Power-Aware Communication for Mobile Computers

Robin Kravets, Karsten Schwan
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
`robink,schwan@cc.gatech.edu`

Ken Calvert
Department of Computer Science
University of Kentucky
Lexington, KY, USA
`calvert@dcs.uky.edu`

Abstract

Recently, the mobile community has focused on techniques for reducing energy consumption for mobile hosts. These power management techniques typically target communication devices such as wireless network interfaces, aiming to reduce usage, and thus energy consumption, of the particular device itself. We observe that optimization of a single device's energy consumption, without considering the effect of the strategy on the rest of the machine, can have negative consequences. We propose power management techniques addressing mobile host communications that encompass all components of a mobile host in an effort to optimize total energy consumption. Specifically, we propose runtime adaptation of communication parameters in order to minimize the energy consumed during active data transfer. Information about the network environment is used to drive such adaptations in an effort to compensate for the effect of dynamic service from wireless communication device on the energy consumed during data transfer. Our results show that power management can often be achieved while maintaining the QoS provided to the application.

1 Introduction

The demand for low-cost, power-efficient computing platforms dominates the mobile computing community, with machines ranging from high-end laptops to small hand-held devices. Yet, these power-efficient mobile computers must support an ever-increasing range of communication services, including internet access, multimedia conferencing and collaborative work. In part, this trend is driven by the fact that wireless communication devices are becoming increasingly common, sometimes even replacing traditional Ethernet cards in mobile computers. Unfortunately, as communication demands increase, so do the demands on the machine's batteries.

The overall goal of power management for mobile computers is to prolong battery life by controlling the energy consumption of the mobile host's various devices. Past research has developed techniques to reduce the energy consumption of individual devices, including disks [DKM94, HLS96, LKHA94], CPUs [GCW95, LS96, WWDS94], and network interfaces [SK96, CPR98, NSYA97, KK98]. All of these techniques aim to reduce the energy consumption of individual devices on the mobile host. This paper builds on such work. It differs from it, however, by not only attempting to minimize the energy consumption of each device, but instead, by doing so in the context of the total energy consumed by the mobile host.

The specific device investigated in our work is the host's communication device. While we also wish to reduce this device's energy usage, our actual goal is to minimize the total energy consumption of a mobile host in the context of its communication. We approach this problem by defining a *communication action* as the entire set of steps or computations involved in performing some communication via this device. We define the energy consumption of a communication action to be the energy consumed by the mobile host plus the energy consumed by the devices being used during the lifetime of the communication action. We then consider the effects of certain device-level power management techniques on the energy consumed by entire communication actions. This paper demonstrates the importance of action-based vs. device-level power management. For instance, by reducing the energy consumption of the communication device, total transfer time may be increased, with the resulting effect of increasing the total energy consumption of the communication action. Clearly, this is not a desirable outcome.

This paper presents techniques for power-aware communication for mobile computers. Central to our work is the aforementioned notion of communication actions, for which we capture both the energy consumed for data transmission by the network interface device (e.g., wireless Ethernet, packet radio modem) and the energy consumed by the host when it generates or consumes this data. Based on this information, we determine the energy cost of transmitting data. Next, we control communication actions' energy consumption by adapting certain parameters of the transmission protocols being used, in response to information about the current state of the network (e.g., loss rate, loss burst size).

Consider the action of transmitting a data object using a stop-and-wait transmission protocol. This protocol attempts to minimize the number of messages transmitted, thereby optimizing the energy consumption of the communication device for the entire data transfer. Alternatively, when using a go-back-N transmission protocol, the communication action may incur additional overhead resulting from unnecessary retransmission of successfully received messages, but it also tends to reduce the action's overall transfer (i.e., completion) time. The integrated power management techniques presented in this paper are designed to consider and capitalize on such tradeoffs concerning energy consumption of communication for mobile hosts.

We demonstrate the effectiveness of power-aware communication techniques with a transmission protocol tailored for wireless communication. In this context, it is intuitive that excessive retransmission and acknowledgments will consume additional energy, because the transmission of these extra messages increases the energy consumption of the wireless network interface device. Recent research in this area has fine tuned transport and MAC layer protocols to minimize this overhead [CPR98, SCAK98, ZR97]. However, the resulting reduction in the number of unnecessary messages is achieved at the cost of an increase in the total transfer time of the data being sent. In other words, for the communication actions implemented by these protocols, the energy consumed by the network interface device is reduced, whereas the energy consumed by the mobile host's CPU and the other devices involved in these actions is increased.

Consider the design of a transmission protocol. The basic mechanisms used to build this protocol concern/affect window size, acknowledgments (ACKs), negative acknowledgments (NACKs), selective acknowledgments (SACKs), FEC and timers. Window size is adjusted to compensate for congestion and to manage flow control. Acknowledgments are used to indicate the state of the receiver. FEC is used to compensate for expected losses, and timers are used to determine certain types of losses. The manner in which these mechanisms are employed defines the behavior of the protocol and therefore determines the energy consumed by the transfer of the data. This paper explores the use of a SACK-based transmission protocol with two modes of transmission. The first mode is a standard SACK protocol [MMFR96, KM97, FF96], while the second mode uses multiple retransmissions in the face of lost messages. The principal experimental results attained with this protocol show that under specific (lossy) wireless network conditions, aggressive retransmission policies can reduce the total energy consumption of the mobile host. Specifically, as losses approach 25%, our techniques result in a 25% energy savings for a high-end laptop, and the predict a 20% savings for a mid-level laptop.

