USENIX Association

Proceedings of the 2002 USENIX Annual Technical Conference

Monterey, California, USA
June 10-15, 2002

Characterizing Alert and Browse Services for Mobile Clients

Atul Adya, Paramvir Bahl, Lili Qiu
Microsoft Research
1 Microsoft Way, Redmond, Washington 98052
{*adya, bahl, liliq*}*@microsoft.com*

Abstract

There is a fair amount of evidence that suggests that Internet access from wirelessly-connected mobile handheld devices is gaining popularity. However, there haven't been too many studies that have focused solely on analyzing the wireless Internet. In this paper, we study the notification and browse services provided by a large commercial web site designed specifically for users who access it via their cell-phones and PDAs. Unlike previous web studies that have analyzed browse services provided over wired networks, we focus primarily on browse and notification services provided over wireless channels. Specifically, we analyze the notification and browser traces to understand the system load, the type of content accessed, and user behavior. We discuss the implications of our findings for techniques such as multicast, query caching and optimization, and transport protocol design.

1 Introduction

Over the last decade the cellular phone industry and the World Wide Web have experienced a phenomenal growth as people around the world have embraced these technologies at a remarkable rate. Today, most major wireless service providers in the United States, Europe, and Japan offer wireless Internet services and many Internet companies provide content that has been adapted to suit the limited display, bandwidth, memory, and processing power of small devices.

Another emerging trend, related to wireless Internet, has to do with how users manage the gigantic information flow that the Internet provides. Realizing that users are being overwhelmed with information, several web content providers allow users to switch their data access model from polling and navigation to notifications or alerts. Instead of periodically browsing through the web sites for potentially useful information, an increasing number of users are adopting the model where they register for information in which they are interested. These users provide a callback address usually in the form of an email address, a cell-phone number, or a pager number, depending on their perceived importance of the information. Whenever the relevant event is triggered, the content provider sends a notification to the user. Examples of some US companies that provide such notifications include Yahoo Mobile, MSN Mobile, AOL Anywhere, and InfoSpace. All of these services allow users to subscribe to alerts for stock quotes, sports scores, lottery, horoscope, calendar events etc. If alert services becomes a popular form of user interaction with the web, it will be critical for content provider and content management companies to handle these notifications efficiently. Proper management of notifications involves understanding which types of notifications are popular, which types of devices are used by subscribers for receiving notifications, the frequency of sending these notifications on a per user basis, etc.

In this paper, we study notification and browse services provided by a large popular commercial web site that is designed specifically for US users who access it via their cell-phones and PDAs. Unlike most previous web studies, which have analyzed browsing services provided over wired networks, we focus primarily on a web server that delivers notification and browsing services over wireless channels. We analyze notification and browser traces to understand the system load, the type of content that is accessed, and user behavior. We believe that our study is important for content providers, wireless ISPs, and web site managers.

We note here that we do not study the performance of the web server subsystem or its architectural design. Instead, we use web server logs to analyze the browse and notification patterns of wireless web users.

The rest of this paper is organized as follows. In Section 2 we review previous work done in the field of web trace analysis. In Section 3, we describe the different ways in which the web site is accessed, the characteris-

tics of the data logs, and the types of analyses we carry out. We present detailed analysis of the notification and browse logs in Sections 4 and Section 5, respectively. In Section 6, we examine the degree of correlation between the usage of browse and notification services. We conclude in Section 7.

2 Related Work

There have been a number of studies on the access dynamics of web servers servicing clients over a wired network. These studies include analyses of web access traces from the perspective of proxies [7, 20, 21], browsers [6, 9], and servers [4, 16]. However, to our knowledge, all previous web workload studies have been conducted for browse services only and there are no published studies on notification services. Consequently, we believe, our analysis of notification services is the first study of its kind.

Even for the browsing services, most studies analyze web servers serving clients over wired networks. There are very limited studies on web servers serving clients over wireless channels. The study closest to ours is the one done by Kunz *et al.* [12], which analyzes network traces generated by a mobile browser application. Specifically, their paper analyzes user behavior (bytes transferred and time spent on the wireless link) based on the notion of a session that was chosen to be 90 seconds; however, a different session period could potentially change their results. The main limitation of their work is the size of the data analyzed: although the traces were collected over a period of seven months, only 80K entries were logged. It is unclear whether the inferences drawn from this study can scale up to large commercial sites. In contrast, we analyzed traces with millions of entries generated over a period of 12 days at a large commercial site. Furthermore, their study also has the limitation that it uses client IP addresses for identifying users; since IP addresses can be reassigned to different users, it is difficult to perform an accurate user-based analysis. In our study, since every entry in the logs contains a unique identifier for every access/notification, we are able to carry out user-behavior analysis more accurately. In addition, our study is broader as we focus on user behavior, server load, content, and document popularity analysis.

Tang and Baker analyzed a seven-week trace of a metropolitan-area packet radio wireless network, and a twelve-week trace of a building-wide local-area wireless network [18, 19]. Both studies focus on how the networks were used, e.g., when the networks were most active, how active the network were, and how often users moved, etc. They did not consider the content or applications for which people used the wireless networks, which is the focus of our paper.

Recently, Balachandran *et al.* [5] analyzed the user behavior and network performance of an IEEE 802.11 based wireless local area network (LAN) using a workload captured at a three day technical conference event. Their study focused on characterizing wireless LAN users for the purpose of coming up with a parameterized model to describe them. Additionally, they carried out workload analysis to address the network capacity planning problem. Their study is very different from ours in terms of analysis, methodology and objectives. While we focus primarily on wireless browse and notification services, they consider all network traffic for improving the network performance. Furthermore, the data-set they captured and analyzed is smaller and significantly different from the web server traces we analyze.

In the sections that follow, whenever appropriate, we refer to related work done by other researchers and compare it with our findings.

3 Data Characteristics

Before presenting the analysis, we briefly describe the different ways in which the web site is accessed, the characteristics of the data logs, and the types of analyses we carried out.

For the web server we used in this study, a single browse request results in exactly one HTTP request to the server. There are no images or other types of content embedded in the page that is transmitted to the client as a result of this request.

In the rest of the paper, we use the term *notification document* to refer to a unique document that may be sent to multiple users; we refer to each such transmission as a *notification message*, which includes duplicates.

