ENERGY-SCALABLE ALGORITHMS AND PROTOCOLS FOR WIRELESS MICROSENSOR NETWORKS

Wendi Rabiner Heinzelman, Amit Sinha, Alice Wang, and Anantha P. Chandrakasan

Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

Wireless microsensor networks lend themselves to trade-offs in energy and quality. In these networks, the individual sensor data per se is not necessarily important to the end user. Rather, it is the combined knowledge of all the sensors that describes what is occurring in the environment. By allowing the algorithms and protocols to adapt the quality of this description, with a corresponding change in energy dissipation, sensor networks can be flexible to the end-user's requirements. In this paper, we provide models for predicting quality and energy and show the advantages of trading off these two parameters. By ensuring that the system operates at a minimum energy for each quality point, the system can achieve both flexibility and energy efficiency, allowing the end-user to maximize system lifetime.

1. INTRODUCTION

Wireless microsensor networks consist of battery-operated sensor nodes that work together to achieve a desired goal. These systems enable the reliable monitoring of a variety of environments for applications that include home security, machine failure diagnosis, chemical/biological detection, medical monitoring, and a variety of military applications [1-4]. The goal of a sensor network is very different than that of a traditional wireless network. In a sensor network, it is not the individual nodes' data that is important, but the combined knowledge that reliably describes the environment the nodes are sensing. Therefore, the end-user only needs access to a function of the data, $\boldsymbol{f(X)}$, rather than all the data, $\boldsymbol{X}$. Data fusion and classification algorithms can thus be used to locally combine correlated data before transmission to the end-user.

Different algorithms and protocols for sensor networks can be evaluated using the following parameters:

- Quality: The quality of the result of a sensor network, $\boldsymbol{f(X)}$, can be measured against the ideal result, given that the source is known (e.g., using matched filter detection).
- Energy: Energy should be minimized at each node to maximize the system lifetime. All aspects of the sensor network, from the physical design to the DSP algorithms and network protocols, should be designed to minimize energy dissipation.

Algorithms and protocols for sensor networks should be scalable, in order to obtain different energy and quality operating points, as the relative importance of different resources and requirements might change over the system lifetime [5]. For example, when energy is plentiful, the network may be required to produce high quality results. As the energy gets depleted, the network may be required to reduce the quality of the results in order to reduce the energy dissipation in the nodes and hence lengthen the total system lifetime. In addition, sensor network protocols should be scalable to respond to events in the environment. Until an event occurs, most of the sensors can remain in the sleep state, with the data from the few remaining sensors providing a coarse quality. Once an event of interest is detected, the system should be able to configure itself so as to obtain very high quality results.

An energy-scalable system should be able to minimize the energy dissipation of the network for a given quality operating point. The scalable algorithms and protocols described in this paper are being implemented as part of the MIT μ-AMPS (micro-Adaptive Multi-domain Power-aware Sensors) system.

2. ENERGY MODELS

In order to predict the energy and quality of different algorithms and protocols, it is important to have accurate models for all aspects of the sensor node. We will specifically discuss models for computation and communication energy dissipation and use these models to simulate performance of our sensor networks.

Currently, there is a great deal of research in the area of low-energy radios. In our work, we assume a simple model where the radio dissipates E_{elec} = 50 nJ/bit in the transmit or receive circuitry and ε_{amp} = 100 pJ/bit/m^2 for the transmit amplifier to achieve an acceptable E_b/N_o (see Figure 1). We assume an r^2 energy loss due to channel transmission since we are looking at relatively short distance transmissions. Thus, to transmit a k-bit message a distance d, the radio expends:

$$\begin{aligned} E_{Tx}(k,d) &= E_{Tx\text{-}elec}(k) + E_{Tx\text{-}amp}(k,d) \\ &= E_{elec}*k + \varepsilon_{amp}*k*d^2 \end{aligned}$$

and to receive this message, the radio expends:

$$\begin{aligned} E_{Rx}(k) &= E_{Rx\text{-}elec}(k) \\ &= E_{elec}*k \end{aligned}$$

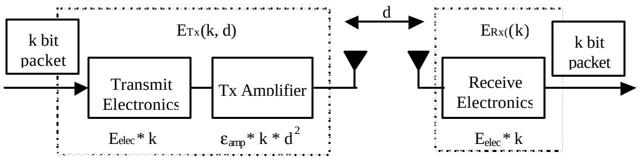

Figure 1. Simple radio energy model.