The final piece of this research proposes the runtime adaptation of communication parameters in order to minimize the energy consumed during active data transfer. Information about the network environment and the energy consumption of various transmission configurations is used to drive such adaptations. Our goal is to compensate for fluctuations in energy consumption due to the dynamic nature of the services available from wireless communication devices.

In Section 2, we describe our power management approach and techniques, and we describe how to quantify and optimize energy consumption. We then discuss power management in the context of transmission protocols in Section 3. Section 4 presents the results from our experiments with a configurable SACK-based protocol. In Section 5, our power management techniques are placed into the context of prior work on communication adaptation. Finally, Section 6 presents some conclusions and future work.

2 Mobility, Communication and Power Management

The desire to conserve energy during active communication has driven diverse communication-centric power management techniques. The goal of such power management is simply to conserve battery lifetime for the mobile host. By aiming our efforts at the whole machine, we can consider all of the factors we have discussed so far. Interestingly, for ongoing communications, it is less important to distinguish the contribution to total energy consumption from each device than to understand the tradeoffs in terms of total energy consumption for different protocol configurations for such communications. In order to be able to quantify such tradeoffs, we present an energy model based on the energy components of a communication action.

This section first discusses the target mobile environment for our research. Next, we discuss in detail how energy consumption may be described to enable online power management for communications. Three sets of measurements are necessary to differentiate base energy usage at idle times, with and without attached network interface devices, from energy requirements during ongoing communications.

2.1 The Mobile Environment

Consider a mobile host that communicates with a stationary base station. Power management techniques must deal with situations in which the mobile host is mostly sending data, receiving data, or some combination of both. Specifically, a sender may be concerned with the energy overhead of transmitting unnecessary retransmissions, while a receiver may be concerned with unnecessary acknowledgments. Both sender and receiver are affected by the total amount of time of the data transfer. We discuss both sender and receiver scenarios in the experiments in Section 4.

The use of wireless links poses a number of unique problems, including novel loss characteristics, synchronization of disconnected operations, and issues involving packet forwarding. These problems pose significant challenges for end-to-end communication protocols. Two types of models have been studied [BPSK96]. The first model exploits the natural hop between a base station and the mobile host for communications crossing the wired-wireless boundary. Namely, standard communication protocols are used by wired hosts to a base station and specialized protocols are used for the final hop from the base station to the mobile hosts [BB95]. The second model utilizes and tunes existing end-to-end protocols, providing help and hints along the way [BSAK95]. We focus on the first model of communication, which allows us to isolate and target the communication between the base station and the mobile host. We do not explicitly investigate the coordinated power management of mobile hosts that communicate with each other via ad hoc networking techniques.

2.2 Energy Consumption Model

Previous work has focused on the energy consumption of individual devices, such as network interface devices [CPR98, SCAK98, ZR97]. We consider an entire communication action, thereby addressing both the network interface and the rest of the mobile host involved in the communication. In effect, we are “charging” the communication action for all system usage that occurs during that action. This paper employs a simple, additive model for the energy consumption of a communication action which ignores concurrency between actions. This simplification is appropriate for mobile hosts not capable of concurrency and also where communication delays make the use of concurrency impractical.

Figure 1 shows a sample energy measurement for a mobile host while the machine is idle, with and without an attached network interface device, and while the machine is actively communicating. Three sources of energy consumption may be differentiated in Figure 1: communication-specific, device-specific and machine-specific. The area under the solid line in the first time period represents the energy consumption of the idle machine, including the CPU, disks, and other components besides the network interface device. This is the *machine-specific* base amount, indicated by power level A. This base amount will always be consumed. During idle times, this amount may be reduced with machine-specific power management techniques, but we assume that while communication is ongoing such techniques will not be invoked. In the next time period in the figure, a wireless network interface device has been activated. Therefore, this area represents the energy consumed by both the idle machine and the idle wireless network interface device. This *device-specific* energy consumption by the network interface device is indicated by the section between power levels A and B. As with the machine-specific power management techniques, device-specific techniques may be used to reduce energy consumption during idle periods in communication. Since we are targeting active communication, we can assume that the device must be turned on during those periods and will consume some average amount of device-specific energy. Finally, when the mobile host is actively communicating, as indicated by the values above power level B, the energy consumed is *communication-specific*. This energy includes the energy consumed by the wireless network interface device for data transmission as well as any energy consumed by the CPU (and other devices involved) during data and protocol processing for transmission and reception. Since the use of the hard disk during data transfer is highly dependent on the application being used, for the remainder of this paper, we assume that the hard disk is not involved in communication. This assumption allows us to focus on the CPU and network interface device. In would be interesting in the future to investigate the interaction between power management techniques for active communication and power management techniques for hard disks.

From Figure 1, it is apparent that transmission-time energy consumption can be partitioned into these three sources. The machine-specific and device-specific components are only affected by the total amount of time required for communication. On the other hand, the communication-specific component, which represents the amount of additional energy needed for actual data transmission,

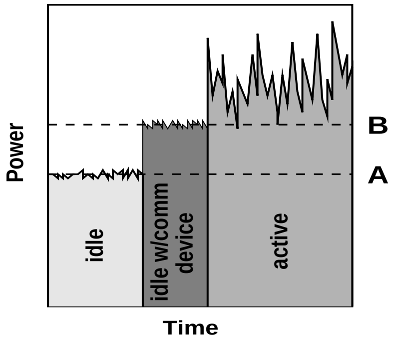


Figure 1: Energy Components of a Communication Action

is affected by two parameters. The first is the amount of energy consumed by the device for data transmission (or data reception for a receiver). This amount will generally increase linearly with data size. In a lossy network environment, this component will also increase due to retransmissions. The second parameter is the amount of CPU processing needed to run the protocols for data transmission or receipt. For the lossy network example, this amount increases due to the computation involved in handling timeouts and preparing messages for retransmission. One important thing to notice is that the area above B is *not* affected by the amount of time it takes to complete the communication action, i.e. transfer the data; if there is a pause in the data transmission, no energy consumption is attributed to the communication-specific component.

