RuralCafe: Web Search in the Rural Developing World

Jay Chen
New York University
715 Broadway
New York, United States
jchen@cs.nyu.edu

Lakshminarayanan Subramanian
New York University
715 Broadway
New York, United States
lakshmi@cs.nyu.edu

Jinyang Li
New York University
715 Broadway
New York, United States
jinyang@cs.nyu.edu

ABSTRACT

The majority of people in rural developing regions do not have access to the World Wide Web. Traditional network connectivity technologies have proven to be prohibitively expensive in these areas. The emergence of new long-range wireless technologies provide hope for connecting these rural regions to the Internet. However, the network connectivity provided by these new solutions are by nature *intermittent* due to high network usage rates, frequent power-cuts and the use of delay tolerant links.

Typical applications, especially interactive applications like web search, do not tolerate intermittent connectivity. In this paper, we present the design and implementation of *RuralCafe*, a system intended to support efficient web search over intermittent networks. RuralCafe enables users to perform web search asynchronously and find what they are looking for in *one round of intermittency* as opposed to multiple rounds of search/downloads. RuralCafe does this by providing an expanded search query interface which allows a user to specify additional query terms to maximize the utility of the results returned by a search query. Given knowledge of the limited available network resources, RuralCafe performs optimizations to prefetch pages to best satisfy a search query based on a user's search preferences. In addition, RuralCafe does not require modifications to the web browser, and can provide single round search results tailored to various types of networks and economic constraints. We have implemented and evaluated the effectiveness of RuralCafe using queries from logs made to a large search engine, queries made by users in an intermittent setting, and live queries from a small testbed deployment. We have also deployed a prototype of RuralCafe in Kerala, India.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; H.3.5 [**Information Systems**]: Web-based services

General Terms

Design, Experimentation, Performance

Keywords

World Wide Web, web search, intermittent network, low bandwidth

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

1. INTRODUCTION

The proliferation of the Web and the Internet has largely remained an urban phenomenon. A significant fraction of rural regions around the world, especially in the developing world, continue to have extremely limited access to the Internet primarily due to economic constraints [15, 6]. The underlying factors for the lack of connectivity are two-fold. First, the purchasing power in these regions is significantly lower than urban areas. Second, none of the traditional wire-line connectivity solutions (fiber, broadband and dial-up) are economically viable for rural regions with low user densities [15, 25]. Even if connectivity is available in the form of satellite networks, the usage rates are exorbitant ($3K per month for 256 Kbps) making it unaffordable.

The recent emergence of new low-cost connectivity solutions using long-range wireless technologies (WiMax [30], long-distance WiFi [19], cellular) and delay-tolerant mechanical backhaul networks provide hope for rural connectivity. In fact, cellular and WiMax networks are increasingly being deployed in rural Asia and Africa. Mechanical backhaul networks [28, 20] which use physical transportation systems have also been deployed in many rural regions. In our prior work [19], we have deployed low-cost WiFi-based rural wireless networks in several countries in Asia and Africa.

A common theme across all these new solutions is that network connectivity is *intermittent* because connectivity is not available or too unaffordable to be used on a continuous basis. For example, Africa does have reasonable cellular coverage in rural areas, the current calling rates ($10 - 50$ cents/min) and data services (1 cent/KB) make it unaffordable to use these networks on a regular basis. Also, in a recent user survey in rural Ghana we found that more than half of rural users can afford less than $1 for calling charges per week [14]. An alternative means of connectivity based on delay tolerant mechanical backhaul networks, are by nature intermittent. Mechanical backhauls have extremely high one-way link latencies lasting a few hours because data is physically moved around. Another cause of intermittency in rural areas is frequent power outages and network failures; most rural regions in Africa experience long power cuts everyday [26].

The overarching problem we seek to address is *how do we extend the Web to the rural developing world?* To address this question, we need to first enable web based applications and services to work on top of intermittent networks. The intermittency in these networks is significantly longer than in traditional networks. Most applications do not work in these intermittent environments, and certain applications may require a complete redesign; the traditional sockets API is not appropriate for intermittent networks and applications require a new communications API [5].

The specific problem we seek to address in this paper is: *how does one support efficient web search over intermittent networks?* Traditionally, web search is an interactive process which requires several rounds of interaction between the user and a search engine, before the user finds the appropriate search response. However, in intermittent environments, multiple rounds of interactive search would be impractical. The problem of intermittent web search in rural contexts has received very little attention within the research community; the TEK [11, 27] and DAKnet [20] projects, are the only efforts we are aware of in this space.

In this paper, we present the design and implementation of RuralCafe, a generic platform that enables intermittent web search in a single round of interaction over a variety of intermittent networks. While it may not always be possible to achieve perfect search in one round, the goal of RuralCafe is to maximize the utility of a search result returned to the user after each round of interaction (similar in spirit to Yahoo's Onesearch [31]).

The primary contribution of RuralCafe is to provide a holistic approach to rethinking web search design for an intermittent setting. Specifically, by modifying the search interface and the search process and exposing the intermittency to the user, RuralCafe hopes to achieve better user-driven targeted one-round search in different intermittent environments. RuralCafe is not intended to be a new search engine or a new search algorithm, but a system that enables users to better interact with a search engine over an intermittent network.

RuralCafe employs three important design choices. First, the current query interface supported by search engines is not expressive enough to support one-round search. RuralCafe exports an alternative search query interface which enable users to enter all the information they know regarding a specific query including customized user-specific search response options. Second, unlike many intermittent systems [24, 2, 16, 20, 17, 4] that hide the intermittency from the application, RuralCafe explicitly exposes the intermittency to the application and to the end-user. Third, RuralCafe allows users to specify their preferences for condensing search responses in bandwidth constrained settings.

In Jan 2009, we deployed RuralCafe in Amrita University in Kerala, India. Our deployment, currently at an initial phase, is intended to serve nearly 10000 students who share a 750 Kbps low-bandwidth Internet connection across 400 terminals. At peak time, the available bandwidth per end-host is less than 2 Kbps. However, we have not yet conducted any detailed user studies based on our deployment. We note that Amrita University is among the few universities in India which has reasonably good Internet connectivity; many other universities are restricted by aggregate bandwidths of 256 Kbps or less.

