Systems for Challenged Network Environments

by

Azarias M. Reda

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2012

Doctoral Committee:

Associate Professor Brian D. Noble, Chair
Professor Paul N. Edwards
Associate Professor Zhuoqing Morley Mao
Assistant Professor Prabal Dutta

To Abiti and Astu

ኃይልን በሚሰጠኝ በክርስቶስ ሁሉን እችላለሁ።

ወደ ፊልጵስዩስ ሰዎች ፬ ፲፫

ACKNOWLEDGEMENTS

My advisor has taught me many things. Two lessons, however, have particularly made this dissertation possible. One was to find something that I could get passionate about—so much so that I couldn't stop thinking about it as I went to bed and woke up. Grad school is hard—projects die and papers that took months to write get rejected. But knowing why I do what I do, and why I care about it has always given me some ground to stand on. Brian made sure that I could work on what I wanted to work on.

The other lesson was teaching me it is easier to ask for forgiveness than it is to get permission (I would later learn that this phrase was due to Grandma COBOL, Grace Hopper). Brian would be the first to tell you that I sometimes went on to work on ideas and projects that didn't sound fruitful. Sometimes, they weren't. But I always knew that I would get a chance to explain myself, and convince him why we should continue working on them. I was free to take risks, and that taught me to take responsibility for my PhD. I didn't appreciate this freedom early on, but I wouldn't have it any other way now.

So, Brian—I sincerely thank you. I owe this dissertation to your brilliant, build-your-own-PhD style of advising.

I would also like to thank Jason Flinn, who co-advised me on some of my earlier projects, and continued to take interest in the later years. It is pleasant to work with smart people who know their stuff. The Mobility group has a great bunch of people who will go out to do amazing things. I was honored to work with you folks. I had wonderful collaborators at Microsoft Research India, Addis Ababa University, LinkedIn and of course, here at Michigan. Thank you all for working with me and I have learned immensely from my interactions with you. My PhD was also supported by the graduate research fellowship program from the NSF, which allowed me to focus on my work.

The faculty and staff in CSE were always helpful. Steve, Dawn and many other dedicated people make life much easier for students like me. Thank you. My only complaint is that I never got the certificate for my “Best Halloween Costume” victory in the fall of 2011.

I was blessed with many wonderful friends—the past few years were a lot better because of you. Thank you for putting up with me and talking to me when I was up and down.

My family has sacrificed so much to get me here. I love you. We did it, praise be to the Lord.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

<u>**Table**</u>

ABSTRACT

Systems for Challenged Network Environments

by

Azarias M. Reda

Chair: Brian D. Noble

Developing regions face significant challenges in network access, making even simple network tasks unpleasant and rich media prohibitively difficult to access. Even as cellular network coverage is approaching a near-universal reach, good network connectivity remains scarce and expensive in many emerging markets. The underlying theme in this dissertation is designing network systems that better accommodate users in emerging markets. To do so, this dissertation begins with a nuanced analysis of content access behavior for web users in developing regions. This analysis finds the personalization of content access—and the fragmentation that results from it—to be significant factors in undermining many existing web acceleration mechanisms. The dissertation explores user access behavior from two vantage points—logs collected at shared internet access sites in Southern India, as well as user activity information obtained from a commercial social networking service with over a hundred million members worldwide. Not only are users accessing increasingly personalized content, the importance of rich media is also growing in emerging markets.

Based on these observations, the dissertation then discusses two systems designed for improving end-user experience in accessing and using content in constrained networks. First, it deals with the challenge of distributing private content in these networks. By lever-

aging the wide availability of cellular telephones, the dissertation describes a system for personal content distribution based on user access behavior. The system enables users to request future data accesses, and it schedules content transfers according to current and expected capacity. This system was deployed in partnership with an internet kiosk in Addis Ababa, Ethiopia. Second, the dissertation looks at routing bulk data in challenged networks, and describes an experimentation platform for building systems for challenged networks. This platform enables researchers to quickly prototype systems for challenged networks, and iteratively evaluate these systems using mobility and network emulation. The dissertation describes a few data routing systems that were built atop this experimentation platform.

Finally, the dissertation discusses the marketplace and service discovery considerations that are important in making these systems viable for developing-region use. In particular, it presents an extensible, auction-based market platform that relies on widely available communication tools for conveniently discovering and trading digital services and goods in developing regions. This market platform was used to provide easy access to two systems designed for challenged networks, and was also deployed in Ethiopia. Collectively, this dissertation brings together several projects that aim to understand and improve end-user experience in challenged networks endemic to developing regions.

CHAPTER I

Introduction

Developing regions face significant challenges in network access. Bandwidth is an important resource in today's global knowledge economy, but it is scarcest where it is most needed—in the developing nations of the world which require low-cost communications to accelerate their socioeconomic development [60]. Even as cellular network coverage is reaching a significant majority of the world's population, good network connectivity remains scarce and expensive in many developing regions [1]. For example, on average, the cost of fixed-broadband as fraction of income per capita is more than an order of magnitude higher in Africa compared to Europe. A number of reasons, such as expensive international bandwidth, inefficient government monopolies, lack of proper peering points, low user density and poor provisioning contribute to this. For many internet users in these regions, poorly connected shared access sites remain the primary alternative of network access. As a result, users have an unpleasant experience when using the internet for anything that involves more than a few hundred kilobytes of data transfer [28].

On the other hand, the digital universe has been expanding rapidly, increasing the demand for efficient transfer of data over the network. In addition, the falling cost of computing and storage devices has accelerated the rate of content generation. Unfortunately, a significant portion of the world's population lacks easy access to digital information—a gap that exemplifies the 'digital divide' between developing regions and the developed world. For example, at a typical internet kiosk with a shared dial-up connection, accessing a single 10MB audio lecture freely available from the University of Michigan's Open Initiative would take several hours to complete, rendering it effectively out of reach for a student who has to pay by the minute for internet access. A video lecture, which often has hundreds of megabytes, is often simply too large to access.

The underlying motivation for this dissertation is investigating mechanisms for improving end-user experience in accessing and consuming content in challenged network environments—environments where bandwidth is often low, latency is high and the price of network access is expensive. To this end, this dissertation leverages widely available resources, alongside observations of data access behavior, in building systems that can better accommodate users in developing regions. For example, cellular devices have attained a critical mass in developing regions with over 5.3 billion mobile subscribers worldwide [1]. Cellular coverage is now available to over 90% of the world's population. On the other hand, users are creatures of habit and establish patterns when accessing content. This dissertation combines these observations in designing mechanisms for improving end-user experience in challenged network environments.

This dissertation makes multiple contributions in this regard. First, it describes a number of studies conducted to better understand content access and web usage behavior in developing regions. Some of the patterns observed are unique to the developing world, often shaped by economic, social and cultural factors, or the brief history of internet citizenship for many users in these environments. Other patterns transcend geographic and economic barriers, and derive from basic human social behavior in communication, sharing and interaction. This dissertation uses two lenses in studying access behavior—a personalized study of web usage from data collected at shared internet access sites in a developing country, as well as an aggregated look from the perspective of a worldwide internet service with millions of users in developing regions. These observations provide a good picture for web usage in developing regions.

Based on these observations, the second contribution is introducing a number of systems for improving end-user experience in challenged network environments. This dissertation incorporates widely available resources in designing systems that can be incrementally deployed on the existing infrastructure. Some of these systems are targeted at users in these regions, while others provide tools for researchers who are building systems for challenged network deployment. In addition, it looks at the service-discovery and marketplace considerations that are important in scaling these systems. Some of these systems have been deployed in developing countries in Asia and Africa, and this document reports on the experience. As developing regions represent important emerging markets for digital services, the systems described in this dissertation provide a few examples in designing appropriate technologies for these markets.

The work in this dissertation falls under a large class of research referred to as Information and Communication Technologies for Development (ICTD). The basic premise for this line of research is that appropriately designed technologies can facilitate efforts towards bringing about development to various regions throughout the world. While global development and the impact of many development programs is often debated, the argument for ICTD is simple and straightforward—by creating an efficient infrastructure for delivering information, services and goods, ICTD research can play an important role in aiding developmental efforts. On the other hand, as computer scientists, developing regions are interesting because they present a unique set of challenges in designing and building systems.

This chapter will begin the discussion with the thesis statement for the dissertation. It also provides an overview of the document and summarizes the contributions in the following section.

1.1 Thesis Statement

The thesis for this dissertation is that:

> *The wide availability of cellular telephones and cheap storage devices, along with patterns in human activity, can be used to significantly improve end-user experience in accessing and using data in challenged network environments.*

This dissertation makes several contributions in validating this thesis. The research builds on prior work in distributed systems, ad-hoc networking and delay tolerant networks to improve data access under poor network conditions. It leverages widely available technologies, weaved with the traditional network stack, to build systems that can be incrementally deployed on the existing infrastructure. In doing so, it will cover three main areas:

1. Understanding usage: The dissertation begins by looking at data access behavior in developing regions. It does this from two vantage points—data access information collected at shared access sites in a developing country, as well as user activity information from a large scale social network with worldwide membership.

2. Improving data transfer: Informed by these observations of web access in developing region contexts, this dissertation describes systems designed for improving end-user experience in two basic usage scenarios—consuming private content and transferring bulk data. It also describes a platform for experimenting with data delivery systems in challenged networks.

3. Making services viable: In order to make these systems viable in developing regions, marketplace and service discovery issues need to be addressed. This dissertation describes a simple, voice and SMS-powered market/auction platform to facilitate the trust and market relationships required for these approaches to scale. This market platform can also be used more generally.

The rest of this document will discuss the contributions along these areas.

1.2 Overview of the Dissertation

This dissertation is organized in three parts, corresponding to the three areas mentioned above. Part One discusses studies in understanding data access behavior in developing regions. There are two chapters in this section, focusing on two different approaches in studying web usage in developing regions. Chapter III describes a large scale study of data access at shared access sites in developing regions. Web usage is getting increasingly personal and fragmented. This makes traditional web acceleration systems that rely on redundancy in user requests for optimizing performance less useful. The study of individual web usage in shared access sites makes three contributions. First, it provides the first personalized, large scale web usage data in developing region contexts. This allows researchers to get a nuanced understanding of access behavior that is not offered by aggregate data. Second, it presents an analysis on this dataset, which provides tangible evidence for describing the increasingly fragmented and personal nature of web access even in developing countries. Finally, based on lessons learned from the analysis, it provides some observations for building effective web acceleration mechanisms in the face of an increasingly personal and media-rich web.

Chapter IV provides a detailed case study in further understanding data access behavior in developing regions by focusing on user participation in online social networks. Online social networks have enjoyed significant growth over the past several years. With improvements in mobile and internet penetration, developing countries are participating in increasing numbers in online communities. This chapter provides the first large scale and detailed analysis of social networking use in developing region contexts. The analysis is based on data from LinkedIn, a professional social network with over 100 million members worldwide. LinkedIn has members from every country in the world, including millions in Africa, Asia, and South America. The goal of this analysis is to provide researchers a detailed look at the growth, adoption, and other characteristics of social networking usage in developing countries compared to the developed world. By extension, this work provides a close-up look of one common data access vertical in developing regions. These observations from chapters III and IV are combined in designing and building systems for assisting with two common usage scenarios—consuming private/personal content and transferring bulk data in challenged network environments.

Part Two of the dissertation describes several systems for improving end-user experience in accessing and using data in challenged network settings. Chapter V describes an infrastructure for distributing private data in challenged environments. Many standard techniques for improving user experience in challenged networks—caching and predictive prefetching—help somewhat, but provide little or no assistance for personal data that is needed only by a single user. This dissertation propose a solution to this problem by extending the idea of individualized content distribution networks that take personal usage patterns into account. The system, called Sulula, addresses this problem by leveraging the near-ubiquity of cellular phones able to send and receive simple SMS messages. Rather than visit a kiosk and fetch data on demand—a tiresome process at best—users request a future visit. If capacity exists, the kiosk can schedule secure retrieval of that user's data, saving time and more efficiently utilizing the kiosk's limited connectivity. When the user arrives at a provisioned kiosk, she need only obtain the session key on-demand, and thereafter has instant access. In addition, Sulula allows users to schedule data uploads. Experimental results show significant gains for the end user, saving tens of minutes of time for a typical email/news reading session. The system also had a small deployment in-country for proof-of-concept, and this dissertation describes lessons learned from that experience.

The next focus is on the challenge of transferring bulk data over poorly and intermittently connected networks. One of the observations in this area was the difficulty of experimenting with systems for challenged networks. Systems designed to work in these settings must support communication through intermittent links and mobile nodes despite poor infrastructural support. Therefore, it is important to experiment with a variety of designs, evaluating them under a variety of circumstances. Unfortunately, these systems tend to be large and complex, with a high implementation bar for even simple functionality. Pure simulation avoids this, but carries with it simplifications that hide many important issues in understanding the overall system performance. To resolve this issue, this dissertation introduces Vivo, a platform that enables researchers to rapidly implement challenged network systems, and reproducibly evaluate them at scale. Vivo, discussed in Chapter VI, has two parts. It provides a toolkit of composable and replaceable component implementations, alongside an evaluation platform with network and mobility emulation to reproducibly evaluate the implemented systems. The toolkit provides a menu of composable components that interact over well-defined interfaces, allowing researchers to focus on specific problems, while providing suitable defaults for other behavior.

Chapter VII presents a new hybrid overlay network that was built atop Vivo. This overlay network, known as Bati, introduces an approach for orchestrating bulk data transfer through a series of hosts with spare storage and diverse network connectivity. By combining natural individual mobility and available network connectivity, Bati forms a hybrid overlay network for delivering bulk data while preserving scalability in the state required to do so. For comparison, two previously suggested data transfer systems are implemented with Vivo. The evaluation shows substantial improvements in using Bati for delivering data, transferring an order of magnitude more data than the network alone, and improving the delivery rate by more than 40% compared to popular ad-hoc networks. In addition to evaluating various design alternatives within Bati, its evaluation is used to demonstrate Vivo's capacity to facilitate repeatable evaluations of live implementations for challenged network systems.

Part Three provides a discussion on the marketplace and discovery issues that are important in making systems like the ones discussed above viable for developing-world use. In the case of Sulula, for example, providing a simple way for users to compare offers from various providers, while transparently arranging for data transfers, is a desirable property for practical use. While electronic commerce has decidedly changed how services and goods are traded in the developed world, the impact remains largely unnoticed in many developing countries. This is often due to low network penetration, lack of locally relevant markets, and requirements for additional facilities (such as credit cards, shipping arrangements etc.) to take advantage of such marketplaces. Even when these markets are established with local content and poor connectivity in mind, they are often specific to a certain domain or community. To mitigate this problem, and allow for wide adoption of digital services, this dissertation introduces an extensible auction-based market platform for use in challenged network environments. This platform enables developers to incorporate a market layer in their applications and open their services to a wider audience, while considering the limited operating environment where communication channels are narrow and potentially expensive. Chapter VIII describes this platform and the empirical and deployment based evaluations.

Finally, Chapter IX will revise the key concepts in the dissertation and provide a summary of the contributions. In addition, it will discuss some future directions for this line of research in the short and long term.

CHAPTER II

Background

This dissertation is built on several projects ranging from information studies in the context of developing regions to systems research and software engineering. This chapter looks at some of the background work that is important in understanding the contributions made in the dissertation. The discussion starts by looking at projects that attempt to understand end user behavior in web access in developing regions, and how these projects relate to the analysis on the impact of personalization in web access. The chapter then describes several systems that have been proposed for improving end-user experience in accessing and using data in challenged networks. These projects provide the underlying work for the systems that will be described in this dissertation. Finally, the chapter discusses marketplace solutions for developing regions, with an eye towards the extensibility that is provided by Robit, the auction based marketplace described in this dissertation.

There have been several studies that consider web usage in developing world scenarios [19, 20, 32, 55, 57, 63]. The datasets for these studies are generally collected from client or proxy level loggers in various contexts and can be characterized as aggregate views into web access in developing countries. Du et al. looked at HTTP traffic captured from internet kiosks in two developing countries [32]. Their results point out various features of web usage in developing countries. Work by Ihm et al. [55] analyzes a large amount of data collected at a global scale and tries to observe differences in developed and developing world traffic. Their dataset comes from a worldwide proxy server that has access to the content of requests in addition to access logs. More specific web access analysis in schools and universities [19, 20] has considered traces obtained from educational environments in developing countries. The dataset in this dissertation differs from these projects because it captures personalized web usage of individuals in a developing country context, allowing for a finer grained analysis of web access behavior. As web experience gets increasingly personalized, a deeper understanding of user behavior is essential in designing appropriate systems for challenged network environments. This dataset is available for researchers to further investigate individual behavior in web access for developing country contexts.

Web usage acceleration is one important class of services for challenged networks. There has been a lot of interest in this area of research [12, 20, 56–58, 85, 98]. In environments where network connectivity is limited and expensive, web acceleration techniques play an important role in improving end user experience. Caching and prefetching are some of the widely used techniques for accelerating web access, with several flavors of implementations and various levels of success. The work in this dissertation provides tangible evidence for considering personalization as an important design factor for such systems. As web usage gets more and more personal, acceleration techniques built atop an aggregate analysis of their user base will be hard pressed to cope with diverse behavior and content. This challenge is well displayed when considering several examples of systems designed for improving web-access experience. Some of these systems attempt to utilize common communication tools for interacting with users.

There are a number of systems that use SMS for accessing advanced services provided over the network, allowing even basic handsets to do things like search [46] and community organizing [41]. Warana Unwired [127] looked at replacing an existing PC-based system for managing information in a sugarcane cooperative with an SMS-based mobile phone system in rural India. The solution enables farmers to get operational information such as sales and orders without having to make a trip to the central office. By sending an SMS messages with formatted text, the farmer would get back a response with the requested information, such as sugarcane output for the season. Some of the work discussed in this dissertation is similar to these systems in using SMS as a bridge to services on the network. It builds on the idea by leveraging the wide availability of mobile phones for delivering private data in challenged networks.

Analysis of aggregated WWW traffic in developing countries [32] has shown that the web traffic pattern is generally in-line with other studies elsewhere, following a Zipf-like distribution. A sizable portion of this traffic is personal in nature, ranging from webmail and educational websites to a large number of niche destinations that form the long tail of the distribution. After studying the traffic patterns for these countries, the authors also suggest some mechanisms that could help to improve performance of web access in those regions, such as client access methods for mail, converting software updates from polling to pushing, curbing irrelevant advertising and offline caching. This dissertation incorporates a number of their suggestions, and makes other performance enhancing techniques easier to implement.

Prefetching data has been widely studied [38, 39, 62, 71, 72, 78, 96] in both file system and web contexts. This idea is extended to private data in challenged network environments. For example, data staging [39] looked at opportunistically prefetching files and caching them on nearby surrogate machines to facilitate secure mobile data access. The system is designed to improve the performance of distributed file systems running on small, storage-limited pervasive computing devices. Although this work is similar to the Sulula system in prefetching content for ease of access later, Sulula is targeted at challenged network environments where bandwidth, rather than storage, is the limiting factor. In addition, it is adapted to usage patterns in developing regions.

HashCache [12], a configurable cache storage engine targeted at the 'next billion', runs at a very small memory footprint enabling multiple terabytes of external storage to be attached to relatively low-powered machines. While efficient cache storage such as Hash-

Cache reduces bandwidth consumption, and thereby the network connectivity expenses, Sulula considers private or personal data where traditional caching mechanisms are not very useful.

Another area of web caching that has been studied well is delta-encoding [86, 107]. This approach looks at transmitting the difference between requested objects rather than whole objects. Although it requires cooperation from the server, this approach can be especially useful when dealing with requested objects that only change slowly. The Sulula provides a simple platform to implement delta-encoding without the cooperation of the data source as it can be incorporated with data aggregation points. In a similar fashion, value-based web caching [104], which uses the content of documents rather than URLs to make cache indexes, is straightforward to implement using Sulula.

Dittorent[106] looked at leveraging peer-to-peer technology to enable offline internet access at modem-speed dialup connection. The solution is implement using a browser plugin which looks up all file requests in the P2P network before downloading them from the internet. If the requested file is available in the network, it is downloaded at modem-speed from local peers. If it is not available, the file is downloaded from the source and then added to the P2P network for future use. However, this approach is unlikely to be useful for private data.

Other approaches for improving connectivity in challenged environments include optimizing high latency links [10] and delay tolerant networking [37]. The former suggests using simple QoS prioritization to provide more fine-grained, adaptive and behavioral classification of network flows rather than purely based on protocol and port, in order to optimize for latency for shorter flows whose performance depends more on latency. DTN proposes a network architecture and application interface structured around optionally-reliable asynchronous message forwarding, with limited expectations of end-to-end connectivity and node resources. Such an architecture can be used to achieve interoperability between networks deployed in mobile and extreme environments that often lack continues connectivity.

The next focus in the dissertation is transferring bulk data in challenged networks. In particular, the dissertation focuses on an experimentation platform targeted at researchers as they build data delivery systems for challenged networks. This platform is known as Vivo. Vivo is a combination of two distinct ideas for experimenting with challenged network systems. It provides a toolkit for rapid prototyping of systems with live-code, and a platform for a full-stack evaluation of these systems through emulation. While these ideas have been explored in various contexts before, other platforms do not streamline both services for challenged network systems. Vivo is best described as a collection of reusable and interoperable components that can be easily composed, modified and replaced for implementing challenged network systems, alongside a platform for evaluating these compositions.

In terms of providing an architecture for building network systems, Vivo is most similar to projects like the Click modular router [88] or the Rover application toolkit [65]. Akin to these projects, Vivo simplifies the process of building systems for a well-defined application area. Unlike previous toolkits, it focuses on the challenges of designing systems for environments characterized by diverse and intermittent connectivity. Vivo is not a reference implementation of an existing protocol, or a monolithic system for iterative improvement [102]. Rather, it relies on well-defined interfaces and dynamic object passing for flexibly composing end systems by focusing only on components researchers find interesting (Section 6.2).

Although there is no true replacement for real-life testbeds like DieselNet [120], which has over 30 networked busses, or vehicular extensions to the ORBIT grid [67], the cost of in-situ evaluations is often prohibitive. For a reasonable middle ground, Vivo borrows concepts from network and mobility emulation frameworks [99, 101, 130, 134] to enable a full-stack evaluation of challenged network systems. It relies on testbeds such as EmuLab or ORBIT for network emulation, and incorporates a mobility layer atop. Optionally, researchers can scale their experiments by using managed clusters like Amazon EC2 with traffic shaping. In emulating mobility, the platform models node encounters rather than faithfully replicating motion trajectories [99, 134]. This allows the representation large geographic areas even on a small cluster, while capturing the essential outcome of mobility in challenged network systems (Section 6.2.3).

Many projects simulate the functionality of challenged network systems and evaluate them using network and mobility simulators. For example, discrete event simulators are commonly used to evaluate opportunistic routing protocols [64, 80]. Mobility generators are available for ns-2, and tools such as JANE [47] provide support for MANETs. More specific simulators like ONE [68] provide a useful set of tools and reference implementations for mobility based systems. The Vivo platform generally aims to preserve the flexibility and reproducibility of simulation based experimentation. However, the focus is providing a platform for rapidly prototyping challenged network systems with live-code, and iteratively evaluating these live implementations.

When it comes to systems for moving bulk data in challenged networks, related work comes from two principal areas—ad-hoc networking and delay/disruption tolerant networks. Ad-hoc networks are concerned with cases where central coordination is not possible, and infrastructure is not present. Nodes must self-organize to deliver messages, often communicating with some radio only when in range. Delay tolerant networks form a confederation of regional networks with in-network storage for delivering messages across boundaries. The following paragraphs discuss several systems from each class and describe how the work in this dissertation relates to them.

Earlier approaches in ad-hoc networks used epidemic style forwarding among mobile nodes [43, 125], which was inefficient in resource utilization. Subsequent approaches [13, 16, 21, 77, 80] improve on epidemic routing by prioritizing how likely packets are going to be delivered from each node, taking into account how often nodes meet each other. Generally, these systems do not focus on connectivity beyond direct wireless links. Some require a form of centralized resource management [49], or flood the network for global information and acknowledgment [16]. In addition, they require the routing computation and state kept at each node to grow linearly with the system, presenting difficulties in scaling to a wide-area system in the developing world. Bati, the system described in this dissertation, builds on mobility lessons learned from these and similar systems, and integrates them into a hybrid network that can use a diverse set of links for delivering messages between nodes, without requiring linear routing state at each node. Bati uses available weak network links for opportunistic transfer of routing information, and good connectivity for route shortcuts and last-mile delivery of messages.

Delay tolerant networks provide an in-network store and forward architecture to identify the shortest path between two nodes in a poorly connected network. The key in accomplishing this goal is finding what nodes to forward messages to, and what link to use. Initial approaches used manually filled routing tables with default links for DTN routers [37]. Other iterations use centralized registers for locating nodes [111], or oracles that can answer system level questions [59], such as the average wait time for a periodic link. Using this information, they assign costs to the links available between nodes, and run a modified version of Dijkstra's shortest path algorithm to determine next hops. While it is unclear how to provide these times under natural mobility, perhaps a more significant difference is that these approaches often assume the endpoints of links are known, and it is only a matter of time until a link becomes available. In Bati's target domain, mobility is unstructured, with uncertain endpoints.

Further improvements in DTN routing looked at reducing the reliance on oracles [64]. The Minimum Estimated Expected Delay (MEED) algorithm improves on the Minimum Expected Delay (MED) algorithm that required on an oracle for discovering the expected delay of links between nodes. MEED estimates this value based on the history of the link. However, this still assumes that link destinations are known ahead of time. As a result, the kinds of mobility most utilized in these systems are fixed schedule principals such as buses [48, 111]. Bati focuses on uncertainty in mobility along with diversity in network links for delivering bulk data. Furthermore, since MEED needs to know the topology of the network, given by the estimated delay of every link between every pair of nodes, it does epidemic routing of link state messages. This might be unduly expensive in low-bandwidth, challenged networks.

DTLSR, a delay tolerant routing protocol for developing regions [27], focuses on network-based routing of packets in challenged networks. The main insight in this work is that there is an underlying stability in the network topology in developing regions, and it can be exploited for making routing decisions. Considering this, they suggest including even failed network links in their MEED style shortest path calculation with a cost proportional to how long a link has failed. DTLSR is a network-only protocol, ignoring principal mobility, and requires link-state announcements.

The work in this dissertation improves on these systems by integrating the nuanced use of a diverse set of network links with unstructured natural mobility, constructing a hybrid overlay network that can be used in two complementary ways. It can deliver bulk data between peers within challenged environments, or to and from well connected ends. The system on an unmodified network stack, and probabilistically learns principal mobility in the system.

Finally, when considering services designed for challenged networks, the dissertation discusses the marketplace requirements in making these systems viable and sustainable in the long run. The auction based market platform discussed in this dissertation attempts to foster competition and innovation in these regions by providing a market platform which transfer one-to-one business transactions to a many-to-many environment. As a result, services can easily be discovered and traded on an open market.

Ethiopia's recent commodities exchange [25], setup by a collaboration of the country's economists and engineers, is a good example of market-based solution that is meant to address the unfairness and lack of information for farmers. Esoco/TradeNet [121], a project in West Africa, enables individuals to get current prices from commodity markets using an SMS platform. Trade at Hand [122] is a project funded by the UN's International Trade

Center in Geneva, and provides daily price information for fruit and vegetable exports in Burkina Faso and Mali. Manobi [83] is another telecom firm based in Senegal that provides real-time agricultural and fish prices to fee-paying customers. BoonaNet [66] is a similar effort in Ethiopia for providing pricing information for commodities like coffee and teff using SMS in local languages. These and similar project are good examples of what open markets and information flow can bring to communities. The marketplace in this dissertation further extends these principles by providing an extensible auction-based plugin market platform that can be used for enabling digital services reach more customers, and build standalone markets for trading other goods.

In summary, the contributions in this dissertation build on numerous projects in computer science, information studies and economics in understanding and improving end user experience in information access. The following chapters, presented in three parts, will provide a detailed look into these contributions and how they relate to the projects described in this chapter.

PART ONE

UNDERSTANDING USAGE

CHAPTER III

Observing web usage at shared access sites

3.1 Introduction

Web usage is growing largely personal. The most popular internet destinations increasingly seem to be those that provide individualized experiences to their users. On the other hand, even traditionally ‘static’ content such as news is increasingly localized and personalized. As the amount of information available on the web grows exponentially [11, 42], personalization is a welcome trend that allows users to focus on what is relevant to them. Just as important, the sheer volume of content available online enables users to choose from a diverse set of services that cater to their personal preferences and interests.

Unfortunately, this creates challenges in designing and building web acceleration mechanisms. These mechanisms are especially important in many developing country environments where connectivity is poor and the network infrastructure is generally challenged. Even when good connectivity is available, it’s often very expensive, and comes with costly bandwidth caps [35, 84]. As a result, effective web acceleration systems that work well within these constraints are still very useful. However, many existing systems are built on basic mechanisms that predate the increasingly personal nature of web usage.

As a result, traditional approaches towards web acceleration find it difficult to perform well. For example, recent studies of web acceleration in developing country contexts have reported that overall cache hit rate from prefetching content for users is very low, often given in single digit percentage points [19, 57]. When considering the cost of prefetching unused content in already-constrained environments, those gains dwindle to almost none. This is perhaps to be expected as many systems are built with an aggregate view of their users, while web experience has been getting more personal. This lack of a nuanced understanding of web usage in developing countries limits the effectiveness of many existing systems.

In order to start tackling this growing problem, this chapter presents three concrete steps. The first contribution is collecting the first personalized, large scale web usage dataset in developing countries. While several datasets with aggregate information about web usage patterns in developing countries exist [19, 32, 55, 63], none of them provides an individualized look into personal web usage. The dataset in this chapter was collected at two sites in southern India over a period of one month, and contains web usage information for about 470 users segmented across several sessions. This data has been anonymized to remove personally identifying information, and is available for researchers to access. This dataset will provide a much needed insight for web system developers targeting developing countries.

This chapter also provides some analysis on this individualized dataset in order to quantify the personal nature of web access in developing regions. While there has always been anecdotal evidence to suggest web usage is getting increasingly personal and dynamic even in developing regions, the analysis provides some tangible evidence to describe the phenomenon. For example, a majority of the users spent more than 40% of their browsing time on private web destinations such as email and social networking sites. Likewise, nearly 60% of data transferred over the month was initiated from domains that were requested by less than 3% of the user base. On the other hand, there is a high similarity between different sessions of the same user, even more than what has been reported for users in well connected environments [75].

Finally, using lessons learned from the analysis, this chapter provides some observations on how to design better web acceleration mechanisms for developing regions. The personal and dynamic nature of web usage presents both a challenge and an opportunity in building web acceleration systems. The challenge is rethinking traditional mechanisms with an aggregate view of the user base to cope with emergent trends in web usage. However, this presents an opportunity in designing personalized web acceleration mechanism that understand and cater to users in developing countries.

3.2 Data collection

The collection of personal data usage must be approached with a great deal of sensitivity. The goal was to collect web usage logs from a diverse set of users, while having some level of repetition among those users. In particular, it would be useful to attribute each browsing session to a unique user, and do so consistently across sessions. While its often easier to access such data in organizations that require their users to log in for web access, that also limits the cross-section of users captured. Instead, this study focused on shared access sites that serve a variety of users. There were two sites for data collection—an internet kiosk with 15 Windows workstations, and a vocational computer education center that trains users across several levels. Both of these sites were located in the northern edge of Bangalore in Karnataka, India.

Recent changes in Indian cyber security laws require internet cafe operators to record the identity of their users before every session. Users are required to provide a multitude of information, including their voter ID and telephone number before accessing the web. This law has also given rise to an ecosystem of kiosk management software with personalized information control. However, the internet cafe in this study was still using a pen and paper form to log users (Figure 3.1). As a result, a separate mechanism for associating identity with browsing sessions was provided.

