

Minimizing Energy Consumption for Wireless Computers in Moby Dick

Paul J.M. Havinga, Gerard J.M. Smit

University of Twente, dept. of Computer Science
P.O. Box 217, 7500 AE Enschede, the Netherlands
e-mail {havinga,smit}@cs.utwente.nl

Abstract

The Moby Dick project is a joint european project to develop and define the architecture of a new generation of mobile hand-held computers, called Pocket Companions. The Pocket Companion is a hand-held device that is resource-poor, i.e. small amount of memory, limited battery life, low processing power, and connected with the environment via a network with variable connectivity. Because battery life is limited and battery weight is an important factor for the size and the weight of the Pocket Companion, energy management plays a crucial role in the architecture. In our architecture we apply several supplementary power reduction techniques on all levels of the system, including the operating system and applications. The two main themes are 1) to avoid wasteful activity, and 2) to exploit locality of reference.

We put considerable effort in reducing energy consumption of communication interfaces. There are several ways to reduce energy consumption: e.g. by system decomposition, using hybrid networking, and by applying power aware MAC protocols.

1 INTRODUCTION

The Moby Dick project [7] is a joint european project (Esprit Long Term Research 20422) to develop and define the architecture of a new generation of mobile hand-held computers. The design challenges lie primarily in the creation of a single architecture that allows the integration of security functions, externally offered services, personality, and communication. Research issues include: security, energy consumption and communication, hybrid networks, data consistency, and environment awareness.

The requirement of portability of hand-held computers and portable devices places severe restrictions on size and power consumption. Even though battery technology is improving continuously and processors and displays are rapidly improving in terms of power consumption, battery life and battery weight are issues that will have a marked influence on how hand-held computers can be used. These devices often require real-time processing capabilities, and thus demand high throughput. The increasing levels of performance and integration that is required will be accompanied by increasing levels of energy consumption. Without a significant energy reduction techniques and energy saving architectures, battery life constraints will limit the capabilities of a Pocket Companion. The main theme of our approach is: enough performance for minimal energy consumption. This is in contrast to the main stream of current research in computer systems which aims at the *highest performance*, and where energy consumption is of minor concern. Even though battery technology, processors and displays are improving continuously in terms of power consumption, the every day use of the Pocket

Companion will require a lot of energy; and thus battery life and battery weight remain important issues. More extensive and continuous use of network services will only aggravate this problem since communication consumes relatively much energy. Research is needed to provide intelligent policies for careful management of the power consumption while still providing the appearance of continuous connections to system services and applications.

Related research

Several researchers have studied the power consumption pattern of mobile computers. However, because they studied different platforms, their results are not always in line, and sometimes even conflicting. Stemm et al. concluded that the network interface consumes at least the same amount of energy as the rest of the system [11]. If the computer is able to receive messages from the network even when it is 'off', the energy consumption increases dramatically. Laptops use several techniques to reduce this energy consumption, primarily by turning them off after a period of no use, or by lowering the clock frequency.

While low-power components and subsystems are essential building blocks for portable systems, little effort has been directed towards *dedicated low-power hardware architectures* by considering the system as a whole. A system wide approach is beneficial because there are dependencies between subsystems, e.g. optimization of one subsystem may have consequences for the energy consumption of other modules.

Different architectures have been proposed to address multimedia computing. These approaches are based on high-performance technology and are mostly simple extensions to current architectures. These systems fail to exploit the opportunities for energy reduction offered by multimedia. Systems like the InfoPad [10] and ParcTab [4] are designed to take advantage of high-speed wireless networking to reduce the amount of computation required on the portable. These systems are a kind of portable terminal and take advantage of the processing power of remote compute servers. This approach simplifies the design and reduces power consumption for the processing components, but significantly increases the network usage and thus also increases energy consumption. These systems also rely on the availability of the network and cannot be used when not connected. UCLA has constructed a network testbed [6] that uses a hardware architecture to localize data for both communication and video. In this way the data streams are reduced and efficiently transferred directly to their destination. Abnous and Rabaey propose an architecture for signal processing applications that is flexible and uses low power [1]. The architecture consists of a control processor surrounded by a heterogeneous array of autonomous, special purpose satellite processors. The computational demand

on the control processor is minimal, its main task is to configure the system and manage the overall control flow of a given signal processing algorithm. The satellite processors perform the dominant, energy-intensive computational tasks of algorithms. The granularity of these tasks is relatively small. Some examples include address generators, multiply-accumulate processors for computing vector dot products, etc.

