Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices

Mark Stemm and Randy H. Katz
{stemm,randy}@CS.Berkeley.EDU
Computer Science Division, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720-1776.

Abstract

Next generation hand-held devices must provide seamless connectivity while obeying stringent power and size constrains. In this paper we examine this issue from the point of view of the Network Interface (NI). We measure the power usage of two PDAs, the Apple Newton Messagepad and Sony Magic Link, and four NIs, the Metricom Ricochet Wireless Modem, the AT&T Wavelan operating at 915 MHz and 2.4 GHz, and the IBM Infrared Wireless LAN Adapter. These measurements clearly indicate that the power drained by the network interface constitutes a large fraction of the total power used by the PDA. We then examine two classes of optimizations that can be used to reduce network interface energy consumption on these devices: transport-level strategies and application-level strategies. Simulation experiments of transport-level strategies show that the dominant cost comes not from the number of packets sent or received by a particular transport protocol but the amount of time that the NI is in an active but idle state. Simulation experiments of application-level strategies that significant energy savings can be made with a minimum of user-visible latency.

1 Introduction

Hand-held devices coupled with wireless network interfaces are emerging as a new way to achieve seamless connectivity. However, these new devices have power, cost, weight and size constraints that are more stringent than most laptop computers. The goal of achieving seamless connectivity while staying within limited size and energy constraints is challenged by the addition of a large power consumer to a personal digital assistant (PDA): a wireless network interface (NI). Current wireless network interfaces consume as much power as an idle PDA. For example, the network interfaces we measured consumed from 350mW to 1300mW when idle, and the PDAs we measured consumed from 700mW to 1200mW when idle. Although much work has been done in reducing the power consumption of other peripheral devices such as disks in laptop computers, [3] [2] [1] [7] [Li94] [8] [4], little work has been done to reduce NI power consumption in handheld devices.

This paper presents detailed measurements of the power and energy consumption of several wireless network interfaces to determine the power/energy drain of devices in their sleep, idle, packet-send and packet-receive states. We then examine two classes of optimizations that can be used to minimize energy consumption of wireless network interfaces: transport level optimizations and application level optimizations. For transport level optimizations, we examine different choices of transport layer protocols, using simulations to examine the relative power trade-offs when sending equal amounts of data from a wired sender to a mobile receiver. We find that the dominant cost in the energy usage of a transport protocol is the time that the transfer takes to complete, not the number of packets sent or received by a particular transport protocol.

For application-level optimizations, we focus on two applications that we expect to be the ``killer apps'' for PDAs: electronic mail and web access. We use real-world traces combined with simulations to experiment with application-specific energy savings strategies. Results show that significant energy savings can be made with a minimum of user-perceivable latency. In particular, for electronic mail applications, the energy consumption can be reduced to the minimum amount: the energy required to retrieve a piece of electronic mail. For web-browsing applications, energy consumption can be reduced by a factor of four with virtually no impact on user-visible latency.

The rest of this paper is organized as follows: Section 2 presents our power measurements of the NIs and PDAs. Section 3 presents our transport-level optimizations designed to reduce NI energy consumption. Section 4 presents our application-specific policies for email applications. Section 5 presents our policies for reducing energy consumption while web browsing, and Section 6 presents conclusions and recommendations for future protocols and network interfaces.

2 Measurements

In this section, we describe the methodology used to measure power consumption of devices and our results for several wireless NIs and PDAs.