Figure 3.1: A user signs in to access the web at a participating shared internet access site

The Event Logger for Firefox [20] plugin was modified to support a login mechanism that asked users for two pieces of credentials, their email and phone number, at startup. The logger collects various pieces of information about web usage, including requests, responses, caching and inactivity. Every event recorded was associated with the individual information which was used to establish identity across sessions. Further modifications were made to include information such as the time users spent on various domains and the site of data collection. This data was periodically uploaded to a monitoring and back up server. At the end of one month, there were about 8 million events recorded from around 470 users.

The second step in the data collection was anonymizing the dataset to remove all personally identifying information. First, users are grouped based on matching credentials, and each user is given a globally unique ID. Each session for a user is identified with another globally unique ID, and every event in the session is associated with it. In order to remove information that could potentially identify users from URL parameters, all parameters are hashed in the URL entry for requests and responses. This anonymized dataset is available as hierarchical record of user-session events.

Understanding the personal nature of web access in developing countries is one obvious use for the dataset. However, several other use cases might be interesting for researchers. For example, it can be used to simulate a realistic service load when evaluating alternatives for web access improvement in developing countries. In addition, it can enable researchers to build and test fine grained access models that represent web usage in these environments. These models are useful in making resource allocation decisions.

3.3 Analysis

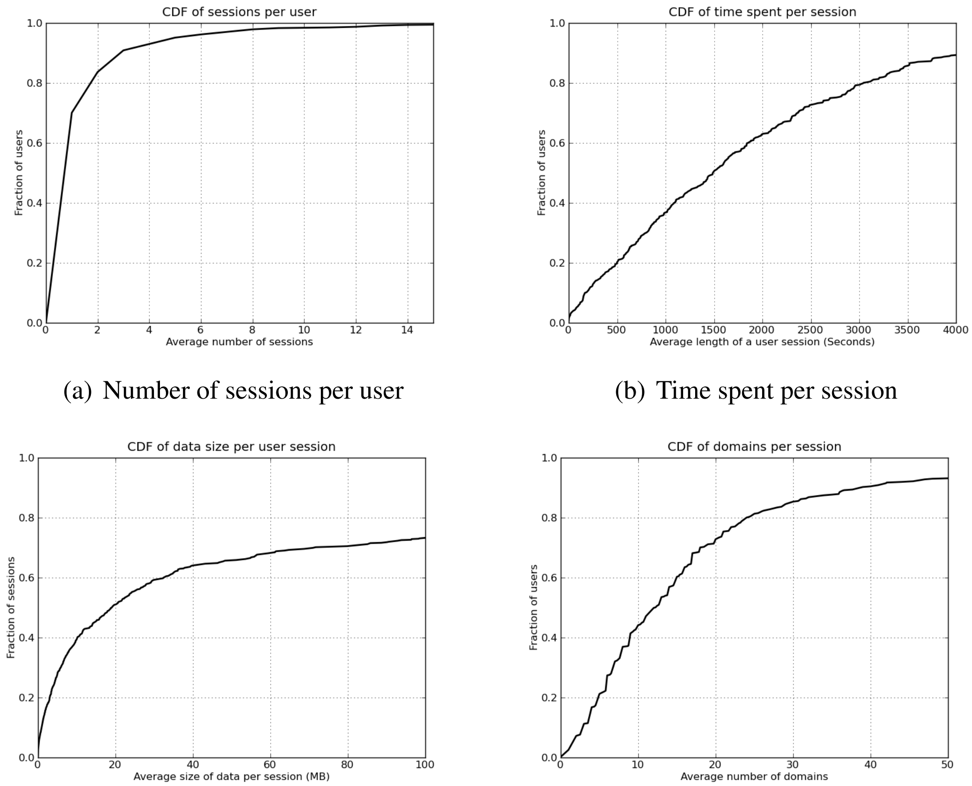
This section provides the analysis on the personalized dataset. It will start by characterizing the data itself, and then proceed to higher level analysis of user behavior. In this chapter, the focus will largely be on personal patterns in the dataset rather than aggregate analysis.

3.3.1 Data characterization

The usage data was collected for a continuous period of 4 weeks between January and February of 2011. It contains a total of 471 users, with 141 of those users having two or more sessions. Figure 3.2(a) shows the distribution of the number of sessions. For about 43 users, there were at least 4 separate sessions recorded.

Each session is characterized by the amount of time a user spent on it. Figure 3.2(b) shows that a majority of the sessions were less than half an hour in length. However, its common to have sessions that are a few hours in length, with 15% of user sessions lasting for at least an hour.

Figures 3.2(c) and 3.2(d) give CDF representations for the amount of data transferred in a session and the number of unique domains that initiated the data transfer. More than 60% of user sessions had less than 15 unique domains responsible for sending data, and



(a) Number of sessions per user

(b) Time spent per session

(c) Data transferred per session

(d) Unique domains requests per session

Figure 3.2: A detailed characterization of the network access data collected during the experiment

transferred less than 30 MBs of data. The size of data transferred varies greatly from one session to other, partly based on the length of the session.

3.3.2 Time spent on private content

The analysis starts out by considering the amount of time users spent on private content. This includes web destinations that require users to log in, such as email and social networking. A very conservative approach was used for identifying private web destinations by looking at only well known email and social networking service providers in India. Then, the time records were filtered for visits to those domains. Realistically, anything that would require credentials to log in, ranging from financial services to job boards, is private by nature. As a result, this representation underrepresents time spent on private content.

Nonetheless, it is found that 40% of the users spent at least 60% of their time on private content, while over 30% spent at least 80% of their time on those services. The complementary CDF in Figure 3.3 shows, what fraction users spent *at least* what fraction of their time on private content.

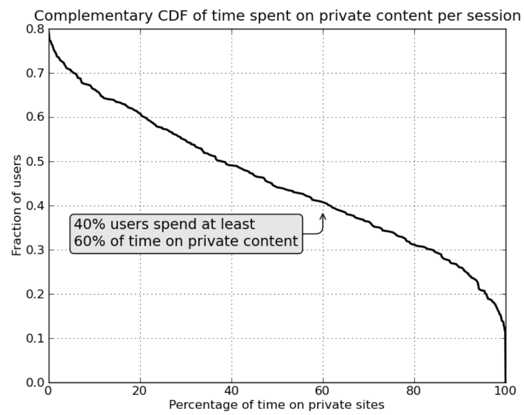


Figure 3.3: The distribution of time spent on private content as a fraction of session length for participating users

3.3.3 Personally interesting content

Private content, however, is only a small manifestation of the personal nature of web usage. The wide availability of diverse content on the web significantly fragments web usage patterns across users—several destinations are only personally interesting, with a small fraction of users requesting them. The analysis considers this phenomena by analyzing how frequently users make requests to various domains. The complementary CDF in Figure 3.4 plots percentage of users making a request against the fraction of requests. For example, the percentage of requests that were made by at least 10% of the users accounted for less than 2% of the total requests. This severe fragmentation has direct consequences on caching and prefetching systems that only consider users in aggregate. As shown in the figure, this is even more visible when each request is weighed with the amount of data transfer it initiated.

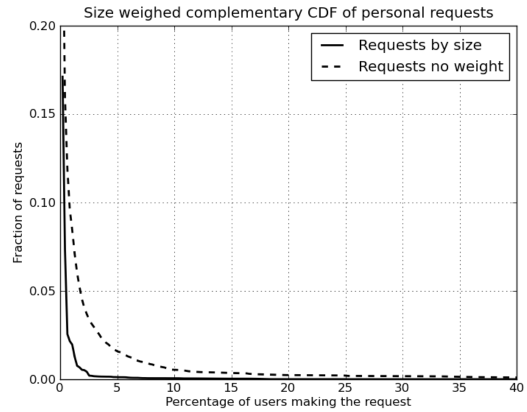


Figure 3.4: The size and relative popularity of requests made by users

Figure 3.5 shows at a 3-way representation of popularity of domains against data size initiated by the domain, and the time users spent on it. Save for a few and well known outliers, most data requests are made by a very small fraction of users. Destinations such as Google and Facebook are requested by a large percentage of people, but account for a small fraction of the total data size, while video content from YouTube accounts for 20% of the total bytes but represents a small fraction of users.

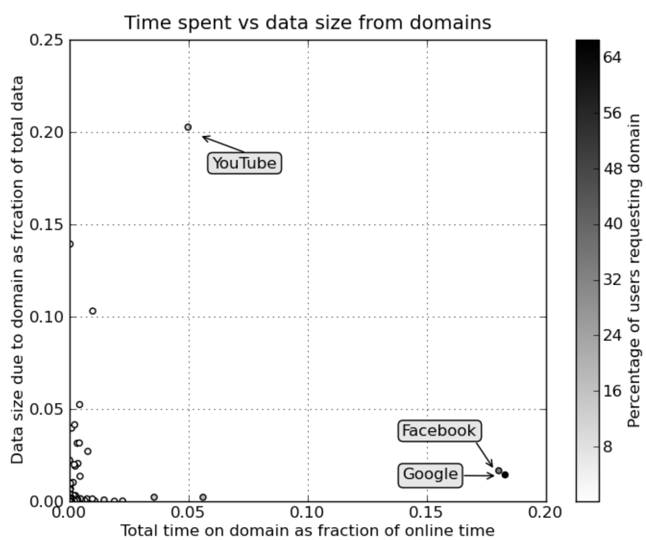


Figure 3.5: A three dimensional comparison of request popularity against the size of requests and the amount of time users spent

3.3.4 Understanding individuals

Since web access in developing countries is increasingly personal and fragmented, the next task was analyzing how similar users are to themselves across several sessions. In addition, it would be useful to compare these results with previous studies of user behavior in well connected environments. The first analysis looks at pair wise Jaccard indices among domain requests per session. The Jaccard index for two sets gives the ratio of the number of items in the intersection of the sets to the number of items in their union. Therefore, for two disjoint sets, the Jaccard index will be 0, while for two identical sets, the Jaccard index is 1. Jaccard indices for user sessions are interesting because they measure how similar sessions are to each other.

The analysis starts out by filtering users that had at least 2 sessions in the dataset, which includes 141 individuals. For each user, the average of the pair wise Jaccard indexes among their sessions. The higher the Jaccard index, the more similar sessions are to each other. Figure 3.6 shows a complementary CDF of Jaccard indexes for two parameters— one weighed by the size of data domains generated, and the other by the amount of time users spent on each domain. These metrics roughly correlate with each other, and 40% of the users had a Jaccard index of at least 0.5. This is significant because it indicates the potential of a personalized web acceleration system to model and understand its users, improving its performance.

Another model that has been commonly applied in studying web access patterns is a Markov model. A first order Markov model assumes the probability of a user transitioning to a particular state depends only on the user's last state, and this probability is the same anytime this state is observed. In the case of web access patterns, a Markov model tries to predict the user's next request based on the last request. This uses slight modification of a first order Markov model that continuously updates the probability distribution of states as more information is obtained about the user. Given a user session with an ordered list of requests, the model predicts the top 3 next requests. Afterwards, the model is provided the

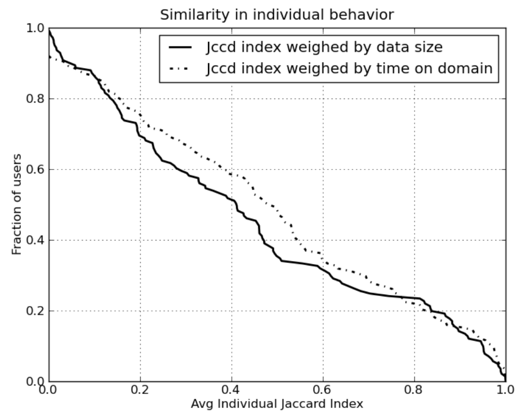


Figure 3.6: The distribution of the average pair-wise Jaccard indexes among sessions of the same user

actual next request, and this is used to slightly modify its probability distribution.

The results in Figure 3.7 show significant success for a Markov models to predict user behavior. On average, the model was able to predict the next request accurately 53% of the time, and the request was in the top 2 and top 3 of the predictions 68% and 75% of the time respectively. This is found to be better than previous studies of web access in developed regions [75], where prediction rates were generally under 40%. However, this is not surprising. Constraints in network access encourage people to mostly focus on important items, and spend their online time on these items. As this reduces the branching factor of browsing, it helps the model predict requests better. Once again, this presents an opportunity for building personalized web acceleration systems that better understand user behavior in challenged network environments.

3.3.5 Associations and clustering

Finally, clustering and association algorithms were run on the global data set to see if they can identify trends that can be exploited in web acceleration. An unsupervised learning algorithm (leader clustering [50]) was applied on a vector representation of the data, where each user had an entry for every domain that is accessed, weighed by the number of times

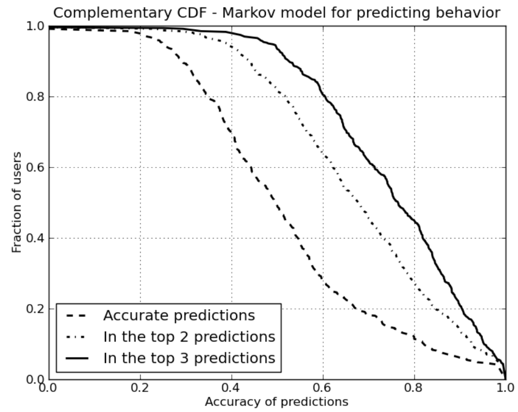


Figure 3.7: The accuracy of a first order Markov model that predicts future access for individuals

it was accessed. Using an Euclidian metric to calculate distance, this attempts to discover clusters of sessions that show similar interests. However, unlike early studies in web access patterns [133], the approach was able to cluster only a small fraction of the total sessions, with a significant majority of clusters including only a single session.

The standard APRIORI algorithm [7] was then run on the dataset to discover association rules between accesses. For example, if users who access site A usually access site B as well, a prefetcher might use this information for intelligently deciding what content to prefetch. While there are some clear associations with high confidence, these tend to be limited to very popular destinations (such as facebook.com ⇒ google.com), or private content (gmail.com ⇒ facebook.com). The results from associations and clustering generally point to the globally divergent nature of access behavior. This is to be expected from the increasingly personal web.

3.4 Discussion

With some of the lessons learned from analyzing a personalized web access dataset, this chapter returns to the design of web acceleration systems for developing regions. When usage is fragmented and personalized, traditional and aggregate mechanisms such as generic caching and prefetching will increasingly find it difficult to perform well. If the last few years are any example, this trend towards more individualized and diversified experience for web users is going to continue. As a result, web acceleration mechanisms need also to be built with personalization as an important component.

While building personalized web acceleration mechanisms will involve various trade-offs with storage, computation and privacy, the potential for improving the end user experience is high. System designers can incorporate personalization in web acceleration mechanisms at different levels and in various capacities. One approach is to personalize prefetching. Predictive prefetching algorithms are an important component of web acceleration [20, 58]. Prefetching, however, has an important constraint in developing regions that the cost of mispredictions is quite high—limited and expensive bandwidth makes it difficult to justify meager gains from aggressive prefetching that consumes a lot of resources. Personalized web usage makes it especially hard to have good, general purpose prefetching algorithms that work well for all users.

Fortunately, however, at the individual level, web usage is not as diverse as it seems. As some of the analysis in this chapter has shown, this is even more so in developing countries. This suggests an important consideration for web acceleration mechanisms should be incorporating user identity in determining what to prefetch. Such a mechanism can be implemented at a client or a proxy level depending on the private or shared nature of access. Identity does not have to be personally identifiable, and a personal prefetcher will have responsibility in providing some privacy guarantees to its users. The dissertation will later discuss how to incorporate some of these observations in building a system for distributing private data in challenged networks in Chapter V.

Another dimension for incorporating personalization could be to build cloud-based tools that guide local clients through web acceleration. This might turn out to be more convenient as it does not tie users to particular access sites. This also gives designers more flexibility to incorporate large scale data in improving individual performance. For example, as a user starts a browsing session, she might be offered to sign in to her account. Without a local administrator having to keep track of individual behavior, the personalization service can provide the required information to the browser that would help it improve web performance to the current user. At the same time, this allows the user to get improved performance across various service providers and platforms.

In summary, this chapter described a study in understanding web usage behavior in developing regions settings, and its implications web acceleration mechanisms. Although web acceleration mechanisms are important in challenged network environments, new approaches are necessary in coping with the changing patterns in data access. In particular, the personalization of web access, and the resulting fragmentation, is often detrimental to mechanisms that rely on the redundancy of content access for improving performance.

As a result, new systems built for web acceleration and improving end user experience in information access need to consider personalization as a key factor. The second part of this dissertation will discuss several such systems that build on these observations. The next focus in understanding user behavior is a case study on the adoption and characteristics of social networking in developing regions. Social networking has been one of the fastest growing segments in web access, and one that is driving personalization. The study presented in the next chapter is based on data obtained from a commercial social networking site with a strong presence in developing regions.

CHAPTER IV

Case study: social networking in developing regions

4.1 Introduction

The analysis of web usage in developing countries, discussed in the previous chapter, points to the increasing fragmentation and personalization of access behavior. One of the key areas where this has been manifested is in online social networks. As a case study in further understanding web usage behavior in developing regions, this chapter focuses on the characteristics and adoption of social networking in these areas. In doing so, we analyze user behavior as well as socio-economic factors that might impact such behavior, ranging from demographics and education to the effects of geographic location in social network adoption.

As mobile and internet penetration improve worldwide [1], users from developing countries are participating in increasing numbers in online communities. The previous chapter and several other studies of web usage patterns in developing country contexts [84] indicate internet users in these areas are engaged in online social networking and communication tools, spending a significant portion of their online time on them. These observations have been made across several usage scenarios, ranging from educational institutions in urban India to remote internet access sites in Africa and Latin America. In addition, surveys and other anecdotal evidence has indicated that users from emerging economies have been driving worldwide membership growth in social networks [18].

This chapter provides the first large scale and detailed analysis of social networking usage in developing country contexts. The analysis is based on profile and activity data from LinkedIn, a professional social networking site with, as of writing, over 100 million members worldwide. Over the past few years, online social networking has been providing a communication platform on a truly global scale unlike anything the world has seen before. Its adoption by people from every corner of the world builds on many interesting patterns and characteristics that reflect the underlying economic, social and cultural makeup of the participants. Using data from a commercial social networking site with a global membership base, this chapter provides an internal look at the social networking phenomena in developing country contexts, and how it compares with the rest of the world.

LinkedIn has members from every country in the world, including several million in Africa. It also has a strong presence in Asia and Latin America, with countries like India and Brazil among the most active in the world. As a professional networking site, LinkedIn also has unique access to information such as career industries and educational level of members. This gives us a rich set of demographic and location data to work with, augmented by detailed activity information for the website. This ranges from how members access the social networking service to how they make connections and interact with other members. This study combines profile information with activity data to analyze several aspects of social networking usage in developing countries.

The analysis in this chapter is presented in the form of several themes that illustrate different dimensions of social networking use. While the study focuses on patterns and characteristics from developing countries, it will also present contextual information from the rest of the world, which provides interesting comparisons emerging from the underlying differences and similarities in the member base. Some patterns are unique to the developing world, often shaped by economic, social and cultural factors, or the brief history and attributes of internet citizenship for many users in these environments. Other patterns transcend geographic and economic barriers, and derive from basic human social behavior in sharing, communication and interaction. The goal of this chapter is to provide researchers a revealing look on the growth, adoption and characteristics of social networking in developing regions, one of the fast growing segments in internet use in those regions. While these characteristics are good indicators of social networking use in emerging economies, it is important to note that the analysis is solely based on data from LinkedIn, only one of several commercial social networks.

This chapter discusses six characteristics and patters, ranging from the interconnectedness of members in various geographic regions to the demographic and educational makeup of participants. Social networks enable members to make connections with other members throughout the world, and the study will begin by investigating how people choose to connect with each other. In particular, the study will look at the geographic locality of social network connections, and trends that emerge from cross-country and cross-continental connections.

The study then considers the overall engagement and activity of members in developing countries, and how it compares with the rest of the world. This can be expressed in several terms, including the growth of personal connections of members in the network, and the frequency and length of visits compared to members from more connected environments. This is augmented by a look at access devices from developing regions. The study will investigate how members are accessing social networking sites from various regions, and how this trend maps the increase in mobile and internet penetration in many developing countries.

Another important component in understanding social networking use is the demographic and educational makeup of members. The study will explore generational bias in internet access and participation, as demonstrated in the age distribution of members across the world. In addition, it looks at gender representation, and how cultural and access barriers are reflected in social networking usage. Alongside demographics, the study investigates the educational background and industry representation of members from various developing countries and corresponding worldwide trends, discussing biases in membership due to economic status and access to technology.

Finally, the study looks at the impact of local languages in social networking participation. It will investigate how content access in a local language influences adoption, and the extent of this influence in various regions. As new languages are introduced, it analyze how it affects usage in developing countries. To show this effect, the study considers pairs of countries with the same national language, but different economic and cultural backgrounds, and explore the correlation between local languages and adoption.

4.2 Dataset

This section describes the process of data collection and data analysis used throughout the chapter. This section will first discuss the pipeline of data collection on the LinkedIn platform, and how data is aggregated from several sources. It will then introduce the data infrastructure used for the analysis, which also supports many LinkedIn services.

4.2.1 Data collection

The study uses two main types of data at LinkedIn. The first form is replicated from production databases, which consists of data mostly provided by LinkedIn members. This data includes member profile information, their education, and their connections with other members.

The second form is activity-based tracking data, which corresponds to logins, pageviews, and user agents. This data is aggregated from production services using Kafka [70], a publish-subscribe system for event collection and dissemination developed at LinkedIn. As of this writing, Kafka is aggregating hundreds of gigabytes of data and more than a billion messages per day from LinkedIn's production systems.

The data analysis presented in this chapter is done over a large amount of data, and is intended to represent interesting characteristics of social networking use in developing

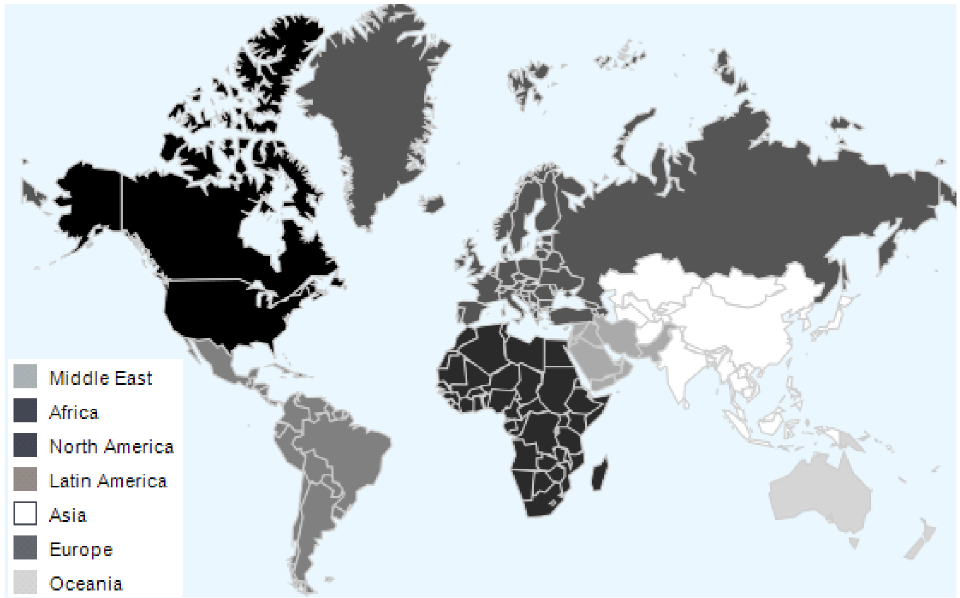


Figure 4.1: Classifying regions of the world for this study

countries. In order to classify countries into groups, the study uses the UN geoscheme for macro geographical regions from the United Nations Statistical Division [3]. Member country information is provided during registration.

This scheme is based on the M49 classification, and is often used for statistical analysis purposes. To better represent socioeconomic differences, the study makes two common adjustments. First, it groups Central America, the Caribbean and South America into "Latin America and the Caribbean" (or "Latin America" for short). In addition, it extracts Western Asia (the Middle East) into a group of its own. While the study has represented Africa, Asia and Latin America separately in the analysis, it refers to their combination as the developing world, and compare their statistics with North America and Europe. Figure 4.1 shows a map representation of this regional classification.

When this chapter represents countries on figures, it sometimes uses the ISO 3166 two letter country code for graphics readability. Further, all chosen countries have at least 50,000 LinkedIn members so as to avoid skew due to sparsity.

4.2.2 Analysis Infrastructure

One of the core pieces of data analysis infrastructure at LinkedIn is Hadoop, an open source implementation of MapReduce [26]. MapReduce provides a framework for processing big datasets on a large number of commodity computers through a series of steps that partition and assemble data in a highly parallel fashion, simplifying the process of writing parallel programs by providing the underlying infrastructure, failure handling, and simple interfaces for programmers.

At LinkedIn, and for the analyses in this chapter, MapReduce is used with two scripting languages on top of Hadoop: Pig [94], a high level data flow language, and Hive [119], a SQL-like language. After aggregations are computed on Hadoop, the resulting data is small enough to be processed by common tools locally on a single machine.

4.3 Data Analysis

This section discusses six themes in understanding social networking usage in developing country contexts. As this work is a comparison of the developing world against the developed world, it normalizes all data to the United States or North America respectively. In a couple of cases, the analysis has had to estimate data, which has been clearly documented.

4.3.1 Connections

The first topic focuses on the composition of connections in the social network for members from various regions. A connection is established when a member requests an invitation with another member in the network and is later approved by the invitee. Online social networks enable participants to establish connections with members from around the world, and this section investigates the makeup of these connections. Note that connections are bidirectional, with each connection linking two members in both directions.

An interesting pattern in analyzing social network connections is the interconnectedness of members within a region, or the locality of relationships. This study expresses the geographic locality of relationships by computing the ratio of connections that are established to members within the same geographic region. When considering macro-geographic classes, the study measures the fraction of connections established within the same macro-geographic region. It then breaks the numbers down by country, and considers connections within the same country.

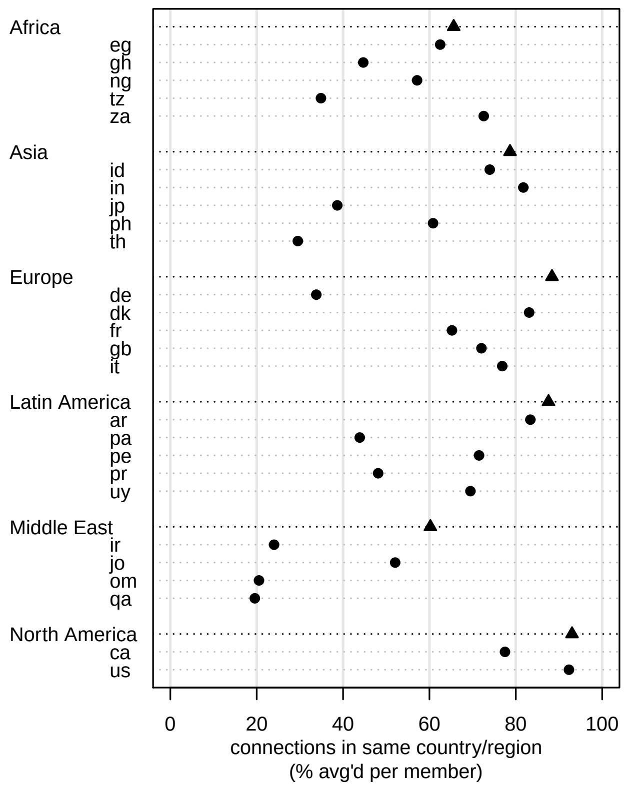


Figure 4.2: Geographic interconnectedness—fraction of connections originating from the same geographic region as members in that region

Figure 4.2 shows the interconnectedness of various regions. For each macro-geographic region represented, it also provides a few selected countries from the same region, and consider connection locality at the country level. Africa and the Middle East have two of the lowest rates of geographic locality: nearly 40% of connections in each respective region is established with members outside the region. Figure 4.3 shows geographic locality of connections for all countries in Europe and Africa on a map. As the dots on each country get larger and darker, connection locality increases.

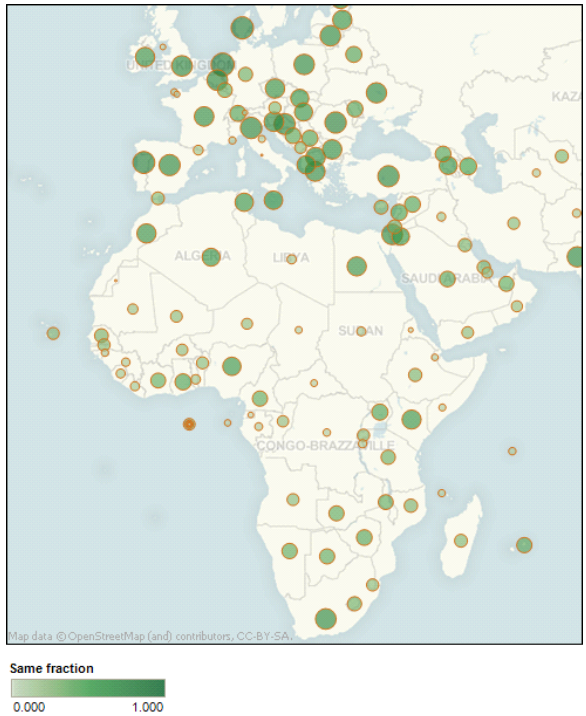
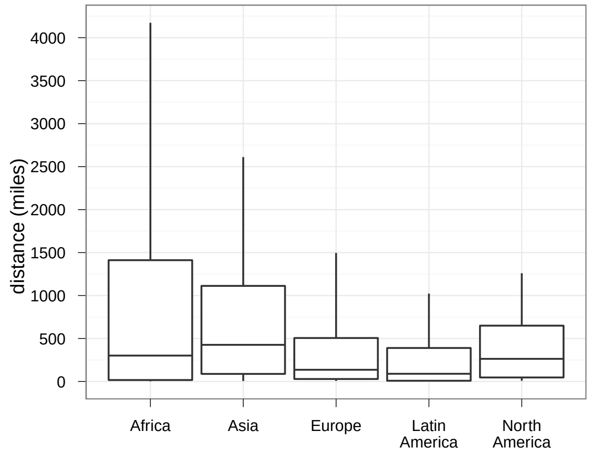


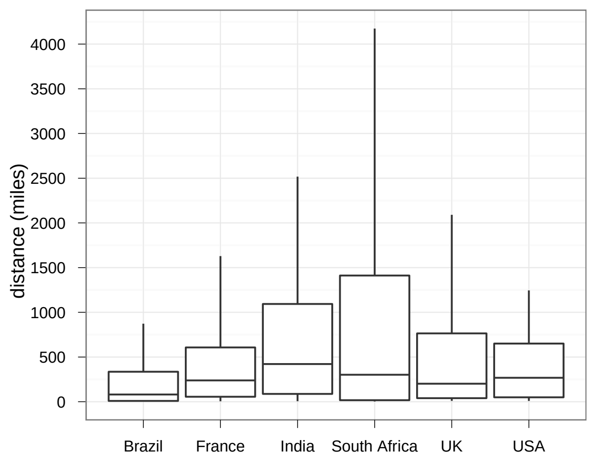
Figure 4.3: Geographic interconnectedness—locality of connections for members in Africa and Europe, computed per country. Bigger dots represent more connections established to members within the same country.

One of the important factors in understanding connection locality is the membership population from each region. Intuitively, as the number of members in a region increases, the chances of establishing a relationship with similarly located members increases. However, this can be balanced out by the increase in membership of other regions, which also provides more opportunities for cross-country and cross-region relationships. There is some correlation between the size of the membership base in a country and the rate of connection locality ($\rho \approx 0.6$, for countries with more than 100,000 members).

