Low power light-weight embedded systems*

Majid Sarrafzadeh[1], Foad Dabiri[1], Roozbeh Jafari[2], Tammara Massey[1], Ani Nahapetian[1]

Computer Science Department[1]
University of California, Los Angeles

{majid, dabiri, tmassey, ani}@cs.ucla.edu

Electrical Eng. Dept. / University of Texas at Dallas[2]
Electrical Eng. and Computer Sci. Dept. / UC Berkeley

rjafari@utdallas.edu

Abstract

Light-weight embedded systems are now gaining more popularity due to the recent technological advances in fabrication that have resulted in more powerful tiny processors with greater communication capabilities that pose various scientific challenges for researchers. Perhaps the most significant challenge is the **energy consumption concern and reliability**, mainly due to the small size of batteries. In this tutorial, we portray a brief description of low-power, light-weight embedded systems, depict several power profiling studies previously conducted, and present several research challenges that require low-power consumption in embedded systems. For each challenge, we highlight how low-power designs may enhance the overall performance of the system. Finally, we present a several techniques that minimize the power consumption in such systems.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; C.3 [**Special-purpose and Application-based Systems**]: Real-time and embedded systems

General Terms

Design, Performance

Keywords

Light-weight embedded systems, sensor networks, power optimization

1.   Introduction

Light-weight embedded systems, recently introduced due to the advancement of fabrication of powerful tiny processors, have the ability to revolutionize capture, processing and actuation in several collaborative and networked systems. The new class of tiny embedded systems has been widely utilized in several domains from medical monitoring applications [1][2] to collaborative object tracking systems [3]. Many systems similar to the aforementioned applications require low-profile, mobile and cost-effective devices. The physical size and the cost-effectiveness immediately deduce several constraints in processing power and communication bandwidth. In addition, it enforces restriction on the size of batteries. These unique limitations require rethinking and reinventing the design process in particularly light-weight embedded systems. Predominantly,

* Second to fifth author names appear in alphabetical order.

power issues become a major concern in the design phase due to the unique properties of such systems. Optimization of power consumption in light-weight embedded systems is no longer just an objective function that is to be minimized. Power optimization is a tight constraint that must be accommodated to deliver a practical system.

2.   System definition

An embedded system is a special-purpose system in which a computer is entirely encapsulated by the gadget it controls. Unlike a general-purpose computer, an embedded system performs pre-defined tasks, usually with very specific requirements and constraints [4]. Since the system is dedicated to a specific task, designers can optimize it, reducing the size and cost of the product. *Light-weight embedded systems* are often referred to low-profile, small size, unobtrusive and portable processing elements with **limited power resources**. Such systems typically incorporate sensing, processing and communications and are often manufactured to be simple and cost-effective. These **low profile** systems usually have limited computational capabilities, memory (storage), speed and I/O interfaces. Despite their low complexity, computationally intensive tasks impede light-weight embedded systems from being deployed in collaborative networks in large quantities. While their sensing capabilities allow for a seamless integration into the physical world, their processor architecture designs yield notable advantages such as reconfigurability and adaptability with various applications and environments. Consumers, on the other hand, constantly demand thinner, smaller and lighter systems with **smaller batteries** in which the battery life is enhanced to meet their lifestyle. Improving the performance of battery life, however, has been always a major scientific challenge for researchers. Due to its criticality, the battery life becomes an objective as opposed to being a constraint in traditional systems. In order to optimize the power consumption in such systems, researchers must understand the major sources of power consumption. Therefore, we present a power profiling study on light-weight embedded systems in the next section. In section 4, we portray a number of challenges with respect to power optimization in such systems. In sections 5 and 6, we discuss two problems that we selected in regards to low-power light-weight embedded system design in details. Finally, Section 7 concludes the paper.

3.   Power profiling

In this section, we present a few widely used embedded systems deployed in several monitoring and mobile applications. Motes from CrossBow [5] are among popular candidates due to their tiny size and on-chip sensing. For applications where interaction with users is desired, Pocket PCs have also been broadly deployed [6]. Pocket PCs, in addition to their portability, incorporate relatively large displays, and communicate using commonly accepted

wireless communication protocols such as Bluetooth and 802.11 Wi-Fi. These communication capabilities facilitate real-time and remote monitoring. Several power profiling studies have been conducted on motes [7] [8]. Pocket PCs behave differently mainly to due to their architecture dissimilarities with motes. Several studies are conducted on power profiling of handheld devices [9] [10] including Pocket PCs.

