Integrated Power Management for Mobile Computers

Abstract

Recently the mobile community has focused on techniques for reducing the energy consumption of mobile hosts. These power management techniques typically target communication devices such as wireless network interfaces, aiming to reduce the usage, and thus the 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 perform runtime adaptation of communication parameters in order to minimize the energy consumed during communication, using information about the network environment to compensate for additional energy consumption due to the dynamic nature of the services available from wireless communication devices. Our results show that power management can be achieved without sacrificing the QoS provided to the application.

1 Introduction

With the increase in the use of mobile computers, researchers have sought techniques for reducing mobile hosts' power requirements. The overall goal of power management is to prolong battery life by controlling the energy consumption of a mobile host's various devices. Past research has developed techniques for reductions in the energy consumption of disk [DKM94, HLS96, LKHA94], CPU [GCW95, LS96, WWDS94], and network interface devices [SK96, CPR98, NSYA97, KK98]. The goal of these techniques is to reduce the energy needs of individual devices. Building on such work, this paper considers the power consumption of a mobile host's communication device in the context of the total power consumed by the host during active communications. The principal contributions of our work are the following:

- *Techniques for integrated power management.* Effective power management needs to consider the effects of each conservation technique on the power required by the entire mobile host. The techniques described in this paper support such integrated power management in conjunction with mobile communications.
- *A model for integrated power measurement.* Integrated power management requires understanding the tradeoffs in using a device in different ways, and how these tradeoffs affect the energy consumption of the entire host. For the communication device on which this paper focuses, this implies the measurement of power consumption for entire 'communications actions', involving the device, the CPU, and others.
- *Novel power management techniques.* The promise of integrated power management techniques is that they can reduce total energy consumption compared to those that address individual devices. The specific technique described in this paper achieves such gains by runtime adaptation of certain communication parameters in response to dynamic changes in the services available from a wireless network. In other words, communication parameters are configured based on the energy savings the communication layer expects to realize for the mobile host, given observed conditions.
- *Experimental validation.* Our power management techniques are implemented using a communication framework that provides dynamic protocol configuration [Tanrfaa, Tanrfab]. Specific protocol parameters can be set and changed through the use of the framework's interface. Results are obtained from actual measurements of a high-end laptop using a WaveLAN to communicate with a base station.

Central to our our work is the definition of the energy cost of a *communication action*, which includes not only the energy consumed by the network interface device, but also that consumed by the mobile host itself. The goal is to minimize the energy cost of a specific communication action given information about the current communication environment and the power requirements of the mobile host. For example, consider the action of transmitting a data object from a mobile host. In this context, the use of compression increases the amount of computation required, thus increasing the energy used by the CPU, while simultaneously decreasing energy consumed by the actual transmission of the data. A non-integrated solution would never consider using compression to reduce communication-centric energy consumption, whereas an integrated technique might decide to use compression, thereby increasing CPU energy consumption to reduce total energy consumption. Conversely, data compress is not efficient on the specific data, then the resulting increase in CPU energy consumption may not be outweighed by the associated decrease in the transmission energy consumption. The integrated power management techniques presented in this paper are designed to permit consideration of such tradeoffs throughout the mobile host's operation.

We demonstrate the effectiveness of integrated power management techniques with a transport 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 transmission 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 are increased.

Consider for the moment the design of a transport protocol. The basic mechanisms we can use to build this protocol are window size, acknowledgments (ACKs, NACKs and SACKs), FEC and timers. The adjustment of window size is used to compensate for congestion and 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. How we use these mechanisms defines the behavior of the protocol and so the energy consumed by the transmission of the data.

This paper explores the use of a SACK-based transport 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 hosts. As losses approach 25%, our techniques show a 25% energy savings for a high-end laptop and predict a 20% savings for a mid-level laptop.

In Section 2, we describe our power management approach and techniques, and place both into the context of our prior work on communication adaptation. We also describe how to quantify and optimize power use. Section 3 discusses power management in the context of transport protocols, and Section 4 presents the results from our experiments with a configurable SACK-based protocol. Finally, Section 5 presents our conclusions.

