Very Low-Cost Internet Access Using KioskNet

S. Guo, M.H. Falaki, E.A. Oliver, S. Ur Rahman, A. Seth, M.A. Zaharia, and S. Keshav
David R. Cheriton School of Computer Science
University of Waterloo
{sguo, mhfalaki, eaoliver, surrahman, a3seth, mazahari, keshav}@uwaterloo.ca

ABSTRACT

Rural Internet kiosks in developing regions can cost-effectively provide communication and e-governance services to the poorest sections of society. A variety of technical and non-technical issues have caused most kiosk deployments to be economically unsustainable [1]. KioskNet addresses the key technical problems underlying kiosk failure by using robust 'mechanical backhaul' for connectivity [2], and by using low-cost and reliable kiosk-controllers to support services delivered from one or more recycled PCs. KioskNet also addresses related issues such as security, user management, and log collection. In this paper, we describe the KioskNet system, outlining its hardware, software, and security architecture. We describe a pilot deployment, and how we used lessons from this deployment to re-design our initial prototype.

Categories and Subject Descriptors: C.2.1 [INetwork Architecture and Design]: Store and forward networks, Wireless communication

General Terms: Design, Economics

Keywords: System design, delay tolerant networks, mechanical backhaul, rural communication, low cost.

1. INTRODUCTION

Rural Internet kiosks in developing regions can provide a variety of services such as birth, marriage, and death certificates, land records, and medical and agricultural consulting to the poorest sections of society. A typical kiosk has a Windows-based PC and a dial-up or VSAT connection to the Internet, and is operated by a computer-literate kiosk owner who maintains the system and assists end users. To effectively serve its users and be profitable to its owner, a kiosk should be highly available and should have a reliable connection to the Internet. Moreover, it should be low-cost, so that it can be sustained with a minimum of user fees. Due to limited electrical power, pervasive dust, mechanical wear-and-tear, and computer viruses, kiosk computers often fail, requiring frequent (and expensive) repairs. Similarly, network connectivity is often lost due to failures in the telephone system, inability to power the VSAT station, or loss of alignment of long-range wireless links. Faced with high costs and unreliable service delivery, customers quickly lose interest, and kiosk deployments are often found to be unsustainable in the long term [1].

KioskNet attempts to make a kiosk more robust without increasing its cost. First, it uses a single-board-computer-based, low-cost, low-power kiosk controller at each kiosk. The controller can communicate wirelessly with another single-board computer mounted on a vehicle (as was pioneered by Daknet [3]). These vehicles carry data to and from a gateway, where data is exchanged with the Internet. This 'mechanical backhaul' [2] avoids the cost of trenches, towers, and satellite dishes, allowing Internet access even in remote areas. In areas where dialup, long-range wireless or cellular phone service is available, the kiosk controller can be configured to use these communication links in conjunction with mechanical backhaul. Second, KioskNet allows refurbished PCs to boot from the kiosk controller. Kiosk controllers are reasonably tamper-proof so they offer reliable virus-free boot images and binaries. We do not use the PC's hard disk, thus avoiding hard disk failures and disk-resident viruses. Moreover, refurbished PCs are cheap and spare parts are widely available.

KioskNet has the following key features:

- The system is low-cost (see Section 5 for details) and appears to be economically viable. We estimate that our system requires a capital expenditure of $100-$700/kiosk, depending on the configuration [1], and an operating expenditure of $70/kiosk/month. These rough estimates include the cost of field technicians and capital depreciation. This is four to ten times cheaper than other solutions.
- The solution is rapidly deployable: we successfully installed a prototype in Anandapuram village, Vishakapatnam district, Andhra Pradesh, India in two days during May 2006.
- Kiosk controllers are low-power (6-8W), therefore they can be run off a solar panel.
- Recycled PCs can run either the (Linux) binaries that are packaged with the kiosk controller, which are guaranteed to be virus free, or can boot into an existing operating system (typically Windows) from their hard drive for stand-alone computing.
- We can provide private and authenticated communication among kiosk users, and between a kiosk user and a secure node in the Internet.
- Our software is shipped in the form of a LiveCD that can be booted on on any Windows or Linux PC. The CD is used to copy OS images directly onto hard drives, which are then installed in single-board computers.