Another interesting way of looking at connection distributions is to consider how far connected members are from each other. Using location information, the distance between members is estimated using the Haversine function [114]. Figures 4.4(a) and 4.4(b) show connection distance for macro-geographic regions and a selection of countries. For each



(a) Regions



(b) Selected Countries

Figure 4.4: Connection distances—distribution of estimated physical distances (in miles) of network connections for members from various regions

region, the distance distribution is computed by considering all connections that originate from the region. On average, Africa and Asia have the two longest distances for connections, and this is also reflected in the individual countries represented. Several reasons, including geographic attributes of the region and the rate of connection locality, affect this distribution.

For connections that do not terminate in the same geographic region, the study then considers patterns in cross-country and cross-continental relationships. Figure 4.5 is a branching map which depicts where outbound connections terminate for a few selected countries. For each country, it shows a few countries where members in the originating country have connections to. The thickness of the line for each arrow corresponds with the fractions of connections that terminate in the destination country. To avoid cluttering, the figure removes self loops, and instead provide the fraction of local connections as a percentage.

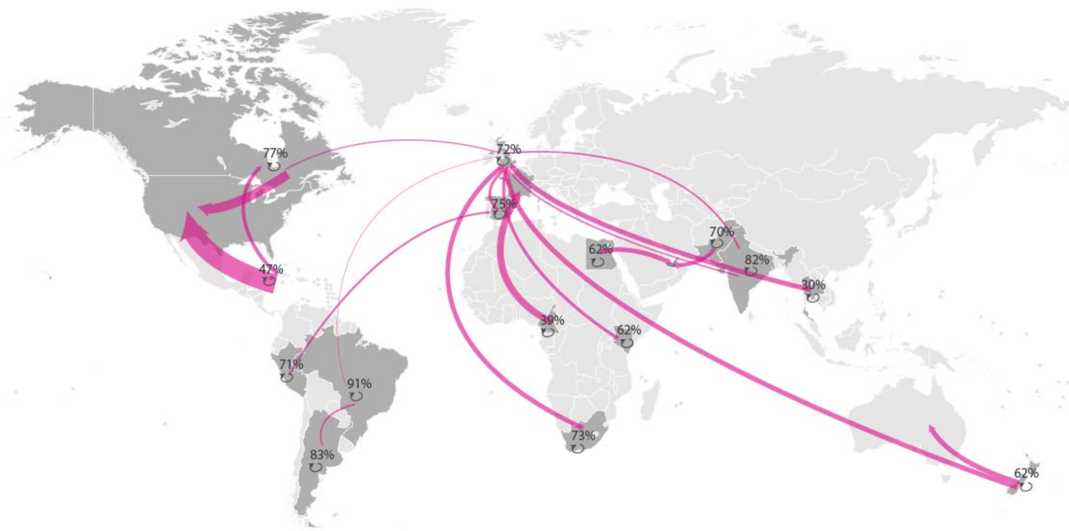
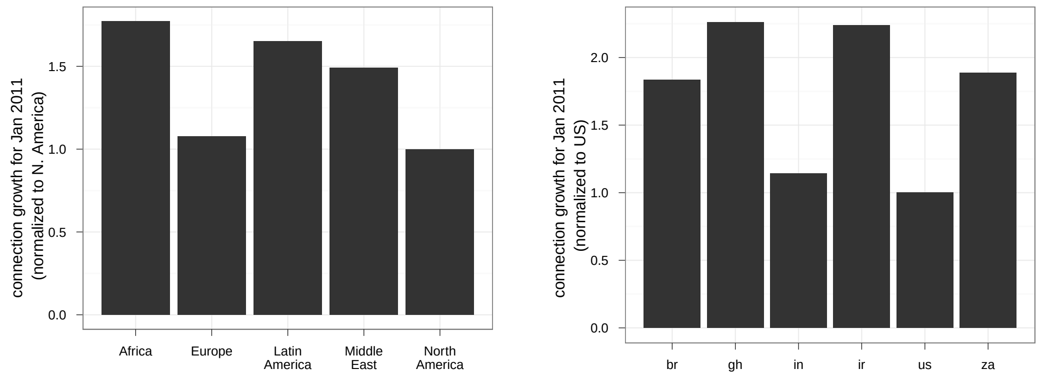


Figure 4.5: Outbound connections—a representation of cross-country social network connections for various countries

4.3.2 Activity

Measuring the activity and engagement of social networking participants is an important component in understanding usage from various regions. This manifests itself in several ways, ranging from how actively members are making connections on social networking sites, to the duration of visits to social networking sites. Several web usage studies in developing country contexts have indicated that users spend a significant fraction of their online time on social networking and communication websites. This section presents a few metrics to further delineate usage across developing countries.



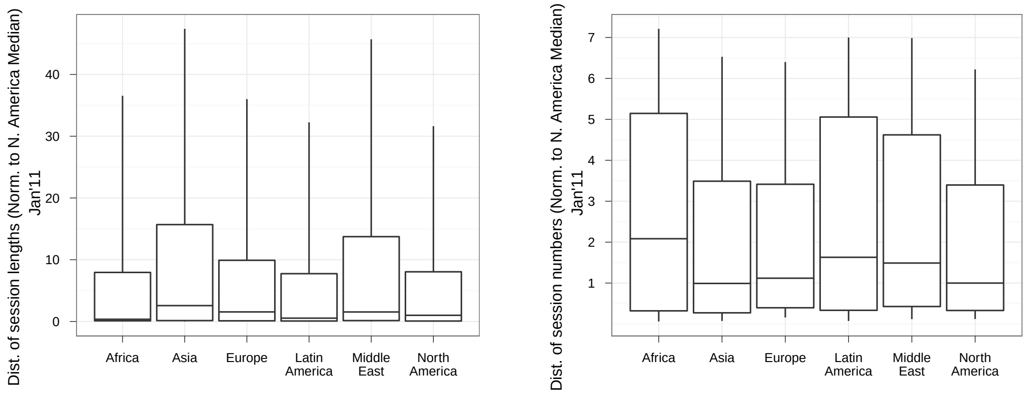
(a) Regions (b) Selected Countries

Figure 4.6: Connection growth—the rate of establishing new connections for members in various regions

An important aspect of social networking activity is the rate of establishing connections. Much of the utility in social networks is driven from communicating with fellow members, and connection growth is a key indicator of member engagement. Figure 4.6(a) presents the normalized rate of connection growth for January 2011 across several regions, which is calculated as the average month over month growth in the number of connections for members from each region. This rate is normalized such that the rate of connection growth for North America is one. Figure 4.6(b) presents the connection growth information for a selection of countries for January 2011, normalized to give the US a value of one.

Connection growth is the fastest in developing regions, which is also reflected in the individual countries represented. One of the main reasons for this is the increasing addition of new members from these regions who are actively making connections on the network. In the early stages of social networking use, members actively add connections to their network. Even then, however, some regions are more active in adding connections. For example, during January 2011, members from Africa are adding new connections quicker than their counterparts in Latin America, although the membership base is growing around 30% faster in the latter.

Another metric this study considers in analyzing activity on the social networking plat-



(a) Normalized session length (b) Normalized visit frequency

Figure 4.7: Member activity—the duration and frequency of member visits from several regions

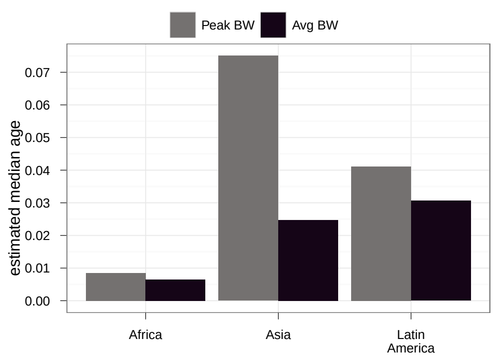
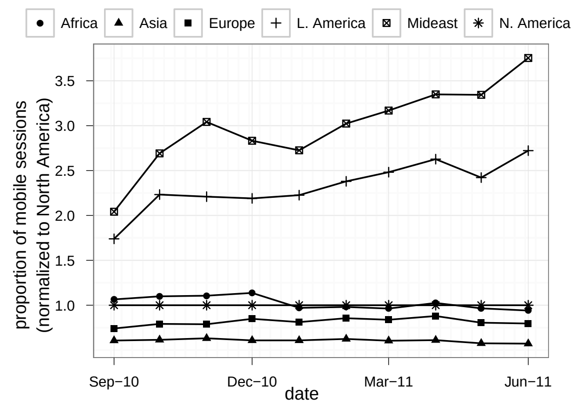


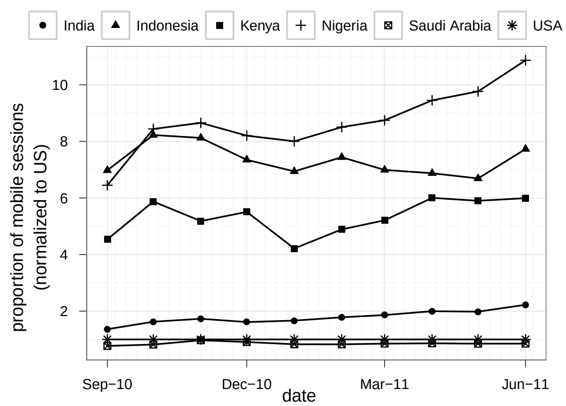
Figure 4.8: Regional bandwidth—peak and average inbound bandwidth from members in developing regions

form is the duration of visits. A session is defined as a continuous user activity with an idle period of at least 30 minutes indicating a new session. Figure 4.7(a) shows the average normalized length of sessions for January 2011. In general, sessions established from developing regions generally last longer. A key factor for these differences is network connectivity, which varies significantly for different regions. For example, low bandwidth connectivity in African and Asian countries requires members from those regions to spend more time interacting to obtain the same information as North American members who have shorter lived sessions.

To better represent this relationship, Figure 4.8 plots the average and peak inbound



(a) Regions



(b) Selected Countries

Figure 4.9: Mobile Access Growth—the ratio of sessions established from mobile devices, each month normalized to the fraction of mobile accesses in the US

bandwidth from accesses in developing regions on July 1, 2011 normalized to North America. These measurements are obtained by a system that monitors LinkedIn's inbound network traffic from several endpoints around the world, in part for detecting and preventing network attacks. These numbers correspond to Akamai's state of the internet report [8] which estimates average connectivity from various regions. As mentioned earlier, the low bandwidth connectivity of members from developing regions is one of the reasons for elongated sessions.

In addition to session duration, visit frequency is an important metric for comparing social networking engagement across developing regions. Figure 4.7(b) plots the normalized, average number of sessions per member for various regions for January 2011. The data is normalized such that members in North America have a median visit frequency of 1. Developing countries generally have a high number of visits per member, although this could be skewed by the influence of newer members. This also correlates with the growth in the number of connections described earlier.

4.3.3 Access devices

Mobile penetration has been one of the singularly most important factors in connecting developing countries locally and across the globe. By the end of 2010, there were more than 5.3 billion telephone subscribers around the world, with nearly a billion of them having access to 3G data services [1]. This growth has been largely driven by Asia and Africa, which have the two highest growth rates in the world. Mobile access is available to nearly 90% of the world population. A number of studies in the developing world indicate that for many people, the phone is the first, and sometimes only, gateway to the internet [31].

In this section, the study focuses on how users access social networking services from various regions. It broadly divides access verticals to mobile and desktop. Mobile accesses include visits that were directly made from mobile web browsers, or through applications that access the social network over a set of API's. In addition to smartphone applications for platforms like the iPhone, Android, and BlackBerry, LinkedIn also has a native Symbian application that runs on Nokia phones, which are more common in developing countries. The fraction of mobile accesses is the metric of interest in this case.

For each region, the ratio of accesses made from mobile devices is computed. To compute access ratios, the study looks at the fraction of sessions that were made from mobile browsers and applications, aggregated by geographic regions. Each ratio is computed as a fraction of accesses from mobile devices in that region to the total accesses from the same region on a monthly basis. Each month has been normalized to the mobile access fraction in North America or the US, respectively.

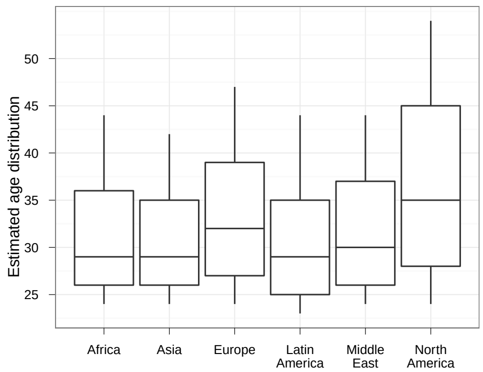
Figure 4.9 shows the ratio of mobile sessions by region and select countries for a period of 10 months, each month normalized to North America or the US, respectively. At a regional level (c.f. Figure 4.9(a)), Latin America and the Middle East have the highest fraction of mobile accesses, and mobile accesses have been increasing significantly. Figure 4.9(b) presents a selection of countries in developing regions with high mobile access ratios. For example, in Africa, Nigeria has some of the highest mobile access rates with nearly 3 times as much as mobile accesses from North America.

4.3.4 Demographics

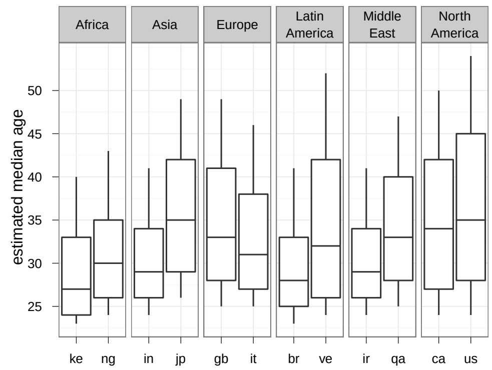
This section focuses on the age and gender composition of social networking participants from developing regions. The age of members is estimated from their profile education information: it is assumeed members were 21 years old when they start their career. While this technique might be a good approximation for members in western countries, a caveat is that its effectiveness might vary in different areas where career start ages might be generally different.

Generational bias is an important factor in online participation. Perhaps more interesting is that this bias tends to operate on a global scale. When looking at the average and median ages of social networking members from across the world, participation is naturally skewed towards people of younger age. The age distribution is also interesting when considering the underlying makeup of the population in different regions. Countries in developing regions generally have a younger population, which is reflected in the social network representation of age groups.

As shown in Figure 4.10(a), the median ages for members from North America are a few years higher than those in Africa or the Middle East. The age distribution for Asia and Latin America is also quite similar, roughly within a year compared to members in Africa. Broadly speaking, younger median ages for members from developing regions correlate with the differences in median ages of the underlying population. Figure 4.10(b) shows a selection of some countries from each region with median ages in the highest or lowest quantiles.



(a) Regions



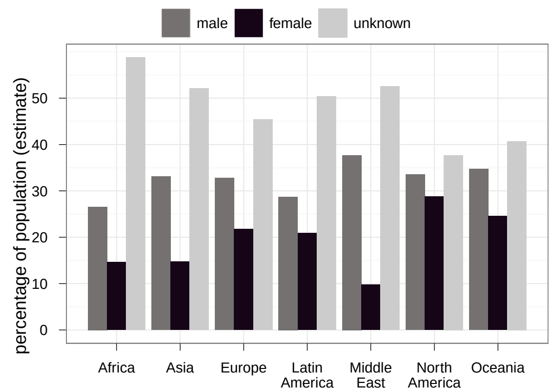
(b) Selected Countries

Figure 4.10: Demographics—estimated median age of members in various regions

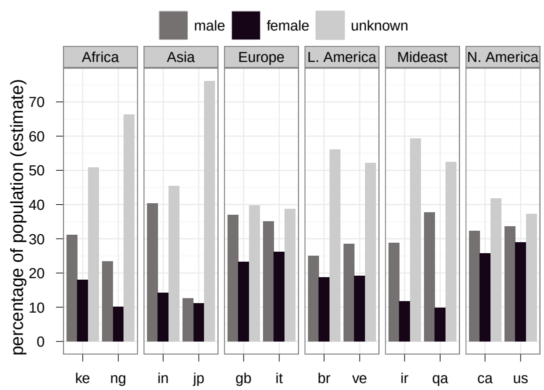
Gender representation is another interesting characteristic for many regions. Gender information is approximated by classifying member first names using a large annotated catalog retrieved from several baby name books. Names that could not be mapped to a gender—the catalog of baby names is biased to Western names—or are ambiguous are labelled “unknown.” In all of the gender information figures, the figures have represented the fraction of users we were not able to map to genders. The unknowns are rather high, so any conclusions should be viewed with some suspicion.

Figure 4.11(a) shows the female membership ratio for a few regions. Globally, males are generally overrepresented by membership, but the differences are more pronounced in many developing countries. Many of these differences can be attributed to social gender roles and economic differences. For example, the Middle East has the lowest female membership ratio in the world, with females making up less than 25% of the total membership base. The ratio is slightly higher for Africa, but with significant differences from country to country.

Figure 4.11(b) shows a selection of countries with various female representations from



(a) Regions



(b) Selected Countries

Figure 4.11: Demographics—approximated gender of members in various regions

several regions. Latin America has one of the highest female ratios in the world, and several Asian countries have gender representation in line with the general population. However, countries like India and Bangladesh have a highly skewed male representation. In Africa, South Africa has one of the higher ratios with nearly 45% female makeup (compared to 49% in the general population [2]). North African countries share similar traits with the Middle East with lower female representation compared to the rest of the continent.

4.3.5 Education and Careers

The fifth theme in the analysis focuses on educational levels and career industries of members. As a professional social network, LinkedIn encourages its members to enter their educational and work history to their profiles. Education levels can include one or more user provided description of the member's educational history. Education levels are described differently across the world. For example a "diploma" in Ethiopia corresponds to a 2 year degree that is equivalent to an associate degree obtained from a community college in the US. To mitigate this problem, education levels are broadly divided to four: high school, college, masters and PhD. When a member has listed more than one education level on their profile, the highest one is picked. Members must also provide an industry when they describe their career. Industry captures a high level classification of career paths, and there are over 120 industries represented on LinkedIn.

The study considers educational levels in several regions in comparison to educational makeup in North America. Figure 4.12 shows the distribution of educational levels in relation to North America or the US respectively for several regions and countries. With the exception of high school graduates, it is interesting to note the near uniform distribution of members with higher educational levels in several regions. Africa and the Middle East have a high fraction of high school graduates in the membership base. When considering Education Indices from the UN Human Development Report [4], it is important to note that professional social networking membership is not representative of the underlying literacy rate and education index in many developing countries. Rather, it is skewed towards to relatively more educated members, which can translates to relative economic affluence, and improved access to connectivity.

Table 4.1 shows the top-5 industries represented from each region. Some differences appear when looking down the list of industries from each region. These differences are more apparent when looking at a selection of countries as shown in table 4.2. As expected, industry representation in a country tends to reflect regionally established industries, such

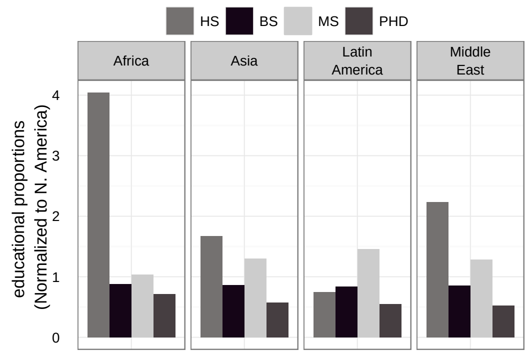


Figure 4.12: Educational levels—the proportion of educational levels making up the membership base in various regions

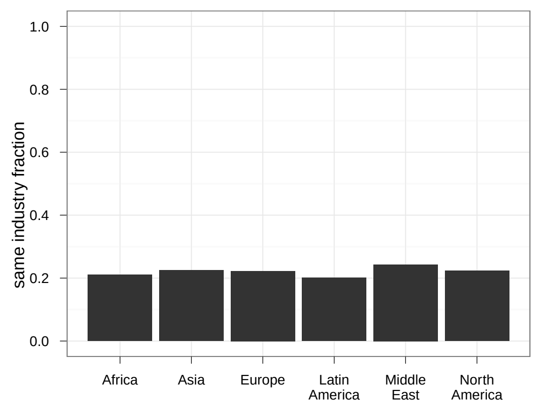


Figure 4.13: Industry interconnectedness—fraction of connections made to members in the same career industry for members in various regions

as oil/energy in Nigeria or computer and software in India. In addition to the volume of professional regionally established industries hire, they also tend to have more international contacts, which has an impact on technology adoption.

Another interesting aspect of represented industries in the network is how members connect across various industries. Figure 4.13 looks at the industry similarity of connections, which is defined in a similar manner as geographic interconnectedness in Section 4.3.1. For each member, the figure computes the fraction of connected members that also work in the same industry as the member. Interestingly, the rate of industry similarity

Region	Top 5 industries
Africa	accounting, banking, education management, information technology and services, telecommunications
Asia	computer software, education management, financial services, information technology and services, telecommunications,
Europe	computer software, financial services, information technology and services, marketing and advertising, telecommunications
Latin America	construction, higher education, information technology and services, marketing and advertising, telecommunications
Middle East	banking, construction, information technology and services, oil and energy, telecommunications
North America	education management, financial services, hospital and health care, information technology and services, real estate

Table 4.1: Top 5 industries by region: ordered alphabetically

Country	Top 5 industries
India	computer software, education management, financial services, information technology and services, telecommunications
Malawi	accounting, banking, education management, information technology and services, non-profit organization management
Nigeria	accounting, banking, information technology and services, oil and energy, telecommunications
Saudi Arabia	construction, hospital and health care, information technology and services, oil and energy, telecommunications
United States	education management, financial services, hospital and health care, information technology and services, real estate

Table 4.2: Top 5 industries by country: ordered alphabetically

remains very close for all the regions considered, with members having only 20–25% of their connections from a similar industry. Intuitively, one might have expected most connections to remain within the same industry, where professional relationships are natural to establish.

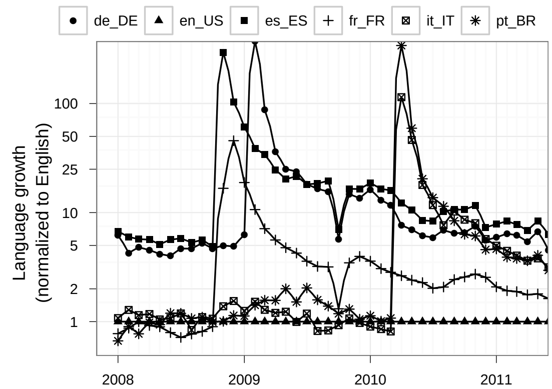


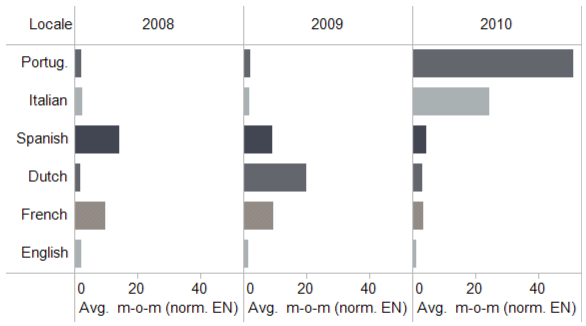
Figure 4.14: Locale growth—month over month growth of languages, with a focus on the impact of newly added languages

4.3.6 Local languages

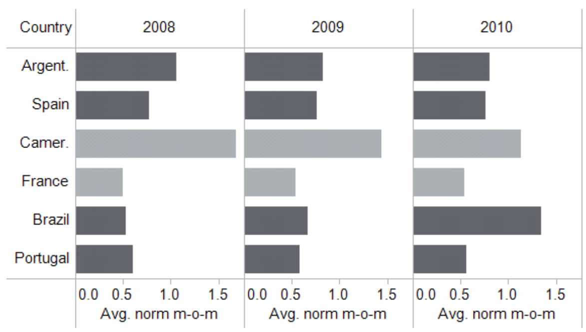
The last topic considered in understanding social networking in developing regions is the impact of local languages on adoption. LinkedIn is available in a multitude of languages, a few of which are local languages for many regions in Africa and Latin America. The analysis focuses on a few chosen languages that have been available to members for at least one year.

This section first looks at the membership growth rate for various languages. Figure 4.14 shows a month-over-month time-line from January 2008 through May 2011 of five languages normalized to English. There is a substantial increase in membership in the language when it is first introduced, which often remains high for about six months before it regresses to the mean.

It is often interesting to see the impact of local languages on particular countries. In order to make some comparisons, this section chooses three pairs of countries from different regions such that the national language for each pair is the same. These include Cameroon and France (French), Argentina and Spain (Spanish) and Brazil and Portugal (Portuguese). The top half of Figure 4.15 shows the average normalized month over month growth of languages for 2008–2010. The bottom half shows the average normalized rate of membership growth for each country, and how the rate changes as languages as introduced. In all of the cases, membership growth responds more positively for countries from developing regions as the national languages are added. Some of this difference can attributed to the overall difference in membership growth across several regions, but the adjustment in the rate of growth a year after the language has been introduced indicates that local languages play an important role in early adoption, and even more so in the developing world.



(a) Locale month over month



(b) Membership month over month

Figure 4.15: Membership with languages—the impact of language introductions on countries with the same national language, but different socioeconomic backgrounds

4.4 Web usage studies

One class of related projects come from web usage studies that provide a macro classification of how users spend time online. There are several web usage studies that have been conducted in developing country settings [20, 32, 84]. Du et. al. [32] evaluated HTTP traffic captured from shared access sites in Ghana and Cambodia. Their results demonstrate several features of web usage in developing countries prior to the widespread adoption of social networks. Another study of internet usage and performance in Zambia [84] points at the increased adoption of social networking and communication tools even in rural villages in Africa. Analysis of web usage in Macha, Zambia, some 350 kilometers from the capital city Lusaka, reveals several interesting findings, including social networking sites as the top visited destinations. More specific web access analysis from a school setting in India [20] also indicate wide usage of email communication and social networking. The work in this chapter complements this body of work by providing analysis on the adoption and usage patterns of social networking in developing regions. As social networking continues to be a dominant web usage scenario around the world, this chapter provides researchers with some insights on the adoption and characteristics of social networking, particularly in developing regions.

A large scale study of web traffic using data collected from a world wide content distribution network (CDN) by Ihm et al. resembles the work in this chapter in the scale of data analysis [55]. Their work analyzes web content that represents one week's worth of browsing data from nearly 350K users across 190 countries. They observe a number of interesting characteristics of web usage in developing regions, including the desire for rich media and differences in download type distributions. In contrast, the work in this chapter focuses social networking usage at a global scale by using data from over a 100 million members that come from every country in the world, with tens of millions of those members from developing regions. This study combines individual profile information with member activity logs for providing researchers a revealing look on some patterns and characteristics of social networking usage in developing regions.

4.5 Discussion

This chapter provided a case study of web usage behavior in developing regions by focusing on online social networks. Using profile and activity data from LinkedIn, this chapter has presented several themes for better understanding various dimensions of social networking in developing regions. Some important observations included the nature of interconnectedness and locality for members and the impact of local languages in social network adoption. Over many metrics, the analysis in this chapter attests to the accelerated growth of social networking in developing regions, and its importance in driving personalization in web access.

As several other studies have also indicated, users in developing regions spend a sizable portion of their time on personal or private destinations, with social networking as a one example. The observations in this part of the dissertation point to at least two important usage scenarios that can benefit from custom designed systems for improving end-user experience as data access grows more personal and media-rich. One is addressing the challenge of distributing private data in challenged networks, while the other is designing mechanisms for efficiently transferring bulk data despite poor network connectivity. The next part of the dissertation will look at systems that were designed with these observations in mind.

PART TWO

IMPROVING DATA TRANSFER

CHAPTER V

Distributing private data in challenged networks

5.1 Introduction

Based on observations of data access behavior in developing regions, we focus on two usage scenarios for improving end-user experience. One is aiding the distribution of private/personal data in challenged networks, and the other is improving bulk data delivery in these environments. This chapter discusses the former.

There have been several proposals for improving information access in challenged networks, ranging from opportunistically prefetching content and storing it for later use [39] to employing peer-to-peer technology to cooperatively download data files and cache them for other users [106]. However, these solutions target data that is requested and consumed by multiple users. Each leverages redundancy in data requests to avoid fetching a piece of data several times, and delivers it from a local cache whenever somebody else requests it, thereby making future requests fast. This is a tried and true approach, used in well connected environments by content distribution networks and most modern browsers. However, this solution cannot help with personal and private data—unfortunately, such data comprises a substantial portion of that consumed in internet kiosks as discussed earlier. In other words, when the data is either personal or only personally interesting, such as email or a school application, a user can not expect other people to have prefetched the data for her. In these cases, users must wait while their data is being fetched from a remote host over a very slow link.

This chapter proposes a solution to this problem by extending the idea of individualized content distribution networks [29] that take personal usage patterns into account. The system, called *Sulula*, provides an infrastructure for distributing private data in challenged networks. Sulula leverages the wide penetration of cellular telephones in developing countries to securely deliver private data to nodes in low bandwidth, high latency environments ahead of usage time. By the end of 2010, there were more than 5.3 billion mobile telephone subscribers around the world, with Africa having the highest mobile growth rate [1]. In Ethiopia, for example, there are about 25X as many mobile subscribers compared to internet subscribers [36]. This allows us to take advantage of widely available services like SMS in the design. To motivate the approach and better explain the solution, consider the following usage scenario.

The user, Elsa, is a bank teller in Addis Ababa, Ethiopia. Elsa spends most of her day at work, and she also takes classes a few nights during the week. She does not have a car and uses public transportation to get to work and back. She has some time during the day when she takes breaks from work, and some more time in the evenings. However, the minibuses she takes home stop running once it gets darker.

Elsa knows of three internet kiosks that she can reasonably get to. One is within walking distance of her work(K_A), one is on her route to work(K_B), and one is close to her home(K_C). It is impractical to check her email and do other work at K_A because the 15-30 minutes break she takes during the day is never enough to walk over there and even properly load her emails. She could break her trip home into two, and stop by K_B, but by the time she is done, it might get too dark, and she might miss the minibuses going home. The only viable alternative is K_C because it is closer to home and she can go there after work. However, that is also true for many others and there is a long wait there. Even after that, there are many people using the lines and the connections are terribly slow.

Elsa, like most people in the cities, has a basic cell phone to make calls and send SMS messages. This is what the solution in this chapter takes advantage of. In the system, she keeps the phone numbers of the three kiosks in her phonebook. About an hour before her break during work, she sends an SMS to the kiosk system at K_A, with her estimated time of arrival, thereby requesting her data to be fetched for her. After looking at the allocated resources, the system informs her whether her request can be serviced or not. If not, it also suggests the earliest available time at which it could be. At the scheduled time, she walks over to K_A and provides her password to access her data, which is ready and waiting for her. She could also do the same with any of the other kiosks depending on her daily plans and routes. By providing an efficient and secure way to deliver private data in challenged networks, the system eliminates most of the wait time when using the internet in those environments. Using service plugins, or *channels*, the infrastructure is able to deliver data from various sources such as email providers, RSS feeds, web portals etc. It also allows internet kiosks to better manage the scarce network resources and provide more value to their customers.

The main contribution of this work is a framework for distributing private data securely to nodes in challenged network environments that leverages usage patterns and an advance notification mechanism. In addition, the framework is used to implement three different channels to deliver personal and personally interesting data from various services. This chapter also reports on a test deployment that was run in Addis Ababa, Ethiopia.

5.2 Design

Sulula provides a framework for distributing personal data in challenged networks environments. At the core of the system is a data transfer infrastructure between two end points with the following properties:

- Supports an advance alert mechanism, triggering data transfer at the request of the owner.
- Supports resource management with a scheduling component that factors in available network resources and priority.
- Operates in challenged networks with low bandwidth and high latency, delivering data ahead of expected need.
- Provides a content agnostic interface allowing applications to embed their own semantics in their data.
- Provides confidentiality, integrity and source authentication - necessary properties for secure transmission of private data.

