Analyzing and Accelerating Web Access in a School in Peri-Urban India

Jay Chen
New York University
jchen@cs.nyu.edu

David Hutchful
Microsoft Research India
davihu@microsoft.com

William Thies
Microsoft Research India
thies@microsoft.com

Lakshminarayanan Subramanian
New York University
lakshmi@cs.nyu.edu

ABSTRACT

While computers and Internet access have growing penetration amongst schools in the developing world, intermittent connectivity and limited bandwidth often prevent them from being fully utilized by students and teachers. In this paper, we make two contributions to help address this problem. First, we characterize six weeks of HTTP traffic from a primary school outside of Bangalore, India, illuminating opportunities and constraints for improving performance in such settings. Second, we deploy an aggressive caching and prefetching engine and show that it accelerates a user's overall browsing experience (apart from video content) by 2.8x. Unlike proxy-based techniques, our system is bundled as an open-source Firefox plugin and runs directly on client machines. This allows easy installation and configuration by end users, which is especially important in developing regions where a lack of permissions or technical expertise often prevents modification of internal network settings.

Categories and Subject Descriptors

H.5.4 [**Hypertext/Hypermedia**]; K.3.1 [**Computers and Education**]

General Terms

Experimentation, Human Factors, Measurement, Performance

1. INTRODUCTION

The emerging regions of the world were left behind by the Internet revolution. More recently, whether for philanthropy or profit, the areas which have traditionally been underserved by the sweeping technological tide are gaining more attention. One central focus of these efforts is in the area of information and communication technology (ICT) for education. Both non-profits and local governments are increasing spending to equip schools with computers and, increasingly, Internet connectivity. Despite the growing emphasis of ICTs for education, relatively little is known about the actual use of the web and the information needs of this segment of the population.

In this paper we aim to analyze and improve web access at a school outside the city of Bangalore, India. The infrastructure at this school is quite good relative to rural areas; the school contains 97 computers which share a 2 Mbps link, and have a 10 GB bandwidth limit per month. Nonetheless, the school suffers from power outages and brownouts, and network connectivity is sometimes interrupted by nearby construction efforts [1].

We initiate our inquiry by analyzing a six-week HTTP trace gathered from 37 of the client machines in the school. While researchers have analyzed web traces from developing regions before [13, 26, 33, 20, 25], they did not focus on an educational environment. In some cases our results corroborate prior findings, while in some dimensions they differ. For example, we observe the shift of web content towards more dynamic pages that complicate offline caching. We were also surprised by the dominance of video and advertising content, even in the absence of a very high-bandwidth connection.

Our second emphasis is in deploying a practical tool to accelerate web traffic in the school environment. This tool leverages prior techniques for aggressive caching, including serving stale pages, as utilized by the C-LINK system in Nicaragua [25]. It also performs client-side prefetching, which has been studied previously [24]. An innovative aspect of our tool is that it is very easy to install: rather than requiring the user to adjust any proxy settings (as is typical of caching and prefetching systems), our tool is bundled as a Firefox plugin that is immediately accessible to non-expert users. Given the many constraints on security, privacy, and local technical expertise in our target environments, a simple browser extension can enable incremental deployments and reduce the administrative overhead of configuring the system. One consequence of utilizing a browser plugin is that all caching and prefetching operations must be client-based, thereby restricting certain optimizations in favor of improved deployability. We also experiment with offline browsing of cached pages during outages using our extension.

We evaluate the effectiveness of web acceleration techniques via a six-week deployment. While similar techniques have been evaluated in a delay-tolerant network [25], we are unaware of comparable deployments in a connected environment. Our high-level result is that, by aggressively caching and prefetching content, our plugin accelerates the overall web browsing experience by a factor of 2.8x (not counting video content, which appears on less than 1% of browsed pages). We observe an overall cache hit rate of 31%; for pages in which content and all embedded objects are cached, the plugin accelerates load time by 9.1x. While prefetching also proved to be helpful, with 23% of prefetched pages eventually being accessed by the user, prefetching accounted for less than 2% of overall cache hits. We also did not demonstrate gains due to offline browsing, due to lim-

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

ited network outages and limited depth of prefetched pages. Overall, our findings suggest that a client-side web accelerator has considerable promise for addressing the real-world network challenges in a developing-region school.

In the remainder of this paper, we begin with background and motivations for our work. We then describe the school, constraints on our deployment, and the resulting set of web acceleration techniques we were able to employ based on those constraints. We go on to present the web traffic analysis and compare and contrast our findings with those in previous work, and to examine how well each of our acceleration techniques performed. Finally, we discuss the implications of our results and directions for future research.

2. BACKGROUND AND RELATED WORK

Web trace data analysis has long history. Our study confirms well-known features of web traffic, including a power law distribution of web requests to their popularity, self-similarity, and a diurnal cycle [5, 49, 6]. Later research on web caching, prefetching, compression, and other acceleration techniques built upon the findings from these traces, highlighting their importance [43, 31].

Relatively little is known about web use in emerging regions particularly among the poorly connected [51]. Research that focuses on traditional constrained web access is either dated [29, 34] or, in the case of more recent work, is in the context of mobile devices [52, 27, 53]. Characterizing traffic patterns of groups of users in emerging regions has only been performed in a few isolated cases at Internet cafes or kiosks in rural areas [13, 26, 33, 20]. One recent attempt has been made to analyze web traffic for emerging regions at larger scales using Content Distribution Networks (CDNs) [21]. However, gathering traffic statistics using client opt-in CDNs biases the sample population since the traffic traces themselves only contain requests that use the CDN.

The lack of adequate basic research on web traffic in these underserved regions is somewhat expected due to the many deployment challenges in these areas [7]. However, a large and growing number of initiatives, systems, applications, and services are targeted at emerging regions each year [11, 4, 40, 30, 39]. Web use in schools is particularly important to study. Education is a central focus of development initiatives [48]. Also, digital literacy is increasingly considered as a requirement for getting a good job in countries such as India, Kenya, and Ghana. Internet access is viewed as a bootstrapping mechanism for both education and e-literacy. Because of this, millions of dollars are poured into connecting schools in developing regions to the Internet, with donations from governments, international organizations, non-profits, and private sponsors. Without a basic understanding of the needs of the target population many efforts are at best misguided and a waste of resources that could be better spent elsewhere.

