Cost-aware Mobile Web Browsing

Sindhura Chava, Rachid Ennaji, Jay Chen, and Lakshminarayanan Subramanian

***Abstract*—The increasing pervasiveness of web browsing on portable devices calls for the careful consideration of how web pages are accessed by mobile users. While flat-rate pricing (independent of the usage) has been the conventional data pricing model in rich countries, mobile roaming users and users in developing regions are typically subject to usage-based pricing based on cost per megabyte downloaded. With the exceptional growth in both the size and complexity of web pages, usage-based pricing can result in high monthly bills even under nominal web usage.**

In this paper, we propose a cost-aware mobile web browsing mechanism that aims to substantially reduce the usage cost incurred for the end-user due to usage-based pricing. The key idea of our approach is to leverage the data plan of the user to compute a *cost quota* for each web request and a network middle-box to automatically adapt any web page to the cost quota. Given the complexity of existing web pages, we leverage a suite of content adaptation techniques to automatically produce the best possible version of a web page for a given quota. Based on a detailed analysis of popular web pages, we show that our system can significantly reduce data costs while extracting the best possible version of a page given the cost constraints.

***Index Terms*—content adaptation, cost-aware web browsing, user experience, pricing, web usage, optimization**

I. Introduction

A large fraction of users in developing regions are subject to usage-based pricing where the prices are significantly higher than flat-rate pricing rates found in developed countries. With the dramatic growth in the complexity of web pages, most usage-based priced users experience high monthly bills even with minimal use because web pages have significantly grown in size and complexity over the past decade. Since 1999, the average web page has increased in size by a factor of 22 (without including video) [6], and a $30-50$ fold increase in web page complexity (objects per page) has been observed [15], [11], [3]. In contrast, under many of the existing data rate plans for mobile users in developing regions, a single web download of a page with a size greater than 2 MB (which is very common) can easily cost over 1 USD. While web designers have resorted to manually redesigning low-bandwidth versions of web pages for mobile users, expecting a large fraction of the web to be manually redesigned for low-bandwidth mobile users is impractical.

In this paper, we describe a *cost-aware mobile web browsing* mechanism that aims to substantially reduce the cost of browsing the web on a mobile device. The key idea of our approach is to adapt web content as a function of the user's mobile pricing plan. Given the pricing plan of a user, the current data usage levels and limits, we dynamically compute a *cost quota* for each web request and use a network middle-box to automatically adapt any web page to the cost quota.

Cost-aware mobile web browsing is fundamentally different from conventional content adaptation techniques [21], [20], [9], [28], [25], [22], [26], [19]. Though mobile web browsing and content adaptation are well studied topics, a fundamental difference in our work in comparison is the notion of cost-awareness, which does not directly arise in existing works. In cost-aware adaptation, the user implicitly or explicitly specifies a quota for a page and our system automatically determines the level of adaptation that suits a specific quota. Based on the anatomy of a web page, we classify the components of a web page and selectively choose and adapt each component to arrive at the best version for a given quota. Cost-aware mobile web browsing also allows a user to explicitly augment the quota for a request to enhance the quality of the downloaded page by simply clicking on an embedded hyperlink (added by the system) to adapted pages. This provides a feedback system to tradeoff between message cost and content fidelity. Finally, our system requires no modifications to the client mobile devices except changing the HTTP proxy settings.

Based on a detailed analysis of our system on popular web pages, we show that our system can significantly reduce the size of downloaded data, and therefore the cost. On average, our system is able to reduce the cost of mobile web browsing by a factor of 2, and our most aggressive level of content adaptation reduces the cost by a factor of 110. We also show that, for each page request, our solution dynamically maximizes the number of web resources to send in order to forward the best possible version of a web page.

II. Related Work

There is a large body of work on web content adaptation [8], [17], [27], [12] and also specifically in the space of content adaptation for mobile devices [9], [26], [25], [18]. While our work leverages existing content adaptation techniques, the primary distinction of our work is the *cost-awareness* dimension. The cost quota (the data allowance) dictates the web page size that the user would want to download without having to pay the overage fee. Here, we outline the closely related works.

The use of proxies for web prefetching is not a new idea [16], [13], and a few developing region specific content adaptation systems have been developed in recent years [27], [12]. These systems do not take into account the ergonomic limitations of mobile devices. To adapt web content for mobile devices, a proxy-based solution in [9] uses RSS feeds, and Opera Mini [4] a commercial proxy-based system pre-renders pages.[1] However, none of these systems adapts content for mobile devices based on connection cost. Furthermore, these systems employ relatively cursory optimizations.

[1] Opera Mini is currently being used by 72% of mobile web users in Africa [5].

