Demo Abstract: Feature-Rich Platform for WSN Design Space Exploration

Bernhard Buchli, Mustafa Yuecel, Roman Lim, Tonio Gsell, Jan Beutel
Computer Engineering and Networks Laboratory
ETH Zurich, 8092 Zurich, Switzerland
{buchli,yuecel,lim,gsell,beutel}@tik.ee.ethz.ch

ABSTRACT

Wireless Sensor Network applications require dependable platforms that deliver correct and reliable operation over long periods. However, application characterization and exact system requirements specification can be complicated due to unknown environmental factors and system limitations. In this demo abstract we present a new approach to design space exploration using a feature-rich system for facilitated experimentation. We argue that results obtained from experimentation with this platform allow rapid specification of optimized sensor systems at reduced cost.

1. INTRODUCTION

New and emerging Wireless Sensor Network (WSN) applications may demand high processing power, complex sensor circuitry, high duty-cycles, and variable-bandwidth communication links to guarantee reliable and dependable operation. The design and operation of such highly application specific, and very resource constraint systems requires numerous trade-off decisions that affect performance, reliability, and system lifetime. Conventional design approaches generally require complete system requirements specifications before a first prototype is implemented. The functionality is then incrementally enhanced, leading to hardware and software revisions before a deployable system is built. However, three main factors complicate the conventional design approach. First, missing or incomplete requirements specifications early in the design process have wide-spread implications. For instance, design choices based on erroneous or inappropriate assumptions may yield unnecessary, or worse, limited system functionality, affecting both the performance and energy consumption. Second, limited resources of networked embedded systems, coupled with limitations due to fixed design choices prohibit extensive design space exploration. Finally, limited visibility into the system significantly complicates the design and debugging process.

In this demo, we describe a new approach to design space exploration of complex WSN systems with many, highly

IPSN'11, April 12–14, 2011, Chicago, Illinois.

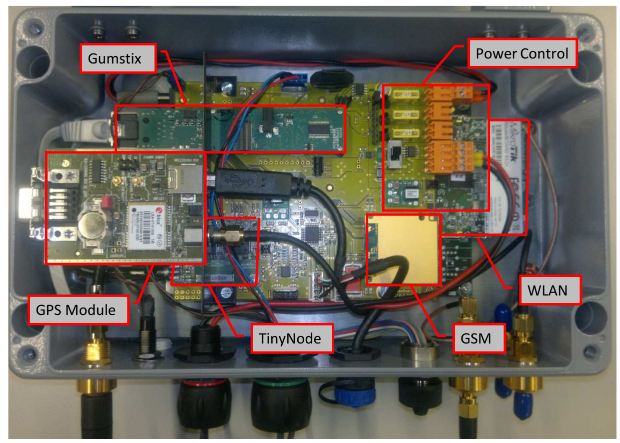

Figure 1: *CoreStation*: Over-provisioning for facilitated design space exploration of WSN systems.

stochastic parameters that depend to a large part on environmental conditions and are therefore difficult to obtain. Our approach eliminates the guess-work during design and deployment phases by leveraging a feature-rich, dependable platform for facilitated experimentation and collection of critical runtime parameters. The generic, yet highly flexible architecture permits on-site system and application profiling before an optimized system is designed and implemented.

2. SYSTEM CONCEPT

To alleviate the aforementioned problems, we employ a system that facilitates experimentation and design space exploration before detailed specifications have to be defined. We leverage the functionality and flexibility of an over provisioned system, called *CoreStation* (see Figure 1), to investigate design trade-offs and collection of critical run-time parameters under real-world deployment conditions.

Our system architecture provides two operating modes. First, the high-performance experimentation mode delivers functionality of a powerful, yet generic prototype, which enables rapid evaluation of design parameters for new or alternative system architectures. In this mode, all system components (see Section 3.1) are at the experimenter's disposal. The software framework (see Section 3.2) provides full support for performance profiling and power tracing. With a high-bandwidth network link and the built-in database connection manager, this mode further permits the *CoreStation* to be deployed as a conventional WSN basestation.

The second mode, similar to [6, 7], is targeted at evaluation under actual deployment conditions. While we assume that the *CoreStation* in performance mode has ample energy supply, the low-power mode considers the system's dependence on finite battery capacity and limited scavenging opportunities. This implies that the *CoreStation* must be able to handle sporadic power outages and support disconnected, autonomous operation without data losses. To this end, data consistency is achieved by storing local copies of sensor readings, logs and traces until a successful synchronization with the master database has been achieved. Reachability, and hence observability, is guaranteed by employing redundant high-bandwidth links together with a WSN mote's ultra low-power radio transceiver. The low-power network integration further permits remote control over power-hungry components. The state of individual high-power modules can be altered using special beacons, which provides increased control and extensive power saving opportunities.