The technical challenges for providing ubiquitous Internet access are now relatively clear. Beyond improving network infrastructure, many ideas and several systems have been implemented to address connectivity challenges. Solutions exist for problems such as power outages as well as intermittent and last-hop network connectivity [46, 15, 35, 47, 9, 4]. It is also understood that even after a connection is established, the quality is by no means adequate; basic problems such as high cost per byte, low bandwidth per person, and high latencies plague even large institutions such as universities that can afford a broadband connection [9, 47, 8]. To make matters worse, modern web pages are designed for dynamic and media-rich content which places greater demands on the network. In conjunction these issues result in an extremely poor user experience in emerging regions, i.e., long and variable page load times, and non-functional websites.

Web acceleration techniques such as caching, compression, and prefetching have been applied in the past to address exactly these bandwidth and latency constraints. The work most closely related to ours is that of Isaacman and Martonosi, who propose aggressive techniques such as caching stale pages and client-side prefetching and evaluate them on network traces [25] as well as via a real deployment in Nicaragua[24]. While their focus is on "collaborative caching" in which multiple clients can share cached pages, their techniques are similar to ours when restricted to a single machine. One of the primary differences between our works is the deployment environment: while we target a synchronous Internet connection, they deploy in an asynchronous delay tolerant network (DTN) [15]. Users behave differently using a DTN because web browsing is converted from a synchronous to an asynchronous activity, which has a non-trivial impact on browsing patterns [8]; in addition, some dynamic and interactive content remains difficult to deliver over DTNs. One contribution of our work is to show that these techniques offer benefits even in a connected environment. We also improve the ease of installation and deployment by packaging our system as a browser plugin, rather than requiring changes to the proxy settings. Finally, our deployment incorporates data from a longer period (six weeks as opposed to five days [25]), enabling detailed analysis. While we later make comparisons regarding the specific results obtained, we emphasize that Isaacman and Martonosi's setting is different across several dimensions, including the deployment environment, user population, and task assignment.

A wide variety of caching algorithms exist [54, 37], and recent research targeting emerging regions has looked at caching architectures for affordable hardware [3]. Tangentially related to our work is Opera Mini, which is designed to accelerate web browsing for mobile devices and primarily focuses on page rewriting and compression, which require an additional proxy component [32]. Prefetching systems are also well studied, but are also dated and untested in this context and require access to a proxy or server support [16, 10, 34, 14]. Furthermore, emerging region specific ideas such as proxying, text-only browsing, offline browsing, and collaborative caching have been suggested and implemented but specialized systems are often difficult to deploy and scale up because they require additional resources or expertise [47, 9, 28, 25, 50, 41, 22].

Lower in the networking stack, transport or session layer protocols such as SCTP [42], SST [45], and SPDY [44] have also been implemented and experimented with, but they are designed as performance optimizations for high speed Internet connections. Internet statistics reports from sources such as Akamai [2] and the International Telecommunications Union (ITU) [23] indicate that, for a variety of reasons, broadband growth in many areas of the world including most of Africa and South America continues to lag behind. As a result, optimizations that focus on slow Internet connections will continue to be relevant, particularly if the optimizations

are deployable with low levels of expertise. Finally, web acceleration techniques that are browser based have been implemented and deployed extensively [19, 17, 12], but to our knowledge no formal study in any emerging-region context has been conducted to evaluate how well these techniques actually perform.

3. THE SCHOOL ENVIRONMENT

3.1 Infrastructure

The study was conducted in a primary school outside of Bangalore, India. We characterize the environment as "peri-urban" because it is situated amongst farmland about 12 miles north of the city center, translating to about a one-hour drive in typical conditions. While the quality of the teachers and the facilities are above typical Indian standards, all the students come from households that earn at or below the poverty level. The school is a kindergarten through 10th-grade institution with 922 students and 51 teachers. As a school that is close to one of the most well-connected cities in India, access to Internet connectivity is relatively good. However, it is still susceptible to unplanned loss of power and Internet connectivity. This is mostly due to either nearby construction work or infrastructural deficiencies on the service provider's end. In such situations, power from UPS systems is made available to the computers of a few important staff members who then access the Internet via satellite data cards.

Under normal operation, the school shares a 2 Mbps link between 97 computers all running Microsoft Windows XP. Of the 97 computers, students have access to 2 labs running 58 machines; the teachers share 11 computers situated in their lounge room; and the rest of the computers are used by administrative staff. All Internet traffic goes through a gateway machine kept in a secure room on the school premises. Only the technology vendor for the school has access to the gateway machine. The Internet connection costs USD 180.00 a month for an upload bandwidth of 0.34 Mbps and download of 0.85 Mbps. Additional usage costs USD 0.004 per megabyte. The typical use rate for this school is 8 gigabytes downstream out of an available 10 gigabytes per month. Metered bandwidth imposes a constraint for how one optimizes Internet access in emerging markets; for example, undisciplined prefetching may cause the bandwidth limit to be exceeded and service to stop altogether.

3.2 Internet Usage

Prior to the experiment, we conducted a survey to determine the nature of Internet usage at the school. We limited our study to the senior computer lab, which has 40 machines and serves 5th-10th grade students and the teachers' lounge, which has 11 computers. Previous observations indicated that these were the most active users of the Internet. There were 30 participants in the survey (11 teachers and 19 students). From the survey results, the sampled students reported using the Internet only 1-4 times per week, on average. This was different than the teachers, who reported daily usage. Both students and teachers reported that, on average, each online session lasted less than an hour.

This means that the students only use computers and access the Internet when they are at school and only during specified computer class times. Only senior students can have unsupervised access to the computers outside scheduled class sessions. We hypothesized that this time-limited and structured access to the Internet would influence the sites visited by the students. This was supported by the survey results where the students reported frequently visiting web-based email sites and accessing curriculum-related websites, but rarely blogs, news, shopping and social networking websites, which are otherwise popular among children their age. The teachers also frequented web-based email and education sites more often than the other types of sites. Further interviews revealed that the high webmail use was a result of a school-wide policy for teachers to share their weekly lesson plans with their heads of departments only through email. The policy was in place to encourage teachers to learn and use the Internet. Both students and teachers also reported visiting video sites – this was the third highest reported destination.

When queried about their response to slow connectivity, both teachers and students indicated that they got frustrated and considered it to be a waste of time waiting for a page to load. In such situations, they reported that they coped by loading only one page at a time. Only one person remarked that they used the browser's features, such as disabling images or javascript, to speed up the page load. When asked about offline browsing, none of the respondents reported having used it.