All data in the system is associated with an owner, which is represented by an opaque ID that is consistent across the system. A typical usage pattern for the system involves data creation or aggregation at the source, data request by the user, data delivery through the Sulula infrastructure and finally data consumption. This section will first discuss some of the guiding principles considered in designing the system.

In the expected usage pattern, users know ahead of time where they want their data delivered, and provide a hint to the system to initiate the data transfer. However, a user is not always guaranteed to consume the data requested, and as a result, there might be cases where a delivered piece of data is never consumed or consumed in time. In addition, most of the data that flows through the system has a certain freshness constraint, which

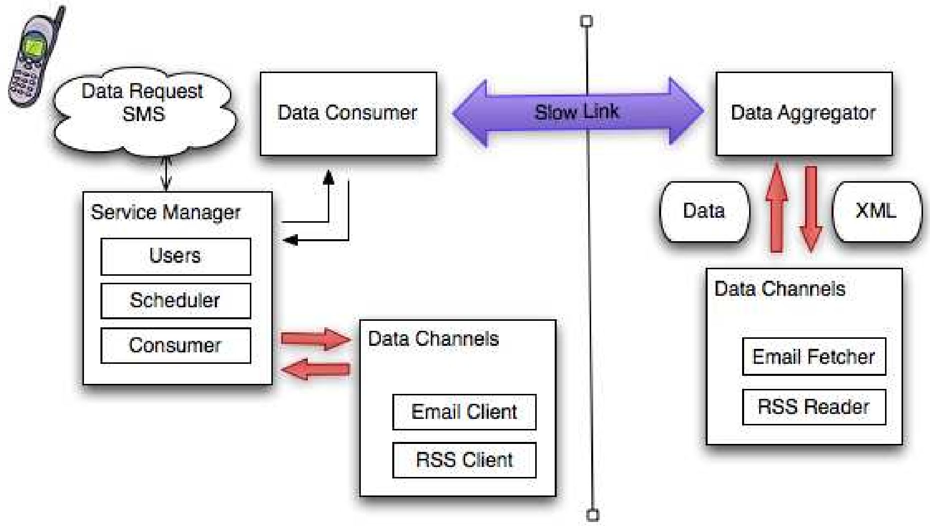


Figure 5.1: Architecture of the Sulula infrastructure

makes data interesting only within a given time frame. Having a time-to-live associated with all delivered data enforces these constraints. In addition, it relieves the data source from having to keep state information about old data, which in most cases is not valuable to the user in any event.

Freshness of data in this case is defined relatively. The nature of challenged networks makes it infeasible to transfer real time, or near real time data. Sulula is best suited for data that stays fresh longer than it takes to deliver it. The system leverages this to schedule transfers based on available resources and expected consumption time of the data. By providing metadata about the data to be delivered, the source assists the resource management and scheduling at the data consumption point. When requested data is ready to be delivered, the data source packages it by compressing and encrypting it. This package is delivered to the consumption point along with the time-to-live. By transferring the bulk of the data before a user gets to the consumption point, the interactive operation that requires a network connection once the user arrives is reduced to obtaining a session key for the encrypted content. Keys are distributed to consumption points only upon the request and authentication of users.

So far, this chapter has described the properties of the data source and the consumption point. A consumption point is where the user interacts with the system. In challenged network environments, this could be an internet kiosk or a library computer that a user has access to. On the other hand, a data source is any service that has some data of interest to a user. This could be an email provider, a social network site, a blog etc. that has either personal or personally interesting data available for the user. Unfortunately, it can not be expected that existing services will adopt support for the system. Instead, Sulula relies on an aggregator, which supports the Sulula interface, but also knows how to interact with existing services, and puts relevant data in the system to be accessed by clients.

The system supports data aggregators by way of data channels. A data aggregator is a server located in a well connected environment that supports the Sulula interface and collects data that is of interest to users. Data collection is done using data channels, which are enabled by plugins that run at the aggregator and at the data consumption end. An email channel, for example, could aggregate a user's email at the server and make it available to the data consumption end using the Sulula infrastructure. Figure 5.1 shows the Sulula system with two implemented channels.

5.2.1 SMS

The International Telecommunication Union's 2010 ICT development index [1] reports that by the end of 2010, there were over 5.3 billion mobile cellular subscribers worldwide, translating into a penetration rate of over 70%, which compares to just about 1.3 billion fixed telephone lines - or 19 per 100 inhabitants. This makes mobile phones easily the most widespread modern technology the world has ever seen. Furthermore, mobile phones may well have the greatest impact on development [34]. In fact, the ICT index states that over 60% of the world's mobile subscribers were from developing countries, with Africa having the highest mobile growth rate—one in four people have a mobile phone, significantly higher than the Internet penetration.

Using SMS to alleviate the acute lack of network resources, therefore, leverages this wide penetration. As most internet consumers also have access to a mobile phone (close to a 100% in the user study described in section 5.5), integrating SMS is not a difficult transition. Having an SMS interface for requesting transfers, rather than something like a fixed schedule of data delivery, allows the users to choose when they want to go to internet kiosks without having to deal with stale data, and gives them the option of comparing multiple service providers before committing to buy a service. On the other hand, it is also important to make adoption by kiosks and other internet providers as simple as possible. For this reason, the system uses a smartphone connected to a commodity PC to serve as an SMS gateway that can receive and send messages while interacting with applications running on the PC. In the deployment, second-hand smartphone bought for less than $50 is used. Such a low price point reduces the barrier to entry for business owners and encourages them to offer Sulula as an option. In addition, this phone could also double as the kiosk cell phone as most kiosks have an additional business letting people make long distance calls.

5.2.2 API

To accomplish the above properties, Sulula provides a pair of API's, one at source of the data, and another at the consumption point, which expose the following functionalities. At the data source end:

- `Add Owner` - Adds an `<owner>`, a unique opaque identifier, to the system. Data is private to its owner.
- `Put Data` - Inserts data for a given user.
- `Ready Data` - Packages (compresses and encrypts) all available data for a user, as well as provides information about the data.
- `Fetch Data` - Delivers data from the source for an owner. This data is encrypted and has a specified time to live, after which point the key becomes invalid.

- `Get Key` - Provides a key for a data transaction by owner, as long as the request is made within the TTL given in the `Fetch Data` request.

At the consumption point:

- `Request Transfer` - Enables an `<owner>` to request for scheduling of data transfer to a particular consumption point. This is exposed through an SMS hook for delivery scheduling, and could also be a network message.
- `Initiate Transfer` - Contacts the data source about available data for an owner and schedules its delivery, and provides information about earliest availability time of data.
- `Ready Data` - Makes data ready for consumption, decrypted and in the state it was stored in `Put Data`.

To enable communication between data channel pairs at the source and consumption point, the system supports two additional operations:

- `Add Channel` - Each `<owner>` can have any number of supported channels associated. Channels are uniquely identified and are described as plugin specific XML messages that enable data collection and aggregation.
- `Deliver Plugin Data` - Plugins can send chunks of data asynchronously to their counterparts on the server. Such data is associated both with the owner of the data and the particular plugin that generated it, and is delivered to the server based on a system wide scheduling of available network resources.

A fully implemented channel provides three operations:

- At the consumption point:

  - Add corresponding channels to the system (with channel details crafted as XML strings that include the information the server component of the plugin would need to collect user data)
  - Given the user data, be able to attach semantic meaning and display it to the user. This often could be as simple as invoking a browser with HTML data, but could also involve presenting email messages, RSS feeds etc.

- At the aggregator end:
  - For a corresponding channel, Put Data in the Sulula system for the owner.

To better explain how channels function in the Sulula system, lets look at how an RSS channel might work. Let's say the system has user <Elsa>.

- At the users end point, an RSS plug-in would use Add Channel to add the following for <Elsa>
  - (RSS, "<RSS feed='www.umich.edu/press.xml'> </RSS>")
  - (RSS, "<RSS feed='www.someblog.com/feeds' user='foo' pass='bar'></RSS>")
- At the data aggeregator, an RSS server-plug-in would use this information to fetch and package instructed data and use Put Data to place it in the system.
- Finally, when data is ready for consumption at the user's end, the system uses the channel plug-in to display it to the user (again, often with a browser).

5.2.3 Uploads

In addition to requesting future visits, Sulula allows users to schedule data uploads. Scheduling such uploads is delegated to data channels. Consider an email channel, for example. When the user is finished with an email session that might have included reading, tagging and composing mail, the channel compiles all data that needs to reach the aggregator so that further network action, such as delivery to the SMTP server, can take place. The channel then uses the `Deliver Plugin Data ()` API call to pass this data to the Sulula infrastructure, which schedules it for asynchronous delivery to the aggregator. The channel counter part on the aggregator can further process this data and communicate the results to the kiosk email client, such as the successful transfer of mail to the recipient.

5.2.4 Capacity

The ability of consumption points to service user requests is an important factor in determining the practicality of the system. The week long observation of a medium sized internet kiosk in Addis Ababa with five PCs and two dial up lines, as summarized in table 5.1 below, shows that kiosks are busier during the late afternoon and early evening hours as students and workers make up most of the consumer base. This leaves ample opportunities for the kiosk to queue and service requests from Sulula users through the rest of the day. In addition, dial up fees are significantly cheaper during off-peak hours (for example, about half off for the 6 pm - 8 am block of the dial-up service provided by the Ethiopian Telecommunication Corporation [35], a monopoly within the country). This increased capacity enables kiosks to take requests for overnight deliveries, at possibly reduced prices. Finally, Sulula supports a transparent scheduling system that notifies users the earliest possible time data would be available if their request can not be serviced by the desired time, which allows users to weigh their options from multiple providers. This could be especially useful when some of the kiosks are already heavily subscribed.

5.3 Implementation

Sulula is implemented as a pair of services that run at the data source and consumption point, and communicate using XML-RPC [132]. Both ends provide API's that plugins and applications can use to deliver data. The data source was implemented in a UNIX

Table 5.1: Customer load at a participating internet kiosk

Two hour block	# of customers
8am - 10am	2
10am - Noon	2
Noon - 2pm	2
2pm - 4pm	2
4pm - 6pm	3
6pm - 8pm	5
8pm - 10pm	5

environment for seamless integration with most online services while the data consumption end runs on a Windows environment for ease of use at internet kiosks and other service providers. Using the provided plugins, the server can also double as a data aggregator. As described shortly, data channels are implemented as simple modifications to existing applications that are used to collect and consume private data.

The data source service consists of data processing, plugin manager and XML RPC server components. The data processing module is responsible for packaging and encrypting data and making it ready for delivery. Sulula uses the 128-bit Advanced Encryption Standard (AES-128) for block encryption and MD5 for hashing. These could easily be replaced with other encryption and hashing mechanisms. The XML RPC server runs on a well-known port and can handle multiple requests simultaneously. The plugin manager interacts with channel plugins, and provides the interface plugins use to manipulate and deliver data to users.

The data consumer service is implemented as a stand-alone multithreaded process that accepts requests from users using an SMS interface, and interacts with plugins as well as the data source. It was implemented on a Windows platform as Windows represents the overwhelming majority of operating systems run on computers used in developing countries [95]. The system was built using the .NET 2.0 framework, and make use of an open source XML-RPC package [24] to interact with the data source. Sulula also usesthe MSR SMS Toolkit [128], an existing SMS gateway solution from Microsoft Research India, to handle and respond to requests from users via a connection to an SMS sending/receiving port. The SMS toolkit is a simple programmable interface to SMS messaging, with hooks to applications that enable integration with various systems. A smartphone communicating with the processes running on the PC using the toolkit serves as the SMS sending and receiving port. Meanwhile, the phone can also be used for making and receiving calls.

All transaction requests involve three SMS messages per scheduled visit. First, the user requests their data to be delivered, along with their estimated time of arrival. A request can have different levels of priority, which would come at different costs, and determine how soon the request would be serviced, as well as the amount of resources allocated to the user's data. The system responds with when the request could be serviced and the cost of the service, asking the user to confirm the transaction within a deadline. This allows the user to gauge the freshness of their data as well as the cost. For example, it might be more valuable for a user to get their data in an hour rather than tomorrow, even at half the cost. Upon confirmation from the user, the system schedules the request for service. A simple SMS session might proceed as follows:

- user data request
  - `<fetch 30 minutes urgent>`
- system confirmation
  - `<Data could be ready in 30 minutes. Cost 4.5 Birr.  Confirm?>`
- user confirmation
  - `<Yes>`

5.3.1 Data Channels

Data channels provide an easy way to deliver personal data to the user through the framework, and represent how Sulula integrates into the existing infrastructure. These are often simple modifications to existing applications enabling them to use the Sulula API. To demonstrate how Sulula supports the plugin architecture and provide working examples of how channels can be effectively used to aggregate and consume personal data, three channels have been implemented on top of the system: an email channel, a news feed channel and a simple HTTP channel. These channels show how the system can be easily integrated to the existing infrastructure and incrementally support various data sources.

5.3.1.1 Email channel

Email represents the majority of private data that is consumed in developing countries [100]. This channel supports email communication using the most common mail protocols. The PyMailGUI email client [81], a lightweight email application written in Python/Tk, was modified to support the Sulula infrastructure. The client side plugin allows for secure registration and displaying of email messages, while the data source plugin interacts with service providers to fetch and send email. The data source plugin uses the owner's credentials to authenticate and fetch mail from a POP3 or IMAP server and send mail using SMTP much like a desktop mail client would. The basic difference between using the email channel and an unmodified desktop email client is that in the system, email administration and display occur at the data consumption end whereas email is fetched and sent at the data aggregator, with the two ends communicating using the Sulula infrastructure. To accomplish this, the application is divided to two parts as shown in Figure 5.2. By employing an asynchronous data transfer mechanism provided through the system, the email channel provides a much more pleasant experience of email communication in challenged environments. The modifications took a total of about 130 lines of Python code to implement.

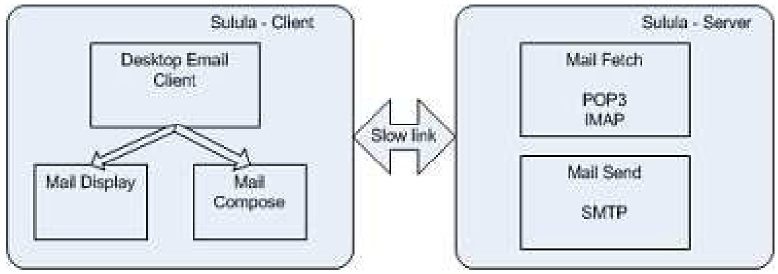


Figure 5.2: Implementing an email channel with Sulula

5.3.1.2 Newsfeed channel

This channel is used to aggregate news feeds in any of the common forms used on the web today, including RSS, RDF and Atom. The NewsFeed [30] news aggregator was modified to use the infrastructure for transferring data in a challenged network environment. As shown in Figure 5.3, the application is broken to two parts, with the aggregator running on the data source and the news displayer running on the data consumption end. As described in the example given in section 5.2, a user can register any news feed that she is interested in using the client plugin, which gets transported to the data aggregator. The newsfeed channel also supports authenticated feeds, which allows users to supply their credentials for accessing private news feeds. News is fetched asynchronously when the system gets a request for the user's data. Less than 50 lines of code were needed to modify this application to use the Sulua system.

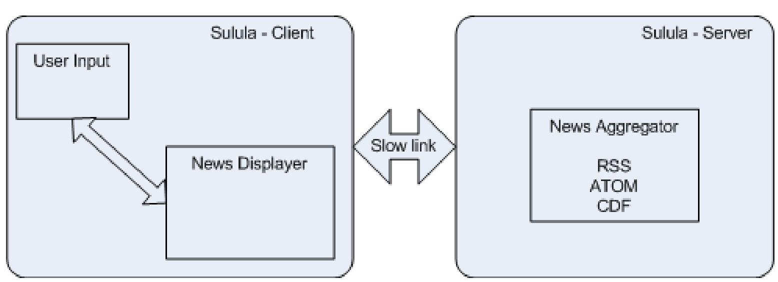


Figure 5.3: Implementing a newsfeed channel with Sulula

5.3.1.3 Simple HTML Channel

This channel provides data aggregation and display for simple HTTP/HTTPS pages. Studies of web access patterns [19, 32] show that there is a long tail of niche destinations that are often requested by a single user or a handful of users at most. This channel is useful for handling such requests that stand to gain little to nothing from conventional caching mechanisms. The client plugin allows users to input webpages they would like to be fetched ahead of time. These could be web portals, educational sites, fast updating news network sites etc. The data aggregator side plugin uses the request to download and package an HTTP/HTTPS website and readies it for delivery. This plugin can also be used as a building block for more sophisticated channels.

5.4 Evaluation and Discussion

An ICTD researcher working in Uganda recently wrote about her experience using the internet there, saying [52]:

> At the moment, I've totally given up on using my 64/64 WiMax+VSAT link via Infocom (which costs $300/month) and I'm using my Warid Telecom GPRS/EDGE modem (cost $60 + $40/month), which incidentally also claims speeds up to 128kbps (16KB/s), but in reality usually sits at about 2-5 KB/s on a good day (I am getting about 1.0KB/s now). The MTN EDGE/HSDPA service ($150 + $45 modem) is a bit of a joke and I have never seen it go above 1-2 KB/s (It's supposed to be 384kbps, or 48KB/s). I can't send emails ...

The main goal in building the Sulula infrastructure is relieving this kind of frustration, which is all too common in developing countries. This section will discuss how the system improves user experience and adds value in these environments.

The evaluation uses a client computer with a Windows Vista operating system running on an Intel Core 2 Duo processor, connected to a phone running Windows Mobile 5 that can send and receive SMS messages. The data aggregator runs the Mac OS X Version 10.5.5 operating system, also with a Core 2 Duo processor. To simulate a challenged network environment, it used a WAN simulator [113] to cap the client's bandwidth at 15Kbps and introduce a latency of 500ms [10] as well as a 5% packet loss rate. These are reasonable numbers that a user might expect when going to an internet kiosk, and often better than the experience at several internet kiosks in Addis Ababa, Ethiopia.

The workload consisted of checking email from a common service provider, Gmail, where there are a total of 5 new email messages in the users inbox, resulting in 1 MB of data transfer. The numbers are based on a recent study of email usage patterns [112]. The user also sends out one email with an attachment of a 200KB file. In total, the user has about 1.2MB of useful data transfer in the email session. In addition, the workload includes one webpage and one blog that publishes a news feed, for a combined data transfer 500KB, to the hypothetical user's routine. The setup is summarized in table 5.2. Based on this setup, the system is evaluated from two perspectives, cost of using the system, including SMS messages, and also in terms of time savings realized.

Table 5.2: Experimental setup for evaluating Sulula

Data size		Bandwidth	Latency
Incoming Email	1MB	15Kbps	500ms
Outgoing Email	200KB		
Web & Newsfeed	500KB		

5.4.1 Performance

The biggest gain from using the Sulula system is significant time savings. Users spend a lot of time waiting in line, as well as slowly accomplishing tasks on the internet. When the scenario is run in an unmodified kiosk environment, in addition to taking 33 minutes on average to complete, the user had to refresh the page and restart downloads a few times because the page would simply hang. Sulula significantly cuts this wasted time by making a user's data available and ready before the user gets to the kiosk, as well as asynchronously transferring user generated data back to the data source. Data access from storage is almost instantaneous, and the only on-demand transaction is retrieving the encryption key from the server, which takes less than a minute. Going back to the previously mentioned user, Elsa, in Addis Ababa, this could mean being able to effectively use her time during the day, and freeing up the evenings for other important activities.

Table 5.3: Time savings from using Sulula during the experiment

Avg. Network Interactive Time	
Existing Kiosks	33 Minutes
Sulula	< 1 Minute

5.4.2 Cost and Pricing

As it stands now, most kiosks in developing countries charge their users by the minute spent connected to the internet. The average price from a visit to Ethiopia was around 0.4 Birr (about US $0.04) per minute. This is slightly higher than the cost of a single SMS message (0.3 Birr) as provided by the Ethiopian Telecommunication Corporation (ETC), which is the only service provider in the country.

A user is charged the per-minute cost regardless of the actual bandwidth delivered. So, if one spends about an hour trying to accomplish a task that is taking much longer than it should, it will cost about 24 Birr all together. Instead of this, the Sulula system encourages charging users based on the amount of data transferred. This is fair to the user because internet kiosks often have multiple users share a single dialup connection, making any of the bandwidth promises irrelevant. Because this also encourages transparency, it would promote business among users who feel like they are paying for undelivered service. Users are also likely to be willing to pay more for such a service because of the significant time savings realized. Finally, the kiosks are free to set up their own prices as they see fit, and let the market determine who has the best value. To facilitate this approach, the system logs transaction information such as data size and date, and makes it available to users and kiosk owners.

The three SMS messages per transaction cost a total of 0.9 Birr. If a kiosk charged 5 cents (0.05 Birr) per 10KB of data transfer, the experimental scenario would cost the user a total of 9.5 Birr, including all SMS messages. This price point is hypothetical, and could be different in the market.

On the other hand, running the scenario in an unmodified kiosk environment on average takes 33 minutes to complete under the network settings described, which translates to 13.2 Birr. Even if kiosks were going to adjust their prices to reflect how much they could have made without using the system, the approach has a more tangible way of measuring the service provided, and by extension its value. This means, in this scenario, kiosks could raise their price as high as 7.1 cents per 10KB, and still not charge the customers any higher than what they already pay, but at a much higher level of satisfaction. In addition, the kiosk can provide different tiers of service based on how soon users want their data, and prioritizing requests accordingly.

Network resources in developing countries are limited. However, researchers can work within a few constraints to increase value both to consumers and service providers. For example, the user might have some preferences such as the location of a kiosk in a certain part of town that does not require taking the city bus or the availability of additional services at a kiosk. In addition, different users will have different time frames and urgency regarding when they want their data available. On the other hand, kiosks can better manage their resources as well. For example, there could be discounts for orders placed in advance while charging more for short notices, peak hours etc. In addition, overnight fetched bits could be cheaper, when demand for the kiosk is low, and bandwidth is generally cheaper.

5.4.3 Limitations

Basic limitations such as high latency in the network environment make delivering interactive data to the user, such as supporting an active browsing session, a bit tricky. One naive way to achieve interactivity in networks with very high latency would be prefetching huge amounts of data. However, this is not feasible since resources are limited to begin with. Nonetheless, applications can develop smarter mechanisms that leverage Sulula to provide an experience that resembles interactivity. For example, plugins could analyze usage history to selectively request what pieces of data to download, and provide a pseudo-interactive experience.

Another limitation in using the Sulula framework is the diversity of private data sources and the quirks in accessing them. For example, how a user authenticates to and consumes data from one social networking website could be different from how data is accessed from an otherwise similar service. One possible solution is implementing a data channel that learns the user's access behavior of various websites within a browser, and uses the information for accessing and presenting data in future sessions. In addition, the wide adoption of open source authentication and authorization techniques, such as OAuth [93] and OpenID [117], allows for more generic solutions.

5.5 Test Deployment

A small in-country deployment of the system was run at a medium sized internet kiosk in the outskirts of Addis Ababa, Ethiopia. The local partner has been operating the business for about a year, and provides a number of office services in the kiosk besides internet connectivity. This section will describe the initial usage study, the telecom infrastructure in Ethiopia, and some details about the experience setting up and running the deployment.

5.5.1 Telecom in Ethiopia

The Ethiopian Telecommunication Corporation (ETC) is the sole provider of communication services in Ethiopia, spanning land lines, mobile phones and internet services. The latest statistical bulletin from ETC[36] shows Ethiopia has one of the lowest telecommunication penetration rates in Sub-Saharan Africa, with both internet and mobile phone subscribers less than 1 in 100. Mobile phone service was introduced in the country in 1999, and internet access only a few years earlier. In spite of this, Ethiopia has one of the highest mobile phone growth rates, with subscriptions increasing by more than 164% during the last two years of the bulletin, which is the third highest in Sub-Saharan Africa next to Nigeria and Angola, according to the ICT Index [1].

Internet subscription, on the other hand, has not been expanding nearly as fast, which can be attributed to the lack of infrastructure and the high cost when available. At the end of 2005, ETC reported less than 18,000 internet subscribers in a country of nearly 80 million. The number of internet users was estimated around 113, 000 and the number of personal computers around 225,000 during the same year. The cost of service ranges from 160 Birr setup and 60 Birr monthly charge for a 900 minute dial-up connection to 111,000 Birr setup and a monthly charge of 48,000 Birr (around $4500) for a 2MB leased connection [35].

These numbers paint a grim picture about the state of affairs regarding network resources in the country, which is also common in many other developing counties. Since the situation is likely to continue for the foreseeable future, systems like Sulula are helpful in improving the experience of people accessing the internet in these countries. As a result, Ethiopia was an ideal place for deploying the system.

5.5.2 Initial user study

At the beginning of the deployment, 12 internet kiosk users were interviewed about a range of issues regarding their internet usage patters and mobile phone access, along with some demographic background. Questions asked included how long they have been using the internet, how often they come to a kiosk, how they use the internet and what some of the inconveniences are. Table 5.4 summarizes a few of the responses.

Some of the relevant takeaways from the study were:

- The average internet user is a young worker with at least a high school education

Table 5.4: Selected responses from initial user study on internet kiosk usage

Average age	25
Average # of visits per week	2.2
Common visit time	late afternoon, evening
Average session length	44 mins
Common uses	Email, Sports, News, Chat
Avg # of email accounts	1.3

- Most internet users were also mobile subscribers
- Webmail was the most dominant use case for internet access
- Most access times overlap, usually during late afternoons and evenings

5.5.3 Deployment experience

5.5.3.1 Language issues

The national language of Ethiopia is Amharic, a Semitic language with its own unique alphabet. English is thought in school as a secondary language, allowing us simple communication with most kiosk customers. However, local translation of documentations and interviews were provided to make sure they were well understood. An Amharic video tutorial was also made for the user-facing part of the system with the help of a partner, and provided translation help with all of the interviews and surveys.

5.5.3.2 Shipping code and troubleshooting

One of the main issues faced in starting up the deployment was the difficulty of transferring program files and patches on demand. The poor network conditions make it prohibitively difficult to transfer big libraries and toolkits over the wire. The initial version the software was physically mailed, along with supporting libraries and tools, which took more than two weeks to arrive. In addition, it was important to adjust how often the software was updated and take careful notice of dependencies and additional requirements. For example, at one point, it was faster to modify the backend of the system to use a different database engine that was available in-country and upload the patch, rather than ship another version of the original database engine that was compatible with the specific OS installation at the kiosk.

Troubleshooting was another sticky point as many helpful tools such as remote desktop or video chat do not work well over the available network. As a result, instant messaging and low-resolution screenshots were the main options for solving some problems that came about during the initial phase of the deployment.

5.5.3.3 Pilot run

After the initial user study, four volunteers were solicited to test Sulula for two weeks. They were encouraged to use dummy email accounts and non-sensitive web content to avoid privacy issues during a testing phase. At the end of the two weeks, the users were asked to fill a survey describing their experience—giving us valuable feedback in making the system work better. The prominent pluses in the survey included the local language video tutorial that explained how to use the system, and the ability to multitask, allowing users to 'go about their business' while having data downloaded or uploaded at the kiosks.

In addition to fixing some bugs, users also asked for a few feature improvements in the system. These include a better and intuitive user interface and a streamlined registration process that could possibly be shared among multiple kiosks. A more technical suggestion included the ability to describe how deep the HTML channel should go in downloading links on a specified page. This goes hand in hand with supporting pseudo-interactive sessions using Sulula.

5.6 Discussion

Traditional caching and prefetching techniques provide little to no assistance for personal data that is needed only by a single user. In challenged network environments, this means users often have to wait while their data is fetched from a remote server over a very slow link. This chapter presented Sulula, a system for distributing private data in challenged networks. Sulula leverages the wide availability of cellular telephones to enable users to request future accesses of data, thereby scheduling data transfer in response to capacity. This approach builds on the observations of increased personalization in data access, and provides mechanisms for a user to specify exactly what content needs to be prefetched. Since the cost of mispredictions in data access is high in challenged networks, this system defers content selection to users.

Having users request content when they need it, rather than relying on a period or automated content transfer, works better in developing regions where routine planning might not work for some users. In Sulula, users request data only when they want to, and are not tied to a schedule. Experimental results show positive results for Sulula, saving users tens of minutes on typical workloads. The evaluations for Sulula consisted of network emulation as well as field deployment in Addis Ababa, Ethiopia.

The next chapter of the dissertation focuses on another challenge in constrained environments—moving bulk data. The acceleration of content generation has demanded efficient mechanisms for transferring bulk data, but challenges in the network infrastructure make this difficult in developing regions. The next chapter focuses on tools for assisting researchers as they design and prototype systems for challenged networks that could help in these regions.

CHAPTER VI

Tools for building systems for challenged networks

6.1 Introduction

The digital universe has been rapidly expanding, and bulk data makes more than 70% of the traffic on the internet [11, 42]. Alongside individualization of content, the internet has become increasingly media-rich. Unfortunately, much of this content is not easily accessible in many developing counties due to the poor network conditions. For example, downloading a 10MB audio lecture freely available from the University of Michigan's Open Initiative would take a few hours to complete in a typical shared dial-up setting of an internet kiosk, rendering it effectively out of reach for a student who has to pay by the minute. A video lecture, which has hundreds of megabytes, is often simply too large to access. Therefore, it is useful to design efficient mechanisms that can assist in the delivery of bulk data in challenged networks.

Fortunately, challenged networks have been receiving increasing attention from the research community in various contexts [53, 80, 89, 90, 120]. The IP suite of network protocols perform poorly when faced with environments characterized by long delays and frequent network partitions [33, 37]. As a result, systems for challenged networks are concerned with enabling communication through intermittent links, mobile nodes and across several regions of connectivity despite poor infrastructural support. Designing systems for challenged networks poses many interesting problems, such as coping with disconnection, efficiently utilizing poorly-provisioned links, incorporating mobility and supporting interoperable naming.

The goal in this part of the dissertation to develop mechanisms for efficient bulk data transfer with developing regions in mind. However, the first challenge that surfaced was the difficulty of experimenting with systems for challenged networks. Systems designed to work in these settings must support communication through intermittent links and mobile nodes in a variety of contexts. Therefore, it is important to experiment with a variety of designs, evaluating them under a variety of circumstances.

Unfortunately, such experimentation is difficult. These systems tend to be large and complex, requiring a significant implementation effort just to route messages from one node to another. This is confounded by the lack of common infrastructure, which implies considerable plumbing even for basic services. In addition, in-situ evaluation of these systems is expensive, time-consuming and difficult to do in a reproducible way. As a consequence, some have turned to pure simulation of systems and their environments to provide explorations in the design space [53, 64, 79, 80, 82, 97].

Such simulation, however, necessarily requires simplifications, and simplifications hide many issues that are often important in understanding the overall system performance. Experimenting with an implementation of the system, rather than its simulation, enables the researcher to factor in numerous networking, systems and mobility considerations that otherwise would have been abstracted away (Section 6.2.1). Often, these considerations are important in analyzing the overall implications of design decisions in the system. For example, researchers might be interested in examining systems issues such as concurrency and resource consumption, or closely exploring security related concerns such as decentralized control of data access and message encryption. Indeed, prior work in the Mobility group at Michigan has often tried to move away from pure simulation in a variety of mobile settings for similar reasons [92, 135, 136].

This chapter introduces a platform that enables researchers to quickly implement challenged network systems, and reproducibly evaluate the implemented systems at scale. This platform is called Vivo. The key contribution of Vivo is allowing researchers to rapidly prototype these systems with live-code, and emulate their environment. In doing so, Vivo significantly reduces the effort needed to implement systems for challenged networks, and facilitates their iterative and reproducible evaluation. Vivo has two parts. It consists of a toolkit of composable and replaceable component implementations, alongside a mobility and networking emulation platform for evaluating the implemented systems. The toolkit provides a menu of composable components that interact over well-defined interfaces, allowing researchers to focus only on the novel contributions in their systems, while providing suitable defaults for other behavior. The emulation platform enables the iterative and reproducible evaluation of systems for challenged networks. In combination, Vivo resolves the experimentation dilemma by presenting an option for quickly developing implementations of systems, while keeping the benefits of reproducible evaluations.