We measured power consumption of two PDAs: the Apple Newton Messagepad 100, and the Sony Magic Link (PIC 1000), and four network interfaces: AT&T's Wavelan PCMCIA card operating at 915 MHz and 2.4 GHz, Metricom's Ricochet Wireless Modem, and IBM's Infrared Wireless LAN card. The AT&T Wavelan is a direct sequence spread spectrum PCMCIA interface with a range of approximately 40m and a user-visible bandwidth of 1.6 Mbits. The Ricochet Wireless Modem is a frequency hopping spread spectrum modem device with a range of approximately 1km and a user-visible bandwidth of 50 kbits. the IBM Infrared card is a diffuse Infrared PCMCIA interface with a range of approximately 5m and a user-visible bandwidth of approximately 850 kbits/sec. The Apple Newton and Sony Magic link are commercially available PDAs. To measure power consumption of the PDAs, we measured the devices while performing tasks designed to stress one subsystem of the device, for example pen input, speaker output, etc. and then averaged the measurements to obtain ``typical'' behavior[1]. We measured the network interfaces while "idling" (powered on but not sending or receiving packets), sleeping (powered off but still connected to the device), and sending and receiving packets of various sizes. We also measured the "wakeup" time, defined as the amount of time from when the device was brought out of its sleep state until the time that the first packet can be sent.

2.1 Methodology

To measure the power consumption for steady state behaviors, we required both current and voltage measurements. We used a digital oscilloscope to measure the voltage and current draw of the various devices. The current draw was actually measured by using a small resistor and measuring the voltage drop across the resistor. For the PDAs and Ricochet modem, which have their own external batteries, we measured the voltage and current at the battery terminals. (Although the Ricochet Modem currently has its own battery, Metricom has plans to make a Ricochet Modem using a PCMCIA form factor). For the PCMCIA NIs, we measured at the power pins coming into the card. For instantaneous operations such as packet transmission and reception, we made several measurements and averaged these together to obtain an average value. The digital oscilloscope produced bitmaps of the instantaneous voltage across the resistor over time. We post-processed the bitmaps to obtain the area under the curve (energy). We also verified that the voltage drops while taking measurements were not large enough to bias our results.

2.2 Measurement Results

Table 1 shows the average power consumption of the two PDAs and the Network Interfaces. An entry of "-" means that the device was not measured in that state, and an entry of "N/A" means that the state is not applicable to the device. The Metricom modem has a unique "wakeup" state; when the Metricom modem turns on, it registers with the network and consumes more power for approximately the first minute of activity. One important observation is that for all possible combinations of network interfaces and PDAs, the power consumed by the NI is comparable to (or even more than) the power consumed by the PDA. This is a clear indication that power management of the NI is essential. Also notice that the Metricom device has a much longer wakeup time than the other NIs. This will affect the usefulness of some of the optimizations described in Section 5.

Device	Sleep Power(mW)	Idle/ (Wakeup) Power (mW)	Wakeup Time (ms)
Wavelan (915 Mhz)	177.3	1318.9	100
Wavelan (2.4 Ghz)	143.0	1148.6	100
Metricom	93.5	346.9/431.03	5000
IBM IR	-	349.6	100
Newton PDA	164.2	1187.8	N/A
Magic Link PDA	312.03	700	N/A
Typical Laptop	-	8000	

