Title: An Integrated Design Environment to Evaluate Power/Performance Tradeoffs for Sensor Network Applications[1]

Authors

Mr. Amol B. Bakshi *(first author, corresponding author)*
Dept. of Electrical Engineering-Systems
University of Southern California
3740 McClintock Ave, EEB 234
Los Angeles, CA 90089
Ph: 213-740-2239, Fax: 213-740-9127
Email: abakshi@usc.edu
Country of Citizenship: India

Mr. Jingzhao Ou
Dept. of Electrical Engineering-Systems
University of Southern California
3740 McClintock Ave, EEB 234
Los Angeles, CA 90089
Ph: 213-740-2239, Fax: 213-740-9127
Email: ouj@usc.edu
Country of Citizenship: China

Prof. Viktor K. Prasanna *(presenting author)*
Dept. of Electrical Engineering-Systems
University of Southern California
3740 McClintock Ave, EEB 200C
Los Angeles, CA 90089
Ph: 213-740-4483, Fax: 213-740-4418
Email: prasanna@usc.edu
Country of Citizenship: USA

Areas:

- Networked Embedded Systems
- Performance Modeling and Simulation for Benchmarking Embedded Systems
- Automated Tools for Embedded System Development

[1] This work is supported by the DARPA Power Aware Computing and Communication Program under contract no. F33615-C-00-1633 monitored by Wright Patterson Air Force Base.

ABSTRACT

Networks of inexpensive, low-power sensing nodes that can monitor the environment, perform limited processing on the samples, and detect events of interest in a collaborative fashion are fast becoming a reality. Examples of such monitoring and detection include target tracking based on acoustic signatures and line-of-bearing estimation, climate control, intrusion detection, etc. The advances in low-power radio technology are making wireless communication within sensor networks an attractive option. However, it is typically difficult or impossible to replenish energy resources available to a portable sensor node, once it is deployed. Maximizing the life of sensor nodes is an overriding priority, and different energy optimization techniques are being developed to addresses computation/communication tradeoffs. A large number of research efforts are focusing on different aspects of the general problem of designing efficient sensor network-based systems - where the metrics to measure efficiency vary from system to system.

With technological advancements such as silicon-based radios expected to become a reality in a few years, designers of sensor network-based systems will be faced with an extremely large set of design decisions. Each choice will affect the overall system performance in ways that might not always be cleanly modeled. In addition to the research challenges in design and optimization, the practical aspects of designing real-world sensor networks will become equally important. For example, the ability of the design framework to allow rapid specification and evaluation of a particular network configuration is crucial for a more exhaustive exploration of the design space. A design environment for future sensor networks should provide tools and formal methodologies that will allow designers to model, analyze, optimize, and simulate such systems. In the context of our work, design and optimization of a sensor network application involves determining the task allocation to different sensor nodes and the inter-node communication mechanism. Design of the sensor node hardware itself is also an area of active research. However, we assume that a set of node architectures is already available to our end-user, and the design problem is restricted to using the available hardware (with flexibilities, if any, such as dynamic voltage scaling) to efficiently implement the target application.

We take a simple, seven-node wireless sensor network for acoustic detection [5] (Automatic Target Recognition) as the case study and demonstrate (i) a modeling and simulation methodology for a class of sensor networks, and (ii) a software framework that implements our methodology. Our formal application model is illustrated in Fig. 1. We use a data flow graph representation to model the computing tasks and their data dependencies. The end-to-end application consists of two types of such data flow graphs: the first type denotes the processing that has to be performed for each sample before it is ready to be 'fused' with results from other sensors, and the second type represents the computing involved in data fusion. Specifically, in our case study, a Fast Fourier Transform (FFT) operation is the only task that is performed on each block of sampled data. The outputs of FFTs from all seven nodes are provided as an input to the collaborative computing part, which consists of delay and sum beamforming (BF), and line-of-bearing (LOB) estimation. The result of collaborative computation in such a cluster model of sensor networks has to be transmitted to some observer. This is accomplished by designating one of the nodes as the cluster-head, which could be equipped with more powerful communication facilities than other sensor nodes. All communication within our cluster is one-hop, and processing of a particular data sample (FFT/BF/LOB) occurs either on its home node, or the cluster-head, or partly on both. Simulating a completely specified instance of the above class of sensor networks involves many challenges. None of the existing network simulators to our knowledge models the internal architecture of the processing nodes in the network. This is because the focus of most network simulators is on protocol development and empirical analysis. Except in areas such as high-speed router design, the node internals have little or no impact on decisions related to protocol design.

