Minimizing Energy for Wireless Web Access with Bounded Slowdown

Ronny Krashinsky and Hari Balakrishnan

Introduction: The capabilities of mobile computing devices are often limited by the size and lifetime of the batteries that power them. As a result, minimizing the energy usage of every component in a mobile system is an important design goal. Wireless network access is a fundamental enabling feature for many portable computers, but if not optimized for power consumption, the wireless network interface can quickly drain a device's batteries. This project seeks to minimize the energy consumed by the wireless network interface for a mobile device generating request/response traffic (e.g., while browsing the Web) over a reliable transport protocol such as TCP. We investigate several interactions between energy-saving mechanisms and network performance, and show that understanding these interactions enables better energy-saving protocols to be designed that have provable performance-energy trade-offs.

Many wireless network interfaces, especially wireless LAN cards, consume a significant amount of energy not only while sending and receiving data, but also when they are idle with their radios powered up and able to communicate. However, because wireless applications typically use the network in bursts, wireless interfaces are designed so they can be disabled when not in use to save energy. As an example, consider the popular IEEE 802.11 wireless LAN specification [2], which describes a power-saving mode (PSM) that periodically turns the network interface off to save energy, and on to communicate and maintain the convenient "always on" network abstraction. When 802.11 PSM is enabled, the access point (AP) buffers data destined for the device. Once every *BeaconPeriod*, typically 100 ms, the AP sends a beacon that indicates whether or not the mobile device has any data waiting for it. The mobile device wakes up to listen to beacons at a fixed frequency and polls the AP to receive any buffered data.

The 802.11 PSM is an example of a *static* power-saving algorithm, since it does not adapt the sleep and awake durations to the degree of network activity; we will refer to it as *PSM-static*. We find that while PSM-static does quite well in saving energy, it does so at significant performance cost. The round trip time (RTT) of a TCP connection can increase substantially with PSM-static, since the effect is to *round up* the RTT to the nearest 100 ms. As shown in Figure 1, this has an especially adverse impact on short TCP connections, whose performance is dominated by the connection RTT. Furthermore, with PSM-static, the power consumed while sleeping and listening for beacons dominates the total energy consumption if the network is accessed only sporadically. For Web workloads, the long (but randomly distributed) idle periods ("think time") end up being most important in terms of energy usage, and PSM-static does not handle this situation well.

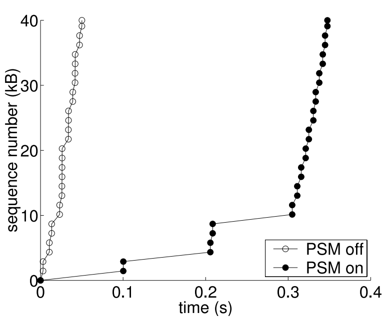

Figure 1: Measured evolution of a TCP connection with and without PSM enabled.

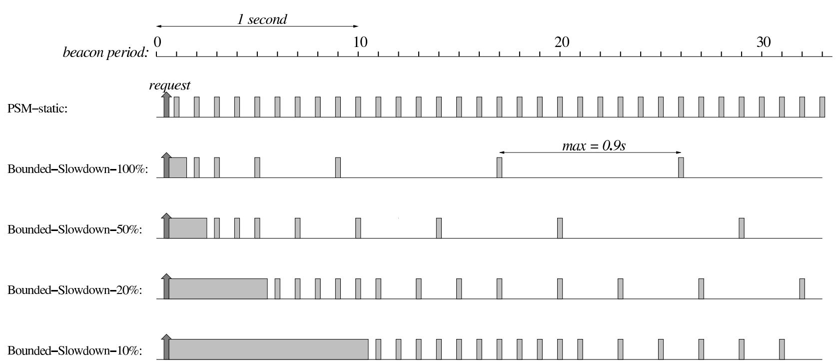

Figure 2: Schematic representation of PSM-static (the 802.11 PSM) and various Bounded-Slowdown alternatives. The arrow indicates a request sent by the mobile device, the initial shaded area indicates when BSD stays awake for a set time after the request, and the shaded bars indicate when the network interface wakes up to listen to beacons.