4. Challenges

In this chapter, we describe several challenges for low-power embedded system design. For each challenge, we begin with a general definition, followed by low-power solutions proposed for traditional systems and then we depict new techniques suitable for light-weight embedded systems suggested by researchers.

4.1 Scheduling for power management

Task scheduling on single or multiple processing elements is considered as one of the most common methods to achieve lower power consumption. In particular, in light-weight embedded systems, scheduling saves power by shutting down devices when they are not operating. Processing elements in embedded systems usually serve different requests at different times. Ordering task execution adjusts the lengths of idle periods and exploits the opportunities for power management [11][12][13]. Several approaches have been proposed for task scheduling on low-power embedded systems that consider highly constrained energy source and environmental sources [14][15][16][17].

4.2 Software power optimization

Software constitutes a major component of today's systems, and its role is projected to continue to grow. In traditional processors, instruction level analysis of a processor aids in developing power consumption models of software execution in processors. Software power evaluation also gives the designers the ability to optimize their programs in terms of power. Common techniques include code compression and coding [18][19][20]. Similar approaches have been applied to embedded systems with tiny processors. Light-weight embedded systems are highly constrained in terms of the memory size available to them. Most work on code compression thus far has focused mainly on the memory optimization. However, code compression has a significant effect on energy consumption. Since a compressed code is smaller in size, fewer accesses to the main memory is required resulting in less energy consumption. Meanwhile, reductions in memory accesses result in the reduction of power dissipation in the bus and interconnections [21][22]. This, in particular, gains more importance in systems with small batteries.

4.3 Low power communication

Communication on buses within a chip as well inter-device wireless communication has been always a major source for power consumption. On-chip bus communication becomes even more challenging in tiny-processors that does not feature advance architecture. A considerable amount of energy is consumed in on-chip interconnect and I/O buses. The main source of power dissipation occurs when the voltage swings in communication lines. Bus coding and encoding techniques can be used to reduce power consumption along with increasing the performance in terms of throughput and latency. These techniques reduce voltage swings along interconnect lines which can result in large power savings [23][24][25][26]. Wireless communication in light-weight embedded systems dominates the total power consumption [7]. Wireless communication distinguishes these systems from other

traditional networks mainly due to the large of number of elements deployed, their power constraints, and their level of mobility. Several techniques have been proposed and various survey articles have been published in this area [27][28][29].

4.4 Low power security

Security protocols mostly involve complex computations and extensive communications. This challenge becomes even more critical in light-weight embedded systems mainly due to the constraints in processing power as well as communication bandwidth. In addition, the limitations in the battery size impede conducting complex tasks that require massive communications. Yet, due to the sensitivity of data communicated in many applications as well as the deployment of such applications in unattended and hostile environments, implementing quality, light-weight security protocols in embedded systems are imperative. Implementation of traditional security protocols in light-weight embedded systems, however, introduces several obstacles due to the resource constrained nature of such systems. More precisely, the constraints include the lack of large storage, the lack of powerful computational elements, a low bandwidth communication infrastructure, and limited power resources [30]. Therefore, various power-aware secure protocols have been proposed. Secure routing schemes have been investigated in [31][32][33] and [34]. Secure protocols also have been studied for data aggregation [35][36][37][38][39][40] and group formation [41][42][43].

4.5 Low power display

Classes of light-weight embedded systems in which interaction with users is essential require displays. The backlight in displays consumes a significant portion of available energy [44]. This becomes even more crucial in light-weight embedded systems where their available energy is highly constrained. Unfortunately, this issue has received little attention from researchers. [45] and [46] have proposed techniques for power reduction in displays. Techniques for low-power graphical user interface (GUI) and low-power human-computer interaction were suggested in [47] and [48].