3.1 Types of Accesses

For browsing, the web site is accessed in three different ways and we categorize the browse accesses based on this usage: *desktop*, *offline*, and *wireless*. Desktop accesses include requests from desktop and laptop machines connected to the web site via wireline networks. Offline accesses are generated due to handheld devices such as PDAs. Companies such as Avantgo and Vindigo

offer services that let users select content from different web sites and download it onto a handheld device for browsing at a later time. The content download occurs when a user synchronizes his/her handheld with a desktop machine and is controlled by a "downloader" program; we refer to these programmatic accesses by the downloader as offline accesses. Wireless accesses occur due to browse actions initiated by users from their cell-phones or wireless devices. Typically, a request from a cell-phone is directed to a "gateway" (operated by the user's service provider) that forwards the message to the web site; this gateway also forwards the reply back to the cell-phone. Thus, from the web site's perspective, it just communicates directly with the gateway machines using the standard HTTP protocol. Since one gateway can serve multiple clients, we do not use IP addresses to identify users; instead, we use a unique identifier assigned to every client that is logged with each access.

Browser Type	No. of accesses	No. of users
Desktop	7,342,206	639,971
Wireless	2,210,758	58,432
Offline	20,508,272	50,968
Misc	2,944,708	1,634

Table 1: User accesses according to browser types

We determine the type of access based on the browser type stored in the log entry corresponding to that access. For example, entries with browser type "Mozilla Windows", "Avantgo", "UP.Browser" are categorized as desktop, offline and wireless accesses respectively. In Table 1 we show the number of accesses according to the browser type (in our case, each access corresponds to a single HTML page). The last category (*Misc*) corresponds to log entries for which the browser type either was empty or contained characters that could not be mapped to any known browser client. The table also shows the number of unique users that were responsible for different types of accesses. Note, the number of desktop users is much higher than the offline and wireless users due to the fact that a large number of users use their desktop machines to register with the web site.

In the case of notifications, there is a client type in the logs that tells us the type of the registered clients. More than 99% of the messages were sent to wireless clients; the remaining were sent to desktop clients.

3.2 Description of Data Logs

We had access to logs for 12 days of web browsing from August 15, 2000 through August 26, 2000. There were approximately 33 million entries in the browse logs. Additionally, we used notification logs from August 20, 2000 through August 26, 2000, which contained 3.25 million entries. For our analysis of the correlation between browse and notification services (Section 6), we obtained additional notification logs and performed the comparison for the period from August 15, 2000 through August 26, 2000.

When a registered user sends a browse request to the web server, a unique identifier corresponding to the user is sent to the server and logged in the web traces (for unregistered users, the id field is empty). We use these identifiers for performing the user-based analysis. Each log record also contains other pieces of useful information along with the user ids, such as the date, time, type of browser, the URL accessed, the data received and sent by the server, etc.

When a notification message is sent, a record is logged in a database. We obtained a part of this database for our analysis. The database entries contained information about the server from where the notification message was sent, a user id, type of the device to which the message was sent (e.g., phone or pager), type of alert, when it was sent, etc.

To efficiently manipulate a large amount of data logs (over 10 GB), we consolidated them into a commercial database system and created indices on columns such as date, user id, and URL. To overcome the limited expressiveness of our database language (in terms of string manipulation), we further processed the database output using Perl scripts.

3.3 Types of Analyses

We now discuss the types of analyses that we perform on the notification and browse logs, and the motivations for doing these analysis.

1. **Content analysis**: We are interested in questions such as: (i) what are the most popular content categories, and (ii) what is the distribution of message sizes? We believe such questions are important to (i) content providers who need to understand better how to prioritize and use the system and network resources efficiently, and to (ii) web site developers who are interested in supporting fast access to popular content.
2. **Popularity analysis**: We are interested in the popularity distribution of notification and browse doc-

uments. In particular, we are interested in comparing these accesses to the well-known Zipf-like distribution as reported in previous web studies [4, 7, 10, 14, 16], and in determining how concentrated are the number of requests/transmissions for popular documents. This has significant implication for the effectiveness of web caching and multicast delivery.

3. **User-behavior analysis**: We are interested in classifying users according to their access patterns. This is useful for personalization, targeted advertising, prioritizing, and capacity planning. Specifically, we look at the following aspects of user behavior:

   - *Spatial Locality*: whether users in the same geographical region tend to receive/request similar notification and browsing content.
   - *Temporal Stability*: whether users are interested in browsing similar documents over time.
   - *User Load Distribution*: how different users place load on the web site; for service providers, this distribution has implications on pricing.

4 Notification Log Analysis

Table 2 shows the overall statistics for the notification logs. In one week, the server sent out 3.25 million notification messages for a total of 295 megabytes. One fourth of the messages sent out were distinct, while the remaining messages had the same content but sent to different users (in some cases, the same message is sent to a user multiple times, e.g., if a user has registered for information to be delivered at specific times and the information has not changed during that period). The significant amount of duplication in messages sent to different users suggests that sending notification via application-level multicast would be useful; Section 4.2 examines this issue in greater depth. There were 200,860 distinct users, of which 99.02% were wireless users. The notifications were sent at the average rate of 323 messages per minute. The peak rate was much higher, approximately 30 times as high as the average rate.

4.1 Content Analysis

We begin our analysis by looking at the content of the notifications sent to various users.

Total messages	3,251,537
Total distinct messages	884,272
Total bytes transmitted	295 MB
Total bytes of unique messages transmitted	71.3 MB
Total number of users	200,860
Total number of wireless users	198,882
Avg. notification rate	322.57 (msgs/min)
Peak notification rate	9502 (msgs/min)

Table 2: Overall statistics for the notification logs for the period from Aug 20 through Aug 26, 2000.

4.1.1 Popular Categories

We classified the notifications into categories based on the subject field, which was recorded in the notification logs. We plotted the number of messages sent for each notification category in Figure 1, and the number of users who received the notification message for each category in Figure 2.

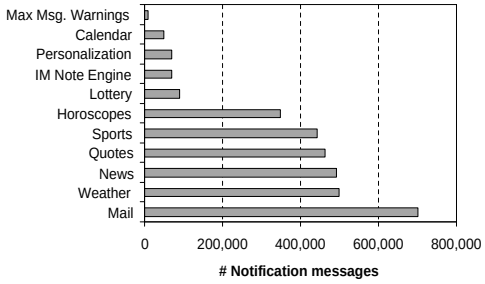


Figure 1: The total number of notifications sent for each category.

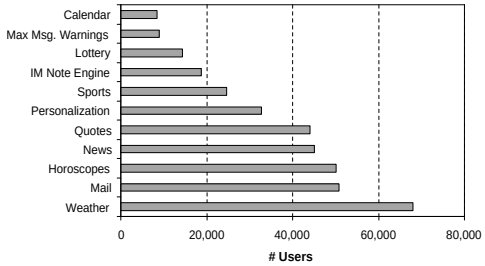


Figure 2: The total number of users who received notifications for each category.