The energy measurement model presented above partitions energy consumption for data transfer into component parts such that each part can be considered independently with respect to its contribution to total energy consumption. Since both machine-specific and device-specific energy consumption only depend on transfer time, it is relatively easy to determine their contribution to total energy consumption. The determination of communication-specific energy consumption is a more difficult problem, since it depends on the configuration of the communication parameters. In Section 4, we will discuss one such example for a mobility-oriented transmission protocol.

3 Power-Aware Communication

The objective of our work is to reduce the energy usage of communication actions. As stated earlier, mobile hosts operate in dynamic network environments. The transmission-time power management techniques we aim to develop must consider the effects of changes in network services on the performance of underlying transmission protocols, and therefore, changes in energy consumption. Specifically, assuming that there is some fixed level of energy consumed by the simple transfer of data in an error-free environment, as errors are introduced into the data transmission stream, energy consumption increases due to three factors: retransmission and processing of lost messages, additional ACK/NAK transmission and processing, and increased transfer time.

The remainder of this section focuses on protocol behaviors in the presence of errors, specifically on the efforts necessary to compensate for lost messages. This is because, in a mobile environment, the possibility of a lost message is much higher than for typical wired environments, and so a protocol that optimizes for lost messages may be more efficient. Our future research may also consider the effects of transmission-time message reordering.

3.1 Transmission-Time Energy Consumption

We can attribute energy consumption to several protocol behaviors. The first is data transmission for a sender or data processing for a receiver. For a sender in a lossless environment, energy will be consumed for each message transmitted. As losses are introduced, more energy will be consumed by retransmitting lost messages. In the best case, the protocol will only retransmit a lost message as many times as needed to have the receiver successfully receive the message. If the protocol is too aggressive, there may be additional transmissions of already received messages. We can call this type of overhead *unnecessary retransmissions*. The second behavior is the amount of time the protocol stalls while waiting for the successful retransmission of a lost message or for some indication that the retransmission has been lost. We call this type of overhead *unnecessary wait time*. If the protocol is aggressive, it may be able to reduce the unnecessary wait time by increasing the number of unnecessary retransmissions. The third behavior is the processing of acknowledgments for a sender and the transmission of acknowledgments for a receiver. The timing of acknowledgments can be used to maintain the flow of data transmission. In the presence of too many acknowledgments, too much time will be spent sending and processing acknowledgments. We call this type of overhead *unnecessary acknowledgments*. In the presence of too few acknowledgments, time will be wasted at the sender while it is waiting for instructions from the receiver. We include this type of overhead in *unnecessary wait time*.

Consider again the breakdown of energy consumption during data transmission, as indicated in Figure 1. Device-specific and machine-specific energy consumption will only be affected by unnecessary wait time, since both are time dependent. Communication-specific energy consumption, on the other hand, is only affected by unnecessary retransmissions and unnecessary acknowledgments. Given this separation of contributing factors, we need to target multiple, and potentially conflicting, parameters.

4 Experimental Evaluation

The hypothesis of our research is that power management techniques aimed at a single device may in the end increase the amount of energy consumed by the mobile host for a communication action. The goal of our experiments is to present a specific example of a situation where minimizing communication-specific energy consumption has a detrimental effect on the total energy consumed by the mobile host. These experiments target applications with large data sets to transmit. These sets may be images, sensor data, or simulation data. We target such data to determine the effect of power management on large data transfers. As a result, our experiments are designed to have the sender actively sending for a long period of time. During this time, we measure the energy consumed by the data transfer. In this section, we describe our experimental setup and present the results from our experiments. In Section 5, we generalize these results to define techniques for adaptive power management.

4.1 Experimental Setup

In order to determine the impact of our power management techniques, we measure the energy consumption of a mobile host under varying conditions. In our experiments, we use a 915MHz Lucent WaveLAN PCMCIA wireless Ethernet card that can transmit data up to 150KBps. It provides three power modes: transmit, receive and suspend, and does not perform power management at the MAC layer. The system is configured as shown in Figure 2, with a wireless Ethernet in a NEC Versa 6360 laptop (the mobile host) communicating with a NEC Versa 6320 (the base station) using a second WaveLAN PCMCIA card, both machines running Linux. The laptop is plugged into a universal power supply (UPS) to filter out fluctuations in wall voltage. The multimeter samples the current 11–12 times a second. From these samples and the output voltage of the UPS, we can determine the energy being used by the computer.

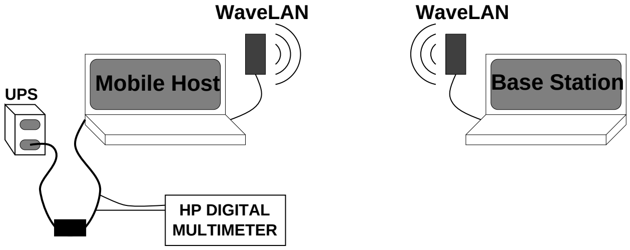


Figure 2: Experimental Setup

To determine the mobile host's energy consumption, we monitor the current being drawn from the transformer by the host. Figure 3 shows the output of the multimeter over time for data being transferred from the NEC Versa. This trace of current readings (11–12 readings a second), when integrated over time, provides us with the total energy and average energy consumed during that time period. In Figure 3, the solid line near 14W represents the power measured for this specific computer when it is idle. In this situation, we define idle to be not communicating, but also not invoking machine-specific power management techniques. From this baseline information about the necessary energy to run a computer, we can compute the "cost" of communication. This cost includes the energy consumed by the communication device and any energy consumed by the CPU due to the communication.