2 Integrated Power Management

A number of communication-related parameters, or 'knobs', affect a mobile host's power usage. In this section, we consider power management in the context of a framework we developed to address the general problem of adapting communication strategies (i.e. configuring a communication layer or setting the 'knobs') to maximize the benefit to the application, within the constraints imposed by the available network services. In earlier research, we have investigated the effects of other 'knobs' (e.g., reliability) on the quality of service delivered to an application [Tanrfaa, Tanrfab]. For this research, we identify relationships among power consumption and the communication variables captured in this communication framework. Next, we describe in detail how power consumption may be described to enable online power management for communications.

Three sets of measurements are necessary to differentiate base power usage at idle times, with and without attached network interface devices, from power requirements during ongoing communications. Interestingly, for ongoing communications, it is less important to distinguish the contribution to total power consumption from each device than to understand the tradeoffs in terms of total power consumption for different protocol configurations for such communications. Finally, we present a simple method for determining appropriate communication parameters based on the energy consumption associated with specific configurations.

2.1 Communication Framework

The power management techniques described in this paper are implemented in the context of an end-to-end *communication layer* [Tanrfaa], 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, using 'payoff functions' at the interface between the configurable communication protocol and each application. Payoff functions characterize the relative value to the application of various quality of service characteristics. In a mobile context today, power management (energy conservation) is typically a primary or even an overriding concern, and we focus on it almost exclusively in the remainder of this paper. Our communication framework, however, is intended to deal with *all* aspects of the service delivered to the application, of which power usage is just one component. As we shall see later, with an integrated approach it need not always be the case that improved power usage comes at the expense of other QoS characteristics.

The structure of our communication-centric resource management framework is depicted in Figure 1. 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 and the 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 in our framework 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 ('set the knobs') 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.

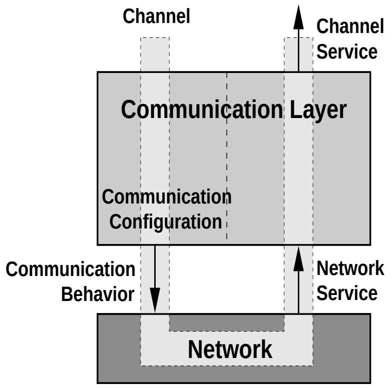


Figure 1: Power Management Architecture

In the power management context considered here, we view the transmission 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.

2.2 Power Measurement Model and Power Optimization

In order to understand how to accomplish power management for communication actions, we first need to define what we mean by the amount of energy consumed by each such action. Previous work in this area has defined this to be the amount of energy consumed by the network interface device itself [CPR98, SCAK98, ZR97]. We extend this definition to consider the energy consumed by the entire mobile host, which includes both the network interface and the rest of the system involved in the communication. Thus, our measurement model evaluates the total energy consumed by the entire system, rather than focusing on the requirements of individual devices.

Figure 2 shows a sample power 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 types of energy consumption may be differentiated when considering the energy consumption of the mobile host in Figure 2: communication-specific, device-specific and machine-specific. The first time period represents the energy consumption of the idle machine (including the CPU, disks, and other components besides the network interface device) over time. This is the machine-specific base amount, indicated by the energy level A, that will generally always be consumed, even when the machine is idle. 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, and the machine will consume some average amount of energy during this time.[1] In the next time period in the figure, a wireless network interface device is activated, and this area represents the energy consumed by 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 energy 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 the 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 energy. Finally, when the mobile host is actively communicating, as indicated by the values above energy level B, the energy used 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. For all of the experiments described in the next section, we attribute consumption above energy level B to the communication actions being performed.

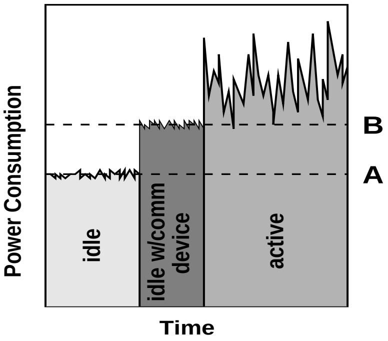


Figure 2: Example Energy Consumption

From Figure 2, it is apparent that transmission-time energy consumption can be partitioned into three sections. The area below A represents the energy consumed by the idle machine, and the area between A and B represents the energy consumed by the communication device. These two areas are only affected by the total amount of time required for communication. On the other hand, the area above B, which represents the amount of additional energy needed for actual data transmission, is affected by two parameters. The first is the amount of energy consumed by the device for data transmission (or data receptions for a receiver). This amount will generally increase linearly with data size. If data compression is performed, this amount will actually be reduced since compression reduces the amount of data to be transmitted. In a lossy network environment, this component will generally 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 compression example, any CPU usage for compression will be attributed to this amount. 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 this area.

