

Energy-Efficient Data Transfer Primitives for Laptops Using Mobile Handhelds

Adishesu Hari, Yuh-Jye Chang, Dimitrios Stiliadis

Alcatel-Lucent Bell Laboratories

Murray Hill, NJ 07974, USA

{Adishesu.Hari, Yuh-Jye.Chang, Dimitri.Stiliadis}@alcatel-lucent.com

Andrea Francini

Alcatel-Lucent Bell Laboratories

Mooresville, NC 28115, USA

Andrea.Francini@alcatel-lucent.com

ABSTRACT

We introduce a novel mechanism for content distribution to large numbers of weakly connected laptops that can be switched off frequently and have intermittent network access. Relying on the user's data-enabled mobile phone and a gateway added to the data path between the laptop and the Internet, the mechanism builds upon two novel data-transfer primitives that efficiently move files across the network even when the laptop is switched off or sleeping, in a way that is fully transparent to the application layer. One primitive targets network-folder-based applications, while the other works for web-based applications. The primitives have been successfully deployed in the field as part of a solution for remote IT management of mobile-employee laptops.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: Data communications.

General Terms

Algorithms, Design, Measurement, Performance.

1. INTRODUCTION

We consider the problem of content distribution between a server node and a large number of loosely connected end devices. An Internet data transfer between an end system and a remote server typically relies on the end system being both powered up and connected to the Internet for the entire duration of the transfer. This requirement causes difficulties when the end system is up only intermittently and network connectivity is sporadic, as is the case for mobile users who can go without a fixed Internet connection for long periods of time. Several attempts have been made to address this problem in the context of *disruption tolerant networks* [11], but their focus is mostly on point-to-point communications.

We present a new mechanism for the distribution of data from one source to a multitude of mobile devices, which can be deployed today at massive scale as an overlay on top of current (3G/4G) wireless technologies. Our goal is to

utilize the extended connectivity of mobile handsets to expedite the distribution of content to computing devices, such as laptops, that are more powerful computationally, but also harder to keep powered on and connected in high-mobility usage scenarios.

Consider, for example, mobile professionals who need to keep their laptops synchronized with their enterprises while spending most of their time on the road. Although it is crucial for an IT organization to keep these laptops up-to-date with patches and software releases, the synchronization is challenging because a laptop can spend long time without connectivity to the enterprise. Any software updates or patches distributed by the IT organization during this time will not reach the laptop. Indeed, it has been documented that a critical patch might take up to 30 days to be deployed in 90% of the enterprise laptops. Given the closing time gap between the discovery of software vulnerabilities and their exploitation, incomplete patch deployments expose the enterprise to significant security risks. Besides patch distribution, several other content-delivery applications suffer from the laptop's sporadic reachability, such as RSS feeds, e-mail, and multi-media content distribution. *We seek an efficient way of transferring application data to and from a large number of mobile laptops that have intermittent uptime and Internet connectivity.* We remark that we cannot assume that all the end-user laptops are concurrently connected to the network at any given time.

The dominant solution for mobile users to stay continuously connected resorts to *wireless wide area network* (WWAN) access technologies, typically by means of a 3G/4G data card. WWAN connectivity, however, does come with its own set of problems. In usual client-server applications, the connection that supports a data transfer must be initiated by the application client, irrespective of whether the data source for the transfer is the server or the client. In applications like patch management, email, RSS feeds, and other forms of data download, the client polls the server periodically. The polling paradigm wastes energy in the laptop, significantly shortening the battery life, and radio resources in the access network. Specifically, the periodic polling activity prevents the laptop from utilizing sleep states and dormancy efficiently in order to reduce energy consumption. In addition, all WWAN technologies allocate radio bearers for IP traffic

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHeld 2010, August 30, 2010, New Delhi, India.

