MIST: Cellular Data Network Measurement for Mobile Applications

Mike P. Wittie, Brett Stone-Gross, Kevin C. Almeroth and Elizabeth M. Belding
Department of Computer Science, University of California, Santa Barbara
Santa Barbara, CA 93106-5110
{mwittie, bstone, almeroth, ebelding}@cs.ucsb.edu

***Abstract*— The rapid growth in the popularity of cellular networks has led to aggressive deployment and a rapid expansion of services. Services based on the integration of these networks into the Internet have only recently become available, but are expected to become very popular. One current limitation to the deployment of many of these services is poor or unknown network performance, particularly in the cellular portion of the network. Our goal in this paper is to motivate and present the Mobile Internet Services Test (MIST) platform, a new architecture to measure and characterize cellular network performance as experienced by mobile devices. We have used MIST to conduct preliminary measurements; evaluate its effectiveness; and motivate further measurement research.**

I. Introduction

Recent years have seen a rapid expansion of data services in cellular networks. As aggressive competition between cellular service providers has lead to decreases in average per-user revenue, network providers began to look at ways to increase non-voice revenue [1]. Studies of customer network usage have indicated the propensity of customers, especially in the US, to adopt new network services hinting at the possibility of increasing data usage via heavier application traffic [2]. The future is also expected to bring an ever increasing integration of various multimedia, packet, and location-based cellular data network services. These services will certainly increase the demands on these networks [2], [1].

The increased customer and service provider focus on data services has attracted the interest of mobile application developers ready to cash in on a new trend. These companies have recognized the opportunity to create revenue generating value chains between themselves and the network providers, and therefore, are rapidly exploring the market space [3]. Popular mobile applications, well supported by the network infrastructure, benefit both the network providers and application developers by driving both data service usage, as well as revenue from application downloads and service fees. For an application to be successful, however, it needs to be brought to the market quickly and offer a high quality service. Beating the competition even by a week can make a major difference in the level of application adoption by the user community. The high quality of an application, equally important to its success as time to market, needs to be assured for a large number of devices and network provider combinations [4].

Studies of the mobile application development processes have identified a large number of challenges to application development time and interoperability [1], [5], [6], [4]. Two main challenge areas have been identified. First, while JAVA is becoming the preferred language of mobile application developers, various mobile JAVA Virtual Machine implementations are riddled with inconsistencies, necessitating custom code optimizations for many platforms [5]. Second, the lack of information about the behavior and performance of different network protocols on different devices and network providers complicates mobile application development [4]. In addition, these performance issues also affect the ability to optimize many application types [6]. While the incompatibilities of JAVA are a problem known throughout the industry, and one that is being addressed, the detailed characterization of the cellular environment as seen by mobile devices remains largely unclear.

In spite of the growing interest in cellular data network performance, from the point of view of mobile device performance, the area remains largely unexplored. Most cellular data network studies have investigated the performance of particular network technologies [7], [8], [9], [10], [11], focusing on network infrastructure optimization and provisioning. Besides the underlying network technology, there are many factors affecting network performance, like radio technology, phone hardware, or network settings. Additionally, it is difficult to extrapolate the current state of technology due to staleness of results or difficulties in comparison across different studies. To aid mobile application developers and identify performance pitfalls on cellular data networks, a number of studies have focused on the relationship between these networks and the transport protocols used by the applications [12], [13]. While these transport protocol studies did offer insight to the networking community, other studies have shown that transport protocols optimized for the wireless environment, more often than not, do not translate to application performance improvements [14]. Indeed, the best way to assure good application usability and performance is to customize application code to the specific platform and network characteristics.

In this paper we propose the Mobile Internet Services Test (MIST) platform, an architecture designed to characterize cellular data network performance as experienced by individual mobile devices. The measurement network performance, network technology, and mobile hardware combinations exposes the effects of the interactions and particularities of the underlying technologies. Since mobile applications are expected to work on a wide variety of platforms, understanding the performance variations is crucial to making application design decisions, thus assuring high application adoption rates within the user community.

MIST is comprised of a mobile application connected to a server back-end. A number of network performance tests are performed between the mobile application and the MIST servers in order to assess network latency, jitter, throughput, and various timeout intervals. Each set of measurement data is saved in a database along with the network and mobile device configuration information used in the test. MIST is lightweight, stateless, and highly scalable. Most importantly, its deployment does not require changing or augmenting the cellular data network infrastructure. It can be deployed on top of mobile devices themselves, configured to measure exactly the network characteristics a mobile application developer is interested in, and report directly on the user experience.

Using a simple set of test data collected by MIST, we present a basic analysis of results for a Samsung MM-A700 mobile phone on the Sprint PCS network operating in Santa Barbara, California. This preliminary data set is used to demonstrate the future opportunities for measurement and analysis rather than a data set for an exhaustive analysis. The goal is to describe the kind of data that can be collected; how it can be analyzed; and what general kinds of conclusions can be drawn from its analysis. We then hypothesize how a detailed understanding of cellular data network performance can be built from an analysis of such collected data. Key questions we answer are: (1) what latencies and data rates exist? (2) what is the variability in network performance? (3) does user mobility affect network performance? (4) what future directions should cellular data network measurements take? By answering these questions we provide important insight into cellular data networks.