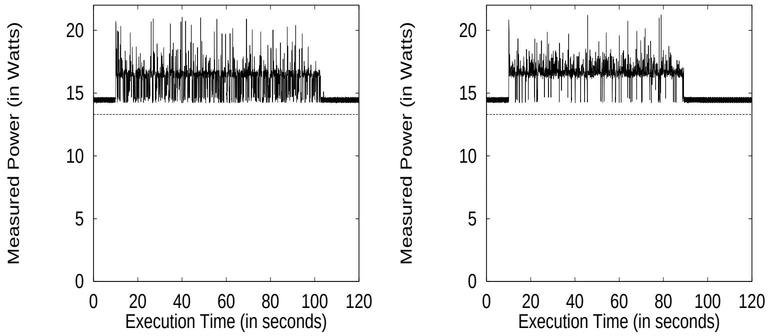
The power measurement model presented above partitions the energy consumption for data transmission into component parts such that each part can be considered independently to determine its contribution to total power consumption. Since both machine-specific and device-specific power consumption are only dependent 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

---

[1] In effect, we are ‘charging’ (no pun intended) the communication action for all system usage that occurs during that action. This is conservative, but is realistic for many applications as long as communication delays remain short enough so that it is not practical to browse the Web, for example. in parallel with some other activity.

on the configuration of the communication parameters. In Section 3, we will discuss one such example for a mobility-oriented transport protocol.

For a more specific example, consider the graphs in Figure 3 which presents two traces for the amount of energy consumed during a communication action. Both figures represent the energy consumed for the transmission of identical amounts of data. The difference lies in the configuration of the communication protocols during transmission. From these examples we can see the contributions of the components of energy consumption to the total energy consumption. Since the transmission time in Figure 3(a) is longer, the machine-specific and device-specific energy consumption is higher than in Figure 3(b). On the other hand, the communication-specific energy consumption in Figure 3(b) is greater due to some property of the communication configuration.



(a) (b) (c)

Figure 3: Comparative Energy Consumption for a Communication Action

2.3 Optimizing Energy Consumption

Through the use of our configurable communication layer, we can configure communication protocols to optimize energy consumption. These parameters are the 'knobs' we can control to effect such optimizations. When using compression, the available parameter is the compression level. When considering transport 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 block size for FEC. In Section 3, we will present a SACK-based transport protocol with a parameter to control the number of retransmissions to send.

In order to understand how to adjust these 'knobs', we have developed a communication adaptation algorithm which we have specialized for power management. Communication adaptation to optimize energy consumption involves two stages. First, power management determines a configuration for a specific point in time, given current network service availability. Second, power management monitors changes in network services. 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 following algorithm determines minimum energy consumption. We demonstrate our algorithm using compression as a sample communication protocol which effects protocol processing, data size and transmission time.

```
Best = $\infty$
for each CompressionLevel $i$ do
    DataSize$_i$ = $F_{\mathrm{comp}}$(CompressionLevel, OrigDataSize)
    TransferTime$_i$ = $F_{tt}$(DataSize$_i$)
    CommunicationPower$_i$ = $F_{\mathrm{comm}}$(DataSize$_i$)
    DevicePower$_i$ = $F_{dev}$(TransferTime$_i$)
```

```
(6)         MachinePower$_i$ = $F_{mach}$(TransferTime$_i$)
(7)         Energy$_i$ = MachinePower$_i$ + DevicePower$_i$ + CommunicationPower$_i$)
(8)         *if* (Energy$_i$ < Best) *then*
(9)             level = $i$
(10)            Best = Energy$_i$
(11)    Compression = level
```

To evaluate individual parameters, we need information about the relationships among the communication variables, as discussed in Section 2. 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. The function F_{comp} (Step (2)) supplies the mapping from compression level (communication configuration) to data size (communication behavior). F_{tt} (Step (3)) supplies the mapping from data size to transfer time (channel service). The next three steps (Steps (4), (5) and (6)) use the results from Steps (2) and (3) to determine the component parts 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, power management chooses the compression 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], power management reevaluates the communication configuration to determine the effects of these changes on energy consumption.