4.6 Low power data management

Power optimization in data management has not been a major focus of researchers conventionally due to the availability of often unlimited energy source in most computation nodes. Yet, limited supply of energy, inadequate processing power, small memory size, power hungry communication and a low bandwidth communication infrastructure impose various challenges for data management in systems which are closely linked with physical environments. In addition, uncertainty in sensor readings due to environmental interference and faults in inexpensive embedded systems may result in the generation of inaccurate information. Several tree-based [49][50][51] and multi-path-based [52][53] query aggregation techniques have been proposed. In [54] researchers have addressed a combined method for in-network data processing. Overall, researchers have outlined several data management metrics in low-power distributed embedded systems which may enhance in-network data processing capabilities.

4.7 Fault tolerance and reliability

The most common approach to fault tolerance is the use of redundancy in systems. Adding a new redundant component creates a new source of power consumption in the system. Therefore, in light-weight embedded systems, efficient fault

tolerant techniques must be applied since the luxury of having redundant components can not be easily afforded. The other barrier is low bandwidth and expensive communication. Fault tolerance cannot be handled in a localized manner since it may not be feasible to collect information from all nodes. Finally, one major objective in manufacturing tiny embedded systems is the fabrication cost. The manufacturers tend to keep the systems cost-effective; therefore, the hardware may not be ultra reliable and reliability concerns mostly must be handled by the applications [55][56][57][58].

5. Minimum skew utilization

Previous research efforts have shown that communication dissipates significant amount of energy [7]. Consequently, design choices and protocols that affect the communication traffic have a great impact on system lifetime. For example, minimum-hop routing algorithms reduce the number of transmissions required to deliver a message at the destination and improve system energy dissipation [59]. In this section, we propose a technique called minimum skew utilization that aims at optimizing the power consumption and enhancing the system lifetime. This is done by evenly distributing node utilization and communication across the network. In our formulation, we attempt to minimize the skew in energy consumption (due to wireless communication) across highly congested nodes. We will show that our formulation will additionally yield a minimum skew distribution of energy consumption across all the nodes. The skew is defined as follows: There exists exponential number of paths connecting source to destination nodes. For every path, we form the following definition. There exists a node in every path that has the highest energy consumption rate (ideally if each path would be isolated from the rest of the network, the energy consumption rate due to wireless communication must be identical throughout the path, however, in reality, nodes may have incoming/outgoing edges from/to the rest of the network. Therefore, the energy consumption rate may not necessarily be uniformly distributed). In every path P_i connecting a source to a destination node, we identify the node with highest wireless communication traffic as p_{imax}. We define the skew of energy consumption as the difference between p_{imax} and p_{jmax} of paths i and j. In the next subsection, we will illustrate how our formulation minimizes the difference of highest traffic across every two paths. Even though, the number of paths is exponential, the upper bound on the number of links with highest traffic is still m (where m is total number of links). The definition of minimum skew is rephrased in Equation 1.

For every Pi and Pi connecting source(s) to destination(s)
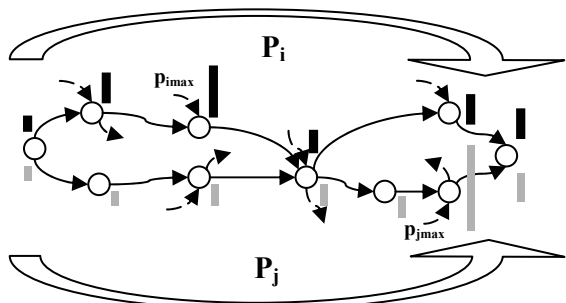$$\text{Minimize} \mid p_{imax} - p_{jmax} \mid \qquad (1)$$

