# Multi-Layer Analysis of Web Browsing Performance for Wireless PDAs

Adesola Omotayo Carey Williamson
Department of Computer Science, University of Calgary
Email: {omotayo,carey}@cpsc.ucalgary.ca

## Abstract

*This paper studies the user-level and network-level performance for a mobile client browsing the Web using a wireless Personal Digital Assistant (PDA). A wireless network analyzer is used to collect traces of the TCP/IP network traffic generated to and from the wireless PDA. Multi-layer protocol analysis is used to assess the Web browsing performance achieved, and identify protocol performance problems. The analyses focus on server response time, document size, HTTP transfer time, TCP performance, and wireless channel quality. The results from our experiments provide insight into strategies that could be used by Web content providers, Web servers, HTTP, and TCP to improve Web browsing performance for mobile users.*

**Keywords:** Wireless Web Performance, Network Traffic Measurement, IEEE 802.11b WLAN

## 1 Introduction

Today's mobile computing devices offer users an unprecedented level of connectivity and convenience. Many mobile users rely on their wireless laptop, Personal Digital Assistant (PDA), or cell phone for Internet access, with Web browsing as one of the primary applications. The widespread availability of WiFi hot spots (i.e., the IEEE 802.11b Wireless LAN standard) makes "anything, anytime, anywhere" information access a reality for many mobile Internet users.

Wireless Web access, however, is not without its performance problems. Wireless channel bandwidth is often limited compared to desktop wired-Internet access, and the wireless channel is typically shared amongst multiple users. In addition, the wireless channel quality can vary significantly with time, with an inherent error rate much higher than that for wired network technologies.

The wireless environment presents many challenges for higher-layer networking protocols, even for relatively simple applications such as Web browsing. The two most relevant protocols are the Transmission Control Protocol (TCP) and the Hyper-Text Transfer Protocol (HTTP). TCP is the primary transport-layer protocol in the Internet protocol stack, providing reliable data delivery for electronic mail, file transfer, and the Web [16]. Some of TCP's control algorithms are arguably optimized for bulk data transfer on a wired network; TCP performance for Web browsing, particularly in a wireless environment, can be quite unpredictable. Furthermore, the HTTP protocol used for Web downloads often leads to inefficient TCP performance [13]. HTTP protocol enhancements such as persistent connections provide significantly better performance, particularly in wireless Web environments [4, 12].

This paper presents a fine-grain analysis of Web browsing performance for a mobile user with a wireless PDA. Our analysis studies the multi-layer protocol interactions that arise when HTTP and TCP/IP operate over an IEEE 802.11b WLAN. A wireless network analyzer is used to collect TCP/IP packet traces of Web workloads generated from a mobile user with a wireless PDA. Analysis of these traces focuses on Web document size, server response time, HTTP transfer time, TCP performance, and wireless channel quality.

Our results identify several protocol-related issues that affect wireless Web browsing performance. These results provide insight into the effectiveness of possible performance enhancements that can be used by Web content providers, Web servers, HTTP, TCP, and IEEE 802.11b.

The remainder of this paper is organized as follows. Section 2 briefly discusses prior related work on WLAN performance measurements. Section 3 describes the experimental methodology for our work. Section 4 presents our measurement results, spanning from the MAC-layer to the HTTP-layer. Section 5 concludes the paper.

## 2 Related Work

There have been many studies characterizing Web workloads, from the viewpoint of clients [6, 9], servers [1, 14], and Web proxies [7, 19]. However, these studies typically consider wired network environments, rather than the wireless environment used in our work.

There are also several studies of wireless network usage in local-area, campus-area, and metropolitan-area environments [5, 10, 15, 17, 18]. These studies provide an understanding of the aggregate traffic demands from many users, using a variety of network applications. For example, Schwab and Bunt [15] use a one-week traffic trace to determine where, when, how, and how much a campus-wide wireless network is being used for applications such as Web browsing and file sharing. Similarly, a 12-week WLAN study at Stanford [18] provides a qualitative description of how users take advantage of the wireless network, and the earlier work by Tang and Baker [17] shows network usage and user mobility over a 7-week period in a metropolitan area wireless network. Results from these studies indicate how and when the networks are used, but they do not discuss application-specific protocol performance issues.

Balachandran *et al.* [5] characterize user behaviour and network performance over 3 days of the ACM SIGCOMM conference in 2001. Their results provide insight into WLAN deployment issues, such as capacity planning, and load balancing across multiple access points. However, they do not present a detailed analysis of TCP protocol effects.

Two studies that explicitly consider mobile devices are those by Adya [3] and Kunz [11]. The former discusses document popularity, content analysis, system load, and user behaviour, while the latter categorizes user behaviours by session characteristics.