Most of the content adaptation work for mobile devices concentrate on content transformation to suit the display of smaller devices. Various content transformation processes like layout change and content format reconfiguration are used in [20] for adapting web content to mobile user agents. The idea of partitioning the web page into blocks (snippets), and processing the information only in the useful blocks by filtering out unnecessary ones is used in [21]. Some content adaptation solutions like those in [28] target only the multimedia content of the web pages. We have leveraged adaptation techniques from automated web re-authoring tools which takes the capabilities of the client and automatically re-authors the web page to fit the client's requirements [10], [14], [8]. Another related class of works focus on user interactive content adaptation where the system takes input from the user to adapt the content based on the user needs and change the adaptation based on user feedback [23], [24], [7].

III. Cost-Aware Mobile Browsing

The main idea behind *cost-aware* mobile web browsing is to associate each web page download with a *cost quota* that is reflective of the number of bytes that the user would need to download the page. Most mobile operators levy high usage charges for web access and in many countries mobile web download cost is very much usage driven. This is particularly true for roaming users. The problem is further exacerbated with the sudden growth in size of web pages over the past few years. This calls for web content adaptation on mobile devices where the page is tailored based on the cost needs.

In a cost-based web browsing model, a user should have the flexibility to specify the cost quota for a web page and the server should be able to provide a condensed version of the page that fits within the specified quota. In our design, cost-awareness is a user controlled metric and is independent of the original web page size. This is a policy decision that we made that emphasizes user control over incurred costs and is independent of the mechanisms used to adapt web pages to the quota. An alternative policy could be to allow the cost to be automatically determined by the system and proportional to the original web page size. Our cost-aware mobile web browsing model is driven by three basic design principles that try to achieve the right tradeoff between minimizing the cost and maximizing page fidelity:

Cost Quota: We require the user to provide both the data plan cost structure and a target budget over a fixed time-period (such as one month). We continuously monitor the web usage of the user and compute a cost quota for each new web request such that the user does not cross the target budget.

Content Adaptation Ladder: Given any web page, we analyze its structure and determine which version of the page is the best fit for a given quota. We derive a basic six-level content adaptation hierarchy for any web-page. The first three level are condensed text versions of the page. The next three levels of the page preserve the basic look of the page, but use different levels of filtering, adaptation and condensing. Based on the six-level content adaptation ladder, we pick the appropriate version of the page that fits within the cost quota.

User Feedback: When the cost quota is very low, the content adaptation can be too conservative for certain web pages and provide very little useful information to the user. In such cases, the user should have the option of requesting an enhanced version of the page. The system should also continuously provide information about current data consumption charges.

To better motivate cost-aware mobile web browsing, we next describe some of the existing mobile plans in developing regions and show how expensive even a single web download can be under the typical data plans. We then present the anatomy of existing web pages to illustrate their complexity and show how often a large amount of unnecessary data is downloaded as part of even a single web request.

A. Usage-based Pricing

Users in developing regions are generally subject to usage-based pricing. In developing regions, the two primary reasons for usage based pricing are limited data capacity and unpredictable demand. Similarly, international roaming users have also traditionally been charged high usage-based fees. These usage-based prices in both developed and developing regions are substantially higher than unlimited plan prices in developed countries. Unfortunately, with spectrum auction prices increasing around the world, cellular providers are looking toward using usage based pricing as a mechanism to increase revenues. Major mobile operators in developed countries have recently shifted to "tiered" pricing plans even for domestic data subscribers rather than unlimited plans. In the United States, AT&T began tiered pricing in June 2010, T-Mobile in June 2011, and Verizon in July 2011 [**?**].

Data Volume	Price (Ksh)	Price per MB (Ksh)
50MB	100	2.00
200MB	250	1.25
500MB	499	1.00
1.5GB	999	0.65
3GB	1,999	0.65
4GB	2,499	0.61
8GB	3,999	0.49
20GB	9,999	0.49
30GB	14,999	0.49

TABLE I
Prepay data bundles for Safaricom, Kenya's largest mobile network. All plans cost 8Ksh per MB after the bundle volume. 83.7Ksh = 1USD (December 2011)

One of the reasons providers use usage-based pricing models for data and voice connectivity is due to the uncertainty in the demand. Typically, regions which use flat-rate pricing will place contractual constraints on users to overcome the demand uncertainty. After crossing the plan limit, the fees of these plans are roughly on the order of 0.1 - 0.25 USD per MB across different providers around the world. At these rates, the relative price is high when compared to the purchasing power parity (PPP) for the region.

Most users in developing regions are subject to different forms of usage based pricing models. The simplest usage based plan is to charge a fixed price per MB of download.

