Promoting Power to a First Class Metric in Network Simulations

Muhammad Hamad Alizai, Georg Kunz, Olaf Landsiedel, Klaus Wehrle
Distributed Systems Group, RWTH Aachen University

Abstract

Accurate prediction of energy consumption early in the design process is essential to efficiently optimize algorithms and protocols. However, despite energy efficiency gathering significant attention in networking research, limited effort has been invested in providing requisite evaluation tools and models. Hence, developers demand powerful evaluation tools to assist them in comparing new communication paradigms in terms of energy efficiency, and minimizing the energy requirements of algorithms. In this paper, we argue for promoting energy to a first class metric in network simulations. We explore the challenges involved in modelling energy in network simulations and present a detailed analysis of different modelling techniques. Finally, we discuss their applicability in high-level network simulations.

1 Introduction

Although power has become an increasingly important factor in the design of distributed systems, it is not dealt with as a key evaluation metric in network simulations. However, measuring the power consumption of distributed algorithms early in the design phase is essential to ensure that later prototypes meet given power requirements. If otherwise these prototypes do not comply with power related requirements, re-engineering them later in the development process is generally a complicated and costly task. Furthermore, previous studies [1], [2] have demonstrated that the choice of algorithm and higher-level software decisions significantly influence the overall power consumption of a distributed system.

In recent years, an array of solutions were proposed for modelling power consumption, ranging from transistor- and micro-architecture level to high-level black-box models. However, most of the existing efforts focus on modelling power consumption at runtime to enable dynamic optimizations in computer systems. For example, runtime models at transistor- and micro-architecture level [3], [4] are used to predict the power consumption of a hardware platform (or a device) and are seemingly not suitable for evaluating large applications from power perspective [5]. Similarly, full-system runtime models benefit from the availability of fine-grained system information, such as CPU utilization, CPU performance counters, and OS-reported utilization of different hardware components. This level of information about the target platform is not at hand in simulations. Therefore, an already difficult task - modelling power consumption - becomes even more complicated to be realized in simulations.

In this paper, we argue for promoting energy[1] to a first class metric in network simulation. The discussion is focused on the following three points: (1) a detailed problem analysis is performed to identify the major consumers of power in distributed applications and protocols, (2) the feasibility of relevant modelling techniques – that capture the impact of hardware to a certain level and the executed code on the energy consumption - is explored to enable energy modelling in widespread networks simulation tools such as ns-3, OMNeT++, and OPNET Modeller, and (3) pertinent use-cases are discussed to highlight the need for considering energy early in the design process to facilitate proper design decisions. We seek purposive feedback to meet the challenges presented in this paper and incorporate power as a first class metric in distributed system design.

We start with exploring existing approaches and their relevance to communication systems in Section 2. A detailed problem analysis is performed in Section 3, before the feasibility and accuracy of different modelling approaches is presented in section 4. Section 5 presents salient use-cases, and we conclude our discussion in section 6.

2 Background

Power consumption has gathered significant attention in networking research. However, the research community has mainly been active in runtime power management [6], [7] and proposing schemes [8][9] for power reduction in networks, especially the Internet. The step that has been bypassed is modelling power in network simulations, which we believe forms the basis to leverage energy-efficiency as a core design component.

Network simulation tools, such as ns-3 and OMNeT++, are predominantly used for protocol evaluation but they do not provide any relevant insight into the power-related behaviour of algorithms. As a result, protocol optimizations are described by traditional factors like throughput, network delays and packet collisions, while power related optimizations are mostly out of question, especially at the earlier stages of the development life-cycle. Recent extensions [27], [28] to these tools concentrate on modelling batteries or provide abstract models to predict the energy consumption of communication related behaviour of a protocol. However, these approaches do not explicitly model the CPU and completely abstract from the hardware effects

[1] We use the term "power" and "energy" interchangeably throughout this paper.

that significantly impact the overall energy consumption of a communication system.

Existing work in simulating power consumption contributes little to the networking domain. For example, cycle- or instruction-accurate simulation models [10], [11] are typically used to evaluate hardware systems by tracking changes in power consumption across each cycle or instruction, respectively. However, it is hard to argument in favour of such simulation models in networking because they rely on micro-architecture level knowledge of a particular hardware platform and possess limited simulation-speed and scalability. Approaches, such as SoftWatt [12] model the power consumption of a complete system by means of validated energy models of each hardware component. However, such approaches are also platform dependent and require the complete implementation of applications and the OS for simulation. Similarly, these approaches target single systems and are unable to model the dynamic behaviour of distributed systems.

