A First Look at Traffic on Smartphones

Hossein Falaki
CENS, UCLA

Dimitrios Lymberopoulos
Microsoft Research

Ratul Mahajan
Microsoft Research

Srikanth Kandula
Microsoft Research

Deborah Estrin
CENS, UCLA

Abstract—Using data from 43 users across two platforms, we present a detailed look at smartphone traffic. We find that browsing contributes over half of the traffic, while each of email, media, and maps contribute roughly 10%. We also find that the overhead of lower layer protocols is high because of small transfer sizes. For half of the transfers that use transport-level security, header bytes correspond to 40% of the total. We show that while packet loss is the main factor that limits the throughput of smartphone traffic, larger send buffers at Internet servers can improve the throughput of a quarter of the transfers. Finally, by studying the interaction between smartphone traffic and the radio power management policy, we find that the power consumption of the radio can be reduced by 35% with minimal impact on the performance of packet exchanges.

Categories and Subject Descriptors
C.2.5 [Computer-Communication Networks] Local and Wide-Area Networks – Internet

General Terms
Measurement, Performance

Keywords
Smartphone traffic, Power management

1. INTRODUCTION

Smartphone traffic represents an increasingly large share of Internet traffic. Cellular traffic is projected to grow 10 times faster than fixed Internet traffic [22] and most of this traffic is generated by smartphones [9]. By next year, smartphone sales are projected to surpass desktop PCs [18].

However, little is known today about the nature of smartphone traffic. Two recent studies have shed valuable light on some aspects of this traffic. Trestian *et al.* study the kinds of Web sites accessed at different times of the day [20]; and Maier *et al.* study HTTP traffic generated by mobile handheld devices (which include music players and personal gaming consoles in addition to smartphones) in homes [15]. Both studies are based on data gathered at a link in the middle of the network. As a result, while they can analyze traffic from a large number of devices, they do not capture a detailed, comprehensive view of individual devices. For instance, the second study misses traffic exchanged by devices through the cellular interfaces or outside of their homes.

In this paper, we report on our ongoing work on detailed characterization of smartphone traffic. Our approach is complementary to that of previous studies—we employ passive sniffers on the device and record all sent and received traffic. Given the difficulty of deploying continuous monitoring on a large number of end user devices, the breadth achievable using this method is limited. But the comprehensive view of smartphone traffic that it provides for monitored devices enables inferences that would otherwise be impossible to make. For instance, we can study how much total traffic a device generates in a day and interaction of its traffic patterns with radio power management.

The results in this paper are based on two datasets. Our primary dataset consists of 10 users across two smartphone platforms. For these users, we deployed a logger that captured packet-level traces. Our other dataset consists of 33 Android users. It contains bytes sent and received by each application in every two-minute window. We have from 1 to 5 months of data for each user.

Using these datasets, we shed light on several aspects of smartphone traffic. By analyzing commonly used ports and applications, we quantify traffic generated by various applications. We find that browsing contributes over half of the traffic, while each of messaging (email, IM), maps and media contribute roughly 10%.

We also find that most smartphone data transfers are small, with the median size being only 3 KB. Such small transfers have many implications. For instance, the overhead of lower layer protocols can be high. We show that for half of the transfers, header bytes constitute over 12% of the total bytes. In the presence of transport security, this overhead grows to 40%. For half of the transfers, lower layer handshakes constitute 20% of the total completion time.

Consistent with controlled experiments with probe traffic [12], we find that smartphone data transfers experience high delays and losses. Unlike controlled experiments, however, our data allows us to also study the impact of these path characteristics on actual smartphone workloads. Focusing on transfers with more than 10 data packets, we find that the median throughput is only 3.5 kbps in the downlink (from the network to the smartphone) and 0.8 Kbps in the uplink.

Analysis of what limits the throughput of smartphone data transfers [23] reveals that packet loss is the primary culprit. But interestingly, a quarter of the downlink transfers are bottlenecked by the size of the sender-side transport buffer. The throughput of such flows can be improved by simply increasing the buffer sizes at servers that communicate with smartphone clients.

Finally, we study the interaction of smartphone traffic with the radio power management policy. We find that the current sleep

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.
IMC'10, November 1–3, 2010, Melbourne, Australia.
Copyright 2010 ACM 978-1-4503-0057-5/10/11 ...$10.00.