The remainder of this paper is organized as follows. In Section 2 we review related measurement efforts. Section 3 presents the MIST system architecture. Section 4 provides an analysis of a basic set of results. Section 5 describes future work. Finally, we conclude in Section 6 with a summary of our findings.

II. Background and Related Work

The need to increase data service revenue by cellular data network providers has created an opportunity for application developers to create a "value chain" connecting customers, mobile applications, and network providers. This relationship, and the drive of application developers to create value through mobile applications, has been studied by Karvonen and Warsta [3]. Mobile applications provide functionality to users who are willing to pay the application developer for that functionality via a download or service fee. As the application becomes popular, network providers benefit from the increase in data traffic. The increase in revenue often times, or at least should, translate into better or newer network infrastructure, in turn, further opening the market to application developers.

In order for an application developer to benefit from value chains, their application needs to reach the user quickly and perform as expected. The importance of rapid deployment and service quality has been detailed by Abrahamsson et al. [4]. While Abrahamsson's mobile application development methodology promises to reduce time to market, the technical problems of mobile application development remain formidable. There is, however, plenty of incentive to startups. Verkasalo presents a number of user behavior studies, indicating user interest in emerging data services and applications, as well as a high rate of adoption [2], [15]. Particularly in the US, mobile data service usage has a high percentage of Instant Message (IM) communications, indicating a trend toward data rather than voice communications.

TABLE I
Network centric mobile application types

Application Type	Challenges	Optimization Techniques
Streaming Media	- high jitter - low throughput	- buffering - layered encoding
Mobile Commerce	- high latency - security	- adaptive protocol design - minimized communications
Pervasive Gaming	- latency varied across systems	- system specific timeout values
Two-way Database	- radio timeout	- keep alive packets
Web Browsing	- low throughput - high load	- phone caching - backoff/queuing algorithms

To expand into the data communication market, a number of mobile application types have been proposed. In Table I we present a brief taxonomy of data centric mobile applications along with their challenges and common optimization techniques. Each application type is sensitive to variations in different network characteristics which can be alleviated with different optimizations techniques.

Streaming media performance on mobile devices, investigated by Walker et al., shows sensitivity to high jitter and low throughput [6]. Application developers can account for jitter, for example, by using buffering techniques. The choice of buffer size, however, is crucial to good application performance; too small a buffer might result in interrupted playback, while a large buffer may delay video start time or simply be infeasible on some memory-constrained devices.

Streaming applications are also sensitive to network throughput. If the network cannot deliver the data required in a timely matter, playback will not be possible in real time and the media will need to be downloaded in its entirety before playback can begin. To cope with this challenge, streaming media is often encoded using layers. Additional bandwidth allows the reception of multiple layers, with each layer increasing the quality of media playback. While the decoding of layered media can be computationally intensive, it is expected to be possible on mobile devices in the near future. While layered media allows application adaptation to reduced throughput, application developers will need to choose layer granularity appropriate for device and network combinations. Finer granularity causes additional computational load, while coarser encodings could result in under-utilization of network capacity, as well as degraded application performance.

The performance of mobile commerce is also affected by high latency and security network considerations. High latency caused by naive protocol design results in long wait times, reducing an application's attractiveness in todays fast paced world. On the other hand, a communication protocol that is needlessly terse might deprive users of a richer experience, also reducing the attractiveness of the application. A fine balance needs to be struck by application developers to give their users the best possible experience within the wait time they are expected to endure. Choices as to the amount of data that can be transmitted during a tolerable wait time depends on the latency and throughput performance of mobile device and network applications.

Additionally, mobile commerce applications also need to meet security and privacy standards users are accustomed to in today's Internet. While knowledge of some device or network vulnerabilities may mean that an application should not be used in certain cases, it is certainly the preferable option to instead giving the user a false sense of security. Finally, since encryption mechanisms are computationally intensive, mobile commerce application developers may want to limit encoded communications on some devices and reduce customer wait times.

Pervasive gaming is a relatively new form of mobile application. While mobile games have been a staple of mobile device software suites, extending these games with network functionality is a relatively new area. This expansion was predicted by Harmer, although the author's vision was limited to games played over long periods of time [1]. Airplay Networks[1] has recently introduced real-time online gaming to the mobile market. One of the requirements of their application is to present a consistent game state to all players. A major challenged faced by the developers was the characterization of latency variation between different mobile devices, network providers, network technologies, and geographical areas.

Mobile applications accessing databases have an additional challenge they need to solve. The quality of the user experience depends on the responsiveness of the application. Users, however, can be unresponsive for some time as they are processing the last query result, for example, when a user retrieves a mobile insurance claim application. During periods of inactivity, many wireless devices go into power-save mode and switch off their radios. Any subsequent transmission will be forced to take a delay penalty while the radio is turned back on. During longer periods of inactivity, the network may deallocate resources given to a mobile device resulting in an even longer delay penalty. Database access applications may want to limit delays by keeping the radio and network allocations alive with dummy transmissions if user response is expected imminently. To make decisions as to the dummy transmission interval and whether or not they are appropriate during a particular state of application execution, applications developers need to know the timeout periods on various mobile devices and underlying networks.