Although the observations and survey focused on the most active Internet users in the school, we believe it to be representative for the rest of the school, and to a large extent, other schools in the area. Ultimately, understanding the costs of connectivity, the patterns of use and the nature of the browsed content provides us with a more informed lens to analyze and improve web access in these types of environments.

4. METHODOLOGY

In order to analyze and accelerate web traffic in the school environment, we implemented a custom tool: the Event Logger for Firefox (ELF). ELF is a free and open source plugin for Firefox that enables logging of Internet traffic as well as web acceleration using aggressive caching and prefetching. The complete ELF extension was implemented in approximately 2,000 lines of code (including comments and blank lines). We describe the requirements and capabilities of this tool in the remainder of this section.

4.1 Traffic Analysis

We initially intended to install a logging mechanism at the school gateway proxy or simply copy the logs from that machine for analysis. After some discussion with the administration we discovered that the network maintenance was being outsourced to an external company, and we were not allowed access to the gateway machine. This may seem unexpected at first, but we have found that being denied access to existing system critical resources such as the network infrastructure or the gateway is common practice at most large institutions.

The ELF plugin for Firefox provided an alternative infrastructure to gather HTTP traces. We were given access to two computer labs at the school where we could install Firefox version 3.6.6 and ELF. The first lab was for students and contained 40 machines, 30 of which were in working order. The second lab contained 11 machines, 7 of which were working. We did not log individual user activity because not

Table 1: Information and statistics logged by ELF. Each per-event log entry is associated with a timestamp, Window ID, and Event ID.

Information	Logging Rate and Accuracy
IP Address Unique Machine ID Disk Cache Size Clock Skew	Per Digest
HTTP REQUEST / POST - URL - Referer HTTP Header - X-Moz: Prefetch HTTP Header	Per Event
HTTP RESPONSE - URL - HTTP Response Code - Content-Type - Content-Size	Per Event
Page Start Load / Page Stop Load - URL	Per Event
Network Online/Offline	Per Event (5 minute)
User Idle/Back	Per Event (1 minute)

every user had a unique login or used the same machine on a day to day basis. The school contained another student lab, and several other machines scattered across offices on the campus.

The log format was in simple plaintext which was flushed to disk or uploaded to a central server in a digest every 5 minutes. If the reporting server was unreachable the logs would remain on disk until it could be contacted again. The logs stored on disk were persistent across browser sessions. We elected to upload the logs to our server for convenience, but this could be disabled and the logs could be copied manually if upstream bandwidth were a concern.

ELF recorded and reported information on a per digest and per event basis. Each digest included aggregate statistics such as cache usage and clock skew. Events such as web requests, responses, cache hits, user idle, and network offline. ELF is configured not to log cookies or any other headers that may be a privacy concern. The complete set of information recorded by ELF is in Table 1.

4.2 Web Acceleration

In addition to understanding web usage in the context of a peri-urban school in an emerging region, we were interested to explore how much web acceleration techniques recommended by prior work would actually improve the user experience. In addition to logging we implemented several of these techniques in ELF.

Caching without Expiration

The first web acceleration technique that we implemented was to sacrifice freshness for speed and offline availability. The basic idea behind this technique is to allow the presentation of potentially stale cached content to users to reduce the amount of traffic transferred over the network. This idea was recommended 4 years ago [13] and deployed in the C-LINK system in Nicaragua [25]; however, until now it is not been bundled as an easily deployable browser extension. To implement this technique, ELF performs several modifications to Firefox's default behavior:

- ELF sets the *check_doc_frequency* to 2 meaning Firefox checks online for an updated version of a file only if it is not in the cache. The Firefox default is 3 which means Firefox checks for an updated file if the cached version is expired.
- ELF changes the cache behavior by extending the cache duration of each file by an extra 30 minutes. The "no-cache" headers in all web responses are also ignored and the expiration is set to 30 minutes. While the expiration notices are ignored by ELF (as per the previous point), they are still important for maintaining the correct cache eviction order used by Firefox.

Prefetching

The second acceleration technique that we implemented was prefetching. The state-of-the-art prefetching algorithms described in the research literature attain high accuracies (over 70% of pretched pages are accessed by the user) [16, 12]. These algorithms typically require deployment at a gateway proxy or server support to achieve these accuracies. Proxies and servers are advantageous because they have more and correlated information to determine browsing patterns. In contrast, off-the-shelf prefetching techniques such as those implemented in WWWOFFLE [50] and Fasterfox [17] perform client-side prefetching which does not require any additional support. Without access to additional resources, proxy-based and server-initiated prefetching and other proxy-based techniques were not deployable in our setting. Instead, we implemented a simple client-based prefetching algorithm based on an existing prefetching extension named Fasterfox.

The basic prefetching algorithm attempts to download files in anchor links on each page loaded by the user. We extended this algorithm to also keep track of the prefetched frontier of web pages, download their embedded objects, and subsequently linked files as well. These prefetches were performed in breadth first search (BFS) fashion where the files at each depth in the BFS tree were ordered in last in first out (LIFO) order to track the latest potentially useful pages for the user. Finally, prefetching only occurs while the user is not actively downloading any pages. We note that even with our modifications, our client-based prefetching algorithm is substantially less accurate than proxy-based or server-initiated prefetching. While proxy-based and server-initiated prefetching would likely perform better, they require more hardware resources and deployment complexity. The main benefit of our client-side prefetching algorithm is that it is easily to deploy.

Offline Browsing

The third and final technique that we implemented was offline browsing[1]. While some form of offline browsing has been implemented previously [50, 47, 9] there has not been a formal evaluation of its effectiveness. To support offline browsing, ELF modifies Firefox in the following ways:

[1]In a preliminary visit to the school, we discovered that the Internet had been down an entire week due to disconnection somewhere upstream. We were therefore very interested in whether offline browsing could benefit the school.

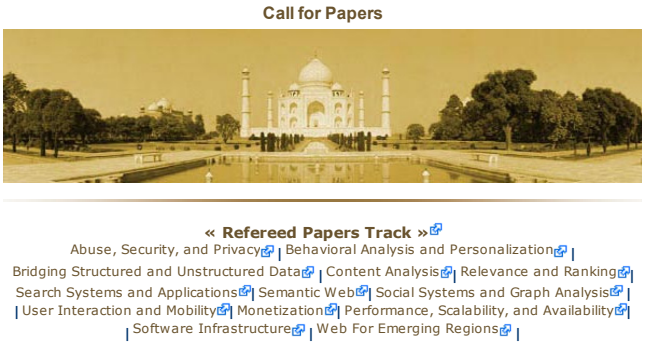

Figure 1: Offline page rendering with ELF cached link indication (light blue external link icons).