TABLE 1. Power Consumption for network interfaces and devices

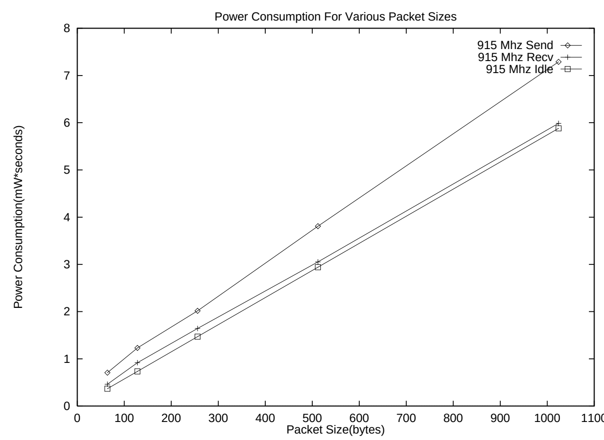

Figure 1. Energy consumption for different packet sizes for 915 MHz Wavelan

Once we had the measurements of the NIs while in their idle and sleep states, we performed more detailed measurements of the network interfaces to determine if the energy consumption differed significantly as the interface sent and received packets of various sizes. Figure 1 and Figure 2 show the results of these measurements for the 915 Mhz Wavelan and Metricom devices, respectively. The **x**-axis shows the packet size in bytes, and the **y**-axis shows the energy consumption in milliwatt-seconds. There are two lines that correspond to sending and receiving packets. Also included is a "baseline" measurement that indicates how much energy is consumed by keeping the interface on and idle for the same amount of time that it takes to send a packet. One obvious feature in the graph is that receiving packets only costs marginally more energy than being idle. This is also true for sending packets on the 915 Mhz Wavelan. For the Metricom device, the cost of sending is approxi-

1. More detailed measurements of the devices can be found at http://www.cs.berkeley.edu/~stemm/power.html.

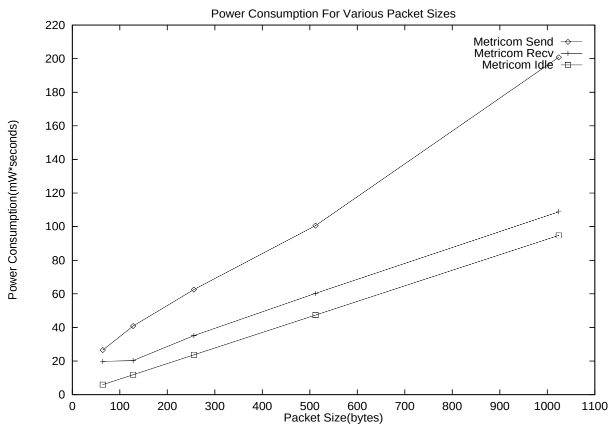

Figure 2. Energy consumption for different packet sizes for Metricom

mately double that of idling for the same amount of time. This would imply that sending is much more expensive than being idle, and network protocols should minimize the number of packets sent. As we will see in Section 3, however, other transport-level considerations have a more significant impact on the energy cost. In addition, for the applications of Section 4 and Section 5, the amount of time that the NI spends sending short acknowledgments is outweighed by the time spent receiving data packets. We believe that these applications or similar ones where the PDA is retrieving rather than sending large amounts of data will be the most common applications on future PDAs.

From these measurements we may conclude that:

1. Receiving packets only costs slightly more than the idle cost.

2. Sending packet costs more than receiving and can be significant when compared to the cost of being idle, but only if the mobile is sending large amounts of data to the wired network.

3 Transport Layer Simulation

In this section, we examine different transport-level protocols and find the energy costs to send data to a mobile receiver for each transport protocol. We start with a simple breakdown of transport-level energy consumption

3.1 Breakdown of Transport-layer Energy Consumption

We can break down the energy consumed to complete a bulk transfer of *b* bytes as follows for a fixed data packet size and a fixed acknowledgment size:

$$Idle = I\frac{b}{B}$$

$$Energy = SendRecv + Idle$$

$$SendRecv = aE_a + dE_d$$

Where a is the number of acknowledgments sent, E_a is the energy cost to send a single acknowledgment, d is the number of data packets sent, E_d is the energy cost to send a single data packet, I is the instantaneous idle power, and B is the effective bandwidth of the transfer. Our goal is to see how these two components of the energy cost change as the choice of transport protocol changes. In all of these simulations, we assume a transfer from a fixed source to a mobile receiver.

We compared four different transport layer protocols in terms of the number of acknowledgment packets they generate, the number of packets that they send to the mobile device, and the amount of time necessary to accomplish the transfer. These were:

1. TCP Reno: Using this protocol, the receiver generates an acknowledgment for every data packet sent.

2. TCP Reno with delayed acknowledgments: Using this protocol, the receiver generates an acknowledgment for every other data packet.

3. Reliable UDP, fixed-size window: Instead of depending on acknowledgments for flow control, this protocol uses rate control in combination with a fixed size error recovery window of size *w*. We used a window size of 10. Each window is acknowledged by the sender with a single selective acknowledgment, and any missing packets in the window are retransmitted by the sender. The receiver sends on average a little more than one acknowledgment for each **w** packets.