Some other plans which charge a fixed cost typically place a low download size limit or place a limit on the usage time, both of which constrain the user. Beyond these limits, most plans revert back to a simple per MB charging model. Table I lists the data plan bundles from Safaricom, Kenya's largest mobile network. Within these data plans, the cost per MB is relatively low (0.023 to 0.006 USD per MB). However, after the bundle limit is exceeded, each additional MB costs approximately 0.10 USD.[2] We found an average popular web page is of size 551.4 KB. Any web page with embedded objects like sounds, video, or flash objects would be several MB in size. Therefore, the cost of even a single download has the potential to be fairly high.

Service provider	Roaming charges per MB
AT&T	$19.96
Verizon	$20.48
T-mobile	$15
Sprint	$19.5

TABLE II
INTERNATIONAL DATA ROAMING CHARGES FOR MOBILE BROADBAND (DECEMBER 2011).

While the focus in this work is on developing regions, it is important to note that the constrained mobile data problem is not isolated to these regions. Table II illustrates the roaming fees per MB of data transfer for the four large cellular providers in the US. We observe that most carriers charge between $15-$20 per MB of data transfer which is extremely expensive.

B. Anatomy of a Web Page

The second troubling aspect of mobile web browsing is the complexity of web pages and the amount of unnecessary information on most web sites. A relatively simple web page contains a combination of several types of objects including HTML, CSS, favicons (shortcut icon), images, Iframes, scripts, Flash objects and embedded objects. Among these web resources, not all of them are directly relevant to what users want to download. It has been observed that the size of web page has more than tripled from 2003-2007 [6]. Web page size has been growing at an astronomical pace especially with the introduction of video.

Based on a simple analysis of the top 100 global web pages, as identified by "Alexa" [1], we find the distribution of objects across web pages as summarized in Figure 1. The average size of web pages is 551.74 KB with 200.41 KB occupied by Javascript files. Our observations corroborate the trends found in other more comprehensive measurement studies [3], [11]. In our analysis, we did not include video files since their size alone is higher than the rest of the page, and even without including the video objects the cost of browsing a single page is high.

[2]Kenya is one of the fastest growing and competitive mobile markets in Sub-saharan Africa, and represents an approximate lower bound on mobile prices.

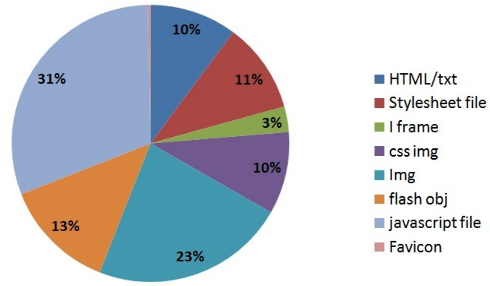


Fig. 1. Web page anatomy of Alexa top 100 pages broken down by bytes.

IV. SYSTEM DESIGN

In this section we describe the design details of our system and how cost-aware mobile web browsing works. In our system, all web requests from a mobile user are directed to a cost-aware HTTP proxy for web access. We explicitly assume that the HTTP proxy resides on a machine with ample network resources that is not bandwidth or cost constrained. The proxy maintains a registry of users and their corresponding cost plan status. The proxy is responsible for handling all the communications between the mobile clients and web servers. Given a new web request from one of the mobile clients, the proxy computes the cost quota of the user and performs different content adaptation techniques to forward the best version possible.

The mobile client device does not require any software modifications. To start using our system, the user manually configures the mobile web browser to connect through the proxy and specifies credentials to use the proxy. In an initial registration phase, a web form is used by the client to communicate cost parameters essential to implementing cost awareness (e.g. the user's pricing plan). All returned pages include an embedded hyperlink to allow the user to request enhanced versions of the page if desired. The adapted page versions are stored at the proxy for a short time to satisfy enhanced version requests.

A. Computing the Cost Quota

Consider a single user who is associated with a target cost budget B per month. Existing cost pricing models can be simplified into one of two types:

Constant rate model: The user pays a standard rate of α per MB of download. Thus, the overall quota of the user $Q = B/\alpha$ MB for a target budget B.

Bundle rate model: The user pays a fixed cost C for a bundle of size Q. Beyond a usage of Q, the user pays a standard overage fees of α per MB.