Copyright 2010 ACM 978-1-4503-0197-8/10/08...\$10.00.

only during active data transfers. Whenever a dormant endpoint sends a poll request, this causes the radio bearers to be reallocated and then released shortly thereafter, leading to inefficient spectrum usage and potential overloads on the signaling path of the network. Last but not least, regular WWAN data cards only operate when the host laptop is powered on.

What is needed is a solution for data transfers to and from mobile laptops that not only ensures connectivity, but also uses energy and spectrum efficiently. The solution should transform what have been traditionally *pull* applications into *push-on-demand* applications. In the interest of seamless deployment, the solution should be designed to coexist with legacy applications and networks. This requires an application- and network-agnostic approach, without any undue programming changes at either the client or the server component of any application. The *DataAccel* concept that we present in this paper embraces all the defining elements of the ideal approach. The *DataAccel* concept and its two data-transfer primitives, called *FileAccel* and *WebAccel*, have been successfully applied to a commercial-grade platform [6] that enhances critical IT applications like patch management, data backup, configuration management, and content distribution.

The rest of the paper is organized as follows. In Section 2 we describe the *DataAccel* primitives. In Section 3 we present experimental results on data transfer latencies and energy consumption. We review related work in Section 4 and draw conclusions in Section 5.

2. DATAACCEL PRIMITIVES

2.1 Overview

We present here the *FileAccel* and *WebAccel* primitives for expediting data transfers to and from laptops with sporadic access connectivity. Both primitives rely on the support of two added infrastructural components between the laptop and the servers it communicates with. We call these components, shown in Figure 1, the *DataAccel handheld* and the *DataAccel gateway*, or shortly, the handheld and the gateway.

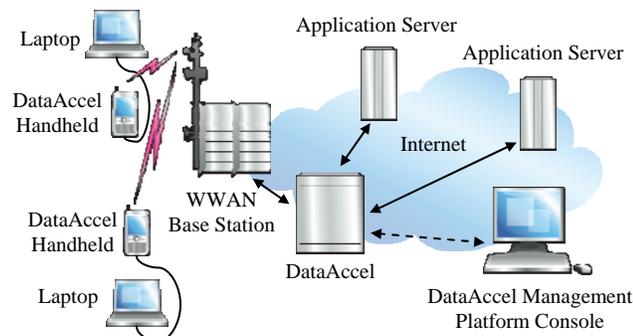


Figure 1. *DataAccel* infrastructure.

The handheld can be a PDA, a smart phone, or simply a cell phone, as long as it features a WWAN interface with

high-speed data capabilities. The battery, processor, persistent storage facilities, and WWAN interface of the handheld make it capable of autonomous operation and data transfers at times when the laptop is powered off. It can also be a 3G/4G data card, provided that it is capable of autonomous operation even when the attached laptop is down or sleeping (the card must include a battery, a processor, and a data storage facility [9]). While use of the handheld's WiFi interface is admitted, a USB cable is our preferred medium for attachment of the handheld to the laptop, because it provides a secure, configuration-free, high-bandwidth data link with an embedded power supply. The *DataAccel* functionality in the handheld is implemented entirely in software and does not require a custom hardware platform.

The gateway is an off-the-shelf network appliance with custom software and is located in the network of the enterprise or service provider that manages the primitives. It mediates access to the application servers on behalf of the laptops. The gateway relays content between the application servers and the handheld over a communication channel to the handheld that can be secured (e.g., using IPsec) and that both the handheld and the gateway can initiate. The same communication channel also provides connectivity to the handheld and associated laptop for the management platform that provisions, maintains, and monitors the primitives. The gateway offers a single point of access to the data path for the execution of all management functions and conveniently excludes the application servers from the perimeter of the *DataAccel* solution. Without the gateway, *DataAccel* software supporting the *FileAccel* and *WebAccel* primitives would have to be installed in every independent source of application files.