[1] All figures are in US dollars

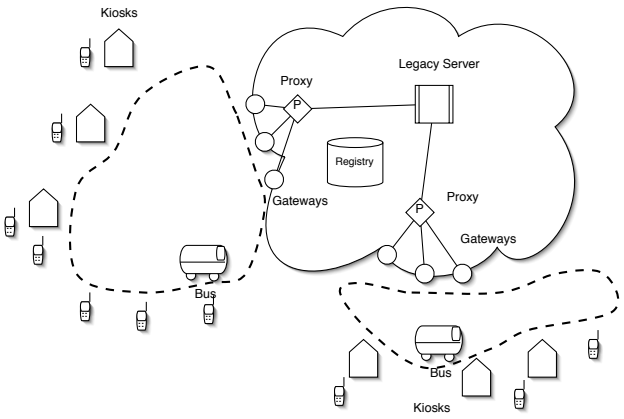

Figure 1: KioskNet overview.

- Our code is free under the Apache open-source license with no patent, copyright or intellectual property restrictions.

We present an overview of the system in Section 2 and its software architecture in Section 3. The security architecture is described in Section 4. We describe the cost structure in Section 5 and our experience with deploying the system in Section 6. Section 7 discusses improvement to our initial design that reflect experiences from the pilot deployment. We present related work in Section 8 and conclude in Section 9.

2. OVERVIEW

KioskNet consists of a set of kiosks that use mechanical backhaul [2] as the primary means of communication to the Internet (Figure1). *Ferries* carry data to and from a kiosk to a set of *gateways* which communicate with a *proxy* on the Internet. The remainder of this section describes these KioskNet elements in more detail.

2.1 Kiosks

Each kiosk has a kiosk controller, which is a server that provides recycled PCs with network boot, a network file system, user management, and network connectivity by means of dial-up, GPRS/SMS, VSAT, or mechanical backhaul. A kiosk controller always has a WiFi NIC. In addition, for most deployments, we expect that kiosk controllers would also provide connectivity by other means, such as GPRS, SMS, VSAT, or a dial-up connection. Our prototype uses headless and keyboard-less low-power single-board computers from Soekris Corporation[2] as kiosk controllers, although the controller functionality can be implemented in any commodity PC.

Two types of users are expected to use a kiosk. Most would use a recycled PC that boots over the network (using a RAM disk) from the kiosk controller, and can then access and execute application binaries provided by the kiosk controller over NFS. Recycled PCs cost approximately $100 and spare parts are widely available worldwide. Moreover, as a shared resource, they are much cheaper than any dedicated resource.

Other users, such as wealthier villagers, government officials, or NGO partners, could access one or more kiosks, or a bus directly, using their own devices, such as smart phones, PDAs, and laptops. Such users would use the kiosk-controller or bus essentially as a wireless hotspot that provides store-and-forward access to the Internet.

The set of kiosks in the same geographical area, and administered by the same entity, comprises a KioskNet *region*. Regions not only have administrative significance, in that all entities in a region are certified by the same certificate authority, but also have routing significance, because bundles are flooded within a region. Figure 1 shows a system with two regions, which could both be managed by a single administrative entity.

2.2 Ferries

Although kiosk controllers can communicate with the Internet using a variety of connectivity options, our focus is on the use of mechanical backhaul. This is provided by cars, buses, motorcycles, or trains that pass by a kiosk and an Internet gateway. We call such entities *ferries*.

A ferry has a single-board-computer that is powered from the vehicle's own battery. This computer has 20-40GB of storage and a WiFi interface. It communicates opportunistically with the kiosk controllers and Internet gateways on its path. During an opportunistic communication session, lasting from 20 seconds to five minutes, we expect 10-150MB of data to be transferred in each direction. This data is stored and forwarded in the form of self-identifying *bundles*. Ferries upload and download bundles opportunistically to and from an Internet gateway.