Hardware specific operations still dominate the overall power consumption in computer and network systems. For example, Wang et. al [13] report that 60% of the energy cost for transmission or reception is accounted for copy operations across the system bus between the kernel and network interface cards. Therefore, most of the existing efforts to reduce the power consumption focus on the development of energy efficient hardware. However, we observe an increasing availability of programmable network hardware, in which functionality is moved from hardware to software. We expect a shift in energy consumption from hardware towards software, therefore increasing the importance of energy efficient software. Hence, efforts at the hardware level need to be complemented equally at the software level to achieve maximum energy savings. In contrast to existing work, we focus on modelling the power consumption of algorithms and protocols in high-level simulations to provide immediate feedback to the developers.

The discussion in this paper motivates the need to develop a fast simulation infrastructure that reflects the power behaviour of algorithms earlier in the design phase and evaluates the benefits of an optimization. We argue that there are two basic requirements for any such simulation infrastructure; (i) it shall enable automated calibration of simulation models with energy information, and (ii) it shall capture the influence of the code - the protocol stack and the OS in particular - and certain hardware effects on energy consumption. To the best of our knowledge, this is the first effort to explore the techniques of simulating power consumption of distributed algorithms and protocols in their entirety. We hope that our analysis will foster further research in this domain.

3 Problem Analysis

Determining the energy consumption of a distributed system requires i) a scalable evaluation architecture, and ii) accurate simulation models. However, satisfying both properties at the same time is obviously a challenging task: For instance, increasing the level of detail of a simulation model also increases its accuracy at the price of a decrease in scalability -- and vice versa. Hence, it is essential to carefully engineer an adequate trade-off between scalability and accuracy.

We believe that network researchers do not require the highly detailed but slow system models used in the hardware system domain. Instead, researchers are typically more interested in comparing different design trade-offs at a qualitative level. Yet, the abstract power models available in current network simulators cannot capture the influence of the executed code and hardware on energy consumption, as they merely count the number of packets transmitted and received. Hence, we argue that this gap between both domains needs to be filled by simulation models that enable network researchers to take design decisions with respect to certain hardware effects.

In the following, we analyze the components of a distributed system node in terms of their energy consumption. The goal of this analysis is to separate the system components which exhibit a significant impact on the energy consumption from those that exhibit just a minor impact. As a result, components with a small impact can be modelled at a higher level of abstraction without significantly adulterating the overall results whereas components with a large impact demand more detailed modelling.

3.1 Hardware

3.1.1 CPU

The CPU is a major contributor to the overall energy consumption of a computer system ranging from 2W of low power mobile CPUs (Intel Atom 1.1GHz) up to 150W of high-end desktop and server CPUs (Intel XEON 3.5GHz Dual Core). Furthermore, modern CPUs exhibit dynamic frequency and voltage scaling techniques which aim for reducing the energy consumption. Hence, it is essential to accurately model the CPU behaviour with respect to a certain platform.

3.1.2 Main Memory

As with the CPU, main memory is a primary computing resource and hence requires explicit modelling. In contrast to the CPU however, a typical memory module of today's desktop computers (4GB DDR3 1066MHz) consumes just about 1-2 W. Due to this low energy consumption, memory hardware usually does not employ dynamic energy saving techniques. As a result, energy models of the main memory may rely on abstract and relatively static modelling.

3.1.3 Disk

A typical hard disk of a computer system consumes 3-15W[2] during write operations, depending on its size, rotation speed and number of platters. This is just a fraction of the energy consumed by CPUs. Furthermore, kernel level protocols (i.e., layer 1 to 4) do not utilize the disk directly, but merely provide network data to the applications. Disk utilization across applications is diverse and ranges from very low (e.g., remote login) to heavy usage (e.g., file sharing). Altogether, modelling the energy consumption of hard disks is necessary, but can be modelled at a higher level of abstraction than the CPU.

3.1.4 Networking Hardware

The energy consumption of networking hardware heavily depends on the transmission medium and the transmission speed. Ethernet controllers for instance consume 270 mW at a link speed of 10MBit/s and up to 1.2W at 1GBit/s (Intel 82541). Since the link speed in wired networks usually remains constant, modelling such networking hardware can be achieved by highly abstract models. However, wireless networks are subject to frequently changing link qualities and hence power consumption due to interference and node mobility. Besides, it is essential to capture different power states of the networking hardware. Hence, wireless networking hardware demands accurate modelling.