Our study is unique in that it provides a fine-grain analysis of protocol performance for wireless Web browsing, identifying inefficiencies from multi-layer protocol interactions.

## 3 Experimental Methodology

### 3.1 Network Environment

Our experiments are conducted on an IEEE 802.11b WLAN in the Department of Computer Science at the University of Calgary. The experimental setup is illustrated in Figure 1. The network operates in *infrastructure mode*, with the Access Point (AP) providing access to the external Internet. The AP is a Netgear WAB 102. The wireless client is a PDA that communicates directly with the AP. These are the only devices present in the WLAN during our experiments.

The PDA is a Compaq iPAQ 3600 Pocket PC running Windows CE (version 3.09348) as the operating system and Internet Explorer as the Web browser. This device has an ARM SA1110 processor and 64 MB Flash RAM. The PDA has a Proxim wireless network interface card (NIC), and a Maximum Transmission Unit (MTU) size of 1500 bytes.

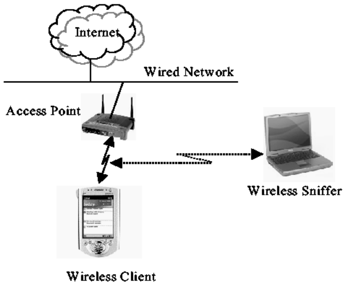


**Figure 1. Experimental Setup for Study**

### 3.2 Workload Generation

Our study uses a very simple workload: a single user with a wireless PDA browsing selected sites on the Internet. The client makes requests for Web pages by typing a URL request into the Web browser, or clicking on a hyperlink. Each request generates TCP packets, which are sent across the WLAN as encapsulated IEEE 802.11b data frames. The WLAN traffic generated during our study is captured using a wireless network analyzer.

The trace analyzed in this paper was collected over a period of 35 minutes on March 3, 2004. During this time, the user browsed Web sites offering news, yellow pages, driving directions, stock quotes, educational resources, and downloadable PDA software. Table 1 lists several of the Web sites used in this study.

**Table 1. Partial List of URLs Used in Study**

| Item | URL | Description |
|---|---|---|
| 1 | www.cpsc.ucalgary.ca | University |
| 2 | www.cemonster.com | Internet Services |
| 3 | www.cnn.com | News |
| 4 | www.forecaster.ca | Sports |
| 5 | www.cnet.com | Computers |
| 6 | www.quickdrive.com | Travel Info |
| 7 | www.handmark.com | Software |
| 8 | www.ehosting.ca | Domain Hosting |
| 9 | mobile.canada.com | Information |
| 10 | weather3.cmc.ec.gc.ca | Weather Info |

### 3.3 Trace Collection

The network traffic measurements were collected using Sniffer Pro 4.60.01, a wireless network analyzer from Sniffer Technologies. This program passively monitors and records all WLAN traffic, enabling protocol analysis at MAC, IP, TCP and HTTP layers. Packet timestamps are recorded with microsecond resolution.

The Sniffer software runs on a wireless laptop. Our laptop is a Compaq Armada E500 with a 1.0 GHz Mobile Intel Pentium III processor, 128 MB of 100 MHz RAM, 9.36 GB disk, and a Cisco Systems Aironet 350 wireless NIC. The network card is configured in promiscuous mode to record all WLAN traffic.

### 3.4 Trace Analysis

The collected packet trace is analyzed using a custom C program called `tcpconnparse`, developed in prior work [8]. The program identifies distinct TCP connections in the trace, reporting the number of TCP packets and data bytes transferred on each connection, as well as the time duration of each connection.

The trace analysis software was extended to handle multiple HTTP transfers on a single TCP connection, as happens on persistent connections [13]. The program was also modified to interpret a TCP RST (reset) packet as equivalent to a TCP FIN packet, since it is known that the Internet Explorer browser often terminates TCP connections with a RST rather than the proper FIN [2]. The customized program groups packets into TCP connections, calculates the number of TCP-layer and MAC-layer retransmissions for each connection, computes the HTTP response time, and calculates the server's response size for each client request.

### 3.5 Performance Metrics

Previous work on Internet browsing for mobile clients focuses mostly on user behavior and content analysis. However, our focus is on the user-perceived Web browsing performance, from a measurement viewpoint on the client's WLAN. The metrics used are:

- *TCP Connection Duration*. This metric expresses the elapsed time for each TCP connection. It is the time difference between the first SYN packet of a TCP connection and the last FIN (or RST) packet of the connection.

- *TCP Connection Type*. There are two types of TCP connections observed in the trace, namely persistent and non-persistent. A persistent connection has multiple HTTP requests and responses on the same TCP connection. A non-persistent connection has only one HTTP request/response interaction.

