International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-4, September 2013

Power Simulation and Power Profiling of Wireless Sensor Networks

Jan Haase

Abstract: ***Wireless sensor networks (WSN) are battery-powered, thus they should be optimized for low energy consumption already at design time. Current tools offer different levels of power simulation, showing the designer where the energy was spent, where hot spots might be optimizable, and so on. However, a semantic gap still remains open: The question why a specific part of the software was executed and therefore energy was consumed. Especially for WSN this question not only encompasses single devices, but whole networks, which can suffer e.g. from network congestion, attenuation, routing errors, lost packets, etc. This paper presents an overview over existing power simulation tools as well as the new approach of power profiling, which collects much data about many aspects that lead to energy consumption in a network of wireless sensor nodes. The outcome is a statistic about the energy consumption for each single generated transaction, from the moment it was generated until the resulting message is finally received. Using this, the designer is able to optimize the whole system, e.g. by running several simulations with different communication protocols, network topologies, duty cycles, or wake-up timings.***

Index Terms**: wireless sensor networks, WSN, power estimation, power profiling, energy efficiency.**

I. INTRODUCTION

The typical wireless device of today is from one of two classes. There are portable consumer products, small, easy to use, fun to use, smart helpers in daily life. Examples are mobile phones, MP3 players, portable electronic games, GPS devices, etc. The second class are industrial cases, e.g. wireless sensors in cars, airplanes, building automation or even at every single item found in a building or private home—the vision of the Internet of Things [1].

As the device itself is wireless, it has to run on battery power. The growing number of features of today's consumer products need more and more power, so that battery lifetimes shrink considerably—e.g. many mobile phones have to be recharged every day [2], and electric cars only have a range of around 100 miles before they have to be recharged, which takes several hours. Batteries lifetime is one of the important problems of today.

In the area of sensor networks, sensors are part of a wireless network using up much energy for sending and receiving messages [3]. Typical examples are wireless light switches in modern buildings, or pressure monitors in car tires, which have to be able to send information about suddenly changing tire pressures.

A low battery may render a safety device like this inoperable and thus be dangerous. Some of these devices are installed at remote or hard to reach places and hence cannot be recharged easily. Energy efficiency is therefore very important in Wireless Sensor Networks (WSN) [4].

To tackle the problem of short battery lifetimes, system designers take the power consumption of the embedded devices into account already at the time when they are developed. One approach is to include power saving features at run time like automatic shutdown of displays or clock gating to switch off unused parts of the device [5]. Another way is the estimation of power consumption for different sets of system parameters (e.g. duty cycles, communication protocols, bus widths, or processor types) in order to choose the design solution having the lowest energy consumption. This is mainly achieved by means of simulation. As the system being built has to be simulated anyway in order to assure the correct function this simulation can be enhanced to include power consumption information. However, simulation is in most cases slow, i.e. in order to simulate all details of a system, the run time of the simulation is often higher by orders of magnitude than the run time of the final device itself [6]. This slowdown is further increased by including power consumption analyses into the simulation.

This paper gives an overview over some existing solutions for fast system and network simulation with respect to power consumption and shows a new approach, namely the Power Profiling, which offers good results for network simulation.

Section III shows the current approaches of simulation in order to get information about power consumption of a system. The following Section IV explains the new approach presented in this paper. The paper then concludes in Section V.

II. GOAL

Many (sensor or actuator) nodes are wireless, therefore they are battery-powered. To prolong the battery lifetime, typically one or more of the following approaches are used:

- Power management at run time
- Frequent energy recharging or battery changing when the battery runs low
- Energy harvesting at run time
- Build power-optimized systems in the first place

This work focuses on the last approach.

Revised Manuscript Received on 30 September 2013.
* Correspondence Author
Jan Haase, Institute of Computer Technology, Vienna University of Technology, Vienna, Austria.

In order to get information about the power consumption of a system being designed, different versions of the system would have to be implemented. However, nowadays most designs are simulated first and it is therefore possible to try different versions before deciding which one to implement physically. This method is called design space exploration.