3.1.5 Video and Audio

In modern desktop computers, the video hardware often is by far the primary consumer of energy. However, in contrast to the resources discussed so far, the video hardware is only of minor importance to network researchers because its energy consumption is seldom influenced by network related operations. As a result, even when considering networking applications such as video streaming, the video hardware can be modelled using a very high level of abstraction. A similar argumentation as for video hardware is true for audio hardware. Both, its energy consumption and its involvement in networking operations are minor. Typical networking applications such as VoIP-softphones usually do not directly utilize the audio hardware, but rather the CPU. Concluding, modelling the energy consumption of audio hardware is not required in network simulation.

3.2 Software

3.2.1 Network Stack

From the perspective of networking research, the network stack obviously constitutes the most important software component. As researchers want to determine the performance and energy consumption of newly developed protocols and algorithms in comparison to existing approaches, the network stack, i.e., the networking protocols, has to be modelled as close to a real implementation as possible. At first, this seems to contradict the central philosophy of network simulation which revolves around abstract models. However, we strongly believe that this is a imperative requirement for achieving accurate results.

3.2.2 Operating System

Even when explicitly excluding the network stack from the operating system in the context of this analysis, a significant amount of time and hence energy is still spent in the operating system [14]. In the context of networking, this is mainly caused by memory management operations that transfer data between user-space and kernel-space as well as NIC and kernel [13]. Furthermore, the operating system provides complex functionality such as process scheduling and device management. While this functionality cannot be accurately modelled in network simulation due to scalability constraints, it is however essential to capture their run-time requirements by means of abstract models.

4 Design Space Exploration

After analyzing the contribution of different components from a power perspective, we now discuss possible approaches to model these power hungry components in high level simulations.

4.1 CPU

The CPU is commonly used as a first-order proxy in modelling the dynamic power consumption at runtime [15]. However, most of the existing approaches heavily rely on runtime metrics such as CPU-utilization and performance counters. The trivial solution to model CPU in simulations is to use expensive emulation, i.e., cycle- or instruction-accurate simulation, which is not feasible in network simulation. We believe that the simulation instrumentation [16], [17] based techniques provide sufficient accuracy and granularity to model the CPU. Our preliminary work in the sensor network domain, such as TimeTOSSIM [16], and efforts like PowerTOSSIM [17] have demonstrated the feasibility of this approach in principle. Moreover, the effort invested in such instrumentation of simulation models is an order of magnitude less than implementing an emulator. Currently, we are generalizing this approach to include simulation platforms from a wide variety of distributed systems. The underlying instrumentation technique is to i) determine the binary instructions corresponding to a source-code line by creating a mapping between simulation source-code and the platform specific executable, and ii) to determine the execution time of these instructions, and thus the CPU utilization, which can be converted into their corresponding power requirements.

The first step, i.e., code-mapping, is only possible when nearly identical code is executed in simulation and on the hardware platform. Thus, it requires a real-world implementation of an application. The second step, i.e., deter-

[2] Survey of Western Digital and Samsung hard drive specs.

mining the execution time and power consumption, is dependent upon the target platform.

4.1.1 Non-pipelined processors

Such processors employ sequential instruction execution without any pipelining and caching strategies, this is a straightforward process, i.e., the execution time of a source-code line can be calculated by summing up the execution cycles of corresponding binary instructions. Thereby, power consumption can directly be derived from these execution cycles [17]. This whole instrumentation process consists of three steps: (1) Parsing the application source code to identify each source code line, (2) instrumenting each line with corresponding execution time, and (3) building the simulation from the extended sources. We implemented a custom C grammar to programmatically change the simulation sources. Our pilot implementation, i.e., TimeTOSSIM, achieves beyond 99% time accuracy in sensor network simulation using a similar approach while outperforming emulation in terms of speed and scalability.

4.1.2 Pipelined processors

For such processors, we use the instructions-per-cycle (IPC) information to derive the power-consumption by using the strong correlation between IPC and power [14]. However, IPC is a runtime metric and cannot be determined by using a static process, such as the one used for non-pipelined CPUs. For this purpose, high-level pipeline models, such as Quick Piping [18], can be used to derive IPC metric. Nonetheless, our initial investigation revealed that the existing work mostly focuses on runtime pipeline modelling, and there is limited effort in providing such pipeline models for high level simulations.