- *Average TCP Connection Throughput*. Throughput is a measure of the average data rate in Kilobits per second (Kbps) achieved by the mobile client during a TCP connection. This value depends on the HTTP response size from the Web server, the server response time, and the network bandwidth.

- *Server Response Time*. This metric expresses the elapsed time between a client's HTTP "GET" request and the first packet of the server's response. This time includes server delay and network round-trip delay.

- *Web Object Size*. This metric indicates the size in bytes of the Web objects downloaded from the server. In general, larger objects take longer to download.

- *HTTP Transfer Time*. The HTTP transfer time is the elapsed time from a client's HTTP GET request to the receipt and acknowledgement of all of the HTTP response data received from the server. This time reflects the efficiency of data transfer across the WLAN.

- *Errors*. Three types of errors are investigated in this study. The MAC-layer error rate indicates Cyclic Redundancy Check (CRC) errors and MAC-layer retransmissions as a percentage of the total data packets seen in the trace. These error rates provide information about the quality of the wireless channel. The TCP-layer error rate expresses TCP packet retransmissions relative to the total packets sent. The user-level error rate expresses the percentage of connection timeouts (aborts) relative to the total number of HTTP requests made by the mobile client.

### 3.6 Data Validation

Several sanity checks were conducted to ensure that the packet trace collected from our experiments was clean, consistent, and complete. For example, all UDP and ICMP packets were removed from the trace, leaving only TCP packets for analysis, and TCP port numbers were checked to ensure that no wrap around occurred during the trace.

Several anomalies were noticed in the data. The first anomaly, as mentioned previously, was the predominant use of TCP RST to terminate a connection rather than a TCP FIN. Only 1.8% of the TCP connections had the expected FIN handshake, while 96.5% used the RST (the remainder were unsuccessful connections). The second anomaly arises from the fact that the trace data collected by the WLAN analyzer may not reflect exactly the same view of the wireless network as the client. For example, this anomaly can manifest itself as successful TCP connection handshakes for which the client's SYN packet was not seen, or MAC-layer retransmissions of frames that were successfully received the first time. This anomaly affects less than 1% of the observed connections.

# 4 Measurement Results

## 4.1 Overview

Table 2 provides a statistical summary of the network trace collected during our experiment. The Web browsing session lasted just over 35 minutes, with 394 successful TCP connections observed. Both persistent (13%) and non-persistent (87%) connections were seen.

**Table 2. Statistical Summary of WLAN Trace**

| Item | Value |
|---|---|
| Trace Duration (min:sec) | 35:33.212 |
| Total TCP Packets | 13,705 |
| Total Data Bytes | 7,216,491 |
| Total TCP Connections | 398 |
| Successful TCP Connections | 394 |

The statistics show that the network is lightly loaded. The average (user-level) data rate for the entire trace is about 25 Kbps (6.4 pkt/sec), with a peak transfer rate of 1.2 Mbps (180 pkt/sec) on the 11 Mbps WLAN.

Figure 2 presents a graphical overview of the network traffic activity during the trace. The solid vertical lines in the graph show the number of TCP packets transmitted on the WLAN in each 1 second interval of the trace, while the lower horizontal dashed line shows the number of simultaneously active TCP connections from the client.

Network usage is clearly bursty, as is typical of Web browsing activity. The browser supports parallel TCP connections, typically with 3-4 connections active at a time, though as many as 10 were observed at one point in the trace. The default socket buffer size was 32 KB for each connection. A total of 56 different IP addresses were seen in the trace, since many commercial Web sites use server clusters, and many Web pages contain advertising banners. About 52% of the TCP packets in the trace were transmitted by the client PDA, which suggests that the TCP implementation in Windows CE does per-packet acknowledgements, rather than the usual TCP Delayed ACKs [16]. The use of Delayed ACKs or some form of ACK consolidation would economize on wireless network usage, and conserve battery power for the wireless device.

Two idle periods are also evident in the trace. These outages occurred when the PDA was rebooted and then reassociated with the AP in the WLAN.

The following subsections present detailed analyses of the trace at different layers of the protocol stack, working top-down from the application layer.