As Figure 1 shows, email, weather, news, stock quotes, sports, and horoscopes are the most popular categories in terms of the total number of notification messages. In comparison, weather, email, horoscopes, news, and stock quotes are the most popular categories in terms of the total number of users (see Figure 2). As we had expected, email alerts were very popular. On the other hand, we had not expected weather-related notifications

to be so popular. Intuitively, one might have expected stock quotes and news to be more popular, especially since users have to explicitly register for different notification types (including weather), i.e., notifications are not being sent due to some default setting on the user-signup page. Another surprise was the low popularity of calendar alerts. For calendar alerts, it is possible that subscribers use handheld devices that are not connected to the wireless Internet, for example, PDAs with pre-installed software to handle scheduled meetings, anniversaries, etc.

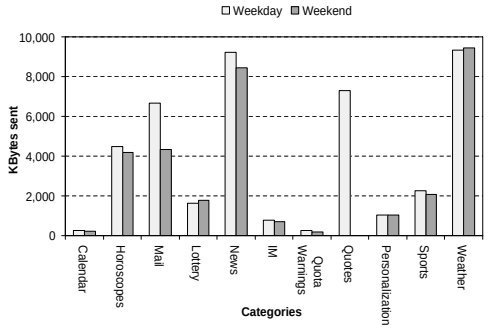


Figure 3: Change of user interest between weekday and weekends

Next we analyzed how user interest changed during the course of a week. Figure 3 shows a comparison between the amount of notification data sent on a weekday versus a day on the weekend. As one would expect, there is a significant difference between the number of stock quote alerts sent during the weekday compared to those sent on the weekend. Similarly, there are fewer mail alerts on weekends; this is probably due to lower levels of work activity that occur on weekends relative to weekdays, resulting in fewer triggering events. For other categories (e.g., sports, weather, horoscopes), the number of notification messages does not vary significantly over weekends and weekdays. We attribute these patterns to the fact that not many users personalize all aspects of their notification portfolio in a very fine-grained manner (for event types such as weather, the web site allows users to select the frequency and the time of delivery).

4.1.2 Notification Message Size and Its Implications

We find that notification messages are small. Specifically, all messages contain less than 256 bytes. We show the message size distribution in Figure 4 to illustrate this point. Consequently, it is important for the delivery protocol to handle small messages efficiently. For example, if the protocol creates a new TCP connection for every notification message, the overhead can be high. In particular, the connection establishment may increase the user-perceived latency by a factor of 3 (i.e., from one half round-trip time to one and a half round-trip time). Assuming the average notification message size to be 128 bytes, the connection setup and tear-down increases the bandwidth usage from 168 bytes per message to 448 bytes per message (i.e., 7 additional packets: 3 packets in the three-way handshake connection setup, and 4 packets in the connection teardown).

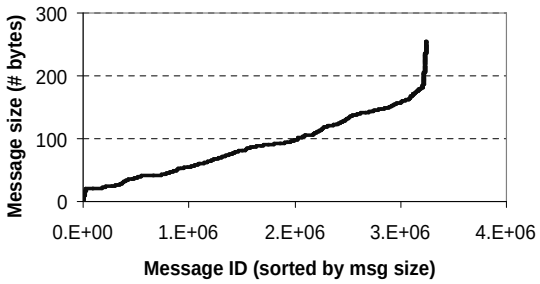


Figure 4: Size distribution of notification messages (including duplicates).

One suggestion for reducing the overhead of connection setup and teardown is to use persistent connections [13], i.e., reuse a TCP connection for multiple transfers. In our case, the servers sending the notification messages can maintain persistent connections with the gateways of the wireless ISPs and then send all messages on this connection.

4.2 Message Popularity Analysis and Its Implications

Several studies have found that web accesses follow Zipf-like distribution: the number of requests to the i^{th} most popular object is proportional to $\frac{1}{i^{\alpha}}$ [3, 4, 6, 7, 10, 14, 16]. The estimates of α range from 0.5 to 1 for web proxy logs [7, 10, 14], and range from 1 to 2 for web server logs [4, 16]. It is interesting to examine whether notification messages exhibit a similar property.

To do the above, we take the following approach: For each notification document, we count the number of notification messages (i.e., copies) that were sent on a given day. We plot the total number of transmissions of a document (i.e., notification messages) versus the popularity ranking of the document on a log-log scale. Figure 5 shows the plot for August 21, 2000. The plots for the other days are similar, and are omitted for brevity. If we ignore the first few notification documents and the flat tail in Figure 5 (as is done in the previous work [6, 7, 16]), we note that the curve fits a straight line reasonably well. We compute the values of α using least-square fitting, after excluding the top 20 doc-

uments and the flat tail (the latter set represents the notification documents that were sent only once or twice). The straight line on the log-log scale implies that the notification documents follow a Zipf-like distribution. We find that for our complete data-set the value of α varies from 1.137 to 1.267 (in Figure 5, the value of α is 1.146). These values are higher than the α in the web proxy logs [7, 10, 14], and lower than (but close to) the α observed for popular web server logs [16].

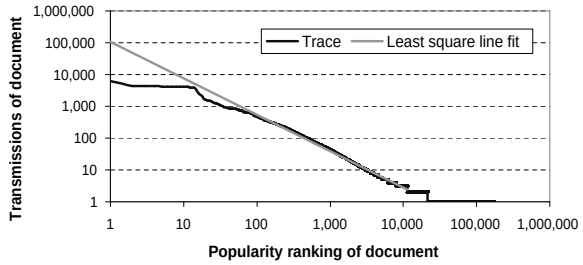


Figure 5: Frequency of notification documents versus ranking in log-log scale (for August 21, 2000).

Figure 6 shows the cumulative distribution of notification documents on August 21, 2000. The top 1% of notification documents (i.e., 1704) account for 54.24% of the total notification messages. In the logs for other days, the top 1% of notification documents account for 54.15% - 63.66% of the total messages. Such a high concentration of messages containing popular documents suggests that using application-level multicast [8, 11, 17, 22] for popular documents would yield significant savings in both bandwidth and server load.

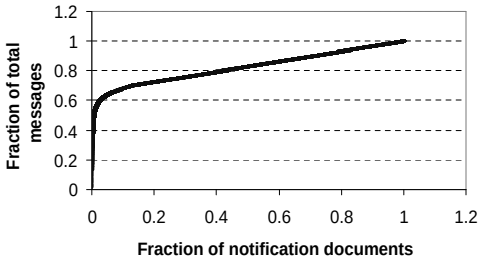


Figure 6: Cumulative distribution of notification messages to documents (for Aug 21, 2000).