Figure 1. Min-skew definition

5.1 Methodology

Given a graph G_s for a network and a specific traffic pattern, the objective is to route the packets so that after the completion of packet routing, the maximum traffic across all nodes is minimized. This objective enhances the system lifetime. We assume that there is a specific node t in G_s serves as a gateway, base station, or destination node, and all packets have to be delivered to node t. Each packet transmitted from a source node to t can be viewed as a unit flow in the network G_s. More precisely, x_{ij} units of flow represents x_{ij} packets transmitted from v_i to v_j. The problem of packet routing is equivalent to finding a feasible network flow in G_s. Intuitively, the objective of minimizing the highest congested node is equivalent to minimizing the maximum flow passing through nodes. In the remainder of this section, we transform G_s to a new graph G_t in which, a formulation based on min-cost flow generates the optimal solution. We proceed to describe the transformation procedure, along with the mathematical properties of our technique.
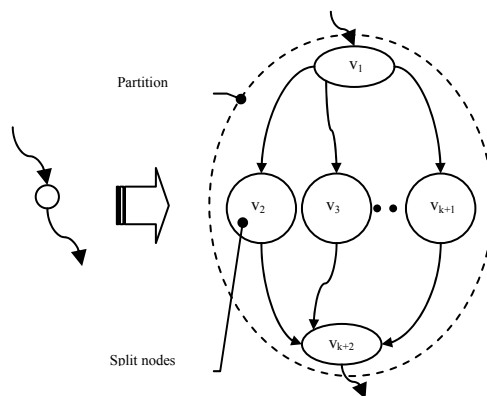

Figure 2. Node partitioning

We construct the network $G_t = (V_t, E_t)$ from the graph G_s according to the following rules. Each node in graph G_s is split into $k + 2$ nodes where k is a tunable parameter that controls the accuracy of the solution. We refer to the resulting set of nodes as a partition. In each partition, two nodes serve as receiver and transmitter (or input and output) and the rest of the nodes are called splits. Figure 2 illustrates an example partition in which, nodes v_1 and v_{k+2} are the receiver and the transmitter of the partition, respectively. Nodes v_2 thru v_{k+1} are the split nodes. The capacity of each node, v_i, is represented by u_i. In order to embed this constraints into the problem, we assign the flow upper bound of $u'_i = u_i/k$ to each of the split nodes. The upper bounds guarantee that a partition cannot pass more flow after relaying u_i units of flow, or equivalently, a node cannot exceed its capacity. The receiver, transmitter, and split nodes are assigned a special sequence of costs for passing the flow. The cost associated with receiver and transmitter nodes of a partition is zero. The costs associated with split nodes of a partition increase from left to right (Figure 2). This directs the min-cost flow technique to utilize the split nodes from left to right, when passing flow through a partition.

The cost on each split node is enforced such that it would be greater than the cumulative cost of the split nodes with smaller indices over all partitions. Intuitively, our cost assignment technique enforces the min-cost flow technique to utilize the split nodes with smaller indices, before trying to utilize a particular

split node. This simple, yet effective idea exhibits the main property of our technique by which, we minimize the traffic across the highly congested links. For simplicity, we define cost rank where cost rank R_i has the following property:

$$R_i > \sum_{j<i} m.R_j \qquad (2)$$

The cost on all the other nodes from the original graph G_s is zero. The loss in edges may also be accommodated easily. For the sake of brevity, we remove the extensions for lossy edges.

We now prove that our cost assignment strategy implies that the min-cost flow technique in G_t corresponds to a routing scheme in G_s that minimizes the traffic across the most highly congested link. Let x_{ij} represent the amount of flow on edge (i, j). Let y_{il} represent the amount of flow going through split node l in partition i. Similarly, let c_{il} denote the associated cost of unit flow passing through split l of node i in G_t. The min-cost flow problem for graph G_t with the given supply and demand vector, can be written as:

$$\text{Minimize} \sum_{\forall i \forall l=1..k} c_{il} y_{il} \qquad (3)$$

The flow that passes through the receiver or transmitter nodes of a partition represents the total flow passing through that partition, or equivalently, it determines the traffic across the original link in G_s. Therefore, the objective is to minimize the maximum amount of flow passing through partitions.

The flow passing through any partition has to pass through its split nodes, and subsequently, min-cost flow solutions utilize the splits with lower costs before higher cost splits. We present three theorems; however, the proofs are omitted for brevity.

Theorem 1: The objective function in Equation 3 minimizes the maximum flow across the links of the graph $G_s = (V_s, E_s)$ with maximum error ε where $\varepsilon <$

$u'_i = u_i/k = 1/k$.

The intuition behind our proposed technique is that the cost assignment on the splits forces the network to route a flow from the lth split of link e_i, if it cannot be routed through any number of other nodes whose $(l-1)^{th}$ splits is empty.

Theorem 2: The solution L, generated by our technique, minimizes the difference of maximum flows across every two disjoint paths connecting a source to a destination node (with tolerance of $\varepsilon = 1/k$ - minimal-skew). In other words, it minimized the difference of traffic across the most highly congested links in the two paths.