2 ENERGY REDUCTION TECHNIQUES

Throughout this paper, we discuss ‘power consumption’ and methods for reducing it. Although they may not explicitly say so, most designers are actually concerned with reducing energy consumption. This is because batteries have a finite supply of energy (as opposed to power, although batteries also put limits on peak power consumption as well). Energy is the time integral of power; if power consumption is a constant, energy consumption is simply power multiplied by the time during which it is consumed. Reducing power consumption only saves energy if the time required to accomplish the task does not increase too much. A processor that consumes more power than a competitor may or not may not consume more energy for a certain program. For example, even if processor A’s power consumption is twice that of processor B, A’s energy consumption could actually be less if it can execute the same program more than twice as quickly as B.

2.1 Design flow

The design flow of a system constitutes of various levels of abstraction. Research has shown that there is no single approach for reducing energy in systems like the Pocket Companion [2]. When a system is designed with the emphasis on power optimization as a performance goal, then the design must embody optimization at all levels of the design flow. In general there are three levels on which energy reduction can be incorporated. The *system level*, the *architecture level*, and the *technological level*. First of all, as the most effective design decisions stem from the architectural and system level, a careful design at these levels can reduce the power consumption considerable. For example, at the system level inactive modules may be turned off to save power. At the architectural level, parallel hardware may be used to reduce global interconnect and allow a reduction in supply voltage without degrading system throughput. Furthermore, we have to use components that use the latest developments in low power technology.

The system and architecture have to be designed targeted to the possible reduction of energy consumption at the gate level. An important aspect of the design flow is the relation and feedback between the levels. Given a design specification, a designer is faced with several different choices on different levels of abstraction. The designer has to select a particular algorithm, design or use an architecture that can be used for it, and determines various parameters such as supply voltage and clock frequency. This multi-dimensional design space offers a large range of possible trade-offs. At the highest levels the most influence on the properties of a design is obtainable. Therefore the most effective design decisions derive from choosing and optimizing architectures and algorithms at the highest levels. It has been demonstrated by several researchers [10] that system and architecture level design decisions can have dramatic impact on

power consumption. However, when designing a system it is a problem to predict the consequences and effectiveness of design decisions because implementation details can only be accurately modelled or estimated at the technological level and not at the higher levels of abstraction.

Energy reduction is not just a problem of the power-conscious hardware designer, but also involves careful design of the operating system and application programs. Furthermore, because the applications have direct knowledge of how the user is using the system, this knowledge must be penetrated into the power management of the system.

2.2 Low power system level design

The two main themes that can be used for energy reduction at system level are to avoid wasteful activity, and to exploit locality of reference.

The component that contributes a significant amount of the total energy consumption is the *interconnect*. Experiments have demonstrated that in designs, about 10 to 40% of the total power may be dissipated in buses, multiplexers and drivers. This amount can increase dramatically for systems with multiple chips due to large off-chip bus capacitance. The power consumption of the interconnect is highly dependent on algorithm and architecture-level design decisions. Two properties of algorithms are important for reducing interconnect power consumption: locality and regularity.

Locality relates to the degree to which a system or algorithm has natural isolated clusters of operation or storage with a few interconnections between them. Partitioning the system or algorithm into spatially local clusters ensures that the majority of the data transfers take place within the clusters and relatively few between clusters. The result is that the local buses are shorter and more frequently used than the longer highly capacitive global buses.

Regularity in an algorithm refers to the repeated occurrence of computational patterns. Common patterns enable the design of less complex architecture and therefore simpler interconnect structure (buses, multiplexers, buffers) and less control hardware.