In an initial test deployment within our lab environment and for a small user-base at Amrita University, we examine the general usability of RuralCafe's various search features. Using two different query logs (one from AOL [18] and one collected in a rural Internet cafe in Cambodia [6]), we show that RuralCafe is able to adapt the response for a query as a function of the type of intermittent network and the amount of network resources available per query. We find that RuralCafe is appropriate and beneficial in environments where either the underlying network intermittency is visible to the user or the available bandwidth per user is severely constrained.

2. MOTIVATION

To motivate the need for RuralCafe, we will first describe different forms of search in rural regions in developing countries and how traditional solutions are unfit for these environments. Next, we describe the results of a recent needs-assessment study we performed in a large university in India where nearly 400 students simultaneously share a low-bandwidth Internet link. The primary result from the study is the traditional process of web search is unsuitable when for very low-bandwidth and intermittent environments.

2.1 Examples of Rural Web Search

Consider these three commonly occurring search scenarios across different types of intermittent networks in rural developing regions:

Shared Budget-constrained Low Bandwidth Networks: In this scenario several users ($100-1000$ users) in an institution connect to the Internet using a low-bandwidth connection (say 128 Kbps) with a usage based charging model. This is a very common case with most university and small business settings in several developing countries around the world. For example, in India, a typical rural Business Process Outsourcing (BPO) unit [23] consists of roughly 50-100 people sharing a 64 Kbps Internet connection. In addition, the usage costs of these links are fairly high and the cost is either determined by the total usage time or number of bits transferred. These networks have limited financial budgets for operation which strictly limits the total uptime. Even when connectivity is available, the available bandwidth per user is extremely small (< 1 Kbps per user); from the perspective of the user, these networks appear inherently intermittent though the period of intermittency is small (on the order of a few minutes). A typical search response from a search engine is in the order of 10 KB and most web pages are in the order of 100KB (including images, scripts, advertisements or other content). To partially alleviate the low-bandwidth problem, a common approach is to use admission control policies to provide a more interactive experience to a few users; even then, the network appears intermittent from the user's perspective.

Search using Messaging links: Given the penetration of cellphones in rural regions, one can envision a scenario where a third-party (or the cellphone provider) provides cellphone users with the ability to perform web search queries using the short-messaging service (SMS) or the multimedia messaging service (MMS). A primary motivation for providing a messaging based search service is that in many rural regions, messaging is significantly cheaper than voice calls or data service. In addition, a user can limit costs of the service based on their search needs. An SMS message has a capacity of 140 bytes and an MMS service provides several KBs worth of data in a single message.

Kiosks using Mechanical Backhauls: United Villages [28, 20], a not-for-profit organization has deployed WiFi-enabled kiosks in various villages in Asia, Africa and Latin America. Each kiosk uses a mechanical backhaul link based on physical transportation. Vehicles such as buses are used to transport bits between the village and the closest town or city. These links have long delays and are operational only for a few times every day (often once) but can transfer gigabytes of data in one trip. Each kiosk also has an operator who acts as an interpreter to help users surf the web.

Several aspects of these scenarios are noteworthy. First, in all these cases, the user is aware of the intermittent network when performing search and is willing to wait correspondingly for a response. Second, the nature of the intermittent links in each of these cases is very varied; while mechanical backhauls are high latency links with the potential for bulk data transfer, the other two cases exhibit low latencies and low bandwidths. In the budget constrained case, network connectivity is available, but the usage is very restricted.

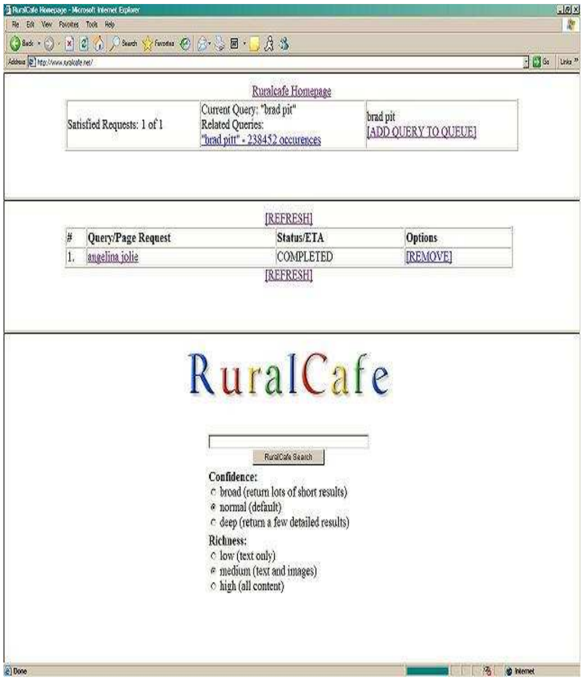
2.2 Browsing Under Poor Connectivity

To better understand the rural setting, we present our primary findings from a needs-assessment study investigating how users' web search and browsing behavior differs when the connection is slow, and whether users employ techniques to alleviate the problem [3]. This study involved 15 subjects on a university campus in Kerala, India, who routinely suffer low-bandwidth or intermittent connections. When compared to typical rural settings, this is actually a "really good" connectivity scenario since the available connectivity is much higher than normal. The network available on the campus was a 750 Kbps connection shared across 3000 faculty and students operating from 400 machines, where during peak hours nearly every machine was being used. Therefore, the worst-case average bandwidth available per machine was approximately 1.9Kbps. This speed is abysmally slow even compared to dial-up (56.6Kbps).

In our study, we installed a Squid proxy to log user queries and responses. In addition, participants were observed and videotaped during the process. Also, screen-capture video of their mouse movements and keyboard entries were recorded. We also interviewed participants using a questionnaire about general reactions to the experience.

We found that caching and compression were helpful in this scenario saving around 50% of the loading time, but were unable to completely insulate the user from the slow connection. Users became frustrated and gave up when pages took several minutes to load. Some users would seek to alleviate waiting times by multi-tasking and loading multiple pages at the same time, effectively queuing requests so they could browse while loading other pages in parallel. These users were the most effective in minimizing the proportion of time they spent waiting for pages to load. The heavy multitaskers only spent 5% to 19% of their time idle while the other users spent 23% to 84% of their time idle. We observed that giving the users an explicit numerical estimate on the amount of time it would take for a page to load rather than the progress bar in the browser could save them considerable amount of time. If users know more precisely how long a page would take to load, they would then be able to do other tasks instead of waiting indefinitely.