4.1.3 Superscalar and multicore processors

The instrumentation process becomes even more complex for multicore platforms because power-consumption can no longer be determined by its power-states, e.g., active or sleeping. In such processors, it is more important to determine what the CPU is actually doing than to determine its power-state. Therefore, power consumption is not a linear function of CPU utilization for such processors. This is because power consumption in processors with shared resources strongly correlates to the processor components utilized for instruction execution. The applicability of instrumentation technique for such processors is an open research question.

4.2 The Operating System

It is imperative to model the power consumption of the OS because it constitutes a major portion of software that dissipates a significant amount of power. Our first step in modelling the power consumption of the OS is to determine its services that are dominant power consumers from a networking point of view. The idea is to model those OS services at different granularity depending on their importance in distributed applications and the variability in power consumption for each invocation. Similarly, it might also be useful to neglect barely used and less power hungry OS services to achieve simulation speedups. On the basis of our initial hypothesis, we divide the OS services into two categories:

4.2.1 The Protocol Stack

The most important OS service that is predominantly used by network applications, as discussed in section 3.2, is the protocol stack. To that end, we propose to import real-world implementations of protocol stacks into network simulations by using tools like Network Simulation Cradle [19] and OppBSD [20]. Such imports enable instrumentation at source-code line level, and thus, the protocol stack can be modelled at the same granularity as the application itself.

4.2.2 OS Routines

Network simulations based on the discrete event paradigm abstract from OS level details. Therefore, OS services other than the protocol stack can either be filtered out as a fixed overhead or OS-routine level profiling can be used to estimate the power consumption. OS-routine level models, such as the one presented by Li et. al. [14], achieve per-routine power estimation errors of less than 6%. Similar modelling techniques can be developed for network simulations that rely on few simple parameters. For example, such parameters could include the number of invocations of each OS routine during simulation. Nonetheless, a detailed evaluation of the impact of non-protocol OS services on the power consumption of network applications is imperative before taking any further design considerations into account.

4.3 Memory

Just like other system components, existing memory models for power consumption are either highly detailed [21] or they rely on runtime metrics. Hence, a significant effort is required to fill this gap between low-level hardware based power models for memory and high-level network simulations. Nonetheless, modelling the memory hierarchy is a design decision dependent on the desired accuracy in network simulation. In a simple system with one-stage memory, such as sensor nodes, power consumption of memory can accurately be derived from CPU utilization. Even in high end server systems with multiple memory hierarchies, CPU utilization based memory models predict power within 10% mean error [15]. This error is due to factors like cache misses and memory swapping which influence the power consumption on each memory access performed by the CPU. We believe that this error is acceptable in high level network simulation provided that CPU utilization is accurately modelled to serve as a proxy for memory power consumption.

4.4 Disk

Considering the fact that network protocols usually do not perform disk related operations, an abstract disk modelling approach is more suitable in network simulation. For example, such approach could include tracking the number of read and write operations and deriving power consumption by using a static power model. We did not find any high level power models for disk utilization during our initial investigation. However, developing an abstract power modelling technique for disks is a relatively simple task compared to modelling a multistage memory hierarchy.

4.5 Network Hardware

Among other devices, the networking hardware apparently consumes considerable energy, especially in battery driven mobile devices and sensor nodes. We identify three main requirements in network simulations to predict the power consumption of network hardware. First, it shall model the energy-state(s) of the hardware device. For example, in the case of a radio chip, this includes transmitting, receiving, power-down, and idle as energy-states. Second, it shall determine the time a device spends in each of its energy-states during simulation. Higher time resolutions would definitely result in more accurate energy predictions. Third, it shall accurately model the amount energy consumed over time by a device in each of its states. The energy-state(s) transitions are usually triggered in the software and can be captured in simulation. The duration a device spends in each of these states can easily be derived from the detailed timing model presented in section 4.1. Figure 1 summarizes the discussion in this section by presenting system components and their proposed modelling techniques.

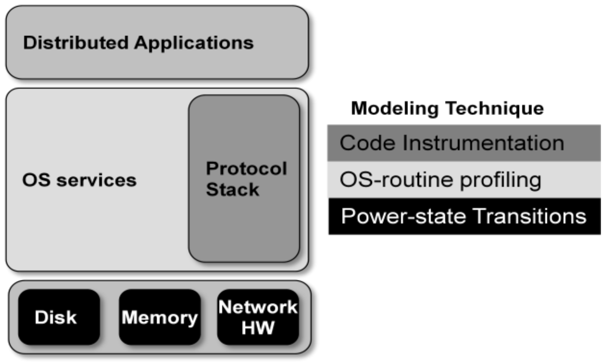

Figure 1: System Components and their corresponding modeling technique

5 Use Cases

This section presents selected use cases that i) underline the need for conducting thorough power consumption predictions early in the development process of distributed systems to facilitate design trade-offs and ii) illustrate the importance of considering the interaction of network nodes by means of network simulation.