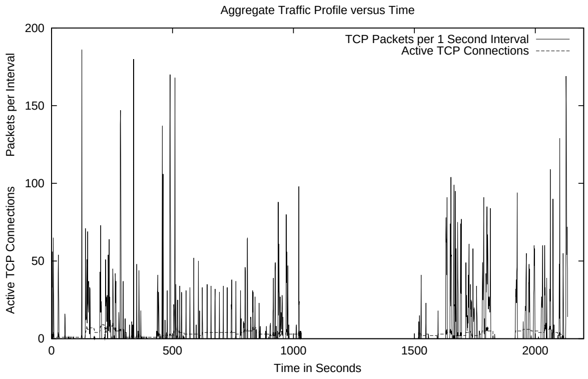


**Figure 2. Time Series Representation of Trace**

## 4.2 HTTP-Layer Analysis

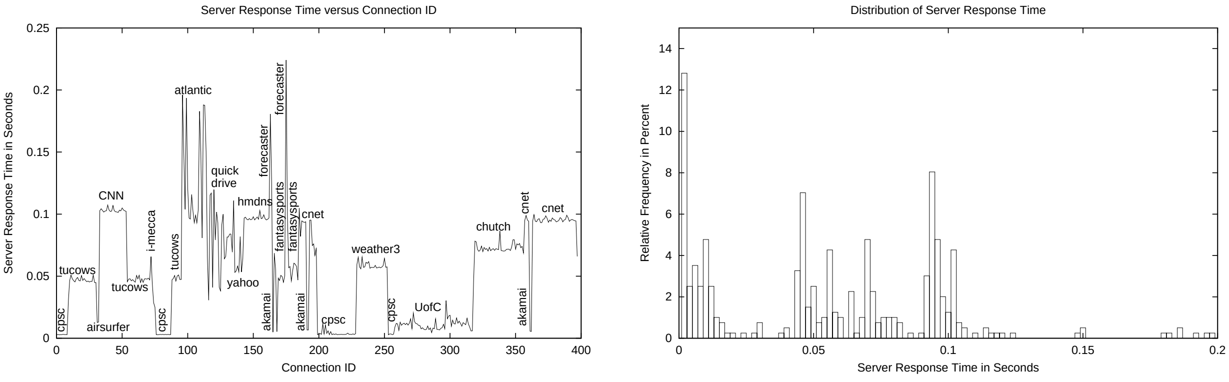
### 4.2.1 Server Response Time

The first analysis studies how quickly Web servers respond to client requests. Figure 3 presents these results. Recall that the server response time is the elapsed time between the mobile client's request and the Web server's first TCP packet (ACK or data) in response to that request.

Figure 3(a) illustrates the server response times experienced by the client. The horizontal axis represents TCP connections in the temporal order that they occurred. The distinct plateaus in this plot represent successive TCP connections to the same Web site during the Web browsing session. Several such sites are indicated by name on the graph.

The response time analysis shows that the server response time for a given Web site is consistent over a time span of several minutes. This observation suggests that the network round-trip time (RTT) dominates the response latency, and that queueing delays at the server or in the network are negligible. One implication of this observation is that caching per-destination state information (e.g., RTT, cwnd) from one recent TCP connection to the next might be effective. While some versions of Linux use this technique to mitigate TCP slow start effects, our trace analysis suggests that Windows CE does *not* do any TCP connection state caching. This optimization could be useful in improving wireless Web browsing performance.

Several spikes are evident in Figure 3(a) as well. More detailed analysis shows that these spikes correspond to two Web sites. One is a sports forecasting Web site in Canada with a long routing path from the University of Calgary. The other is a domain hosting and VoIP service provider in the USA, which is probably a very busy server. Our results indicate that Web server response times are generally consistent, but can vary for some Web sites, depending on server and network load.



(a) Response Time by Connection ID

(b) Response Time Distribution

**Figure 3. Server Response Time Results**

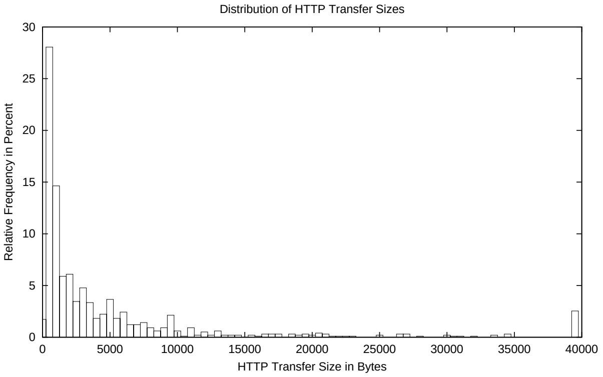


**Figure 4. Distribution of Web Object Sizes**

Figure 3(b) shows the distribution of server response times observed. The peaks in this distribution correspond to the plateaus in Figure 3(a). Most of the server response times are less than 200 milliseconds.

### 4.2.2 Web Object Sizes

The next analysis studies the sizes of the objects downloaded by the mobile user. The relationship between document type and size is also studied.

A total of 967 Web objects were downloaded during the trace. The smallest object size observed was 35 bytes, while the largest was 530 KB. The median object size was 1,549 bytes, and the mean was 6,956 bytes. The Web objects are typically small, though a few larger ones skew the mean of the distribution. These observations are consistent with those for general Web workloads [1], although the Web content analyzed here is targeted especially for mobile users.