4. Reliable UDP, unlimited window: This is a special case of the above UDP scheme when the flow control window is equal to the number of packets sent.

The primary difference between these schemes is in the number of acknowledgments sent by the mobile device and the number of times that a duplicate packet will be received by the sender.

3.2 Methodology

The scenario we used was a three node network including a source, base station, and receiver. The source and base station were connected with a high bandwidth, low error rate link, and the base station and mobile were connected with a lower bandwidth, higher error rate link. We simulated the TCP protocols using the Network Simulator ns [9]. We simulated the Reliable UDP schemes by deriving formulas that showed the number of packets sent and received for a given bulk transfer size and packet error rate. To compare the protocols, we kept track of the total length of the transfer and the values of a and d. We then used the information extracted from the data in Figure 1 and Figure 2 to generate the energy drain for each packet sent and received as well as the energy cost for the entire transfer.

3.3 Simulation Results

Figure 3 shows the contribution that *SendRecv* makes to the energy cost for a variety of transfer sizes for the 915 Mhz Wavelan. The **x** axis shows the transfer size, and the **y** axis shows the energy cost in mW-seconds. These results show that the UDP protocols, which send fewer acknowledgments, use less energy. When the contribution from *Idle* to the total energy cost is included, however, (Figure 4), the total trans-

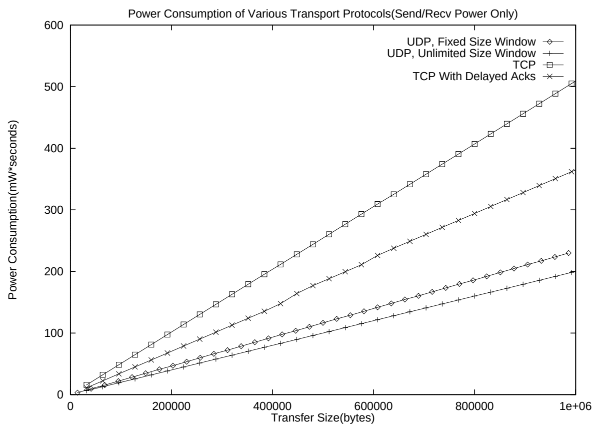

Figure 3. Energy for different transport protocols only including *SendRecv*

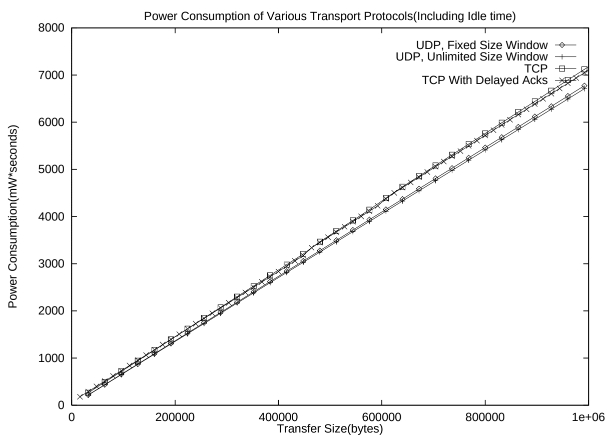

Figure 4. Energy for different transport protocols including *SendRecv* and *Idle*

fer time comes into play. Because the amount of time that the receiver waits for packets from the sender to arrive is much larger than the (relatively) small amount of time that the receiver actually sends or receives packets, the idle cost dominates the cost to send or receive packets, and the difference between the transport protocols is eliminated.

3.4 The Effect of Error Rate on Energy

Section 3.3 shows that *Idle* makes the greatest contribution to final energy cost, and that for low error rates, the different transport protocols behave similarly. Figure 5 shows the effect of a higher error rate on energy consumption. In the presence of a high packet error rate, the difference is more significant. As shown in [BSAK95], TCP mistakes packet losses for congestion and reduces the transmission rate. From a power standpoint, this decreases the value of B and increases the total energy cost. A more intelligent scheme that does not mistake wireless packet losses for congestion would not have this problem.