For *multimedia* applications in particular, there is a substantial reduction in energy consumption possible as the computational complexity is high, have a regular and spatially local computation, and the communication between modules is significant. Improving the energy efficiency by exploiting *locality of reference* and using *efficient application-specific modules* therefore has a substantial impact on a system like the Pocket Companion.

3 MOBY DICK ARCHITECTURE

We use several supplementary approaches to reduce energy consumption in the Moby Dick architecture. In this paper we focus on the system architecture, networking, and the integration of Quality of Service and power management. Our approach is based on extensive use of power reduction techniques at all levels of system design and has a number of premises:

- An architecture with a general purpose processor surrounded by a set of heterogeneous programmable modules, each providing an energy efficient implementation of dedicated tasks.

- A reconfigurable internal communication network that exploits locality of reference and eliminates wasteful data copies.
- A system design that avoiding wasteful activity: e.g. by using *autonomous* modules that can be powered down individually and are data driven.
- A wireless communication system designed for low energy consumption by using *intelligent* network interfaces that can deal efficiently with a mobile environment, by using a power aware network protocol, and by using a MAC protocol that minimizes the ‘on-time’ of network interfaces.
- A *Quality of Service framework* for integrated management of the resources of the Pocket Companion in which each module has its own - dedicated - *local* power management. The operating system will control the power states of devices in the system and share this information with applications and users.

3.1 Pocket Companion system architecture

At the system level the locality of reference principle is applied to divide the functionality of the system into dedicated modules. When the system is decomposed into application-specific coprocessors the data traffic is reduced, for instance because unnecessary data copies are removed. For example, in a system where a stream of video data is to be displayed on a screen, the data can be copied directly to the screen memory, without going through the main processor.

Furthermore, general purpose processors often have to perform tasks for which they are not ideally suited. Although they can perform such tasks, dedicated implementations may take considerably shorter executing time, and might be less energy demanding. Application-specific integrated circuits (ASICs) or dedicated processors in combination with a general purpose processor offer an attractive alternative approach. A system designer uses the processor for portions of algorithm for which it is well suited (e.g. initialization), and applies an application-specific coprocessor (e.g. custom hardware) for other tasks.

Figure 1 shows an example of such a system. Each subsystem

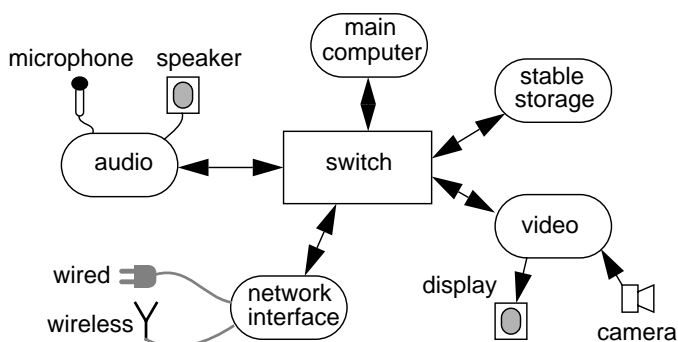


Figure 1: Pocket Companion system architecture

needs to be designed to balance the goals of the whole system and not just to optimize a specific subsystem. Each module must be optimized - apart for its main task - for energy consumption. It can have a separate clock, voltage control, power-down modes, and dedicated features to save energy. This is a good example of the difference between power and energy: although the application-specific coprocessor may actually consume more power than

the processor, it may be able to accomplish the same task in far less time, resulting in a net energy savings. The application specific coprocessor can offload the main processor with task like JPEG and MP3 decoding, encryption, and some network protocol handling. An MPEG chip can handle video much more energy efficient than a general purpose processor. A processor needs a lot of energy hungry bus cycles to perform the same operation, and hence requires a lot of power.

3.2 Energy reduction techniques for communication

The wireless network interface of a mobile computer consumes a significant fraction of the total power [11]. It is not sufficient to minimize the energy consumption during transmission and reception of data only, since measurements show that on typical applications like a web-browser or e-mail, the energy consumed when the interface is on and idle is more than the cost of transceiving packets. This is because the interface is generally longer idle than actually receiving and transmitting packets.