Obviously, a fast design process with fewer redesigns dramatically shortens the time-to-market and therefore increases profits. If therefore some inacceptably high energy consumption due to some specific parts of the model (e.g. some memory which was dimensioned too small or usage of an inefficient communication protocol) can be identified early, the designer greatly benefits. Furthermore, an system with an optimized and thereby lower energy consumption even is of use if some of the other aforementioned approaches like energy harvesting are used, as a lower consumption leads to lower requirements for an energy harvester and fewer recharge-cycles.

The simulators thus have to be enhanced to simulate not only the function and some other non-functional properties (e.g. size, hot spots) but also energy consumption. This leads to the method of power simulation.

III. POWER SIMULATION

Figure 1 shows a typical wireless sensor. A device like this normally consists of a sensor part, a transceiver part, a microcontroller and some connection inbetween. The microcontroller activates the sensor, monitors the sensor data, and generates some message to be sent wirelessly using the transceiver. In many cases additional memory enables the microcontroller to keep a history of sensor data and therefore detect trends or unusual readings. A protocol processing unit helps generating the message packets. Finally, a power management unit keeps the energy consumption of the whole device low by e.g. switching off all unused parts (clock-gating) and/or controlling the sleep phases in the duty cycle.

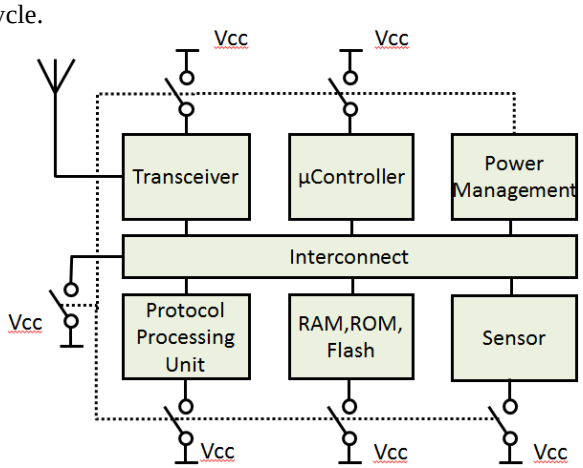

Figure 1 A typical wireless sensor consists of a sensor, a transceiver with an antenna, some microcontroller

Wireless sensor devices like this are deployed in many areas like automotive/car, avionics and space craft environments (distance sensors, pressure sensors, temperature sensors, rain sensors, light sensors, radiation sensors, etc.), buildings (movement sensors, light sensors, intruder detection, fire detection, etc.) [7-9], traffic (street signs, traffic lights, freight containers, etc.), and many more.

In the following subsections some simulation environments for different levels are presented.

A. Step 1: Hardware simulators

A first step for power simulation of wireless sensor networks is the simulation of single nodes. As the nodes are battery-powered, the power consumption of the whole node is of interest, e.g. for estimations of how long a battery with a given capacity will live.

The model of the wireless sensor node is typically simulated on hardware level, i.e. the hardware components are modeled and simulated. However, in most cases the components' energy consumption is only calculated in an "full power", "low power", "sleep mode", or "off" state. This allows only statistics over time, but without knowledge on what components will run for how long, as the concrete simulation use cases cannot consider software at this level.

Well-known simulators for the hardware level are listed in Table 1.

Table 1 Hardware/Circuit Simulators Summary Table

Name	since	Features
SPICE [10]	1973	Functional simulation, integrity checking
HSPICE [11]	1979	Synopsys, based on SPICE
ModelSim [12]	1980s	Mentor, graphical output
Spectre [13]	1990	Cadence, based on SPICE, Verilog-A
SpecC [14]	1999	System-level Design, based on C
SystemC [15]	2000	C++ class library, modelling and simulator in one, HDL, discrete event driven, OSCI/Accellera, IEEE-standard since 2005/2011
SystemC AMS [16]	2000s	Based on SystemC, system-level, mixed signal, timed dataflow, simulation of analog parts