Figure 4 shows the distribution of Web object sizes. Most (90%) of the Web objects downloaded by the mobile client are smaller than 10 KB, and only 2.5% exceed 40 KB. This observation suggests that Web proxy caching of content from wireless portals could effectively reduce network latency for mobile Web users. In fact, several graphics for the `cnet` Web site were served by an Akamai CDN (TCP connections 362 and 363) in Figure 3.

A separate analysis studied Web object types. The three most prevalent file types downloaded are GIF, JPG, and HTML. One obvious inference from this is that many Web pages include embedded objects. In particular, graphic files are widely used on the Web. The least frequently observed image file type is PNG, a patent-free competitor to GIF that offers several improvements. However, few Web browsers support PNG, and it is rarely used on Web sites.

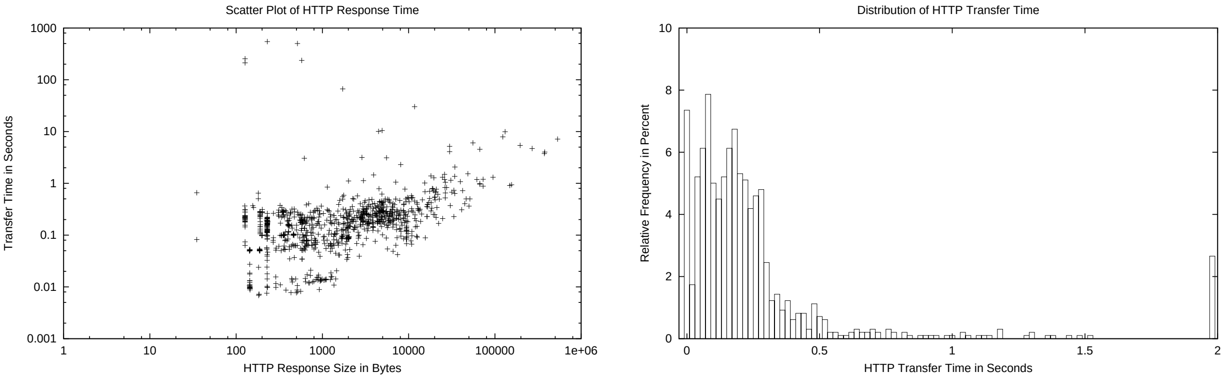
The largest objects transferred were executable files, containing downloadable software for PDAs. Providers of such software should consider compressing their executable files into more compact file formats, since mobile devices have limited memory and wireless network bandwidth.

### 4.2.3 HTTP Transfer Time

The next analysis studies HTTP transfer time: the elapsed time from when a mobile client makes a GET request to when the client has all of the corresponding response data from the Web server (using 1 or more network packets).

Figure 5(a) shows a scatter-plot of the HTTP transfer time versus the Web object size for each transfer. The expected trend is that larger Web objects take longer to download. While this trend is present in the data, the density of points in the graph make it hard to see. There are also several outliers among the data points. A few small Web objects had excessively long[1] HTTP transfer times.

[1] Further examination of the trace shows that these transactions used persistent connections to two different popular commercial Web sites, one running GWS/2.1 Web server software on Linux and the other running AkamaiGHost on Linux. We are not yet able to explain these anomalies.



(a) Scatter Plot of HTTP Transfer Times

(b) HTTP Transfer Time Distribution

**Figure 5. HTTP Transfer Time Analysis**

Figure 5(b) shows the distribution of HTTP transfer times. In general, the HTTP transfer times are low. About 96% of the transfers complete in less than 1 second, and only 2.5% require longer than 2 seconds. Given the Web object size distribution discussed previously, a high proportion of the server's responses fit in a single TCP packet. These transfers typically complete in less than 1 second. Even large transfers complete in 10 seconds or less.

## 4.3 TCP-Layer Analysis

### 4.3.1 TCP Connection Type

The first TCP-layer analysis focuses on classifying the observed connections as either persistent (i.e., same TCP connection used for multiple HTTP transfers) or non-persistent (i.e., only a single HTTP transfer before the TCP connection is closed). Analysis of the trace shows that 53 of the TCP connections (13%) were persistent, while 341 (87%) were non-persistent. As many as 73 HTTP requests were seen per connection, though only 4% of the TCP connections sent more than 10 HTTP requests. About 65% of the 967 HTTP transfers occurred on persistent connections.

Persistent connections offer several advantages over non-persistent connections, particularly in a wireless environment [4, 12]. First, the control packet overhead of TCP handshaking can be amortized across multiple HTTP transfers. Second, multiple request-response interactions on the same TCP connections allow TCP ACKs to be piggybacked on TCP data packets, reducing the number of network packets. Third, persistent connections allow subsequent HTTP transfers to exploit TCP state information (e.g., cwnd) established by earlier HTTP transfers on the same connection, avoiding repeated TCP slow starts.