We simplify both these models using a simple quota Q for the target budget. In the bundle rate model, if the user exceeds the specified quota Q over a month, we request the user to specify a new target budget B beyond the fixed cost and compute a new quota as B/α MB for future requests during the month. Let N represent the average number of requests issued by the user in the previous months and T represent the billing cycle-period of one month ($T = 30days$). Let t

represent the current time that has elapsed during this time period. Let $n(t)$ represent the number of requests made so far and $q(t)$ is the current quota consumed. Given that users may have variable usage characteristics across months, we do not constrain the number of user requests in our model. We assume that the requests from a user arrive as a memoryless process at an unknown rate. Using the short-term history as a sample, we predict the future number of requests during the time-period $T - t$ left in the current monthly cycle as: $n(t) \times (T - t)/t$. For the current request, this assumes that the rate of requests for this time cycle is represented by $n(t)/t$. We can estimate the quota of a new web request as:

$$((Q - q(t)) \times t)/(n(t) \times (T - t))$$

This estimation may not be accurate when $n(t)$ is very small because there is little information to predict the number of requests during the month. In this case, we allocate a web request quota of Q/N based on the previous month history of N. When $q(t) > Q$, we revert to the basic text version of a web page unless the user explicitly requests an improved version.

B. Content Adaptation Ladder

Given the anatomy of a web page, we introduce a six-level content adaptation hierarchy for a given web page. The first three levels of the Web page which form the *text-only ladder* focus on textual versions of the page while the final three levels which form the *advanced ladder* focus on adaptation and filtering techniques that preserve the basic structure of the page.

1) Text-only ladder: The three levels of the text-only ladder of a web page are:

Snippet page: In the snippet version of a page, we extract the most important textual snippet of the web page requested by the user. In the preprocessing step, we extract the main body of the web page by removing redundant data, advertisements and split the block-level HTML components into a sequence of paragraphs. To extract the main text snippet, we extend the basic mechanism proposed in [29], which uses a simple averaging mechanism to smooth the word count graph at a paragraph level. We extend this scheme to include the influence of a desired number of neighboring paragraphs. We normalize the word count of every paragraph by convolving the word count function with a Gaussian function with a fixed width, that is, if $\{n_i\}_{i=1}^{N}$ is the normalized word count as a function of the paragraph number, we do the following:

$$\hat{n}_i = \frac{\sum_{k=-p}^{p} n_{i-k} e^{-\frac{2k^2}{p^2}}}{\sum_{k=-p}^{p} e^{-\frac{2k^2}{p^2}}} \quad (1)$$

where p is the range : number of samples to be considered before and after the current sample. A salient feature of the Gaussian smoothing function is that the current sample is given the highest weight and the weight of the neighboring samples decays smoothly with distance. This feature helps in distinguishing the main text with higher probability.

Text-only version: In the text-only version, we strip all unessential HTML tags and present a condensed HTML (with a plain CSS) containing the textual content of a page. This is done by repeatedly using the above mentioned procedure of main text snippet extraction to extract the next level of main snippets. We then take the top 5 results of the main text snippet extraction and form a text-only version of the requested web page.

Page summarization(Level 0): At this level, we present a summarization of the web page where we show the important headings with little more information about them. This method filters out everything but the headings and the paragraphs next to each heading. This technique works very well for news web pages or other pages with textual data.

2) Advanced Ladder: The advanced ladder supports three levels as outlined below.

Level 1: HTML, CSS, Iframe, Relevant Javascripts, Images in headings

Level 2: Level 1 + Images(compressed, down-sampled), Iframe

Level 3: Level 2 + Embedded objects

Level 1 extracts and presents only the critical components of the page while Level 2 is a filtered and adapted version of the page (removing ads and unnecessary scripts). Level 3 presents an adapted version of the page with no filtering. While generating these different versions of the pages, we performed a few critical optimizations to enhance load times and reduce the web page size.

HTML rewriting: Minification is the practice of reducing the size of the code thereby improving the load times. Minification usually involves removing white spaces (space, tab, newline) and comments. In the case of Javascript, this improves response time performance because the size of the downloaded file is reduced. This contributed to over a $5 - 10$ KB reduction in web page sizes.

Rendering time optimization: Moving the Cascaded Stylesheet to the header of the HTML allows progressive rendering. Similarly, moving the Javascripts to the end of the HTML code enhances user experience since it would enable the light weight components of the web page to be downloaded first.

Ads and favicon filtering: Javascripts that contain particular keywords are potential signatures of advertisements. To separate scripts that are related to a page and those that are not, we constructed a large list of keywords that we used for identifying scripts related to advertisements. Similarly, we distinguish images related to the web page from those that are not based on the name, location of the images and location of the image pointer in the HTML. We also remove favicons across all adaptation levels which saves $2 - 10$ KB.

Image and embedded objects filtering: We use the standard approaches of down-sampling images and removing embedded objects at the appropriate levels of the adaptation ladder which may include multimedia content, like sounds, videos, flash etc., in web pages. The typical size of a flash object in web pages is 85.4 KB which contributes 13% to the average web page size.

C. User Feedback

Every web request response is associated with attached "cached" hyperlinks at the top of the page which allow the user to request enhanced versions of a page present in the proxy cache. This is important for when the user crosses the target quota and is automatically assigned a text-version of page with a quota of 10 KB (our default value). The user is presented four pointers: (a) Original page; (b) Level 2 page; (c) Level 1 page; (d) Text-only version.

