TinyBench: The Case For A Standardized Benchmark Suite for TinyOS Based Wireless Sensor Network Devices *

Mark Hempstead, Matt Welsh, David Brooks
Division of Engineering and Applied Sciences, Harvard University
Email: mhempste, mdw, dbrooks@eecs.harvard.edu

Abstract

The growing wireless sensor network research community lacks a standard method for evaluating hardware platforms. Traditional benchmark suites do not sufficiently address the needs of sensor network designers. This work provides motivation for a benchmark suite and details an approach for benchmarking TinyOS compatible hardware. To aid the development of future hardware architectures, we propose the creation of a standard single node benchmark suite, based on both real applications and "stressmarks." We present sample benchmark results and call for further work in this area.

1. Introduction

The wireless sensor network community is growing rapidly. New academic and commercial sensor node platforms are introduced every year, with more advanced designs currently under development. Like the early days of desktop microprocessor design, there are no metrics currently available to effectively compare sensor node platforms and aid in the choice of the "best" platform for a given deployment. Similarly, sensor network hardware designers do not have a representative set of applications to facilitate circuit and architectural decisions.

This work proposes the creation of *TinyBench*: a standardized benchmark suite for TinyOS[1] based wireless sensor network hardware. We argue that the best platform for a given deployment is not solely based on performance, but rather, on the fit of the entire hardware system to the requirements of the application to be deployed. We propose that a single node benchmark suite should be made up of both *stressmarks*, (applications that stress a particular component of the system), as well as real applications.

∗ This material is based upon work support by the National Science Foundation under grant number 0330244. Any opinions, finding, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

TinyOS was created to simplify code development and function across multiple platforms. Because TinyOS is emerging as the standard development platform for wireless sensor networks, there poses a possible calling to leverage this platform in order to design a benchmark suite.

2. Benchmarking Sensor Network Nodes

Because wireless sensor networks do not fit traditional performance based benchmarks suites, a new set of benchmarking goals is required. First, because of the large variety of sensor network devices (including SoC and commodity designs), it is important to create a benchmark suite which can operate across all platforms. Secondly, because sensor network deployments operate within a fixed energy budget, a benchmark suite must characterize the energy consumption of typical tasks. Lastly, sensor networks respond to the environment, often with real time requirements. Therefore, a good benchmark suite will characterize the latency of common tasks.

Historically there were several different classes of benchmarks used to characterize desktop processors, from real applications to synthetic benchmarks such as Whetstone. We introduce four different classes of benchmarks specific to sensor network nodes.

1. Hardware Component Level Microbenchmarks
2. Node Level Stressmarks
3. Node Level Real Applications
4. Network Level Real Applications

Component level microbenchmarks test one individual component of the system (i.e. the radio) in order to characterize that particular component, which is useful for the construction of detailed power models. We define a stressmark as a benchmark that exercises one particular area of the hardware device. The device is stressed to the breakpoint - revealing its full capability. Node level real application benchmarks test actual deployable applications on one single node. Benchmarks constructed from network level

Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)
0742-1303/04 $ 20.00 IEEE

BENCHMARK:	Radio
DATE:	05/20/2004
HW PLATFORM:	Mica2
TinyOS VERSION:	1.1.0
CODE SIZE ROM:	9358 bytes
CODE SIZE RAM:	378 bytes
TIME to Complete:	727.18 ms
AVERAGE POWER:	39.81 mW

Table 1. Example TinyBench Benchmark Report

real applications reflect the true nature of a wireless sensor network environment.

For this particular benchmark suite, we propose the use of node level stressmarks and node level real applications. The other two classes are useful but are difficult to measure.

Benchmarking wireless sensor network hardware requires a completely different approach than traditional performance based measurements. Depending on the class of application, a developer would be concerned with different hardware characteristics. Table 2 displays a sample TinyBench report. We envision a report of similar construction provided for sensor network application developers and hardware designers. Users of the TinyBench reports would then be able to parse out information, such as power consumption, that they are interested in.

3. Preliminary Results

We evaluated four test applications on three sensor network platforms (Mica, Mica2, Mica2dot). These test applications include:

- Surge[4] - A multihop routing application
- TinyDB[2] - a declarative query engine application
- ECC[3] - a CPU intensive stressmark
- Radio - sends 10 messages back to back, stressmark

Figure 1 displays average power for the sample benchmarks. The data shows that the newer and smaller mica2dot platform is, on average, more power efficient. Most current TinyOS applications do not power down during idle periods which caused the real applications (Surge, TinyDB) and stressmarks to exhibit similar power consumption.

The results present are a sample of the platform and architectural comparisons that would be possible after the creation of TinyBench. Our data analysis pose several issues to be addressed: because the platforms we tested use the same microcontroller and similar radio chips the results were similar. Sensor network applications traditionally do not have a fixed point of completion which is required for effective benchmarking.

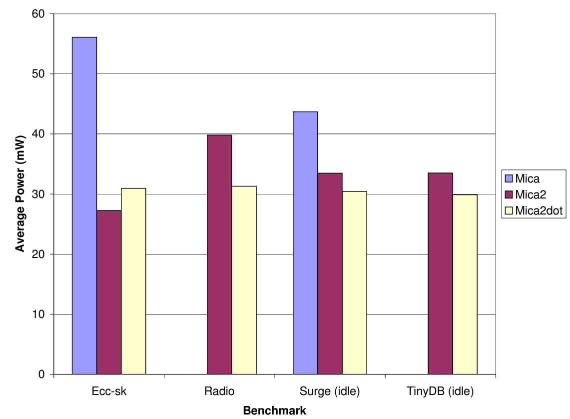

Figure 1. Average Power for the sample benchmarks *The Mica mote suffered a hardware failure and as a result it was not possible to measure two of the benchmarks.*

4. Conclusion and Future Directions

This work fills the void within the sensor network hardware community for a standard benchmark suite. Our research presents a benchmark selection and evaluation methodology as well as sample benchmarks and results. We hope that these findings will motivate the research community to submit benchmark applications and begin using TinyBench for platform selection and to motivate future hardware development.

The next major steps are as follows: we need to write several sets of cross platform stressmarks while managing to appeal to the sensor network community with a *call for benchmarks.* With that, the first version of the TinyBench benchmark suite and reporting format should be released. In addition, several platforms other than the Mica should be tested. Finally, TinyBench would greatly contribute to the future of hardware development by facilitating the construction of an architectural level simulator.

References

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.

[2] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks. In *Proc. the 5th OSDI*, December 2002.

[3] D. Malan. Crypto for tiny objects. Technical Report TR-04-04, Harvard University, January 2004.

[4] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. 2003.