *Redirect* of Table 3, we found that some websites have multiple levels redirections. For example, m.weather.com will be redirected to xhtml.weather.com and then to mw.weather.com. In some cases, users are redirected to another URL which is quite similar to the original one. In other cases, web objects have been moved to a new location, and the original URL simply redirects the incoming re-

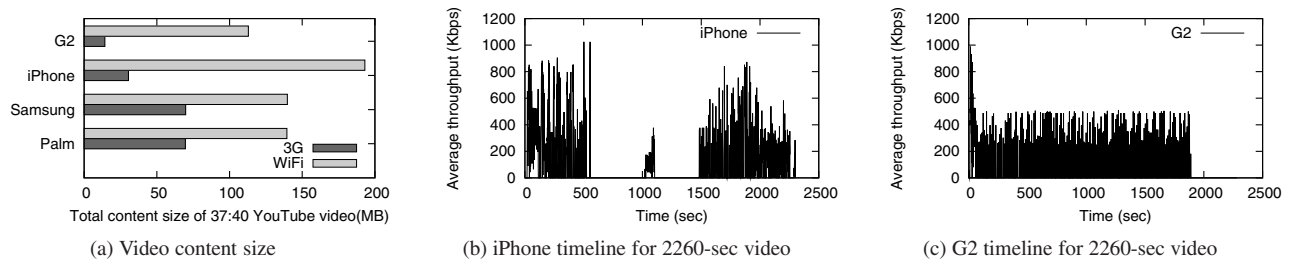


Figure 8: Streaming video performance: content size, download strategy.

quests to the new location. We think some of these redirections are unnecessary and can be eliminated with better webpage design.

6.6 Web browsing via 3G networks

Figure 7 shows the case study for two groups of URLs listed in Table 3. *Group A* corresponds to the URLs that have concise and simple contents, e.g., `m.ebay.com` (*URL a*) contains 7 objects with a total size of 58.6 KB. Many of these websites are search engines or portals to social networking sites, including `www.google.com`, `m.bing.com`, `m.myspace.com`, and `m.facebook.com`. *Group B* consists of websites with rich contents, e.g., `mapquest.com` (*URL b*) has 19 objects with a total size of 135.1 KB. Other websites in the group include online map (`maps.google.com`), information exchange (`mobile.craigslist.org`), and news (`m.fox.com` and `m.cnn.com`).

There are two sets of data in Figure 7. One set is collected when each smartphone visits the real URLs via 3G networks. To eliminate the differences in downloaded contents and network conditions, each phone also visits the replicated URLs via WiFi with a RTT of 400 ms to emulate the typical 3G network conditions.

It is clear that all smartphones experience smaller page loading time for the simple URL in Figure 7(a) compared with that for the content-rich URL in Figure 7(b). We break down the page loading time into four parts: TCP transfer, TCP idle, DNS lookup, and TCP handshake. The size of `mapquest.com` is larger than that of `m.ebay.com`, resulting in longer TCP transfer time. Moreover, `mapquest.com` contains more contents to render and more complex Javascripts to execute, leading to longer TCP idle time. The DNS lookup time and TCP handshake time contribute to less than 10% of page loading time, which are negligible.

We further observe that the Palm (Sprint), Samsung (Verizon), and HTC (AT&T) phones experience much longer page loading time for `mapquest.com` compare to iPhone (AT&T) and G2 (T-Mobile). This is likely due to their slower Javascript execution speed, as shown in Figure 5.

In the WiFi experiments, all the phones download the same contents and experience the same network conditions. As a result, the TCP transfer time differences among all phones are small. However, we can still observe significant page loading time differences, mostly due to the gap in TCP idle times. We further note that their relative ranking is consistent with the ranking of Javascript execution speed in Figure 5.

Summary: First, we found that higher browser concurrency enables the phones to better utilize available network bandwidth, hence reducing page loading time. Second, server configurations and content optimization play a major role in web browsing performance. Compression tends to always help under typical 3G network conditions. However, a few popular websites are employing sub-optimal server configurations and page designs. Third, we

found the bottleneck for web browsing performance often lies in the phone itself rather than in the 3G network.

7. OTHER MOBILE APPLICATIONS

In this section, we study two other popular mobile applications, streaming video and VoIP.

7.1 Streaming video

We downloaded a 37-minute long video using a YouTube player on each phone. Figure 8(a) shows the size of the video downloaded using TCP via WiFi and 3G on each phone. As expected, the video size is smaller for 3G than for WiFi, because both the video server and 3G carrier can customize video based on network conditions to ensure good user experience. Interestingly, the video size for 3G also varies across carriers: it is the smallest for T-Mobile, followed by AT&T, Verizon, and Sprint.

Figures 8(b)(c) show the representative time series of video download throughput for iPhone and G2 via 3G networks. The timeline of iPhone exhibits a distinct pattern with clear pauses. It initially downloads a portion of the video at a high rate, then stops before downloading the remaining portions. We conjecture that the download stops when the buffered content exceeds certain threshold, and resumes after the buffered content falls below another threshold. The purpose is likely to accommodate the limited phone memory and to save energy usage associated with the 3G interface. Another observation is that iPhone always terminates the TCP connection every 10-20 seconds and then establishes a new one on demand. We conjecture that iPhone attempts to put the 3G interface into low power state to save energy.