Finally web browsing, an application already becoming widespread, also has unique challenges in cellular environment. Due to the constrained or varied bandwidth available on mobile devices, websites for these devices need to be tailored to the expected network performance. Similar to streaming media, different encoding techniques can improve user experience. To complicate matters, mobile device websites will likely need to handle flash crowds. One such scenario is a stadium full of fans trying to download a replay of the latest play. The requests may need to be queued without degrading user experience. Knowing the network characteristics may allow application developers to predict queue times and offer that information to waiting fans to reduce their frustration and improve user experience.

Fine tuning of application code to ensure consistent behavior across platform types and network providers requires the knowledge of their performance. The accurate measurement and characterization of cellular data networks, needed for application code optimization, has been difficult, largely due to the number of service providers, diversity of the fundamental protocols, and the proprietary nature of network architectures. There have been several studies of Global System for Mobile Communications (GSM) networks [7], [8], [9]. There have also been several studies that focused primarily on Code Division Multiple Access (CDMA) technology pioneered by Qualcomm [10], [11]. These studies have been primarily focused on the needs of network infrastructure optimization and provisioning and provide little insight into network performance from the user perspective.

There has also been some work on the characterization of CDMA networks and Evolution-Data Optimized (EVDO) technology in particular. While Claypool's work provides important insight into the suitability of the EVDO technology for streaming and interactive applications, it does not account for variation between different mobile devices or network setting of different providers [12].

Broadband Reports[2] has implemented a simple tool to estimate wireless network performance on individual mobile devices. The interface on their web site permits a user to select a packet size varying between 5KB and 600KB and then send a packet to estimate latency and downstream bandwidth. An inadequacy of this approach is that it only performs a single HTTP-based test where the user connects to the website and downloads a single packet through the phones built-in web browser. There are a number of limitations with the Broadband Reports test. First, measuring network performance using HTTP packets does not give a good view of application traffic performance due to large HTTP overhead with respect to usually small data packet traffic [16]. Second, the Broadband Reports test relies on JAVA Script, a functionality not enabled on most mobile devices for security reasons. Finally, the Broadband Reports test is not adaptable to application developer needs, a key advantage of MIST.

III. System Architecture

The Mobile Internet Services Test (MIST) platform enables mobile application developers to characterize the data delivery performance of cellular data networks as delivered to

[1] http://www.airplaynetworks.com/

[2] http://www.broadbandreports.com/

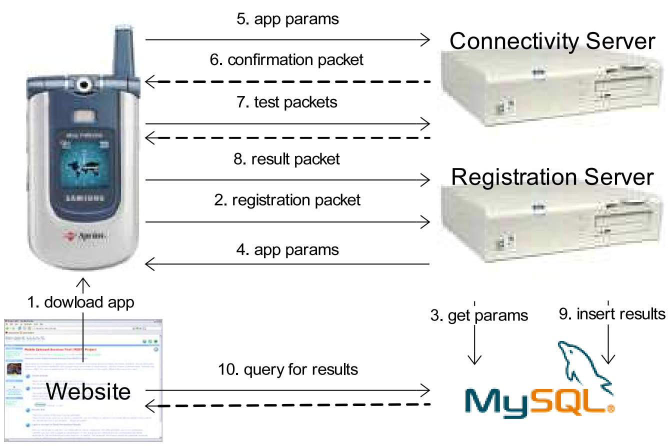

Fig. 1. The MIST architecture and communication protocol.

mobile devices. Mobile application developers need to know a range of network performance characteristics, including latency, jitter, throughput, and network timeout delays. Of particular importance is how these values vary in different network provider, network technology, and mobile device combinations. To enable data collection from users across the range of technologies available, MIST offers a lightweight measurement architecture that can be deployed on a wide range of JAVA-enabled mobile devices.

The MIST measurement platform is designed with scalability, accuracy, and ease of deployment as primary goals. We address challenges of measurement system scalability in three ways. We minimize the measurement application footprint to conserve space and improve performance on challenged mobile platforms, *i.e.*, cell phones. We implement a lightweight server architecture to allow a large number of mobile devices to perform tests concurrently using a single server. Finally, we design a communication protocol that can distribute measurement activity to accommodate a very large number of clients and maintain geographically accurate measurement. In this section, we describe how the MIST architecture achieves its implementation goals and describe the design tradeoffs.

The diagram in Figure 1 illustrates the major components of our system and the steps in MIST's communication protocol. The measurement application operating on the mobile device first collects information about the mobile device, service provider, and test location. The registration server communicates with the phone application to gather user input data and accumulate the results after a test run has been completed. The registration server also manages a connection to the web site database where the user data, test results, and application configuration information are stored. The web site enables users to maintain accounts to review the tests run by their devices. The web site also provides tools for data visualization. Finally, the connectivity server is responsible for communicating with the mobile device to measure network performance.