In the following sections, we use the results from the transport level simulation to experiment with application-specific policies for reducing energy consumption of network interfaces.

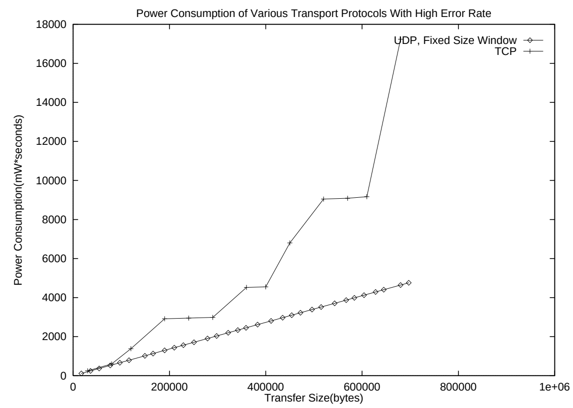

Figure 5. The effect of wireless losses on energy consumption

From our transport-level measurements, we have learned that:

- The dominant energy cost of any transport protocol is not the number of packets sent or received but the amount of time that the transfer takes to complete.
- This property means that the energy cost can increase significantly in the presence of wireless losses, where a receiver must wait for a TCP sender to recover from packet losses.

The results from our transport-level measurements are used in the application-specific experiments of Section 4 and Section .

4 Mail Simulation

In this section, we describe application-specific optimizations that can be used to reduce the energy compositions of network interfaces while using electronic mail applications. We start with a brief description of the trace data used for the experiments.

4.1 Data Collection

We used the user population of the Computer Science Division at UC Berkeley to measure mail activity. The arrival times and sizes of mail messages appearing in the Division mail spool was collected. This trace was used as a sample workload to the simulations of Section 4.2.

4.2 Simulation Setup and Results

In our strategy for reducing energy consumption, the PDA wakes up periodically, bringing its NI from a sleep to idle state and checks for new mail. Like approaches in [5], [6], and [11], the availability of new mail is broadcast periodically so the PDA does not have actually transmit any packets to check for mail.

We define the *attention span* as the amount of time that the PDA waits before waking up and checking for new mail. We ran the simulation for attention spans ranging from 60 seconds (1 minute), to 600 seconds (10 minutes), in 15 second increments and measured the average energy consumption

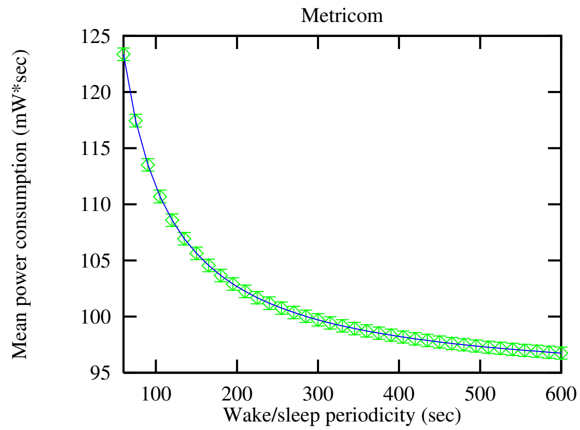

Figure 6. Energy vs. attention span for Metricom

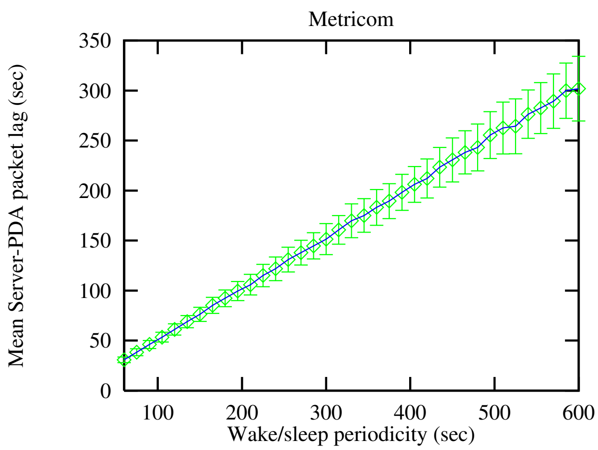

Figure 7. Staleness vs. Attention Span for Metricom

for the user population and the average "staleness", the lag between the time the mail message enters the mail spool and the time that the PDA discovers that the message has arrived.