Persistent connection usage varies by Web site. Figure 6 shows the number of HTTP requests on each TCP connection, using the same horizontal axis as Figure 3(a). Some Web content providers (e.g., CNN, quickdrive, akamai, google) support persistent connections, while some (e.g., tucows, atlantic, fantasysports, weather3) apparently do not. In the latter case, the PDA browser issues many separate TCP connections, serially or in parallel, to the same Web site over a time span of several minutes, with each connection downloading one Web object.

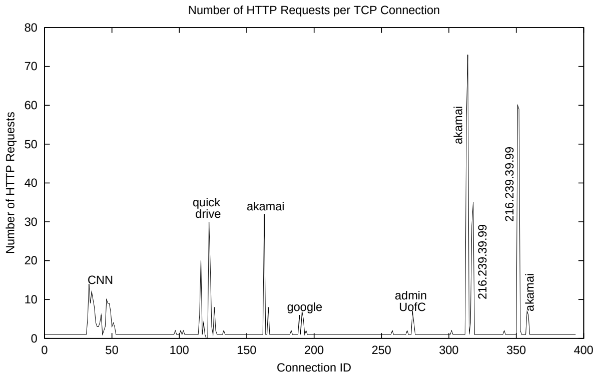


**Figure 6. HTTP Requests vs. Connection ID**

Persistent connections improve the overall network efficiency. Wireless Web browsing would be faster if persistent connections were used for all Web sites.

### 4.3.2 TCP Connection Duration

Figure 7 presents results from TCP connection-level analysis. Figure 7(a) shows the distribution of packets per connection. Figure 7(b) shows the distribution of data bytes per connection, and Figure 7(c) shows the distribution of TCP connection durations.

The main observation from Figure 7 is that most TCP connections are brief. Approximately 75% of all connections sent fewer than 20 packets, and only 6% sent more than 100 packets. The mean was 35 packets per connection; the fewest sent was 8 packets, and the most was 653 packets. About 80% of the connections sent fewer than 10 KB, and only 8% sent more than 50 KB. Approximately 75% of all connections lasted less than 1 second, and 87% lasted less than 10 seconds. This observation is consistent with the earlier observations that most TCP connections are non-persistent, and most Web object transfers are small. In other words, Web content designed for mobile users is small and usually downloads quickly. While 10% of connections last longer than 30 seconds, over 98% of the connections complete within 100 seconds.

Four TCP connections lasted 300 seconds or more. Further study of the trace reveals that the long-duration connections were to Web sites with a large value for the persistent connection timeout. These connections were thus idle for most of their lifetime, prior to the TCP connection termination. Tighter setting of the persistent connection timeout is recommended for these sites.

### 4.3.3 TCP Connection Throughput

The final TCP-layer analysis combines HTTP transfer size information with TCP connection duration to study the average throughput for TCP connections. Throughput is expressed in Kilobits per second (Kbps). Higher values reflect more efficient usage of the network.

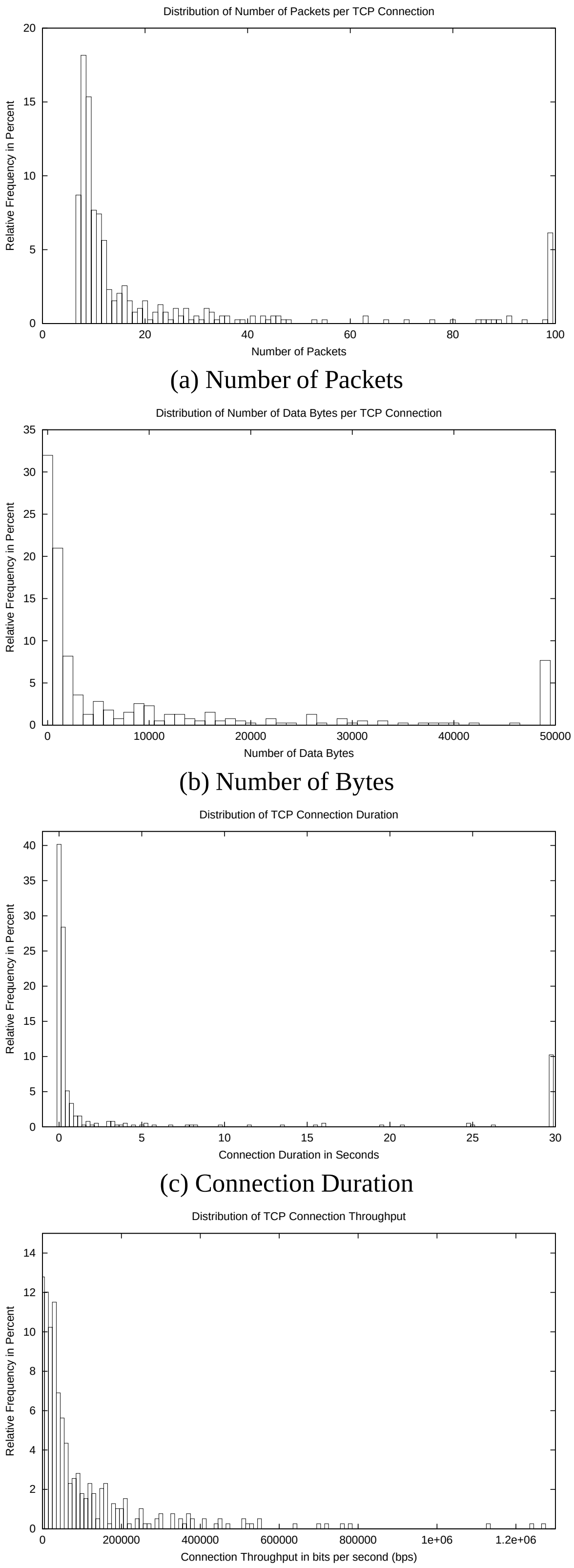
Figure 7(d) presents the results of the TCP connection throughput analysis. The results show that 95% of the connections had throughputs below 400 Kbps. This throughput is low compared to the 11 Mbps physical-layer data rate in IEEE 802.11b.