A possible optimization is to distribute a set of caches over the Internet to form an overlay multicast tree rooted at the notification server. When a notification message needs to be sent to multiple recipients simultaneously, it can be sent over the overlay tree and also stored at the caches that it traverses. These caches can help in offloading the retransmission work (say, due to a client coming online) from the server: when the same copy of notification needs to be sent at a later time, the caches closest to the receiver can forward the message

Note that even though the current notification traffic is not significant, as the popularity of notification services increases, bandwidth usage will become an important factor for scaling the notification system. Consequently, optimizations such as application-level multicast will become more important.

We also observed that the concentration of notification messages to documents becomes less pronounced as the number of the documents considered increases. For example, the top 7.6% – 42.0% of the documents account for 80% of the total messages, and the top 45.1% – 71.0% of notifications account for 90% of the total messages. This implies that a large performance benefit can be obtained by multicasting only the most popular notification documents.

4.3 User Behavior Analysis

We now study two aspects of user behavior: (i) the spatial locality of user interest, and (ii) the distribution of load that users place on the server.

4.3.1 Spatial Locality

Spatial locality of user interest is about determining whether people in the same geographical region tend to receive similar notification content. To carry out our analysis we take the following approach. We define a notification message to be locally shared if at least two users in the same cluster receive the notification. We compare the degree of sharing using geographical clustering and four random clusterings. In the geographical clustering case, clients in the same city are clustered together. In the random clustering case, clients are clustered randomly with the cluster size being the same as in geographical clustering. We obtained the geographical location of users using a registration database which contains zip code information for each user. The zip code information is not clean — some users supplied invalid zip codes; we filter out all the zip codes that are not 5 digits. 14% of the users supplied such invalid zip codes. In the remaining entries, it is still possible to have zip codes that do not match the actual user location, but the fraction is likely to be small. Furthermore, when computing the degree of local sharing, we exclude the cities to which fewer than 100 notification messages were sent over the course of the week.

As shown in Figure 7, clients residing in the same city have significantly more sharing in notification content compared to the clients picked at random. We also compared geographical clustering with three other random clusterings and observed similar results. The higher degree of sharing in notification messages for clients in the same geographical region indicates that localized services are popular for notification services. For example, people living in New York are interested in receiving notification messages about weather or events in New York. The geographical locality in notification content implies that placing servers (i.e., either notification server replicas or servers in an overlay network that provide application-level multicast) close to popular geographical clusters can be useful in reducing network load.

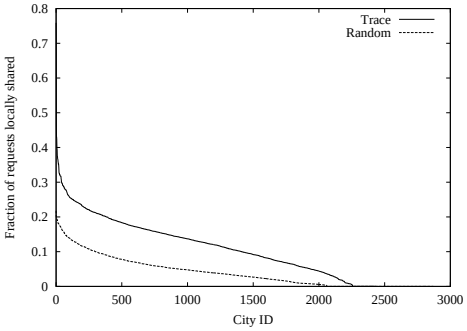


Figure 7: Compare the local sharing between random clients and clients that are geographically close together.

4.3.2 Load distribution of different users

On average, we observed that a user receives 2.3 notification messages containing a total of 0.2 KBytes per day, and 16.1 notification messages containing 1.4 KBytes of data per week. There is a significant variation in the clients' usage — during the week that we studied, some clients received over 1000 messages (containing as high as 0.1 MB of data), while other clients received fewer than 10 messages containing as little as a few hundred bytes of data.

Figures 8 and 9 show the total number of messages and the total number of bytes received by different users on a log-log scale, respectively. Both curves fit very well with a straight line (i.e., follow Zipf-like distribution), except at the tail where there is a sudden drop. We compute the values of α using least-square fitting, after excluding the sharp drop at the tail. The value of α is 0.4437 when usage is defined as the number of messages; when usage is defined as the number of bytes, its value is 0.4567.

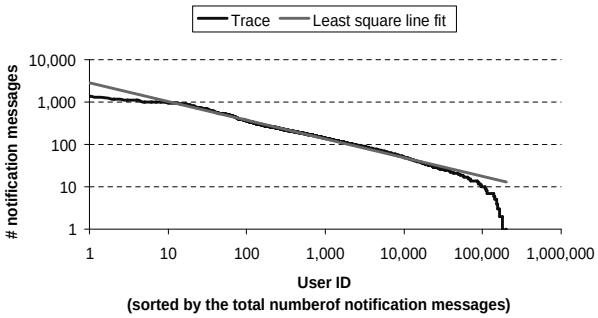


Figure 8: The total number of notification messages received by different users.

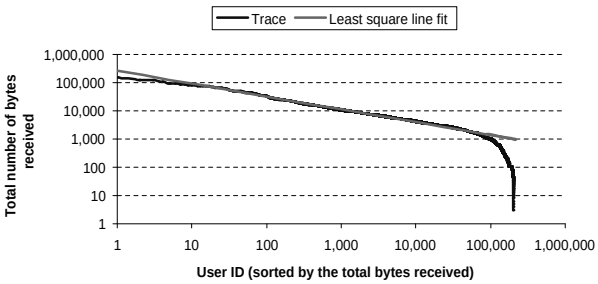


Figure 9: The total number of notification bytes received by different users.

To further study how usage is distributed across different clients, we plot the cumulative distribution of client usage in Figure 10. As the figure shows, the top 5% of the clients received 28% of the notification messages, and 25% of the notification bytes; the top 10% of the clients received 40% of the notification messages, and 38% of the notification bytes. It is clear that a small fraction of users consume a significant fraction of the system and network resources. It is also interesting to note that the CDF curves are similar for the two different ways of defining usage. The similarity of the curves shows that each user receives a similar number of bytes per message.

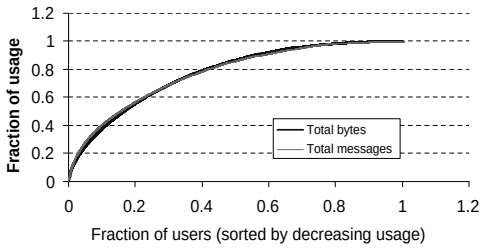


Figure 10: Cumulative distribution of different clients' usage.