In addition to modeling the energy dissipation of the radio, it is important to accurately estimate the energy requirements of the computation (often done in software) that must be performed on the nodes. Sensors should be able to estimate the energy requirement of an application, make decisions about their processing ability based on user-input and sustainable battery life, and configure themselves to meet the required goals. The concept of "energy aware" software is integral to such systems. For example, based on the energy model for the application and the system lifetime requirements, the sensor node should be able to decide whether a particular application can be run. If not, the node might reduce its voltage using an embedded DC/DC converter and run the application at reduced throughput or run at the same throughput but with reduced accuracy. Both these configurations would reduce energy dissipation and increase the node's lifetime. These energy-accuracy-throughput tradeoffs necessitate robust energy models for software based on parameters such as operating frequency, voltage and target processor.

We have proposed the following parameterized model for software and verified this model on the StrongARM SA-1100 microprocessor:

$$E_{tot}(V_{dd}, f) = C_{tot}V_{dd}^2 + V_{dd}(I_0 e^{\frac{V_{dd}}{nV_T}})(\frac{N}{f}) \qquad \text{(Eq. 1)}$$

where C_{tot} is the total capacitance switched by the program and N is the number of cycles the program takes to execute. These parameters can be obtained from the energy consumption data for a particular supply voltage, V_{dd}, and frequency, f, combination. The model can then be used to predict energy consumption for different supply-throughput configurations in energy-constrained environments. The leakage current model is processor dependent; for the StrongARM SA-1100, the parameters I_0 and n are 1.196 mA and 21.26 respectively. Table 1 shows the performance of our model compared to actual energy data for several DSP routines that may be performed on the sensor nodes. The maximum error for the programs we tested was less than 5%.

Table 1. Software energy model performance.

DSP Routine	Meas. Energy (mJ)	Model Parameters			Error (%)
		C_{tot} (mF)	N ($x10^6$)	C_{tot}/N	
fft	53.89	28.46	43.67	0.65	1.24
dct	0.10	0.05	0.08	0.66	4.22
idct	0.13	0.06	0.10	0.66	2.59
fir	1.23	0.67	0.97	0.70	3.28
tdlms	21.29	12.13	17.10	0.71	1.91

3. SCALABLE DSP ALGORITHMS

Energy scalability at the algorithm level is highly desirable because a large range of both energy and quality can be achieved by varying algorithm parameters. However, the energy-quality characteristics of DSP algorithms may not be optimal due to data dependencies. We will show how algorithmic transformations can be used to improve the energy-quality curve for the specific case of beamforming algorithms.

Beamforming algorithms can be used to aggregate highly correlated data from multiple sensors into one representative signal. The advantage of beamforming is twofold. First, beamforming is used to enhance the desired signal while interference or uncorrelated sensor noise is reduced. This leads to an improvement in detection and classification of the target. Second, beamforming reduces redundant data through compression of multiple sensor data into one signal. Figure 2 shows a block diagram of a wireless network of M sensors utilizing beamforming for local data aggregation.

We have studied various beamforming algorithms that fall under the category of "blind beamforming'' [6]. These beamformers provide suitable weighting functions, $w_i(n)$, to satisfy a given optimality criterion, without knowledge of the sensor locations. In this paper, we will show energy scalability for the Least Mean Squares (LMS) beamforming algorithm. The LMS algorithm uses a minimum mean squared error criterion to determine the appropriate array weighting filters, and is highly suitable for power aware wireless sensor networks.