Figure 6 shows the average energy consumption as a function of the attention span.As the attention span increases, the energy consumption decreases. Figure 7 shows the corresponding "staleness", which increases linearly as a function of the attention span. The results are quite promising; with an approximate staleness of two minutes, the power consumption drops by 20%. This attention span reduces the energy consumption to the cost of retrieving the email messages.

5 Web Access Simulation

In this section, we describe optimizations that can be used to reduce energy consumption for Web browsing applications. We briefly describe the data trace collection below.

5.1 Trace Collection and Processing

We used traces of HTTP traffic at UC Berkeley as input to a simulator which experimented with different power savings strategies. For each workstation, we kept track of the start times and transfer sizes for each outstanding HTTP connection. For each user, we divided time into *work* (when at least 1 outstanding connection was outstanding) and *think* (when no connections are active) phases. These post-processed traces form the input to the simulation.

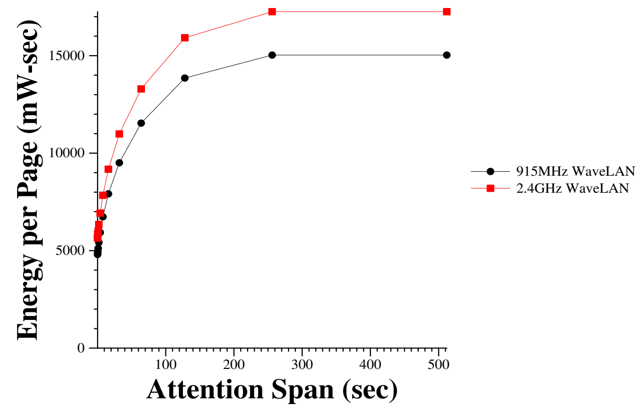

Figure 8. Energy per page as a function of Attention Span for Wavelan NIs

5.2 Power Saving Strategy

The power saving strategy evaluated in this section attempts to reduce effective power consumption during the think time portions of the traces. We turn off the network interface after the user has been in a think phase for more than a certain amount of time (called the *attention span)*. It stays in that state until the user sends data (in this case, a HTTP request) from the interface. It is important, however, to distinguish between large think times and times where the user has stopped using the application. For our measurements, we specified a maximum attention span of 5 minutes, after which we considered the PDA to have been turned off. Any think times more than 5 minutes were excluded in the simulation. In this way, we do not falsely claim energy savings when the user would have simply turned of the device immediately.

5.3 Simulation Setup and Results

The web simulation uses the NI and transport-level measurements from Section 2 and Section 3 as well as the traces described above.

The outputs of the simulation are two metrics of performance:

- Average energy cost, in mW-seconds, of an HTTP page retrieval.
- The average latency for the initiation of a Web page access. This measures the average amount of time to complete the first HTTP request of a work phase (with the assumption that Web page accesses are a single html document followed by a number of inline images)

Figure 8, Figure 9, and Figure 10 show the simulation results for the Wavelan and Metricom devices. Figure 8 shows the energy per page as a function of the attention span for the Wavelan NIs, and Figure 9 shows the response time as a function of attention span. For the Wavelan NI, we can see