There are a number of techniques that can be applied for the system and architecture design of a wireless communication system that can be used to reduce power consumption. A careful design of *all* network layers is required. In this section we will elaborate on the techniques used in the Moby Dick project for reducing energy consumption in communication. There are several ways to achieve this: e.g. by system decomposition, by using hybrid networking with low power short range networks, and by applying power aware MAC protocols.

System decomposition

In normal systems much of the network protocol stack is implemented on the main processor. Thus, the network interface and the main processor must always be on for the network to be active. Because almost all data is transported through the processor, performance and energy consumption is a significant problem.

In a communication system locality of reference can be exploited by decomposition of the network protocol stack and cautious management of the data flow. This can reduce the energy consumption for several reasons:

- First, unnecessary data copies between successive layers of the protocol stack can be eliminated. This eliminates wasteful data transfers over the global (!) bus, and thus saves much dissipation in buses, multiplexers and drivers.
- Secondly, dedicated hardware can do basic signal processing and can move merely the necessary data directly to its destination, thus keeping data copies off of the system bus. Moreover, this dedicated hardware might do its tasks much more energy efficient than a general purpose processor.
- Finally, a communications processor can be applied to handle most of the lower levels of the protocol stack, thereby allowing the main processor to sleep for extended periods of time without affecting system performance or functionality.

This decomposition can also be applied beyond the system level of the portable: in our approach certain functions of the system can be migrated from the portable system to a remote server that has plenty of energy resources. This remote server handles those functions that can not be handled efficiently on the

portable machine. For example, a base station could handle parts of the network protocol stack in lieu of the mobile. The remote server has a private dedicated communication with the mobile so that the mobile units can use an internal, light weight, protocol to communicate with the base station rather than TCP/IP or UDP. The net result is saving in code and energy. In such a system it is also efficient to adapt the protocols for the specific environment it is used in. For example, wireless networks have a much higher *error rate* than the normal wired networks. In the presence of a high packet error rate, some network protocols (such as TCP) may overreact to packet losses, mistaking them for congestion. This leads to backing off to a lower transfer rate which increases the energy consumption because it leads to a longer transfer time. Any protocol that leaves a mobile receiver idle unnecessarily wastes energy. The limitations of TCP can be overcome by a more adequate congestion control during packet errors [9]. Buffering of data on a base station can be used to perform only local retransmissions that are caused by errors in the wireless network.

In order to save energy a normal mode of operation of the mobile will be a sleep or power down mode. To support full connectivity while being in a deep power down mode the network protocols need to be modified. Store-and-forward schemes for wireless networks, such as the IEEE 802.11 proposed sleep mode, not only allow a network interface to enter a sleep mode but can also perform local retransmissions not involving the higher network protocol layers. However, such schemes have the disadvantage of requiring a third party, e.g. a base station, to act as a buffering interface.

In the higher level protocols of a communication system caching and scheduling is used to control the transmission of messages. In a situation with varying and multiple network connectivity it may be wise to prefetch some information or postpone the actual transmission until the quality of the connection is better, or until another, more power economic, network is available. An application can for example schedule the times to turn on the processor when it is connected to a wired network so that the application can download information from the network when it consumes less energy or does not need its batteries.

Hybrid network control and locality

Portable computers need to be able to move seamlessly from one communication medium to another, for example from a GSM network to an in-door network, without rebooting or restarting applications. Applications require that networks are able to determine that the mobile has moved from one network to another network with a possible different QoS. The network that is most appropriate in a certain location at a certain time depends on the user requirements, network bandwidth, communication costs, energy consumption etc. The system and the applications might adapt to the *cost* of communication (e.g. measured in terms of ampère-hours or telephone bills).

Over short distances, typically of up to five metres, high-speed, low-energy communication is possible. Private houses, office buildings and public buildings can be fitted with 'micro-cellular' networks with a small antenna in every room at regular intervals, so that a Pocket Companion never has to communicate over a great distance — thus saving energy — and so that the bandwidth available in the aether does not have to be shared with large

numbers of other devices - thus providing high aggregate bandwidth. Over large distances (kilometres rather than metres), the Pocket Companion can make use of the standard infrastructures for digital telephony (such as GSM).

