KioskNet - Low-cost Internet Access for Developing Regions

M.H. Falaki, C. Ho, U. Ismail, A. Leong, R. Luk, E. Oliver, S. Ur Rahman, and S. Keshav
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Categories and Subject Descriptors: C.2: Network Architecture and Design

General Terms: Design, Reliability, Security

1. OVERVIEW

The current connectivity technologies for rural kiosks in developing countries, such as dial-up, Very Small Aperture Terminals (VSAT) and long-distance wireless, tend to be both expensive and failure prone. We propose the use of buses and cars as "mechanical backhaul" devices to carry data to and from a village and an Internet gateway. Building on the pioneering lead of Daknet [2], we have designed a comprehensive solution, encompassing naming, addressing, forwarding, routing, identity management, application support, and security [1]. We believe that this architecture not only meets the top-level goals of low-cost and robustness, but also exposes fundamental architectural principles necessary for any such design.

Although kiosk controllers can communicate with the Internet using a variety of connectivity options, our focus is on the use of mechanical backhaul, which is provided by cars that pass by a kiosk and also an Internet gateway. We call such entities ferries. A ferry has a small, rechargeable, battery-powered computer with 20-40GB of storage and a WiFi card. It opportunistically communicates with the kiosk controllers and Internet gateways on its path.

Ferries upload and download data opportunistically to and from an Internet gateway, which is a computer that has a WiFi interface, storage, and an always-on connection to the Internet. The gateways are likely to be present in cities having 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 proxy on the Internet. It also downloads data from the proxy on behalf of kiosk users, and transfers them opportunistically to the appropriate ferry, governed by a routing protocol. In our implementation, we use a Soekris computer both for ferries and for gateways.

Existing Internet services are typically unable to deal with delays and disconnections. Therefore, we propose the use of a disconnection-aware proxy that hides disconnection from legacy servers. The proxy is resident in the Internet and essentially has two halves. One half establishes disconnection-tolerant sessions with applications running on a recycled PC or on a mobile user's device. The other half communicates with legacy servers. Data forwarding between the two halves is highly application dependent. To support application-specific protocols, we allow applications to instantiate an application-specific plug-in at the proxy.

Finally, the last component of our architecture are the legacy servers that are 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.

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

2. VIDEO APPLICATION

We demonstrate the operation of the KioskNet system through a disconnection tolerant video application called *oTube*. oTube allows KioskNet users to upload and download videos to the Internet without an end-to-end connection to any server on the Internet. oTube consists of two components: an application that runs on the kiosk and a plug-in at the proxy. The kiosk component provides the user with a graphical user interface for requesting content from an online video sharing site. The kiosk component also allows users to record videos, which may be uploaded to the Internet. When a download or upload is initiated by the user, oTube communicates with a local Opportunistic Communication Management Protocol (OCMP) daemon running on the kiosk. OCMP encapsulates the oTube data as *bundles*, which are stored in persistent storage to survive hardware failures or shutdowns while waiting for a ferry. When a passing ferry is detected, OCMP transfers the opaque oTube data to the ferry. Like the kiosk, the ferry also stores the data in persistent storage. When the ferry is within range of an Internet gateway, the data is transferred to the proxy. On the proxy, the OCMP daemon invokes the oTube plug-in, which uses the data contained within the bundle to communicate with an Internet video site. After a successful transaction, the proxy sends an acknowledgment all the way back to the user via the gateway, the ferry, and the kiosk controller, in the same fashion.

3. REFERENCES

[1] S. Guo, M.H. Falaki, E. Oliver, S. Ur Rahman, A. Seth, M. Zaharia, U. Ismail, and S. Keshav. Design and implementation of the kiosknet system. In Proceedings of the International Conference on Information and Communication Technologies and Development (ICTD 2007), pages 300–309, 2007.

[2] A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. Computer, 37(1):78–83, 2004.