To better understand the low throughput, the mobile client's HTTP requests were studied. As noted previously, many TCP connections request a single embedded object from a Web page with many embedded objects. The cost of making multiple TCP connections to obtain the embedded objects limits the effective throughput. In other words, the low throughput is a consequence of small HTTP transfer sizes, non-negligible RTTs, TCP slow start effects, and non-persistent TCP connections. Using persistent connections would significantly improve the throughput.

Throughput varies greatly from one connection to another, even for connections to the same Web site. The variation arises from different HTTP transfer sizes for each of the non-persistent connections.

## 4.4 MAC-Layer Analysis and Error Analysis

Our final analysis focuses on the quality of the wireless channel, and any performance anomalies that arise from this. The accompanying error analysis spans multiple layers of the protocol stack.



(a) Number of Packets

(b) Number of Bytes

(c) Connection Duration

(d) Connection Throughput

**Figure 7. TCP Connection-Level Analysis**

The total number of MAC-layer retransmissions observed was 533. These retries affected 407 packets (3.0%), from 159 connections (40%). About 4% of the connections had more than 5 MAC-layer retry events. Most packets (78%) were successful on the first retry, with only 5 packets requiring more than 3 retries. The most retry attempts observed for a packet was 6. The burstiness of the MAC-layer retransmission pattern is similar to that of the overall traffic in Figure 2. In addition, the wireless analyzer reported CRC errors for 0.04% of the total packets. CRC errors are evenly distributed throughout the trace, reflecting the randomness of wireless channel errors. These observations suggest that the wireless channel quality does not have a major impact on wireless Web browsing performance in our experiment.

At the TCP layer, a total of 26 TCP data packet loss and retransmission events were observed. This represents an average loss ratio of 0.2%. We think these packet losses occurred on the Internet, prior to reaching the WLAN. The packet losses affected 12 TCP connections, with only 2 TCP connections having more than 3 packet loss events. Most losses were recovered using fast retransmit, while several incurred a TCP coarse timeout.

More detailed analysis of the trace showed many connections with superfluous duplicate ACKs, two connections with zero-window update events, and several TCP connections with out-of-order packets arriving at the WLAN. The latter phenomenon can happen if packets take different paths through the Internet. About 10 such connections were observed, from two different Web sites. The most out-of-order packets observed on a connection was 11, among 300 server packets received on that connection.

At the user layer, 1% of the attempted connections were unsuccessful: they failed during the opening handshake. A total of 96.74% of the attempted connections were successful, with opening and closing handshakes. The remaining 2.26% of connections performed the opening handshake successfully but had no TCP packets for the closing handshake. These connections are classified as "aborted connections". The anomalous connections have minimal impact on our analysis.

## 5 Summary and Conclusions

This paper presents a fine-grain analysis of Web browsing performance for a mobile user with a wireless PDA. A wireless network analyzer is used to collect TCP/IP packet traces of Web workloads generated from a mobile user with a wireless PDA. While the experimental setup and workload are simple, the detailed analysis of the traces focuses on the multi-layer protocol interactions that arise when HTTP and TCP/IP operate over an IEEE 802.11b WLAN.