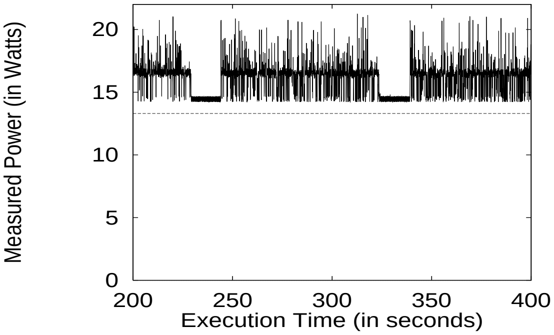


Figure 3: Sample Output from Multimeter

According to specifications from the manufacturer [Luc96], the power requirements of the WaveLAN card are those shown in Table 1, Column 2. Column 3 in Table 1 shows the power requirements measured during our experiments. The measurements for receive mode are taken while the computer was idle, which implies no extra disk or CPU activity. We observe that our measurements of the power required while the device is in either mode are very close to the documented specification.

State	Documented	Measured
WaveLAN - suspended	0W	0W
WaveLAN - receive	1.48W	1.52W
WaveLAN - transmit	3.00W	3.10W

Table 1: Power Requirements of the Lucent WaveLAN PCMCIA Wireless Ethernet card.

From our measurements, we measured the NEC Versa around 13.3W when idle. With the WaveLAN card inserted, the power increased 14.8W when the machine is idle. From this information, we consider any power over the 13.3W idle measurement to be contributed to the ongoing communication. Additionally, we note that the power often peaks over 16.3W, which is the expected power necessary for transmission. This additional power over 16.3W can be attributed to the use of the CPU during active data transmission. As mentioned earlier, we also consider this power to be attributed to the active communication.

4.2 Energy Consumption for TCP

The design of TCP and other such transmission protocols has included reliability as well as congestion and flow control. Since such protocols have been developed in the context of the Internet, much effort has been put into congestion detection and recovery as well as compensating for out-of-order message delivery. As a result, TCP has been optimized to react to lost messages as an indication of congestion. In such cases, TCP backs down and reduces its transmissions in an effort to ease the congestion. In contrast, losses in a mobile environment are often caused by interference, not congestion, and so should be aggressively retransmitted without backing down. Additionally, TCP makes an initial assumption that a missing message at the receiver may still be en route and will arrive out of order. Since, in a wireless environment, the probability of message loss is significantly higher than the probability of message reordering, protocols should be optimized for treating missing messages as lost.

Acknowledgments are used by the receiver to indicate to the sender which messages have or have not been successfully received. Acknowledgments can be either positive (ACK), to indicate a successfully received message, or negative (NAK), to indicate a potentially lost message. TCP, for example, uses ACKs to indicate the last successfully received message, and continues to send an ACK for the same message when lost messages have been noticed. In order to compensate for potentially out-of-order messages, TCP delays its ACKs for a period of time. This allows the potentially en route data to arrive without causing an unnecessary retransmission, but delays the retransmission of actual lost messages. Additionally, the sender will not retransmit a message until it has received three duplicate ACKs for the same message, adding additional delay to the retransmission for actual lost messages.

In order to determine the effect of message loss on energy consumption for TCP, we set up a simple experiment run between two mobile hosts using wireless Ethernet devices. We transferred 7MBytes of data using TCP and measured the energy consumed by the whole computer. We simulated lost and corrupted messages by having the receiver randomly drop message at the device driver level. As the number of lost messages increased to 15%, the communication-specific energy consumption increased 20%, but the transfer time (and so the device-specific and machine-specific energy consumption) increased over 500%.

Figure 4 presents the three components of energy consumption for an active TCP connection. The scale on the left hand side is the ratio of the increased energy consumption of the communication action due to loss to the energy consumption of the communication action with no loss. Figure 4(a) shows the communication energy consumption as message loss increases. As we can see, the amount of energy used for data transfer increases at least linearly with the loss rate. This can be expected since the sender needs to retransmit those lost messages. It is important to note here that TCP tends to minimize the number of unnecessary retransmissions. The middle graph in Figure 4(b) shows us the communication-specific plus the device-specific energy consumption. With this graph, we can already see the effect of having to maintain the network interface device in an active mode for an extended period of time: although the energy consumption for transmitting messages has been kept low, the energy consumption for the device has increased significantly. Now add in the energy used by the rest of the system (Figure 4(b), top graph); note the dramatic change in scale from Figure 4(a) to Figure 4(b). The top and middle graphs appear to increase exponentially with loss rate. The reason is that both the device-specific and machine-specific components are linear in time, while TCP's congestion-control mechanism, which doubles the retransmission timeout on successive losses, effectively causes the total transfer time to grow exponentially with increasing loss rate.