The *FileAccel* primitive executes folder-based data transfers. The *WebAccel* primitive executes web-based data transfers. Both primitives support time-shifted, reliable download data transfers. The *FileAccel* primitive also supports time-shifted, reliable upload data transfers. Time shifting is useful for increasing the utilization of wireless bandwidth in off-peak hours, since it allows the server to schedule the delivery of content at times when the network might not be highly utilized. By offloading the data transfers to the handheld, the primitives reduce the consumption of energy in the laptop, because they prevent it from having to remain powered on for extended periods of time only to make those transfers possible.

2.2 FileAccel

The *FileAccel* primitive targets applications that transfer data using shared network folders or drives, such as Microsoft Systems Management Server (SMS). The *FileAccel* primitive reduces the completion time of file exchanges between the laptop and the application servers in the Internet core, equally supporting file exchanges in the uplink direction (from the laptop to the application

server) and in the downlink direction (from the application server to the laptop). The key idea behind FileAccel is to replace the direct file transfer between the application server and the laptop with a more efficient three-step transfer process: between the application server and the gateway, between the gateway and the handheld, and between the handheld and the laptop. Note that no code changes are required at the application server in the network or at the application client in the laptop. The application server must be configured to allow the gateway to exchange data with its network folder, while the application client must be configured to fetch and store its data from/into a laptop folder, rather than a network folder.

Without FileAccel, a file exchange between the laptop and the application server can complete only if during the entire time needed to transfer the file the laptop is powered on (*laptop availability condition*) and connected to an access network (*laptop connectivity condition*). With the FileAccel primitive, the exchange can complete even if the laptop availability condition and the laptop connectivity condition are not satisfied simultaneously.

The FileAccel primitive relies on dedicated software modules installed in the laptop, in the handheld, and in the gateway. The modules handle the file exchanges between the application server and the laptop as the combination of file exchanges over the server-gateway, the gateway-handheld, and the handheld-laptop connectivity segments. The handheld and the gateway can exchange files whenever the handheld is powered on and connected to an access network, independently of the power state of the laptop. The laptop and the handheld can exchange files whenever they are both powered on and connected with each other, independently of the availability of an access network that enables connectivity to the gateway.

In our implementation of FileAccel, the dedicated software modules are off-the-shelf folder synchronization clients (*rsync* [7]). The laptop and the handheld allocate one directory per transfer direction to applications in a configurable set. The gateway allocates one directory per application in the downlink direction and one directory per application per laptop in the upstream direction. In every node, the folder synchronization client cooperates with its peer in the adjacent node to keep the respective directories synchronized. In the laptop and handheld the folder synchronization client removes a file from the respective directory as soon as its transfer to an adjacent node is completed. In the gateway, since the same downlink directory serves multiple handhelds, an aging mechanism controls the removal of files from those directories.

When an application server makes a new file available at the gateway, copies of the same file are quickly transferred to the handheld instances that are powered on and have network connectivity. Every other handheld receives the file as soon as it establishes connectivity with the gateway, provided that the aging timer of the file has not expired.

For handheld instances that are within WWAN coverage but in sleep state, the gateway can dispatch *short message service* (SMS) messages that wake up the handhelds without changing the power state of the laptop. The handheld can then receive the file as it establishes direct connectivity with the gateway. This way the handheld is relieved from maintaining persistent connectivity with the gateway to ensure the prompt delivery of new data. Once a file has reached the handheld, it passes to the laptop as soon as the latter powers on.

For transfers in the opposite direction the sequence of intermediate steps is almost identical, with only one difference in the immediate removal of the new file from the dedicated directory in the laptop immediately after the file is successfully transferred to the handheld.

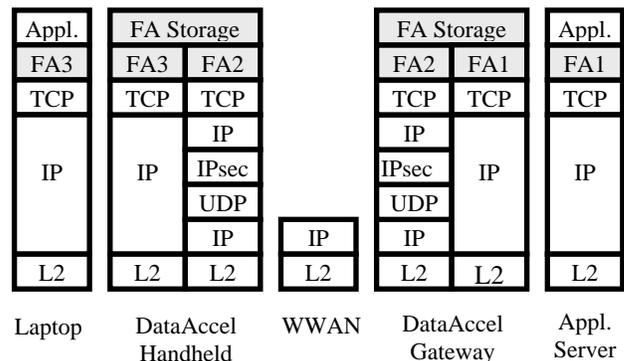


Figure 2. Protocol stacks with FileAccel primitive.