2.3 Gateways

A gateway is a computer that has a WiFi interface, storage, and an always-on connection to the Internet. Gateways are likely to be present in cities with DSL or cable broadband Internet access. A gateway collects data opportunistically from a ferry and stages it in local storage before uploading it to the Internet through the proxy. A region may have more than one gateway.

2.4 Proxy

We expect that most communication between a kiosk user and the Internet would be for existing services such as Email, financial transactions, and access to back-end systems that provide government-to-citizen services. Existing servers that provide such services typically cannot deal with long delays and disconnections. Therefore, we need a disconnection-aware proxy that hides end-user disconnection from legacy servers. We assume that there is one proxy per region.

The proxy is resident in the Internet and has two halves. One half establishes disconnection-tolerant connection sessions with applications running on recycled PCs or on mobile users' devices. The other half communicates with legacy servers. Data forwarding between the two halves is highly application-dependent; for example, a proxy that fetches email from a POP server on behalf of a user needs to implement the POP protocol. To support application-specific protocols, we allow an application to create an application-specific plugin at the proxy. This plugin can download data from the Internet on behalf of kiosk users. This data is then

[2]http://soekris.com

transferred to an appropriate gateway (as determined by a routing protocol) and subsequently handed off to a ferry using opportunistic communication for delivery to a kiosk. In the other direction, the plugin receives data from the gateways and transfers these to legacy servers.

2.5 Legacy servers

The last component of our architecture, the legacy servers, is typically accessed using TCP/IP and an application-layer protocol such as POP, SMTP, or HTTP by a proxy. We do not require any changes to legacy servers.

3. COMMUNICATION ARCHITECTURE

KioskNet software runs on proxies, gateways, ferries, kiosk controllers, and cell phones/PDAs. The overall communication architecture is sketched below and depicted in Figure 2; the interested reader will find a more detailed description in [2].

The base communication layer is TCP/IP that runs on wired or wireless network interfaces present at every element in the system. Most elements also run the Delay-Tolerant Networking overlay provided by the DTN reference implementation [4]. This provides disconnection-tolerant end-to-end connectivity. We modified the DTNRG DTN 2.0 reference implementation to add flooding-based routing within each region.

Although DTN provides disconnection-tolerance, it lacks many important services. It does not provide the ability for a kiosk-controller, cell phone, or proxy to use application-specific policies to choose from one of many network interfaces, nor does it support mobility for users who may choose to move from one kiosk to another. It does not provide application-specific plugins at the proxy. Finally, DTN does not support seamless interconnection with legacy servers.

These capabilities, instead, are provided by the opportunistic connection management protocol or OCMP [5]. Each type of available communication path is modeled as a connection object (CO) in OCMP. For instance, DTN is encapsulated as a DTN CO. There are similar COs for a TCP connection bound to each type of NIC (GPRS, EDGE, WiMAX, dial-up etc.). We also support a CO for an 'SMS NIC', which allows communication over an SMS channel. OCMP allows a sophisticated policy manager to arbitrarily assign bundles to transmission opportunities on COs.

OCMP works in conjunction with the TCA-admin component, which is responsible for mobility management. Each user has a hierarchical GUID, which is in the form shown in Figure 2. The TCA-Admin component registers this GUID with a DNS-based back-end every time a user changes location. A gateway queries this back end, using standard DNS resolution to determine the peer gateway where it should send a bundle.

Note that identical Java-based OCMP protocol stacks run on cell phones and kiosk controllers. The only difference is that the DTN protocol stack runs only on kiosk controllers, and not on cell phones. This is because the DTNRG reference implementation, which is written in C++, cannot run on a cell phone. If a Java-based DTN implementation becomes available for cell phones, the cell phone protocol stack can be made the same as that on the kiosk controller.

Besides these infrastructural components for network communication, the KioskNet software suite supports considerable additional functionality. This includes support for user account management, creation and dissemination of per-user public and private keys, a disconnection-tolerant shell (similar to [6]), an application-specific cell-phone based control channel between the proxy and each kiosk, tools for automatic database synchronization between a proxy and each kiosk controller, support for email at each kiosk, and support for application-specific plugins that allow content generated at a kiosk to be uploaded to blogger.com and flickr.com. These are described in more detail on the KioskNet website [7].