We noticed that users often had problems entering queries perfectly the first time. Responses for results came fairly quickly due to the mostly text nature of the query results pages. However, in the mechanical backhaul scenario making a malformed or vague query would cause a huge amount of delay before the mistake was corrected. We also found that users were often looking for a specific piece of information, and were not interested in the richer media such as images or movies that would be embedded in each page. In contrast, when images were desired, users had no problems selecting the image search explicitly.


RuralCafe Homepage - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.ruralcafe.net/
Go Links
RuralCafe
Done
Internet

Ruralcafe Homepage

Satisfied Requests: 1 of 1	Current Query: "brad pit" Related Queries: "brad pitt" - 238452 occurences	brad pit [ADD QUERY TO QUEUE]

[REFRESH]

#	Query/Page Request	Status/ETA	Options
1.	angelina jolie	COMPLETED	[REMOVE]

[REFRESH]

RuralCafe Search

Confidence:
○ broad (return lots of short results)
● normal (default)
○ deep (return a few detailed results)

Richness:
○ low (text only)
● medium (text and images)
○ high (all content)

Figure 1: RuralCafe Query Interface

3. RETHINKING THE SEARCH PROCESS

An important aspect of the RuralCafe design is rethinking the search process in the context of intermittent networks. The traditional model of interactive web search is not feasible in intermittent contexts since the user does not have the luxury of continuously refining the query to find what he wants. Therefore, the philosophy of RuralCafe is to maximize the utility of search response for every round of search. To achieve this goal, the RuralCafe design uses four key design principles:

1. *Rethink the Search Interface:* The traditional search interface is not expressive enough for intermittent search. RuralCafe uses a modified search interface that explicitly exposes the underlying network intermittency to the users and enables users to express their search intent and requirements in a greater level of detail. Part of the challenge is also to make the search interface relatively simple, since users typically do not prefer complicated search interfaces. In addition, any redesign of the search interface should not require any modifications to the browser software at the end hosts.

2. *Local Query Refinement and Search:* A significant fraction of user queries are often ill-specified and ambiguous. To maximize the utility of a search response every query should be meaningful and as well specified as possible. One way we help the user achieve this is by asking him to explicitly specify whether the search query is directed or not (deep or broad). Another way assist the user is by performing local query refinement and local search. RuralCafe uses a large local cache which is pre-populated and enables users

to exhaustively search the local cache before issuing search query over the network. Local search enables users to appropriately refine their query to reduce the number of search rounds that require the network. RuralCafe uses a repository of "popular search phrases" to support query expansion and also correct potential errors in queries.

3. Adapting to different Intermittent Environments: RuralCafe uses an intermittent link model to encapsulate different types of intermittent networks under a common set of parameters. This enables RuralCafe to easily adapt across different intermittent environments and also quantify the available network resources for each query.

4. Tailoring the Search Response: RuralCafe tailors the search response for a query as a function of the user search preferences and available network resources. Depending on the type of query, RuralCafe prefetches an appropriate set of related pages to a query that enables users to locally search. In addition, RuralCafe employs different filtering and compression routines to prefetch several pages within the available quota for a query.

In the rest of this section, we primarily focus our discussion on the RuralCafe search interface and describe the rationale for the modified interface. We discuss the design in detail in Section 4 .

3.1 RuralCafe Search Interface

Figure 1 illustrates the expanded search interface of RuralCafe. From the user's standpoint, the interface is explicitly tailored for intermittent search. Every user is associated with a *queue* of pending search requests and a *history* of previously processed search requests. Each processed search query is associated with a pointer to the set of search responses for that query and the set of "pre-fetched objects" corresponding to the search query. RuralCafe explicitly reports the current status of every search request which denotes the expected waiting time for each query (as illustrated in Figure 1).

RuralCafe uses a two-step process before a user issues a search query. In the first step, the user is expected to refine the search query based on local search. As part of this process, RuralCafe provides related documents from the local cache. In addition, RuralCafe provides pointers to "related query terms" corresponding to a query based on a collection of pre-populated list of popular search terms. This can also be used for correcting grammatical errors as illustrated in an example search query for "brad pit" in Figure 1. In this case, RuralCafe proposes spelling corrections to popular terms that are incorrectly spelled. After refining a query, the user explicitly adds the query to the queue of outstanding requests. A user also has the flexibility to remove or retry previously issued queries with different query options.

Every query in RuralCafe is specified by three parameters:

- Expanded list of query terms
- Type of query response (broad, normal or deep)
- Richness of search result pages (low, medium or high)

The *expanded list of query terms* is derived by the user after local query refinement and localized search. Specifically, RuralCafe uses a pre-populated list of popular shingles based on the Linguistic Data Consortium (LDC) dataset [12] to aid users to locally expand the list of query terms and also correct spelling errors in queries. While advanced search options may be useful as exported by existing search engines, these options are rarely used in practice. The key design goal of the RuralCafe search interface is to keep it simple yet expressive.

RuralCafe requires two additional parameters from the user on the *type of response* and *richness* to drive the search process. The *type of query response* option allows a user to both characterize the type of search query and the overall type of search response. We define "deep" queries to be well-specified while "broad" queries as not well-specified; this option determines how RuralCafe selectively prefetches search response pages. The *richness of query response* characterizes the type of content that the user anticipates in response to a query (text, images, documents, videos etc.) thereby determining how to selectively filter content in the search response.

Search Interface Rationale: To understand the rationale behind the modified search interface, it is essential to understand how do users perform web search in terms of search styles and goals. Table 1 shows the significant goal divisions of a thorough study of search goals by Rose et al. [22] from Yahoo. For this categorization Rose et al. opted to ask the user to manually specify the goal category along with each search for the purposes of their study. The results of common navigational search queries imply that a single page is requested repeatedly and could easily be cached. The classification of informational directed, informational undirected, and resource tasks can be re-interpreted as tasks where the user knows specifically what they are looking for ("deep") and ones where the user does not ("broad"). Based on this characterization, we derived the *type of response* query option in RuralCafe.

This begs the question: are users capable of providing meaningful and expressive queries? The Time Equals Knowledge (TEK) [11, 27] project, an email based asynchronous search system deployed in Solomon Islands shows that users are indeed capable of providing expressive and meaningful queries when operating in intermittent environments. The local search and query refinement aspect of RuralCafe is primarily to aid the user in identifying related search terms corresponding to a query. In RuralCafe, we explicitly avoid the problem of automated query expansion. While there has been significant amount of work on automated query expansion and query replacement, we anticipate these enhancements to be integrated into the search engine. Therefore, we restrict RuralCafe to user-driven query expansion by suggesting related popular terms for each query.