The first step to model the energy-quality (E-Q) scalability of a system is to identify the parameters of the algorithm that can be changed to achieve energy scalability. For example, in the LMS beamforming algorithm, we have identified four parameters that can be used to scale energy and quality: number of sensors, filter length, voltage supply and adaptation time [7]. The second step is to implement the algorithm such that the energy-quality relationship has two desirable traits: the quality on average is monotonically increasing as energy increases and the E-Q curve is concave down:

$$Q(E_1) \geq Q(E_2) \quad \text{if} \quad E_1 \geq E_2 \qquad \text{(Eq. 2)}$$

$$\frac{d^2Q(E)}{dE^2} \leq 0 \quad \text{for} \quad 0 \leq E \leq E_{\max} \qquad \text{(Eq. 3)}$$

where $Q(E)$ is an accurate model of the algorithm's average quality as a function of computational energy. These constraints lead to intelligent energy-scalable systems. An E-Q curve that is concave downward is highly desirable since close to maximal quality is achieved at lower energies. Conversely, a system that has a concave upwards E-Q curve can only guarantee high quality at the maximum amount of energy.

Algorithmic transformations can be used to improve the E-Q characteristics of a system. For example, we will show that the LMS beamforming algorithm can be transformed to obtain desirable E-Q curves, providing better energy-scalability. Figure 3 shows our testbed of 6 sensors for this example. We perform beamforming on the sensor data, measure the energy dissipated on the StrongARM SA-1100, calculate the matched filter output (quality), and determine the E-Q relationship as we vary the number of sensors in beamforming and as the source moves from location A to location B.

In Scenario 1, we perform beamforming without any knowledge of the source location in relation to the sensors. Beamforming is done in a pre-set order <*1,2,3,4,5,6*>. The parameter we use to scale energy is *k*, the number of sensors in beamforming. As *k* is increased from 1 to 6, there is a proportional increase in energy. As the source moves from location A to B, we take snapshots of the E-Q curve, shown in Figure 4. This curve shows that with a preset beamforming order, there can be vastly different E-Q curves depending on the source location. When the source is at location A, the beamforming quality is only close to maximum when *k=5,6*. Conversely, when the source is at location B, the beamforming quality is close to maximum when *k=2*. Therefore, since the E-Q curve is highly data dependent, desirable E-Q scalability cannot be guaranteed.

An intelligent alternative is to perform some initial pre-processing of the sensor data to determine the desired beamforming order for a given set of sensor data. Intuitively, we want to beamform the data from sensors that have higher sensor signal energy. We propose the *most significant first transform*, which can be applied to many algorithms to improve E-Q characteristics. To find the desired beamforming order, first the sensor signal energy is estimated from the sensor data. Then the sensor signal energies are sorted using a quicksort method. The quicksort output determines the desired beamforming order.

Figure 5 shows the E-Q relationship when an algorithmic transform is used. In this scenario, with the most significant first transform, we can ensure that the E-Q graph is concave down, thus improving the E-Q characteristics for beamforming. However, there is a price to pay in computation energy. If the energy cost required to compute the sensor signal energies and quicksort is large compared to LMS beamforming, then the extra scalability is not worth the effort. For our example, the overhead computational energy was 8.8 μJ, only 0.41% the required energy for 2 sensor LMS beamforming.

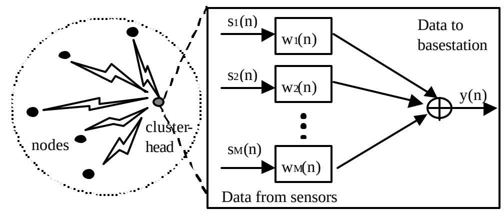

Figure 2. Beamforming algorithm.

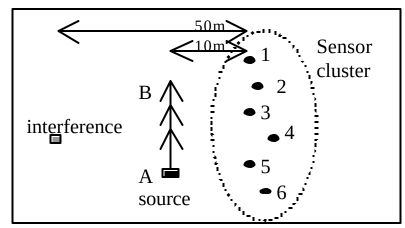

Figure 3. The testbed of sensors to show E-Q scalability.