4. SECURITY

KioskNet's security architecture is designed to meet the requirements of four distinct groups:

- *KioskNet Franchisers*: Franchisers, usually non- governmental organizations (NGOs) deploying KioskNet, are concerned with the integrity of their infrastructure nodes (gateways, mobile routers, kiosk controllers and proxies) and would want to detect, if not prevent, the misuse of their infrastructure.

- *KioskNet Franchisees*: Franchisees (i.e. kiosk operators) are concerned with the security of their kiosk terminals and would want protection against malware and also would want to prevent any attacks launched through KioskNet.

- *KioskNet Users*: Users are concerned with the confidentiality and integrity of their data despite using untrusted ferries and snooping kiosk operators.

- *Application Service Providers*: Depending on the type of service they provide, ASPs may want franchisers to guarantee the integrity of their software when deployed on a KioskNet. Examples of such software might include tax payment and land registry systems operated by the government or microfinance banking applications.

We satisfy these requirements through a combination of standard cryptographic techniques such as PKI and more recent developments such as transparent encrypted file systems. For instance, to enable the authentication and end-to-end encryption of in-flight data, all the entities named above are issued unique credentials including a 2048-bit RSA private key and a corresponding public key certificate.

Certificates are issued and signed in a hierarchical fashion, forming chains. A secure central root CA server at the University of Waterloo issues certificates to franchisers, who then issue certificates to franchisees and ASPs operating in their region. Franchisees can then certify users registered at their kiosks. Similarly, all infrastructure nodes are issued unique credentials by the franchisers that maintain them. Public key certificates for users, franchisees and ASPs are periodically broadcast throughout a franchiser's region through the use of a Whitepages public key database maintained at the proxy and replicated at all kiosk controllers.

The security of KioskNet infrastructure is ensured through the use of digital signatures on all remote commands and software updates issued by franchiser administrative personnel. Further, franchisees are not given root access to deployed kiosk controllers, preventing them from modifying the software on these systems. An encrypted root directory

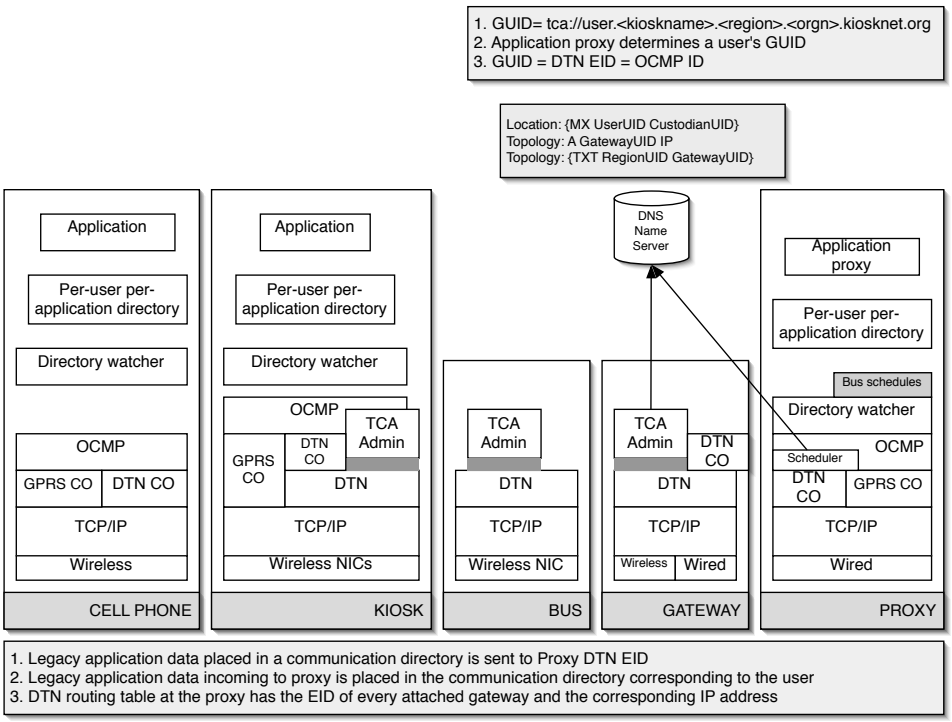

Figure 2: Software architecture.

on each infrastructure node prevents attackers from removing the device's hard disk and booting it with a Live CD to access the node's private key. Industry-standard practices such as the use of intrusion detection systems and firewalls can be additionally used to protect infrastructure nodes against remote attack through their network interfaces.