B. Step 2: Hardware and software simulators

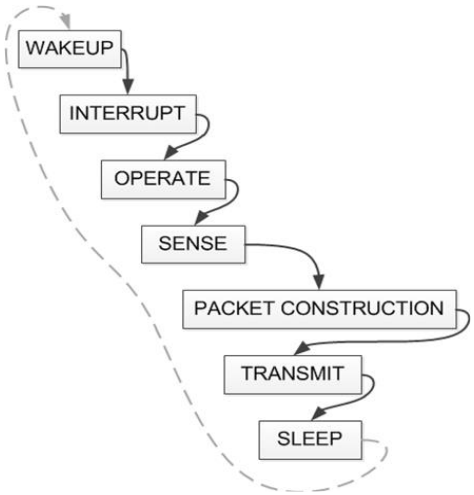

Figure 2 A duty cycle of a typical sensor

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

However, simulation of hardware components is not enough:

- Only a specific node is focused
- Detection of possible hot spots
- Sleep time vs. active time

Some software instructions or states affect more than one hardware component at a time. In order to cover this in a second step, the hardware system with its software has to be simulated simultaneously.

Normally, a sensor node's operation cycle can be described as a state machine. Typically, it looks similar to Fig. 2: The sensor is in sleep mode until it wakes up (due to a specific time or a wakeup signal), gets an interrupt, switches to operate mode, senses the data it has to measure (e.g. a temperature or pressure value), then it generates a message packet in order to transmit it to some central unit or other node and goes to sleep again.

Furthermore, simulation of the hardware and software includes instruction set simulation (ISS). This enables the designer to simulate each instruction in detail.

Table 2 Hardware and Software Simulators Summary Table

Name	since	Features
MATLAB Simulink [17]	1984	MathWorks, Hardware/Software Co-Simulation
Avrora + AEON [18–21]	2005	Simulation for AVR microcontroller (Atmel), extensible; AEON (extension) includes radio propagation model (without noise model)
CooJa [22–26]	2006	Contiki OS Emulator, Java-based, very flexible, heterogeneous networks can be modeled
TOSSIM and PowerTOSSIM [27–29]	2008	TinyOS, only for MICA2 motes, OS level simulation, not cycleaccurate, no network wide power simulation

C. Step 3: Hardware and software and network simulators

Before the arising of WSNs, several simulator extensions already existed for ad-hoc mobile networks, being among the most representatives those developed in Monarch project [30] or GloMoSim [31] and its commercial version QualNet [32]. Soon, extensions for those existing simulators were developed, just before specific tools, designed with WSNs in mind, were released. Table 3 outlines some of these tools [33].

D. The semantic gap

All the tools described in the preceding Sections are able to collect data about energy consumption on different levels. However, they can only answer the question, where was energy consumed.

However, the programmer intending to optimize the system currently being designed needs answers to the following questions:

- Why was energy consumed?
- In which operations was energy consumed?
- What circumstances influenced the amount of energy consumed?
- What where the reasons for high energy consumption?

These questions describe the semantic gap for the programmer. If he could get specific answers to these questions, it would be possible to optimize the system on a higher level, e.g. by comparing different timing or communication protocols for network transactions. In the following Section, power profiling is presented as a solution for the programmer's semantic gap.

IV. POWER PROFILING

The goal of power profiling is to fill the programmer's semantic gap described in Section III.D. For this, data is collected at simulation time on several levels in order to be interpreted as a whole.

Higher levels trigger activities in lower levels, i.e. a network transaction will trigger transmitting some message in the sender node as well as receiving the message in the receiver node, both activities leading to some energy consumption in both nodes' transceiver units on hardware level. Furthermore, the receiver will have to listen for the message for some time (which also costs energy) and if there were communication problems like noise or a too weak signal the message will have to be resent. All this information is not collected by standard power simulation strategies. Power consumption data is collected on

- Network level
- Software level
- Hardware level

The different levels are depicted in Fig. 3 [51].