The cumulative load imposed by all users (in terms of number of messages and the number of bytes sent by the servers) is shown in Figure 11. The figure shows that the number of messages and the number of bytes are fairly constant during weekdays but exceed the number sent

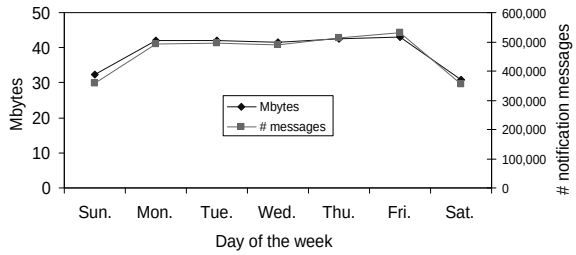


Figure 11: Number of bytes and messages served by the notification servers during the days in the week

during the weekend. This confirms what one would expect, i.e., information alerts are more frequently generated when people are working.

4.4 Summary

Our analysis shows that notification messages are small, popular documents account for a significant fraction of the messages, and there exists a high degree of sharing in geographical regions. System designers need to develop transport protocols that can send such messages in a reliable, efficient and secure manner. For example, an overlay network consisting of geographically placed caches along with application-level multicast can reduce the total network bandwidth requirements and server load. We also observed that there is a significant variation in clients' usage of notification services. Service providers can design pricing plans according to the needs of the clients and also specialize content based on geographical location.

5 Browser Log Analysis

In this section, we present our browser logs analysis. In our earlier work, we performed analyses on document content and popularity, distribution of user sessions, and system load [1]. For the sake of completeness, we first summarize the major findings of our previous analysis, and then study the temporal stability and spatial locality of user accesses, as well as the distribution of the load placed by different users on the web server.

5.1 Summary of previous analysis

In [1], we analyzed the browser log collected during the period from August 15, 2000 through August 26, 2000. During this time the web server received 1.6 – 3.2 million requests per day from 64,000 – 98,000 distinct clients. Below is a synopsis of our major findings:

1. The distribution of document popularity does not closely follow Zipf-like distribution, where a document is defined as a unique URL or as a unique URL and parameter pair. The majority of requests are concentrated on a small number of documents. In particular, we found that 0.1% – 0.5% of the documents (i.e., approximately 121 – 442) account for 90% of the requests.

2. More than 60% of the pages accessed at the web server are due to offline PDA users and less than 7% of the accesses are due to wireless clients; the remaining accesses are due to desktop clients for registration and customization services.

3. Our analysis for the distribution of reply sizes showed that most of the replies to wireless clients are less than 3 KBytes. For offline clients, most of the replies are less than 6 KBytes. The reply size distribution for the two types of clients is similar.

4. Our user session analysis showed that users tend to have short sessions when interacting with the web site: 95% of the sessions were less than 3 minutes. We empirically determined the session-activity threshold to be somewhere between 30 to 45 seconds (i.e., if no request is received from a client for such a duration, it implies that the old session has ended).

5. Our category analysis showed that stock quotes, news, and yellow pages are the top categories accessed by wireless clients. For offline clients, help is the most popular category followed by news and stock quotes.

6. We observed that the relative importance of different categories did not change between weekdays and weekends (except stock quotes and sports). However, the amount of data accessed over the weekend drops by approximately 45%.

These findings have the following performance implications:

1. The high concentration of requests to popular documents in the browser log implies that caching the results of popular queries would be very effective in reducing the web server load.

2. Since most replies sent to wireless and offline users are small (3 – 6 KB), the wireless web server should be highly optimized in sending short replies, e.g., optimizing TCP slow start and re-start [15, 23] can be useful in this environment.

3. Our heuristic, based on user session analysis, to determine the session-inactivity period can be useful to wireless service providers who want to reclaim IP addresses. Our analysis showed that IP addresses may be reclaimed more quickly than the time period determined in earlier work [12].

5.2 New Analysis

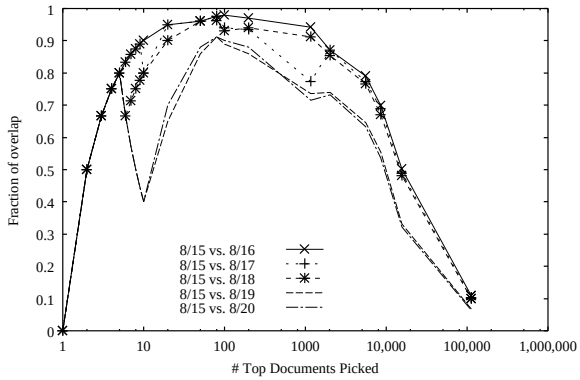
We now present a user-based analysis of the browser logs (based on the unique identifier associated with each browse request). We examine temporal stability, spatial locality, and usage variation across different users.

5.2.1 Temporal Stability

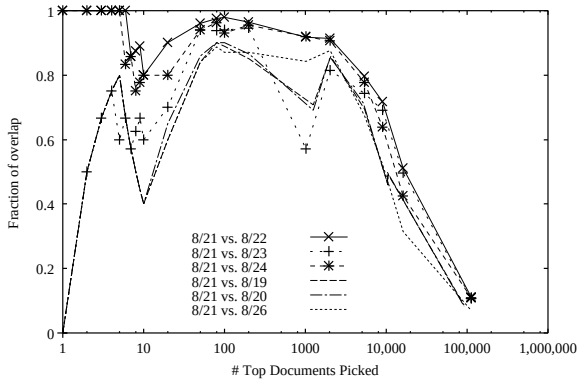
In the section, we analyze whether users are interested in a similar set of documents on different days. To answer this question, we pick the N most popular documents from each day, and compare the extent of the overlap. Since all the web pages are dynamically generated, a document is defined as a combination of a unique URL name and the query parameters (i.e., two requests with the same URL with different parameters are considered as different document requests). We will use the terms document and query interchangeably in this section.

First we study the requests from all users, i.e., including wireless, offline, and desktop users. Figure 12 (a) and (b) plot the overlap between weekdays August 15 (Tuesday) and August 21 (Monday) versus other days (i.e., both weekend days and weekdays) (In Figure 12 (a) and (b), the curves with points are for pairs of weekdays, and those without points are for a weekday and weekend.) Figure 13 plots the overlap between weekend days. Note that the x-axis data value for the top N case does not always correspond to exactly N in the graphs. The reason is that when we consider the top (say) 100 documents, the next few documents after these documents may also have the same frequency as the 100^{th} document; since we include these documents as well for the "top 100" data point, it sometimes results in a small mis-match of the plotted points.