Our results identify several protocol-related observations regarding wireless Web browsing performance. First, persistent HTTP connections are rarely seen in our wireless Web browsing experiments. Persistent connections offer many advantages for throughput, response time, and network efficiency. Taking advantage of these efficiencies is important in a bandwidth-limited wireless access network. Second, there is good potential for caching TCP connection state information in wireless Web browsing. This optimization is especially relevant when parallel TCP connections are used to retrieve embedded objects for a Web page. Since network path properties such as RTT are stable for several minutes, exploiting this information can improve TCP throughput and user response time. Third, content developers for wireless portals are doing a conscientious job in designing Web sites for their target audience. Web object sizes are reasonable, and object download times for users are typically 1 second or less. Nevertheless, there are still opportunities for compressed file formats, as well as more compact image types. Finally, a wireless access network such as IEEE 802.11b is really not an impediment to good user-perceived Web browsing performance. In our measurements, most anomalies observed were attributable to the Internet and its protocols, rather than to the WLAN itself.

Future work will extend this study to a much larger workload generated by a broader set of users.

## Acknowledgements

The WLAN traffic measurements in this paper were collected with the help of Guangwei Bai and Nayden Markatchev. Financial support for this work was provided by iCORE (Informatics Circle of Research Excellence) in the Province of Alberta, and by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

## References

[1] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", *IEEE/ACM Transactions on Networking*, Vol. 5, No. 5, pp. 631-645, October 1997.

[2] M. Arlitt and C. Williamson, "Analysis of TCP Reset Behaviour on the Internet", submitted for publication, November 2003.

[3] A. Adya, P. Bahl, and L. Qiu, "Analyzing the Browse Patterns of Mobile Clients", *Proceedings of the First ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, pp. 189-194, November 2001.

[4] G. Bai, K. Oladosu, and C. Williamson, "Performance Issues for Wireless Web Servers", *Proceedings of the International Workshop on Mobile, Wireless, and Ad Hoc Networks* (MWAN), Las Vegas, NV, June 2004.

[5] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan, “Characterizing User Behavior and Network Performance in a Public Wireless LAN”, *Proceedings of ACM SIGMETRICS*, Marina del Rey, CA, pp. 195-205, June 2002.

[6] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, “Changes in Web Client Access Patterns”, *World Wide Web Journal*, 1999.

[7] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web Caching and Zipf-like Distributions: Evidence and Implications”, *Proceedings of IEEE INFOCOM*, New York, NY, pp. 126-134, March 1999.

[8] R. Epsilon, J. Ke, and C. Williamson, “Analysis of ISP IP/ATM Network Traffic Measurements”, *ACM Performance Evaluation Review*, Vol. 27, No. 2, pp. 15-24, September 1999.

[9] F. Hernandez-Campos, K. Jeffay, and F. Donelson-Smith, “Tracking the Evolution of Web Traffic: 1995-2003”, *Proceedings of the 11th IEEE/ACM MASCOTS Conference*, Orlando, FL, October, 2003.

[10] D. Kotz and K. Essein, “Analysis of a Campus-Wide Wireless Network”, *Proceedings of ACM MOBICOM*, Atlanta, GA, September 2002.

[11] T. Kunz, T. Barry, X. Zhou, J. Black, and H. Mahoney, “WAP Traffic: Description and Comparison to WWW Traffic”, *Proceedings of ACM MSWiM*, August 2000.

[12] K. Oladosu, *Performance and Robustness Testing of Wireless Web Servers*, M.Sc. Thesis, Department of Computer Science, University of Calgary, August 2003.

[13] V. Padmanabhan and J. Mogul, “Improving HTTP Latency”, *Computer Networks and ISDN Systems*, Vol. 28, pp. 25-35, December 1995.

[14] V. Padmanabhan and L. Qiu, “The Content and Access Dynamics of a Busy Web Site: Findings and Implications”, *Proceedings of ACM SIGCOMM Conference*, Stockholm, Sweden, pp. 111-123, August 2000.

[15] D. Schwab and R. Bunt, “Characterising the Use of a Campus Wireless Network”, *Proceedings of IEEE INFOCOM 2004*, Hong Kong, March 2004.

[16] W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994.

[17] D. Tang and M. Baker, “Analysis of a Metropolitan-Area Wireless Network”, *Proceedings of ACM MOBICOM*, Seattle, WA, pp. 13-23, August 1999.

[18] D. Tang and M. Baker, “Analysis of a Local-Area Wireless Network”, *Proceedings of ACM MOBICOM*, Boston, MA, pp. 1-10, August 2000.

[19] A. Wolman, G. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. Levy, “On the Scale and Performance of Cooperative Web Proxy Caching”, *Proceedings of ACM SOSP*, December 1999.