After discussing the relationship between transport protocols and mobility, the remainder of this paper describes an empirical approach to the implementation of such a power management framework. Section 3 discusses the effects of transport protocol configuration on energy consumption. In Section 4, we present the results from experiments with a specific transport protocol designed for energy-efficient data transmission.

3 Energy Efficient Mobile Communication

The desire to conserve energy during active communication has driven diverse power management techniques. The goal of 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. Given information about the energy consumption of the machine and the device and characteristics of the data stream (loss and reordering statistics), we can determine how best to configure the transport layer protocol. This section discusses the effects of transport protocol design on energy consumption and presents a breakdown of energy consumption information for transport protocols that can be used in our power management framework.

3.1 The Mobile Environment

In order to successfully manage power, we need to consider the environment in which the mobile host exists. If the mobile host is communicating primarily with a stationary base station, then we will only need to target our power management efforts at the one mobile host. On the other hand, if the mobile host exists in some type of ad hoc network where it is communicating with other mobile hosts, we will need to consider power management techniques for both ends of the communication.

For this discussion, we will focus on a mobile host that primarily communicates with a stationary base station. In this case, power management techniques may differ if the mobile host is mostly sending or receiving data or some combination of both. The parameters of a transport protocol may differ for power management techniques aimed at a sender versus those aimed at a receiver. For example, a sender may be concerned with the energy overhead of transmitting unnecessary retransmissions, while a receiver may be concerned with the overhead of transmitting unnecessary acknowledgments. Both sender and receiver will be concerned with the amount of time the transmission takes. We consider both sender and receiver scenarios in the experiments discussed in Section 4.

The introduction of wireless links into communication systems based on wired links has posed a number of problems. These problems include different loss characteristics and different bandwidth capabilities on the wired and the wireless line, 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 existing in the communication route to a mobile host. 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]. In this paper, 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.

3.2 Mobility and Transport Protocols

Transmission time power management focuses on the energy consumed by the active communications of a mobile host in a dynamic network environment. In this section, we will discuss the effect of changes in network service on the performance and therefor on the energy consumption of transport protocols. We assume that there is some fixed level of energy consumed by the simple transmission 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 transmission time. We approach this problem by considering the effects of changes in the parameters of the transport protocol being used and their effect on these three factors.

The design of TCP and other such transport protocols has included transmission 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 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. This lead us to consider the use of 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.

Basic timers are also used to determine when the sender should send a retransmission of a presumably lost message. The timer value for this situation should be based on an estimate of the round trip time. The accuracy of the timeout value determines the performance of the protocol. For the sender, a timeout value that is too short will cause the unnecessary retransmission of potentially successfully received messages, while a timeout value that is too long will unnecessarily delay the retransmission of lost messages.

3.3 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. 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 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 the 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*.

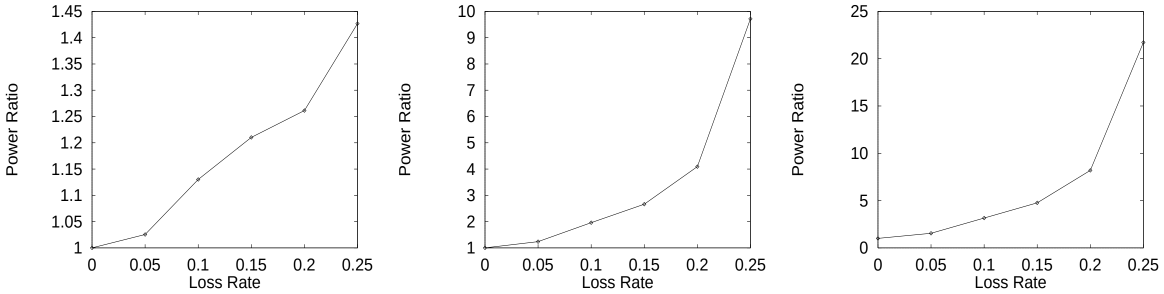
Consider again the breakdown of energy consumption during data transmission, as indicated in Figure 2. Device-specific and machine-specific energy consumption will only be affected by unnecessary wait time, since both are time dependent. Communication-specific energy consumption, on the other hand, is only affected by unnecessary retransmissions and unnecessary acknowledgments. Given this separation of contributing factors, we need to target multiple, and potentially conflicting, parameters.