5.1 Design Trade-offs

Promoting energy consumption to a first class metric adds a further dimension to the design process. In this section, we exemplary discuss typical design trade-offs that weigh the complexity of data coding against its energy efficiency.

5.1.1 Physical and Link Layer

Turbo codes [22] aim to increase the reliability of wireless transmissions on the physical layer. They allow for correcting transmission errors by applying iterative coding and decoding schemes to the transmitted data. Incrementing the number of iterations increases both the effectiveness and the energy consumption of turbo codes. Hence, the design trade-off lies in finding an appropriate number of iterations. In addition, network coding [23] denotes an algorithmic approach that mixes data at intermediate network nodes while allowing receivers to deduce the original data. By merging multiple messages into one, the overall traffic volume of a network decreases at the cost of coding effort. As with turbo codes, the trade-off lies in balancing the coding effort with the reduction in network traffic.

5.1.2 Network Layer

The security oriented protocols IPSec and HIP reside on or next to the network layer and make use of complex cryptography. Since security is a highly sensitive issue, there is usually no real trade-off between the complexity and strength of a cryptographic algorithm and its energy consumption. However, accurate energy predictions help to estimate the impact of cryptography particularly on low powered devices such as sensor nodes and mobile devices.

5.2 Network Interaction

Due to the interaction of network nodes, distributed systems exhibit a highly dynamic behavior which directly influences their energy consumption. However, existing tools for predicting the energy consumption focus primarily on individual and isolated systems. The following use cases illustrate the importance of combining network simulation with accurate and calibrated energy models.

5.2.1 Home Computers

Home computers are often powered on just to maintain reachability (e.g., instant messengers, VoIP-softphones) or provide peer-to-peer services (e.g., file-sharing, anonymization). Developing new energy efficient protocol extensions [8] and paradigms such as service delegation [24] is a crucial step towards greening the Internet. In order to accurately study their impact on energy consumption as well as their algorithmic correctness, realistic (i.e., large scale) network models in conjunction with precise energy models are required.

5.2.2 Network Core

A significant amount of energy is consumed by devices located in the network core such as repeaters, switches and routers. While such devices typically run 24/7, they are subject to highly fluctuating utilization loads resulting in diverse energy consumption patterns. Isolated power estimations however often do not consider such utilization patterns. In contrast, network simulation naturally models utilization patterns based on user behaviour and traffic models [25].

5.2.3 Sensor Networks and Mobile Devices

The wireless communication of mobile devices is heavily influenced by external factors such as interfering nodes or node mobility. Interference for example affects the energy consumption by inducing retransmissions of packets. Furthermore, node mobility causes frequent changes in transmission power due to varying distances to other nodes. Network simulation forms the substrate for accurately modelling those effects [26] which are otherwise non-trivial to integrate with today's energy profiling tools.

6 Conclusion

This paper underlines the need of promoting energy to a first class metric in network simulations. As pointed out earlier, existing research focuses either on runtime modelling or very detailed power models based on micro-architecture level knowledge of a system. This gap between high-level simulation and detailed power modelling techniques is a lot bigger than we initially expected when we started working on modelling power in network simulations. Nonetheless, there is none denying the fact that this gap has to be filled up if we want power consumption to become a major design metric in communication systems. The CPU is the most important component from a power perspective. It is very well suited to act as a proxy for modelling the power consumption of other system components as well. Concluding, this paper intends to spread a “call for help” to fill this void in network simulations and promote power to a first class metric.

7 Literature

[1] V. Tiwari, S. Malik, A. Wolfe, and M. T.-C. Lee, “Instruction level poweranalysis and optimization of software”, In VLSID ’96, 1996, Washington, USA.

[2] A. Sinha, A. Wang, and A. Chandrakasan, “Algorithmic transforms for efficient energy scalable computation,” 2000. Available: citeseer.nj.nec.com/sinha00algorithmic.html

[3] C. Isci and M. Martonosi, “Runtime power monitoring in high-end processors: Methodology and empirical data,” In MICRO 36, 2003, Washington, USA.

[4] R. Joseph and M. Martonosi, “Run-time power estimation in high performance microprocessors,” In ISLPED ’01: New York, NY, USA, 2001.