Finally, the search response should be tailored to the type of content the user anticipates. However, network resources are often constrained in intermittent environments thereby restricting the type of information that can be fetched. To strike a balance between the two, RuralCafe allows the user to specify the richness setting on content quality (”low”, ”medium”, and ”high”). Based on this setting and the available network resources, RuralCafe adapts the type of filtering and compression routines that are invoked for the search response.

4. RURALCAFE DESIGN

In this section, we describe the overall design of RuralCafe. We begin with the RuralCafe setup and query process. We then elaborate on three key design aspects of Ru-

Table 1: Types of Search Queries

Search Goal	Description	Examples
Navigational	Return a website I have in mind	aloha airlines
Directed (deep)	I want to know the answer to some question	what is a supercharger
Undirected (broad)	I want to learn anything and everything about my topic	color blindness

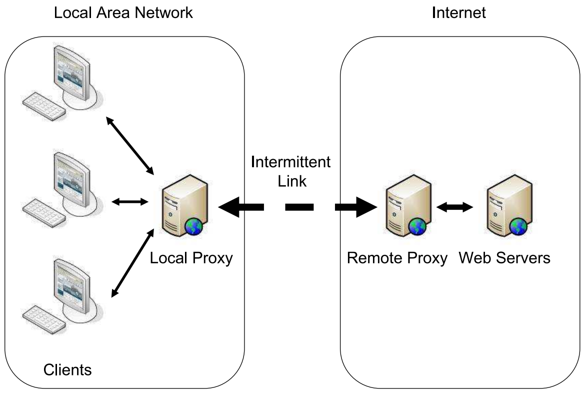


Figure 2: Basic RuralCafe setup

ralCafe: (a) local search and query refinement; (b) tailoring the search response; (c) adapting to different intermittent environments.

4.1 RuralCafe Setup and Query Process

The basic setup of RuralCafe is illustrated in Figure 2. RuralCafe uses a simple proxy architecture comprising of end-hosts within a village that connect to the Internet over an intermittent link using intelligent proxies. One intelligent proxy is placed at either end of the intermittent link. All the traffic to and from the end-hosts traverse the local proxy before being injected on the intermittent link and tunneled to the remote proxy. The remote proxy connects to the Internet using a direct network connection. In practice, the intermittent link could be a multi-hop delay tolerant network with the intelligent proxies placed at the edge of the DTN. If a single end-host directly connects through an intermittent link, as in the case of cellphones, then the local proxy functionality is placed at the end-host. End-hosts are configured to connect to the local proxy by default. The local proxy redirects the user to the expanded search interface when a search engine is requested. When possible, the local proxy is equipped with a large local store which the client can locally search. The entire RuralCafe functionality is deployed at the two proxies.

The user's home page is set to *www.ruralcafe.net*, and the user requests that page from the local proxy. The local proxy serves the search summary page containing the RuralCafe query interface, status of outstanding search requests and pointers to the history of prior search results. The history of prior search results includes the search responses and the set of prefetched pages corresponding to the query. Corresponding to each outstanding query, RuralCafe provides the expected wait time for obtaining a response (based on the intermittent network characteristics). As described earlier, a user issuing a new query first locally searches and refines the query before adding the query to the search queue. The status of outstanding requests is auto-refreshed by the local proxy. The current RuralCafe implementation maintains state about the search status of each user; the user can check the past query results (along with pre-fetched pages) over several days in the system. In the absence of space in the local proxy cache we use a simple history based approach to evict cache entries.

4.2 Local Query Refinement and Search

Query Refinement: We populate the local proxy with a subset of the popular N-grams from the Linguistic Data Consortium (LDC) [12]. This dataset provides a large corpus of *shingles* (a collection of consecutive words) otherwise referred to as N-grams along with their web frequencies as published by popular search engines. While this dataset may not be up to date, the relative popularities of most N-grams are maintained over time. We use a popular LDC shingle dataset to perform two optimizations. The first optimization is to suggest associated popular query terms to the user corresponding to a search query. RuralCafe, then allows the users to choose appropriate query expansion terms from a list of popular terms. To identify the appropriate popular terms, we first identify popular N-grams (in most cases $N = 1$) within the search query (list of consecutive terms within the query) and found extensions to each such N-gram that is also popular. For example, if (ab) is a popular 2-gram, we search for all popular 3-grams of the form (xab) or (abx). The second optimization is to correct potential grammatical errors on individual terms and provide alternatives to these terms to the user. The complete LDC dataset is fairly large (more than 300GB). We restrict the local dataset to only popular shingles which have a minimum frequency of $100K$ on the Web. Even in mobile phone environments (where the phone acts as its local proxy), the popular shingles can easily be loaded within a 2 GB flash that can be used as a local data store.

Local Search: Currently, we use a simple approach to perform local search. Each page cached by the local proxy is associated with a set of terms which includes the list of query terms which fetched the page and a list of important terms in the document. Our current implementation fetches the important keywords in a document including the titles, search keywords, names, section headings and keywords with references. The list of terms in a new query is compared with the list of terms associated with each document and the local search similarity is simply a match of the number of common words above a certain threshold (we use a threshold of 2 for multi-term searches and a threshold of 1 for single-term searches). In the future, we intend to support elaborate local search features (by mining the terms of the downloaded documents) to enhance the search capabilities during disconnected periods.

4.3 Tailoring the Query Response

The local proxy forwards the expanded search along with the query options to the remote proxy. The local proxy in

RuralCafe associates each query with a *search quota* that represents the maximum number of bytes that is allocated as the overall response budget for a query. This quota is calculated based on the available network resources and the number of outstanding requests. Generating the response to a query at the remote proxy involves two steps: (a) issuing the appropriate search query to a search engine to gather search responses; (b) determining what information to be pre-fetched within the allocated quota for a query.

Processing a Query: RuralCafe can support three different forms of user queries: *simple queries, composite queries* and *contextual* queries. A simple query Q is just a collection of terms and this query is forwarded to the search engine as constructed. A composite query is specified in the form of $(M, R_1, R_2, \ldots R_n)$, where M is the main query and each R_i is a supporting term. Such a query can be encoded as a regular expression with each R_i combined using an "OR" clause and this regular expression based query can be issued as an advanced search to a search engine. The composite query is most useful when each R_i represents a specific aspect of the main query M and the individual supporting terms are not directly related. One example of a composite query is if a user would want to download two different papers by a specific author.