One challenge of implementing the multi-level adaptation approach is that the proxy requires to first download the entire web page before performing the adaptation. This is especially a concern for proxy-side rendering systems like Opera Mini. To maximize parallelism and minimize rendering time, we perform on-the-fly adaptation at the proxy which issues requests in *batches*. Given a quota, if the adapted version of a web page containing all objects downloaded so far achieves the quota, the proxy stops downloading more objects in the page. The batches of object downloads directly correspond with the three levels in our advanced ladder. In the first batch, we download objects corresponding to Level 1, Level 2 in second batch and Level 3 in third batch.

V. Implementation

The prototype of our system is implemented in 2000 lines of Java code. The architecture of the system consists of a proxy and local storage to persist user information. As mentioned in Section IV, our system does not require software modification of the user's browser nor running any additional processes on the mobile client device.

A. Proxy

To service page requests,the proxy has three main responsibilities. First, the proxy listens for incoming connections on port 7500 and spawns a new thread to process the request. In addition to parsing the HTTP request header, the thread reads the user's cost parameters from the disk and stores the state of the user in memory. Second, the thread receives web pages and dynamically computes the cost quota. The thread uses the structure of the web page to determine the types and the number of web resources in the page. Using the cost quota, the content of the web page is adapted and filtered using the techniques described in Section IV. Finally, the thread sends the web page to the user followed by updating the user's cost information.

B. Local Storage

The second module of the system consists of persisting the state of registered users. The storage used is a light weight database that consists of XML files that store the registered users and the their corresponding download budget states. The parameters that the proxy saves are mainly the user ID, the user's data plan, and the cost quota values. For performance, the threads servicing the requests update a user's record after forwarding the web pages to the client.

VI. Evaluation

In this section, we evaluate the behavior and the performance of our system using the most popular web pages in the Internet listed by Alexa as the sample sites for our analysis. Our evaluation shows three key results. First, we show that the content adaptation ladder component of our system can automatically learn the structure of web pages and extract the most meaningful components of a page within the provided quota. Second, we show that our system can significantly reduce the size of web pages for different levels of adaptation and also provide a substantial cost savings. Associated to this, our system can also provide the best possible version for a specified byte quota for a page. Finally, we perform a small-scale user study and obtain feedback from users on the usability of the system; most users expressed positive impressions in the system usability but some users reported minor presentation problems for a few web pages which were excessively filtered when users had very small quotas.

Adaptation ladder: To study the behavior of the system in the for the different adaptations levels (0,1,2), we manually tested the system across a wide range of sites. In our tests, we find that our system is able to successfully extract the critical components of a page even when the quota was small. We highlight this with three examples.

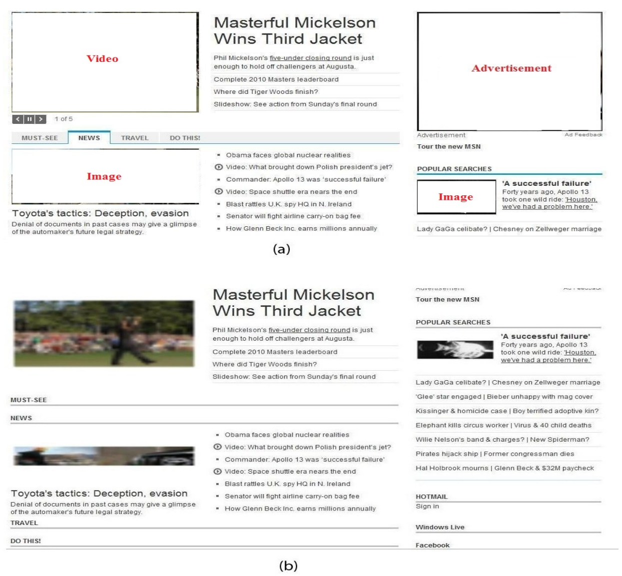
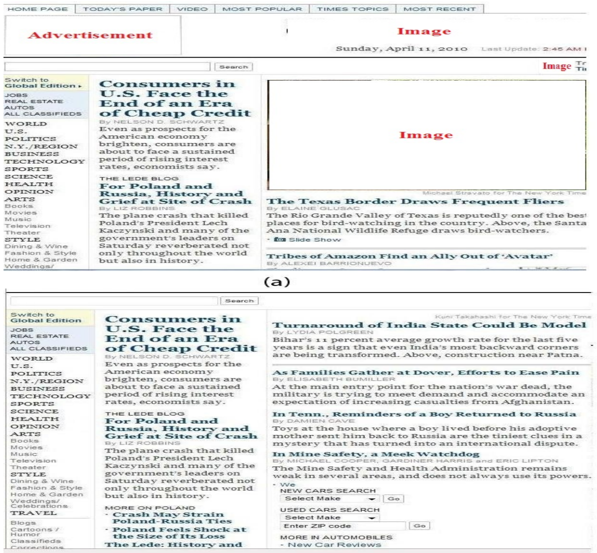


Fig. 2. (a) Original page. (b) Level 2 adaptation. Visually, the images are downsampled, and the page retains most of its original formatting.