Despite the overhead and potential difficulties arising in the redesign (and related verification) to map resources and software to the final target platform, early experiences with the *CoreStation* have produced insights that can facilitate the definition of system requirements specifications and related trade-offs for new and improved architectures.

3. SYSTEM IMPLEMENTATION

Figure 1 shows the *CoreStation*'s hardware platform enclosed in a water-proof box with EMP-protected antennas, power, and debug connectors. The following two sections provide an overview of the hard- and software architecture.

3.1 Hardware

To accommodate for the evaluation of diverse application scenarios, the hardware architecture is partitioned into three layers: power and interfacing, processing, and networking.

Power and Interfacing. The Baseboard accepts 5.5V-18V unregulated input and generates multiple software controlled and monitored power domains. In addition to on-board temperature, humidity, voltage and current sensors, two USB interfaces, one configurable serial interface, and one I^2C bus for connecting external sensors are available. The Baseboard further provides interfaces and communication planes for the Gumstix embedded PC, a TinyNode [5] (or compatible) mote, and a GSM modem. Accurate time across power cycles is maintained by a battery-backed RTC. Finally, for on-site maintenance a serial debugging console can be accessed without the need to open the enclosure.

Processing. The Gumstix verdex pro XL6P [1] represents the main processing element of the *CoreStation*. The 600MHz XScale CPU, together with 128MB of RAM, and 8GB of Flash memory provides enough resources to run a Linux operating system (OS), with most of the features and utilities available on desktop distributions. In the current set-up we further utilize a TinyNode184 mote, which features an MSP430 low-power processor running the TinyOS operating system. The Gumstix and TinyNode can communicate via the TinyOS serial protocol over UART.

Networking. Our setup employs two redundant high-bandwidth links, WLAN and GSM, which provide reliable access and visibility into the system. The *CoreStation* further supports ethernet or satellite links. The TinyNode's radio permits integration into ultra low-power Dozer [4] networks. Special Dozer beacons sent via the low-power network enable remote-control over the processing elements.

3.2 Software

The custom software framework, called *BackLog*, runs on top of the Linux OS. It provides a plug-in API, power control module, plug-in execution scheduler, and optionally handles connection to GSN [2] backend servers. Built-in monitoring and statistics collection, together with high-level language support and well-established utilities (ssh, snmp, ntp, etc.) prove to be invaluable for rapidly implementing functionality and analyzing system performance trade-offs. *BackLog* further provides communication interfaces with, and control over hardware components. The ability to interact with individual components, such as monitoring or reprogramming the TinyNode, polling sensors, or enable/disable wireless links, extends the system functionality to an on-site testbed of low-power mote based sensor systems.

4. CASE STUDY

The *CoreStation* has been used sucessfully to define the requirements of a high-alpine, GPS-based land movement monitoring system [3]. On-site experimentation with the *CoreStation* and varying operating parameters, such as number and length of measurement periods, sampling frequency, and component duty-cycles, have led to the definition of an optimized system architecture based on a TinyNode network mote equipped with an off-the-shelf GPS receiver. At a GPS duty-cycle of 12%, this optimized system achieves highly accurate relative positioning while requiring a daily energy supply of just under 860mWh. A comparable GPS logger has 64% higher energy demands despite lacking a radio that would enable online data collection. The energy consumption by the *CoreStation* with appropriate duty-cycles to achieve the same results is 18Wh.

5. CONCLUSION

In this demo we describe a flexible platform that facilitates requirements analysis and specification of WSNs before application specific system parameters have to be defined.

6. ACKNOWLEDGMENTS

The work presented was supported by NCCR-MICS, a center supported by the Swiss National Science Foundation under grant number 5005-67322, and Nano-Tera.ch.

7. REFERENCES

[1] Gumstix. http://www.gumstix.com.

[2] K. Aberer, M. Hauswirth, and A. Salehi. A middleware for fast and flexible sensor network deployment. In *VLDB: Very Large Data Bases*, 2006.

[3] J. Beutel, B. Buchli, F. Ferrari, M. Keller, L. Thiele, and M. Zimmerling. X-sense: Sensing in extreme environments. In *Proceedings of Design, Automation and Test in Europe, 2011 (DATE 2011)*, Grenoble, France, 2011.

[4] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *ACM/IEEE IPSN*, 2007.

[5] H. Dubois-Ferrière, L. Fabre, R. Meier, and P. Metrailler. Tinynode: a comprehensive platform for wireless sensor network applications. In *ACM/IEEE IPSN*, 2006.

[6] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, and D. Estrin. Emstar: a software environment for developing and deploying wireless sensor networks. In *Proceedings of the 2004 USENIX Technical Conference*, Boston, MA, 2004.

[7] N. Sharma, J. Gummerson, D. Irwin, D. Ganesan, and P. Shenoy. Srcp: Simple remote control for perpetual high-power sensor networks. In *Proceedings of the 6th European Conference on Wireless Sensor Networks*, pages 1–16, Cork, Ireland, 2009.