Implementing a system with the Vivo toolkit is akin to assembling components to make a whole system, which liberates researchers to innovate only in their areas of interest. The toolkit consists of a composable collection of modules that provide as much of the common infrastructure as possible (such as networking, storage, error propagation etc.), alongside a number of example implementations for higher level services. These modules interact over a set of well-defined interfaces that give researchers the flexibility to easily swap any of the various modules with their own implementations, incorporate new modules or simply modify existing ones. This allows for rapid prototyping and experimentation of a variety of designs, and direct comparison between alternative approaches. In addition, a component-based architecture better supports changing and unexpected workloads as new application domains emerge for these systems.

Vivo provides an accompanying network and mobility emulation platform for evaluating systems built atop this toolkit. In-situ evaluation of data delivery systems for challenged networks is often prohibitively expensive for many scenarios. As a moderate middle ground, Vivo relies on a managed testbed (either local, or remotely accessible, such as EmuLab [130] or ORBIT [101]) for network emulation, and incorporates a mobility layer that can be driven by real-life traces or probabilistic mobility models. Importantly, Vivo need not model specific motion trajectories, because these systems tend to have few participants over large physical distances. Instead, the platform emulates node encounters, capturing the essential outcome of mobility in the context of challenged networks. Thus, the platform is able to represent many participants and large geographic areas even on a small cluster. In turn, researchers can perform a full-stack evaluation of their system, with an option to scale easily.

This chapter begins the discussion by comparing Vivo with existing projects, focusing on its contribution as a platform for live-code experimentation of challenged network systems. It then describes challenged networks more precisely, and characterize existing data delivery systems for these networks. This will specify the kind of systems that could be built with Vivo (Section 6.2). It continues with a detailed description of the platform itself. In particular, it focuses on how to achieve flexibility in system design (Section 6.3). The evaluation for Vivo show that it reduces the effort in building challenged network systems, and facilitates their iterative and reproducible evaluation. This chapter demonstrates the ease of building systems with Vivo by implementing two previously proposed data delivery systems with it. The next chapter will provide a more detailed account of the design and evaluation of a new hybrid overlay network for bulk data that was built with Vivo. It will also show how Vivo enables researchers to evaluate the impact of their design decisions by analyzing the three systems.

6.2 Design

Challenged networks, such as those common in developing regions, are characterized by unreliable and poor infrastructure, where connectivity comes and goes, and the effective bandwidth among network endpoints is usually low. The lack of always-on infrastructure implies high packet drop rates and frequent transmission failures. Consequently, local connectivity is often better than using the infrastructure. As cell phones become more affordable globally, many users are carrying with them a device with modest storage, local communication and computational capacities. In combination, challenged networks are distinguished by diverse and intermittent connectivity atop poorly provisioned infrastructure.

Due to the failure of IP to work well in these environments, several systems have been proposed for tackling the challenge of delivering data despite unreliable connectivity. Vivo was designed to aid in building such systems for live-code experimentation. But what sort of systems can ideally benefit from Vivo? To answer this question precisely, this section first look at the underlying assumptions made about data routing systems for challenged networks. These assumptions are based on analysis of many existing systems for challenged networks, and attempt to capture shared principles across a range of them. In turn, these assumptions allow us to build a composable set of reusable modules for implementing similar systems old and new.

Vivo assumes data delivery systems for challenged networks consist of end-users and nodes. Nodes are defined as devices capable of storing messages. Typically, a node might provide additional functionality on top of storage, such as computation. In addition, some of these nodes might be connected to the network in varying capacities. Nodes could also be mobile or stationary. To give a few examples, a thumb drive—commonly used to carry data in shared access sites in developing regions [14]—can be considered a storage node for moving data through mobility, although it does no computation or networking. On the other hand, a kiosk computer is a network and computation capable node, although stationary. By relaxing the definition of a node in the typical system, Vivo is able to support a range of mobility and network based routing protocols.

Vivo assumes data delivery systems allow end-users (or their applications) to send messages to other end-users. To facilitate transfer, it assumes end-users and nodes are identified by system-wide aliases. Vivo makes no further assumptions about what aliases mean, and different systems might utilize different naming schemes in routing messages. For example, aliases might resolve to an underlying dynamic addressing mechanism. Finally, it assumes end-users access their messages from nodes. Systems might associate end-users with multiple nodes, enabling message access from multiple destinations. By extension, when an end-user sends a message to another end-user, she indicates both the recipient alias as well as one or more node-aliases where the recipient accesses his messages. The routing protocol attempts to deliver messages to destination nodes where recipient end-users will be able to access them.

Many systems for challenged networks satisfy these assumptions [13, 15, 37, 64, 80, 111, 120, 125]. A large class of these systems operates as an overlay above the transport layer of the networks they interconnect by providing services such as in-network storage, interoperable naming and coarse-grained message passing. In order to deal with intermittent connectivity, a store-and-forward architecture is often an essential component of these systems. Furthermore, many of these systems attempt to aid the network infrastructure with individual mobility of end-users (and storage nodes that are carried alongside) in transferring data. Based on these underlying assumptions, Vivo provides a platform that simplifies the process of implementing and evaluating such systems for challenged networks.

The Vivo toolkit is composed of modules that interact over a set of callback interfaces. These modules provide generic infrastructure and a number of example implementations for various services in challenged network systems. The interfaces used to communicate between modules have both synchronous and asynchronous (event-driven) components. Well-defined interfaces allow researchers to easily compose and modify underlying modules in the resulting system. For example, a system might typically modify/replace modules that are involved in intra-node message routing, while reusing existing buffer management or analytics modules. This chapter demonstrates how this flexibility is utilized in building different systems with Vivo (Section 6.4).

In order to provide more flexibility in designing systems, the platform use dynamic objects for passing data across modules. It takes advantage of a feature in modern programming languages such as Python, C# and Ruby that gives developers the flexibility to modify objects with new attributes and methods at runtime, while maintaining the same signature for callback interfaces. This allows modules to pass additional data and code, while honoring the underlying interface for communication. This is a blend between Lampson's advice to keep interfaces stable [76], and the need to build an evolving set of systems using the toolkit. As a result, the toolkit can be used to build a nearly functional system out of the box, with the option to easily customize it to the builder's needs.

Over the next few sections, this chapter will discuss the components that make up Vivo in more detail. The system toolkit consists of six broad classes of modules, each module implementing a set of services for a potential system. These include groups of modules for providing basic functionality such as network communication, mobility management and runtime analysis. At a layer above these, a group of modules assists in facilitating message routing in the network system. Other components deal with services such as configuration, system startup and integration with the emulation platform. The discussion will begin with a selection of these modules and how they interact with each other. Later on, the chapter will discuss the evaluation platform that is used to iteratively experiment with systems built atop the Vivo toolkit. Figure 6.1 summarizes the main components in Vivo.

6.2.1 Why live experimentation?

Live experimentation of systems for challenged networks enables researcher to explore a number of issues that would otherwise be hidden by simulation. Often, these considerations are important in analyzing the overall implication of design decisions in the system. Through its composable architecture, Vivo isolates and exposes many of these issues—spanning systems, networking and mobility—allowing researchers to focus on areas they find interesting. A few examples of issues that could be considered due to live implemen-

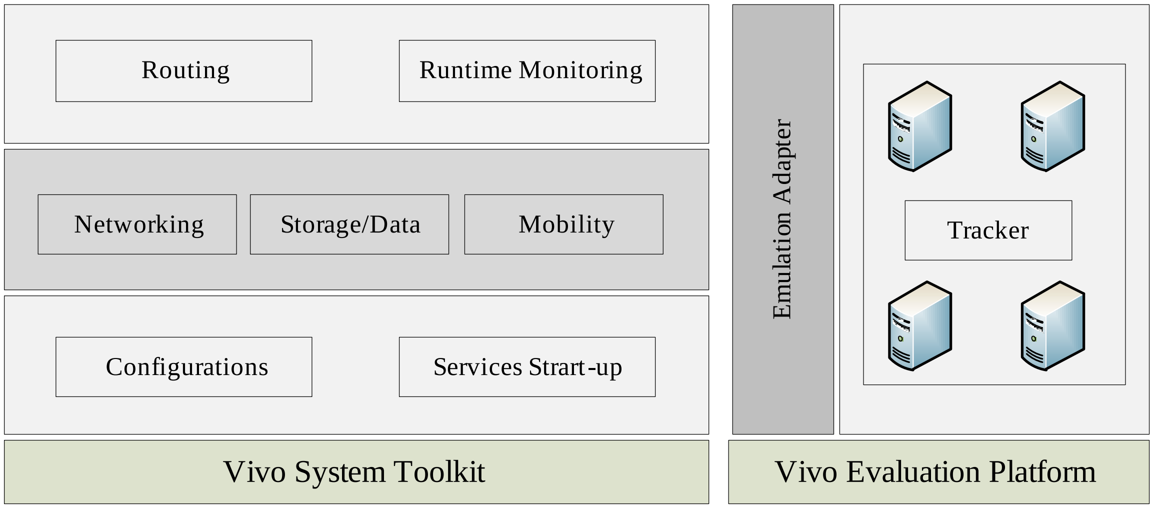


Figure 6.1: Main components in the Vivo platform for challenged network systems

tations include:

Storage management: Encoding and decoding messages, managing underlying storage and databases, data compression etc.

Systems: process management, concurrency, monitoring resource (memory, computation) consumption, performance optimization etc.

Platform compatibility: adapting to various platforms, diverse resources, capacity etc.

I/O: File I/O, user interaction models, communicating network failures etc.

Extensions: designing for plug-ins, external components, leveraging existing source code etc.

Logging: advanced logging, incorporating standard debugging tools etc.

Security: access control to messages, encryption, key distribution etc.

As challenged network systems support new workloads and application domains, many more interesting issues are likely to emerge. Vivo's component-based architecture can adapt to such advances, and live experimentation aids researchers in analyzing the impact of these issues.

6.2.2 The Vivo toolkit

The modules in the Vivo toolkit are broadly organized based on six classes of functionality: networking, mobility, storage, routing, runtime analysis and configuration. In addition to these core set of modules, Vivo provides a group of modules to liaise between the end-system and the evaluation platform. These groups of modules are organized into a few functional layers. At the foundation of the platform are three groups of modules that provide basic functionality common among data delivery systems: network management, mobility management and data/storage management.

Network management: consists of several modules for managing a node's network connectivity, if any. These include characterizing network links, sending and receiving application level messages, and managing information about other nodes in the system. To utilize a common parlance, the Vivo platform refers to the known information about another node as its *contact information*. Through interfaces exported by these modules, the rest of the system can get informed about networks and contacts, as well as deal with network failures.

Buffer management: provides a set of modules to work with data processing and management in the system. For store-and-forward architectures common in challenged networks, buffer management is an important issue. Some common operations include cross-platform data encoding and providing a simple interface to query the state of the buffer at a node. With additional interfaces for storing, retrieving or removing messages, a researcher can focus on implementing the system's high level buffer management strategy rather than the underlying details.

Mobility monitoring: consists of several modules to aid in documenting, processing and modeling the mobility of nodes in the system. Systems are often interested in capturing and modeling node encounters rather than motion trajectory. These modules lay the groundwork for mobility management by defining a format for exchanging mobility history, raising events as other nodes are encountered and providing example implementations for mobility models. In turn, these modules expose mobility information to the rest of the system through callback interfaces.

A layer above these is the routing group that provides several modules to facilitate message forwarding in the network. While the details of the routing protocol implementation are left to the system designer, these modules provide a number of hooks into the rest of the system to easily obtain information that might aid in making routing decisions. These include understanding network availability, node mobility and current storage capacity. Vivo also provides tools for processing messages as they are initially inserted into the system by end-users. As with the rest of the toolkit, researchers are free to compose only the modules they find useful or modify ones that are close enough in operation.

Other high-level modules allow a researcher to understand the system better through runtime analysis, providing a central place for monitoring resource consumption, errors generated or statistics about message delivery in the system. Each node collects information about its own view of the system, which can be aggregated to give a high level picture of performance, as demonstrated in the evaluation. At a lower level, Vivo also provides a number of modules to assist in configuring the system and starting up the various comprising services. The key design goal remains providing as much of the common infrastructure as possible, such that the system building effort can focus on the pieces that are novel to the project at hand. In section 6.3, the discussion will return to the implementation details and supported interfaces of some of these modules.

6.2.3 The Vivo Emulation Platform

Accompanying the Vivo system toolkit its emulation based evaluation platform. This platform was designed to streamline the process of evaluating systems built atop the Vivo toolkit. In particular, it focuses on enabling researchers to evaluate their systems as is, without having to duplicate their effort on a separate approach. As a result, the emulation platform is mainly concerned with providing a suitable interface to existing network emulation frameworks with challenged network systems in mind. An important aspect of this design is catering to the message passing paradigm, and encounter-based mobility models that characterize many data delivery systems in these environments.

Vivo's evaluation platform defers network emulation to an underlying framework such as EmuLab or ORBIT. These frameworks allow for a high fidelity representation of network conditions given a description of network connectivity for participating nodes. In addition, researchers might choose to set up Vivo on their own managed cluster with available traffic shaping tools. For example, this might be useful if one needs to run experiments locally, or scale experiments beyond resources available at shared research facilities. The basic requirement in using managed clusters is that they be able to enforce network connectivity requirements for a given node. Experiments have been conducted on the EmuLab framework, as well as on a large scale cluster using Amazon EC2. Running experiments on other network emulation platforms is fairly straightforward, and involves configuring the underlying network accordingly.

Modeling mobility is the other important component in evaluating systems for challenged networks. An approach for emulating mobility on a stationary grid is to use spatial switching or system migration for mimicking motion trajectories [99, 134]. In the case of message delivery systems, however, it is observed that mobility is interesting only to the extent that nodes encounter (and pass messages to) each other. As a result, there is no need to explicitly model motion. Instead, it suffices to represent the impact of nodes meeting each other in the wild. What this means is that if the independent mobility models of two nodes dictate that they encounter each other at some time-step, the emulation platform arranges for the two nodes to communicate directly over the network and exchange messages. As a result, the emulation platform allows researchers to model mobility in large geographic areas even a relatively small cluster.

The Vivo evaluation platform can support arbitrary models for guiding node mobility in the system. In many cases, node mobility is a proxy for human mobility, which drives mobile nodes in a data delivery network. A number of mobility models have been suggested for capturing how such nodes move in the environment, ranging from Lempel-Ziv and Markov models to those mimicking working day mobility [17]. The emulation platform takes a routine that generates next-locations for mobile nodes in the system. This routine should implement an appropriate mobility model and return the expected location of a mobile node for a given time-step. These locations can be represented in terms of frequented destinations (such as home, internet kiosk, library etc) or geographic coordinates. Nodes encounter each other when they share the same destination or come within a specified distance during some customizable time-step. In many cases, a portion of the participating nodes will be mobile while others are stationary.

Bringing network and mobility emulation together, a Vivo emulation session consists of general purpose computers running live code, each emulating a node in the target environment. To monitor and coordinate participating nodes, Vivo uses an *emulation tracker*—a machine which uses a well-connected side channel for interacting with nodes in the system. The network characteristics between any two nodes are emulated by the underlying cluster, while mobility is emulated through staged encounters. The emulation tracker runs the mobility models for all nodes in the system, and for each discrete time-step during the emulation session, determines encountering nodes. To emulate node encounters, Vivo notifies the corresponding nodes, and enables them to communicate with each other over a separate LAN and exchange messages. Nodes can shape this communication channel to represent the characteristics of their encounter. Vivo comes with two built-in models for driving node mobility. One is based on a large scale study that tracked over 100,000 users for six months [45], and the other is based on recorded mobility traces from the CROWDAD collection [69].

The emulation tracker is also used for configuring and monitoring nodes. It is often the case that nodes have varying network connectivity with respect to other nodes in the network, as specified by the experimenter. However, the separate high-bandwidth LAN for communicating with the emulation tracker is maintained irrespective of the node's connectivity in the experiment. The tracker uses this channel to configure nodes, start applications, manage mobility and monitor the experiment session. Once a session is finished, the tracker collects runtime and message delivery information provided by participating nodes which can be used for aggregate analysis. Section 6.3 discusses how the evaluation platform interacts with systems built atop Vivo, and in the next chapter, the dissertation demonstrate how it enables researchers to reproducibly evaluate their design decisions.

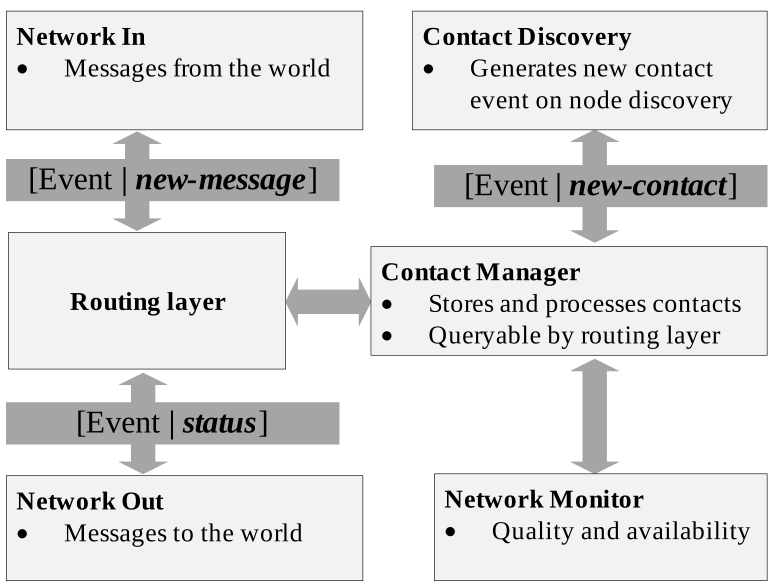


Figure 6.2: Summary of networking modules

6.3 Implementation

Vivo is implemented as a set of composable modules that communicate over defined interfaces. This section delves into some more detail by describing a subset of these modules and the interfaces they expose. The discussion is organized by groups of functionality. It will briefly describe a few modules per group, and provide a summary of events, interfaces and dynamic objects these modules use. These dynamic objects allow researchers to extend interfaces by passing additional data and code without changing the call signature.

6.3.1 Networking modules

This group of modules implements network related functionalities for data delivery systems, summarized in Figure 6.2. Vivo puts emphasis on isolating functionality to simplify module composition. For instance, the **network monitor** module provides a set of tools for characterizing network connectivity to other nodes in the system. As such, it can respond to queries in regards to network availability and quality. However, this module does not have memory, and relies on other modules to organize and store the aggregated network related information.

To assist in organizing known information about other nodes in the system, researchers can utilize the **contact manager** module. This module makes the contact information for any known node readily accessible through an interface. It registers to the *new-contact* event which is raised by the **contact discovery** module when a new node is discovered. This might be caused by other events in the system, such as encountering a node, or receiving a message from a previously unknown node. The contact discovery node has no memory, and storing contact information is left to the contact manager. The contact manager can utilize the network monitor to associate network quality with nodes. This chapter provides an example implementation for managing contacts using an OceanStore-style attenuated Bloom filter [103], which can efficiently respond to membership queries about known contacts.

Dynamic Object	Description
node-info	An object used for capturing a node's identity, such as the alias and network address
network-quality-info	An object for capturing the network characteristics of a remote node
Event name	**Callback signature**
new-contact	void *NC-handler*(node-info)
new-message	void *NM-handler*(message, node-info)
message-status	void *MS-notify*(message-id, status)
Synchronous interfaces	**Description**
void *ship-message*(message,node-info,status-handler)	Send a message through the network
network-quality-info *measure-network* (node-info)	Measure the network connectivity of a remote node
network-quality-info *get-network-info* (alias)	Get network quality information for a measured node

Table 6.1: Dynamic objects, callbacks and interface exposed by the networking modules

Two separate modules exist for managing network transfers. The **network out** module receives a message from the routing layer, and forwards it to the requested contact over the network. The routing layer can register a callback to be notified about the status of network transfers. An example implementation of a network queue for handling failures through prioritized retries is provided. The **network in** module, on the other hand, receives messages from the network, and passes them to the routing layer. A module interested in learning about new network messages would register a callback for the *new-message* event from this module. This module also triggers the contact discovery module to record potentially new contacts for nodes that are discovered through the network.

Table 6.1 summarizes events, interfaces and dynamic objects used to communicate across these modules.

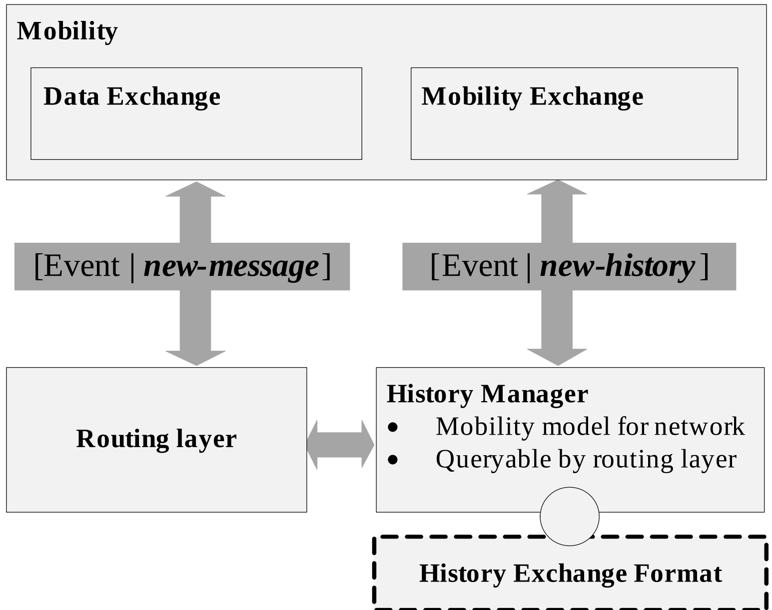


Figure 6.3: Summary of mobility modules

6.3.2 Mobility modules

Node mobility provides an important mechanism for moving data in challenged networks. This group of modules, summarized in Figure 6.3, provides tools for assisting in mobility management. For example, the **history exchange** module facilitates exchanging mobility histories when two nodes encounter each other. It also generates a *new-history* event to notify other modules. The history exchange module does not keep track of mobility history itself and defers mobility modeling to other components.

A routing protocol that utilizes mobility for data delivery often needs to work with a mobility model. The **mobility history manager** is an ideal module to implement such models. Using mobility history observations from the history exchange module, the researcher can provide a custom mobility model for use by the routing layer. This module supports an interface for passing mobility information using a dynamic object that can be tailored as needed. An example implementation builds on the previously mentioned study of human mobility [45].

On the other hand, the **data exchange** module facilitates message exchange when nodes encounter each other. When messages are received from another node during an encounter, this module generates a *new-message* event that a routing module might register to. In addition, it provides a simple interface for transferring messages to another node through mobility. In Table 6.2 summarizes the new interfaces and events that are used in this group of modules.

6.3.3 Routing modules

The last set of examples discussed deals with routing messages. These modules interact with the rest of the system in forwarding messages, summarized in Figure 6.4. Messages that are delivered to the node are processed by the **incoming message processor**. This module can subscribe to the *new-message* event to capture incoming messages. The module has two specialized functions: parsing incoming messages according to the message encoding format, and invoking the message acknowledgment routine for the system. The parsed message is then passed to the node buffer.

When end-users insert messages for delivery, the **new message handler** encodes them according to a specified message encoding format. The module provides an example message encoding format based on RFC 2045 [40]. In addition, the module often interacts with tools in the data processing group of modules for functionalities such as encryption and hashing. At the other end, the **outgoing message processor** module continuously monitors the message buffer for messages that need to be forwarded along, and makes decisions on how to forward them. It relies on hooks into the networking, mobility and buffer modules for making its routing decisions. In turn, this module can be queried by other modules (such as analytics) for message passing statistics.

New interfaces and dynamic objects used by these modules are summarized in Table 6.2. As mentioneded in section 6.2.2, there are additional groups of functionality in the toolkit, including buffer management and runtime operations (configuration, startup and analysis). The interested reader is referred to the system documentation for a discussion of

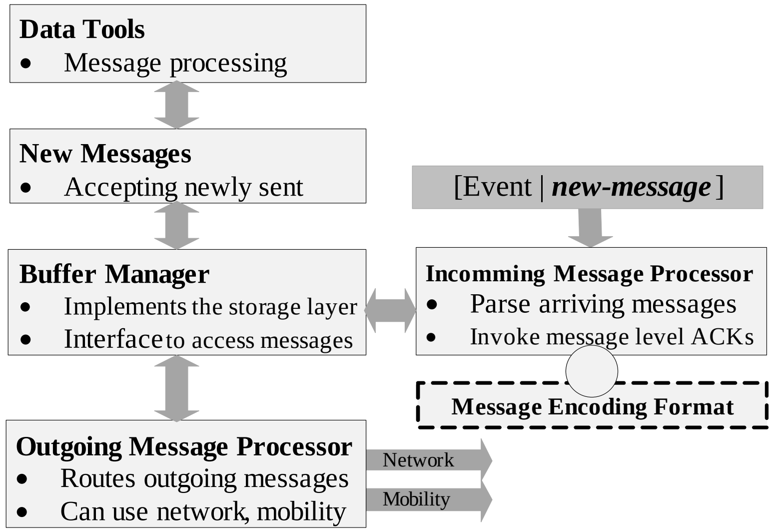


Figure 6.4: Summary of the routing layer

these modules and a complete reference.

6.3.4 Emulation adapter

To take advantage of the evaluation platform, a system needs to support a set of interfaces for responding to communications from the emulation tracker. The network communication between nodes is shaped by the underlying emulation framework. On the other hand, the tracker notifies nodes when an encounter happens between a pair of them, prompting them to communicate directly. An example **emulation adapter** module provides interfaces for accepting notifications from the tracker and exchanging messages upon an emulated encounter. Messages are often bundled with the routing information the system will need in forwarding them to their destination. Researchers are free to customize the contents of these bundles. One option is to use a standard format that has been proposed for bundle switched networks [109]. The basic operations on message bundles are:

```
SendBundles([b1, b2,...])
RecvBundles([b1, b2,...])
```

Dynamic Object	Description
mob-history	An object for passing the mobility history of nodes. Systems can customize the exchange
mob-prediction-object	Captures information about expected future mobility such as near-future encounters.
message	Captures the content and meta-data of a message within the system. Customizable by the system.
Event name	**Callback signature**
new-history	void *NH-manager*(alias, mobility-history)
Synchronous interfaces	**Description**
void *ship-message*(message,node-info,status-handler)	Send a message through using mobility
mobility-pred-object *get-prediction*(alias)	Request information about expected future mobility of the node
void *add-message*(message, kind)	Adds a message to the buffer. Describes whether it is outgoing or incoming.
messages *get-outgoing-messages*( )	Used to access a list of outgoing messages from the buffer.

Table 6.2: Dynamic objects, callbacks and interfaces exposed by the mobility and routing modules

In addition, participating nodes can collect runtime data which will be aggregated for system-wide analysis. The emulation tracker attempts to invoke `CollectInfo()` for each node at the end of an emulation session. Researchers can take advantage of the runtime analysis modules in the toolkit to collect such data. The evaluation platform provides tools to assist in analyzing emulation results collected from nodes. These tools can be used to generate statistics such as delivery rates, number of hops, transfer duration and buffer utilization. These can be augmented with custom analysis for processing the aggregated data.

6.4 Implementing systems with Vivo

The evaluation of Vivo attempts to convey two basic points about the platform: ease of system implementation and enabling reproducible experiments. As such, the evaluation is a mix of demonstrations and case studies that highlight the utility Vivo brings to researchers. This chapter discusses the implementation of two previously proposed systems using Vivo. It will describe how these systems take advantage of existing components in Vivo, and make their own modifications towards a live implementation. In the next chapter, a more detailed discussion will be provided for a new hybrid overlay network for bulk data that was built with Vivo. This will also demonstrate how Vivo enables researchers to evaluate the impact of their design decisions by reproducibly analyzing the three systems with it.

6.4.0.1 An epidemic routing network

One simple data delivery network implemented on Vivo is based on an epidemic routing protocol. Epidemic algorithms work by performing a pair-wise exchange of information as nodes encounter each other in a network. Several flavors of epidemic forwarding have been suggested for ad-hoc networks [125]. Nodes buffer messages even if there is no contemporaneous path available to the destination. When nodes meet each other, they exchange a summary of the messages kept by each node. A message is requested from the other node if it has not been previously seen by the current node, and if there is buffer space available. Epidemic networks eventually deliver messages to their destination by spreading them through the network a pair at a time. This network was implemented using a subset of the components in the Vivo toolkit. The system does not need the network modules to operate, but modifies the routing layer to reflect the epidemic algorithm. The mobility modules in the system deal with message exchange with encountered nodes. As each forwarding decision is made on a per-encounter basis, there is no need for an explicit mobility model. The system uses a FIFO queue to manage the buffer at each node, which is provided by buffer management modules in Vivo. Whenever a new message arrives to a full queue, the message that has stayed the longest in the queue is dropped. The implementation of the epidemic routing network can be used as a baseline for more complex ad-hoc networks. The system has a simple routing algorithm, and the effort to put together a live implementation using Vivo reflects that. Yet, the system can be run natively on nodes, and be evaluated through emulation.

6.4.0.2 A modified probabilistic routing network

The second delivery network implemented is based on probabilistic routing of messages that relies on the non-random characteristics of individual mobility to improve delivery rates [80, 131]. This network defines a delivery predictability metric for use in data forwarding. For each node and destination pair in the network, this metric indicates how likely the node will be able to deliver messages to the destination. The delivery predictability metric in the mobility history module is implemented, and it uses a dynamic object to pass the metric to the routing layer. Most of the effort in building this system was in implementing its routing protocol.

The routing layer is modified to reflect the probabilistic transmission algorithm. When two nodes encounter and the corresponding event is triggered, the routing modules use the message exchange component for transferring messages. A message is copied to a neighboring node only if the delivery predictability of the message's destination is higher at the recipient node. In addition, the nodes exchange a summary of their delivery predictability information using the history exchange module. This is used to appropriately update the metric. Each node starts out having the same delivery predictability to all other nodes in the network. As nodes encounter each other, this value is updated to reflect the frequency of encounters, the duration between encounters as well as the transitive nature of delivery.

To make the discussion more concrete, Listing 1 shows a pseudo-code of the changes in the mobility and routing modules, with emphasis on interaction to the rest of the system through Vivo's interfaces. If two nodes, A and B, encountered each other.

Listing 1 Implementing a probabilistic routing system

Receive *new-contact* event
    Triggers mobility, networking module
Mobility module:
    Wait for *new-history* event
    Obtain *mob-history* dynamic object
    Update delivery predictability (P):
        $P(A,B)_{new} \leftarrow P(A,B) + (1 - P(A,B)) \times \alpha$
        foreach x B knows about:
            $P(A,x)_{new} \leftarrow P(A,x) + (1 - P(A,x)) \times$
                $P(A,B) \times P(B,x) \times \beta$
        comment: α and β are empirically set
Routing module:
Outgoing processor
    Invoke *get-mob-prediction*
        Uses mobility history module
        Returns *mob-prediction-object*
    If $P(A, dest) < P(B, dest)$:
        Invoke *ship-envelope*
            Uses message-exchange module
Incoming processor
    Receive *new-message* event
        Contains *message* dynamic object
    Parse using encoding format
        Uses message handler module
    Invoke *add-message*
        Uses the buffer-manager module

With only a few hundred lines of code, building this system with Vivo required an order of magnitude less code than the previously reported prototype, while allowing the builder to focus on the new ideas in the system. Furthermore, the Vivo architecture simplifies adding new components. To demonstrate the ease of modifying existing systems to support new features with Vivo, the network adds a new module for recommending routes to nodes. The idea behind this approach is that, as the system analyzes node mobility patterns and frequented locations, it can make route recommendations that are tailored to the node. The goal of such a recommendation would be to encourage participants to visit certain locations which improve the overall delivery rate. In the common case, a participant is not expected to follow these recommendations. Therefore, the challenge is ensuring the routing algorithm works well in the common case, but benefits when individuals do follow these suggestions. The recommendation engine is implemented as a separate module that obtains mobility information from the system, and provides route recommendations to the routing module. The composable architecture of Vivo simplifies such designs.