Both the connectivity and registration servers are implemented in JAVA 1.5, with a particular effort toward making the implementations compatible with older versions of the JAVA Virtual Machine (JVM). Geographically distributed connectivity servers can provide regional network connectivity, allowing a mobile device to connect to the nearest server in order to provide the most accurate cellular data network measurements. Our goal is to use location-aware connectivity server assignment in order to minimize the delay and jitter effects caused by the wired portion of the end-to-end path, thereby making the measurements of the wireless route portion more accurate. Additionally, multiple connectivity servers reduce the load on any one server, thereby eliminating any delay due to the server handling too many simultaneous measurements.

Our phone application is implemented in the JAVA Mobile Environment (J2ME) with support for the Mobile Information Device Profile (MIDP) 2.0 standard[3]. The main difference between MIDP 2.0 and its predecessor, MIDP 1.0, is that MIDP 2.0 offers UDP and TCP sockets in addition to MIDP 1.0's HTTP connector. The HTTP connector requires that all data be wrapped in HTTP headers, which adds significant overhead, especially to small packets, thus reducing the accuracy of latency measurement. Additionally, since MIDP 2.0 and TCP are envisioned as the standard for future mobile applications, we chose them as the basis for our platform.

We have placed an emphasis on keeping the server implementations lightweight, contributing to the servers' responsiveness and increasing measurement accuracy. Our lightweight design also promotes system scalability, as more mobile devices can connect to the server simultaneously before overloading server components. To achieve the above goals, we have moved most of the measurement functionality onto the mobile application.

Each MIST measurement run is initiated by the mobile application on the mobile device. The application registers with the registration server and obtains a set of test parameters. The test parameters specify the test identifier; the number and types of tests to be executed; the number and size of packets for each test; and the transmission interval information. The application then transmits the parameters to the connectivity server to assure appropriate replies and begins the tests. During the tests, the specified type and number of test packets are transmitted between the application and the connectivity server. The test packets are timestamped four times: during client transmission, server reception, server transmission, and finally client reception. Following a synchronization algorithm presented by Mills [17], we calculate the offset between the client and server clocks using:

$$\text{offset} = \frac{server\_rx - client\_tx + server\_tx - client\_rx}{2}$$

From the calculated offset, MIST can adjust timestamp values as if the mobile device and server clocks were synchronized during the test, which in turn allows us to calculate accurate uplink and downlink latencies. The test packet results are kept on the phone, until a test run is complete, at which time they are reported to the registration server. This method allows the connectivity server to keep no state

[3]http://jcp.org/aboutJAVA/communityprocess/final/jsr118/

per client, reducing its memory requirements. Additionally, the registration server can receive results in bulk, minimizing its communication overhead and allowing bulk inserts to the application database.

To promote reuse and adaptability of our application, we allow application developers to customize MIST's tests and parameters. MIST's configurability is a major advantage over applications like the Broadband Reports mobile test which follow the one-test-fits-all philosophy. Application developers can create custom test runs, where they can adjust parameters like test duration, number of packets, packet size, as well as metrics collected, like latency, jitter, throughput, and timeout interval delay. We believe that test customization will lend itself to MIST's adoptability, allowing users to download the MIST application only once, not every time test parameters change. MIST can also store old test runs along with their parameters, assuring that old test data can always be retrieved and interpreted correctly, providing application developers with more than just the most recent view of the network.

In the future we envision a number of extensions to the MIST architecture bringing additional utility to the mobile application developer. MIST uses separate registration and connectivity servers to allow centralized test configuration and data storage repository, but distributed test execution. We envision using a number of connectivity servers, with each mobile device running its test against the geographically closest server. This paradigm is expected to give network measurements representative of modern network services. To reduce the impact of network and reduce hot spots, Akamai[4], for example, has developed solutions to provide for the geographic distribution of media content. We believe that future data services will be deployed geographically and want MIST to provide the option of geographically aware performance measurement to mobile application developers.

Another architecture improvement is the use of MIST as a traffic trace generator and emulator. Mobile application developers may want to test their applications against the real delays of cellular data networks. MIST can be used to record a custom traffic trace, which can then be replayed offline to test application performance and stability. MIST could also be used as a traffic emulator, allowing mobile application developers to test the performance of custom traffic traces on a diverse set of mobile devices and networks.

IV. Data Analysis

To demonstrate MIST's capabilities of providing insight into the performance of cellular data services as experienced by a particular device, we present the analysis of a small data set collected in the Santa Barbara, CA area. Our primary goal in this analysis is to show that MIST can obtain useful data, which can be analyzed to characterize the performance of data networks as experienced by individual wireless devices. Our second goal, was to identify non-obvious network behavior, which, if unexplainable with currently collected data, would be the basis for expanding MISTs collection capabilities.

[4]http://www.akamai.com/

The data presented in the following sections has been collected using a Samsung MM-A700 cellular phone on the Sprint PCS network. We ran the MIST tests on the UCSB campus under stationary conditions and on a nearby section of US Route 101 at 60 mph. The Sprint PCS cellular data network in the testing area is a CDMA2000 network using the 1xRTT data communication protocol[5]. Using traceroute, we were able to locate the first hop router in Los Angeles, CA.