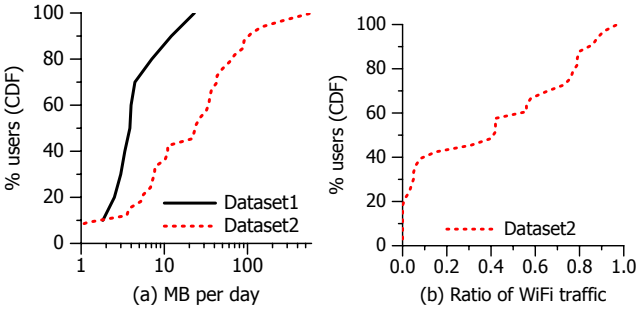

Figure 1: (a) Smartphone traffic per day. (b) Ratio of traffic sent on the WiFi interface.

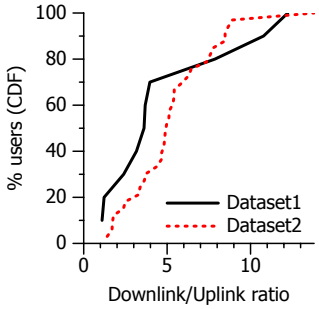

Figure 2: Ratio of downlink to uplink traffic.

timers, that is, the idle period after which the radio will go to sleep, are overly long. By reducing them based on current traffic patterns, radio power consumption can be reduced by at least 35% with minimal impact on performance.

2. DATASETS

Our results are based on two sets of traces. The first data set consists of packet level traces from 10 smartphone users across two different platforms. Our second data set contains application level traffic information from 33 Android users.

Dataset1 Our first dataset is from 8 Windows Mobile (HTC Touch) users and 2 Android (HTC Dream) users. It contains packet-level traces, including link layer headers, for data sent and received by the smartphone. We collected these traces using *Netlog* on Windows Mobile and *tcpdump* on Android. The traces were stored locally and uploaded at regular intervals using the USB connection.

All users are knowledge workers. Each had an unlimited data plan with their carrier (7 AT&T, 3 T-Mobile). The users were resident in two different cities in the USA.

Across all users, there is 532 days of data. For individual users, the data varies from 26 to 84 days.

Dataset2 Our second dataset is from 33 Android (HTC Dream) users. It was collected using a custom logging tool that provides an application-level view of smartphone traffic. Every two minutes, it records the number of bytes sent and received by every process that runs on the smartphone. This tool is available to other researchers by request and more details on its operation are available in [10].

The set of users consists of 17 knowledge workers and 16 high school students. As for user interactions [10], we did not find statistically valid differences among the two demographics with respect to traffic. We thus present their results jointly. Each participant was provided an unlimited voice, text and data plan through T-Mobile. The users were resident in the same city in the USA.

Across all users, there is 1660 days of data. For individual users, the data varies from 49-147 days.

A key limitation of our work is the small user populations of our datasets, even though they provide independent vantage points on smartphone traffic characteristics. Expanding our set of users is a subject of ongoing work.

3. TRAFFIC COMPOSITION

In this section, we study the basic makeup of smartphone traffic, starting with volume per user. Figure 1(a) shows how much traffic is exchanged per day by users. This amount is 2-20 MB in Dataset1 and 1-500 MB in Dataset2. Compared to residential broadband traffic this is roughly one order of magnitude smaller [1].

Two factors may explain the differences in the two datasets. One is that Dataset1 is dominated by Windows Mobile, while Dataset2 is exclusively Android. It is likely that the Android OS and users generate more traffic. The heaviest user in Dataset1 is in fact an Android user. In earlier work, we found that Android users interact with their devices more heavily than Windows Mobile users [10]. For instance, the median session length of Android users is more than twice that of Windows Mobile users.

The second factor, not unrelated to the first, is that many users in Dataset2 use WiFi heavily. Dataset1 does not provide direct information on the interface (WiFi or cellular) used by individual packets. But by observing interface addresses and path delays—cellular delays are much higher—we conclude that WiFi usage was minimal among those users. In Dataset2, we can reliably identify the share of WiFi traffic using information about interface state.