Theorem 3: The lexicographically sorted solution of minimal-skew routing is unique.

5.2   Summary and open problems
We generated several benchmarks based on random geometric graphs with relatively a large number of nodes to illustrate the effectiveness of our technique. For simplicity, we assumed that communication capacity across all links is uniform. One hundred sensors are placed within areas of size 160×40, 160×60, 160×80, 200×40, 200×60, 200×80 and 200×100. In all networks, three source nodes are placed on far left side of the square area while the destination nodes/ gateways (three) are placed on the right side of the square. This particularly assists us to place the

source/destination nodes not within the close proximity of each other. The connectivity between nodes is determined by unit-radius disk model. The reception rate on communication links is chosen randomly between 80% and 100%. The locations of the nodes are generated conforming to a random uniform distribution over several size areas. For every particular area size, twenty benchmarks are generated with three source and three destination nodes. To compare our scheme against other routing algorithms, we consider a shortest path routing algorithm based on minimum cost flow (which is equivalent to the case where we have only one split ($k=1$)).

Overall, the average maximum communication traffic is reduced by a factor of 4 for $k=4$ compared to min-cost shortest path approach ($k=1$). The average delay of our scheme (for k =4) is about 20% greater than the min-cost shortest path. The average delay is not vastly increased due to existence of multiple edge disjoint paths between sources and destinations. The average delay also increases slightly as the k (number of splits) is increased. In general, highly connected networks provide a large number of parallel paths between nodes, which is of our interest and enhance the flexibility of data routing.

For the minimum skew utilization, a number of open problems must be solved to allow truly deployment of the technique in distributed and light-weight embedded systems. Firstly, the distributed version of this technique must be explored. Secondly, in highly dynamic networks where the quality of links may lively change, a fast optimal or sub-optimal solution is desired. Finally, the effect of several cost series on split nodes may be studied.

6.   Static Voltage Scheduling
In this section we address the problem of static voltage scheduling in light-weight, low-power and high-performance systems. The scheduling problem refers to the assignment of a supply voltage level to each module in the architecture representing the system. The objective is to minimize the energy consumption for a given computation time or throughput constraints or both. Intuitively, the voltage scheduling problem can be stated as a timing management problem. In a given application with distinct constituting blocks, the maximum tolerable power reduction of individual blocks is desired, while the timing constraints of the system is not violated. These blocks are often modeled as nodes in a directed acyclic graph (DAG) where edges represent the dependency among modules.

6.1   Methodology
We present an optimal methodology for static voltage scheduling for low-power and high-performance systems. The main property of the methodology is the unified formulation with linear size number of constraints in the optimization problem as opposed to exponential number of constraints in previously proposed techniques. Our formulation can be applied to dynamic voltage scaling on single or multiple resources. Moreover, this problem formulation results a convex optimization problem.

An interesting observation is proven indicating that in the optimal voltage scheduling of a DAG, the delay between any node u and the output t is independent of the choice of the path taken between them and is unique. This property of the optimal solution leads us into generating only linear number of constraints. We prove this property in the following theorem.
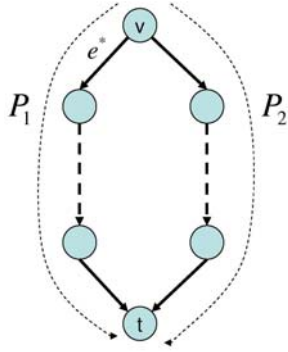
Figure 3. Figure for theorem 4

Theorem 4: In the optimal voltage scheduling of a DAG, the distance between any node v and the output t is independent of the choice of the path taken between them and is unique.

Proof: Suppose the claim is not true, i.e. there exists a node v where its distance to t through path P_1 is less then P_2 (see Figure 3). The intuition behind the proof is that, the larger delay assigned to an edge yields in more power saving which is the objective of the optimization problem. If P_1 is shorter than P_2, there exists an edge ($e*$) in P_1 that can be slowed down and still not violate the timing constraint because P_2 is on the critical path from v to t. One immediate candidate for $e*$ is the first edge in P_1. Increasing the delay of $e*$ by $d_{P2} - d_{P1}$ will not cause a timing violation and therefore, reduces the total power dissipation.