In order to determine the effect of message loss on energy consumption, we set up a simple experiment run between two mobile hosts using wireless Ethernet devices. We transmitted 12MBytes 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 transmission time (and so the device-specific and machine-specific energy consumption) increased over 500%.

Since TCP is a conservative transport protocol, it reacts to loss by slowing down the transmission of new data. It can be argued that this type of protocol is inefficient for wireless networks where losses of up to 30% can be experienced. Additionally, much of the increase in time can be attributed to the effect of congestion control techniques. We simply use this as an example to show the overhead for communication-specific energy consumption for a protocol that is known to be inefficient in the face of excessive loss and that is aimed to minimize unnecessary retransmissions.

Figure 4 presents the three levels 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 transmission 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. Figure 4(b) shows us the communication-specific plus the device-specific energy consumption. In 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(c)); note the dramatic change in scale from Figure 4(a) to Figure 4(c). Curves (b) and (c) 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.

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. In Section 4, we will discuss a more wireless-friendly transport protocol and show similar, though not as dramatic, results.



(a) (b) (c)

Figure 4: Energy Consumption for TCP

4 Experiments

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.

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 5, 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.

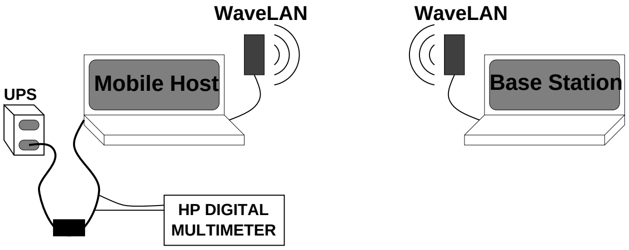


Figure 5: Experimental Setup

To determine the mobile host's energy consumption, we monitor the current being drawn from the transformer by the host. Figure 6 shows the output of the multimeter over time for data being transmitted 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 6, the solid line near 14W represents the power consumed by 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 we collect 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 and hard disk due to the communication.

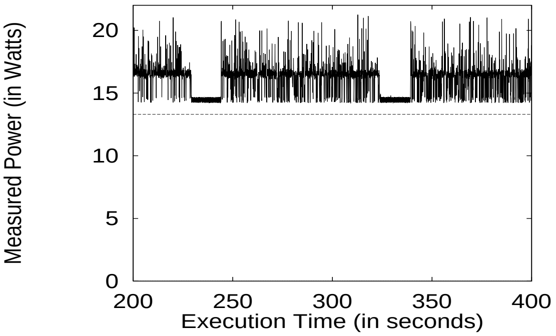


Figure 6: Sample Output from Multimeter

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 implied 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 determine that the NEC Versa consumes around 13.3W when idle. With the WaveLAN card inserted, the laptop consumes 14.8W when idle. From this information, we consider any energy consumed over the 13.3W idle energy consumption to be contributed to the ongoing communication. Additionally, we note that the energy consumption often peaks over 16.3W, which is the expected energy consumption for a transmission. This additional energy consumption over 16.3W can be attributed to the use of the CPU during active data transmission. As mentioned earlier, we also consider this energy consumption to be part of the overall communication energy consumption.

4.2 Results

The goal of our experiments is to demonstrate the tradeoff between minimizing the number of unnecessary retransmissions vs. minimizing the unnecessary wait time and to 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. This protocol has two options it can choose between. 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 transmission time in environments where, if the first retransmission gets lost, the second retransmission still has a good chance of being transmitted successfully. Figure 7 shows the expected running times of such a protocol for each option. As we can see, the use of double retransmissions can reduce the transmission time by 5% of the original transfer time for 10% loss and by 53% for 25% loss. This determination of transfer time based on the specification of protocol parameters can now be used in our optimization algorithm described in Section 2. This information allows us to map retransmission policy (communication configuration) to transfer time (channel service). Information about transfer time also allows us to determine both device-specific and machine-specific energy consumption.