A contextual query is specified using a keyword W (within a known set of keywords) representing a specific context which drives a specific automated filtering or extraction process at the remote proxy. Contextual queries are also useful for extracting files of specific types or content from specific domains; for example, Google allows queries of the form "filetype:pdf" or "filetype:ppt" or "site:foobar.com" in the advanced search options. Contextual queries may also be useful for SMS-based queries where the final response from the search engine should be limited to 140 bytes. For example, if the context is specified as "phone number" or "address", then we can use the context to extract phone numbers or addresses from the top search results. We have currently written SMS specific extractors only for a limited set of contexts; doing a detailed contextual query analysis is part of future work.

Customizing the response: For a given query Q, the results gathered by the remote proxy can be classified into three basic categories: (a) Search results; (b) Downloaded or prefetched pages; (c) Embedded media content. The search results refer to the top-M search results along with brief summaries. Downloaded pages refer to the basic files associated with web pages (HTML of various flavors, CSS, ASP, etc.) present in the search result pages. Embedded media content refer to the embedded files (images, audio, video, and other unidentifiable files) in the page.

The overall search quota for the query Q is divided between these categories depending on the *type of response* and *richness* options. For a broad query, a significant portion of the quota is allocated to search results and a smaller portion for downloaded pages and embedded content. For a deep query, a significant fraction is allocated to downloaded pages.

The richness preference of a query response is dictated by the user preference, but ultimately limited by the available quota for a query. For "low" richness responses, we extract only the individual links on search results pages, excluding the various cached links, link previews, advertisements, and other page content. Similarly, for downloaded pages, we do not include the return of embedded media content for each page. Only the bare-bone page HTML for the downloaded pages are prefetched. For "medium" richness of response, we allow embedded images to be prefetched in addition to the text-based files. The large and miscellaneous embedded content including audio, video, and other unidentified files are included to be prefetched in a query response only if the corresponding quota is available and the user explicitly requests a "high" quality of response.

4.4 Intermittent Network Adaptation

We require RuralCafe to work seamlessly across a variety of different intermittent networks. We use a simple model that can be used to parameterize different types of intermittent links based on the *bundle* concept from the Delay-Tolerant Networking (DTN) architecture [7]. Transmissions across a link are batched into bundles where within each bundle we can pack information up to a pre-specified maximum bundle size s. Each bundle upon transmission is associated with a delay d and bundle transmissions are separated by time-periods T. To model budget constrained links, we use a parameter N to represent the maximum number of bundles that can be transmitted within one day. N and T are related in that one parameter imposes a constraint on the other. Typically for an intermittent link, we would use one of the two parameters as the primary parameter to determine the number and spacing between bundle transmissions. In essence, we associate every intermittent link with four parameters: (a) bundle size s; (b) delay d; (c) time-period T; (d) maximum number of bundles N. While we assume s to be a fixed value, d and T can be variable. We use the value of s to determine the search quota and the values of d, T, N to calculate approximate response times.

Estimating Search Quota and Response Time: The local proxy batches queries into "sessions" where a session is a collection of queries issued together within a bundle. Hence, the net quota for a session is s. In mechanical backhaul networks, a session is purely dictated by the arrival timings of the physical transportation; here, a session is a set of queries issued between two arrival events. In budget-constrained links, a session is dependent on N, the maximum number of bundles allowed per day and the arrival rate of user queries. RuralCafe does not make distinctions across different queries. Given a session with K queries, RuralCafe assigns a common quota of s/K for every query for a bundle size s for the link. Note that in the case of SMS or MMS based messaging links, we treat each query separately and set s to be the maximum allowable message size in the underlying network. We calculate the response time as a linear function of the existing queue size and the parameters d, T and N.

5. IMPLEMENTATION DETAILS

The RuralCafe prototype is implemented in 10000 lines of C# code. Both proxies are multi-threaded and cache results on local storage. The client browser is located on the same LAN as the local proxy and is configured to use it as the proxy server.

5.1 Local Proxy

The two responsibilities of the local proxy are to service page requests, and to give estimates on how long the pages will be returned.

5.1.1 Client Interface

As mentioned in the design, we chose not to modify the web browser of the client beyond setting its home page and proxy settings to the local proxy machine. The query interface that is presented is simply a page served by the local proxy that resembles a Google search page with the additional settings for breadth or depth and media richness available. The search interface also consists of a list of the client's previous searches in reverse chronological order along with whether the response package has been received, and an estimated time for arrival for the package if it has not. To issue a query the user fills in the search fields and associated radio buttons and presses search. Then the browser issues the request to the local proxy. The format of the request is similar to that of a Google search request except with additional terms defining the breadth and quality setting associated with the query.

5.1.2 Servicing Requests

The local proxy listens for connections on port 8080 from users and spawns a new thread to service each HTTP request. Each request is served only from the local cache. This is an important distinction as the time spent by the user is not dependent upon the quality link to the Internet. The local proxy maintains a queue of requests per client IP address; the thread adds the latest request into the this list, and time-stamping it with the request time. The thread then serves the requested page from the cache if it exists, otherwise it returns a the default search page. The user is then free to wait for the request to be satisfied, refreshing the page until it is complete and clicking on the link when it is, or to continue searching for other content.

After the thread serves the default page, it forwards the request out to the remote proxy and awaits delivery of the requested page. The client's browser is forwarded back to the default search page. Once the requested page is returned, the local proxy unpacks the page into the local cache and updates the status of the request in its internal state, and the thread is destroyed. The next time the client requests the page either directly or via the link presented on the default search page, it will be served immediately from local cache. The local proxy is also stateful, so if a user logs out of the machine and then returns later his queries will have made progress, and returned results are available for browsing.

The local proxy communicates the quota allocated for the query to the remote server, along with the preferences associated with the search or URL request. The request parameters set by the user are simply embedded into the search URL by the web form to be parsed by the remote proxy.

5.2 Remote Proxy

The primary role of the remote proxy is to service requests made by the local proxy, to prefetch pages filtering out page components according to search request preferences, and finally to return the results to the local proxy. Unlike the local proxy, the remote proxy maintains much less state and is a prefetching agent that speaks the same protocol across the intermittent link as the local proxy and fetches query results on its behalf.