Figure 2 shows how a web page looks after being adapted from the original into Level 2 format. The advertisements and images in the original screenshots have been edited out due to copyright and replaced with labels. We observe that the Level 2 web page is very similar to the original both in terms of format and content, but the page size is approximately half the size.

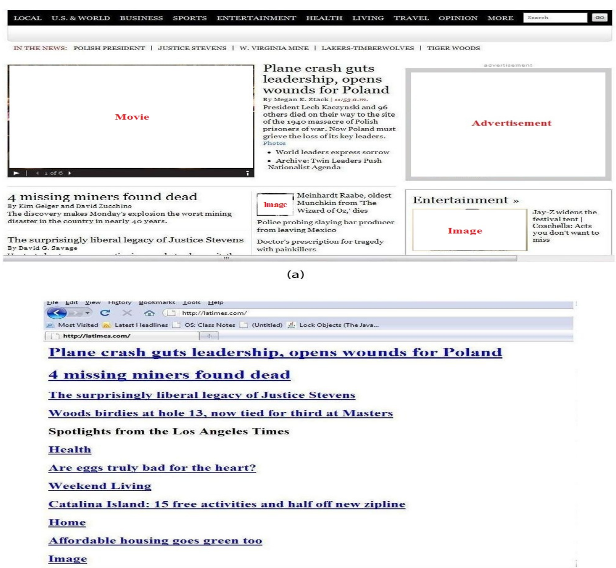
Figure 3 illustrates how a web page looks visuall after being adapted into Level 1 format. This adapted page still conveys the same information content, but is over three times smaller than the original.



(a)

(b)

Fig. 3. (a) Original page. (b) Level 1 adaptation. Visually, the images completely removed, but the page retains some of its original structure.



(a)

(b)

Fig. 4. (a) Original page. (b) Level 0 adaptation. Visually, the page is completely stripped of formatting, structure, and multimedia.

Figure 4 shows the Level 0 adaptation of a webpage. Virtually all formatting information is removed. Only the bare text and hyperlinks are still present.

In these examples, we can clearly observe that our system is able to extract the relevant important information while filtering out irrelevant information at the different adaptation levels. The adaptations performed by our system rendered the page in a presentable and readable style across all web sites we tested. Our system works especially well for news web pages as text summaries and related images can be easily extracted. For sites such as flickr [2] whose content is primarily images, the per-byte reduction factors after adaptation are not as significant. We hope to optimize our algorithms for multimedia centric websites in future work.

TABLE III
CUMULATIVE DISTRIBUTION OF THE SIZE REDUCTION FACTOR FROM THE ORIGINAL PAGE.

Type	50%	75%	90%	95%
Snippet	110	200	300	350
Level 1	3	7	15	20
Level 2	2	5	10	16

Size reduction: To measure the size reduction, we ran our system across the top sites from Alexa for different levels of adaptation. Table III summarizes our results. The table shows the cumulative distribution of the reduction factor page size when using our system for different adaptation levels at 50%, 75%, 90% and 95% of the CDF. We observe that in the median case, Level 1 and Level 2 adaptations provide a size reduction by a factor of 3 and 2 respectively while snippet adaptation can provide a reduction by a factor of 110. The reduction factor of page size increases up to 20, 60, and 350 times for adaptation levels (1, 2, and snippet) at 95% of the CDF.

Cost savings: To measure the cost savings of our system, we consider the scenario where an average user makes N web requests per day. For simplicity, we consider the user to pick one of the Alexa sites for the analysis. We run the system for 2 settings: (a) With a quota of 10 MB for the day; (b) With a quota of 5 MB for the day. For a user who visits $N = 100$ web pages per day, the conventional web browsing system yields roughly 42.4 MB of download data. In contrast, our system was able to significantly reduce the amount of data downloaded. For large values of N (¿100), our system slightly over-shot the specified quota since the per page quota was too small to accommodate even the lowest level of adaptation. Figure 5 shows the distribution of web page requests across various levels of adaptation for the two different quota constraints. For example, in Figure 5(a) 54% of the pages forwarded to the client with a 10 MB budget are Level 3 adapted versions. When the user has only 5 MB budget left, 53% of the pages returned are Level 1 adapted versions.