Looking at Figure 12 (a) and (b), we make the following observations: first, the overlap between different days is significant. For example, the overlaps are over 80% for the top 100 documents, and mostly over 70% for the top 1000 documents. This indicates that the set of popular queries remains relatively stable, and suggests that we can cache a stable set of popular query results or opti-



(a) Overlap between a weekday versus other days.



(b) Overlap between another weekday versus other days.

Figure 12: Temporal stability of document ranking for weekdays

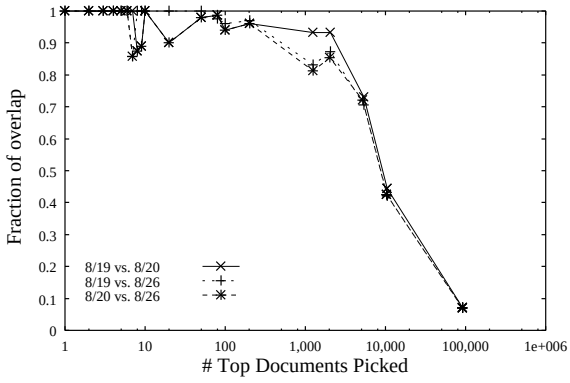


Figure 13: Temporal stability of document ranking between weekend days

mize the data layout to improve the performance of these queries. For example, workload-based techniques can be used to generate indices and materialized views automatically for a database [2]; these techniques are largely applicable if the database query workload is relatively stable (which is the case for our browser queries).

Second, the overlap initially fluctuates with the increasing number of documents picked, and then decreases when the number of top documents picked is over 100. The initial fluctuation is probably due to the fact that although very popular documents tend to remain popular, their relative ranking does change over time. However, as we further increase the number of documents, we may include some less popular documents. Since these documents are less likely to remain popular than very popular documents, the temporal overlap decreases. This phenomenon was also observed in [16].

Third, the overlap between pairs of weekdays is generally higher than the overlap between a weekend day and a weekday. The overlap between two weekend days is even higher. This is consistent with our intuition, and suggests that we should use past weekday workload to predict future weekday workload, and likewise use past weekend workload to predict future weekend workload.

We also examine the requests coming from only the wireless users, and find the results are very similar. As before, the set of popular queries remains stable over time. The stability is especially high when we consider the most popular queries. In addition, there is a significant difference between the access pattern on weekdays versus that on weekends.

5.2.2 Spatial locality

In this section, we consider the following question: do people in the same geographical region tend to issue a similar set of queries. We employ the same approach as is used in studying the spatial locality for notification services (described in Section 4.3.1).

Figure 14 compares the fraction of documents that are shared within a geographical cluster and within four random clusters, when we consider requests from all the users (excluding users with invalid IDs). The figure shows that the curve for the geographical clusters overlaps with those for random clusters. This overlap indicates that the degree of sharing between geographical clustering and random clustering is comparable, and the correlation between users' interest in browsing over wireless channels and their geographical location is weak.

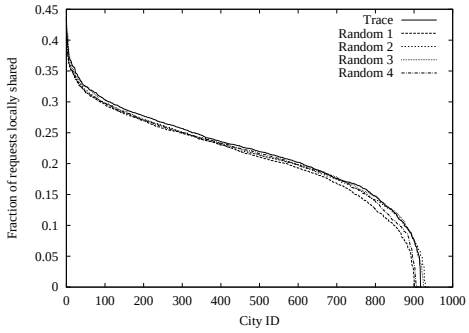


Figure 14: Local sharing between random sets of clients and clients that are geographically close together.

A possible explanation for the weak correlation is that the popular browse content has global interest. In particular, as mentioned in Section 5.1, 0.1% - 0.5% of the URL and parameter combinations (i.e., about 121 – 442 unique combinations) account for 90% of the requests. With such a high concentration of user interest on a few documents, even when clients are picked at random, they share many requests; therefore, the geographical locality becomes insignificant. A similar phenomenon has been observed in a study of a popular news server [16], where the authors observed that the significance of domain membership becomes diminished during a popular event. A major distinction between that study and ours is the way in which users are clustered: in that study, users are clustered based on their DNS names, whereas in our study we cluster users based on their geographical region, e.g. the city in which they reside.

A natural question follows – why is there such a high concentration of interest in popular documents that even when clients are picked at random they share many documents? Examination of the most popular URLs and parameters shows that they include the front pages for email login, news, sports, weather, lottery, and the signup application, as well as some popular stock quote queries. Intuitively, these queries are very popular to all users regardless of their physical locations.

The lack of geographical locality implies that the web server's content can be replicated without keeping in mind the geographical location of the clients.

We performed the same spatial locality analysis to requests issued only by wireless clients. Figure 15 summarizes the results. With geographical clustering, wireless clients have slightly more sharing of documents than with random clustering; however, the distinction between the two clusterings is much less significant than

the difference observed for notification documents. This result suggests that using geographical locality of wireless users as input for optimizing performance (or providing content) will yield limited success.

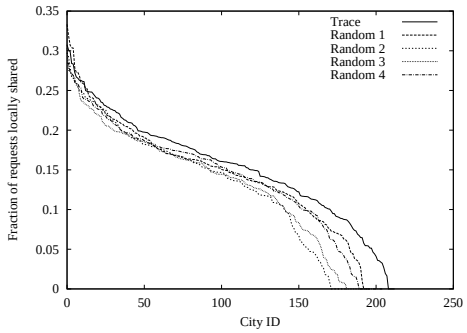


Figure 15: Comparison of local sharing between random sets of wireless clients and wireless clients that are geographically close together.

5.2.3 Load distribution of different users

In this section, we study the distribution of loads placed on the web server by different users. Our earlier analysis [1] examined the difference in load distribution between wireless users and offline users. Now we look at the load distribution at a more fine-grained level — at a per-user level.

Figure 16 and Figure 17 show the total number of accesses and total number of data requested by different clients, respectively (users with invalid identifiers were discarded). As the figures show, there is a significant variation in the load placed by different users on the web server: some users request several orders of magnitude more documents/data than other users. The accesses from only the wireless clients reveal similar property. Thus, service providers can consider designing different pricing plans that to cater to the widely varying needs of different users.

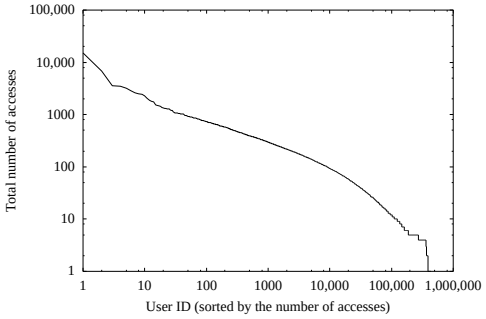


Figure 16: Total number of accesses made by different users.