6.5 Discussion

When designing systems for challenged networks, it is important to experiment with a variety of designs and evaluate them under a variety of circumstances. This chapter discussed Vivo, an experimentation platform for systems in challenged networks. Vivo consists of a toolkit of modules that interact over well-defined interfaces alongside an evaluation platform for experimenting with systems built atop this toolkit. The evaluation platform in Vivo provides a combination of network and mobility emulation. While these approaches are designed to simplify the iterative and reproducible evaluations of systems for challenged networks, researchers will need to consider the risks in relying on models for representing real-life environments. The network and mobility emulation mechanisms deployed in Vivo are based on previous studies of network experimentation and human mobility. It is important to understand the tradeoffs in these approaches, and determine if they provide the desired level of fidelity in representing the experimentation environment.

The EmuLab evaluation platform, which uses off-the-shelf PCs and a general purpose operating system for network emulation, has been verified along two dimensions [130]. The first was ensuring machines are able to keep pace when emulated links are operating at full speed. Several capacity tests have shown EmuLab's ability to easily support high bandwidth links on large clusters. On the other hand, the accuracy and precision of emulations was verified through a series of experiments using a representative range of delay, bandwidth and packet loss rate values, coupled with high packet rates. These experiments consider both large and small packets. EmuLab has also been verified against a wide area network using macro benchmarks that were run on live internet nodes. Therefore, for many practical purposes, EmuLab can provide a high fidelity network emulation framework. Nonetheless, the Vivo emulation platform is independent of the underlying cluster, and researchers can provide their own emulation cluster with the desired degree of fidelity.

Mobility emulation is the other important component when considering the faithfulness of the evaluation platform. The primary mobility model provided with the emulation platform is one based on a large scale human mobility study that looked at the patterns for over a 100,000 individuals over a period of six months [45]. The underling mobility routines discovered from this study reaffirm previous lessons in human mobility [116]. In addition, subsequent analysis of the same dataset with population segmentation for various regions, age groups, genders etc. have shown that the models used in the emulation platform hold for a wide range of people [115]. However, the models provided in Vivo might not be sufficient for all needs. Various regions and usage contexts might require other mobility models that are more faithful to the system participants. Researchers are able to provide their own mobility models by designing them as routines that generate mobility patterns for the emulation tracker. To the extent human mobility patterns could be captured with a mobility model, Vivo can be used to recreate such mobility within experiments. However, researchers should be aware that it is not always possible to completely capture all elements of human mobility with a model, and should consider field trials if complete faithfulness is desired.

The following chapter will further demonstrate the utility Vivo brings to researchers—and the choices researchers are able to make—by discussing a new hybrid overlay network that was built atop the Vivo platform. This network combines individual mobility and diverse network connectivity in delivering data in challenged networks. The discussion will also cover how the network was evaluated using the Vivo platform, and further clarify the network and mobility emulation considerations in experimenting with such systems for challenged networks.

CHAPTER VII

A hybrid overlay network for bulk data

7.1 Introduction

The previous chapter discussed Vivo, a platform for building and experimenting with systems for challenged networks. This chapter looks at a system for transferring bulk data that was built atop Vivo. This system is called *Bati*. Bati is a hybrid overlay network that delivers bulk data by leveraging two complimentary mechanisms—the *natural mobility* of users along with the *diverse, intermittent connectivity* available to them as they move. Bati combines and expands on ideas from delay-tolerant and ad hoc networking, with an eye towards scalability in the face of highly uncertain endpoints and the movements and connectivity between them. Data can travel with moving principals using excess storage, or through clusters of well-connected machines; weak connectivity is used for control plane activity, providing efficient resource management. Bati, requires no global state, no message or acknowledgement flooding, and can tolerate imperfect predictions in individual nodes' destinations, arrival times, and arrival frequencies.

The central observation in Bati is that principals have a small number of locations that they frequently visit. Each mobile node tracks its K most popular destinations, and the expected inter-arrival time between them. To compute the distance between two destinations, the total flow of principals between them is combined. However, Bati limits the state required to do so by remembering only the most frequent destinations in combination precisely, and approximating the remainder with a conservative estimate of infrequent destinations. This preserves the scalability of the system, devoting resources to the most likely routing alternatives.

This mobility model of “distance" is augmented by recognizing that even within the developing world, clusters of good connectivity exist and local connectivity often far exceeds remote connectivity. Routing to any destination within a cluster is equivalent to reaching all of them. In turn, data can be *staged* within specific nodes in a cluster to take advantage of mobility patterns that are not precisely rooted at the origin node, but that can be reached easily by intra-cluster means. A single metric combines the contributions of mobility and network connectivity for routing decisions.

Because delivery is uncertain, Bati allows multiple packets in flight to any particular destination. Rather than try to precisely notify all nodes of delivery, Bati employs an inverted-ACK technique. Data starts with a soft timeout, which is either checked or (in the absence of any connectivity) extended as necessary. These control-plane operations can be done even in resource-poor networks typically found in the developing world with only a nominal overhead.

Bati was implemented as a multithreaded process that utilizes various components in the Vivo toolkit. Bati, alongside the other two systems discussed in the previous chapter, was evaluated using Vivo’s emulation platform. In addition to evaluating design alternatives within Bati, its evaluation is used to demonstrate Vivo’s capacity to facilitate a repeatable evaluation of live implementations. When researchers evaluate their system with emulation, they observe the impact of design decisions at various layers—spanning networking, systems, data processing, routing etc—allowing them to experiment with issues at various levels. The experiments discussed here were set up on the EmuLab platform.

7.2 Design

For an outside observer who is not aware of an individual's motivation and schedule, human mobility can easily appear to be random and unpredictable. Yet, despite the deep-rooted desire for spontaneity and change, human daily mobility can be characterized by regularity. The success of a Bati depends on marrying the strengths of a diverse set of network links with a sufficiently accurate model for natural human mobility. Luckily, scientific analysis shows true randomness in human behavior to be a rather infrequent phenomenon [115]. As to the diverse set of available network links, Bati shows a broad classification of capacity can go a long way in establishing an efficient division of *purpose*.

Bati uses an individual mobility model to represent how principals move among nodes, and incrementally update a distributed, system-level transition model that represents how close nodes are connected with each other as a result of mobility. All available network connectivity is also utilized in delivering data. Some links might be best suited for shipping data, while others might be more appropriate in orchestrating roles. The following sections describe how Bati parses mobility, builds an efficient and distributed transition model, and exploits link diversity in delivering data.

7.2.1 Individual Mobility

People are creatures of habit and often have certain repeated patterns that can be learned probabilistically. Several projects use Markov models for mobility prediction [22, 116]. While a second order Markov model has been shown sufficient for predicting a principal's short term mobility, it often falls short when considering more than a few steps in the future [91]. In particular, Bati is more concerned with where a principal is going to be in a day—or a week—than in the next few minutes. As a result, a Markov model does not quite capture the information needed for use in this system.

Instead, Bati uses results from a large scale study of individual mobility patterns that looked at a trajectory of more than 100,000 anonymized mobile phone users whose location was tracked for a six-month period [45]. The study finds that human trajectories show a high degree of temporal and spatial regularity, each individual being characterized by a time-independent distinctive travel distance and a significant probability of return to a few highly frequented locations. In particular, the cumulative return probability of an individual to a previous location is characterized by several peaks at 24, 48 and 72 hours. In addition, the probability of finding a user a location with rank L, where L represents the L^{th} most visited location for an individual, is well approximated by $P(L) \sim \frac{1}{L}$, independent of the number of locations visited.

The model starts out by keeping track of the average interarrival time for the top-K nodes a principal visits, and how long it has been since the last visit. The interarrival time is defined as the time it took for a principal to get back to a node when at least one other node was visited in the interim. This gives us the notion of nodes a principal is likely to visit soon. The model combines this with the observed standard deviation of the interarrival time to bound the likelihood of a visit to a node in a given time window as follows: when a principal visits a node, it orders the top K other nodes in the principal's history based on the latest estimated time the principal is going to visit the nodes with a threshold probability P. Intuitively, as the principal moves in the system, the nodes it is likely to visit soon start bubbling up to the top of the list. For each node in the principal's top-K list, the model provides:

$$M[n] = \{\mu_t, \sigma_t, T, C\} \tag{7.1}$$

Where μ_t is the average interarrival time, σ_t is the standard deviation and T is time of last visit. C is the confidence score that describes how similar the recent trend has been to the overall pattern. It is measured by recording how many of the last N observed visits occurred within the estimated bound.

The model has been adapted to deal with phase changes and new patterns. If a principal visits a node before the due time, the average interarrival time and the standard deviation are affected accordingly. On the other hand, if the principal is significantly past-due to a node, it is reflected in the system by extending the average expected time to the node by a multiple of the standard deviation. This extension gets incrementally bigger to reflect more permanent changes in patterns. The confidence score is used to gauge how much the parameters are altered: the lower the confidence score, the more drastically the parameters are changed, allowing for quick learning. While it might be useful to add route fingerprints to deal with *multiple, established* sub-patterns of a principal with respect to a single node, the empirical analysis in section 7.3 suggests that the current model is sufficient for capturing mobility patterns in the system.

7.2.2 Collective Mobility

Another important piece in the system is the collective transition model that describes the *mobility connectedness* of a node to others in the system. As a node is visited by a number of principals, the transition model is incrementally updated to reflect the current state of the system.

Consider a principal Z that moves from node A to B with an average trip time $T_{(Z,A \to B)}$. The quantity $\delta_{(A \to B)}$ is defined to represent the closeness from A to B. $\delta_{(A \to B)}$ describes the *avergage, collective* trip time in going from A to B due to all principals that visit node A. For these purposes, the average trip time of a principal between two nodes can be sufficiently estimated from its interarrival times at the nodes without having to keep a separate record.

If there was only one principal in the system, then:

$$\delta_{(A \to B)} = T_{(Z,A \to B)} \tag{7.2}$$

Now imagine another principal, Y, that also moves between A and B with an average trip time $T_{(Y,A \to B)}$. It is important to update the relative closeness factor, $\delta_{(A \to B)}$ between A and B, to reflect the component contributed by the new principal.

This is equivalent to calculating the average waiting time for a bus between two stations when there are busses running at different intervals. If two busses service a station, and the first bus leaves from the station every n seconds while the second bus leaves every m seconds, it can be shown that the average waiting time, W, for a randomly arriving passenger is:

$$W = \frac{n \times m}{n + m}$$

Using the same approach, the closeness as a result of two principals Z and Y, moving between A and B is given as:

$$\delta_{(A \to B)} = \frac{T_{(Z,A \to B)} \times T_{(Y,A \to B)}}{T_{(Z,A \to B)} + T_{(Y,A \to B)}} \tag{7.3}$$

This can be generalized to incrementally update the relative closeness between any two nodes as the system learns more about principal mobility in the system. Given the old closeness between two nodes as $\delta_{(old)}$, if a principal's expected trip time was adjusted from $T_{(old)}$ to $T_{(new)}$, the resulting new collective closeness as a result of this update, $\delta_{(new)}$, can be shown to be:

$$\delta_{(new)} = \frac{1}{\frac{1}{\delta_{(old)}} + \frac{1}{T_{(new)}} - \frac{1}{T_{(old)}}} \tag{7.4}$$

This allows for a simple way to gauge and update closeness between nodes as principals move in the system, and study their mobility patterns. Closeness is calculated in a distributed manner where each node is responsible for maintaining part of the transition model that describes how connected the node is to its top-K closest nodes in the system. K is by default 25, but could be configured according to space availability. To keep track of the long tail of not-so-well-connected nodes, Bati uses attenuated Bloom filters, in a manner similar to OceanStore [103]. This has an n-width array of standard Bloom filters, where the n^{th} filter represents an n^{th} level of connectedness – giving us a reference-table like property. Since these filters are used to represent nodes in the tail that are not well connected to the current node, the small false positive rate from the Bloom filter is easily outweighed by the space savings.

Since nodes cannot be expected to always be available for updating and querying their slice of the transition model, principals are used as carriers of not only data, but also system transition information. This is possible because a principal visits only a small fraction of all nodes in the system. As a principal visits one of its top-N frequented nodes, it makes a copy of the most recent slice of the transition model kept at the node. If an update to a node is not possible because the node is currently offline, the update is kept on the principal, and merged later when the principal visits the node. On the other hand, when making routing decisions, the latest copy carried by the principal is used. This copy will be slightly out of date, but the freshness of the model is proportional to the frequency at which the principal visits the node in question. As a result, the most recent copies will be from nodes the principal visits often, and will probably visit soon as well, while the most stale copies will be from less frequented nodes—a desirable property.

7.2.3 Network usage

While bandwidth is scarce in developing regions, not all nodes in the system are poorly connected to the network, and local connectivity usually far exceeds remote connectivity. In addition, most nodes have a low bandwidth, high latency link that can be selectively used when available. Bati can utilize all available connectivity to aid in the delivery of messages in this hybrid network. In particular, when available, low bandwidth links are used for maintaining more up to date routing information at nodes, while the occasional fat pipe is used for routing shortcuts and last-mile delivery.

Bati uses weak links in two ways. First, a node can subscribe for remote network updates about changes in a principal's mobility patterns, subject to network capacity. This subscription is tiered so that the weaker a node's link, the fewer network updates it will receive. This is an optimization, as updates can always be merged when the principal revisits the node. Second, weak links are used to opportunistically acknowledge envelopes.

Since network resources are not always available or reliable, synchronous acknowledgment of envelopes is not practical, and should not be required for correctness. Instead, Bati uses opportunistic, collective, and asynchronous ACKs for space optimization, as an acknowledged envelope no longer needs to be kept in the buffer or be transmitted to future principals. From the node's standpoint, an envelope is kept in the buffer until it is ACKed or eventually kicked out by envelopes that are more likely to be delivered.

The fundamental difficulty in ACKing in networks like Bati is route multiplicity. Traditional ACKs, where nodes acknowledge envelopes as they receive them, do not work well here because a node cannot truly ACK an envelope without flooding the system, as more principals with extra copies could keep showing up. However, it is important to stop the propagation of already delivered messages to conserve resources. Bati uses *inverted ACKing*, a simple and resource customizable strategy targeted at decentralized delivery systems. When envelopes are inserted in the system, they are given a time-till-acknowledgment (TTA), which can be based on average delivery time of envelopes. As long as an envelope is encountered within its TTA, it is simply stored and forwarded. As the TTAs for envelopes at a node starts to expire, they are batched for collective and opportunistic ACK requests from their destinations. An ACK for an envelope consists of its unique ID, 16 bytes in size. If an envelope has not been delivered, or an ACK is not possible due to network failures, the TTA is extended up to a limit. While this approach allows some envelopes to continue propagating for some time even after delivery, it trades network usage for local storage—an advantageous exchange in most developing-world environments.

Some nodes in the hybrid network have better connectivity than others. When good connectivity is available, it is used for establishing route shortcuts through *data staging*, and for last-mile delivery of envelopes to their destination. If an envelope makes it to an intermediate node that is well connected to the final destination, the network, rather than mobility, is used to deliver the envelope.

Data staging positions data envelopes at other, well-connected nodes, in order to gain from expected principal mobility in the system. As a simple example, imagine two nodes A and B that are well connected to each other, and a third node C that is in a challenged network environment but 'close' to B due to principal mobility. Now, imagine there was a data envelope originating from A and addressed to node C. In this case, it would be beneficial to use the well connected network to stage the envelope from A at B, so that it can take advantage of the likely path to C. Bati accomplishes this using *clusters* of nodes that can communicate efficiently though the network and *hot links*—strong mobility connectedness in the transition model.

7.2.4 Routing

When a node encounters a new envelope, the available network resources are examined to determine if the envelope can be directly delivered or gainfully staged as described above. All remaining envelopes are routed using the mobility of individuals in the system. Given the information from the node's collective mobility model, and the mobility models available from the principal, a node determines an ordered list of envelopes to be carried by the principal as follows:

Step 1: The nodes a principal is expected to visit before coming back to the current node are classified into groups based on how soon the principal is expected to visit the respective node, as given by its mobility model. Group formation ensures that the further out the estimated time of arrival is, a larger set of nodes will belong in the group.

Step 2: A majority of the data capsule's storage space is provisioned equally among the node groups. This equal division ensures that the envelopes that will most benefit from going to the nodes a principal is going to visit sooner will have more real estate on the capsule than envelopes destined for subsequent nodes. A node might be bumped to the next group if the confidence for its estimated time of arrival is below a threshold.

Step 3: For each group, the node determines, of all the remaining envelopes, those that would most benefit from going to any of the nodes in the group. This means, the destination of the envelope needs to be closer at one of the nodes in the group than at the current node. Within this set, envelopes are ordered based on how close they would get to their destination. Any unused capsule space from each group is appended to the next group's slice.

Step 4: The remaining space in the data capsule is filled with a randomized set from the left over envelopes. This is to account for the uncertainty in the mobility of the principal, as well as in the collective mobility models, and benefit from it.

7.3 Evaluating systems with Vivo

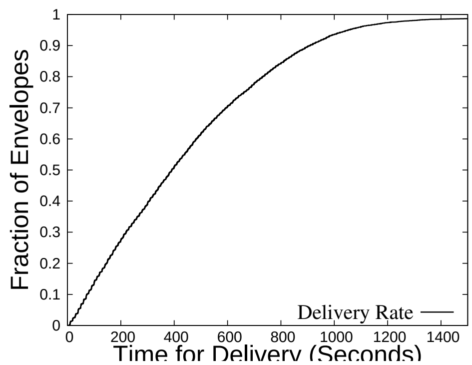
Bati was implemented as a multithreaded, cross-platform user level library with multiple components that were modified from the Vivo system toolkit. This section takes a look at the two systems implemented earlier—an epidemic network and a probabilistic routing network—as well as Bati. The Vivo emulation platform is used to compare these systems, and evaluate design alternatives within Bati. The experiments discussed here were set up on the EmuLab platform. It is also fairly straightforward to configure other managed clusters for Vivo.

The experiment had 40 nodes for use by the system, about half of them representing mobile principals. To guide mobility, a built-in probabilistic model based on the temporal and spatial locality of human mobility [45] is used. This model has been shown to give good estimates regardless of age, gender, population density or rural versus urban environments [115]. In the experiment, each node had between 2 and 4 frequently encountered nodes driven by this mobility model. There are three broad classes of network connectivity in the experiments. Half of the nodes have no network connectivity, while the remaining have some connectivity. Out of these, most are connected poorly over a 15Kbps link with 300ms latency. A smaller fraction has good connectivity with 1.5Mbps bandwidth and 60ms of latency. These numbers are derived from Akamai's state of the internet report for global connectivity [8].

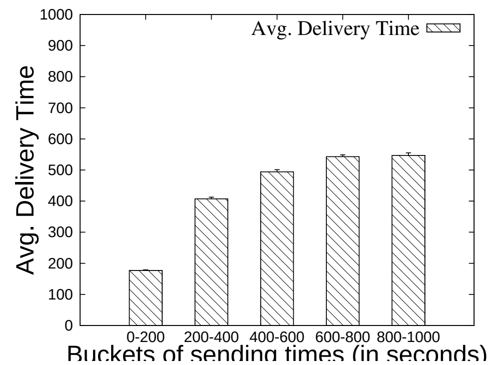
For an individual experiment, the system is started and allowed to run for 500 seconds. After that, for the next 1000 seconds, each participant sends two messages to two randomly selected destinations every 10 seconds. The system is then allowed to run for another 1000 seconds before it's terminated and information is collected. These represent some of the configurations possible with the Vivo evaluation platform. Each of the experiments was run four times, and this section reports on the aggregate analysis. Experiment reproducibility is one of the key benefits in using Vivo. At the same time, emulation exercises the entire stack of the implemented system, helping researchers understand the full impact of their design decisions, and lending some reality to evaluations over pure simulation.

7.3.0.1 Base run

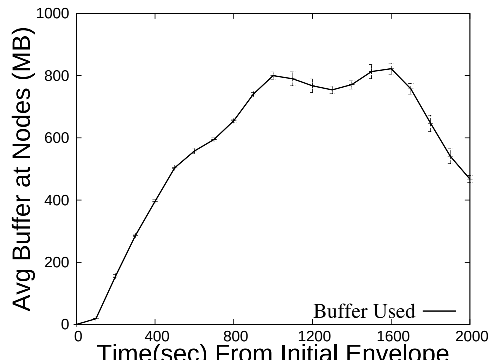
For comparison against different configurations, a base run is established for Bati's performance. For this run, 1MB messages are sent, with the buffer for mobile nodes at 100MB. 20% of nodes in the evaluation are configured to have good connectivity. The collective ACKing interval is set to be 400 seconds. Figure 7.1 exhibits the analysis for the base run. The CDF shows more than 50% the enevelopes were delivered within 500 seconds, and more than 90% with in 1000 seconds. Figure 7.1(b) shows the average delivery time for envelopes sent earlier in the run is shorter than those later, and plateaus to around 500 seconds for later envelopes. This is because buffer space is more plentiful earlier in the run. Figure 7.1(c) shows the average buffer used at stationary nodes, taken every 100 seconds at all nodes. Buffer used at the nodes also plateaus after about mid way through the run, and declines afterwards. As envelopes get delivered in the system and are ACKed, they are kicked out of non-destination nodes, which reduces the buffer footprint. The number of network hops made by envelopes is proportional to the availability of good connectivity while mobility hops are determined by the individual movement patterns and available principal buffer space.



(a)



(b)



(c)

Figure 7.1: A base run for evaluation experiments

7.3.1 Comparing systems

The case study begins by comparing the performance of Bati with the epidemic and probabilistic networks. The experiment sends messages of size 1MB in a system where mobile nodes have storage capacity of 100MB each. This comparison configures 20% of nodes in the evaluation to have good connectivity, as described above. Figure 7.2 plots the cumulative delivery rate of messages over time in each of the three networks. As expected, epidemic routing lags in delivery rate of messages. The hybrid network is able to leverage both network connectivity and individual mobility, allowing it to deliver about 40% more messages compared to probabilistic networking under the same scenarios.

The evaluation then compares Bati to using the available network links efficiently, with in-network storage for delay tolerance. Considering the amount of bytes transferred from delivering the envelopes in 2000 seconds of the base run with a 1MB envelope size, it takes

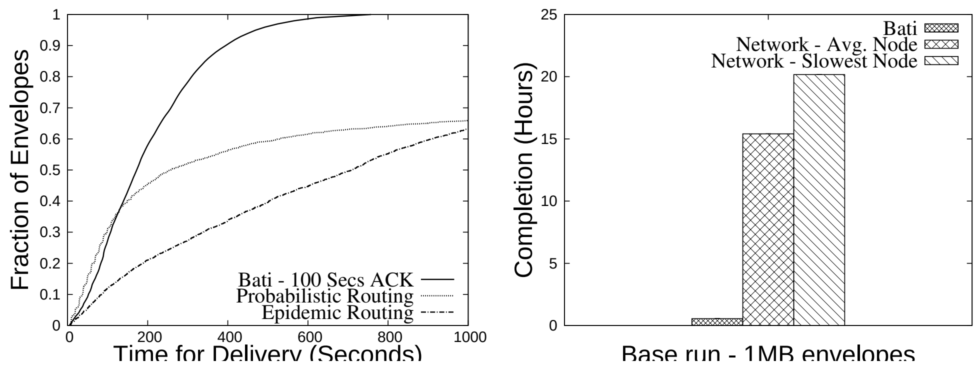


Figure 7.2: A three system comparison of envelop delivery

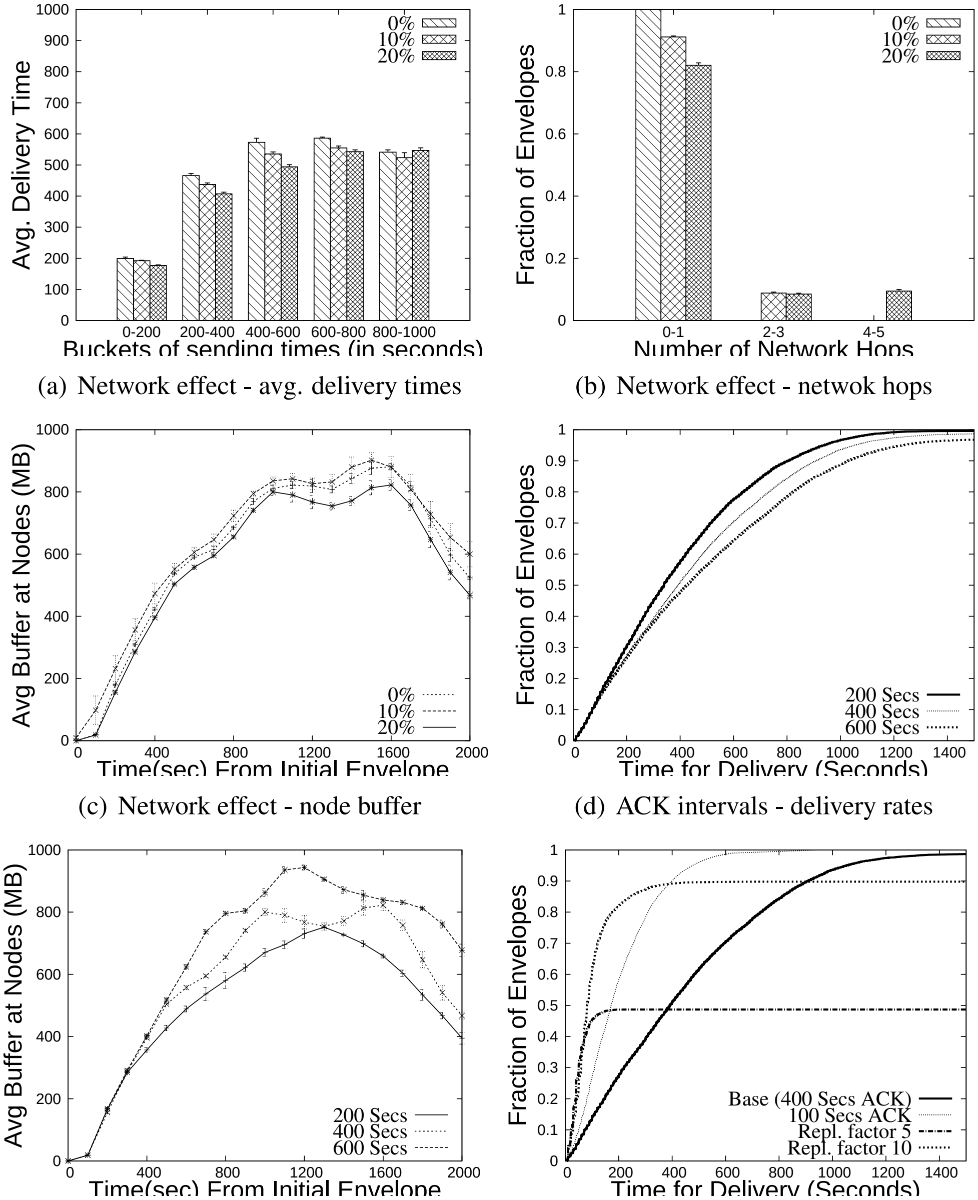
the average node upwards of 15 hours to accomplish the same using the network alone. A weakly connected node takes slightly over 20 hours to deliver envelopes sent from it during the run. The savings with using Bati are even higher (nearly two orders of magnitude) for a 5MB envelope run, where Bati's ability to leverage unstructured mobility serves it well. Figure 7.2 summarizes the results.

7.3.2 System characterizations

This section will investigate the impact of various research decisions in Bati. This will also demonstrate how Vivo's emulation platform simplifies the process of analyzing design decisions.

7.3.2.1 The effect of network penetration

In order to gauge the effect of good network penetration in the system, this section varies the percentage of nodes that are well connected in the system from 0% to 20%, while keeping everything else similar. The average number of mobility hops for an envelope decreases and the average number of network hops increases as the penetration rate increases, shown in figure 7.3(b). This is because more envelopes are able to use the network rather than mobility for delivery. Buffer space used at stationary nodes, shown in 7.3(c), also slightly reduces as connectivity increases because more envelopes are deliv-



(a) Network effect - avg. delivery times

(b) Network effect - netwok hops

(c) Network effect - node buffer

(d) ACK intervals - delivery rates

(e) ACK intervals - node buffer

(f) ACKs vs. replication limits

Figure 7.3: Effects of Network penetration and ACK intervals on performance

ered using the network, which takes a shorter time and allows nodes to kick out envelopes sooner. This also reduces the average delivery times of envelopes as shown in figure 7.3(a).

7.3.2.2 The effect of ACKing intervals

Another factor considered in the evaluation is different ACKing intervals. Leaving everything else similar in the base run, the experiment looks at 200 second, 400 second and 600 second ACKing intervals. As shown in figure 7.3(e), the most significant tradeoff between these runs is buffer space at nodes. The more frequent ACKs are, the less space that is used at nodes. In a similar fashion, less and less delivered envelopes are propagated through Bati with the increase in frequency of ACKs. This is reflected in the slightly higher delivery rate for the 200 second run over the others as seen in figure 7.3(d). The number of mobility hops also slightly increases with increased ACKing intervals. The number of network hops, however, is not affected very much because network delivered envelopes do not need a separate ACK anyway. The small increase in network hop count with higher ACKing intervals is due to envelopes that are delivered through mobility propagation to nodes with good network connections, which end up staging or delivering the envelopes through the network.

As described earlier, one of the design principles in Bati is to use collective ACKs and likelihood of delivery for buffer space management rather than a strict limit on hop-counts or replicas. Although this approach results in envelopes propagating for some time even after delivery, thereby using some buffer space, it delivers more envelopes in the long run. To illustrate this point, the experiment runs two simple variation of Bati where the number of replicates propagated by a node is initially set to 5 and 10. These approaches in fact have a quicker start in delivering messages, but also introduce an asymptote in the delivery rate. This asymptote depends on the size of the system, and how far envelopes have to travel to their destination. As a result, Bati opts for the more graceful scheme. On the other hand, with a 100 second ACKing interval, Bati achieves the quick start property, while still maintaining a high delivery rate. On average, the size of a collective ACK was 9.7KB for the 400 second interval and 3.7KB for the 100 second interval—representing 1.3% and 1.8% of a weakly connected node's available bandwidth respectively. Figure 7.3(f) compares the results.

7.3.3 Using recorded traces

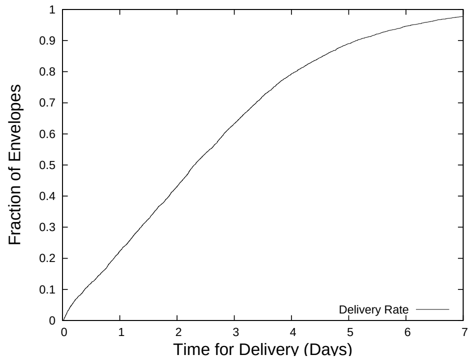
After evaluating Bati using Vivo's built-in probabilistic model to guide mobility, it was interesting to investigate as to how the system would operate under recoded mobility traces. Ideally, the two should be equivalent as the mobility model used is an abstraction of real world mobility. For recoded mobility, the CROWDAD mobility data from Dartmouth College [69] was used. The data was collected nearly continuously for two years, and has a record of more than 6000 traced users as they move about more than 500 nodes. Data collected for February of 2003, the last full month available in the dataset, was used because it potentially has the highest coverage of nodes and principals. There are more than 1700 active principals and more than 500 nodes visited during the month.