The Moby Dick project does research on low-power short range wireless ATM networks based on near-field RF coupling [6]. We have demonstrated a wireless link that delivers a bandwidth of 1 Mbps per cell. Cells have the size of a single office room.

A nano-cellular system has a number of advantages:

1. Because the distance between base stations and mobiles is small, a low transmission power is sufficient.
2. A small cell size can be used to locate mobiles and/or people in a building. A condition is that the location boundaries are well-defined. This can be obtained when the transmissions does not pass through walls (very high-frequency RF or IR) or have a rapid spatial decay of field strength (such as near-field RF coupling). Knowing where people are is the key principle for building location-aware applications.
3. Small cells imply that users can utilise the full capacity of distinct cells, thereby attaining a high bandwidth density (Mbps/m²). The total aggregate bandwidth of an entire office building will be the number of cells times the bandwidth per cell.

Power aware MAC protocol

The structure of current wireless networks may cause needless energy consumption. In a wireless system the medium access protocols can be adapted and tuned for low energy consumption. We use a power aware TDMA protocol that coordinates the delivery of data to receivers. The basic objective is that the protocol tries to minimize all actions of the network interface, i.e. minimize 'on-time' of the transmitter as well as the receiver. A base station is responsible for traffic scheduling. Mobiles with scheduled traffic are indicated in a list, which allows mobiles without traffic to rapidly reduce power. As switching between states (i.e. off, idle, receiving, transmitting) consumes time and energy, the number of state-transitions have to be minimized. By scheduling bulk data transfers, an inactive terminal is allowed to doze and power off the receiver.

A base station dictates a frame structure within its range. A frame consists of a number of data-cells and a traffic control cell. (Note that the frame structure influences the latency. This is an example of a trade-off between energy consumption and QoS.) The traffic control is transmitted by a base station and contains the information about the subsequent data-cells, including when the next traffic control cell will be transmitted.

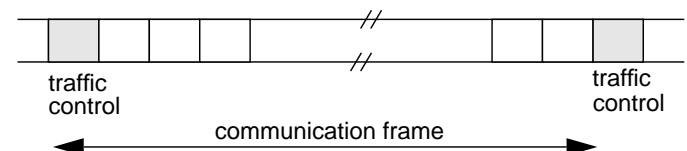


Figure 2: example of a TDMA frame structure

The approach described above leads to a number of low-power mechanisms, each with their particular advantages.

1. Mobiles with scheduled traffic are indicated in the traffic

control cell, which allows mobiles without traffic to rapidly reduce power. This mechanism requires that all mobiles and the base station are synchronized to be able to receive the traffic control cell in time.

2. By explicitly scheduling all bulk data transfers, a terminal is allowed to doze (and power off the receiver) as long as the network interface is reactivated at the scheduled time to transceive the data at full speed.
3. The overhead and energy consumption involved in the frame mechanism with traffic control, can be reduced when the frame size can be adapted to the situation. For example in case of a room in which only one mobile communicates, the frame size can be increased. There is a trade-off between frame size and the latency. When a low latency is required the frame size can be adapted accordingly.
4. Some applications require a distribution of information to a group of users. A multicast or broadcast mechanism can reduce energy consumption since the information is sent only once, and the base station - with plenty of energy - disseminates the information. Notice that the performance (total aggregated throughput) is increased as well. The network and MAC protocols need to be adapted for these mechanisms.