Figure 2 illustrates the protocol stacks in the laptop, handheld, wireless wide-area network (WWAN), gateway, and application server when the data transfer between laptop and server is executed with the support of the FileAccel primitive. The figure emphasizes the time-shifted nature of the data transfers between laptop and application server by depiction of three distinct FileAccel connectivity segments (FA1 between gateway and server, FA2 between handheld and gateway, and FA3 between laptop and handheld) along the data path. We have ruggedized the FileAccel implementation against storage corruption and overflow by applying checksums to the transferred data.

2.3 WebAccel

The WebAccel primitive reduces the time needed to complete the web-based distribution of a file to a known population of laptops. While the FileAccel primitive handles file transfers based on shared folders in both directions, the WebAccel primitive handles web-based file distributions, where multiple clients, residing in their respective mobile laptops, retrieve copies of the same file from a common application server. Without WebAccel, each laptop would independently request the file from the server and each individual file download would only be completed at times when the laptop availability condition and the laptop connectivity condition are both satisfied.

Web caching can be used to effectively reduce the resources and time needed to complete the distribution of a file to the fixed set of laptops. However, traditional web caching does not grant exemption from the requirement of simultaneous satisfaction of the laptop-availability and laptop-connectivity conditions. The WebAccel primitive, on the contrary, successfully attains such exemption by combining traditional web caching (with storage of a file copy in the DataAccel gateway) with a form of proactive caching that distributes copies of the file to the connected DataAccel handhelds independently of the power states of the respective laptops. The WebAccel primitive elects the handhelds as mobile caches, ready to pass the new file to the respective laptops as soon as they power up.

```

if (file downloaded by laptop) {
    remove file from handheld cache;
    mark file entry in local file list as "delivered";
}
if ((new file list advertised) or
    (request timer expired)) {
    set request timer;
    request new file list from gateway;
}
if (new file list received) {
    if (entries in local file list are not in new file list) {
        remove entries from local file list;
    }
    if (entries in new file list are not in local file list) {
        add entries to local file list;
        request from gateway files of new entries;
        if (no space in handheld cache for new files) {
            remove oldest files from handheld cache;
        }
    }
}

```

Figure 3. Pseudo-code for handheld operation of WebAccel.

The WebAccel primitive builds on dedicated software modules installed in the gateway and in the handheld. The WebAccel software in the gateway consists of off-the-shelf software for standard web-cache functionality, augmented with code for maintaining, advertising, and distributing to the DataAccel handhelds the list of files currently stored in the gateway cache. The gateway cache adds a new entry to the list the first time it receives a request for the corresponding file from one of the handhelds. Only requests associated with a configurable set of applications generate new entries in the file list. The proactive advertisement of the file list to the handhelds at every change of the list contents constitutes the main element of novelty in the DataAccel gateway versus ordinary web-cache software.

The handhelds include off-the-shelf cache software, functionally extended to maintain local copies of the list of

cached files. Every time the handheld recognizes from a list update that it must add a new entry to its version of the file list, it issues a request for the corresponding file to the DataAccel gateway. After receiving a file from the gateway, the handheld cache holds it until the laptop retrieves it, either immediately or right after powering up. For proper interoperation with the WebAccel primitive, the laptop only needs to have its application clients configured with the handheld as their cache proxies.