While we were successful at developing the MIST platform, using it to collect data still proved to be a significant challenge. In particular, we discovered a number of wireless device shortcomings and network limitations. The majority of cellular devices we had access to did not have a data plan we could use, and another large portion were only able to run JAVA under MIDP 1.0. The remaining three were on Verizon, Cingular, and Sprint PCS networks, the first two of which place restrictions on the downloading of unregistered mobile applications. In the future, we expect that the majority of cellular data network users will have unlimited data plans and MIDP 2.0 devices. Further, we hope to either be able to register our application with cellular service providers, or that the providers will eliminate their download restrictions. Therefore, in the remainder of this section, we only present a sample set of results. This data set, nevertheless, show the capabilities of our system to provide the insight necessary to characterize cellular data networks.

A. *MIST Data Collection*

The MIST tests we ran consisted of three types of tests: latency, throughput, and timeout interval. A summary of our tests and the associated parameters are given in Table II. The latency tests measured the uplink and downlink delay as effected by packet size. Two of our latency tests were configured to send packets at one second intervals. We performed a third experiment as a control, where the transmission interval was based on a Zipf distribution of 20 values with a mean of one second. Zipf distributions have been shown to accurately model user reactions over time [18]. We used a Zipf distribution as a representative non-constant distribution to show periodicity of transmissions had no effect on network response time. In selecting the mean value, we chose one second since it reasonably represents an active user's reaction time. As a result of our testing, we found that if the interval fell below 200 msec, packets were queued on the user device.

We measured throughput values based on 10,000 byte packets.We found this value to be large enough to measure throughput bandwidth, but small enough such that it did not overflow mobile device memory. The 10,000 byte packet is representative of application sizes users download, but also allow developers to estimate the download time of streams while keeping the throughput test duration small. We prefer to measure throughput directly, by measuring bulk data transmission time, rather than estimating network performance using packet pair throughput estimation.

[5]http://kb.pcsintel.com/

TABLE II
MEASUREMENT TEST CONFIGURATIONS

Test Type	Upsize (bytes)	Downsize (bytes)	Test Length (packets)	Interval (seconds)	Zipf Distribution Alpha	Zipf Distribution Length
Latency	500	500	20	0.5	NA	NA
Latency	1000	1000	20	0.5	NA	NA
Latency	1000	1000	20	variable	1	20
Throughput	10,000	0	1	NA	NA	NA
Throughput	0	10,000	1	NA	NA	NA
Timeout	500	500	20	1-20, 1 sec. incr.	NA	NA

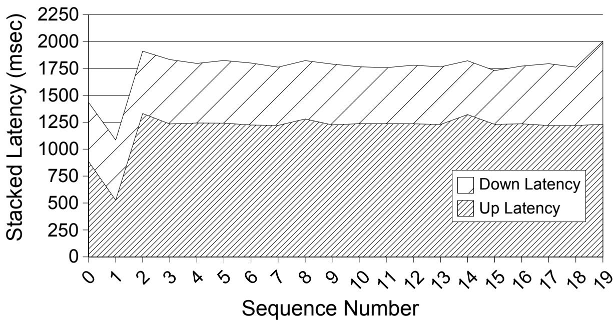

Fig. 2. Round trip time, downlink, and uplink latency results.

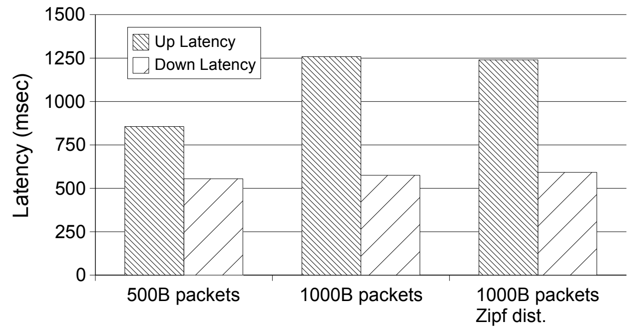

Fig. 3. Uplink and downlink latency as effected by packet size and send interval.

Finally, we measured component timeout intervals. Mobile device manufacturers extend battery lifetime by shutting down power-hungry components when they are not being used. In addition, cellular data network providers allocate network resources to mobile devices through flow admission protocols. These resources are freed when the mobile radio is not in use. We measure any timeout intervals by transmitting small packets at increasing inter-transmission delays. Our timeout test measured packet latency associated with each interval, allowing us to estimate the performance cost of bringing a timed out component back or forcing resources to be reallocated. A summary of our tests is presented in Table II.

While we have found that the MIST tests using the nominal parameters presented in Table II are insightful, a developer might want to adjust them to more closely represent mobile application behavior. While our focus is the broad investigation of network characteristics, developers may be more interested in network performance for a specific load scenario. Our system allows for simple parameter adjustment though changes to the registration server test configuration database.