Figure 1(b) shows the ratio of WiFi traffic in Dataset2. We see that the median ratio is almost 0.5 but it varies widely across users. While the bottom 20% do not use WiFi at all, the top 20% use it for more than 80% of their traffic. These results also imply that depending on the user population smartphone studies based on only cellular traffic [20] or only WiFi traffic [15] can miss a significant fraction of device traffic.

We now study the composition of smartphone traffic from other perspectives.

Downlink vs. uplink Figure 2 shows the ratio of downlink (from the network to the smartphone) to uplink traffic. There is a wide variation among users, caused likely by diversity in application usage, from downlink traffic equaling uplink traffic to it being over 10 times the uplink traffic. The average across all users for downlink to uplink ratio is 6:1. This high asymmetry, indicating a strong bias towards downloads, has implications for provisioning access technologies for smartphones. It is comparable to asymmetry in residential broadband traffic in Europe [14] and Japan [11] but is well above the subset of "peer-type heavy-hitters"[11] whose ratio is close to 1:1.

Despite differences in total traffic exchanged the downlink to uplink ratios in the two datasets are similar. This suggests that the mix of activities that generate network traffic may not be disparate for the two cases. We study these next.

Common ports [Dataset1] Ports provide insight into user activities on smartphones. Identifying applications using ports may be inaccurate in some cases (e.g., peer-to-peer), but it is a simple indicator that works well overall [14].

Table 1 shows for Dataset1 all ports that carry over 0.1% of the

	Bytes (%)	Packets (%)
HTTPS (443)	43.88	31.66
HTTP (80)	37.48	22.16
IMAP4S (993)	15.21	39.32
DNS (53)	1.08	2.31
IM (5000-01)	0.69	0.32
Android-Mkt (5228)	0.48	1.10
IPv6local (5355)	0.22	0.88
DHCP (67)	0.22	0.24
NetBIOS (137-39)	0.17	0.60
other	0.57	0.37

Table 1: Ports (in parenthesis) used by IP packets in Dataset1.

	Bytes (%)
Browsing	58.02
Media	10.82
Messaging (Email, IM)	10.33
Maps	8.51
System	5.83
Social networking	4.18
Games	0.36
Productivity	0.15
unknown/other	1.79

Table 2: Traffic generated by applications in Dataset2.

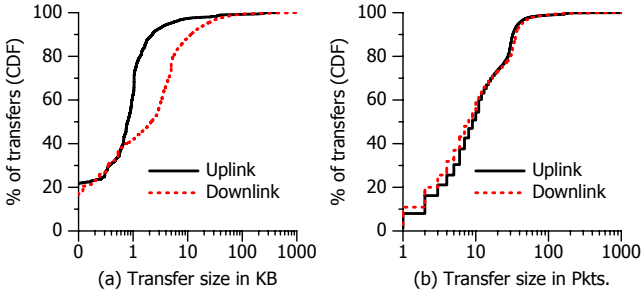

Figure 3: Transfer sizes in Dataset1. The x-axes are log scale.

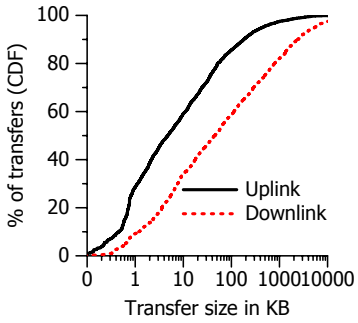

Figure 4: Transfer sizes in Dataset2. The x-axis is log scale.

bytes. We see that the dominant ports correspond to HTTPS, HTTP, and IMAP4S. These results suggest that the main traffic generators on smartphones of these users are browsing and email. HTTPS is used by secure Web sites and email servers (including Exchange). HTTP of course is used heavily as part of browsing and for downloading data of various kinds. IMAP4S is the secure version of the IMAP protocol for email. While IMAP4S has the most packets, it is third with respect to the number of bytes. This implies a bias towards small packets, likely generated as part of frequently polling for (often non-existent) new email. Many Dataset1 users had two configured email accounts—a push-based work account and a polling-based personal account. Increased adoption of push-based email, by which clients are notified when new email arrives, will change the nature of email traffic.

The 37% share of HTTP traffic that we find is roughly half of that observed for mobile handheld devices in homes [15] and 50% less than that in residential broadband traffic [14].