In contrast, G2 shows a different behavior by periodically downloading small chunks of the video every 10 seconds. The Samsung and Palm phones behave similarly with a slightly longer interval of 20 seconds between downloads. This is likely motivated by the fact that users sometimes do not watch the entire video and may skim through certain parts of the video. Such downloading patterns can also help to save energy. Our initial study shows that the video players on different phones employ different policies to fetch video. This merits more detailed future study.

7.2 VoIP

We carry out a simple VoIP experiment on the Samsung (Verizon) and Palm (Sprint) phones given their uniform support for Skype. During the experiment, the same music file is played on both the phone and the desktop, when the two are in a Skype call. The volume is kept the same to have similar voice input. Figure 9 shows that the throughput for both phones via 3G is nearly identical, as the same coding rate is used. The throughput is higher under WiFi than under 3G, as different amount of data is transferred depending on the network being used. This reflects how Skype tries

to vary the encoding rate according to the network condition to achieve good perceived voice quality.

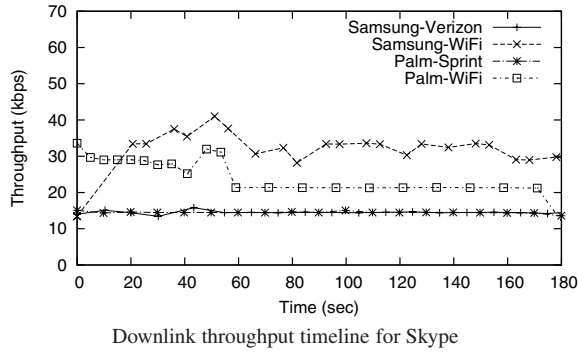


Figure 9: VoIP performance

8. CONCLUSION

In this paper, we characterized the performance of network applications on smartphones in a way that is relevant to end users, cellular operators, smartphone vendors, application developers, as well as content providers. We comprehensively studied 3G network performance by leveraging our widely-deployed measurement tool, *3GTest*. We carefully devised a set of experiments to quantify how application performance, in particular web browsing, is impacted by various factors, and where the performance bottleneck is. Our analysis provides insight into how network operators and smartphone vendors can improve 3G networks and mobile devices, and how content providers can optimize mobile websites.

Our work represents an important step towards a better understanding of the performance of 3G networks and smartphone applications. As future work, we will continue to collect data from *3GTest* and study the network and application performance differences across various locations. We also plan to study how webpage structure and web object dependency affect page loading time.

9. ACKNOWLEDGEMENTS

We thank all the reviewers and especially our shepherd Bill Schilit for their constructive comments and suggestions. This work is partially funded by NSF CNS-0643612, DARPA Computer Science Study Panel, and ARMY Young Investigator Award.

10. REFERENCES

- [1] Alexa Top 500 Global Sites. <http://www.alexa.com/topsites>.
- [2] Best Practices for Speeding Up Your Web Site. <http://developer.yahoo.com/performance/rules.html#minify>.

- [3] Canals Press Release: R2009112. <http://www.canalys.com/pr/2009/r2009112.htm>.
- [4] FCC Consumer Broadband Test. <http://broadband.gov/qualitytest/>.
- [5] ICSI Netalyzr. <http://netalyzr.icsi.berkeley.edu/>.
- [6] Smartphone 3G Test (3GTest). <http://www.eecs.umich.edu/3gtest>.
- [7] Speedtest.net. <http://www.speedtest.net/>.
- [8] SunSpider JavaScript Benchmark. <http://www2.webkit.org/perf/sunspider-0.9/sunspider.html>.
- [9] M. Balakrishnan, I. Mohamed, and V. Ramasubramanian. Where's That Phone?: Geolocating IP Addresses on 3G Networks. In *Proceedings of IMC*, 2009.
- [10] 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.
- [11] M. C. Chan and R. Ramjee. TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation. In *Proc. of MOBICOM*, 2002.
- [12] J. Chesterfield, R. Chakravorty, J. Crowcroft, P. Rodriguez, and S. Banerjee. Experiences with Multimedia Streaming over 2.5G and 3G Networks. *Journal ACM/MONET*, 2004.
- [13] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley, and R. Cruz. Modeling TCP in a Multi-Rate Multi-User CDMA System. In *IFIP-Networking 2007*, 2007.
- [14] 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.
- [15] L. Masinter. The "data" URL Scheme. RFC 2397, 1998.
- [16] 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.
- [17] R. Mahajan, M. Zhang, L. Poole, and V. Pai. Uncovering Performance Differences in Backbone ISPs with Netdiff. In *Proceeding of NSDI*, 2008.
- [18] K. Mattar, A. Sridharan, H. Zang, I. Matta, and A. Bestavros. TCP Over CDMA2000 Networks : A Cross-Layer Measurement Study. In *PAM*, 2007.
- [19] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proc. ACM SIGCOMM*, 1998.
- [20] W. L. Tan, F. Lam, and W.-C. Lau. An empirical study on 3g network capacity and performance. In *Proc. IEEE INFOCOM*, 2007.
- [21] 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.
- [22] W. Wei, C. Zhang, H. Zang, J. Kurose, and D. Towsley. Inference and Evaluation of Split-Connection Approaches in Cellular Data Networks. In *PAM*, 2006.
- [23] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz. Primary Users in Cellular Networks: A Large-scale Measurement Study. In *DySpAN*, 2008.
- [24] Z. Zhuang, T.-Y. Chang, R. Sivakumar, and A. Velayutham. A3: Application-Aware Acceleration for Wireless Data Networks. In *Proc. of ACM MOBICOM*, 2006.