3.3 QoS and power management

If one investigates by what methods applications can adapt their QoS, one notices that, in order to bridge substantial changes in resource allocation (CPU, energy consumption and network bandwidth are the resources most affected), merely changing parameters is not sufficient. Better results can be obtained by changing algorithms. In the Moby Dick architecture, Quality of Service (QoS) is a framework to model integration and integrated management of all the system services and applications in the Pocket Companion. The consumption of resources by one application might affect other applications, and as resources run out, all applications are affected. If the availability of a resource changes, whether it be a file, CPU cycles, or energy consumption, applications that use them are notified, and they can adapt their behaviour. For example an application that maintains a distributed diary would request, for its highest QoS, to make use of a consistent view of its files, but, if this cannot be made available due to a network partition, it would accept an inconsistent version as the next best thing. Since communication bandwidth, energy consumption and application behaviour are closely linked, we believe that a QoS framework is a sound basis for integrated management of the resources of the Pocket Companion.

In order to integrate power awareness in the QoS framework, changes must be made to hardware, drivers, firmware, operating system, and applications. The system needs to be flexible, and have several implementations of a function of which one can be chosen depending on the QoS and available resources. The operating system will control the power states of devices in the system and share this information with applications and users.

4 CURRENT STATUS AND FUTURE DIRECTIONS

Currently we are designing the system architecture of the Pocket Companion. Application specific coprocessors will be built with standard components, like microcontrollers and

FPGAs. Because the system is decomposed of dedicated application specific subsystems that are connected with each other via the switch, energy consumption is reduced and - although not a primary goal - performance is increased. A testbed is being build to verify our statements and evaluate the results.

In cooperation with Nedap N.V. we have built a wireless link based on near-field radio [8]. We have designed, analysed and simulated a real-time, energy efficient MAC protocol for this link [5]. We are able to implement, test and evaluate several mechanisms for low power MAC protocols. Several measurements will be made e.g. concerning energy consumption, error rates and mode switching delays. The flexibility of our network interface allows experimenting with different system decompositions and network protocols.

As hardware architecture and system software are related, we also experiment with system software. The operating system *Inferno* from Lucent Technologies [3] is quite well suited for this purpose. In *Inferno* applications and system may be split easily - even dynamically - between client and server. Certain functions of the system can be migrated from the portable system to a remote server. The remote server handles those functions that can not be handled efficiently on the portable machine. Using *Inferno* and the hardware architecture of the Pocket Companion gives us a system that is modular and can be adapted to performance and power requirements. We are working on real time scheduling and a QoS manager in *Inferno*.

5 REFERENCES

- [1] Abnous A, Rabaey J.: "Ultra-Low-Power Domain-Specific Multimedia Processors," Proceedings of the IEEE VLSI Signal Processing Workshop, San Francisco, October 1996.
- [2] Havinga P.J.M., Smit G.J.M.: "Low power system design techniques for mobile computers", internal report University of Twente, 1997
- [3] "Inferno reference manual", Lucent Technologies 1997, document id: TM01FR10, see also <http://cruel.com/vanni/>
- [4] C. Kantarjiev et al.: "Experiences with X in a wireless environment", Mobile and location-independent computing symposium, Cambridge MA, August 1993.
- [5] Linnenbank, G.R.J. et al.: "A request-TDMA multiple-access scheme for wireless multimedia networks", Proc. MoMuC-3, 1996.
- [6] W. Mangione-Smith, et al.: "A low power architecture for wireless multimedia systems: lessons learned from building a power hog", Proceedings of the international symposium on low power electronics and design, August 1996
- [7] Mullender S.J., Corsini P., Hartvigsen G. "Moby Dick - The Mobile Digital Companion", LTR 20422, Annex I - Project Programme, December 1995 (see also <http://www.cs.utwente.nl/~havinga/pp.html>).
- [8] van Opzeeland, M., Poortinga, R.: "Design and realization of a wireless ATM-network", Msc. thesis University of Twente, Enschede, May 1997.
- [9] L.Rizzo: "Effective Erasure Codes for Reliable Computer Communication Protocols", ACM Computer Communication Review, Vol. 27- 2, pp 24-36, April 1997
- [10] Sheng S., Chandrakasan A., Brodersen R.W.: "A Portable Multimedia Terminal", IEEE Communications Magazine, pp. 64-75, vol. 30, no. 12, Dec., 1992.
- [11] M. Stemm, et al. "Reducing power consumption of network interfaces in hand-held devices", Proceedings MoMuC-3, 1996.