B. Latency Analysis

We introduce our latency analysis with the following scenario. Twenty test packets were sent between the mobile device application and the connectivity server. Each transmission was timestamped four times allowing us to separate the uplink and downlink latencies. The data is shown in Figure 2.

The graph uplink and downlink latencies are stacked to show the Round Trip Time (RTT) as the topmost line. These three metrics are of primary importance to application developers designing real-time or delay-sensitive applications. They need to know what delay the user will experience and the effect of packet size.

To answer these questions, MIST measures latency as effected by changes in packets size. The averaged results of three twenty packet test-runs are presented in Figure 3. We varied packet size in the first two measurement configuration, from 500 to 1000 bytes. In the third measurement, we used 1000 byte packets transmitted according to a Zipf distribution. We used this last test as a control to show that our latency measurements are not affected by any periodic anomaly. We used a Zipf distribution as a representative non-constant distribution to show periodicity of transmissions does not affect our results.

The latency measurements graphed in Figure 3 show that uplink latency dominates round trip time. The effect is magnified when the packet size is doubled. We believe the difference between latencies is due to differences in CDMA spreading codes. Third Generation (3G) technology requirements, guiding the CDMA2000 design, indicate that more bandwidth needs to be allocated on the downlink to account for the disproportionately high download traffic of Web browsing [19]. While this traffic duality may not represent application data traffic [16], our traffic is nevertheless affected by the 3G requirements. We also observe, that varying of packet size from 500 bytes to 1000 bytes has little effect on the downlink latency. We have noticed that 1000 byte packets often require two transmissions on the uplink, while the downlink transfer of 1000 byte packets does not result in fragmentation.

In addition to the differences between uplink and downlink latencies, we have also found there to be large variations between individual packet latencies. Table III shows the minimum, maximum, and average values for the latency

TABLE III
LATENCY STATISTICS (MSEC)

	500B				1000B				1000B Zipf			
	W/out Mobility		W/ Mobility		W/out Mobility		W/ Mobility		W/out Mobility		W/ Mobility	
	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down
Min.	330	484	306	315	354	499	395	459	550	490	444	466
Max.	3379	1485	985	1460	2513	1205	2561	3043	1644	1330	1833	1989
Avg.	856	555	775	665	1258	575	1310	698	1239	592	1296	629

experiments. These values range from 306 milliseconds to over 3 seconds. The large variation in latency values makes it difficult to both characterize and predict network latency behavior. Application developers need to be aware that there is no simple answer to what delay their users will experience, and need to write their software with quite a bit of tolerance for delay.

We have also investigated the effects of mobility on latency. Figure 4 presents a histogram of uplink latencies of the 500 byte and 1000 byte packets collected for the 93103 ZIP code between 12:00pm and 1:00pm. By fixing the geographic area and time of day, we hoped to perform the experiments under similar signal strength and network load. The results presented in Figure 4 show how the accumulated data from three test runs performed under stationary conditions and 60 mph highway mobility. These results are augmented in Table III with average, minimum, and maximum values. While the values in Table III show that the average, minimum, and maximum values are similar for both the stationary and mobile measurements, the latency histogram in Figure 4 shows larger variability for the mobile tests. The data collected in the stationary tests centers around values close to the average for both the uplink and downlink measurements. The mobile test data, on the other hand is more evenly distributed, making latency characterization in mobile environments even more challenging.

The observed differences in uplink and downlink latency behavior are important for mobile application developers, who need to adjust for varying delays of application packets depending on the link transmission direction and packet size. For latency sensitive applications, like streaming media, pervasive gaming, and mobile commerce, making smart traffic shaping decisions to account for varying latencies can greatly increase application usability.

C. *Throughput Analysis*

To assure uniform performance of multimedia network applications, application developers need to combine latency and jitter network characteristics with measurements of network throughput. We have measured uplink and downlink throughput by timing the upload and download of a 10,000 byte packet. The 10,000 byte packet is representative of application sizes users download, and allows accurate bandwidth measurement while keeping the test duration small. To more accurately measure network, as well as mobile device performance, we chose to measure throughput of a bulk data transfer, rather than by using packet pair bandwidth estimation. Packet pair bandwidth estimation estimates network throughput by identifying bottleneck performance from the temporal spread of a small packet pair [12]. While packet pair estimations can work well in some settings, they do not stress the network allocation policies or mobile hardware and can be overly optimistic.

TABLE IV
THROUGHPUT STATISTICS (KBPS)

	Uplink	Downlink
Min.	0.79	1.82
Max.	4.47	11.17
Avg.	2.05	4.67

Table IV presents the minimum, maximum, and average throughput values taken in the 93106 ZIP code over a period of 10 days. We observe that the average downlink throughput is over twice the average uplink throughput. We believe this large difference is the result of CDMA coding allocations made by the service providers. Another interesting observation that can be made from Table IV is the large difference between the average and the maximum throughput values. While the uplink average throughput is fairly evenly placed between the minimum and maximum, the downlink direction has a disproportionately larger maximum throughput value as compared to its average value.