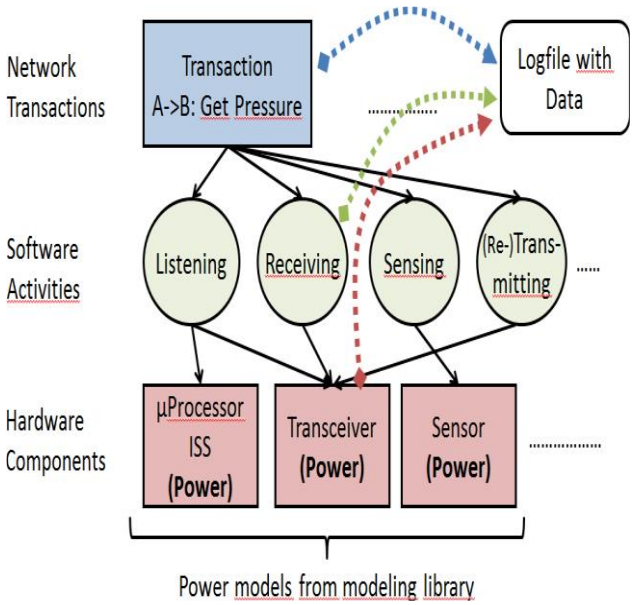

Figure 3 Power Profiling collects data from different layers

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Collecting data for the simulation

Power profiling aims to provide meaningful information associated with software and network power consumption in order to optimize it. Figure 4 depicts the different directions in which low level power consumption data has to be abstracted, giving examples of some abstractions made in all the three perspectives: hardware, software, and network [53].

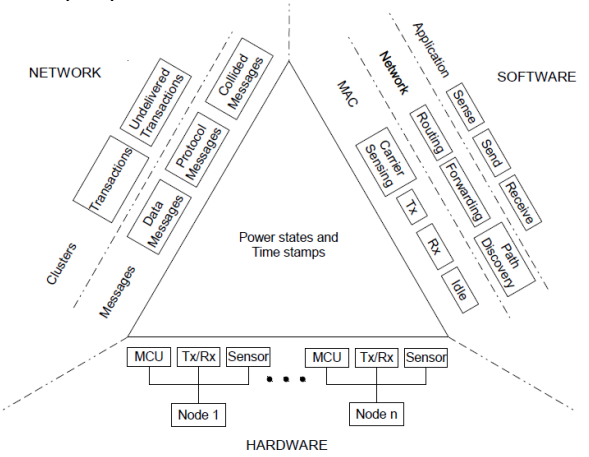

Figure 4 Data about power states and states transitions is recorded in three different directions: hardware, software, and network

On the hardware level, data about hardware modules like sensors, microcontrollers, transceivers is collected, for each node involved. On the software level, actions like sending, receiving, forwarding, routing, etc. are monitored and data collected. On the network level data about whole transactions, message to be sent from which node to which node, possible collisions, lost messages (that will have to be resent), etc. is collected.

Typical simulations of whole systems are already quite complex and therefore take some time. By introducing power profiling, this complexity is increased even more, as much data about the simulated run of the system is generated and collected. An obvious question is whether power profiling is still feasible in terms of simulation speed. A solution to this problem is use of Transaction Level Modeling (TLM) [54,55].

Although TLM is intended for simulation of bus communication, the behavior in terms of simulation, of both a bus-based system and a Wireless Sensor Network, is not that different [56]. The communication channel, e.g. the air, acts in a similar way as a bus do. In TLM, transactions are passed from initiators to targets, going through any number of interconnects. Extrapolating this components to node-to-node communication, the sender node will be the initiator, the receiver will be the target, and the air will be an interconnect, which modifies the transaction by adding attenuation, delay, noise and interferences [57,58].

The simulation has been provided with a state machine class which must be set up by the programmer, defining all the states, with their corresponding power consumption values, and all possible transitions, which will be later the only permitted by the simulator. A state machine can be associated to any module created by the programmer. State machines are automatically registered in the system, so that any transition is automatically reported to the power logger. This relieves the responsibility of reporting changes in power consumption from the programmer, which must only define the state machine and trigger the transitions.