5.2.1 Servicing Requests

Once the remote proxy receives a request it spawns a new thread to serve the request. First, the remote proxy parses in the URL for the parameters and preferences of the request (quota, type, richness). Then the proxy checks its cache for the page, and requests a fresh copy if necessary. If a page needs to be downloaded, the proxy awaits a response just like a normal web proxy. After the proxy receives the page from the server, it stores the page in its cache. Pages are stored locally to simplify the composition of content to be sent back to the local proxy. The remote proxy then creates a single archive (a package) to store the results of the request. If the page requested was a search page, the proxy follows the policy settings given by the local proxy and prefetches the appropriate number of search results to the package. The proxy then fills up the rest of the space with the pages in the search results as best it can according to the breadth and richness setting of the request. If the quota indicates that there is extra space after this, the current policy has the proxy continues to fill in pages and/or embedded elements until the quota is filled. Finally, once the package is full, the remote proxy sends the result back via a HTTP Response headers to the local proxy including a special header "encoding-type=gzip-package" to indicate that it is a gzip package.

5.2.2 Prefetching and Prioritization Solutions

There are clearly many alternative policies for prefetching pages including: filtering and condensing content, or recursively retrieving page links. There are also more sophisticated prefetching algorithms available [21], but these are beyond the scope of this paper.

The remote proxy is also in a position to make decisions about the prioritization between different queries for the link bandwidth back to the local proxy. Our current implementation supports only a simple FIFO allocation where the responses to requests are returned according to when they have been successfully fetched. In the future, we plan to experiment with other allocation strategies based on number of query terms, size of the downloaded search results, differentiating across different types of link characteristics. The key is to keep the algorithms simple so that users have a good idea of what to expect from each of the different search settings. Pricing will dictate the actual bandwidth available, but for the purposes of this work we consider bandwidth allocation across kiosks, users, and terminals to be fixed.

6. EVALUATION

It is difficult to completely evaluate the benefits of RuralCafe without a detailed user study. We have recently deployed RuralCafe at Amrita University in Kerala, India where a low-bandwidth connection serves nearly 10000 students. This low-bandwidth setting is exactly the environment RuralCafe is designed for, and we plan on conducting an extensive study on the benefits of RuralCafe in the near future. In this section, we present the results from evaluating RuralCafe using simulations, benchmarks, and a usability study with a testbed in our lab. We evaluate RuralCafe across three metrics: (a) usability; (b)time saved; (c) adaptability to different network constraints. The analysis for usability and reduction in search rounds is summarized, whereas the results on RuralCafe's adaptability to different network constraints is shown in more detail. In our simulations and benchmarks we use search query logs from search queries collected in an Internet cafe in Cambodia used in a study by Du at al. [6].

6.1 Usability and Deployment

In Jan 2009, we deployed RuralCafe to provide enhanced web search and web browsing capabilities in Amrita University. We intend to perform a detailed user study based on our deployment in the future. To briefly describe our deployment setup, Amrita University and Institute of Medical Sciences together share a 8 Mbps Internet connection, of whcih the connectivity to many units within the institute are bandwidth restricted. Nearly 10000 students share Internet connectivity using 400 machines; this sub-network is bandwidth constrained by 750 Kbps. Hence, during peak time usage, the average bandwidth available per end-host is less than 2 Kbps which is an ideal setup for RuralCafe. We note that Amrita University is among the few universities in India which has very good network connectivity; in many other regions, the bandwidth connectivity is restricted by 128 kbps or 256 Kbps in aggregate.

Currently, much of our usability experiences is from a testbed deployed in our lab and a small user-base in Amrita who have helped us in deploying the system. In our lab setup, we use an artificially constrained link with a search round trip time taking 1 to 5 minutes and with a quota of 1MB. RuralCafe was used in this setting for day to day search queries by students and staff. We solicited feedback on the user experience and the cognitive load of learning and performing search using our new interface.

In our testbed, we artificially limited the bandwidth quota available to 1MB per search in our testbed, and allowed students and staff to use RuralCafe to perform search queries. In general, users quickly understood the queuing concept and adapted to the prefetching of pages along with the estimated time until results are returned rather than on-demand search. Users were able to refine queries locally using the query construction framework, and make progress in their search sessions using the cache while the network was bottlenecked. Though the local search was a bit cumbersome at first, the query construction assistance and estimated number of results returned provided enough detail for users to construct queries that more accurately expressed their intentions. Users at both Amrita and our lab, in general, found the estimated time for completion as a useful bit of information.

6.2 Idle Time Reduced

The amount of time saved is clearly dependent the connection quality to the Internet; the slower the connection, the more time is saved by minimizing search rounds over the Internet. While the connection quality varies with the setting, the number of search rounds does not. To get a lower bound on the number of search rounds are in a typical search session we performed simulations using our search logs and compared this with previous results. In these simulations we aggregate the search queries into search sessions by checking for matching terms across neighboring queries. A search session is indicative of the number of rounds of interaction of a user for a single search. Our results showed that roughly 80% of the queries were present in search sessions that were not satisfied in a single round (i.e. more than one query per session). In fact more than 30% of the search sessions involve at least four queries. In a larger study from AOL [18], Pass et al. found that 28% of queries are re-formulations of the previous query, and that in these cases the query is reformulated an average of 2.6 times. In the intermittent world, searches that require more than a single round may increase the time for meaningful results by a factor of minutes, hours, and even days. This indicates that providing an interface that gives the user results in one round would increase the practicality of one-round web search.

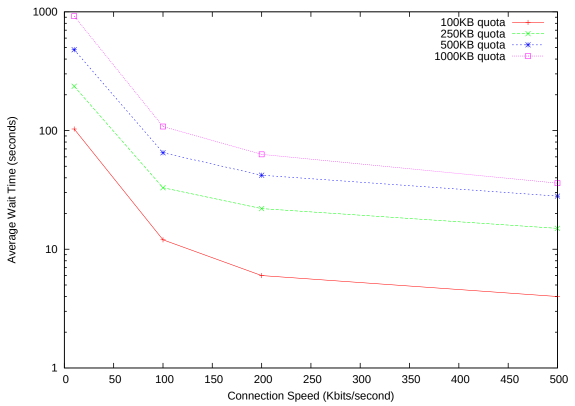


Figure 3: Average Wait Time vs Connection Speed

In our testbed the most common feedback was that users spent more time navigating the suggested search terms along with the cached search results to come up with a more clearly specified query. However, after adding the desired query to the queue, users felt comfortable leaving the search to run and coming back at a later time. While it is true that query construction assistance and local search of the cache sometimes causes users to perform more total search rounds per session, by the time the search query actually uses the network the query is so well refined that often only a single round is necessary.