The histogram of throughput measurements, presented in Figure 5, shows another interesting performance feature. The throughput values in both uplink and downlink directions seem be concentrated around two local maxima for each direction. The bimodal throughput behavior shown by our tests is likely the result of adaptive spreading code allocation in response to link quality in CDMA networks [13].

The throughput measurements collected during this study indicate that throughput in the 93106 ZIP code is not satisfactory for streaming applications. The Sprint PCS network deployed in the Santa Barbara, CA does not take advantage of the latest CDMA technologies like EVDO, placing a limit on handset performance. Mobile application developers can adjust to these realities by either drastically reducing the encoding rate of streaming media, or preferring to transfer the media to the mobile device before initiating playback.

D. *Timeout Analysis*

Our final test investigated mobile device dormancy and timeout periods. We sent small packets at intervals consecutively increasing by one second. The results of our test are plotted in Figure 6. The graph shows uplink transmission latency as effected by the increasing inter-packet transmission

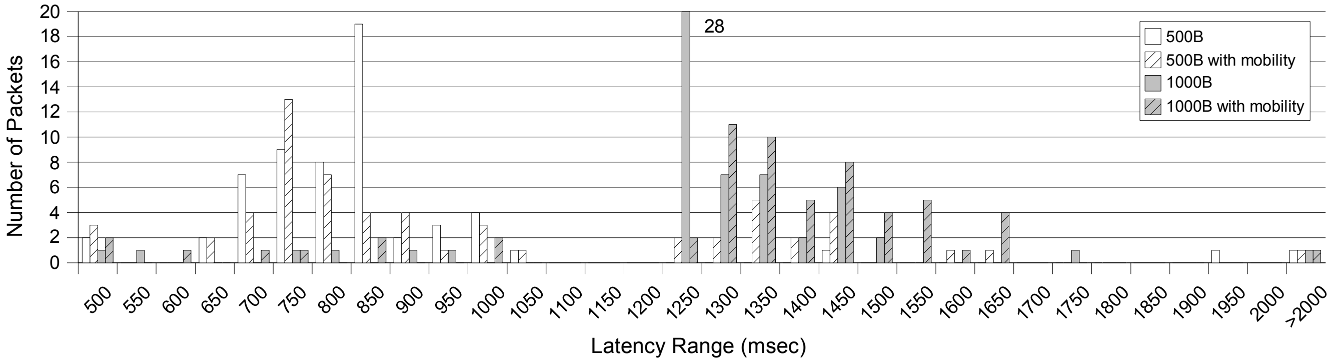

Fig. 4. Uplink latency histogram showing the effects of packet sizes and mobility on latency distribution.

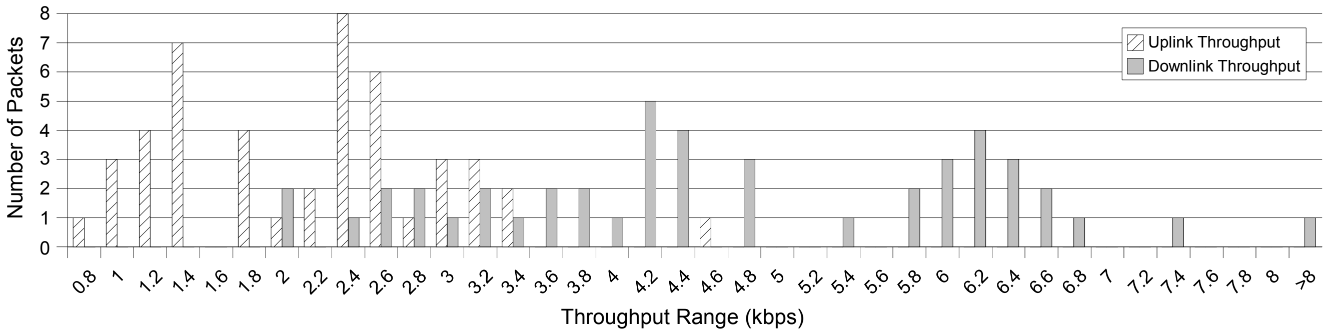

Fig. 5. Uplink throughput histogram showing the effect of packet size on throughput.

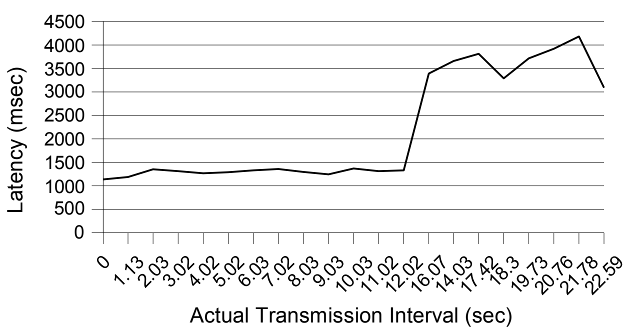

Fig. 6. Latency as effected by transmission interval.

interval. Packet latency oscillates between 1 and 1.5 seconds until the transmission interval reaches 12 seconds. Latency values for packet transmission intervals larger than 12 seconds have much higher latencies, on average, about 3.5 seconds. We believe this result is due to the radio on our mobile device timing out and going dormant. In our future work, we plan to investigate whether the timeout values vary across mobile devices and manufacturers.