Table 3 Hardware, Software, and Network Simulators Summary Table (from [33])

Name	Based on	Language	Since	Propagation model	Other features
ns-2 [34,35]	–	C++	2009	Free space, two ray ground, shadowing	Huge protocol library
SensorSim [36]	ns-2	C++	2001	Free space, two ray ground, shadowing ns-2 protocol library, hardware-in-the-loop	
Prowler [37]	MATLAB	MATLAB	2004	Deterministic and probabilistic models	Accurate channel estimation
J-Sim [38,39]	–	Java	2006	Free space, irregular terrain (Longley-Rice) and two ray ground	Mobility model, hardware-in-the-loop
VisualSense [40]	Ptolemy II [41,42]	Java	2008	Erasure, limited range and distance power loss models	Generic propagation model for communication and sensor channels, TinyOS emulation through Viptos.
SENSE [43,44]	COST [45]	C++	2006	Free space, limited range	Some component implementations
PAWiS [46]	OMNeT++ [47]	C++	2008	Deterministic Attenuation	Module library, visualization tools, dynamic network behavior through Lua
SNOPS [48,49]	SystemC	C++	2011	Deterministic and probabilistic models	Interferences, HW system models in SystemC
IDEA1 [50]	SystemC	C++	2011	No specific model is available	HW system models in SystemC

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Activities are completely user-defined. The simulator just registers them so that the information about current activities are attached to every power report. In the case of the transaction tracking, a transaction identity is also included in all power reports. Besides, an extension can be added to the generic payload, which contains the accumulated power consumed by the transaction and maintains its history, which can be easily reconstructed later, to evaluate unfortunate forks. It also provides information about the transaction consumption in execution time [59]. The result is a statistic showing for each generated transaction the energy consumption it triggered throughout the whole network.

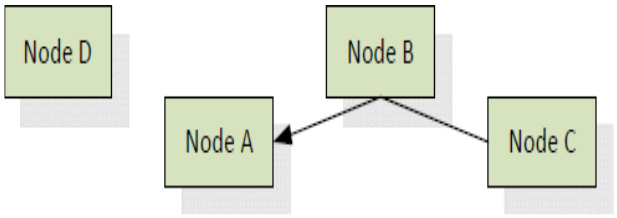

Figure 5 A simple network topology: Node C sends some request to node A

An exemplary transaction could be like this:

1. The central node C decides that sensor node A should send its current sensor value (consumption in microcontroller of C)
2. C generates a message to A containing the data request for A's sensor value (consumption in microcontroller and protocol processing unit of C)
3. The resulting message is sent to A, but routed via node B (consumption in transceivers of C and B)
4. Node B has to identify the destination of the message, find out it's not for B, and resend it to node A (consumption in microcontroller of B, transceiver of B, transceiver of A)
5. Node A receives the message and reads the request (consumption in transceiver and microcontroller of A)
6. Node A triggers a sensor reading (consumption in sensor of A)
7. Node A creates a reply message to central node C, having the same message ID as the request (consumption in the microcontroller and protocol processing unit of A)
8. Node A sends the message to C via A. However, node D is also in proximity of A and therefore also receives the message (consumption in transceivers of B and D)
9. Node D has to identify the destination of the message, find out it's not for D, and a resend would go via node A, therefore it just drops the message (consumption in microcontroller of D)
10. Node B has to identify the destination of the message, find out it's not for B, and resend it to node C (consumption in microcontroller of B, transceiver of B)
11. Node C receives the message (consumption in transceiver of C)
12. Node C reads the reply and maybe stores the sensor value or starts another transaction, e.g. a command to another node (consumption in microcontroller of C)

This case might be even more complex when sleep times of nodes are considered (listening only at specific time frames, several resends until the messages ar received, etc.)

A wakeup receiver [60] can help lowering the energy consumption of a node which is not the destination of a received message: It reads the destination and possibly forwards it to the next node. Only if the message was sent to this device, the wakeup receiver wakes up the microcontroller (which consumes much more energy than the wakeup receiver).

V. CONCLUSION AND OUTLOOK

The approach of power profiling presented in this paper offers possibilities to collect energy consumption data in order to optimize devices at design time. A special focus lies on the optimization of wireless sensor networks, where the total energy consumption of transactions is introduced, including all energy consumed in involved nodes (hardware, software, and network). This approach enables the designer to even optimize on a higher level, e.g. the network topology or communication protocols.