6.3 Adaptability to Different Networks

Prefetching Flexibility: Using the search logs available to us, we wanted to explore whether RuralCafe could offer a gradation of services for variable quota levels ranging from over 1 MB (backhauls) to only 100 KB (dialup). We found that for larger quota sizes, RuralCafe can prefetch several search result pages as part of the search response to significantly enhance the subsequent local browsing experience. For small quota sizes, the preferences would still allow RuralCafe to cheaply return valuable pieces of information. Specifically, for each bandwidth budget we wanted to know the grade of service a user could expect to achieve using RuralCafe.

For this analysis we first measured the sizes of the results returned by typical queries. From our search logs we fetched the search results page for 1000 queries to www.google.com. These search result pages contain links to target pages and short descriptions of the target page contents. Each search result page had an average of 13 of these links to target pages. We downloaded the target pages on each of these search result pages to get an idea of the relative sizes of pages and their contents. Along with all pages downloaded we included the embedded content. pages. Figure 4 is a histogram of the file sizes. The Y axis is in log scale and cut off at 10000 files to preserve the scale, and file sizes are cut off at 200KB.

Table 2 summarizes the average sizes of the types of con-

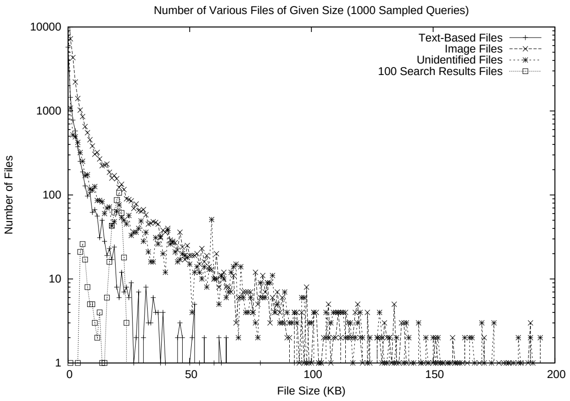


Figure 4: Composite of sizes of all page types

Table 2: Average Page Sizes

Page Type	Uncompressed	Compressed
Search Results (Full)	25.0KB	7.7KB
Search Results (Text)	23.6KB	6.9KB
Target Page (Full)	162.2KB	102.8KB
Target Page (Text)	60.2KB	18.8KB

tent we downloaded along with the compressed sizes. We observe that filtering out images reduces target page sizes a great deal (62.9% less) as compared to search result pages (5.6% less). Compression helps reduce search result pages and target pages dramatically in size especially text pages (70%). These results indicate that different classes of service may be provided by using filtering schemes that match user preferences.

Cost of Search: What does an average search query cost and what does a response contain? As an example, if we have a 100KB search result budget, using the values in Table 2, RuralCafe would be able to return 10 search pages with 10 results and short descriptions per page. The same 100KB budget could alternatively be used to return a full target page if desired. Requests of these types would cost only fractions of cents in a variety of different settings using 100KB budgets over satellite in India, and smaller 10KB budgets over GPRS in Africa. Even at 10KB budget sizes useful results can be returned without any specialized filtering or information extracting algorithms.

We can also compare RuralCafe responses with those of normal web requests. If the search result budget is set to be equal to the size of one page then RuralCafe's per query cost is the same as normal web searches. However, if the budget is set to be larger than a typical result then the remote proxy will prefetch additional pages. Prefetching pages causes each query to cost more, but each response would have an increased chance of satisfying the request in a single iteration. The search budget offers a natural tradeoff between the acceptable per query cost and the expected number of round trips. Moreover, since queries may be more precisely specified in RuralCafe, each byte retreived is more "valuable" than the bytes in a normal web response.

Search Response Time: To understand how well RuralCafe performs in different network conditions we performed microbenchmarks under different emulated settings.

Table 3: RuralCafe Component Processing Time (1MB budget, 10Kbps)

Component	Time Spent
Local Proxy (processing)	under 1s
Remote Proxy (processing)	under 1s
Remote Proxy (downloading pages)	18s
Remote Proxy (transmitting pages)	810s

For different search result budgets we varied the bandwidth of our emulated link while keeping the network latency under 10ms. We performed 1000 search queries using the default RuralCafe richness and prefetching settings. Figure 3 shows the resulting average wait time depending on the connection speed. We confirm that RuralCafe is capable of providing service across the spectrum of connection speeds.

Since RuralCafe has not been optimized for speed, some processing time is necessary at the proxies to service each query. For completeness, Table 3 summarizes the time spent on various tasks by RuralCafe for a 1MB search response budget. We observe that the overhead is small compared to the time spent transferring data across the slow 10Kbps connection. The time spent on processing by the remote and local proxies becomes even more negligible as the round trip time increases.

7. RELATED WORK

We classify related work into four categories: (a) delay tolerant network design, (b) web search in intermittent contexts, (c) Web caching, and (d) low bandwidth adaptation.

Delay tolerant network design: The Delay Tolerant Networking (DTN) architecture [7] models each intermittent link as a time-varying delay tolerant link where each link is associated with a delay and a capacity both of which vary with time. Correspondingly, the DTN research group has developed several routing and addressing protocols to route packets between any pair of nodes within the network [10]. RuralCafe operates independent of the DTN routing algorithms at the underlying layer.

One theme where RuralCafe differs in philosophy from DTN protocols is to expose the intermittency to the end-user. DTN shields the higher layers from the unreliable network layer and from the application's perspective, DTN replaces IP by providing an interface of URI's and DTN demultiplexers on top of TCP and UDP in the network stack [7]. By explicitly exposing the intermittency to the user, RuralCafe is able to enable the end-user to steer the search process. Our belief is that users would perform more intelligent and specified searches from constrained intermittent environments.