When the first laptop in the managed set requests a file for one of the applications under WebAccel coverage, the handheld first relays the request to the gateway, and then the gateway forwards it to the target server. Upon receiving the file from the server, the gateway stores it in the cache and advertises the resulting update in the file list to all handhelds that are currently reachable. The gateway then dispatches the file list to all handhelds that subsequently request it. Finally, the gateway fulfills every request for a file copy that it receives from the handhelds. The transfers of the file copies to all handhelds that are currently reachable occur independently of the power state of the associated laptops, and typically before the laptops generate explicit requests for the files. For handhelds that are in sleep state when the update of the file list is first advertised, the gateway can dispatch SMS messages over the WWAN infrastructure to wake them up and have them issue their requests for the updated file list. The pseudo-code of Figure 3 presents algorithmic elements of the WebAccel operation in the DataAccel handheld.

3. PERFORMANCE

3.1 Data Transfer Speed

Both FileAccel and WebAccel stage the transfer of data from the network to the laptop over three hops. For the hop between the application server and the gateway, the transfer completion latency is the sum of the delay of the gateway in detecting the availability of new content at the server and the data transfer time from the server to the gateway. In our current implementation, the gateway polls the server once every two minutes. If the application server is inside the same domain or enterprise as the gateway, the data transfer is usually almost instantaneous. If the application server is in a public network, it can rely on high-speed Internet connectivity that enables download rates in the megabit-per-second range.

The hop between the laptop and the handheld is over the USB link, which is operative when both of its endpoints are up. While USB 2.0 has a theoretical data transfer capability of 480 Mbps, the actual data transfer rate is very much below this limit and depends on the *communications device class* (CDC) emulated over USB. We use the Ethernet USB CDC to expose the handheld-to-laptop USB link as a point-to-point Ethernet link.

The hop from the gateway to the handheld includes the wireless link that anchors the handheld to the WWAN. The

data transfer time is made of two components. First, the gateway needs to signal the availability of new content to the handheld, so that the handheld can initiate its downlink transfer. If the handheld is up, the gateway sends the new-content notification directly to the handheld. If the handheld is down, the gateway dispatches an SMS message, which takes a variable amount of time to reach the handheld, depending on the network operator and load. In our use of the FileAccel and WebAccel primitives we have observed SMS message delays ranging from a few seconds to many hours, though tens of seconds is the typical measure. The second component of the transfer time is the transmission time over the relatively slow wireless link, which depends on the specific WWAN technology supported by the handheld, on the data plan of the end user, and on the network load, handheld position, and local weather at the time of the transfer.

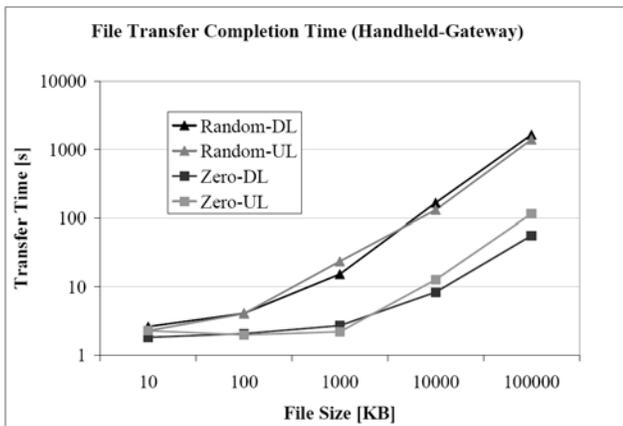


Figure 4. File transfer time between handheld and gateway.

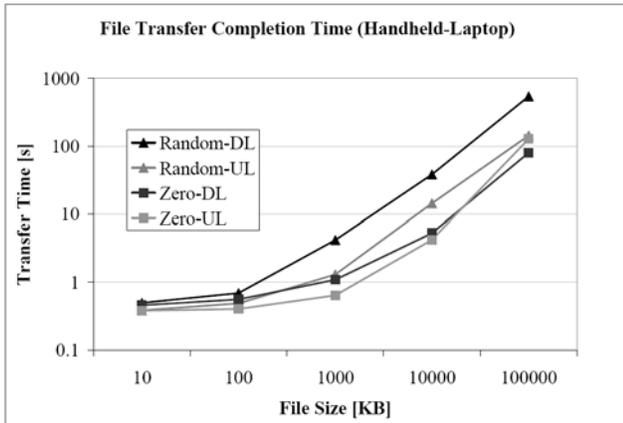


Figure 5. File transfer time between handheld and laptop.