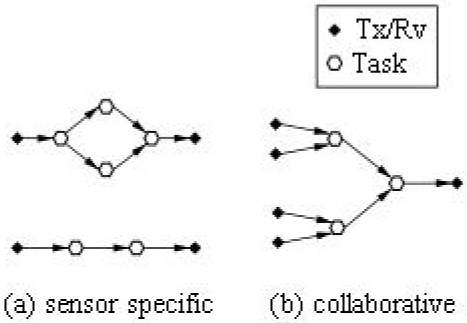

Figure 1: Application Model

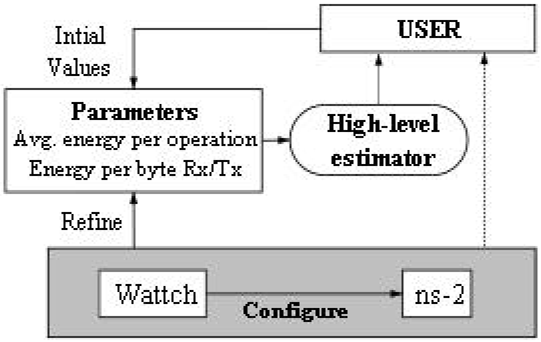

Figure 2: Simulation and model refinement

Also, processor simulators do not model the environment outside the chip boundary. Therefore, to obtain detailed and accurate performance estimates for the entire system, we propose a technique to automatically generate network scenarios based on results from low-level node simulations. The network simulator is configured using the generated scenarios, and the individual simulation results are merged and presented to the end-user as a whole. Such a 'horizontal' simulation is accomplished through the use of a central data repository for model information, which means that the simulators never have to directly interact with each other. The simulators we integrate provide estimates about energy consumption, thereby assisting in a power/performance analysis of a specific system configuration. Our design framework facilitates multi-granular simulation, i.e., simulating the same system configuration by using simulation models at different levels of abstraction. Typically, coarse-grained models provide rapid estimates, but need to make approximations about system behavior that might not be very accurate. For such a scenario, we demonstrate a form of analytical model refinement (see Fig. 2), i.e. the data from low-level simulations can be automatically processed to 'distill' parameter values used by high-level simulators. Naturally, the exact processing has to be specified by someone with knowledge of the analytical model semantics.

Our design environment provides the following capabilities to the user:

- To graphically describe the target application, node architecture, network configuration, and task-to-node mapping.
- To change (reconfigure) the system model to explore alternate designs. Some of the parameters that can be manipulated by the designer include receive/transmit power of the radio, voltage/frequency setting of the processor, cluster geometry, propagation models, etc.
- To automatically simulate a design using a coarse system model.
- To automatically configure and execute low-level simulators for the node (Wattch [3]) and the network (ns-2 [4]) and obtain system-wide energy and latency estimates.
- To automatically update high-level model parameters using low-level simulation statistics.
- To graphically visualize simulation results and facilitate (manual) identification of power/performance bottlenecks in the design.

This work is an illustration of the general approach of the MILAN [1] project. A modeling and simulation framework based on the first version of the 7-node ATR system model was implemented in [2]. The primary focus of that work was a prototype demonstration of simulator integration and model refinement. Therefore, the system model itself lacked generality. Also, we use a relatively more detailed version of the high-level estimator implemented for [2]. This work represents a significant step towards the ultimate goal of a design environment for automatic optimization and synthesis of sensor network applications.

REFERENCES

[1] A. Bakshi, V. K. Prasanna, A. Ledeczi, A. Agrawal, J. Davis, B. Eames, S. Mohanty, V. Mathur, S. Neema, G. Nordstrom, C. Raghavendra, and M. Singh, “MILAN: A model based integrated simulation framework for design of embedded systems,” *ACM SIGPLAN 2001 Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, June 2001.

[2] A. Bakshi, J. Ou, and V. K. Prasanna, “Power-aware embedded system design using the MILAN framework,” of embedded systems,” *IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT)*, May 2002.

[3] D. Brooks, V. Tiwari, and M. Martonosi, “Wattch: A framework for architectural-level power analysis and optimizations,” *International Symposium on Computer Architecture (ISCA)*, June 2000.

[4] Network Simulator - ns-2, http://www.isi.edu/nsnam/ns

[5] N. Srour and J. Robertson, “Remote netted acoustic detection system: final report,” *ARL Technical Report*, ARL-TR-706, April 1995.