A mobile application developer can adapt their application to these findings by either using periodic keep-alive messages to prevent timeouts, or planning for longer delays after periods of user inactivity. Adjusting to the mobile device and network timeout intervals correctly can have a major impact on the wait times perceived by users, thus improving mobile application performance.

V. Future Work

During our data collection we identified a number of measurement directions for future work. First, there are numerous outstanding questions as to the behavior of cellular data networks from the point-of-view of mobile application developers. While we were able perform tests and obtain data for the Sprint PCS network using one mobile device, to make the study exhaustive we need to perform similar tests using other service provider/mobile device combinations. Results from these tests would allow application developers to make their applications robust to varying latency, packet size, and timeout values. Additionally, there are questions of scale. Is the performance of the network impacted by the number of users accessing the server? Does load on the network created by dense usage areas affect network performance at the MAC layer?

Second, we would like to improve MIST with the addition of geographically distributed measurement, as well as trace collection and emulation features. We believe that these functions will allow mobile developers to, not only characterize cellular data networks, but actively field test their traffic loads before application deployments. The improved understanding of the network is likely to positively impact application usability and adoption rate, ultimately resulting in value chains benefiting application developers, network providers, and customers.

VI. Conclusions

We have presented MIST, a first-of-its-kind cellular data network measurement platform, focusing on the needs of mobile application developers, rather then network infrastructure optimization and provisioning. We have analyzed sample data collected with our tool, showing large variability in cellular data network characteristics. We have concluded, by motivating the need for further measurement efforts from the point-of-view of mobile application developers and network researchers.

References

[1] J. A. Harmer, "Mobile multimedia services," *BT Technology Journal*, vol. 21, no. 3, pp. 169–180, July 2003.

[2] H. Verkasalo, "Empirical observations on the emergence of mobile multimedia services and applications in the U.S. and Europe," in *Proceedings of the 5th international conference on Mobile and ubiquitous multimedia (MUM)*, December 2006.

[3] J. Karvonen and J. Warsta, "Mobile multimedia services development: value chain perspective," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia (MUM)*, October 2004.

[4] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jaalinoja, M. Korkala, J. Koskela, P. Kyllonen, and O. Salo, "Mobile-D: an agile approach for mobile application development," in *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications (OOPSLA)*, October 2004.

[5] G. Lawton, "Moving JAVA into mobile phones," *IEEE Computer*, vol. 35, no. 6, pp. 17–20, June 2002.

[6] M. D. Walker, M. Nilsson, T. Jebb, and R. Turnbull, "Mobile video-streaming," *BT Technology Journal*, vol. 21, no. 3, pp. 192–202, July 2003.

[7] R. Chakravorty and I. Pratt, "WWW performance over GPRS," in *IEEE Mobile Wireless Communication Networks*, September 2002.

[8] R. Chakravorty, J. Chesterfield, P. Rodriguez, and S. Banerjee, "Measurement approaches to evaluate performance optimizations over wide-area wireless networks," in *Fifth Passive and Active Measurement (PAM) Workshop*, April 2004.

[9] R. Chakravorty, "Performance issues with general packet radio service," *Journal on Communications and Networks*, vol. 4, no. 2, pp. 266–281, June 2002.

[10] C. Williamson, E. Halepovic, H. Sun, and Y. Wu, "Characterization of CDMA2000 cellular data network traffic," in *30th Annual IEEE Conference on Local Computer Networks*, November 2005.

[11] E. Halepovic and C. Williamson, "Characterizing and modeling user mobility in a cellular data network," in *Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN)*, October 2005.

[12] M. Claypool, R. Kinicki, W. Lee, M. Li, and G. Ratner, "Characterization by measurement of a CDMA 1x EVDO network," in *Proceedings of the 2nd annual international workshop on Wireless internet (WICON)*, August 2006.

[13] M. Ghaderi, A. Sridharan, H. Zang, D. Towsley, and R. Cruz, "TCP-aware resource allocation in CDMA networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom)*, September 2006.

[14] Z. Zhuang, T.-Y. Chang, R. Sivakumar, and A. Velayutham, "A 3: application-aware acceleration for wireless data networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom)*, September 2006.

[15] H. Verkasalo, "Handset-based monitoring of mobile customer behavior," Master's thesis, Helsinki University of Technology, 2005.

[16] I. C. Y. Ma and J. Irvine, "Characteristics of WAP traffic," *Wireless Networks*, vol. 10, no. 1, pp. 71–81, January 2004.

[17] D. Mills, "Improved algorithms for synchronizing computer network clocks," *IEEE Transactions Networks*, vol. 3, no. 3, pp. 245–254, June 1995.

[18] A. Dan and D. Sitaram, "An online video placement policy based on bandwidth to space ratio (BSR)," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, May 1995.

[19] M. W. Oliphant, "The mobile phone meets the Internet," *IEEE Spectrum*, vol. 36, no. 8, pp. 20–28, August 1999.