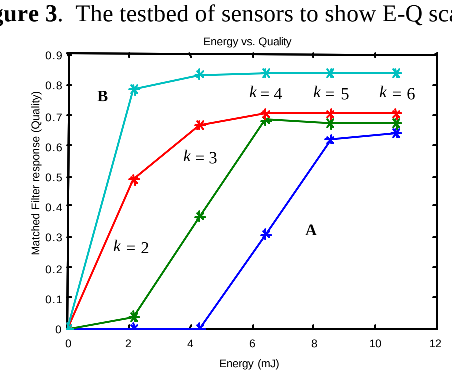

Figure 4. The E-Q snapshot as the source moves from location A to B for the scenario of LMS beamforming with a pre-set order of sensor data.

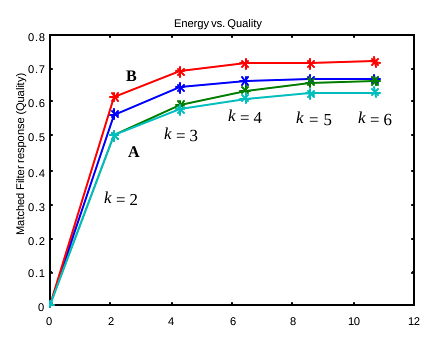

Figure 5. The E-Q snapshot as the source moves from location A to B for the scenario of LMS beamforming with significance ordering of sensor data.

4. SCALABLE NETWORK PROTOCOLS

We have developed a sensor network communication protocol called LEACH (Low Energy Adaptive Clustering Hierarchy) [8]. LEACH is a cluster-based protocol that uses local data fusion and classification to greatly reduce the amount of information that must be transmitted to the base station. Using LEACH, the sensor nodes transmit their data to a local cluster-head, where beamforming is performed to fuse the data, as shown in Figure 2. The beamformed signal is then transmitted to the end-user.

Since the cluster-heads in LEACH are local control centers, the LEACH framework provides several means for achieving energy-scalability. One method of scaling the protocols is to precede data transfers with high-level negotiation using meta-data descriptors, as in the SPIN protocol [9], to ensure that only data that provides new information is transmitted to the cluster-head. This “LEACH with negotiation” protocol can be used with the energy-scalable beamforming algorithm. Using this protocol, each sensor node computes its signal energy and sends this small amount of information (meta-data) to the cluster-head. The cluster-head performs the quicksort to determine which sensors have the highest quality data and requests data from only the k highest energy sensors out of N total sensors. Only nodes that receive requests will transmit data. This saves $N-k$ nodes from energy-draining transmissions when their data is not needed.

We ran the LEACH algorithm with and without the negotiation stage and computed the total energy dissipated in the system, as shown in Figure 6. Since the highest quality for a given energy is achieved when the quicksort method is used to determine which signals to beamform, the cluster-head must obtain all the data and perform the signal energy and quicksort computations. Therefore, the energy in the LEACH case accounts for the transmission of all N signals to the cluster-head, the signal energy computation, the quicksort computation, and the LMS beamforming of k signals. The LEACH with negotiation algorithm energy accounts for the signal energy computation for all N signals, the transmission of N meta-data messages containing the signal energy to the cluster-head, the quicksort computations, the transmission of k request messages from the cluster-head to the nodes, the transmission of k data signals back to the cluster-head, and the LMS beamforming of these signals. Therefore, the LEACH with negotiation algorithm trades off transmission of several small meta-data messages to ensure no extra data is transmitted that will not be used at the cluster-head. This extra negotiation significantly reduces energy dissipation in the network, by about 70% when $k<N$. However, when $k=N$, the extra meta-data negotiation adds energy dissipation to the system (less than 1%). Therefore, the network should perform negotiation only if $k<N$.

Figure 7 shows the average E-Q curves that can be achieved for LEACH with and without data negotiation, using the testbed shown in Figure 3. This shows that while both algorithms are energy-scalable and satisfy the criterion in Eqns. 2 and 3, LEACH with negotiation is more energy-efficient (the E-Q curve is steeper) and hence has more desirable E-Q scalability.