The large volume of traffic that uses HTTP or HTTPS perhaps indicates the trend among smartphone applications to tunnel their data through these protocols, even when such data would not normally be considered HTTP payload (e.g., music, video and, social apps).

Applications [Dataset2] Our second dataset lets us directly observe what applications are generating smartphone traffic. We partition applications into several categories that are shown in Table 2. "System" includes applications that are part of the OS (e.g., package manager, backup), and "Productivity" includes applications for calendars, alarms, and document handling (e.g., Office, PDF reader). The meanings of the other application categories are what their names suggest.

We see in the table that browsing dominates smartphone traffic. As in Dataset1, messaging is also a significant contributor. Media and maps are other major contributors. These applications tend to use HTTP and HTTPS for transport, but the application-level view lets us quantify their contribution independently.

4. TRANSFER SIZES

In this section, we study the sizes of individual data transfers, which impact throughput as well as power consumption [2]. We identify individual transfers using TCP flows. A TCP flow is identified using IP addresses and ports. Packets of a flow without an extended idle period (1 minute), are considered as part of one transfer; flows with long idle periods are considered separate transfers [6]. Long idle periods can arise within a TCP flow if the client (e.g., email application) maintains an open connection to the server, to avoid the TCP connection setup overhead for each transfer.

Figure 3 shows the CDF of transfer sizes in bytes and packets across all users in Dataset1. The size in bytes includes the bytes contributed by TCP and IP headers. While the mean transfer size is 273 KB sent and 57 KB received, most transfers are extremely small. When considering both directions cumulatively, 30% of them have fewer than 1K bytes and 10 packets. These results are consistent with those of Maier *et al.* for HTTP traffic from handheld devices [15].

Figure 4 shows that Dataset2 is dominated by small transfers as well. We define a transfer size differently in this case. Dataset2 has bytes sent and received by individual applications in 2-minute long intervals. For each application, we combine contiguous intervals with non-zero data exchanged as one transfer. If an application exchanges data over multiple TCP connections, this definition aggregates data across those connections into one transfer. Despite this aggregation, the graph shows that most transfers are small.

The small transfer sizes that we observe have many implications. Given the high amount of energy consumed by the 3G radio to go from sleep to ready state and from idle to sleep state (§6), there can

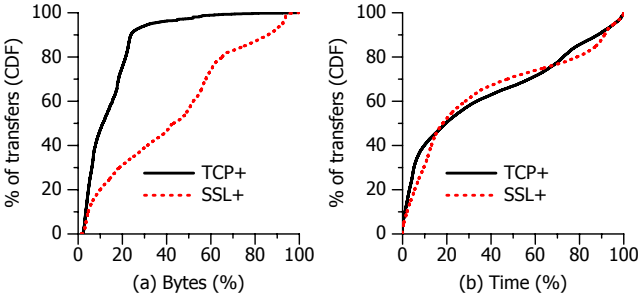

Figure 5: The overhead of layers below TCP and SSL (inclusive) in Dataset1.

be a high energy overhead associated with them. Another implication is that a scheme like Catnap [8] is unlikely to reduce radio power consumption. Catnap puts the radio to sleep during transfers but is effective only for long transfers. In §6, we present a simple method that can reduce radio power consumption by 35% by appropriately setting radio sleep timers.

Yet another implication of small transfers is that the already high overhead of lower-layer protocols can dominate. This overhead manifests as extra bytes that must be transmitted as headers as well as extra time that it takes to complete handshakes.

We quantify these overheads at the transport (TCP) and transport security (SSL) layers. According to our analysis 96% of smartphone traffic is TCP-based and more than half uses SSL (through HTTPS and IMAP4S). In Figure 5, "TCP+" captures overhead of TCP and all layers below it. "SSL+" captures overhead of SSL and all layers below it, and it is computed only for SSL-based transfers.

Figure 5(a) shows that the median TCP+ overhead at byte-level is 12%, i.e., more than one in ten bytes is devoted to TCP or lower layer headers. SSL further increases overhead. The median SSL+ overhead is 40%, and 20% of the transfers have an overhead that is twice that amount.