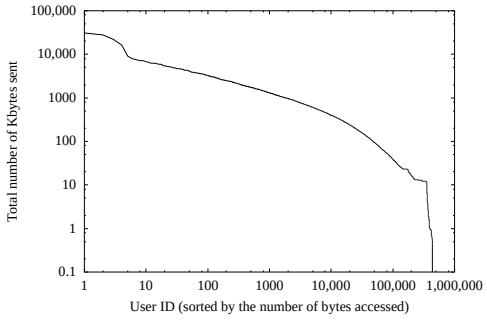


Figure 17: Total number of data received by different users.

Figure 18 shows the inter-arrival time between requests coming from the same user. The requests generated from the offline users are much more bursty than those from the wireless users: 97% of the requests from the offline users have 1 second or less inter-arrival time, whereas only 9% of the requests from the wireless users have comparable inter-arrival time. We observe very bursty traffic for offline PDA users because their requests are generated by the downloader program rather than a human being; these users also generate significantly more requests than wireless users. If not handled appropriately, such bursts can delay wireless users unnecessarily. The web site designers can address this problem in a number of ways. For example, they can provide higher priority to wireless users or restrict the burst of offline user requests to a few front-door servers (servers that handle incoming HTTP requests). An orthogonal efficiency issue that needs to be addressed is the synchronization protocol for PDAs, i.e., instead of sending a large number of small requests, the synchronization protocol could batch all these requests into a single request and reduce the server load and roundtrip latency.

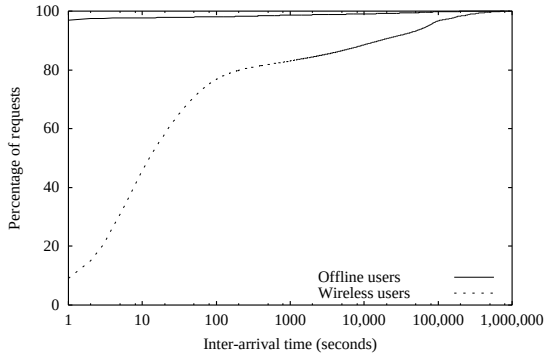


Figure 18: CDF of inter-arrival time between consecutive requests from the same user.

6 Correlation between notifications and browsing

Having studied both the notification logs and the browse logs, it is useful to understand whether there is any correlation between the browsing and notification activities of users. We are interested in answering questions such as: (i) do users utilize one of the services significantly more than other services, and (ii) does their interest in particular content categories differ across the two services. We use the notification and browser logs, both spanning from August 15, 2000 through August 26, 2000 for the following analysis.

6.1 Correlation in the amount of usage

Figure 19 shows the average number of notification messages versus the number of browse requests, and the average number of browse requests versus the number of notification messages. There is little correlation between the two variables: the number of notification messages fluctuates widely with the number of browse requests; similarly, the number of browse requests also shows no obvious trend with respect to the number of notification messages. The correlation coefficient between these two variables is 0.265 when considering all users, and 0.125 when considering only wireless users. The low correlation coefficients implies that web site designers cannot predict a user's browsing activity based on his/her notification activity, and vice versa.

6.2 Correlation in popular content categories

We now look at the question whether users are interested in a similar set of content categories across the two services. To answer this we take the following approach: first, we classify notification messages and browsing accesses into different categories. (The details of categorizing notifications are described in Section 4.2, and the details of categorizing browse accesses are described in our earlier work [1].) Then for each individual user, we pick the top N content categories in browsing and top N content categories in notification (if the next few categories after the N^{th} category have the same frequency of access as the N^{th} category, we include those categories as well for the top N case).

Figure 20 shows the percentage of users who have at least some overlap between their top N browse and notification categories. The degree of overlap is much higher when we consider wireless users only. For example, for

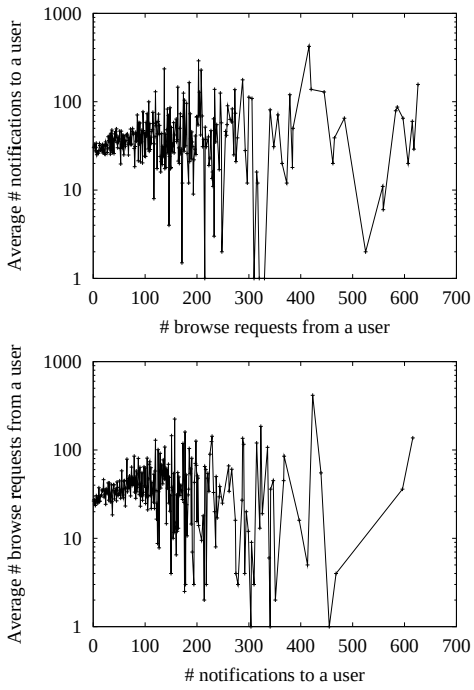


Figure 19: Correlation between the number of browse requests and notifications of wireless users.

the top 3 categories, the percentage of overlapped users is less than 10% when considering all the users, and around 50% when considering only the wireless users. On the other hand, even when considering wireless users only, the number of overlapped users is never more than 65%.

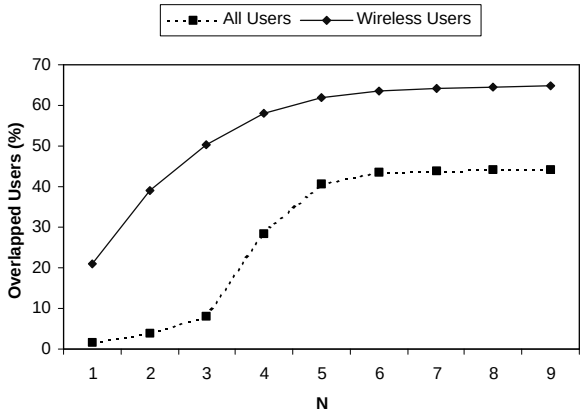


Figure 20: Number of users who have overlap between their top N browsing categories and top N notification categories.