REFERENCES

1. J. Conti, "The internet of things," Communications Engineer, vol. 4, no. 6, pp. 20 –25, 2006.
2. A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda, "Characterizing and modeling user activity on smartphones," Northwestern University, Tech. Rep., Mar. 2010.
3. H.-Y. Zhou, D.-Y. Luo, Y. Gao, and D.-C. Zuo, "Modeling of node energy consumption for wireless sensor networks," Wireless Sensor Network, vol. 3, pp. 18–23, Jan. 2011.
4. A.Willig, "Recent and emerging topics in wireless industrial communications: A selection," Industrial Informatics, IEEE Transactions on, vol. 4, no. 2, pp. 102 –124, May 2008.
5. Q. Wu, M. Pedram, and X. Wu, "Clock-gating and its application to low power design of sequential circuits," Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, vol. 47, no. 3, pp. 415 –420, Mar. 2000.
6. E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Mario, and J. Garcia-Haro, "Simulation scalability issues in wireless sensor networks," Communications Magazine, IEEE, vol. 44, no. 7, pp. 64 – 73, July 2006.
7. D. Bruckner, J. Haase, P. Palensky, G. Zucker, „Latest trends in integrating building automation and smart grids", IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, pp. 6285-6290, 2012
8. M. Rathmair, J. Haase, "Simulator for Smart Load Management in Home Appliances", Proceedings of SIMUL 2012, The Fourth International Conference on Advances in System Simulation, pp. 1-6, 2012
9. J. Wenninger, J. Moreno, J. Haase, "Model Based Design of Smart Appliances", book chapter in "Embedded Systems for Smart Appliances and Energy Management", pp. 41-51, Springer 2013
10. L.W. Nagel and D. Pederson, "Spice (simulation program with integrated circuit emphasis)," EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M382, Apr 1973. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/1973/22871.html
11. "Synopsys HSPICE," homepage: http://www.synopsys.com/tools/Verification/AMSVerification/ CircuitSimulation/HSPICE/Pages/default.aspx.
12. "Mentor ModelSim," homepage: http://model.com.
13. "Cadence Spectre," homepage: http://www.cadence.com/products/rf/spectre circuit/pages/-default.aspx.
14. R. Dömer, A. Gerstlauer, and D. Gaijski, "SpecC language reference manual (version 2.0)," http://www.ics.uci.edu/ specc/reference/SpecC LRM 20.pdf.
15. "SystemC," Open SystemC Initiative. [Online]. Available: http://www.systemc.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