The following observation is immediately inferred from the above theorem: The delay of each path in the optimal solution from the primary input node s to primary output node t is equal to T. Now that the distance of every node to the destination is independent of the path taken, let t_i be a variable assigned to each node v_i that represents its distance to t. We call t_i the distance variable of node v_i. In other words, t_i is the delay of the system from node v_i to the output. Therefore, the delay of each node in the graph can be bounded by linear number of constraints.

In the next step, we show that the feasible solution space is in fact convex. With a convex objective and a convex feasible region, there can be only one optimal solution, which is globally optimal. All delay constraints are linear and can be viewed as planes, bounding the solution space. It is trivial that non-finite planes yield in a convex solution space. To show the convexity of the objective function which we prove that each term is convex. It can be easily shown that each term is proportional to the inverse of the delay squared which is trivially convex. Since the sum of convex functions is convex, the objective function is convex.

These voltages obtained from the optimal method might all have different values and therefore result in large number of power supplies with various voltage levels which may not be available. Current technologies, allow the designer to utilize only a few number of voltages. We propose to use the nearest neighbor mapping technique and map the optimal voltage of each node to its next available level. Experimental results show that this yields a very good power saving while the timing constraint is violated by a small fraction.

6.2 Summary and open problems

We evaluated the performance of our techniques on benchmarks from TGFF[60]. An average of 43.96% power reduction was gained for unbounded supply voltage assignment along with 40%

average power saving where discrete voltage levels are available. The results illustrate the efficiency of using the continuous optimal voltage scheduling for discrete case and observed that even limited number of voltage levels (less that 8) can provide us with near optimal power reduction.

In future, the optimum voltage scheduling method may be extended for dynamic voltage scheduling. Furthermore, developing design rules that assist developers with voltage scheduling at the design stage may be investigated. In addition, the effect of voltage level shifters on performance and their related optimization problems may be studied.

7. Conclusion

Power consumption and battery life is among the most critical concerns in light-weight embedded systems. Improving the performance of battery life, however, has been always a major scientific challenge for researchers. In order to optimize the power consumption of such systems, researchers must first understand the major sources of power consumption. In this paper, we presented power profiling studies conducted on several embedded systems. We furthermore portrayed a number of challenges that require low-power solutions and depicted three techniques that enhances the power lifetime.

8. References

[1] R. Jafari, A. Encarnacao, A. Zahoory, F. Dabiri, H. Noshadi, and M. Sarrafzadeh, "Wireless Sensor Networks for Health Monitoring," *MobiQuitous '05: Proc. Second Ann. Int'l Conf. Mobile and Ubiquitous Systems,* 2005.

[2] R. Jafari, F. Dabiri, P. Brisk, and M. Sarrafzadeh, "Adaptive and Fault Tolerant Medical Vest for Life-Critical Medical Monitoring," *SAC '05: Proc. 2005 ACM Symp. Applied Computing,* pp. 272-279, 2005.

[3] Aslam, J., Butler, Z., Constantin, F., Crespi, V., Cybenko, G., and Rus, D. 2003. Tracking a moving object with a binary sensor network. In *Proceedings of the 1st international Conference on Embedded Networked Sensor Systems* (Los Angeles, California, USA, November 05 - 07, 2003). SenSys '03. ACM Press, New York, NY, 150-161.

[4] Daniel Gajski, Frank Vahid, Sanjiv Narayan, and Jie Gong, *Specification and Design of Embedded Systems*, Prentice Hall, 1994.

[5] Crossbow technology inc. *http://www.xbow.com*.

[6] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. 2000. Sensing techniques for mobile interaction. *In Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology* (San Diego, California, United States, November 06 - 08, 2000). UIST '00. ACM Press, New York, NY, 91-100.

[7] Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 188-200, New York, NY, USA, 2004. ACM Press.

[8] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214-226, New York, NY, USA, 2004. ACM Press.

[9] Vijay Raghunathan, Trevor Pering, Roy Want, Alex Nguyen, and Peter Jensen. Experience with a low power wireless mobile computing platform. In *ISLPED '04: Proceedings of the 2004 international symposium on Low power electronics and design*, pages 363-368, New York, NY, USA, 2004. ACM Press.