We now compare the extent of the overlap by varying N from 1 to the total number of categories. The results are shown in Figure 21. The figure shows the average percentage of overlap between two categories, where the average overlap is computed as follows:

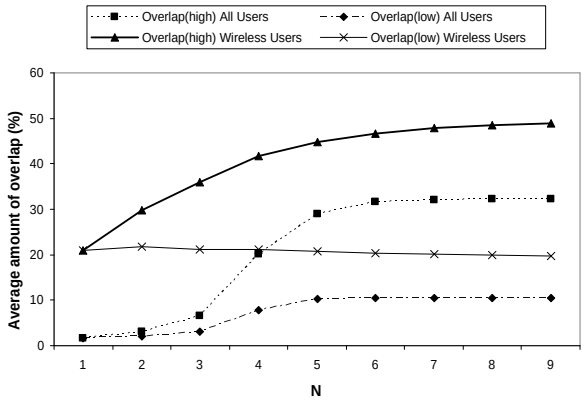


Figure 21: Correlation between the number of browse requests and notifications of wireless users.

$$overlap_{high} = \frac{\sum_i \frac{\#categories\ overlapped\ for\ user_i}{min(N, min(BC, NC))}}{relevant\ users}$$

$$overlap_{low} = \frac{\sum_i \frac{\#categories\ overlapped\ for\ user_i}{min(N, max(BC, NC))}}{relevant\ users}$$

where BC denotes the number of browse categories, NC denotes the number of notification categories, and relevant users refers to those users that have at least one browse record and one notification record in the respective logs. We show the results for only the top 9 categories, since the values beyond that are stable.

Essentially these ratios compute the percentage of overlap for each individual user, and then take the average of these percentages over all wireless users or all users. Since not all users have at least N browsing or notification categories, we compute $overlap_{high}$ and $overlap_{low}$, where the former computes the percentage of overlap by using the minimum of BC and NC, and the latter uses the maximum of BC and NC. The figure shows that the amount of overlap is considerably higher when considering only wireless users. For example, for the top three categories, the overlap is less than 7% when considering all users. In comparison, for wireless users, the $overlap_{low}$ and $overlap_{high}$ values are 21% and 36%, respectively. We also observe that the effect of increasing N is small. Even when N is 8, the percentage of overlap is less than 50% for wireless users.

The above results indicate that wireless users have moderate correlation in the way they use browse and notification services. In comparison, the correlation is much lower when considering all users. This is because the most popular browsing categories for desktop users are sign-up services, direction, and general help, whereas notification is usually not used to deliver these types of content. On the other hand, some wireless users are interested in both browsing and receiving notifications about emails, stock quotes, personalization, news and sports. However, the degree of correlation is limited, and service providers cannot solely rely on a user's notification profile to determine what content he/she may be interested in browsing.

7 Conclusions

Internet access via small handheld devices is expected to increase tremendously in the next few years. In this paper, we analyzed the access patterns of a large web site designed primarily for wireless and handheld mobile devices. The web site provides both browse and notification services. To our knowledge, this is a first-of-a-kind study that analyzes notification services. It is also first in analyzing user behavior using a commercial web site. We believe this is an important first step in the direction of understanding the dynamics of wireless Internet services.

8 Acknowledgements

We thank Mike Spreitzer for carefully reading our paper several times and helping us improve its quality significantly. We also thank the anonymous reviewers for their detailed comments and suggestions that improved the paper in several aspects.

References

[1] A. Adya, P. Bahl, and L. Qiu. Understanding the Browse Patterns of Mobile Clients using a Popular Web Server. *ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[2] S. Agrawal, S. Chaudhuri, and V. Narasayya. Automated Selection of Materialized Views and Indexes for SQL Databases. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB00)*, Sep 2000.

[3] M. Arlitt and T. Jin. Workload characterization of the 1998 World Cup Web site. *IEEE Network*, pages 30 – 37, May/June 2000.

[4] M. F. Arlitt and C. L. Williamson. Internet Web Servers: Workload Characterization and Performance

Implications. In *IEEE/ACM Transactions on Networking*, pages 631–645, 1997.

[5] A. Balachandran, G. Voelker, P. Bahl, and V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *Proceedings of ACM SIGMETRICS '02*, 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. In *Proceedings of INFOCOMM '99*, March 1999.

[8] Y. Chu, S. Rao, S. Seshan, and H. Zhang. Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture. In *Proceedings of ACM SIGCOMM 2001*, August 2001.

[9] C. Cunha, A. Bestavros, and M. Crovella. Characteristics of WWW Client-based Traces. *Technical Report TR-95-010*, April 1995.

[10] S. Glassman. A Caching Relay for the World Wide Web. In *Proceedings of 1st WWW Conference*, May 1994.

[11] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and Jr. J. W. O'Toole. Overcast: Reliable Multicasting with an Overlay Network, 2000.

[12] T. Kunz, T. Barry, X.Zhou, J.P.Black, and H.M.Mahoney. WAP Traffic: Description and Comparison to WWW Traffic. *ACM Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, August 2000.

[13] J. C. Mogul. The Case for Persistent-Connection HTTP. In *Proceedings of ACM SIGCOMM 95*, August 1995.

[14] N. Nishikawa, T. Hosokawa, Y. Mori, K. Yoshidab, and H. Tsujia. Memory-based Architecture for Distributed WWW Caching Proxy. In *Proceedings of 7th WWW Conference*, April 1998.

[15] V. N. Padmanabhan and R. Katz. TCP Fast Start: A Technique for Speeding up Web Transfers. In *Proceedings of IEEE Globecom'98*, November 1998.

[16] V. N. Padmanabhan and L. Qiu. The Content and Access Dynamics of a Busy Web Site: Findings and Implications. In *Proceedings ACM SIGCOMM 2000*, August 2000.

[17] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel. ALMI: An Application Level Multicast Infrastructure. In *Proceedings of USITS 2001*, March 2001.

[18] D. Tang and M. Baker. Analysis of a Metropolitan-Area Wireless Network. In *Proceedings of ACM MobiCom 99*, pages 13–23, August 1999.

[19] D. Tang and M. Baker. Analysis of a Local-Area Wireless Network. In *Proceedings of ACM MobiCom 2000*, August 2000.

[20] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, M. Brown T. Landray, D. Pinnel, A. Karlin, and H. Levy. Organizational-based Analysis of Web-Object Sharing and Caching. In *Proceedings of USITS '99*, October 1999.

[21] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, M. Brown T. Landray, D. Pinnel, A. Karlin, and H. Levy. On the Scale and Performance of Cooperative Web Proxy Caching. In *Proceedings of SOSP '99*, Dec. 1999.

[22] H. Yu, L. Breslau, and S. Shenker. A Scalable Web Cache Consistency Architecture. In *Proceedings of ACM SIGCOMM '99*, 1999.

[23] Y. Zhang, L. Qiu, and S. Keshav. Speeding up Short Data Transfers: Theory, Architectural Support, and Simulation Results. In *Proceedings of NOSSDAV'2000*, June 2000.