Rural kiosks operated by franchisees are protected against viruses and other malware by being forced to boot from read-only disk images stored in tamper-proof kiosk controllers. Because only franchiser administrative personnel are permitted to update these disk images, franchisees can be assured of the integrity and security of the operating system and applications running on their kiosks.

The measures taken to protect rural kiosks described above also provide ASPs with assurance of the integrity of the platform their applications are deployed on. Additional security can be provided by ASPs issuing signed certificates for their application binaries, allowing users and franchisees to verify their integrity as required.

User data stored in kiosk controllers is secured by creating encrypted virtual volumes for each user's home directory keyed with the user's kiosk login password. These volumes are stored in encrypted form on the kiosk controller and exported over NFS for mounting at kiosk terminals when users login with a valid password. Linux's Pluggable Authentication Module (PAM) is used to automate the decryption of these volumes when users login and their encryption when users logout. Users can transparently read and write to their encrypted home directories through our use of the Linux DM-Crypt disk encryption module. Because user data, including private keys, is stored in these encrypted home directories, even attackers with root access are unable to view or modify the data.

In-flight user data that requires privacy and authenticity is encrypted and signed at kiosk terminals before it is transferred to the kiosk controller for forwarding to other KioskNet infrastructure nodes along its way to the proxy. This ensures secure user data cannot be read, fabricated or tampered with while in transit within KioskNet.

When combined, the security measures described above serve to protect KioskNet against a diverse set of attacks, ranging from simple wireless packet sniffing to more sophisticated attacks that involve removing an infrastructure node's hard disk and booting it with a Live CD to gain root access and read or modify the data stored in it.

More details of this solution can be found in [8].

5. COST STRUCTURE

By design, our solution is extremely low-cost. For instance, we estimate that to provide minimal connectivity to a population of about one million people will require a total capital expenditure of only about $300,000 or 30 cents/person. More extensive coverage will probably cost ten times as much, but still less than a one-time cost of five dollars a person.

We now present some cost figures. These figures are merely indicative because much depends on the actual deployment environment, and issues such as the rate of interest for small business loans, the import duty rate on electronics, and purchase volumes.

Using off-the-shelf technology, the cost of an average kiosk (which does not require solar power) would be about $450. The main costs at a kiosk are for a single-board computer

(such as a Soekris net4501with an 802.11a/b/g mini-PCI Atheros wireless card) which costs about $250, for power remediation (using car batteries), which costs about $100, and for a $100-recycled PC. Note that this cost would be lower with volume purchases. Moreover, the cost of a single-board computer will be lower if local single-board computer manufacturers can be found, or if the single-board computer is replaced with an XO laptop [9]. On the other hand, costs can be higher if there is need for solar cells (which cost around $150), or high-power external antennas, which can add another $250 to the cost.

Assuming an initial capital expenditure of $450, the operational expense, including the cost of field technicians and capital depreciation on an 18-month schedule is about $65/month. The main costs are for a field technician, who can service about 20 kiosks, and the cost of capital depreciation. Even assuming 10% penetration of a target population of 2500 users, with a service charge of $3.00 a year, an operator can break even. Additional profit can be generated by charging more per user, by increasing penetration, or by offering additional services, such as computer literacy or digital photographs.

6. PILOT DEPLOYMENT

One of us (Seth) deployed a prototype of our solution in Anandapuram, a village in South India, during the week of May 16th, 2006. Each kiosk already had a Windows XP PC. We deployed a Soekris net4801 at the kiosk, with a 40 GB Toshiba hard disk drive for local storage. The system was connected to a roof-mounted omnidirectional antenna.