The experiment was setup as follows. For each run, there were 21 nodes picked randomly for which the principals that visit these nodes were considered. In order to reduce the resource consumptions of the emulation, two simplifying reductions were made. The emulation was limited to:

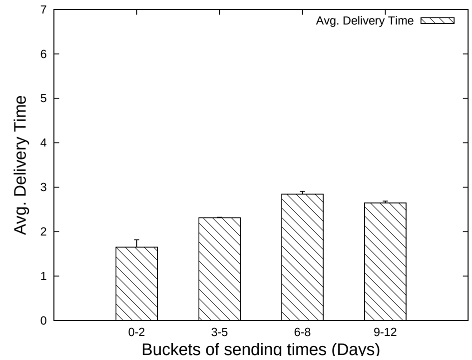
1. The top 30 mobile principals visiting the nodes
2. The first 500 steps of these principals

Both reductions work against the system because the more principals emulated, and the more they move around, the more chances there are for data transfer. The long tail of unemulated principals could have served as additional carriers in the network. The warm up, message sending and running portions of this experiment were divided in a similar ratio as the first set of experiments. As such, the system is initially run for a week, then for the next 12 days, every time a principal visits a new node, it sends two messages to randomly selected nodes. After that, the system is allowed to run for the rest of the trace. The experiment was run four times.

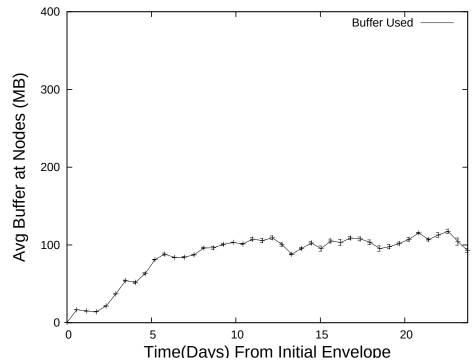
The results obtained are fairly equivalent to those in previous mobility model, as shown in figure 7.4. More than 50% of the envelopes were delivered within 3 days, and more than



(a)



(b)



(c)

Figure 7.4: Using recorded mobility traces for evaluating delivery

90% within a week. The average delivery time was between 2 and 3 days for the various clusters of sending times. Buffer and hop counts also have a predictable behavior as shown in figure 7.4(c).

7.4 Discussion

This chapter provided an in-depth discussion of Bati, a hybrid overlay network built atop the Vivo platform for challenged networks. The key insight in Bati is combining the unstructured and uncertain mobility of individuals with the diverse connectivity they encounter as they move in their environment. This chapter is also used as vehicle to demonstrate the utility Vivo brings to researchers in prototyping systems for challenged networks, and evaluating them in an iterative and reproducible manner. Bati builds on several prior projects and observations into repetitions in human behavior. Evaluations based on conservative estimates of available resources show significant savings in using Bati for delivering bulk data compared to using the network alone, while improving the deliver rates compared to systems that primarily rely on mobility for moving data.

This chapter concludes the discussion of tools and mechanisms for improving data access in challenged network environments. The next part of the dissertation turns to the market and service discovery considerations that are important in making these systems viable for developing-country use. Providing services like personal data aggregation carry monetary cost, and an efficient mechanism for trading these services is a requirement for scalable use. To facilitate a wide adoption of digital services, this dissertation introduces an extensible auction-based market platform for challenged network environments that can also be used more generally.

PART THREE

BUILDING SUSTAINABLE SERVICES

CHAPTER VIII

An extensible market platform for challenged networks

8.1 Introduction

If digital services targeted towards developing regions are going to be sustainable, there needs to be an efficient market mechanism for trading them. An open and competitive marketplace can bring enormous value for the consumer, while fostering innovation and growth across a wide variety of businesses. In the case of Sulula, for example, providing a simple and extensible way to allow users to compare offers from various providers, while transparently arranging for data transfer is ideally required. As a result, it would be useful to design a trading platform that is appropriate for these regions.

While electronic commerce has fundamentally changed how services and goods are traded in the developed world, its impacts are far and apart in developing countries. This is often due to low network penetration, lack of locally relevant markets, and requirements for additional facilities (such as credit cards, shipping arrangements etc.) to take advantage of such marketplaces. Even when these markets are established with local content and poor connectivity in mind, they are often specific to a certain domain or community. To mitigate this problem, and allow for wide adoption of digital services, this chapter proposes an extensible auction-based market platform for use in challenged network environments. Such a platform should enable developers to incorporate a market layer in their application and open their service to a wider audience, while considering the limited operating environment where communication channels are narrow and potentially expensive.

This chapter introduces *Robit*, an extensible auction-based market platform for operation in challenged environments. Robit enables developers to incorporate a market layer in their application, and open their service to a wider audience. Providers and consumers will benefit from the greater number of transaction options and product information brought by the added market structure, resulting in a social surplus increase [61, 118, 126]. Robit develops from experiences of building market oriented digital services for challenged network environments, and a survey of auction based marketplaces. The market platform uses a customizable auction mechanism based on a modified second-price auction structure. Robit is targeted towards challenged network environments, where communication channels are narrow and potentially expensive.

The user-facing communication channels in Robit are SMS and voicemail, leveraging the wide penetration of cellular telephones in developing counties. By the end of 2010, there were more than 5.3 billion mobile telephone subscribers around the world, with developing regions having the highest mobile growth rate [1]. In Ethiopia, for example, there are about twenty five times as many mobile subscribers compared to internet subscribers [36]. This allows us to take advantage of widely available services like SMS in the system design. In addition, voice based solutions, such as VioKiosks [5] and the Talking Book [108] have shown tremendous success in developing regions, as the barrier to participation is minimal.

To show how Robit can be used to add a market layer for services and goods, Sulula has been modified to use Robit for an auction-based marketplace. In addition, this chapter describes a standalone marketplace developed with Robit for buying and selling goods that takes advantage of the various features Robit provides. The next section will next briefly discuss the role Robit plays in these contexts.

8.1.1 Use case: digital applications

As discussed in Chapter V, Sulula takes into consideration the wide availability of cellular phones in developing countries and personal usage patterns. Rather than visit a kiosk and fetch data on demand—a tiresome process at best—users request a future visit at a kiosk using their SMS capable cell phones. Sulula provides the secure infrastructure to fetch a user's private data from a data source to the consumption point ahead of usage time. When the user arrives at a provisioned kiosk, she need only obtain the session key on-demand, and thereafter has instant access. By scheduling data downloads and uploads with respect to resource availability, Sulula enables kiosks to manage their resources well, while saving users tens of minutes of waiting time downloading private data.

Sulula, however, requires users to establish business relationships with particular kiosks. For example, if a user knows of three kiosks that she might be interested in getting her data at, she needs to:

- Physically register at each kiosk, and establish a trust relationship with each one
- Know the contact information and services provided by each kiosk
- Individually contact each kiosk to see if capacity exists, and the prices are reasonable

This process is inefficient at best. Because users have the burden of contacting and comparing offers from kiosks individually, on top of having to pre-register at each one, competition in the marketplace is hindered. In addition, this mechanism makes Sulula difficult to use when individuals are mobile and do not have a frequented 'home' kiosk—severely limiting its utility to travelers. Such users do not have pre-established business relationships with local kiosks of the visited area. Furthermore, Sulula requires each kiosk to be capable of having the facility to process and respond to SMS messages, which might require expensive gateways for busier kiosks. This can reduce the adoption rate by kiosks, and thereby reduce the number of service providers supporting the Sulula platform.

In the context of Sulula, Robit decouples the mechanism of providing service from the marketplace requirements of fostering competition and openness. As such, rather than users establishing business relationships with individual vendors, all communication is done through a central marketplace that facilitates the exchange of digital services. The marketplace serves as the convergence point where users and kiosk owners come together to trade, all done through low bandwidth communication and SMS messages.

Vendors register with the marketplace and provide identifying information such as location. When a user would like to schedule data delivery, she contacts only the marketplace, describing her preferred location and estimated time of arrival, using SMS. Robit then takes over the request, and conducts a dynamic (real-time) auction for service providers in the preferred location. Vendors are allowed to fine-tune their offerings, run promotional periods etc. The scheduling mechanism at kiosks ensures that a kiosk can in fact service a request. Once offers are collected from various vendors, they are filtered and forwarded to the user. The user can consider various factors, such as location, price and timeliness of service, in deciding which kiosk to use and then informs the marketplace. The winner of the auction is notified, and the session key required for prefetching data is securely transferred to the kiosk. The user then can head to the location at the described time, knowing her data is ready and waiting for her.

8.1.2 Use case: standalone market

Robit was also used to build a standalone auction-based market for buying and selling goods. The auction platform allows merchants to run auctions for their goods, increasing the exposure of their items and the efficiency of trade, even in the face of poor networking conditions. Robit makes a number of adjustments to the vanilla auction infrastructure to adapt to the limiting constraints present in these environments. In addition to a simple SMS interface that allows users to accomplish a number of marketplace activities, Robit is augmented by a voicemail gallery system that gives participants the opportunity for more descriptive communication about the goods and services offered in the marketplace. Furthermore, to mitigate the lack of wide and on-demand communication channels, Robit auctions are structured around a right of first refusal rather than a contract to buy. The auction structure of Robit will be discussed in section 8.2.

The preliminary user study in Addis Ababa, Ethiopia, suggests a potential need for such a service. It is particularly difficult to buy and sell used goods, as most of the trade happens through word-of-mouth or brokers who charge steep fees for their services of bringing buyers and sellers together. In addition, comparison shopping requires visiting markets in different areas and finding similar items. The initial user study and pilot deployment are discussed in section 8.4.

Robit provides a simple interface to its auction based digital marketplace, enabling buyers, merchants and vendors to exchange services and goods in a competitive environment. It is structured around the fundamental constraints present in challenged network environments, and leverages widely available tools to achieve its purposes. Robit can be used to make digital services like Sulula widely available to users without having to establish individual business relationships with all potential vendors. It can also be used to build an auction platform for exchanging other goods, and it provides a number of features and technologies that make the adoption easier in low-bandwidth, high latency environments.

8.2 Design

The main design principle behind Robit is providing an open and competitive marketplace by utilizing widely available communication channels in challenged environments. To this end, Robit limits the resources required to maintain and operate the infrastructure.

Robit features an extensible structure, allowing various additional services to plug on top of the basic auction infrastructure. To demonstrate this, this chapter will discuss the shim layer built on top of Sulula that allowed it to benefit from the platform provided by Robit. In addition, the various principles and mechanisms used in building a network challenged digital marketplace for use in developing regions are discussed.

To use the Robit infrastructures, digital services like Sulula provide small plugins at the market side as well as the end point, such as kiosks or business centers. The main principles in providing an auction platform for digital services are:

- Providing a transparent architecture to give users a simple way to purchase services
- Allowing for a real-time auction framework that can utilize up to date market information
- Operating without requiring users to have a pre-established business relationship with specific vendors once they are registered with the marketplace, and
- Fostering competition and openness

The market-side plugin for a digital service augments Robit with the service-specific logic to deal with requests. When requests are sent to the Robit listening component, they are forwarded to this plugin for further processing. In the case of Sulula, this layer checks with the data source that has data of interest for the user, and obtains meta information about available data for the user, such as the size of the data and a transaction key for further inquires. Using this information, the layer formulates solicitation for offers (SFOs) to pass to the marketplace. Once these requests are structured as SFOs, Robit contacts eligible registered vendors for their latest offers in providing service.

The dynamic bidding is handled at kiosks with the vendor-side plugin that provides the market functionalities to Sulula. This shim layer hooks into the Sulula stack to get the latest resource schedules and price offers set by the vendor, and responds to the SFO. A response to an SFO includes the price a kiosk is willing to accept for the data size specified, as well as the latest time it is able to service the requests. This information is collected from a number of participating vendors.

Upon receiving these offers, Robit passes back the most viable offers to the market-side plugin, which are recorded and passed on to the users. Users receive these offers in the form of a collective SMS message summarizing their options. A user responds using the unique identifier for an offer within a set limit of time in which the offer is good. Once an offer is selected, the winning kiosk is notified and securely given the transaction key so that it can download data. Data is transmitted from the source using the secure infrastructure provided by Sulula. The 128-bit Advanced Encryption Standard (AES-128) is used for block encryption and MD5 for hashing. Figure 8.1 shows the plugin architecture used by the marketplace.

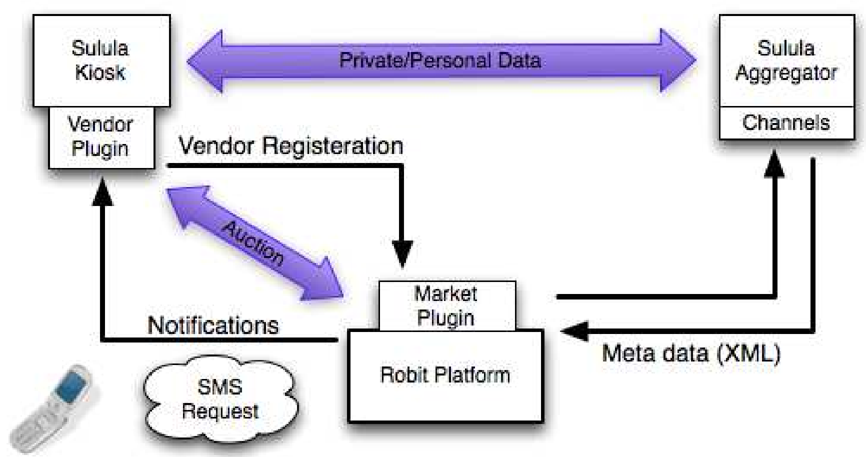


Figure 8.1: Modifying the Sulula data distribution network to benefit from the Robit auction platform

Robit in addition was used to build a standalone marketplace for tangible goods in a community. Its low barrier to entry and low cost make it a good complement to the current status-quo of physical or word-of-mouth markets. In providing this standalone auction marketplace, the important design principles are:

- Building an easy and customizable interface to buying and selling
- Leveraging widely available technologies, and less reliance on good network connectivity
- Establishing a flexible market structure that leaves buyers and sellers in control of their business, and
- Self-sustaining the service with incentives for the marketplace operator

Participants in the standalone market are uniquely identified using their cell phone numbers. This establishes a somewhat permanent identifier that can be used to manage payments and communication. In addition, this allows for easy plug of micropayment systems such as M-PESA [87] into the system. The marketplace is structured such that sellers are charged a small fee to list their items. This fee can be adjusted based on operation expenses of the entity running the marketplace, or even waived if the service is provided by an NGO or a community organization. In either case, enabling services to be profit oriented, or at least self sustaining, is a critical component of a lasting impact that stays longer than pilot study periods.

The standalone market has two main user-facing components. The first piece is an SMS-powered item listing and browsing component that enables buyers and sellers to come together around the marketplace. While building a web-based market initially was initially considered, the system opted for the more widely available means in many of these environments. Listing an item on the market is a simple task that can be done with two text messages, one describing the meta-information about the item, such as name, reserve price, length of auction and location, and another giving more details about the item.

However, no sequence of SMS messages could provide enough information for buyers regarding the item for sale. As a result, the main purpose of the listing text messages is providing a searchable database that only describes the key identifying characteristics of an item. This infrastructure supports Unicode characters, and items can be listed and searched using the local SMS supported language. Ideally, such information will be augmented by pictures and video that describe the exact details of the item. However, this is not practical with the poor network conditions and the lack of access to internet communication for many people. For this reason, the marketplace is augmented with a voicemail based product gallery that is used to further describe items.

The simple voice based system is run concurrent to the marketplace. When a user lists an item for sale using SMS messages, she is given a unique auction ID and a corresponding PIN and encouraged to call the system. When the seller calls the system, she is asked for the ID and PIN of the auction, and allowed to record or modify the item description. The length of the description can be set by the market operator. Once this description is recorded, it is made available as an additional communication channel for buyers to learn more about the item.

When a buyer expresses interest in buying an item using a text message, given as a description of the item, with a preferred location and price range, Robit does a search for available item and provides a summary of the title and auction IDs of matching auctions. The buyer then can learn more about particular items by asking for the details available on the marketplace. Optionally, the buyer can also call the system to check if there is further voicemail description left by the seller, identified by the auction ID. While a voice description will not replace pictures, video or an in-person visit, it suffices to give the buyer a good idea about the item. Furthermore, by structuring the auction around the right for first refusal by a buyer, the Robit marketplace caters to the inherent information gap due to the narrow communication channel in these environments.

8.2.1 Auction mechanism design

Robit uses an auction design tailored specifically to address the aforementioned constraints associated with mobile marketplaces. The auction mechanism adopts the second-price auction policy [129][1] : the highest bid buyer pays the auctioneer or the seller the second highest bid for the auctioned item that he or she wins. This payment system renders the second-price auction framework *incentive compatible*, meaning bidders' optimal bidding strategy is to bid exactly how much they value the auctioned item [54].

After examining the item in greater depth, the winner is permitted to withdraw from the auction without paying any penalty. This feature thus compensates for *information asymmetry*, when one side of the market has more information than the other, rising from the nature of the mobile marketplaces [9]. In other words, buyers' right of first refusal helps to prevent scenarios where sellers provide false information or purposefully create exaggerated impression about their items. Although revealing detailed item information to selective group of high-bid buyers and allowing them to resubmit their bids can also alleviate the effects of asymmetric information in Robit, such auction mechanisms require more time and effort from the buyers, thus discouraging them from bidding altogether.

From the previous discussion, one may assume that Robit's second-price auction mechanism can eliminate strategic elements in buyers' bidding decisions, consequently improving the marketplace's ease of use. However, Robit's permission for auction winners to withdraw their bids violates this incentive compatibility property in the case of malicious bidders, whose interest is not to maximize their own utility but to crowd out as many buyers as possible. In particular, the enactment of the first refusal right may expose sellers and buyers to attacks by malicious buyers who can crowd out legitimate buyers by submitting artificially high bid and then withdrawing from the auction. The time and monetary cost of sending SMS messages and monitoring auctions may in part deter such behavior.

In order to further discourage these malicious buyers in high-priced items, Robit imposes a refundable deposit requirement, $d \in [0, 1]$ of a buyer's bid, that will be collected at the beginning of the auction and returned in full when it concludes. To reserve the winner's right of first refusal in cases where later revealed details about the item do not

[1]eBay, a prominent internet auction website, also implements its own version of second-price auction [105]

meet the buyer's expectations, the market always refunds the buyers' deposit, regardless of the auction's outcome. This deposit requirement can further prevent malicious buyers from participating in several auctions simultaneously. In particular, the empirical study examines the effect of different deposit requirement levels d on social surplus for items of different values. The results show buyers may benefit more from medium to high deposit rates, while sellers may not, prompting auction operators for Robit to determine their marketplaces' deposit rate that is optimal for both buyers and sellers. Note that although an entry fee will likely effectively deter this malicious behavior, the system restrains from imposing extra barriers that may undermine Robit marketplaces' accessibility and usability. Payments and deposits could be made using existing micropayment facilities or through in-person dealings.

8.3 Implementation

The auction component of Sulula is mostly implemented using C# and the .NET platform. It has a number of sub-components that define the various modules in the system that provide different functionalities. Open source technologies were used whenever possible to reduce the time of development as well as increase the quality and reusability of pieces. The major components in the auction platform of Robit are the communication manger, the SMS manger, the search manager and the market manager.

The communication manager handles network based communications with other components in the system. It also provides an easy way to extend the communication channels to new components. In the context of Sulula, this component is used to manage the communication with various vendors, as well as data sources that have the private data that users are interested in. In the standalone market, the communication manager talks with components that deal with the voicemail gallery. As such, this component has to understand a number of protocols as it interacts with different entities. The system makes use of an open source XML-RPC package [24] besides other remote procedure call facilities present in the .NET platform.

The SMS manager deals with sending and receiving text messages, while communicating with the market manager that deals with the internals of the auction. There were a few options here. One could include a commercial grade SMS gateway that is able to handle high intensity traffic as used by several companies, or outsource the actual SMS sending and receiving using services like Twilio [123], which provide a simple way to integrate SMS in applications without dealing with SMS directly. However, a simple and self-contained solution is desired to begin with, which would allow for a quick deployment and small barrier to entry. On this front, some of the options were the Gnokii project associated with Nokia [44] and the MSR SMS Toolkit [128]. The system uses the MSR SMS Toolkit because of its easy integration to .NET projects. The toolkit is an existing SMS gateway solution from Microsoft Research India, used to handle and respond to requests from users via a connection to an SMS sending/receiving port. The SMS toolkit is a simple programmable interface to SMS messaging, with hooks to applications that enable integration with various systems. A smartphone communicating with the processes running on a PC serves as the SMS sending and receiving port. Meanwhile, the phone can also be used for making and receiving calls.

The SMS manager has two subcomponents. One listens to and forwards requests, while the other operates a separate notification thread that provides current information to participants in the market. This is used to let users know when important events, such as auction closing or unlisitng, happen in the system. The SMS listener uses a command parser to interpret text messages and perform requested actions. Users are able to register, sell, bid, unlist, add and get details about auctions among other things.

The search manger performs guided search for users regarding items present in the marketplace. This is a crucial component because the opportunity for communication is narrow with buyers, as everything happens through SMS messages. This component indexes the title, location, price and detailed description of auctions and performs a natural language

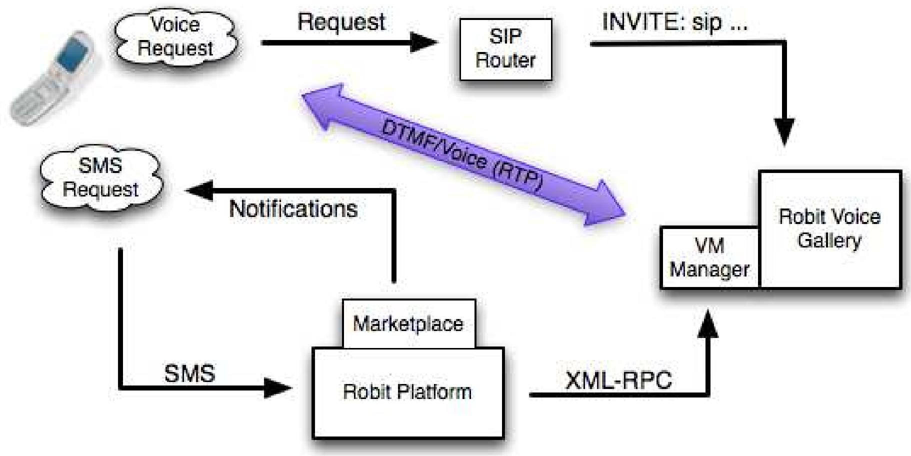


Figure 8.2: Robit and the Voicemail Gallery

search guided by provided constraints. It ranks the found items using a relevance measure, while limiting for the preferred location and the price range buyers are looking for. A summary of the top (by default 3) matching items is given back to the user using an SMS message. This summary includes the auction ID, title, location and the current highest bid on the item along with the remaining time. The buyer then can either request for the detail or check if there is a voicemail gallery for the item. Currently, locations search is limited to city/town levels due to the lack of structured addresses and naming conventions. However, recent advances in mapping technologies for developing countries [73] provide a promising avenue for finer grained approaches.

The market manager is the component that implements the various auction strategies that were described above. At the bare minimum, it keeps track of open and closed auctions in the system, and performs needed system housekeeping. This component interfaces with the SMS manager to learn about requests as well as notify users of events. It interfaces with the search manager when identifying relevant auctions that a user might be interested in. It consists of two additional threads that monitor auctions that are nearing to closing time, and those transactions that have been completed and need to be logged for future reference.

Another important component in the marketplace is the voicemail gallery for items.

This component implements an interactive voice response system that augments the main auction platform. With the increased adoption of VoIP, voice and telephony solutions are increasingly getting integrated with networked applications. This enables applications to have intuitive user interfaces that are very easy to customize to different cultures and environments. The Session Initiation Protocol (SIP) is at the core of many of these solutions, and the solution in Robit is SIP powered as well. The system builds on the open source SIP Express Media Server (SEMS) and SIP Express Router(SER) [110] projects, and implements mostly in C++, and added the needed marketplace features.

Calls to the system are switched using the SIP router, and terminated using the media server. The media server interacts with the marketplace to learn about open auctions, and assigned PINs for sellers. As users call the system, they are guided through customizable voice prompts to either check a voicemail gallery for an item, or create a new voicemail gallery for an open auction in the marketplace. Parts of this IVR system are implemented in Python that wraps the internal C++ modules. The prototype was run on a Pentium 3 machine with only 512MB of memory, and functions properly even on such low powered computers. Calling the voicemail system from the public switched telephone network (PSTN) requires an SIP-PSTN gateway, which is easy to implement, but could be pricey to start with. For pilot deployments, it is possible to use SIP compliant softphones which are widely available in internet kiosks as they are often used to make VoIP calls over the network. Figure 8.2 shows how the marketplace and voicemail gallery work together.

To make customization as easy as possible, all voice prompts are recorded as simple WAV files and placed in a user modifiable folder. Each prompt is clearly identified by a describing name, and customizing the prompt to a local language is only a matter of replacing the prompts with a different recording, while leaving the name unchanged.

8.4 Evaluation

Robit has been evaluated in a few different ways to analyze its utility in a challenged environment setting. This section will first discuss the auction market simulations performed to compare how different deposit requirements affect auction outcomes and the social surplus created. It then discusses a user study conducted in Addis Ababa regarding the utility of an auction based market place in developing countries like Ethiopia. The study gave us important insight into how conventional markets operate, and was important in revising the design for Robit. One of the interesting findings was the difficulty people have in buying and selling used goods outside of their social circles. Finally, the section will describe a small in-country pilot deployment that was performed. There were local translators and a video tutorial that gives a quick overview of the Robit platform, with the standalone market as an example.

8.4.1 Auction simulation

The first evaluation empirically studies how the deposit requirement $x \in [0, 1]$ affects the social surplus for two types of goods: high-valued and low-valued. In particular, a Robit market scenario of 100 buyers and 100 sellers who all sell the same good is simulated. Each seller s holds an auction for his or her one item, whose value r_s (from the seller's perspective) is also the auction's *reserve price*. Each buyer b is endowed with e_b dollars, and cannot borrow money at any time, guaranteeing $e_b \geq 0$. A randomly chosen subset of 20 buyers participate in each auction. It is important to note that the evaluation analyzes the case where buyers are involved in several auctions at a time, which might be unlikely in the common case. However, in doing so, the evaluation focuses on the interesting properties of the system in deterring malicious users who are trying to game the market. For most people who participate only in a handful of auctions at a time, the deposit requirement will have almost no impact, which is the intended purpose of the approach.

A buyer sends a text message to Robit asking for details about the item. Buyer b's valuation for seller s's item is specified as $x_{b,s} = v_b + \alpha_{b,s}$ where v_b represents b's average valuation for the good, and $\alpha_{b,s}$ reflects b's preference for the particular item sold by s. After sending his or her bid message to the seller, buyer b will be charged a deposit of $x_{b,s}d$, which will be refunded in full after the auction regardless of the final outcome. The seller then chooses the winner $b*$ and the final price x_s*, as described in Section 8.2.1, and moreover, provides detailed item information to $b*$, which could involve an in-person visit. After considering this information, buyer $b*$'s valuation for the item is updated to $x'_s = r_s + \beta_s$. $b*$ will exercise the right of first refusal if and only if $x'_{b*,s} \leq x_s*$, and will pay for the item otherwise.

A Robit marketplace operator could charge a small fee for sellers to list their items. Since these charges are the sellers' fixed *sunk costs* [126], these listings will not be included in the analysis. The cost of sending a single SMS in Ethiopia, as provided by the Ethiopian Telecommunication Corporation, is around \$0.03 [36], and this value is used in the simulation. Note that each of the participating buyers in an auction sends at least two messages to first ask about the status of the auction and then to submit his or her bid. The winning bidder sends an additional message notifying whether he or she would like to pay for the item.

The empirical study consists of two scenarios: in scenario A, the auctioned items have smaller values ($r \sim 4 \times N(1, 0.25), \beta \sim 3 \times N(1, 0.25), v \sim 6 \times N(1, 0.25), \alpha \sim 0.5 \times N(1, 0.25), e_b \sim 20 \times N(1, 0.25)$), while auctioned items in scenario B have higher values ($r \sim 21 \times N(1, 0.25), \beta \sim 3 \times N(1, 0.25), v \sim 22.5 \times N(1, 0.25), \alpha \sim 0.5 \times N(1, 0.25), e_b \sim 40 \times N(1, 0.25)$). 5% of the buyer population is assigned as malicious; each of these malicious buyers will bid twice their valuation for the item. Note that malicious buyers do not necessarily withdraw their bids in all auctions, since they are charged the second highest bid price, and more importantly, are also interested in purchasing auctioned items like others. The above parameters were calibrated so that at least more than half of the auctions are successful on average in all settings, which mirrors the fact that a

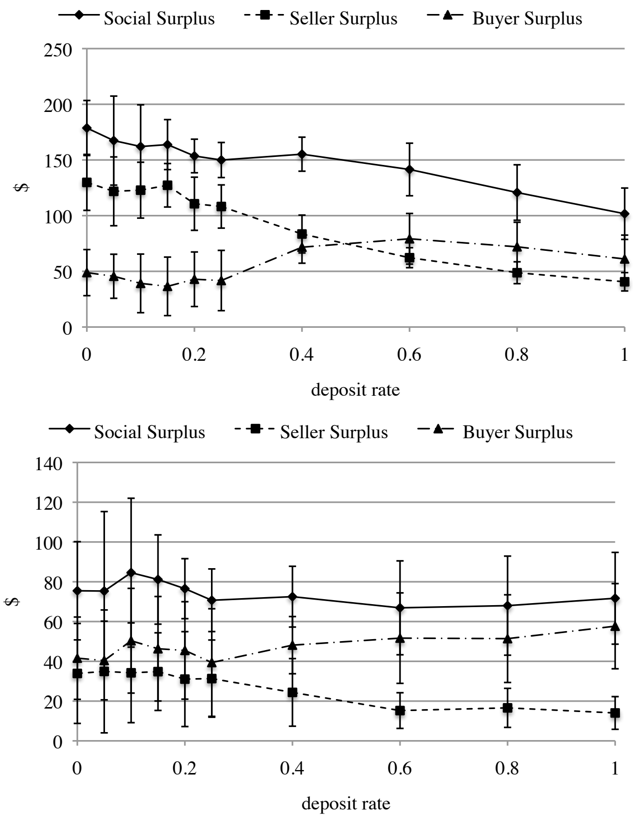


Figure 8.3: Results of buyer, seller and social surplus from market simulation

majority of items listed on popular auction sites are sold.

The auctions' outcomes were evaluated by three measures: i) seller surplus $\delta_s = \sum_s I_s(x_s * -r_s)$ where $I_s = 1$ if s sold his or her item and 0 otherwise, ii) buyer surplus $\delta_b = \sum_s I_s(x'_s - x_s*)$, and iii) social surplus $\delta = \delta_s + \delta_b$. As depicted in Figure 8.3, social surplus declines as the deposit rate increases in scenarios A and B, and so does buyer surplus. At the same time, seller surplus increases with d and levels off around $d \in [0.6, 1]$. Intuitively, higher deposit rates help to keep malicious buyers from crowding out legitimate buyers, and help the genuine buyers to win more auctions, resulting in more successful auctions. However, since malicious buyers are also interested in buying items whose final price is less than the buyers' valuations, their increasing absence also means lower final prices. In addition, fewer participants, both malicious and genuine, can submit bids for each auction as a result of higher deposit requirements. These concurrent effects therefore help explain the decrease in social surplus and in seller surplus and increase in buyer surplus for both scenarios A and B. An auction designer's job is then to choose the deposit rate that is acceptable to both buyers and sellers, while keeping the social surplus at a reasonable level.