Approach: A PSM protocol addresses the following fundamental question: *When should a wireless interface go to sleep, and when should it be awake?* Based on our observations of the adverse and unexpected interactions that occur when a TCP connection is superimposed on PSM-static, we consider the problem of optimizing energy consumption

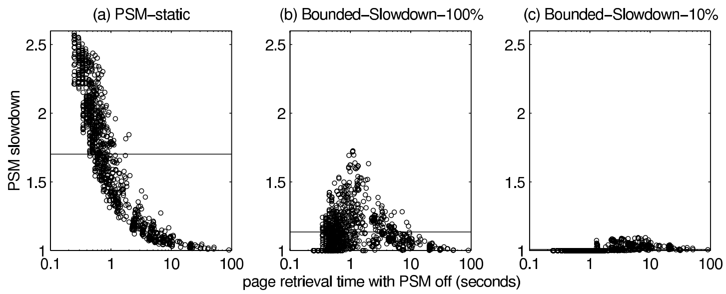

Figure 3: Per-page slowdown for three PSM alternatives. Each marker represents a single Web page; the x-coordinate is the retrieval time with PSM off, and the y-coordinate is the slowdown when PSM is on. The horizontal lines show the means.

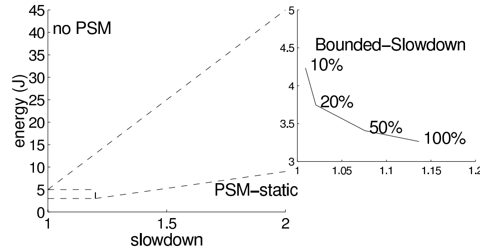

Figure 4: Mean per-page energy versus slowdown for various PSM alternatives.

under the constraint that interactive request/response performance does not degrade by more than a known amount. Specifically, we address the following problem: *Find an algorithm that minimizes energy consumption using the sleep and wake-up primitives such that any RTT does not exceed* $(1+p)\cdot R$*, where R is the original RTT and p is a tunable parameter that controls the maximum percentage slowdown.*

Our solution to this problem results in the *Bounded-Slowdown (BSD) protocol*. Figure 2 shows the behavior of PSM-static and the BSD protocol for various values of p (these are labeled as $100\cdot p$ percent). BSD stays awake for a short period of time after a request is sent, and adaptively listens to fewer beacons when the link remains idle. Staying awake reduces communication delay but increases energy consumption, while listening to fewer beacons saves energy but can increase delay.

Progress: To evaluate the effectiveness of the BSD protocol and compare it to PSM-static, we use simulations based on real-world Web traffic and power parameters from a commercially available 802.11b card.

Figure 3 compares the performance of PSM-static with two BSD alternatives based on a server RTT of 40 ms. The figure shows that PSM-static has the greatest negative impact on pages with fast retrieval times. These are slowed down by up to about 2.5× which is the penalty for extending a 40 ms RTT to 100 ms. BSD-100% shows a large improvement, and does bound the worst-case slowdown to be smaller than 2×. In fact, all of the slowdowns are far less than this bound because the protocol keeps the network interface awake for 100 ms after the mobile device sends data, so fast RTTs are not slowed down at all. BSD-10% further improves performance and shows almost no slowdown.

Figure 4 shows the trade-off between page retrieval time and energy consumption. PSM-static reduces energy by 11× compared to no PSM, but does this at the cost of increasing average Web page retrieval times by 70%. BSD-100% increases average Web page retrieval times by only 14% over the base performance with no PSM, and simultaneously uses 9% less energy than PSM-static (and 12.5× less than no PSM)—both its performance and energy usage are always better than the 802.11 PSM. BSD-20% essentially eliminates the slowdown in Web page retrieval times while only using 5% more energy than PSM-static.

Research Support: This work was funded in part by DARPA PAC/C award F30602-00-2-0562, and an equipment grant from Intel.

References:

[1] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," in *MobiCom 2002*, Atlanta, GA, September 2002.

[2] IEEE Computer Society LAN MAN Standards Committee, *IEEE Std 802.11: Wireless LAN Medium Access Control and Physical Layer Specifications*, August 1999.