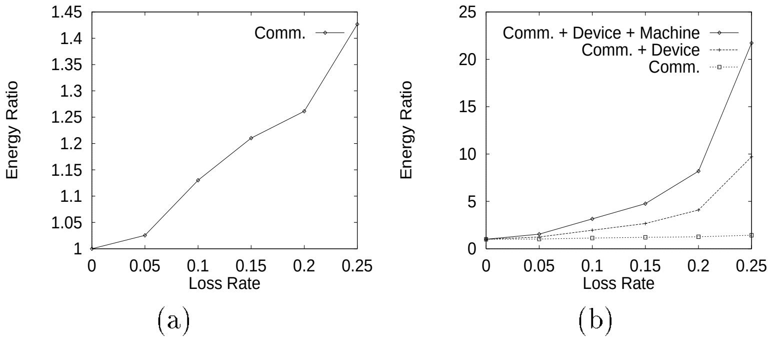


Figure 4: Energy Consumption for TCP

It can be argued that TCP is inefficient for wireless networks where losses of up to 30% can be experienced. Issues with the use of TCP in mobile environments are well-known and are not the subject of our work. The point of this example is simply to illustrate that we need to consider issues beyond identifying the overheads associated with transmitting messages. We next discuss a more wireless-friendly transmission protocol and show similar, though not as dramatic, results.

4.3 Energy Consumption for SACK

The previous results with TCP lead us to consider the use of selective acknowledgments (SACKs) to indicate all successfully received messages to the sender [MMFR96, KM97, FF96]. By using a SACK, the receiver can aggressively tell the sender which messages have been received and which messages can be assumed lost or reordered. The use of SACKs also leaves a large amount of flexibility in the design of the sender. The standard sender could be optimized for loss compensation, but, through the monitoring of the packet stream, could be adapted to a more out-of-order friendly protocol in such situations. For the rest of this discussion, we are going to focus on the efforts necessary to compensate for lost messages. In the future, it would be interesting to consider the effect of transmission-time message reordering on such a protocol and its energy consumption.

The goal of these experiments is to demonstrate the tradeoff between minimizing the number of unnecessary retransmissions vs. minimizing the unnecessary wait time and so evaluate the effect of this tradeoff on the total energy consumption of a mobile host. In this section, we consider a mobile host that primarily transmits data.

To demonstrate such an energy tradeoff, we implement a SACK-based transmission protocol. There are two operating modes for this protocol. In the normal case, the protocol sends a simple retransmission when it determines a message has been lost. The second option is to send two retransmissions of the same message back-to-back. This option can significantly reduce transfer time in environments where, if the first retransmission gets lost, the second retransmission still has a good chance of being transmitted successfully. Figure 5 shows the running times of such a protocol for each option. These times are the average of 5 runs of the experiment. Each experiment transferred a 7MByte block of data, with loss rates varying from 0 to 25%. As we can see, the use of double retransmissions can reduce the transfer time by 5% of the original transfer time for 10% loss and by 53% for 25% loss. Information about transfer time also allows us to determine both device-specific and machine-specific energy consumption.

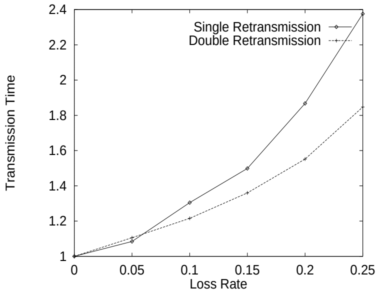


Figure 5: Running Time

To understand how retransmission policy and transfer time translate to energy consumption, we can examine the results similarly to our earlier examination of TCP. Figure 7(a) shows the communication-specific energy consumption for both options of the SACK protocol for a mobile host that is primarily sending data. The results for a receiver are similar and have been omitted for brevity. As we would expect, double retransmissions cost more energy than single retransmissions due to an increase in the number of unnecessary retransmissions. Figure 6 compares the number of unnecessary retransmissions used for each configuration for a 7MByte data transfer. Since transfer time does not play a part in communication-specific energy consumption, the time saved by the use of double retransmissions does not help reduce the communication-specific energy consumption. If we now include the device-specific energy consumption (Figure 7(b)), we can see that the energy consumption for single and double retransmissions has almost evened out. This can be explained by the fact that the device-specific energy consumption is solely dependent on transfer time. Finally, the inclusion of the machine-specific energy consumption shows us that the use of double retransmissions can actually save energy (Figure 7(c)). For both Figures 7(b) and 7(c), increased energy consumption for single retransmission is due to increased unnecessary wait time.

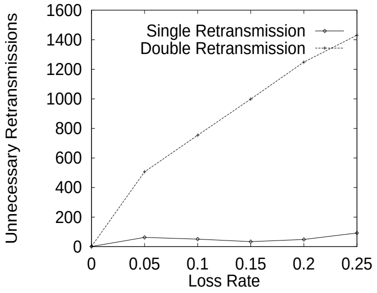
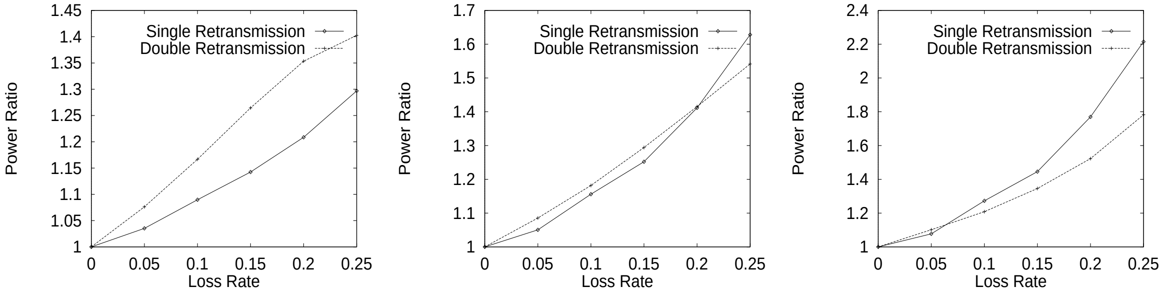


Figure 6: Unnecessary Retransmissions



(a) (b) (c)

Figure 7: Energy Components

The experimental results presented in this section utilize a high-end laptop with high run time energy consumption for the machine itself. From these results, we can also predict the energy consumption of more energy efficient machines. For example, the previous experiment was performed on the NEC Versa, which consumes around 13.3W when idle. If we consider a machine like the Toshiba Libretto 60 which consumes about 7W when idle, we would expect to see results like those in Figure 8. This demonstrates that our energy consumption model enables power management techniques across machines with different energy consumptions.