We present performance data obtained from the implementation of the primitives in commercial handhelds using the Android platform [3]. In Figure 4 we plot downlink and uplink transfer times for the FileAccel primitive over the hop that includes the bottleneck wireless link (we use an HSPA link in the WCDMA 1900 MHz band). We display equivalent data for the USB link between handheld and laptop in Figure 5. We transfer two types of files with multiple sizes (from 10 KB to 100 MB).

The first file type, called *Random*, contains random data, while the second type, called *Zero*, contains all zeros. As can be seen in Figure 4, in both directions the transfer of random files takes much longer to complete than the transfer of zero files. This is because we have data compression enabled in the *rsync* utility that supports the FileAccel primitive. Data compression enables faster transfers for files with compressible content like text documents. Compression also minimizes the usage of wireless bandwidth, which is highly beneficial with data plans that charge per byte transmitted or place caps on data transfer volumes. Figure 5 also shows faster file transfers for the *Zero* files versus the *Random* files. In this case the uplink speeds generally exceed the downlink speeds. This is because the laptop, with a more powerful processor than the handheld, can sustain higher data compression throughputs. The only exception is the 100 MB *Zero* file, whose upload speed is limited by the slower write speed into the *secure digital* (SD) card of the handheld.

3.2 Energy Consumption

With our primitives the handheld can exchange data with the network when the laptop is powered down. As a result, the laptop can always be turned off when not in use, e.g., at night or during breaks or meetings. It is not even necessary to power on the handheld, as long as it can be woken up with an SMS message. As a result, the end user can power up the laptop only for interactive use of its applications.

In our measurements, the handheld power consumption in sleep mode ranges between 10 mW and 20 mW, depending on signal strength conditions and wireless network configuration. At peak data-transfer rates, the power consumption is only 2 W. On the other hand, we measured between 18 W and 25 W of idle power for most laptops with Intel Centrino Duo technology. If we assume an average idle power of 20 W per laptop and 8 hours per work day of power-off or standby time, our primitives can save at least 160 Wh per work day and 92 kWh per year.

4. RELATED WORK

4.1 Industry

Many tools exist for handheld-to-desktop synchronization (e.g., Microsoft ActiveSync for Windows-based products), but they target a different use case than our primitives. They are in fact designed to synchronize peers, i.e., to move content between an application in the desktop and the same application in the handheld. Thus, they require instances of the same application to exist at both ends. Furthermore, the synchronization is application-specific: the application must be modified to plug into the *application programming interface* (API) of the synchronization tool. With both FileAccel and WebAccel, there is no need to implement the application in the handheld, nor is there any programming requirement at either the laptop or the handheld.

With the Intel Active Management Technology (iAMT) for mobile and desktop computers, an auxiliary processor remains in active state while the main processor is sleeping, so that data can be downloaded to the computer without waking up the main processor. This technology is specific to Intel processors and only works with Ethernet and WiFi networks, not with WWAN interfaces, and is therefore unsuitable for mobile data download. Also, no data transfer primitives are provided in the auxiliary processor. Our solution does not have these limitations.