8.4.2 Initial user study

Detailed interviews were conducted with twelve people, ages 17 to 52, in Addis Ababa, Ethiopia to understand some of the everyday market issues they deal with. The group consisted of students, business operators, government and private sector workers. The interview asked about their buying routines and how they compare prices etc. New item purchases in the last three months for the participants ranged from clothing and shoes to mobile phones, computers, furniture and accessories. These are bought from either outdoors markets, such as *Merkato* in the heart of Addis Ababa, or shops and boutiques found throughout the city. All participants said they compare a few places before making a purchase, and they use their judgment of the quality of the item alongside the price point comparisons to decide between alternatives. A small number of participants also said they use the internet for items like computers to get a feel for what the price should be. This however gives only a ball park number as many factors are involved in determining the actual price at the marketplace. Price negotiation is an important part of the buying process. Only a few shops have fixed, advertised prices, and buyers have to negotiate and compare prices to get the best value. Anecdotally, of the several stores visited in the city, fixed prices—very common in the US and many other countries—were found only in some of the high-end stores, and negotiation was the norm everywhere else.

Most of the participants also have bought used items in the past. These include electronics, furniture and home appliances. Most of these are bought either through a social contact or through second-hand shops that buy and sell used items. Some participants also used brokers for this purpose. Price negotiations are even more important in this context because there are not enough similar items in the social buying circle to compare price points. Most of the people interviewed have also tried to sell items at some point in the past. The process is similar in buying used items as it involves finding takers either through social circles and brokers or taking it to a used item shop for sell. A few people also mentioned posting the item and contact information on a notice board.

About half of the participants have used brokers for buying and selling items. The broker fee ranges from 2%-10% of the final selling price. Brokers are especially important in buying items that are not commonly traded, such as cars and large appliances. However, they are also used in making smaller purchases like mobile phones and personal computers. Experiences with brokers widely vary, but most people are content with the process, but for the sometimes steep prices.

All participants knew about auctions and about a third have participated in one. Auction notices are common on the national television and radio broadcasts, often advertised by banks and the customs bureau. Auctions are regarded as an excellent mechanism for getting the best prices for items because many buyers can participate and see the item for sale. The sentiment was particularly strong with those participants who tried to sell bigger items in the near past. Most of the participants were willing to try an auction based system for buying and selling their own items.

8.4.3 Pilot deployment

Following the user study in the country, a pilot deployment was set up in Addis Ababa. Local partners and volunteers were invaluable in preparing a local translation of the system. The deployment started out by translating the Robit documents to Amharic. There also was a local language video tutorial that explains the key elements of the system, and relates it to the current experience. In addition, the voicemail gallery was customized to operate in Amharic.

The pilot deployment took place in a local internet kiosk which was doubling as the market operator. This is one potential avenue for increasing the Robit adoption. Another alternative is for a separate entity, such as a local shop or small business, to administer the marketplace. Several mock auctions were run over a few weeks, while letting the volunteers try out the system and give us feedback.

The feedback received was technical as well as social. The value a marketplace like Robit could bring to users was clearly noticed during the pilot deployment. A lot of people thought augmenting the system with a web component might also be useful as the those who have internet access could use it even better. Other feedback was about the experience people had when interacting with the SMS interface. On the other hand, some suggested various modifications to the market process, including holding the item auctioned at the market operator, or adding a way to verify an item listed for sale indeed belonged to the seller. This was meant to deter stolen goods from being sold on the marketplace. Other feedbacks included enforcing timeliness and building trust on the system over time. These are essential considerations in full deployment of the service.

8.5 Voice based systems

The Robit market benefits Sulula, and other digital services targeted to developing counties, by adding an auction based market platform which transforms one-to-one business transactions to a many-to-many environment that fosters competition and innovation. By providing a market platform, Robit enables developers to build services that can be traded on an open market while using widely available communication tools such as SMS and voice. This section looks at a few other examples of voice based systems as they relate to the voicemail gallery in Robit.

There are a number of voice based projects that enable communication within a community and can be easily shaped to the local needs. The main advantage of these systems is customization. The VoiKiosks [5] project from researchers at IBM India leverages their previous work in VoiServ [74] which mimics the structure of the WWW, but using voice.

This is enabled through the Hyperspeech transfer protocol (HSTP) [6] which defines a protocol to seamlessly connect telephony voice applications. HSTP enables the users to browse across voice applications by navigating the Hyperspeech content in the application, which enables the building of voice enabled sites, called VoiceSites.

VoiKiosks used this technology to develop community specific voice based 'bulletin boards', where users can listen to and record messages. This approach is a response to the observation that the usefulness of the internet information for people in developing counties and the many villages within is very limited because relevant content is often not available on the internet [51]. One of the services VoiKiosks provide is an advertisement for professional services, where people can call in and post the services they provide and how to reach them, much like a classified ad on a magazine. Interested takers can then call the provider using the contact information. Other voice enabled solutions targeted at developing countries include the Talking Phone project from Literacy Bridge [108] and the Freedom Fone [23] project currently implemented in Zimbabwe and South Africa. Robit builds on a number of principles from similar systems that showed that adding low-technology, but widespread interfaces to applications can boost its adoption in developing countries. These lessons were applied in the area of an auction-based marketplace that can be used for the exchange of digital services or tangible goods.

8.6 Discussion

Given systems that were designed to work well in challenged network environments, this chapter considers what it would take to make them viable and sustainable in developing region contexts. To that end, this chapter described Robit, an extensible auction-based market platform for challenged network environments. Robit enables application developers to expose their services to a wide audience using a market layer that interacts with users over widely available communication tools. By doing so, it attempts to create an ecosystem where users can conveniently discover and trade digital services and goods despite infrastructural challenges. The Robit architecture makes it easy to modify it for various services and purposes. As examples, this chapter discusses two applications—Sulula and a standalone marketplace—that benefited from the market services that are provided by Robit.

This chapter concludes the last topic covered in this dissertation. The next chapter will provide a brief summary of the main contributions in this dissertation, as well as future directions for this line of research. Thinking about computing technologies in the context of developing regions is a fairly new discipline—one that was enabled by a number of recent improvements in access and affordability. As a result, it remains a fertile area for conducting research that addresses challenges unique to the developing world.

CHAPTER IX

Conclusion and future directions

Challenged network environments, endemic to developing regions, make even simple network tasks unpleasant and rich media prohibitively difficult to access. This dissertation discussed several contributions towards validating the thesis that the wide availability of cellular telephones and cheap storage devices, along with patterns in human activity, can be used to significantly improve end-user experience in accessing and using data in challenged network environments. The document was presented in three topical areas which discussed understanding user behavior, building systems for improving user experience, and providing tools for making these systems viable in developing regions.

The first study in this dissertation leads to contributions in understanding web usage behavior in developing region settings, and its implications for web acceleration mechanisms. Web acceleration mechanisms are especially important in challenged network environments where connectivity is limited or expensive. However, traditional solutions that rely on the redundancy of user requests for improving performance find it increasingly difficult to be effective as web usage gets personalized and fragmented. This study has resulted in a large scale personalized web usage data collected in a developing country context that is available for researchers. The analysis on this dataset provides some tangible evidence for describing the personal nature of web access in developing regions. For example, although web access behavior is quite diverse when considering users in aggregate, the patterns tend to be self-similar when looking at users individually. These observations are important in designing systems that are appropriate for challenged networks, and the dissertation has provided several examples along these lines.

The next study in the dissertation looked at a case study of web usage behavior in developing regions with a focus on online social networks. Social networking has been one of the fastest growing segments in web access that is also driving personalization. The study focused on access behavior of users, as well as the social, economical and cultural factors that impact some of this behavior. Using profile and activity data from LinkedIn, a social networking site with over a 100 million members worldwide, this study has presented several themes in social networking usage in developing regions. Some of the themes discussed include nature of interconnectedness and geographic locality for members, the activity and engagement level of members in developing regions, as well as access verticals for content from various regions.

As several other studies have also indicated, users in developing regions spend a sizable portion of their time on personal or private destinations, with social networking as a one example. The observations in this part of the dissertation point to at least two important usage scenarios that can benefit from custom designed systems for improving end-user experience as data access grows more personal and media-rich. One is addressing the challenge of distributing private data in challenged networks, while the other is designing mechanisms for efficiently transferring bulk data despite poor network connectivity.

When it comes to personal data that is needed by a single user, traditional caching and prefetching techniques provide little to no assistance. In challenged network environments, this means users often have to wait while their data is fetched from a remote server over a very slow link. This dissertation presented Sulula, an infrastructure for distributing private data in challenged network environments that leverages the near ubiquity of mobile phones for scheduling future requests. This system is based on the observation that many standard techniques for improving user experience in accessing content rely on the redundancy in content access. By using data channels, Sulula can easily integrate with the existing infrastructure and incrementally support various data sources. Sulula was implemented with three data channels—an email channel, a news feed channel and an HTML channel, that use the framework to deliver private or personal data from common sources.

The experimental results show Sulula can save users tens of minutes in a typical email/news reading session in challenged networks. In addition, the system facilitates a fair and transparent pricing structure that can improve how digital services are traded in developing regions. In addition, the dissertation described a small pilot deployment in Addis Ababa, Ethiopia with positive results. Sulula can be used alongside traditional caching and content distribution techniques to improve internet access experience in challenged network environments.

The next focus in the dissertation was the transfer of bulk data in challenged networks. In particular, the dissertation discusses an experimentation platform for prototyping systems for challenged networks, and evaluating them iteratively. This is important because these systems are expected to operate under a variety of circumstances. The system described in this dissertation, Vivo, is a platform for simplifying such experimentation. The key contribution of Vivo is allowing researchers to rapidly prototype systems for challenged networks with live-code, and evaluate them reproducibly. Using a menu of composable ingredients, Vivo liberates researchers to focus on their area of interest in building systems, thereby reducing the effort needed to do so. These systems can be directly evaluated on Vivo's network and emulation platform. In demonstrating Vivo's utility to researchers, the dissertation discussed three example systems that were built atop Vivo. Two of there were previously proposed by other researchers, while the third was a new hybrid overlay network known as Bati.

The dissertation demonstrated how researcher would go about building systems like Bati, and how the Vivo emulation platform enables them to reproducibly and iteratively evaluate their design decisions. Bati leverages estimates of natural mobility along with available network resources to transfer Data. The results with conservative estimates of available resources show savings in using Bati for delivering bulk data compared to using the network alone, while improving delivery rates compared to solutions that only use ad-hoc routing between mobile peers.

In conclusion, the first two parts of the dissertation focus on understanding data access behavior and building systems for improving user experience based on these observations. The next part of the dissertation turned to the market and service discovery considerations that are important in making these systems viable for developing region use. Providing services like personal data aggregation carry monetary cost, and an efficient mechanism for trading these service is a requirement for scalable use. To facilitate a wide adoption of digital services, the dissertation introduces an extensible auction-based market platform for challenged network environments that can also be used more generally.

Open and competitive markets bring enormous value to communities. The internet has enabled such markets and changed how people buy and sell goods. However, these markets are not usually available in challenged network environments due to poor infrastructure and lack of relevant content. While some isolated solutions targeted to challenged environments exist, they are often confined to certain domains or communities. Appropriately designed marketplaces are especially important in building sustainable services in developing regions.

The last chapter in this dissertation described Robit, an extensible auction-based market platform for challenged environments. Robit incorporates ideas from economic and auction theories with widely available communication tools to create a market platform that can be used by developers in various applications. Robit is a platform, and as such, can be used to build a market layer in a number of applications. These could be marketplaces for trading digital services or goods. To demonstrate how Robit can be used to add a market layer to applications, the Sulula system was modified for taking advantage of the marketplace platform. In addition, the dissertation analyzes a standalone auction-based market for tangible goods built atop Robit.

Combined together, this dissertation brought together several projects that aim to understand and improve end-user experience in accessing and using data in challenged networks. Designing systems for developing regions is often driven as much by performance metrics as it is by practicality and appropriateness. The many projects described in this dissertation focus on practicality by trying to understand user data access behavior, and designing systems appropriate to the target audience.

The study of computing technologies for development is a research area in an early phase. Due to improvements in the buying power of individuals and the cost implications of Moore's Law, access to computing technology has accelerated in developing regions. Yet, developing regions present a unique set of challenges—spanning infrastructure, literacy, language etc.—that render many existing computing solutions unsuitable to these regions. Fortunately, ICTD continues to attract academic interest from several institutions, with multidisciplinary interests from economics, policy and information studies. As a result, it presents a fertile opportunity to conduct research with great impact and visibility.

Moving forward, one medium term research goal is to work at the intersection of computer science and the study of information communication technologies for development. Based on qualitative and quantitative analysis of information technology use in emerging economies, it would be interesting to build systems that cater to some of the unique challenges. Over the coming years, some interesting areas of research include systems for intermittent connectivity, low-cost wireless connectivity options, mobile systems and applications for developing regions, power efficient systems, and security challenges for communication technologies in developing regions.

One particular area of interest is working on technical solutions to assist in the economic development of moderately educated individuals in developing regions. It is interesting to observe that the youth literacy rate in Sub-Saharan Africa has been significantly improving, reaching as high as 98% in Zimbabwe and well over 60% on average [124].

By leveraging widely available communication resources, it would be useful to research appropriately designed technologies that can create or facilitate economic opportunities for moderately educated individuals—able to read, write in their local language and perform basic arithmetic—in developing regions. There are at least three important challenges that need to be addressed in this regard. These include improving access to information, designing simple, secure and scalable payment systems in support of local and online economic opportunities, and overcoming language barriers through appropriate interfaces.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Measuring The Information Society, 2010. International Telecommunication Union.

[2] The CIA world factbook. `http://www.cia.gov/cia/publications/factbook`, August 2011.

[3] Standard country and area codes classifications (M49), 2011. United Nations Statistics Division.

[4] United Nations Human Development Report, 2011. United Nations Development Programmee.

[5] Sheetal Agarwal, Arun Kumar, Amit Anil Nanavati, and Nitendra Rajput. Voikiosk: increasing reachability of kiosks in developing regions. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1123–1124, New York, NY, USA, 2008. ACM.

[6] Sheetal K. Agarwal, Dipanjan Chakraborty, Arun Kumar, Amit Anil Nanavati, and Nitendra Rajput. Hstp: hyperspeech transfer protocol. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 67–76, New York, NY, USA, 2007. ACM.

[7] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[8] Akamai Reports. The State of the Internet, 2nd quarter, 2009.

[9] G.A. Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. *The quarterly journal of economics*, 84(3):488–500, 1970.

[10] Yaw Anokwa, Colin Dixon, Gaetano Borriello, and Tapan Parikh. Optimizing high latency links in the developing world. In *WiNS-DR '08*, pages 53–56, New York, NY, USA, 2008. ACM.

[11] N.B. Azzouna and F. Guillemin. Analysis of ADSL traffic on an IP backbone link. In *Global Telecommunications Conference,IEEE*, pages 3742–3746 vol.7, Dec. 2003.

[12] Anirudh Badam, KyoungSoo Park, Vivek S. Pai, and Larry L. Peterson. HashCache: cache storage for the next billion. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 123–136, Berkeley, CA, USA, 2009. USENIX Association.

[13] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. volume 37, pages 373–384, New York, NY, USA, 2007. ACM.

[14] Prasanta Bhattacharya and William Thies. Computer viruses in urban indian telecenters: characterizing an unsolved problem. NSDR '11.

[15] David Bild, Yue Liu, Robert Dick, Z. Morley Mao, and Dan Wallach. Using predictable mobility patterns to support scalable and secure manets of handheld devices. MobiArch '11.

[16] John Burgess, Brian Gallagher, David Jensen, and Brian N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM*, April 2006.

[17] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *WCMC 2002*.

[18] Pew Research Center. Global Publics Embrace Social Networking. *Pew Global Attitudes Project*, 2010.

[19] Jay Chen, Saleema Amershi, Aditya Dhananjay, and Lakshmi Subramanian. Comparing web interaction models in developing regions. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV '10, pages 6:1–6:9, New York, NY, USA, 2010. ACM.

[20] Jay Chen, David Hutchful, William Thies, and Lakshmi Subramanian. Analyzing and Accelerating Web Access in a Shool in Peri-Urban India. In *WWW 2011*. ACM, 2011.

[21] X. Chen and A.L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *POMC 2001*, pages 21–27, Newport, RI, USA.

[22] Christine Cheng, Ravi Jain, and Eric van den Berg. Location prediction algorithms for mobile wireless systems. 2003.

[23] Bev Clark and Brenda Burrell. Freedom Fone: Dial-up Information Service. In *International Conference on Information and Communication Technologies and Development (ICTD)*, Doha, Qatar, 2009.

[24] Charles Cook. XML-RPC for .NET. `www.xml-rpc.net`.

[25] The Ethiopia Commodity Exchange, 2010. `www.ecx.com.et`.

[26] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51:107–113, January 2008.

[27] Michael Demmer and Kevin Fall. DTLSR: delay tolerant routing for developing regions. In *NSDR '07*, New York, NY. ACM.

[28] The Digital Divide at a Glance, 2005. World Summit On the Information Society.

[29] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. Globally Distributed Content Delivery. *IEEE Internet Computing*, 6(5):50–58, 2002.

[30] Martin Doege. NewsFeed Aggregator. `home.arcor.de/mdoege/newsfeed/`.

[31] Jonathan Donner, Shikoh Gitau, and Gary Marsden. Exploring mobile-only Internet use: results of a training study in urban South Africa. *International Journal of Communication*, 5:574–597, 2011.

[32] Bowei Du, Michael Demmer, and Eric Brewer. Analysis of WWW traffic in Cambodia and Ghana. In *WWW '06*, pages 771–780, New York, NY, USA, 2006. ACM.

[33] Robert C. Durst, Gregory J. Miller, and Eric J. Travis. Tcp extensions for space communications. *Wirel. Netw.*, 3, October 1997.

[34] Leaders: The real digital divide - Technology and development. *The Economist*, 374:9–10, 2005.

[35] Ethiopia Tel. Co. Services, 2009. `www.telecom.net.et/services`.

[36] Annual Statistical Bulletin, 2005. Ethiopian Telecommunication Agency.

[37] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03*, pages 27–34, New York, NY, USA, 2003. ACM.

[38] Li Fan, Pei Cao, Wei Lin, and Quinn Jacobson. Web Prefetching Between Low-Bandwidth Clients and Proxies: Potential and Performance. pages 178–187, 1999.

[39] Jason Flinn, Shafeeq Sinnamohideen, Niraj Tolia, and M. Satyanaryanan. Data Staging on Untrusted Surrogates. In *FAST '03*, pages 15–28, Berkeley, CA, USA, 2003. USENIX Association.

[40] N. Freed and N. Borenstein. Rfc 2045: Multipurpose internet mail extensions (mime), 1996.

[41] Frontline SMS. SMS Solution for NGO's. `www.frontlinesms.com`.

[42] J. Gantz. The diverse and exploding digital universe. Technical Report White paper, IDC, 2008.

[43] Natalie Glance, Dave Snowdon, and Jean-Luc Meunier. Pollen: using people as a communication medium. *Comput. Netw.*, 35(4):429–442, 2001.

[44] Gnokii: Open Source Tools for Your Mobile Phone, 2009. `www.gnokii.org`.

[45] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-Laszlo Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[46] Google. Google SMS. www.google.com/sms.

[47] Daniel Gorgen, Hannes Frey, and Christian Hiedels. Jane-the java ad hoc network development environment. In *Proceedings of the 40th Annual Simulation Symposium*, 2007.

[48] S. Guo, M. H. Falaki, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, U. Ismail, and S. Keshav. Design and implementation of the kiosknet system. In *ICTD '07*.

[49] Haggle Project. www.haggleproject.org.

[50] J. Hartigan. *Clustering Algorithms*. John Wiley and Sons, New York, 1975.

[51] Richard Heeks. Failure, success and improvisation of information systems projects in developing countries. *The Information Society*, 18, March 2002.

[52] Mellisa Ho. The internet (or lack there of), 2009. www.ictdchick.com/blog.

[53] Hsu-Chun Hsiao, Ahren Studer, Chen Chen, Adrian Perrig, Fan Bai, Bhargav Bellur, and Aravind Iyer. Flooding-resilient broadcast authentication for vanets. MobiCom '11.

[54] L. Hurwicz. On informationally decentralized systems. *Studies in resource allocation processes*, page 425, 1977.

[55] Sunghwan Ihm, KyoungSoo Park, and Vivek S. Pai. Towards understanding developing world traffic. In *NSDR 2010*, pages 8:1–8:6.

[56] Sunghwan Ihm, KyoungSoo Park, and Vivek S. Pai. Wide-area network acceleration for the developing world. In *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, pages 18–18.

[57] Sibren Isaacman and Margaret Martonosi. Potential for collaborative caching and prefetching in largely-disconnected villages. In *WiNS-DR 2008*, pages 23–30.

[58] Sibren Isaacman and Margaret Martonosi. The C-LINK System for Collaborative Web Usage: A Real-World Deployment in Rural Nicaragua. In *NSDR 2009*. ACM, 2009.

[59] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *SIGCOMM '04*, Portland, Oregon. ACM.

[60] Mike Jensen. Open Access: Lowering the costs of international bandwidth in Africa. *APC Issue Papers*, 2006.

[61] Robert Jensen. The Digital Provide: Information (Technology), Market Performance, and Welfare in the South Indian Fisheries Sector. *The Quarterly Journal of Economics*, August 2007.

[62] Zhimei Jiang and Leonard Kleinrock. Web Prefetching in a Mobile Environment. *IEEE Personal Communications*, 5:25–34, 1998.

[63] D. L. Johnson, Elizabeth M. Belding, Kevin Almeroth, and Gertjan van Stam. Internet usage and performance analysis of a rural wireless network in macha, zambia. In *NSDR 2010*, pages 7:1–7:6, 2010.

[64] Evan P. C. Jones, Lily Li, and Jakub K. Schmidtke. Practical routing in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 6(8), 2007.

[65] A. Joseph, J. Tauber, and M. Kaashoek. Mobile computing with the rover toolkit. *IEEE Trans. on Computers*, 46(3), 1997.

[66] Samuel Kinde Kassegne, Ashenafi Assefa, and Mieso Denko. BoonaNet: SMS and IM Mediated Price Information for Crops in Ethiopia, 2007. `www.digitaladdis.com/sk/BoonaNet.ppt`.

[67] S. Kaul, K. Ramachandran, P. Shankar, S. Oh, M. Gruteser, I. Seskar, and T. Nadeem. Effect of antenna placement and diversity on vehicular network communications. In *Sensor, Mesh and Ad Hoc Communications and Networks*, 2007.

[68] Ari Keranen, Jorg Ott, and Teemu Karkkainen. The one simulator for dtn protocol evaluation. In *Proceedings of the 2nd Intl. Conference on Simulation Tools and Techniques*, 2009.

[69] David Kotz, Tristan Henderson, Ilya Abyzov, and Jihwang Yeo. CRAWDAD trace set dartmouth movement (v. 2005-03-08). http://crawdad.cs.dartmouth.edu, 2005.

[70] Jay Kreps, Neha Narkhede, and Jun Rao. Kafka: A distributed messaging system for log processing. In *Proceedings of 6th International Workshop on Networking Meets Databases (NetDB)*, Athens, Greece, June 2011.

[71] Tom M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul. Exploring the Bounds of Web Latency Reduction from Caching and Prefetching. In *USENIX Symposium on Internet Technologies and Systems*, 1997.

[72] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao. Oceanstore: An architecture for global-scale persistent storage. pages 190–201, 2000.

[73] Arun Kumar, Dipanjan Chakraborty, Himanshu Chauhan, Sheetal K. Agarwal, and Nitendra Rajput. Folksomaps - towards community driven intelligent maps for developing regions. In *ICTD'09: Proceedings of the 3rd international conference on Information and communication technologies and development*, pages 85–94, Piscataway, NJ, USA, 2009. IEEE Press.

[74] Arun Kumar, Nitendra Rajput, Dipanjan Chakraborty, Sheetal K. Agarwal, and Amit Anil Nanavati. VOISERV: Creation and Delivery of Converged Services through Voice for Emerging Economies. In *WOWMOM*, pages 1–8. IEEE, 2007.

[75] Benjamin Lambert and Omid Fatemieh. Generating intelligent links to web pages by mining access patterns of individuals and the community, 2005.

[76] Butler W. Lampson. Hints for computer system design. In *Proceedings of the ninth ACM symposium on Operating systems principles*, SOSP '83, pages 33–48, New York, NY, USA, 1983. ACM.

[77] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for DTNs. In *INFOCOM*, 2006.

[78] Hui Lei and Dan Duchamp. An Analytical Approach to File Prefetching. In *In Proceedings of the USENIX 1997 Annual Technical Conference*, pages 275–288, 1997.

[79] Qinghua Li, Sencun Zhu, and Guohong Cao. Routing in socially selfish delay tolerant networks. In *INFOCOM 2010*.

[80] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7, July 2003.

[81] Mark Lutz. PyMailGUI. `www.rmi.net/ lutz/`.

[82] Alireza Mahdian, Hansu Gu, Faris Kateb, Sravan Thokala, and Shivakant Mishra. Gps assisted adhoc routing using cellphones for poorly connected areas. NSDR '10.

[83] Mabobi Inc., 2010. `www.manobi.net`.

[84] K. W. Matthee, G. Mweemba, A. V. Pais, G. van Stam, and M. Rijken. Bringing Internet connectivity to rural Zambia using a collaborative approach. In *ICTD '07*, pages 47–58, 2007.

[85] Scott Michel, Khoi Nguyen, Adam Rosenstein, Lixia Zhang, Sally Floyd, and Van Jacobson. Adaptive web caching: towards a new global caching architecture. *Comput. Netw. ISDN Syst.*, 30:2169–2177, November 1998.

[86] Jeffrey C. Mogul, Fred Douglis, Anja Feldmann, and Balachander Krishnamurthy. Potential benefits of delta encoding and data compression for http. *SIGCOMM Comput. Commun. Rev.*, 27(4):181–194, 1997.

[87] O. Morawczynski and G. Miscione. Examining trust in mobile banking transactions in Kenya: The case of m-PESA in Kenya. *IFIP WG 9.4-University of Pretoria Joint Workshop*, 2008.

[88] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router.

[89] Akira Nagata, Shinya Yamamura, and Masato Tsuru. Web browsing over multiple heterogeneous challenged networks. In *Proceedings of the 6th ACM workshop on Challenged networks*, 2011.

[90] Samuel C. Nelson, Gautam Bhanage, and Dipankar Raychaudhuri. Gstar: generalized storage-aware routing for mobilityfirst in the future mobile internet. MobiArch '11.

[91] Anthony J. Nicholson and Brian D. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom*, 2008.

[92] B. D. Noble, M. Satyanarayanan, G. T. Nguyen, and R. H. Katz. Trace-based mobile network emulation. SIGCOMM '97.

[93] OAuth Core Workgroup, OAuth Core 1.0, 2007. `http://oauth.net/core`.

[94] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig Latin: a not-so-foreign language for data processing. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1099–1110, Vancouver, BC, Canada, June 2008.

[95] Operating System Market Share - Africa. `www.netapplications.com`.

[96] Themistoklis Palpanas and Alberto Mendelzon. Web Prefetching Using Partial Match Prediction. 1998.

[97] Andreea Picu and Thrasyvoulos Spyropoulos. Distributed stochastic optimization in opportunistic networks: the case of optimal relay selection. In *Proceedings of the 5th ACM workshop on Challenged networks*, 2010.

[98] Michael Rabinovich and Oliver Spatscheck. *Web Caching and Replication*. Addison-Wesley, 2001.

[99] Kishore Ramachandran, Sanjit Kaul, Suhas Mathur, Marco Gruteser, and Ivan Seskar. Towards large-scale mobile network emulation through spatial switching on a wireless grid. In *Proceedings of the ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*, 2005.

[100] Aishwarya Ratan and Savita Bailur. Welfare, Agency and 'ICT for Development'. In *ICTD '07*, pages 119–130, 2007.

[101] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Comm. and Netw. Conference*, 2005.

[102] DTNRG: IRTF dtn research group bundle protocol, 2011. `http://www.dtnrg.org`.

[103] S.C. Rhea and J. Kubiatowicz. Probabilistic Location and Routing. In *INFO-COM'02*.

[104] Sean C. Rhea, Kevin Liang, and Eric Brewer. Value-based web caching. In *WWW '03*, pages 619–628, New York, NY, USA, 2003. ACM.

[105] A.E. Roth and A. Ockenfels. Last-minute bidding and the rules for ending second-price auctions: Evidence from ebay and amazon auctions on the internet. *American Economic Review*, 92(4):1093–1103, 2002.

[106] Umar Saif, Ahsan Chudhary, Shakeel Butt, Nabeel Butt, and Ghulam Murtaza. Internet for the Developing World: Offline Internet Access at Modem-speed Dialup Connections. In *ICTD '07*, pages 76–87, 2007.

[107] Anubhav Savant and Torsten Suel. Server-friendly delta compression for efficient web access. pages 303–322, 2004.

[108] Cliff Schmidt and Andrew Azaabanye Bayor. Health, Agriculture, and Literacy with a $10 Rural Audio Computer. In *International Conference on Information and Communication Technologies and Development (ICTD)*, Doha, Qatar, 2009.

[109] K. Scott and M. S. Burleigh. Rfc 5050: Bundle protocol specification, 2007.

[110] SIP Express Router and SIP Express Media Server, 2010. `www.iptel.org`.

[111] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. MobiCom '06.

[112] Sam Shah and Brian D. Noble. A study of e-mail patterns. *Softw. Pract. Exper.*, 37(14):1515–1538, 2007.

[113] Shunra. Desktop VE Wan Simulation. `www.shunra.com`.

[114] Roger W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68(2):159, 1984.

[115] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Laszlo Barabasi. Limits of predictability in human mobility. *Science*, 327(5968), 2010.

[116] L. Song, D. Kotz, R. Jain, and Xiaoning He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM'04*.

[117] specs@openid.net. OpenID Authentication 2.0, 2007. `http://openid.net/developers/specs/`.

[118] George J. Stigler. The economics of information. *Journal of Political Economy*, 69:213, 1961.

[119] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: a warehousing solution over a Map-Reduce framework. *Proceedings of the VLDB Endowment*, 2:1626–1629, August 2009.

[120] Xiaozheng Tie, Arun Venkataramani, and Aruna Balasubramanian. R3: robust replication routing in wireless networks with diverse connectivity characteristics. MobiCom '11.

[121] Esoko/TradeNet, 2010. `www.esoko.com`.

[122] Trade at Hand, 2010. `www.intracen.org/trade-at-hand`.

[123] Twilio Cloud Communications: Web Service API for building Voice and SMS Applications, 2010. `www.twilio.com`.

[124] Global Education Digest, 2011. The United Nations Educational, Scientific and Cultural Organization (UNESCO).

[125] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical report, Duke University, 2000.

[126] H.R. Varian. Intermediate microeconomics: a modern approach. *New York*, 1999.

[127] Rajesh Veeraraghavan, Naga Yasodhar, and Kentaro Toyama. Warana Unwired: Replacing PCs with Mobile Phones in a Rural Sugarcane Cooperative. *Information Technologies & International Development*, 5(1), 2009.

[128] V.Goyal and S.Blagsved. SMS server toolkit, 2007. `www.codeplex.com/smstoolkit`.

[129] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of finance*, 16(1):8–37, 1961.

[130] Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, and Abhijeet Joglekar. An integrated experimental environment for distributed systems and networks. In *OSDI'02*.

[131] Jeff Wilson. Probabilistic routing in delay tolerant networks. Master's thesis, Baylor University, 2007.

[132] Dave Winer. XML-RPC Specification. `www.xmlrpc.com/spec`.

[133] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the fifth international World Wide Web conference on Computer networks and ISDN systems*, pages 1007–1014, 1996.

[134] Hayoung Yoon, JongWon Kim, Maximilian Ott, and Thierry Rakotoarivelo. Mobility emulator for dtn and manet applications. WINTECH '09.

[135] J. Yoon, M. Liu, and B. D. Noble. Random waypoint considered harmful. INFOCOM'03.

[136] J. Yoon, B.D. Noble, M. Liu, and M. Kim. Building realistic mobility models from coarse-grained traces. MobiSys'06.