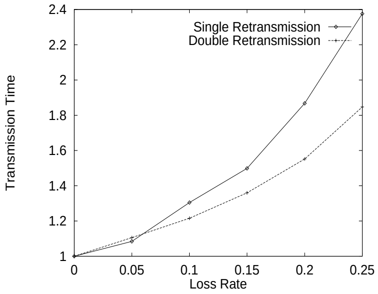


Figure 7: Running Time

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 9(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, the double retransmission costs more energy than the single retransmissions due to an increase in the number of unnecessary retransmissions. Figure 8 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 9(b)), we can see that the energy consumption for the single and double retransmission has almost evened out. This can be explained by the fact that the device-specific energy consumption is solely dependent on transmission time. Finally, the inclusion of the machine-specific energy consumption shows us that the use of the double retransmissions can actually save us energy (Figure 9(c)). For both Figure 9(b) and Figure 9(c), the increased energy consumption for the single retransmission is due to the increase in the unnecessary wait time. The breakdown of energy consumption can now be fed back into our optimization algorithm and used to determine the appropriate protocol configuration for a given loss rate.

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

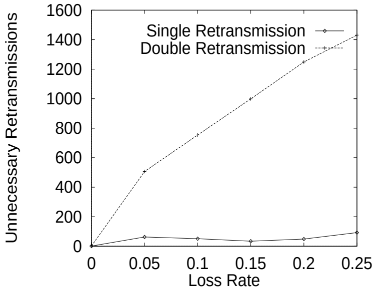


Figure 8: Unnecessary Retransmissions

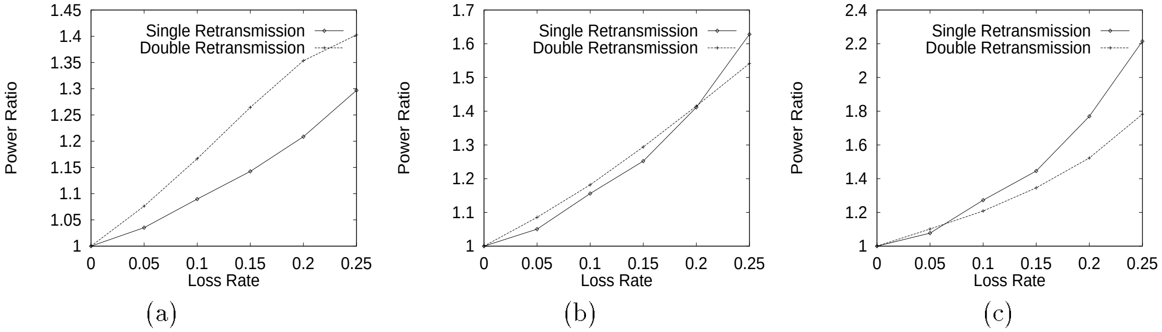


Figure 9: Energy Consumption Components

consumes around 13.3W when idle. If we consider a machine like a the Toshiba Libretto 60 which consumes about 7W when idle, we would expect to see results like those in Figure 10. This shows us, that although we need to have knowledge about the specific machine we are using, such techniques are able to solve problems across machines with different energy consumptions.

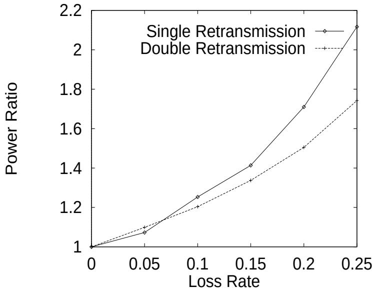


Figure 10: Total Energy Consumption

A limitation of our experimental results is the assumption that losses are random. The scheme presented in [ZR97] introduces an energy-efficient error control strategy for mobile communications, which delays retransmissions when the channel is impaired. Those 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. This is based on information about the loss characteristics of fading wireless channels. By extending our experiments to use a simple Rayleigh fading model for error loss, we observed results similar to those attained for a channel with random losses. From this observation, we believe we can extend our results to more generic loss models in the future.

5 Conclusions

This paper presents communication-centric power management 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 transmission. 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.

[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 and 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.

[Tanrfaa] Title and authors names removed for anonymity.

[Tanrfab] Title and authors names removed for anonymity.

[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.