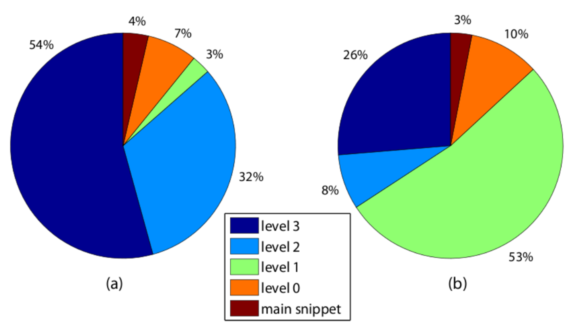


Fig. 5. Distribution of 100 web page requests. (a) Quota left: 10MB (b) Quota left: 5MB

Small-scale user study: We evaluated the merits of our proxy with a small-scale user study comprising of seven participants with no prior exposure to our system. Participants were graduate students in the US who were presented with a hypothetical situation where they would soon run out of monthly data and our proxy is adapting webpages for this scenario. Participants were shown the roaming charges of the major service providers and the typical costs they would incur for different levels of mobile usage under roaming. Each participant was allotted about 15 to 30 minutes browsing on an iPhone rather than a cheaper device to minimize contamination of our results by dissatisfaction due to device limitations. Quota numbers were not used in this study, and we manually set the levels of adaptation so that participants were able to try all levels of adaptation.

At the end of the study, each participant was then given a semi-structured interview where they were asked (a) their overall satisfaction with the experience given the scenario (on a scale from 1 to 5); (b) whether they would use such a system; (c) other feedback and comments. Table IV summarizes our findings. The overall response was positive. Four of the participants in the user study readily expressed interest in using a system like this. Among the participants who responded with only moderate satisfaction, we found that dissatisfaction was primarily due to the failure of our participants to realize that when left with very small quotas, web pages are relatively big. As a result of this, one participant was surprised by the snippet versions of pages and the other participant found the system to be too aggressive. Users did not observe any noticeable delay in browsing due to out intermediating proxy.

User	Overall satisfaction (0-5)	Use the proxy
1	3.5	Maybe
2	4	Yes
3	3.5	Yes
4	3	No
5	4	Yes
6	4	Yes
7	3	Maybe

TABLE IV
USER STUDY RESULTS

VII. CONCLUSIONS

Most mobile users in developing regions are subject to usage based pricing data rate plans in some form or another resulting in high usage costs. In this paper, we presented a cost-aware mobile browsing framework that automatically adapts web content as a function of a user's data pricing plan and usage levels to substantially reduce data costs. In addition to reduced costs, our approach enhances user browsing experience by forwarding the best possible version of a page for a given data allowance and based on user feedback. Our evaluation illustrates the significant cost savings that our system can provide for mobile web users.