- ELF sets the disk cache size to 500 MB uniformly across all machines to increase the amount of offline browsing possible. The default cache size for our version of Firefox was 50 MB. Any existing files in the cache were left in place, and no changes to the other cache size settings were made.
- ELF modifies the "Cache-Control" header for all web responses in Firefox so that the "max-age=31536000" which indicates that files should be evicted only after they are 1 year old.
- ELF checks the machine is online or offline by sending a HEAD request to a popular search portal every 5 minutes. If the host was unreachable, ELF would put Firefox in offline mode.
- In offline mode, ELF modifies the rendering of all pages using a style sheet to indicate the links that refer to pages that exist in the cache to assist users in avoiding dead links while browsing offline. A screenshot of the offline browsing modification is in Figure 1.

Additional Techniques

Other techniques suggested by previous work involving a proxy server were not investigated here due to the lack of access to the gateway proxy at the school. Blacklisting and other filtering was also not implemented (though we easily could have) to avoid contaminating our traffic analysis results. We note that the techniques we implemented are highly synergistic with each other both in terms of purpose and their respective modifications to Firefox.

5. TRAFFIC ANALYSIS

In this section we present some analysis of the traffic traces. We find several well-known results that corroborate with existing web literature, some that corroborate with web in developing regions literature, and some findings that are completely novel. In this and the next section we compare our results with the two pieces of prior research that are the most closely related to ours by Du et al. [13] and Johnson et al. [26].

Our logs were taken over a 48 day period between September 6 and October 24, 2010. New data continues to be captured for analysis. Our trace contains 1,723,146 log events with 665,571 HTTP requests by users, of which 325,037 were downloaded. There were 207,011 cache hits, 37,809 unresponded requests, and 95,714 (canceled/incomplete/failed). A total of 178,634 unique URLs were requested from 5,025 unique domains and subdomains, amounting to over 13.3 GB of data. Note that since we modified the caching behavior of the clients, the caching numbers reflect those changes. Also, prefetched objects are not included in any of the results in this section.

Table 2: Top requested domains or sub-domains.

Domain or Sub-Domain	Requests
mail.google.com	16.71%
www.google.com	4.35%
*.fbcdn.net	4.52%
*.google.co.in	3.64%
safebrowsing-cache.google.com	1.67%
*.doubleclick.net	2.53%
js.geoads.com	1.43%
ad.yieldmanager.com	1.41%
*.yimg.com	2.16%
www.google-analytics.com	1.00%
other	59.58%

5.1 Time and Place of Access

Files downloaded over a period of 2 weeks are shown in Figure 2. We see the expected diurnal pattern which is truncated each day due to the hours the school is open. Sundays also exhibit no activity. This pattern was largely consistent except between October 10 and October 17 when nearly no activity occured due to a holiday.

The usage level of the machines is split between the two labs. The machines in the teacher's lab $(31 - 36)$ viewed approximately 1 to 3 orders of magnitude more pages than those in the student lab $(1 - 30)$ in terms of web pages requested. The per byte results are similar. Less than 10% of requests failed or were incomplete which is lower than that in the 12.5% observed by Johnson et al. at a rural village in Zambia.

5.2 Internet Destinations

The distribution of requests across domains or sub-domains is shown in Table 2. Web mail, search portals, and advertising dominate the number of requests. Unlike recent work at rural Internet cafes by Johnson et al., webmail requests dominate rather than Facebook. We assume that this is simply due to the difference in demographic. Unlike the previous result, the demographic of our users is strictly school faculty and students and is unlikely to contain traffic from international visitors. Prior work by Du et al. found more portal requests than mail and advertising requests. In our trace, mail requests were the most frequent followed by portals then advertising.

5.3 Classification by Type and Size

The MIME types of all files downloaded are tabulated in Table 3. The proportion of images to html/plaintext is similar to those in previous studies in rural Internet cafes and kiosks. However, the dominance of video 72.15% is perhaps surprising for an environment without high-bandwidth Internet access. Looking more closely at the video traffic, we found that like in rural Internet cafes, the videos are mostly

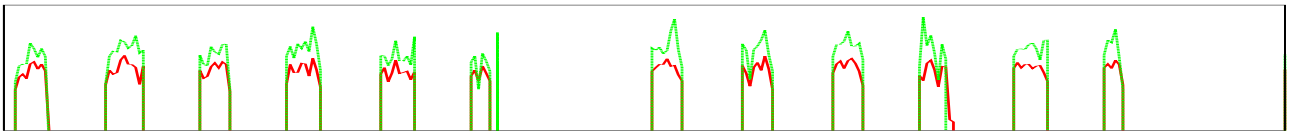

Figure 2: Semi-log plot of requests/hour (red) and KBytes/hour (green) over a period of 2 weeks. The expected diurnal pattern is clear for days when school was in session.

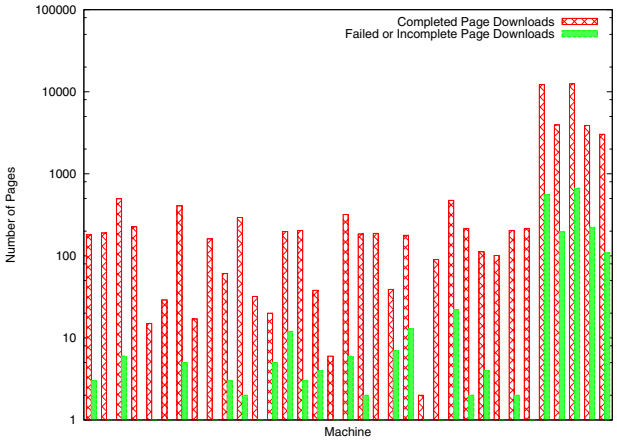

Figure 3: Page loads per machine.