16. "SystemC AMS," homepage: http://www.accellera.org/downloads/standards/systemc/ams.
17. "MATLAB Simulink," homepage: http://www.mathworks.de/products/simulink.
18. UCLA Compilers Group, "Avrora: The avr simulation and analysis framework," 2011. [Online]. Available: http://compilers.cs.ucla.edu/avrora/
19. B. L. Titzer, D. K. Lee, and J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," in Proceedings of the 4th international symposium on Information processing in sensor networks, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005.
20. R. de Paz Alberola and D. Pesch, "AvroraZ: extending Avrora with an IEEE 802.15.4 compliant radio chip model," in Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, ser. PM2HW2N '08. New York, NY, USA: ACM, 2008, pp. 43–50.
21. Center for Adaptive Wireless Systems, "Avroraz: enabling ieee 802.15.4 compliant emulations," 2011. [Online]. Available: http://citavroraz.sourceforge.net/docextensions.html
22. F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in Local Computer Networks, Proceedings 2006 31st IEEE Conference on, Nov. 2006, pp. 641 –648.
23. J. Eriksson, F. Österlind, N. Finne, A. Dunkels, and T. Voigt, "Accurate power profiling for sensor network simulators," 2008.
24. J. Eriksson, F. Österlind, N. Finne, A. Dunkels, N. Tsiftes, and T. Voigt, "Accurate networkscale power profiling for sensor network simulators," in Wireless Sensor Networks, ser. Lecture Notes in Computer Science, U. Roedig and C. Sreenan, Eds. Springer Berlin / Heidelberg, 2009, vol. 5432, pp. 312–326, 10.1007/978-3-642-00224-3 20.
25. F. Österlind, J. Eriksson, and A. Dunkels, "Cooja TimeLine: a power visualizer for sensor network simulation," in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 385–386.
26. C. A. Boano, K. Römer, F. Österlind, and T. Voigt, "Demo Abstract: Realistic Simulation of Radio Interference in COOJA," in European Conference on Wireless Sensor Networks (EWSN 2011), Feb. 2011.
27. P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," in Proceedings of the 1st international conference on Embedded networked sensor systems, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 126–137.
28. V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in Proceedings of the 2nd international conference on Embedded networked sensor systems, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 188–200.
29. E. Perla, A. O. Cath´ain, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "PowerTOSSIM z: realistic energy modelling for wireless sensor network environments," in Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, ser. PM2HW2N '08. New York, NY, USA: ACM, 2008, pp. 35–42.
30. D. B. Johnson, "Validation of wireless and mobile network models and simulation," in In Proceedings of the DARPA/NIST Network Simulation Validation Workshop, May 1999.
31. X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a library for parallel simulation of largescale wireless networks," in Proceedings of the twelfth workshop on Parallel and distributed simulation, ser. PADS '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 154–161.
32. S. N. Technologies, "Qualnet." [Online]. Available: http://www.scalablenetworks. com/products/qualnet/
33. J. Haase, J. Molina, and D. Dietrich, "Power-aware system design of wireless sensor networks: Power estimation and power profiling strategies," Industrial Informatics, IEEE Transactions on, vol. 7, no. 4, pp. 601 –613, Nov. 2011.
34. "The network simulator - ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/
35. K. Fall and K. Varadhan, "The ns manual," 2010. [Online]. Available: http://www.isi.edu/nsnam/ns/doc/ns doc.pdf
36. S. Park, A. Savvides, and M. B. Srivastava, "SensorSim: a simulation framework for sensor networks," in Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, ser. MSWIM '00. New York, NY, USA: ACM, 2000, pp. 104–111.
37. G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi, "Simulation-based optimization of communication protocols for large-scale wireless sensor networks," in Aerospace Conference, 2003. Proceedings. 2003 IEEE, vol. 3, Mar. 2003, pp. 1339–1346.
38. "J-Sim Official." [Online]. Available: http://sites.google.com/site/jsimofficial/
39. A. Sobeih, J. Hou, L.-C. Kung, N. Li, H. Zhang,W.-P. Chen, H.-Y. Tyan, and H. Lim, "J-Sim: a simulation and emulation environment for wireless sensor networks," Wireless Communications, IEEE, vol. 13, no. 4, pp. 104 –119, Aug. 2006.
40. P. Baldwin, S. Kohli, E. A. Lee, X. Liu, Y. Zhao, C. C. T. Ee, C. Brooks, N. V. Krishnan, S. Neuendorffer, C. Zhong, and R. Zhou, "VisualSense: Visual modeling for wireless and sensor network systems," UCB ERL Memorandum UCB/ERL M04/8, Tech. Rep., 2004.
41. "Ptolemy II," UC Berkeley EECS Dept. [Online]. Available: http://ptolemy.berkeley.edu/ptolemyII/
42. E. A. Lee, "Finite State Machines and Modal Models in Ptolemy II," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-151, Nov 2009. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-151.html
43. G. Chen, J. Branch, M. Pflug, L. Zhu, and B. Szymanski, "SENSE: A Wireless Sensor Network Simulator," in Advances in Pervasive Computing and Networking, B. K. Szymanski and B. Yener, Eds. Springer US, 2005, pp. 249–267, 10.1007/0-387-23466-7 13.
44. C. of Pervasive Computing and Networking, "SENSE: Sensor Network Simulator and Emulator." [Online]. Available: http://www.ita.cs.rpi.edu/sense/
45. G. Chen and B. Szymanski, "COST: a component-oriented discrete event simulator," Winter Simulation Conference, vol. 1, pp. 776–782, 2002.
46. S. Mahlknecht, J. Glaser, and T. Herndl, "PAWiS: Towards a Power Aware System Architecture for a SoC/SiP Wireless Sensor and Actor Node Implementation," in Proceedings of 6th IFAC International Conference on Fieldbus Systems and their Applications, 2005, pp. 129–134.
47. "OMNeT++ Network Simulator Framework," OMNeT++ Community. [Online]. Available: http://www.omnetpp.org/
48. "SNOPS - Sensor Network Simulation by Power Simulation," project funded by FFG (FITIT program), Austria.
49. J. Moreno Molina, J. Haase, and C. Grimm, "Energy consumption estimation and profiling in wireless sensor networks," in ARCS '10 - 23th International Conference on Architecture of Computing Systens 2010 Workshop Proceedings, Feb. 2010, pp. 259–264.
50. W. Du, F. Mieyeville, and D. Navarro, "IDEA1: A SystemC-based system-level simulator for wireless sensor networks," in Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on, June 2010, pp. 618 –622.
51. J. Haase, J. Molina, and C. Grimm, "High level energy consumption estimation and profiling for optimizing wireless sensor networks," in Industrial Informatics (INDIN), 2010 8th IEEE International Conference on, July 2010, pp. 537 –542.
52. Markus Damm, Jan Haase, Christoph Grimm, Fernando Herrera, and Eugenio Villar. 2008. Bridging MoCs in SystemC specifications of heterogeneous systems. EURASIP J. Embedded Syst. 2008, Article 7 (January 2008), 16 pages. DOI=10.1155/2008/738136 http://dx.doi.org/10.1155/2008/738136
53. Haase, J., Molina, J. M., & Dietrich, D. (2011). Power-aware system design of wireless sensor networks: Power estimation and power profiling strategies. Industrial Informatics, IEEE Transactions on, 7(4), 601-613.
54. "Transaction-Level Modeling," Open SystemC Initiative - Transaction-Level Modeling Working Group. [Online]. Available: http://www.systemc.org/downloads/standards/tlm20/
55. J. Aynsley, "TLM-2.0 Language Reference Manual," Open SystemC Initiative, Tech. Rep., 2009.
56. J. Haase, M. Damm, J. Glaser, J. Moreno, and C. Grimm, "Systemc-based power simulation of wireless sensor networks," in Specification Design Languages, 2009. FDL 2009. Forum on, sept. 2009, pp. 1 –4.
57. M. Damm, J. Moreno, J. Haase, and C. Grimm, "Using transaction level modeling techniques for wireless sensor network simulation," in Design, Automation Test in Europe Conference Exhibition (DATE), 2010, march 2010, pp. 1047 –1052.
58. J. Haase, M. Lang, C. Grimm, "Mixed-level simulation of wireless sensor networks" in Proceedings of Forum on Specification and Design Languages (FDL) 2010.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