Figure 5(b) shows the time overhead. TCP+ is measured as the time between the first SYN and the first packet that contains non-TCP bytes. If the radio is asleep when the SYN is sent, this measure will include the time to wake up the radio. In the next section, we quantify the radio waking overhead separately. SSL+ is measured as the time between the first TCP SYN and the first packet that contains non-TCP, non-SSL bytes. Thus, in addition to the TCP handshake, it includes any time needed for SSL key exchange.

We see that the time overhead too is significant. The median overhead of TCP is 20%, i.e., a fifth of the total transfer time is spent waiting for TCP handshake to complete.

Surprisingly, SSL does not add much to the time overhead beyond TCP, which points to the effectiveness of SSL session key caching for smartphone workloads. Smartphones frequently talk using SSL to the same server (e.g., email server). Cached session keys enable quick connection establishment without the overhead of full key exchange. Most of the additional overhead due to SSL appears to be due to their larger headers. In Figure 5(b), the SSL+ overhead appears slightly lower than TCP+ in some cases because the two curves are computed over different sets of transfers.

In summary, we find that most smartphone transfers are small. Such transfers have a high energy cost and amplify the overhead of lower-layer protocols. One way to avoid this overhead, which we will investigate in the future, is to aggregate transfers across applications and across time. Using a proxy in the cloud can facilitate such aggregation.

5. PERFORMANCE

We now investigate the performance of TCP transfers. We study observed round trip time, throughput, and retransmission rate as well as estimate what limits the transfer throughput. We use only Dataset1 because Dataset2 does not have the granularity of information needed for this analysis. As almost all of Dataset1 traffic is 3G-based (§3), our analysis sheds light on the performance that is seen by smartphones when using the 3G interface in real operating conditions.

5.1 Round trip time

We estimate the RTT of a transfer as the difference between the SYN and SYN-ACK packets. Accurate inference of RTT using data and acknowledgment packets is complicated by delayed acknowledgments. If multiple SYN packets are transmitted for a transfer, we use the last SYN packet. In some cases, the SYN-ACK packet may be sent by a proxy in carrier network instead of the contacted server. Even in these cases, we get a good estimate if the dominant component of the RTT is the wireless delay [12].

We explicitly correct for a source of error that would otherwise significantly overestimate network RTT. If the radio is asleep when the transfer is initiated, it includes the times it takes for the radio to wake up and synchronize with the tower. To weed out this impact, we focus on transfers that are initiated when the radio is in full power mode. We identify such transfers as those that are initiated within 3 seconds of the previous transmission or reception. The idle period after which the radios go into a lower power mode is well above this threshold (§6).

The "Trailing" curve in Figure 6(a) shows the RTTs observed by such transfers. We see that the median is 125 ms but 10% of the transfers observe an RTT of over 0.5 seconds. Such high variance in RTT is consistent with controlled experiments [12]. A TCP flow that experiences high variance in RTT will suffer from delayed response to congestion among other things. Such variance can stem from a host of factors including link layer retransmissions (that are not visible to us), network congestion, and overloaded equipment inside the carrier network.

The "All" curve in the graph represents RTTs computed for all transfers, not just those initiated when the radio is awake. The difference in the two curves quantifies the overhead of radio wake-ups. The difference is 400 ms at the median and 1.7 seconds at 90th percentile. The variation stems from the variable amount of time the radio takes to fully synchronize with the tower and a wake-up may already be in progress because of another transfer.

5.2 Retransmission rate

We now study how frequently packets are retransmitted in TCP transfers. Retransmissions are identified using sequence numbers and provide a good estimate of path loss rate. Their rate can differ slightly from loss rate due to TCP dynamics such as spurious timeouts.

Across all transfers, the uplink retransmission rate is 3.7% and the downlink rate is 3.3%. These loss rates are much higher than those for wired paths. The median average loss rate seen from the SLAC laboratory during 2008 was less than 0.1% for North America and less than 1% for most of the world [7]. However, our observed wireless loss rate is similar to those inferred using controlled experiments [12]. We show below that packet loss is the main bottleneck for TCP throughput.

Figure 6(b) shows the retransmission rates for individual transfers. This graph is based only on connections that send more than 10 data packets in a given direction. We see that roughly 60% of the connections experience no retransmissions. But 25% of them