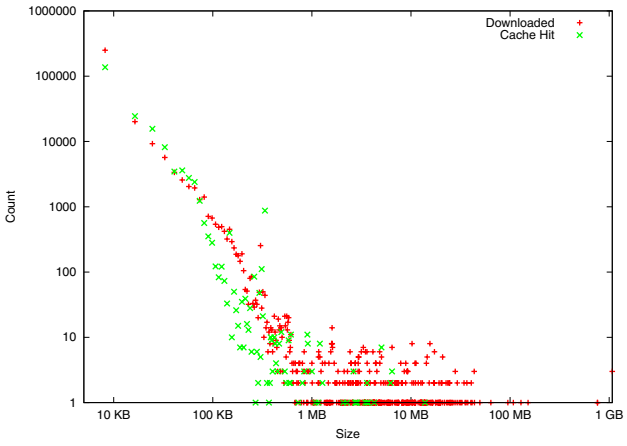

Figure 4: Logarithmic plot of file size distribution.

Table 3: Traffic by MIME Type.

Mime Type	Requests	Bytes
images	**47.46%**	14.11%
html/plaintext	22.30%	1.58%
javascript	20.49%	2.61%
other	5.79%	5.01%
css stylesheet	2.59%	0.43%
shockwave/flash	1.91%	1.83%
video	0.19%	**72.15%**
audio	0.14%	1.20%
msoffice	0.09%	0.31%
pdf	0.04%	0.71%
compressed	$< 0.01\%$	0.06%
binary	$< 0.01\%$	$< 0.01\%$

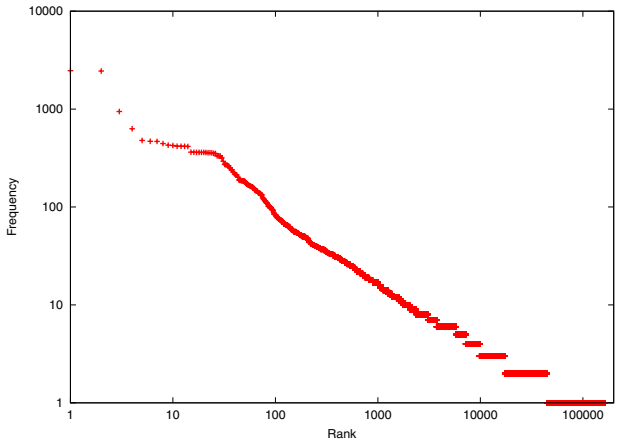

Figure 5: Logarithmic plot of URL frequencies by rank. The expected Zipf-like distribution is clear.

from google.com or youtube.com. A total of 360 YouTube videos were requested. Of the 360 YouTube videos, 190 were unique and the most popular video was requested 10 times. As with previous results, YouTube video traffic could be dramatically reduced if the videos were cached at the gateway proxy. The actual contents of those videos include educational content, news, TV programming, music videos, and arts and crafts.

Compared to prior work we also found more javascript files downloaded (20% of requests) and nearly no binaries. The amount of shockwave/flash and audio files downloaded is also still relatively low. Finally, the number of MS Office and PDF documents downloaded is negligible.

Figure 4 shows a logarithmic graph of the size distribution of the URLs downloaded and cache hits at the clients aggregated into 8 KB buckets. We find very similar results to those found by Du et al., with a couple of notable exceptions. We find fewer objects in the midrange around 1 MB, and there are also a few outliers of large files around 1 GB in size that were video downloads. We also observe that the cache hits roughly track the downloaded file patterns but are generally smaller. Also, the largest cache hit is only approximately 14 MB in size.

Figure 5 shows a logarithmic graph of the URL frequencies by rank. The distribution is roughly in line with those in previous studies, with a Zipf-like distribution. We hypothesize that the deviation from the Zipf distribution at the highest ranked URLs $(1 - 100)$ may be due to our smaller population size.

5.4 Web Performance

We found that the average time taken for a page to load including rendering time was 3.69 seconds, with a standard deviation of 9.53 seconds. The minimum was 0 seconds, and the maximum was 366 seconds. We also found that the variability is larger across the student machines due to older hardware and presumably malware infections. However, due to the higher number of requests by the teachers' machines, our results are dominated by those faster page load times. This is an interesting result that is not found in most studies due to proxy-based logging mechanisms which cannot capture complete client browser statistics. We take a closer look at the factors that affect page load time in the next section.

6. WEB ACCELERATION RESULTS

While most caching and caching optimizations are simulated or implemented at the gateway proxy, we follow the approach of implementing the caching at the client itself [25, 24]. There would undoubtedly be a higher cache hit rate, better prefetching accuracy, and better offline browsing if our techniques were implemented at the gateway proxy. Our numbers should therefore be interpreted as a lower bound for each of these web accelerations if implemented at a proxy. However, our results also represent a realistic measure of what is possible without proxy support which in our experience is often the case due to infrastructure constraints, security and privacy concerns, and lack of local technical support. We briefly relate our results to previous findings by Isaacman and Martonosi [25] where it is illuminating, but reiterate that any direct comparison between two different systems in different environments and usage scenarios should be made with care.

6.1 Caching

Modifying caching to tradeoff freshness for speed is one of the main web acceleration techniques suggested for constrained networks. We find that with all of the modifications, our overall cache hit rate is 31.1%. The average cache size over the duration of our trace was 110 MB with a standard deviation of 85 MB, a minimum of 0 MB, and a maximum of 383 MB.

Recall that ELF modified the nocache and serves pages from the cache regardless of their expiration time. Compared to Du et al.'s work simulating the aggregate cache hit rate from 6 community information centers the cache hit rate is similar to the unmodified cache rate (approximately 43%) for a cache of the same size (500 MB). Our cache hit rate is relatively high given that the previous simulation was from a trace both 5 years old where dynamic content was less prevalent, and also for aggregate traffic across a set of machines rather than a single client. While we stress that any direct comparison is tenuous, we observe that our cache hit rate 31.1% is higher than that of the local cache-hit rate observed by Isaacman and Martonosi of up to 19%, and lower than their combined collaborative and local cache-hit rate of 78% [25].

Table 4 shows the cache hit rate broken down by MIME type. Comparing this table with Table 3 we can examine the content types for which caching performs well, and those where caching could be improved. We observe that the percentage of cache hits for javascript and shockwave/flash are markedly higher than their counterparts for downloaded MIME types. Videos from YouTube and some other sites are not being cached at all resulting in only 4.56% of cache hits being video content. Images and html/plaintext are not cached as often as they are requested, which indicates that the cache could be improved for those content types. In terms of bytes, images and html/plaintext are being cached more than in Table 3, but this result is most likely due to inflation caused by the lack of video cache hits.