A limitation of our initial experimental results is the assumption that losses are random. In

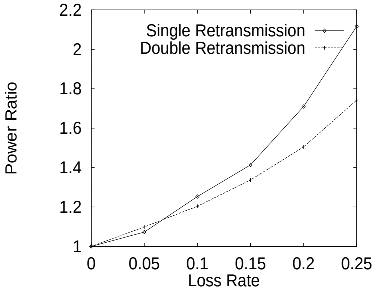


Figure 8: Total Energy Consumption

ongoing research, we use a more realistic loss model, based on the assumption that successive packet losses are correlated. We implemented a packet loss model based on the model presented by Zorzi [ZRM98]. In this model, success or failure of an individual packet corresponds directly to the state of a binary Markov channel. For a given packet length and transmission rate, transition probabilities between the two states depend solely on the specified average packet loss and the mobile host's speed. For our experiments, a slow mobile host (i.e., 2-10Kph) will experience slow fading, which is characterized by long average burst lengths. A faster mobile host (i.e., 25-50Kph) will experience fast fading, which is characterized by shorter average burst lengths and approaches a random loss model.

Our initial studies show that for fast fading, the effect of using the SACK protocol with double retransmissions is similar to the results presented above. This is due to the fact that for fast fading the average bust length is quite short. The benefit of using double retransmissions comes from situations where the first retransmission gets lost, but is actually at the end of a loss burst. In this case, and so the second retransmission will be successfully transmitted. In the case of slow fading, the average loss burst length is quite long, which means that it is unlikely that the first retransmission will be at the end of a loss burst. Double retransmissions for slow fading actually increase energy consumption, the mobile host is paying the overhead of unsuccessfully transmitting the second retransmission. This is the argument used in [ZR97]. This scheme introduces an energy-efficient error control strategy for mobile communications, which delays retransmissions when the channel is impaired. These results show that, since the channel is impaired and it is likely that the retransmission will also get lost, delaying the retransmission improves its chances of success.

These results demonstrate that it is important to have knowledge of the state of the communication channel in order to make intelligent power management decisions. To this end, the next section presents techniques for adaptive power management based on such information.

5 Towards Adaptive Power Management

The goal of this research is to design adaptive power management techniques aimed at minimizing the energy consumed during data transfer. The experiments described above demonstrate the effects of different configurations of a transmission protocol with varying channel qualities. Our next step is to use this information to adapt to changes in channel quality. In this section, we consider such power management in the context of a framework we developed to address the general problem of adapting communication strategies to maximize the benefit to the application, within the constraints imposed by the available network services. We identify relationships among energy consumption and the communication variables captured in this communication framework. Finally, we present a simple method for determining appropriate communication parameters based on the energy consumption associated with specific configurations. In earlier research, we have investigated the effects of other parameters (e.g., reliability) on the quality of service delivered to an application [KCKS97, KCS98].

5.1 Communication Framework

The power management techniques described in this paper are implemented in the context of an end-to-end *communication layer* [KCS98], which configures the operation of its communication protocols based on the perceived benefits to the applications using it and the constraints imposed by the underlying network service. The novel characteristic of this layer is its ability to maximize the value of the services provided to individual applications in the face of changing network characteristics. In a mobile context, power management (energy conservation) is typically a primary or even an overriding concern, and we focus on it almost exclusively in this section. Our communication framework, however, is intended to deal with *all* aspects of the service delivered to the application, of which energy usage is just one component. As we have seen in Chapter 4, with an integrated approach it need not always be the case that improved energy usage comes at the expense of other QoS characteristics.

The structure of our communication-centric resource management framework is depicted in Figure 9. The model defines several sets of interrelated variables, some of which can be controlled by the communication layer:

- *Communication Configuration:* the protocols and mechanisms used to enhance the service received from the underlying network layer. These variables capture what application-level and transport-level protocols are used (e.g. use of an ARQ protocol vs. Forward Error Control), as well as how protocol parameters are configured (e.g. block size when FEC is used). The communication layer directly controls this configuration.
- *Communication Behavior:* what the network "sees" from the communication layer. This includes traditional QoS dimensions like transmission rate, burst size, etc. These are determined by the communication configuration, and are thus under indirect control of the communication layer.
- *Network Service:* what the communication layer "sees" from the network on the receiving side. This is quantified by variables that include loss rate, delay, jitter, and cost. The parameters for network service are determined by the network but may be affected indirectly by the communication behavior. For example, an increased transmission rate may result in an increased loss rate.
- *Channel Service:* the end-to-end behavior ultimately "seen" by the application. This is determined by the network service, with enhancements applied by the communication layer.

In general, the values of the variables in these sets quantify system behavior over time intervals, rather than instantaneously.

The communication layer has several responsibilities. First, it must accurately account for the relationships among the variables described above. Second, it must monitor the values of those variables (e.g. loss rate) that are under its direct control, via periodic measurement or direct reporting from the network. Third, it must configure the protocols and parameters to maximize value to the application. And finally, it must communicate with its remote peer to ensure that consistent configurations are used at both ends.