4.2 Research

Various studies [4, 10] have confirmed that hosts can achieve substantial energy savings by aggressive use of low-power sleep states. The *Somniloquy* system [2] introduces an enhanced network interface that can offload energy-intensive file downloads from the host. Since the enhanced network interface is Ethernet-based, the solution remains unsuitable for mobile environments. Also, Somniloquy achieves energy savings by reducing the probability that the computer may have to wake up in response to network traffic, without denying that possibility entirely. The system embeds application stubs in the enhanced network interface so that the interface can handle simple application messages that would otherwise wake up the host, but still defers more complex messages to the host. Conversely, the FileAccel and WebAccel primitives allow the host to sleep for arbitrary amounts of time, without requiring special application stubs and without ever waking up the host. The *power proxy table* (PPT) scheme in [5] elects a rule-based proxy in the network to handle the connections of a sleeping host. Like Somniloquy, PPT needs application-specific stubs in the proxy to handle network traffic. Cell2Notify [1] calls a smartphone through the cellular network to wake up its WiFi interface. Our solution's reliance on SMS messaging relieves the gateway from including an onerous PSTN interface. While SMS messages are generally slower than direct voice calls in reaching the handhelds, the latency increase is not large enough to cause negative impact on the overall file-transfer application. Tetherless communications [8] is a new paradigm for data transfer between weakly connected systems based on intermediate persistent storage. The paradigm relies on the cooperation of nomadic peers travelling across a delay-tolerant network. Our approach does not require a new communications infrastructure.

5. CONCLUSIONS

We have presented the FileAccel and WebAccel data transfer primitives for energy-efficient data dissemination to laptop groups using WWAN handhelds as intermediary nodes. FileAccel presents a network folder as a local laptop folder, whereas WebAccel serves web content from a nearby web proxy. As far as we know, these are the first energy-efficient data transfer schemes that are application-agnostic and WWAN-friendly, and that do not require the

laptop to wake up for a data transfer. An added advantage is that the primitives do not require the activation of *tethering* (typically offered at a premium) in the service subscription of the handheld, thereby cutting down on service deployment costs. The inclusion of a network gateway in the solution centralizes and therefore simplifies its management. While the primitives can be used for a variety of applications, they are chiefly well-suited for pushing patches and updates to the laptops. Our implementation of the primitives is an adaptation of a larger project [9] that uses a battery-powered smart data card to deliver a variety of enterprise services to the laptop, including remote kill, geo-location, asset tracking, and online backups [6].

6. ACKNOWLEDGMENTS

We thankfully acknowledge the Alcatel-Lucent team that implemented the FileAccel and WebAccel primitives in the OmniAccess 3500 Nonstop Laptop Guardian product.

7. REFERENCES

- [1] Agarwal, Y., et al. 2007. Wireless Wakeups Revisited: Energy Management for VoIP over WiFi Smartphones. Proc. of the 5th Int. Conf. on Mobile Systems, Applications and Services (June 2007). MOBISYS '07.
- [2] Agarwal, Y., et al. 2009. Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Use. Proc. of the 6th USENIX Symposium on Networked Systems Design and Implementation (Apr. 2009). NDSI '09.
- [3] Android. <<http://www.android.com>>.
- [4] Gunaratne, C., Christensen, K., and Nordman, B. 2005. Managing Energy Consumption Costs in Desktop PCs and LAN switches with Proxying, Split TCP Connections, and Scaling of Link Speed. Int. Journal of Network Management (Oct. 2005).
- [5] Nedeveschi, S., et al. 2009. Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems. Proc. of NDSI '09 (Apr. 2009).
- [6] OmniAccess 3500 Nonstop Laptop Guardian. <<http://www.alcatel-lucent.com/omniaccess3500/>>.
- [7] Rsync. <<http://samba.anu.edu.au/rsync/>>.
- [8] Seth, A., et al. 2005. An Architecture for Tetherless Communications. Proc. of the Dagstuhl Seminar on Disruption Tolerant Networking (Apr. 2005).
- [9] Stiliadis, D., et al. 2007. Evros: A Service-Delivery Platform for Extending Security Coverage and IT Reach. Bell Labs Tech. J., 12, 3 (Sep. 2007), 101-119.
- [10] Webber, C., et al. 2006. After-Hours Power Status of Office Equipment in the USA. Energy 31, 14 (Nov. 2006), 2823-2838.
- [11] Zhang, X., et al. 2007. Study of a Bus-Based Disruption-Tolerant Network: Mobility Modeling and Impact on Routing. Proc. of the 13th Int. Conf. on Mobile Computing and Networking (Montreal, QC, Canada, Sep. 2007). MOBICOM '07.