Table 4: Cache hits by MIME Type.

Mime Type	Hits	Bytes
images	**35.97%**	24.16%
javascript	34.18%	**34.79%**
html/plaintext	14.77%	4.53%
css stylesheet	8.14%	4.53%
shockwave/flash	4.75%	24.97%
other	2.14%	1.71%
video	0.03%	4.56%
msoffice	0.02%	0.22%
pdf	0.01%	0.36%
audio	$< 0.01\%$	0.17%
compressed	$< 0.01\%$	$< 0.01\%$
binary	$< 0.01\%$	$< 0.01\%$

To measure the impact of caching on the user experience, we compare the load times for URLs that were accessed in both cached and uncached states. As a webpage can contain many embedded objects (such as images), we consider a page to be "completely cached" if both its content and all of its embedded objects are in the cache (this accounts for 35% of total page requests). Likewise, a page is "completely uncached" if neither its content nor any of its embedded objects are in the cache (this accounts for 40% of page requests). In our trace, we identify 716 URLs that were requested in both completely cached and completely uncached states. Comparing the load times for these two states yields an estimate of the performance impact of caching.

Our results are as follows. On average, each page requires 3.3s to load when completely uncached, but 0.36s to load when completely cached. In other words, caching a page and its contents leads to a 9.1x speedup in web access, on average.[2]. Given that the overall cache hit rate for objects is 31%, one could expect caching to offer a proportionate speedup of the user's overall experience: in this school environment, **our tool offers a general speedup of 2.8x across all HTML pages**. We note that this figure does not apply to video content, however, since very few videos are cached; video represents less than 1% of the page requests but 72% of the bytes transferred.

6.2 Prefetching

The number of prefetched files in our trace was 95,714 files compared to 325,037 files downloaded by user initiated requests. Despite the aggressive prefetching algorithm we implemented, only 22.7% of all downloaded files were prefetches. Since our prefetcher only downloads files when users are inactive (i.e., when there are no pending requests originating directly from the user), this relationship also in-

[2]While we presented the speedup using the arithmetic mean, using the geometic mean suggests a comparable speedup of 10.0x.

Table 5: Prefetched Files by MIME Type.

Mime Type	Hits	Bytes
html/plaintext	**93.22%**	**55.36%**
images	2.85%	7.58%
other	2.66%	1.90%
pdf	0.62%	17.06%
shockwave/flash	0.58%	0.33%
audio	0.04%	9.22%
msoffice	0.03%	0.69%
video	$< 0.01\%$	2.67%
compressed	$< 0.01\%$	5.19%
javascript	0.00%	0.00%
css stylesheet	0.00%	0.00%
binary	0.00%	0.00%

dicates that users spent a significant fraction actively waiting for pages to load.

The prefetching accuracy is defined as the percentage of prefetched files which later result in a cache hit. We found the prefetching accuracy in our traces to be 16.13% over the course of six weeks. This figure is low as expected from a single client only prefetching algorithm whereas Isaacman and Martonosi found that with collaborative prefetching achieved 48% accuracy over the course of two days. The percentage of cache hits due to prefetching in our system is only 1.8%. This figure corroborates closely with previously observed results of 2% by Isaacman and Martonosi [25]. These results suggest that while prefetching more pages may improve the percentage of cache hits due to prefetching, the limiting factor to client-side prefetching is still the accuracy of the algorithm.

The prefetched files by MIME type are shown in Table 5. Our simple client-based prefetching algorithm only downloads the links on the pages viewed by the user, and only after all links on the page are retrieved are the embedded objects on the linked pages downloaded. As new page requests from the user come in, new links are added, and the user likely browses too fast for the prefetching algorithm to download all of the links and start downloading the embedded objects on the linked pages.

Our results suggest that if more bandwidth were available or a more intelligent algorithm were employed, it would be beneficial to prefetch images and javascript files. More advanced prefetching algorithms are outside the scope of this work, but one simple improvement for future work would be to estimate the likelihood that a user will follow a particular link on a page, and to prefetch content from the link with a priority that is in proportion to that likelihood.

As with our caching results, what we are really interested in is the impact of prefetching on the user experience. Unfortunately we do not have enough data points to do a rigorous analysis of completely uncached versus completely cached (and prefetched) pages, though we illustrate the general relationship between load time and percentage of prefetched embedded objects in in Figure 6. We can observe from these results that prefetching is helpful if files are chosen properly.

6.3 Offline Browsing

While our original impression was that the network connectivity was frequently down in the school, our logs indicated very few periods when the browser was used in offline mode. We did log a negligible number of cache hits (12) during offline browsing, though only 8 machines were used while offline, for a total of two hours of logged outages. We intentionally did not train students and teachers regarding offline mode, so they might not have attempted to browse cached pages while the network was down. However, even if they had attempted to use the browser, with the current rate of prefetching they could expect to follow at most 1 or 2 links before requesting pages that were unavailable.[3] We hope to revisit offline browsing in other environments in future work.

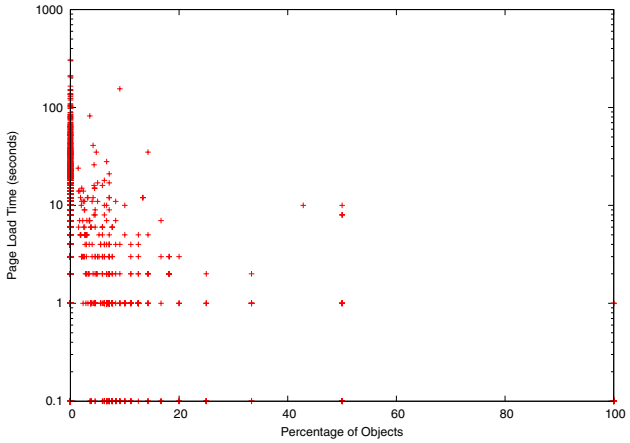

Figure 6: Page load time versus per-page prefetch rate (fraction of requests that hit in the cache and the file was cached due to a prefetching event).

7. DISCUSSION

In this section we briefly discuss some of the implications of our results, other unimplemented optimizations, and directions for future work.