[10] Marc A. Viredaz and Deborah A. Wallach. Power evaluation of a handheld computer. *IEEE Micro*, 23(1):66-74, 2003.

[11] Jing-Jang Hwang, Yuan-Chieh Chow, Frank D. Anger, and Chung-Yee Lee. Scheduling precedence graphs in systems with interprocessor communication times. *SIAM J. Comput.*, 18(2):244-257, 1989.

[12] Hesham El-Rewini and T. G. Lewis. Scheduling parallel program tasks onto arbitrary target machines. *J. Parallel Distrib. Comput.*, 9(2):138-153, 1990.

[13] Behrooz Shirazi, Mingfang Wang, and Girish Pathak. Analysis and evaluation of heuristic methods for static task scheduling. *J. Parallel Distrib. Comput.*, 10(3):222-2232, 1990.

[14] Yung-Hsiang Lu, Luca Benini, and Giovanni De Micheli. Low-power task scheduling for multiple devices. In *CODES '00: Proceedings of the eighth*

international workshop on Hardware/software codesign, pages 39-43, New York, NY, USA, 2000. ACM Press.

[15] Niraj K. Jha, Jiong Luo, "Battery-Aware Static Scheduling for Distributed Real-Time Embedded Systems," *Design Automation Conference* , pp. 444-449, 2001.

[16] Rakhmatov, D. and Vrudhula, S. 2003. Energy management for battery-powered embedded systems. *Trans. on Embedded Computing Sys.* 2, 3 (Aug. 2003), 277-324.

[17] Kansal, A., Potter, D., and Srivastava, M. B. 2004. Performance aware tasking for environmentally powered sensor networks. *In Proceedings of the Joint international Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, June 10 - 14, 2004). SIGMETRICS '04/Performance '04. ACM Press, New York, NY, 223-234.

[18] A. Peymandoust, T. Simunic, and G. de Micheli. Low power embedded software optimization using symbolic algebra. In *DATE '02: Proceedings of the conference on Design, automation and test in Europe*, page 1052, Washington, DC, USA, 2002. IEEE Computer Society.

[19] Eui-Young Chung, Luca Benini, and Giovanni De Micheli. Source code transformation based on software cost analysis. In *ISSS '01: Proceedings of the 14th international symposium on Systems synthesis*, pages 153-158, New York, NY, USA, 2001. ACM Press.

[20] Arpad Beszedes, Rudolf Ferenc, Tibor Gyimothy, Andre Dolenc, and Konsta Karsisto. Survey of code-size reduction methods. *ACM Comput. Surv.*, 35(3):223-267, 2003.

[21] Haris Lekatsas, Wayne Wolf, and Joerg Henkel. Arithmetic coding for low power embedded system design. In *DCC '00: Proceedings of the Conference on Data Compression*, page 430, Washington, DC, USA, 2000. IEEE Computer Society.

[22] A. Parikh, Soontae Kim, M. Kandemir, N. Vijaykrishnan, and M. J. Irwin. Instruction scheduling for low power. *J. VLSI Signal Process. Syst.*, 37(1):129-149, 2004.

[23] Mircea R. Stan and Wayne P. Burleson. Bus-invert coding for low-power i/o. *IEEE Trans. Very Large Scale Integr. Syst.*, 3(1):49-58, 1995.

[24] Benjamin Bishop and Anil Bahuman. A low-energy adaptive bus coding scheme. In *WVLSI '01: Proceedings of the IEEE Computer Society Workshop on VLSI 2001*, page 118, Washington, DC, USA, 2001. IEEE Computer Society.

[26] Peter Petrov and Alex Orailoglu. Low-power data memory communication for application-speci‾c embedded processors. In *ISSS '02: Proceedings of the 15th international symposium on System Synthesis*, pages 219-224, New York, NY, USA, 2002. ACM Press.

[25] Peter Petrov and Alex Orailoglu. Low-power instruction bus encoding for embedded processors. *IEEE Trans. Very Large Scale Integr. Syst.*, 12(8):812-826, 2004.

[27] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks, 2002.

[28] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 181-190, New York, NY, USA, 1998. ACM Press.

[29] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 2, 2004.