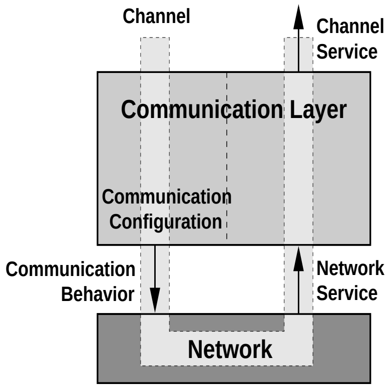


Figure 9: Power Management Architecture

In the power management context considered here, we view the transfer of each piece of data as an independent communication action for which application satisfaction is to be maximized, and assume that value to the application is primarily determined by energy consumption. The energy required to complete the action is a function of the variables listed above. In order to solve the optimization problem of minimizing energy consumption, two relationships must be examined:

- The relationship among *communication configuration*, *communication behavior*, *network service* and *channel service*: how the service delivered to the application is affected by protocol configuration, communication characteristics and the network service. For example, the use of retransmission as an error control strategy can decrease loss rate, but may increase delay. The use of forward error control, on the other hand, can decrease loss, but may require more bandwidth.
- The relationship between energy consumption and *communication behavior* and *channel service*: how protocol processing and transfer time affect energy consumption, the latter being the quantity to be optimized.

These relationships are affected by changes in communication configuration and network service. The characterization of both relationships is a major challenge in the design and implementation of the power management techniques. It might be realized through analysis, simulation, measurement, or (most likely) some combination of all three.

5.2 Optimizing Energy Consumption

Through the use of our configurable communication layer, we can configure communication protocols to optimize energy consumption. We can control these parameters to effect such optimizations. When considering transmission protocols, especially error control functionality, these parameters include the techniques used for error control (ARQ vs. FEC) as well as the parameters for these techniques. The parameters for an ARQ-based protocol might be window size or timer values, while they might be redundancy level for FEC. In Section 4, we presented a SACK-based transmission protocol with a parameter to control the number of retransmissions to send.

In order to understand how to adjust these parameters, we have developed a communication adaptation algorithm which we have specialized for power management. Communication adaptation to optimize energy consumption involves two stages. First, the correct configuration for a specific point in time is determined, given current network service availability. Second, network services are monitored for changes in service quality. The effects of such changes on energy consumption are considered to determine a new configuration.

Given a fixed network resource specification, we consider a range of possible operating parameters and choose those that minimize energy consumption. The algorithm in Figure 10 determines minimum energy consumption. We demonstrate our algorithm using FEC as a sample communication protocol, which effects protocol processing, data size and transfer time. We are currently investigating the effect of FEC on energy consumption in the context of our communication framework.

To evaluate individual parameters, we need information about the relationships among the communication variables, as discussed in Section 5.1. What we want is a mapping from communication configuration, communication behavior and network service to channel service. In general, such a mapping may be formulated empirically (using profiling techniques prior to running the application or during execution, i.e. based on recent history), or in some cases it may be devised analytically. For example, we could use the experimental results from Section 4, which allows us to map retransmission policy (communication configuration) to transfer time (channel service). For the algorithm described in Figure 10, we consider the effect of forward error control on energy consumption. The function F_{FEC} (Step (2)) supplies the mapping from redundancy level and original data size (com-

```
(0)   Best = $\infty$
(1)   for each RedundancyLevel $i$ do
(2)       $\text{DataSize}_i = F_{\text{FEC}}(\text{RedundancyLevel, OrigDataSize})$
(3)       $\text{CodingTime}_i = T_{\text{FEC}}(\text{RedundancyLevel, OrigDataSize})$
(4)       $\text{TransferTime}_i = T_{tt}(\text{DataSize}_i, \text{ChannelState})$
(5)       $\text{CommunicationEnergy}_i = E_{\text{FEC}}(\text{RedundancyLevel}, \text{OrigDataSize}) +$
              $E_{\text{comm}}(\text{DataSize}_i, \text{ChannelState})$
(6)       $\text{DeviceEnergy}_i = E_{dev}(\text{CodingTime}_i, \text{TransferTime}_i)$
(7)       $\text{MachineEnergy}_i = E_{mach}(\text{CodingTime}_i, \text{TransferTime}_i)$
(8)       $\text{Energy}_i = \text{MachineEnergy}_i + \text{DeviceEnergy}_i + \text{CommunicationEnergy}_i$
(9)       if ($\text{Energy}_i < \text{Best}$) then
(10)          level = $i$
(11)          Best = $\text{Energy}_i$
(12)  RedundancyLevel = level
```

Figure 10: Algorithm for Determinig Minimum Energy Consumption for Data Transmissions using FEC

munication configuration) to transmission data size (communication behavior). T_{FEC} (Step (3)) supplies the mapping from and redundancy level original data size to coding time (channel service) and T_{tt} (Step (4)) supplies the mapping from data size and channel state to transfer time (channel service). The next three steps (Steps (5), (6) and (7)) use the results from Steps (2), (3) and (4) to determine the components of the energy consumption. Finally, the communication framework evaluates the expected total energy consumption associated with communications that operate with these service parameters. These steps are repeated to determine the energy consumption for each possible operating point (as indicated by the loop in Step (1)). From these operating points, the communication framework, which is responsible for power management, chooses the redundancy level that minimizes total energy consumption.

The second stage of communication adaptation reacts to changes in network resources. As such changes are detected, either through information from the network or through online monitoring [RSYJ97], the communication configuration is reevaluate to determine the effects of these changes on energy consumption.