From our results, freshness vs speed is a good tradeoff for users behind constrained network connections. We realize that this tradeoff violates RFC2616 [18], but, as suggested by prior work, users in these settings are unlikely to be interested in minute changes in content on sites other than news, webmail, and Facebook [13]. We also found that prefetching, while beneficial, could be further improved, and client-end prefetching algorithms are an interesting avenue for future work. We found that offline cache browsing is only useful for visiting previously viewed pages. While some prior work has suggested that less experienced web users perform history-based browsing [38], this appears to be a corner case in our setting. We hypothesize that another shortcoming of offline browsing is that the rate at which pages are prefetched or cached cannot keep up with the pages browsed while offline. Under a constrained setting, a user will eventually get a cache miss and fail to make progress. Local search and offline content aggregation are promising options to address this shortcoming of offline browsing [8]. Also, we still consider highlighting dead links as a useful user interface feature given the caveats above.

[3]Isaacman and Martonosi make exactly this observation in their often offline C-LINK deployment.

There are several other optimizations that we considered, but were unable to or did not wish to implement in this work to avoid contaminating our traffic results. Time shifting (e.g., downloading large volumes of content overnight for possible viewing the next day) is not feasible in this setting due to limits on monthly bandwidth consumption. If bandwidth were not a concern, time shifting the prefetches or downloading even while user requests were taking place could be helpful. Text-based browsing requires a proxy server to pre-render pages prior to transferring them over the bottleneck link. The client could potentially interface with such a proxy, and this is an area for future work. Compression requires server support which is unlikely to be ubiquitous until web server software comes pre-configured to use compression. Caching of dynamic content would be tremendously useful for offline browsing and improving cache hit rate, particularly with our caching modifications. In the research literature, existing dynamic web caches exist, but they are primarily focused on server-side or proxy-based implementations. Leveraging the existing implementation in most browsers to "save webpage" could be one easy solution. Blacklisting of advertisement files would improve performance, but filtering pages would have severely contaminated our traffic results. We will be incorporating or deploying the adblock extension in the near future [36].

Finally, there were two optimizations that could have been useful that we did not implement, but have been studied independently. Collaborative caching across networks behind upstream bottlenecks such as ours could have been a way to gain the greater benefits of caching without access to the gateway proxy [25, 24]. Caching partially downloaded or chunks of content could also be useful particularly for YouTube videos and highly synergistic with cooperative caching. We hope to add these caching optimizations to our extension in the next version. While the prevalence of video may appear to undermine the benefits of our optimizations, we note that the video downloads are too slow to be viewed in real-time. This suggests that people are loading the videos first and watching them after completion, and in terms of user experience, these load times are not as disruptive as those during a more real-time task such as web browsing.

On the clients themselves, browser modifications are an ideal choice for the many deployment constraints in emerging regions. Beyond the deployment and scaling issues we have discussed in Section 2, awareness of the solution itself is also a problem. Even after the extension is made available for free download it is unlikely to be discovered by a local sysadmin who does not read research papers! To overcome this obstacle, the extension we implemented in this work would ideally be integrated into the browser itself so that the software comes pre-packaged with constrained network web optimizations. To address the lack of user expertise, the browser could track page load times and turn on the various web optimizations as the user experience starts to degrade. Users of the browser with good connections or tech savvy users who elect to turn off auto-tuning would not be affected.

We do not claim that our extension completely solves all of the web access issues behind constrained networks, but we do believe it covers a wide variety of options deployable at clients and is an important first step. Our results and prior work indicate that the optimizations requiring gateway proxy support are likely to be even more beneficial. Such optimizations could be deployed by extending off-the-shelf web proxies such as Squid.

8. CONCLUSION

In our traffic trace we found many similarities to prior work from emerging regions, but also some interesting differences due to either the unique demographic, urban setting, or more recent data. We discovered a dominance of videos (72% by bytes), and significantly more javascript files (20% by requests) than previously measured. We also found that webmail was the most common request followed by portals and then advertisements.

We implemented many of the web accelerations suggested previously as a Firefox plugin – freely available and easily installed on any client machine. We found that serving stale pages works well in practice and prefetching helps to some extent, though offline browsing was not useful in the context of the school. We quantified the benefits offered by our tool via a six-week deployment, demonstrating a cache hit rate of 31% and accelerated viewing of cached pages by 9.1x. Taken together, this implies an average acceleration of 2.8x for the user in browsing (non-video) web pages.

Overall, our primary contribution is an easy-to-use tool, requiring no expertise on the part of system administrators, that leverages prior techniques to have demonstrable impact in a resource-constrained educational setting.

9. REFERENCES

[1] Private Conversation with Faculty Regarding Network Outages.

[2] Akamai. http://www.akamai.com/.

[3] A. Badam, K. Park, V. Pai, and L. Peterson. Hashcache: Cache storage for the next billion. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 123–136. USENIX Association, 2009.

[4] A. Balasubramanian, Y. Zhou, W. Croft, B. Levine, and A. Venkataramani. Web search from a bus. *Second Workshop on Challenged Networks(CHANTS)*, pages 59–66, 2007.

[5] T. Bray. Measuring the web. *Proceedings of the Fifth International World Wide Web Conference*, 1996.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. IEEE, 2002.

[7] E. Brewer, M. Demmer, M. Ho, R. Honicky, J. Pal, M. Plauche, and S. Surana. The challenges of technology research for developing regions. *Pervasive Computing, IEEE*, 5(2):15–23, 2006.

[8] J. Chen, A. Dhananjay, S. Amershi, and L. Subramanian. Comparing Web Interaction Models in Developing Regions. In *Proceedings of the 1st Annual Symposium on Computing for Development (DEV)*. ACM, 2010.

[9] J. Chen, L. Subramanian, and J. Li. RuralCafe: web search in the rural developing world. In *Proceedings of WWW*, pages 411–420, 2009.

[10] X. Chen and X. Zhang. Coordinated data prefetching by utilizing reference information at both proxy and web servers. *ACM SIGMETRICS Performance Evaluation Review*, 2001.

[11] M. Demmer, B. Du, and E. Brewer. TierStore: A Distributed Storage System for Developing Regions. *FAST*, 2008.

[12] J. Domčnech, J. Gil, J. Sahuquillo, and A. Pont. Web prefetching performance metrics: A survey. *Performance Evaluation*, 63(9-10):988–1004, 2006.

[13] B. Du, M. Demmer, and E. Brewer. Analysis of WWW traffic in Cambodia and Ghana. *Proceedings of WWW*, pages 771–780, 2006.

[14] D. Duchamp. Prefetching hyperlinks. In *Proceedings of the 2nd conference on USENIX Symposium on Internet Technologies and Systems-Volume 2*, 1999.