These results reflect the relative costs of computation and communication. If the cost for computation is much more than the cost for communication, there will be minimal relative energy savings using this approach. However, if computation is cheap compared to communication, this approach can save tremendous amounts of energy.

5. CONCLUSION

We have shown that sensor networks can achieve different points on the energy-quality curve by implementing adaptive algorithms and protocols. Accurate models for computation and communication energy dissipation in a sensor node enable the system to determine how to adapt in order to meet a given quality or energy requirement. The techniques discussed here (e.g., algorithmic transformations such as the *most significant first* and protocol scaling through event-driven operation and data negotiation) can be applied to different algorithms and protocols to achieve desirable energy-scalability, providing the flexibility to meet the varying needs of the end-user.

ACKNOWLEDGEMENTS

The authors would like to thank M. Bhardwaj and S.-H. Cho for valuable discussions. This work was supported in part by DARPA. W. Heinzelman is supported by a Kodak Fellowship and A. Wang is supported by a Lucent Fellowship.

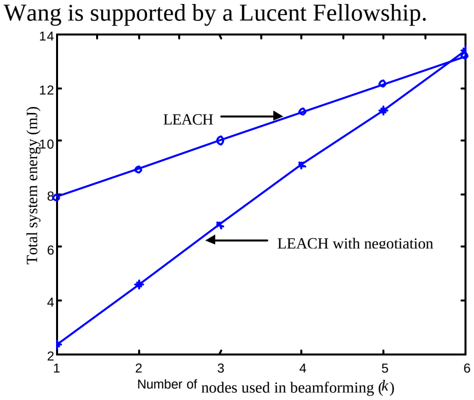

Figure 6. Using negotiation to precede data transfer in LEACH can significantly reduce the energy dissipation.

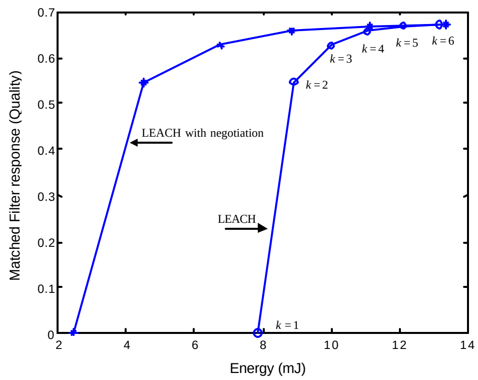

Figure 7. E-Q curve for LEACH with and without data negotiation.

REFERENCES

1. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Scalable Coordination in Sensor Networks,” *Proc. Mobicom ’99*, Seattle, WA, Aug. 1999, pp. 263-270.
2. J. Kahn, R. Katz, and K. Pister, “Mobile Networking for Smart Dust,” *Proc. Mobicom ’99*, Aug. 1999, pp. 271-278.
3. A. Chandrakasan et al., “Design Considerations for Distributed Microsensor Systems,” *Proc. CICC ’99*, May 1999.
4. G. Asada et al., “Wireless Integrated Network Sensors: Wireless Systems on a Chip,” *Proc. 1998 European Solid-State Circuits Conference*.
5. S. Nawab et al., “Approximate Signal Processing,” *Journal of VLSI Sig. Proc. Systems*, Vol 15, No. 1, Jan. 1997.
6. K. Yao et al., “Blind Beamforming on a Randomly Distributed Sensor Array System,” *IEEE Journal on Selected Areas in Communication*, vol. 16, no.8, Oct. 1998.
7. A. Wang, W. Heinzelman, and A. Chandrakasan, “Energy-Scalable Protocols for Battery-Operated Microsensor Networks,” *Proc. SiPS* ‘99, Oct. 1999, pp. 483-492.
8. W. Heinzelman, A. Chandrakasn, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” *Proc. HICSS ‘00*, Jan. 2000.
9. W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” *Proc. Mobicom ’99*, Aug. 1999, pp. 174-185.