59. J. Moreno, J. Wenninger, J. Haase, and C. Grimm, “Energy profiling technique for networklevel energy optimization,” in AFRICON, 2011, 2011, pp. 1–6.
60. J. Glaser, J. Haase, M. Damm, and C. Grimm, “Investigating power-reduction for a reconfigurable sensor interface,” in Austrochip. Institut für Elektronik - TU Graz, 2009, pp. 27–32.

AUTHOR PROFILE

Jan Haase received the Diploma degree in computer sciences at Goethe University, Frankfurt, Germany. He then became a Research Assistant at the Department of Technical Informatics, Frankfurt University. There he focused on the field of computer architectures, dynamic and distributed parallel computation, middlewares, and embedded systems, resulting in his Ph.D. thesis on the scalable, dataflow-driven virtual machine (SDVM).

Since 2007, he is a Senior Researcher and Project Leader at Vienna University of Technology. His research interests include hardware/software co-design, design methodologies for heterogeneous (AMS+HW/SW) embedded systems, wireless sensor networks, power consumption optimization, and automatic parallelization.

Dr. Haase is author and coauthor of more than 90 reviewed publications. He is currently appointed IEEE Region 8 Conference Coordinator. He is an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and of the IEEE Industrial Electronics Tech News and served and serves as Program (Co-) Chair or Member in several program committees of international conferences like IECON, AFRICON, DATE, DAC, FDL, CSNDSP, ICIEA, Austrochip, and smaller workshops.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)