[15] K. Fall. A delay tolerant network architecture for challenged internets. *Proceedings of SIGCOMM*, 2003.

[16] L. Fan, P. Cao, W. Lin, and Q. Jacobson. Web prefetching between low-bandwidth clients and proxies: potential and performance. In *Proceedings of the 1999 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 178–187. ACM, 1999.

[17] FasterFox. http://fasterfox.mozdev.org/.

[18] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. RFC 2616: Hypertext transfer protocol–HTTP/1.1, June 1999. *Status: Standards Track*.

[19] Google Web Accelerator. http://webaccelerator.google.com.

[20] E. Huerta and R. Sandoval-Almazán. Digital literacy: Problems faced by telecenter users in Mexico. *Information Technology for Development*, 13(3):217–232, 2007.

[21] S. Ihm, K. Park, and V. Pai. Towards Understanding Developing World Traffic. In *NSDR 2010 4th workshop on networked systems for developing regions*.

[22] S. Ihm, K. Park, and V. Pai. Wide-area Network Acceleration for the Developing World. In *Proceedings of the USENIX Annual Technical Conference (USENIX.10)*, 2010.

[23] International Telecommunication Union. http://www.itu.int/.

[24] S. Isaacman and M. Martonosi. Potential for collaborative caching and prefetching in largely-disconnected villages. In *Proceedings of the 2008 ACM workshop on Wireless networks and systems for developing regions*, 2008.

[25] S. Isaacman and M. Martonosi. The C-LINK System for Collaborative Web Usage: A Real-World Deployment in Rural Nicaragua. In *NSDR 2009 3th workshop on networked systems for developing regions*, 2009.

[26] D. Johnson, E. Belding, K. Almeroth, and G. van Stam. Internet usage and performance analysis of a rural wireless network in Macha, Zambia. In *Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions*, pages 1–6. ACM, 2010.

[27] M. Kamvar, M. Kellar, R. Patel, and Y. Xu. Computers and iphones and mobile phones, oh my!: a logs-based comparison of search users on different devices. In *Proceedings of WWW*, pages 801–810, 2009.

[28] Loband. http://www.loband.org.

[29] T. Loon and V. Bharghavan. Alleviating the latency and bandwidth problems in WWW browsing. In *Proceedings of USENIX Symposium on Internet Technology and Systems*, 1997.

[30] W. Lu, M. Tierney, J. Chen, L. Subramanian, and B. Rao. SMSAppStore: SMS-based Mobile Applications Made Easy. In *Proceedings of the 1st Annual Symposium on Computing for Development (DEV)*. ACM, 2010.

[31] J. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy. Potential benefits of delta encoding and data compression for HTTP. In *Proceedings of the ACM SIGCOMM'97 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 1997.

[32] Opera Mini. http://www.opera.com/mobile/download/.

[33] B. Oyelaran-Oyeyinka and C. N. Adeya. Internet access in africa: empirical evidence from kenya and nigeria. *Telemat. Inf.*, 21(1):67–81, 2004.

[34] V. Padmanabhan and J. Mogul. Using predictive prefetching to improve World Wide Web latency. *ACM SIGCOMM Computer Communication Review*, 26(3):22–36, 1996.

[35] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer. WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks. *NSDI*, 2007.

[36] A. Plus. https://addons.mozilla.org/en-US/firefox/addon/1865/.

[37] M. Rabinovich and O. Spatscheck. Web Caching and Replication. *SIGMOD Record*, 32(4):107, 2003.

[38] A. Ratan, S. Satpathy, L. Zia, K. Toyama, S. Blagsvedt, U. Pawar, T. Subramaniam, and A. Ratan. Kelsa+: Digital Literacy for Low-Income Office Workers. 3rd IEEE/ACM International Conference on Information and Communication Technologies and Development, Doha, 2009.

[39] A. Reda, Q. Duong, T. Alperovich, B. Noble, and Y. Haile. Robit: An Extensible Auction-based Market Platform for Challenged Environments. In *Proceedings of the 4th international conference on Information and Communications Technologies and Development (ICTD 2010)*, pages 801–810. ACM, 2010.

[40] A. Reda, B. Noble, and Y. Haile. Distributing private data in challenged network environments. In *Proceedings of the 19th international conference on World wide web*, pages 801–810. ACM, 2010.

[41] U. Saif, A. Chudhary, S. Butt, N. Butt, and G. Murtaza. Internet for the developing world: Offline internet access at modem-speed dialup connections. In *Proc. Intl. Conf. on Information and Communication Technologies and Development (ICTD), India*, 2007.

[42] SCTP. http://www.sctp.org/.

[43] W. Shi, E. Collins, and V. Karamcheti. Modeling object characteristics of dynamic web content. *Journal of Parallel and Distributed Computing*, 2003.

[44] SPDY. http://www.chromium.org/spdy/spdy-whitepaper.

[45] SST. http://pdos.csail.mit.edu/uia/sst/.

[46] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, and E. Brewer. Beyond Pilots: Keeping Rural Wireless Networks Alive. *NSDI*, 2008.

[47] W. Thies et al. Searching the world wide web in low-connectivity communities. *Proceedings of WWW*, 2002.

[48] United Nations Millenium Development Goals. http://www.un.org/millenniumgoals/.

[49] A. Woodruff, P. Aoki, E. Brewer, P. Gauthier, and L. Rowe. An investigation of documents from the World Wide Web. *Proceedings of the Fifth International World Wide Web Conference*, 1996.

[50] World Wide Web Offline Explorer. http://www.gedanken.demon.co.uk/wwwoffle/.

[51] S. Wyche, T. Smyth, M. Chetty, P. Aoki, and R. Grinter. Deliberate Interactions: Characterizing Technology Use in Nairobi, Kenya. *CHI*, 2010.

[52] X. Xie, G. Miao, R. Song, J. Wen, and W. Ma. Efficient browsing of web search results on mobile devices based on block importance model. In *Proceedings of Pervasive Computing and Communications*, pages 17–26, 2005.

[53] J. Yi, F. Maghoul, and J. Pedersen. Deciphering mobile search patterns: a study of yahoo! mobile search queries. In *Proceedings of WWW*, 2008.

[54] L. Zhang, S. Michel, K. Nguyen, A. Rosenstein, S. Floyd, and V. Jacobson. Adaptive Web Caching: Towards a New Global Caching Architecture. *Third International Caching Workshop, June*, 1998.