REFERENCES

[1] Alexa the web information company. http://www.alexa.com/.
[2] Flickr. http://www.flickr.com/.
[3] Let's make the web faster - google code. http://code.google.com/speed/articles/web-metrics.html.
[4] Opera mini. http://www.opera.com/mobile/download/.
[5] State of the mobile web (2011). http://www.opera.com/smw/.
[6] Websiteoptimization.com. http://www.websiteoptimization.com/.
[7] P. Baudisch, X. Xie, C. Wang, and W. Ma. Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 91–94. ACM, 2004.
[8] T. Bickmore, A. Girgensohn, and J. Sullivan. Web page filtering and re-authoring for mobile users. *The Computer Journal*, 42(6):534–546, 1999.
[9] A. Blekas, J. Garofalakis, and V. Stefanis. Use of rss feeds for content adaptation in mobile web browsing. In *Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility?*, pages 79–85. ACM, 2006.
[10] O. Buyukkokten, O. Kaljuvee, H. Garcia-Molina, A. Paepcke, and T. Winograd. Efficient web browsing on handheld devices using page and form summarization. *ACM Transactions on Information Systems*, 20(1):82–115, 2002.
[11] J. Charzinski. Traffic Properties, Client Side Cachability and CDN Usage of Popular Web Sites. *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, 2010.
[12] J. Chen, L. Subramanian, and J. Li. Ruralcafe: web search in the rural developing world. *Proceedings of the 18th International Conference on World Wide Web*, 2009.
[13] X. Chen and X. Zhang. Coordinated data prefetching by utilizing reference information at both proxy and web servers. *ACM SIGMETRICS Performance Evaluation Review*, 2001.
[14] Y. Chen, W. Ma, and H. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Proceedings of the 12th international conference on World Wide Web, May*, pages 20–24, 2003.
[15] J. Domčnech, A. Pont, J. Sahuquillo, and J. Gil. A user-focused evaluation of web prefetching algorithms. *Computer Communications Review*, 30(10), 2007.
[16] L. Fan, P. Cao, W. Lin, and Q. Jacobson. Web prefetching between low-bandwidth clients and proxies: potential and performance. *Proceedings of the ACM SIGMETRICS International Conference on Measurement and modeling of computer systems*, 1999.
[17] A. Fox and E. Brewer. Reducing www latency and bandwidth requirements by real-time distillation. *Computer Networks and ISDN Systems*, 28(7-11):1445–1456, 1996.
[18] Y. Hwang, J. Kim, and E. Seo. Structure-aware web transcoding for mobile devices. *Internet Computing, IEEE*, 7(5):14–21, 2003.
[19] J. Korva, J. Plomp, P. Määttä, and M. Metso. On-line service adaptation for mobile and fixed terminal devices. In *Mobile Data Management*, pages 254–259. Springer, 2001.
[20] T. Laakko and T. Hiltunen. Adapting web content to mobile user agents. *Internet Computing, IEEE*, 9(2):46–53, 2005.
[21] E. Lee, J. Kang, J. Choi, and J. Yang. Topic-specific web content adaptation to mobile devices. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 845–848. IEEE Computer Society, 2006.
[22] T. F. MediaLab. Web content adaptation - white paper, 2004.
[23] P. K. Mishra, K. Baliyan, Deepshika, and S. Singh. User interactive web content adaptation for mobile devices. In *2010 Impact of Globalization and Privatization on meeting India's IT HR needs*, March 6-7, 2010.
[24] I. Mohomed, J. Cai, S. Chavoshi, and E. de Lara. Context-aware interactive content adaptation. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 42–55. ACM, 2006.
[25] I. Mohomed, J. Cai, and E. De Lara. Urica: Usage-aware interactive content adaptation for mobile devices. *ACM SIGOPS Operating Systems Review*, 40(4):345–358, 2006.
[26] I. Mohomed, A. Scannell, N. Bila, J. Zhang, and E. De Lara. Correlation-based content adaptation for mobile web browsing. In *Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware*, pages 101–120. Springer-Verlag New York, Inc., 2007.
[27] W. Thies, J. Prevost, T. Mahtab, G. Cuevas, S. Shakhshir, A. Artola, B. Vo, Y. Litvak, S. Chan, S. Henderson, et al. Searching the world Wide Web in low-connectivity communities. *Proceedings of the 11th International Conference on World Wide Web*, 2002.
[28] D. Zhang. Web content adaptation for mobile handheld devices. *Communications of the ACM*, 50(2):75–79, 2007.
[29] B. Zhou, Y. Xiong, and W. Liu. Efficient web page main text extraction towards online news analysis. In *e-Business Engineering, 2009. ICEBE'09. IEEE International Conference on*, pages 37–41. IEEE, 2009.

Sindhura Chava is currently working as a Technology Analyst at a financial firm in New York. Her interests lie in operating systems, firmware engineering and computer architecture. She received a B.E. degree in Electronics and Instrumentation from BITS-Pilani, India and a M.S. degree in Computer Science from New York University. Contact her at 715 Broadway, Room 706, New York, NY 10003; sindhura.chava@gmail.com.

Rachid Ennaji currently works as a Software Engineer at Amazon, Inc. He is interested in different aspects of distributed computing and systems. He received a M.S. degree from New York University in 2011, and a B.S. degree in Computer Science from Al Akhawayn University in Ifrane in 2009. Contact him at 715 Broadway, Room 706, New York, NY 10003; rachid.ennaji@nyu.edu.

Jay Chen is a Visiting Assistant Professor at NYU - Abu Dhabi. He received a PhD in Computer Science from New York University (NYU). His research is in the area of Information and Communication Technology for Development (ICTD). He is a researcher at the Center for Technology and Economic Development (CTED) in Abu Dhabi. He has a background in Systems and Networking, and his current focus is on providing affordable information access for clients with low/poor connectivity and sustainable energy. Contact him at Department of Computer Science, NYUAD, Sama Tower 1307, Abu Dhabi, U.A.E.; jchen@cs.nyu.edu.

Lakshminarayanan Subramanian is an Assistant Professor in the Courant Institute of Mathematical Sciences at NYU. His research interests are in the areas of networks, distributed systems and computing for development. He co-leads the Networks and Wide-Area Systems(NeWS) group (which investigates software solutions for distributed systems, wireline and wireless networking, operating system, security and privacy, technologies and applications for the developing world) and the CATER Lab at NYU (which focuses on developing and deploying low-cost, innovative technology solutions to some of the problems in developing regions in terms of communication, healthcare and microfinance). Contact him at 715 Broadway, Room 706, New York, NY 10003; lakshmi@cs.nyu.edu.