[5] C. Hu, D. A. Jimenez, and U. Kremer, “Toward an evaluation infrastructure for power and energy optimizations”, In IPDPS ’05, Washington, DC, USA, 2005.

[6] N. H. Zamora, J.-C. Kao, and R. Marculescu, “Distributed powermanagement techniques for wireless network video systems” In DATE ’07, San Jose, CA, USA, 2007.

[7] C. Gunaratne, K. Christensen, and B. Nordman, “Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed,” Int. J. Netw. Manag., vol. 15, no. 5, 2005.

[8] P. Eronen, “TCP Wake-Up: Reducing Keep-Alive Traffic in Mobile IPv4 and IPSec NAT Traversal,” Nokia, Tech. Rep., 2008.

[9] L. Irish, , L. Irish, and K. J. Christensen, “A ”green tcp/ip” to reduce electricity consumed by computers,” in Proceedings of IEEE Southeastcon, 1998.

[10] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A framework for architectural-level power analysis and optimizations,” in In Proceedings of the 27th Annual International Symposium on Computer Architecture, 2000.

[11] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye, “Energy-driven integrated hardware-software optimizations using simplepower,” SIGARCH Comput. Archit. News, vol. 28, no. 2, 2000.

[12] S. Gurumurthi, A. Sivasubramaniam, M. J. Irwin, N. Vijaykrishnan, M. Kandemir, T. Li, and L. K. John, “Using Complete Machine Simulation for Software Power Estimation: The SoftWatt Approach” In HPCA’02, Washington, DC, USA, 2002.

[13] B. Wang and S. Sinsh, “Computational Energy Cost of TCP,” in Proc. of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, March 2004, pp. 785–795.

[14] T. Li and L. K. John, “Run-time modeling and estimation of operating system power consumption,” SIGMETRICS Performance Evaluation Review, vol. 31, no. 1, 2003.

[15] S. Rivoire, P. Ranganathan, and C. Kozyrakis, “A comparison of highlevel full-system power models.” In HotPower, 2008.

[16] O. Landsiedel, H. Alizai, and K. Wehrle, “When Timing Matters: Enabling Time Accurate and Scalable Simulation of Sensor Network Applications“, In IPSN 08, 2008.

[17] V. Shnayder, M. Hempstead, B. rong Chen, G. W. Allen, and M. Welsh, “Simulating the Power Consumption of large-scale Sensor Network Applications” In SenSys’04, 2004.

[18] C. W. Milner and J. W. Davidson, “Quick piping: a fast, high-level model for describing processor pipelines,” SIGPLAN Not., vol. 37, no. 7, 2002.

[19] S. Jansen and A. McGregor, “Simulation with Real World Network Stacks”, In WSC, 2005.

[20] R. Bless and M. Doll, “Integration of the FreeBSD TCP/IP-stack into the discrete event simulator OMNet++”, In WSC, 2004.

[21] E. Schmidt, G. von C¨olln, L. Kruse, F. Theeuwen, and W. Nebel, “Memory power models for multilevel power estimation and optimization,” IEEE Trans. Very Large Scale Integr. Syst., vol. 10, no. 2, 2002.

[22] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1,” IEEE International Conference on Communications, 1993.

[23] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network Information Flow,” IEEE Transactions on Information Theory, vol. 46, no. 4, 2000.

[24] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta, “Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage”, In NSDI’09, 2009.

[25] B. A. Mah, “An Empirical Model of HTTP Network Traffic,” in Proc. of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies, 1997.

[26] T. Camp, J. Boleng, and V. Davies, “A Survey of Mobility Models for Ad Hoc Network Research,” Wireless Communications and Mobile Computing, vol. 2, 2002.

[27] Energy Framework: http://www.sics.se/~lmfeeney/software/energyframework.html

[28] L. M. Feeney, “An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks”, In Mobile Networks and Applications, Vol. 6, 2001.

[29] Margi, Cintia B. and Obraczka, Katia, “Instrumenting Network Simulators for Evaluating Energy Consumption in Power-Aware Ad-Hoc Network Protocols”, In MASCOTS 04, 2004.

[30] Daniel Weber, Johann Glaser and Stefan Mahlknecht, “Discrete Event Simulation Framework for Power Aware Wireless Sensor Networks”, In INDIN, 2007.

[31] Feng Chen, Isabel Dietrich, Reinhard German and Falko Dressler, , "An Energy Model for Simulation Studies of Wireless Sensor Networks using OMNeT++", Praxis der Informationsverarbeitung und Kommunikation (PIK), vol. 32 (2).