[30] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Seventh Annual International Conference on Mobile Computing and Networks (MobiCOM 2001)*, Rome, Italy, July 2001.

[31] J. Deng, R. Han, and S. Mishra. Insens: Intrusion-tolerant routing in wireless sensor networks, 2002.

[32] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243-254, 2000.

[33] P. Papadimitratos and Z.J. Haas. Secure routing for mobile ad hoc networks.

[34] Sapon Tanachaiwiwat, Pinalkumar Dave, Rohan Bhindwale, and Ahmed Helmy. Poster abstract secure locations: routing on trust and isolating compromised sensors in location-aware sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 324-325, New York, NY, USA, 2003. ACM Press.

[35] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Ku mar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263-270, 1999.

[36] L. Hu and D. Evans. Secure aggregation for wireless networks, 2003.

[37] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks, 2002.

[38] Bartosz Przydatek, Dawn Song, and Adrian Perrig. Sia: secure information aggregation in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 255-265, New York, NY, USA, 2003. ACM Press.

[39] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. Medians and beyond: new aggregation techniques for sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 239-249, New York, NY, USA, 2004. ACM Press.

[40] F. Ye, H. Luo, S. Lu, and L. Zhang. Statistical en-route detection and filtering of injected false data in sensor networks, 2004.

[41] Andreas Grlach, Andreas Heinemann, and Wesley W. Terpstra. Survey on location privacy in pervasive computing.

[42] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2-3):293-315, September 2003.

[43] Sandro Rafaeli and David Hutchison. A survey of key management for secure group communication. *ACM Comput. Surv.*, 35(3):309-329, 2003.

[44] J. Flinn, K. Farkas, and J. Anderson. Power and energy characterization of the itsy pocket computer, 2000.

[45] Inseok Choi, Hojun Shim, and Naehyuck Chang. Low-power color tft lcd display for hand-held embedded systems. In *ISLPED '02: Proceedings of the 2002 international symposium on Low power electronics and design*, pages 112-117, New York, NY, USA, 2002. ACM Press.

[46] S. Pasricha and S. Mohapatra. Reducing backlight power consumption for streaming video applications on mobile handheld devices, 2003.

[47] Lin Zhong and Niraj K. Jha. Graphical user interface energy characterization for handheld computers. In *CASES '03: Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems*, pages 232-242, New York, NY, USA, 2003. ACM Press.

[48] Lin Zhong and Niraj K. Jha. Energy efficiency of handheld computer interfaces: limits, characterization and practice. In *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 247-260, New York, NY, USA, 2005. ACM Press.

[49] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper.Syst. Rev.*, 36(SI):131-146, 2002.

[50] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122-173, 2005.

[51] Y. Yao and J. Gehrke. Query processing in sensor networks, 2003.

[52] Jeffrey Considine, Feifei Li, George Kollios, and John Byers. Approximate aggregation techniques for sensor databases. In *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*, page 449, Washington, DC, USA, 2004. IEEE Computer Society.

[53] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. Synopsis diffusion for robust aggregation in sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 250-262, New York, NY, USA, 2004. ACM Press.

[54] Amit Manjhi, Suman Nath, and Phillip B. Gibbons. Tributaries and deltas: efficient and robust aggregation in sensor network streams. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 287-298, New York, NY, USA, 2005. ACM Press.

[55] Thomas Clouqueur, Kewal K. Saluja, and Parameswaran Ramanathan. Fault tolerance in collaborative sensor networks for target detection. *IEEE Trans. Comput.*, 53(3):320-333, 2004.

[56] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli. Fault tolerance techniques in wireless ad-hoc sensor networks, 2002.

[57] Jonathan L. Bredin, Erik D. Demaine, MohammadTaghi Hajiaghayi, and Daniela Rus. Deploying sensor networks with guaranteed capacity and fault tolerance. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 309-319, New York, NY, USA, 2005. ACM Press.

[58] Murat Demirbas. *Scalable design of fault-tolerance for wireless sensor networks*. PhD thesis, The Ohio State University, 2004.

[59] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.

[60] D. Rhodes R. Dick and W. Wolf. Tgff: Task graphs for free. *In CODES '98*: Proceedings of the CODES, pages 97–101, 1998.