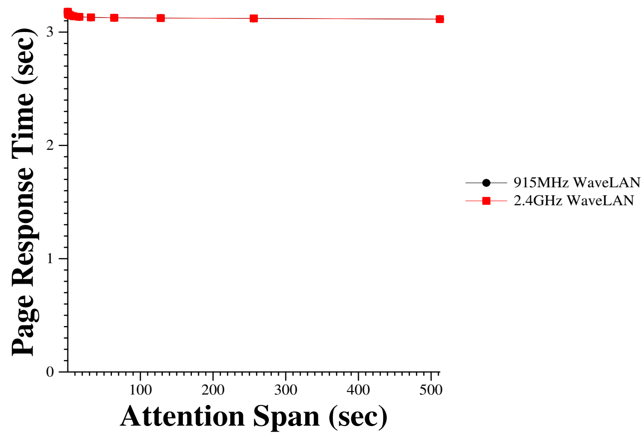

Figure 9. Response time as a function of attention span for 915 Mhz and 2.4 Ghz Wavelan

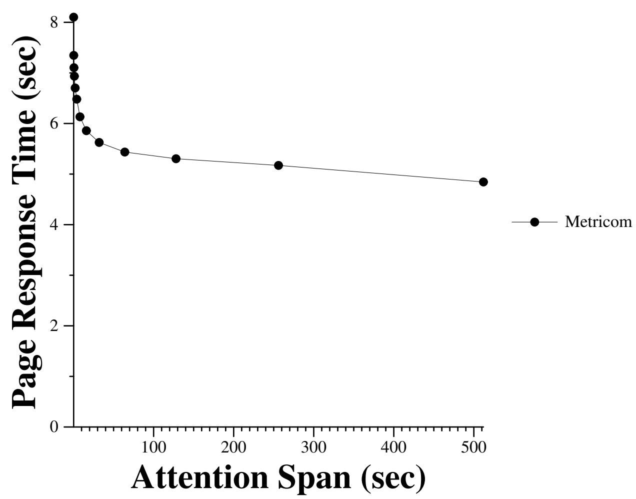

Figure 10. Response time as a function of attention span for Metricom

that increasing attention spans lead to significant reductions in energy costs with no user-visible increase in latency. Figure 10 shows the response time as a function of attention span for the Metricom NI. This illustrates the effect of a large sleep->wakeup transition. For shorter attention spans, the 5 second delay as the interface is powered on has a user-visible latency. For larger attention spans, however, the latency to retrieve the web page dominates.

6 Conclusions/Recommendations

Our measurements of PDA and Network Interface power and energy consumptions show that Network Interfaces consume a significant fraction of the total power on a PDA. Additional measurements for sending and receiving packets of various sizes indicate that the power consumed when the interface is on and idle is virtually identical to the cost of receiving packets. For some interfaces, the cost of sending packets can be significant when compared to the cost of being idle, but application- and transport-level considerations make the idle cost the dominant cost.

Although the choice of transport layer can have a significant impact on the number of packets sent and received by the mobile device, the actual power difference is minimal. This is because the energy consumed simply by keeping the network interface on during the transfer contributes the most to the final energy cost. In the presence of a high packet error rate, however, current TCP sender implementations overreact to packet losses, mistaking them for congestion. This slows down the transfer rate, which increases the amount of time that the transfer takes and the amount of energy consumption by the network interface.

Simulations show that for email, our optimizations can reduce the energy consumption to the minimum possible: the energy required to receive messages. For web browsing, fast sleep-idle transitions allow significant power savings with no impact on user-visible latency. Even for interfaces with longer sleep-idle transitions, however, significant power savings can be achieved with less aggressive management of the network interface.

6.1 Recommendations for Future Networks Interfaces and Protocols

Current generation transport and link-level protocols may need some tuning to minimize the power cost of network interfaces. Any protocol that leaves a mobile receiver idle unnecessarily (such as TCP's backoff in the presence of wireless losses) wastes power. Even when the protocol is performing correctly, inefficient link-layer scheduling may be the problem; a link layer that allocates 2 Mb on a contention basis for 10 mobiles causes each of them to consume 10 times as much power (100 times as much power total!) as a base station that uses a TDMA scheme to coordinate delivery of data to receivers. Recent work has proposed more intelligent link-layer schemes to handle this problem [10]. The valuable lesson is that network interfaces can consume a significant fraction of the power budget of PDAs, and this requires smart software and applications to make sure that battery lifetime is not needlessly shortened.