Power came from a 42 AH deep discharge car battery that was charged by two 1200 mA (12V) Powerflex solar panels mounted on the roof of the kiosk. We could also have run our system from AC mains and relied on battery or solar power only for backup.

In the car (see Figure 3), we used power from the car battery, but through an inverter and the Soekris power supply, to mitigate against voltage spikes. The car had a magenetically mounted omni-directional antenna.

The gateway was in Vishakapatnam. Because the van was parked below the computer room, it was necessary to place the omni antenna outside the building. Figure 3 is a composite figure showing the deployed system.

The purpose of the pilot deployment was to gain confidence in the physical system (antennas, power supplies, single board computers) and their ability to operate with minimal infrastructure and in poor operating conditions – temperatures in the van reached almost 50 C! The software infrastructure in the pilot, though, was not well tested. In the last year, we have thoroughly stress-tested every component of the system, and we released a robust implementation on July 20, 2007. We plan to release the security, SMS, and DNS components of our system in the fourth quarter of 2007.

7. DISCUSSION

When we started work on this project in 2005, we made several speculative design decisions. Based on our experiences with the prototype deployed in the field, we subsequently changed our mind regarding several key architectural components. This section describes the changes we had to make, and why. We hope that these experiences will help other groups heading down similar paths.

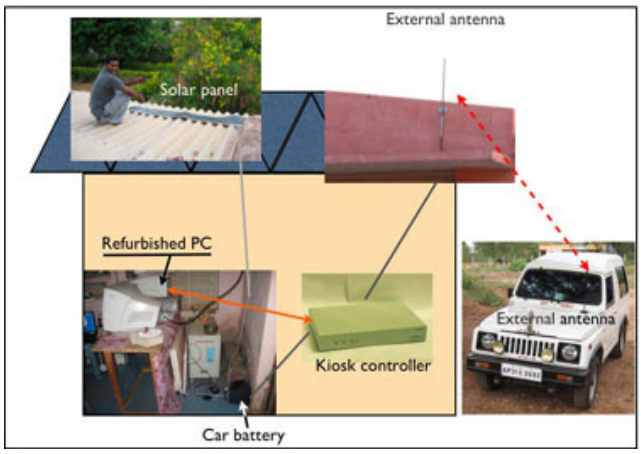

Figure 3: Composite picture of the pilot deployment.

7.1 IBC vs. PKI

The initial design of the system provided privacy by means of Hierarchical Identity Based Cryptography (HIBC) [10]. This allows a kiosk user to send authenticated and encrypted messages to another user without the need to know that user's public key. Although academically interesting, using IBC turned out to be problematic in practice. IBC is essentially controlled by a single entity (Voltage Inc), which does not release source code and has stringent licensing conditions for commercial use. We therefore decided to replace HIBC with our own PKI. There is a wide assortment of open-source tools available for PKI, and we were able to use them to build our own PKI in a matter of a few developer-months.

7.2 Flat names and DHT vs. Hierarchical names and DNS

Our initial design used flat names and a DHT as a Home Location Register to keep track of user location. Again, although this is academically interesting, we found that the DHT we used (OpenDHT) was both slow and unstable. Moreover, OpenDHT is hosted on PlanetLab nodes that are not found in most developing countries. From a technical perspective, a DHT does not allow us to delegate location management for sets of users to third parties. We therefore decided to use hierarchical names for users (of the form user.kioskname.regionname.organizationname.kiosknet.org). This allowed us to use stable, well-tested, and fast off-the-shelf DNS implementations for location management - the location of a user is just an MX record that points to its kiosk. We can also delegate part of the name space to the organization responsible for a deployment. We think that these two benefits more than compensated for the loss of a flat name space and an infinitely-scaleable DHT.