Web search in intermittent contexts: Intermittent links occur in a variety of scenarios outside of the developing world context. These include flaky wireless links [8, 2], mobile nodes continuously changing access points [33], vehicular networks in urban settings [2, 16], postal networks [29]. RuralCafe is a proxy level solution seeking to provide web search and browsing across all types of network connectivity. In the context of the developing world, the TEK system [11, 27] provides a non-interactive search mechanism where a user types in a search query using E-mail and the search result is asynchronously sent back to the user through E-mail. DakNet [20] is a top down approach that leverages TEK to allow users to perform everyday search queries, browse the web, and check email. DAKnet uses physical transportation links such as buses to transport bits. DakNet addresses the issue of bringing the Internet to rural areas by using simple store-and-forward mechanisms of SMTP. caching and a web proxy interface. RuralCafe does not modify the application itself, but leverages proxies to deal with the intermittency of the network in an application specific manner.

Web search from a bus (Thedu) [2, 1] is also an interesting approach to a similar problem of performing web searches in an urban environment while mobile open access points are readily available. Though the problem domain is different, the elements of prefetching and result prioritization are in common with our solution. Thedu focuses on predicting the likelihood of prefetching hits based on query-likelihood models, we avoid the complex problem of deciphering user intention altogether as it is independent of our communication protocol and application design.

Web Caching: Web caching is a very well studied topic over the past two decades and there have been several caching optimizations that have been proposed for low-bandwidth networks [32, 21, 9]. The work by Du [6] analyze web access traces from Cambodia to analyze the effectiveness of simple caching strategies in developing regions. A followup work by Isaacman and Martonosi [9] show the potential for collaborative caching and prefetching in rural developing regions. Specifically, their result shows that prefetching appropriate pages can enhance the power of local cache-based search in rural traces. These caching and prefetching strategies can be used in RuralCafe to enhance the local search mechanism.

Low-bandwidth content adaptation: Similar to web caching, there have been several works in the space of content adaptation in low-bandwidth networks that RuralCafe can leverage. Specifically, there have been several works on lossy and loss-free compression routines for different forms of media (video, audio, images) which can be applied in RuralCafe. A detailed description of these techniques is outside the scope of this paper. Loband [13] is a system that enables users to view filtered text-only versions (or text +images) of web pages in low-bandwidth environments.

8. CONCLUSION

In the developing world, especially rural areas, network connectivity is unreliable, and bandwidth is a scarce commodity. While some connectivity is being established using newer technologies, high rates continue to cause bandwidth and constant connectivity to be rare. As a result, typically synchronous activities such as web search and browsing become frustrating or practically impossible to use. This problem is only exacerbated by the typical user's search pattern of iteratively refining queries. Traditional techniques such as compression and caching are not enough to hide the intermittent nature of these extreme networks. In order to address the problem of web search in these environments we built RuralCafe, a system that is specifically designed to enable users to efficiently perform web search queries in one round across an intermittent network.

RuralCafe desynchronizes the search process by performing many search tasks in an offline manner. The actual retrieval of information using the network is only done when absolutely necessary, and the user is given precise feedback on when to expect results. RuralCafe also allows users to easily express the type of query response they expect so the bytes retrieved have a higher likelihood of actually being useful. We have shown that RuralCafe enables efficient web search over a range of different intermittent environments. A prototype of version of RuralCafe is currently deployed at Amrita University in Kerala, India.

9. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback which has improved the content and presentation of this paper. We also thank the students and staff at Amrita University for facilitating our study and hosting our initial deployment.

This material is partially based upon work supported by the National Science Foundation under the following grants: CNS-0832153 and CNS-0831934.

10. REFERENCES

[1] A. Balasubramanian, B. Levine, and A. Venkataramani. Enhancing Interactive Web Applications in Hybrid Networks. *ACM MOBICOM*, 2008.

[2] 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.

[3] J. Chen, L. Subramanian, and K. Toyama. Web Browsing under Poor Connectivity. *CHI Proceedings on Human Factors in Computing Systems (Work In Progress Session)*, 2009.

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

[5] M. Demmer, K. Fall, T. Koponen, and S. Shenker. Towards a Modern Communications API. *Hotnets*, 2007.

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

[7] K. Fall. A delay tolerant network architecture for challenged internets, 2003.

[8] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: a distributed mobile sensor computing system. *Sensys*, pages 125–138, 2006.

[9] S. Isaacman and M. Martonosi. Potential for Collaborative Caching and Prefetching in Largely-Disconnected Villages. *ACM MOBICOM WiNS-DR workshop*, 2008.

[10] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. *SIGCOMM Comput. Commun. Rev.*, 34(4):145–158, 2004.

[11] L. Levison, W. Thies, and S. Amarasinghe. Providing Web search capability for low-connectivity communities. *ISTAS*, pages 87–91, 2002.

[12] Linguistic Data Consortium. http://www.ldc.upenn.edu.

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

[14] A. Meacham. The Case for SmartTrack. *NYU Technical Report*, 2008.

[15] S. Mubaraq, J. Hwang, D. Filippini, R. Moazzami, L. Subramanian, and T. Du. Economic analysis of

networking technologies for rural developing regions. *Workshop on Internet Economics*, 2005.

[16] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11 b for" automobile" users. *INFOCOM*, 2004.

[17] J. Ott and D. Kutscher. Bundling the Web: HTTP over DTN. *WNEPT*, 2006.

[18] G. Pass, A. Chowdhury, and C. Torgeson. A Picture of Search. In *First Intl. Conf. on Scalable Information Systems*, 2006.

[19] 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.

[20] A. Pentland, R. Fletcher, and A. Hasson. DakNet: rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

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

[22] D. E. Rose and D. Levinson. Understanding User Goals in Web Search. In *WWW*, May 2004.

[23] Rural BPO. http://www.icmrindia.org/casestudies/catalogue/Businesstm.

[24] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. *Mobicom*, pages 334–345, 2006.

[25] L. Subramanian, S. Nedevschi, R. Patra, S. Surana, A. Sheth, and E. Brewer. Rethinking Wireless for the Developing World. *Hotnets*, 2006.

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

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

[28] United Villages. http://www.unitedvillages.com.

[29] R. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the Postal System into a Generic Digital Communication Mechanism. *SIGCOMM*, 2004.

[30] WiMAX forum. http://www.wimaxforum.org.

[31] Yahoo One Search. http://mobile.yahoo.com/onesearch.

[32] 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.

[33] X. Zhang, J. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. *Mobicom*, pages 195–206, 2007.