This section has presented techniques for adaptive power management based on information about the state of the network. A mobile host can use information about the effects of changes in transmission protocol parameters on the actual transmission of the data as well as on the energy consumption of the data transfer. These effects can be determined beforehand, as in the experiments presented in Section 4, and used in the optimization algorithm presented above to minimize energy consumption for data transfer.

6 Conclusions

This paper presents power-aware communication techniques aimed at energy conservation across all components of a mobile host. By understanding the effects of changes in communication parameters on the energy consumption of the entire mobile host, not only the network interface device, we are able to minimize energy consumption through appropriate parameter choices. Previous work was aimed at minimizing the energy consumption of a individual network interface device by regulating retransmissions at the transport level. The results presented in this paper demonstrate that although the energy consumed during the transmission of the extra messages is not negligible, it is not always significant in the context of the amount of energy consumed by the entire mobile host during data transfer. Our results demonstrate that under lossy network conditions, aggressive retransmission can save a high-end laptop up to 25% over a less aggressive retransmission policy.

With the dramatic increase in the use of mobile computers, the demand for more energy-efficient computers will continue to increase. Due to the diverse uses of these machines, we can only hope to accomplish successful power management through understanding the needs of the individual machines. Inherent in this diversity of use is a diversity of communication requirements, which lends itself to the type of power management techniques described in this paper.

References

[BB95] Ajay Bakre and B.R. Badrinath. I-TCP: Indirect TCP for mobile hosts. In *IEEE International Conference on Distributed Computing Systems (ICDCS) '95*, 1995.

[BPSK96] Hari Balakrishnan, Venkata Padmanabhan, Srinivasan Seshan, and Randy Katz. A comparison of mechanisms for improving TCP performance over wireless links. In *Proceedings of the SIGCOMM '96 Symposium*, 1996.

[BSAK95] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy Katz. Improving TCP/IP performance over wireless networks. In *First ACM International Conference on Mobile Computing and Networking (MOBICOM)*, November 1995.

[CPR98] Imrich Chlamtac, Chiara Petrioli, and Jason Redi. Energy conservation in access protocols for mobile computing and communication. *Microprocessors and Microsystems Journal*, 1998.

[DKM94] F. Douglis, P. Krishnan, and B. Marsh. Thwarting the power hungry disk. In *Proceedings of the 1994 Winter USENIX Conference*, January 1994.

[FF96] Kevin Fall and Sally Floyd. Simultation-based comparisons of tahoe, reno, and SACK TCP. *Computer Communications Review*, July 1996.

[GCW95] K. Govil, E. Chan, and H. Wasserman. Comparing algorithms for dynamic speed-setting of a low-power CPU. In *First ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 1995.

[HLS96] D. Helmbold, D. D. E. Long, and B. Sherrod. A dynamic disk spin-down technique for mobile computing. In *Second ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 1996.

[KCKS97] Robin Kravets, Ken Calvert, P. Krishnan, and Karsten Schwan. Adaptive variation of reliability. In *the Seventh IFIP Conference on High Performance Networking (HPN'97)*, April 1997.

[KCS98] Robin Kravets, Ken Calvert, and Karsten Schwan. Payoff adaptation of communication for distributed interactive applications. *Journal on High Speed Networking: Special Issue on Multimedia Communications*, 1998.

[KK98] Robin Kravets and P. Krishnan. Power management techniques for mobile communication. In *Fourth ACM International Conference on Mobile Computing and Networking (MOBICOM'98)*, 1998.

[KM97] S. Keshav and S.P. Morgan. SMART retransmission: Performance with overload and random losses. In *Proceedings of IEEE INFOCOM 97*, 1997.

[LKHA94] K. Li, R. Kumpf, P. Horton, and T. Anderson. A quantitative analysis of disk drive power management in portable computers. In *Proceedings of the 1994 Winter USENIX*, 1994.

[LS96] J. R. Lorch and A. J. Smith. Reducing processor power consumption by improving processor time management in a single-user operating system. In *Second ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 1996.

[Luc96] Lucent Technologies. *WaveLAN/PCMCIA Card User's Guide*, October 1996.

[MMFR96] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgement options. Request for Comments RFC 2018, Internet Engineering Task Force, October 1996.

[NSYA97] B. Narendran, J. Sienicki, S. Yajnik, and P. Agrawal. Evaluation of an adaptive power and error control algorithm for wireless systems. In *IEEE International Conference on Communications (ICC'97)*, 1997.

[RSYJ97] Daniela Rosu, Karsten Schwan, Sudhakar Yalamanchili, and Rakesh Jha. On adaptive resource allocation for complex real-time applications. In *18th IEEE Real-Time Systems Symposium, San Francisco, CA*. IEEE, December 1997.

[SCAK98] Krishna Sivalingam, Jyh-Chang Chen, Prathima Agrawal, and Shalinee Kishore. Design and analysis of low-power access protocols for wireless and mobile ATM networks. *ACM/Baltzer Journal on Special Topics in Mobile Networking and Applications (MONET)*, June 1998.

[SK96] Mark Stemm and Randy Katz. Reducing power consumption of network interfaces in hand-held devices. In *Third International Workshop on Mobile Multimedia Communications (MoMuc-3)*, December 1996.

[WWDS94] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for reduced CPU energy. In *Proceedings of the First Symposium on Operating System Design and Implementation (OSDI) '94*, November 1994.

[ZR97] Michele Zorzi and Ramesh Rao. Error control and energy consumption in communications for nomadic computing. *IEEE Transactions on Computers (Special Issue on Mobile Computing*, March 1997.

[ZRM98] Michele Zorzi, Ramesh Rao, and Laurence Milstein. Error statistics in data transmission over fading channels. *IEEE Transactions on Communications*, 46:1468–1477, November 1998.