7 Acknowledgments

Thanks to Paul Gauthier and Daishi Harada, who worked on earlier versions of this work. Thanks also go to Bruce Mah from UC Berkeley for providing us with the WWW traces. This work is supported by DARPA contract DAAB07-95-C-D154 and grants from the California MICRO Program, Hughes Aircraft Corporation, Metricom, and AT&T.

References

[1] Fred Douglis, Frans Kaashoek, Brian Marsh, Ramon Caceres, Kai Li, and Joshua Tauber. Storage alternatives for mobile computers. In *Proc. 1994 Symposium on Operating Systems Design and Implementation*. OSDI, November 1994.

[2] Fred Douglis, P. Krishnan, and Brian Marsh. Thwarting the power-hungry disk. In *Proceedings 1994 Winter USENIX Symposium*, 1994.

[3] Fred Douglis and Brian Marsh. Low power disk management for mobile computers. Technical Report MITL-TR-53-93, Matsushita Information Technology Laboratory, April 1993.

[4] D.P. Hembold, D. E. Long, and B Sherrod. A Dynamic Disk Spin-down Technique for Mobile Computing. In *Proc. MOBICOM '96*, November 1996.

[5] T. Imielinski, S. Viswanathan, , and B.R. Badrinath. Energy efficient indexing on air. In *Proceedings ACM SIGMOD*, 1994.

[6] Ravi Jain and John Werth. Airdisks and airraid:modeling and scheduling periodic wireless data broadcast. *Computer Architecture News*, 23(4):23–28, September 1995.

[7] Kester Li, Roger Kumpf, Paul Horton, , and Tom Anderson. A quantative analysis of disk drive power management in portable computers. In *Proceedings 1994 Winter USENIX*, pages 279–291, San Franscisco, California, Winter 1994.

[8] Brian Marsh. Systems issues for mobile computing. Technical Report MITL-TR-50-93, Matsushita Information Technology Laboratory, Feburary 1993.

[9] ns – LBNL Network Simulator. http://www-nrg.ee.lbl.gov/ns/, 1996.

[10] W. M. Smith and P.S. Ghang. A Low Power Median Access Control Protocol for Portable Multi-Media Systems. In *Proc. Third Workshop on Mobile Multimedia Communications (MoMuC-3)*, Sept 1996.

[11] Stanley Zdonik, Michael Franklin, Rafael Alonso, and Swarup Acharya. Are "Disks in the Air" just pie in the sky? In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 12–19, Santa Cruz, California, December 1994.

Mark Stemm (ACM S 1995, IEEE S 95) is a Ph.D. student in Computer Science at the University of California at Berkeley. His research interests are in the areas of wireless and wide-area computer networks, mobile computing, and operating systems. He received his B. S. degree in Computer Science with University and Departmental honors from Carnegie Mellon University in 1994 and his M.S. degree in Computer Science from the University of California at Berkeley in 1996.

On the WWW, his URL is: http://www.cs.berkeley.edu/~stemm. His e-mail address is: stemm@cs.berkeley.edu.

Professor Randy H. Katz is a leading researcher in computer system design and implementation. His research experience has spanned numerous disciplines. He has written over 120 technical publications on CAD, database management, multiprocessor architectures, high performance storage systems, and video server architectures. He was responsible for developing the concept of Redundant Arrays of Inexpensive Disks (RAID), now a $3 billion industry segment. Katz's recent research has focused on wireless communications and mobile computing applications.

From January 1993 through December 1994, Katz was a program manager and deputy director of the Computing Systems Technology Office of ARPA. He was responsible for "wiring the White House" to the Internet, and also assisted the Clinton Administration in formulating policies related to the National Information Infrastructure and wireless technologies.

He is a member of the ACM and a senior member of the IEEE.