7.3 Mechanical backhaul vs. Use of all interfaces

When we started our work, we assumed that the only way to reach a kiosk would be using mechanical backhaul. In fact, kiosks are increasingly being reached by GPRS, and soon, will also have WiMAX coverage. Therefore, we de-

cided to support a wide variety of connectivities, with mechanical backhaul reserved for slow and delay-tolerant data. It turns out that using SMS for a control channel brings numerous benefits, such as allowing us to detect ferry failures, and to alert kiosks to turn on their WiFi interface in anticipation of a ferry arrival. We believe that this acceptance of multiple-connectivity makes our system more widely applicable..

8. RELATED WORK

Our work is most closely related to, and was inspired by, the pioneering work by Daknet [3, 11]. However, we differ from Daknet in several ways. To begin with, Daknet focuses only on communication, but KioskNet also supports a computing platform based on recycled PCs. KioskNet leverages DTN for disconnection tolerance, and adds PKI for privacy, confidentiality, and integrity. Moreover, KioskNet supports multiple network interfaces at each kiosk.

KioskNet's goal of low-cost Internet access is shared by the pioneering work by CorDECT [12] and two well-known long-range wireless projects- Digital Gangetic Plains Project [13] and WildNet [14]. These are essentially communication technologies, and can be integrated into KioskNet (as connection objects) with little effort. With KioskNet, mechanical backhaul can be used to supplement long-range wireless for delay-insensitive data, such as video content distribution, email, and database updates.

The use of mechanical backhaul has also been studied in pioneering work on data ferrying [15], and recent work on DieselNet [16]. However, the focus of these projects has primarily been on routing - instead, we take a whole-systems perspective for the specific purpose of rural connectivity.

9. CONCLUSIONS

Rural communities worldwide can benefit from low-cost Internet access. KioskNet attempts to meet this need by leveraging mechanical backhaul. However, to build a usable and useful solution, focusing only on the communication path is inadequate. Our solution, therefore, provides not only communication, but also a variety of related components, such as security, user management, and log collection.

On another note, our first design was academically sound, but far too complex in practice. By carefully examining the problem, and using well-tested existing solutions, we have been able to dramatically reduce complexity but without much reduction in functionality. We suspect that many other academic projects could benefit from a similar technical re-appraisal.

10. REFERENCES

[1] K. Toyama, "Review of Research on Rural PC Kiosks." [Online]. Available: http://research.microsoft.com/research/tem/kiosks/Kiosks%20Research.doc

[2] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2006, pp. 334–345.

[3] "United villages," 2007. [Online]. Available: http://www.unitedvillages.com/

[4] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing Delay Tolerant Networking," *Intel Research, Berkeley, Technical Report, IRB-TR-04-020, Dec*, 2004.

[5] A. Seth, M. Zaharia, S. Keshav, and S. Bhattacharyya, "A policy-oriented architecture for opportunistic communication on multiple wireless networks," 2006. [Online]. Available: http://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/06/ocmp.pdf

[6] M. Lukac, L. Girod, and D. Estrin, "Disruption tolerant shell," *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, pp. 189–196, 2006.

[7] "The kiosknet project," 2007. [Online]. Available: http://blizzard.cs.uwaterloo.ca/tetherless/index.php/KioskNet

[8] S. U. Rahman, "Kiosknet security," 2007. [Online]. Available: http://blizzard.cs.uwaterloo.ca/tetherless/index.php/Security_Architecture_Overview_%28draft%29

[9] "One laptop per child (olpc)," 2007. [Online]. Available: http://www.laptop.org/

[10] A. Seth and S. Keshav, "Practical Security for Disconnected Nodes," *Proceedings of First Workshop on Secure Network Protocols (NPSEC)*, 2005.

[11] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, no. 1, pp. 78–83.

[12] A. Jhunjhunwala, B. Ramamurthi, and T. Gonsalves, "The role of technology in telecom expansion in india," *Communications Magazine, IEEE*, vol. 36, no. 11, pp. 88–94, 1998.

[13] "Ruralnet (digital gangetic plains: Dgp) 802.11-based low-cost networking for rural india," 2007. [Online]. Available: http://www.cse.iitk.ac.in/users/braman/dgp.html

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

[15] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 187–198, 2004.

[16] "Umass dieselnet," 2007. [Online]. Available: http://prisms.cs.umass.edu/dome/index.php?page=umassdieselnet