Dipl.-Ing. Philipp Maria Glatz, Bakk.techn.

# Efficient Network Coding Middleware for Energy Harvesting Wireless Sensor Networks

Dissertation

vorgelegt an der
Technischen Universität Graz

zur Erlangung des akademischen Grades
Doktor der Technischen Wissenschaften
(Dr.techn.)

durchgeführt am Institut für Technische Informatik
Technische Universität Graz
Vorstand: O. Univ.-Prof. Dipl.-Ing. Dr. techn. Reinhold Weiß

Graz, im August 2011

## EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am .................................... ....................................

(Unterschrift)

## STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.................................... ....................................

date (signature)

## Kurzfassung

Drahtlose Sensor Netzwerke - Wireless Sensor Networks (WSNs) - sind aufgebaut aus kleinen eingebetteten Geräten, die kooperativ die von ihnen geforderte Anwendung implementieren. Bei diesen handelt es sich meist um manuell schwer durchführbare oder gefährliche Aufgaben, die autonom in entlegenen Gebieten abgearbeitet werden müssen. Die in solchen Szenarien entstehenden Kosten für den Austausch von Batterien und für Modifikationen vor Ort übersteigen meist die Kosten, die durch die Hardware selbst entstehen. Um solchen Kosten vorzubeugen werden ressourcenschonende Verfahren bezüglich Energie und Bandbreite benötigt.

Diese Verfahren gliedern sich typischerweise in Power-Aware Computing und Energy Harvesting. Power-Aware Computing für WSNs bedeutet, Daten möglichst effizient zu übertragen oder die Menge der Daten zu reduzieren um Energie zu sparen und den hohen Leistungsverbrauch des Funkmoduls möglichst selten notwendig zu machen. Energy Harvesting beschäftigt sich hingegen damit bereits umgesetzte Energiereserven wieder aufzufüllen. Ziel dabei ist es, die Belastung durch die Anwendung so anzupassen, dass ein durchgehender energieneutraler Betrieb möglich ist.

In dieser Arbeit liegt der Fokus auf energieneutralen WSNs durch den Einsatz von Energy Harvesting System (EHS) Technologie. In diesem Kontext werden Optimierungen der Netzwerkprotokolle für WSNs vorgestellt und integriert, die auf Network Coding (NC) beruhen. Lokales, autonomes, skalierbares und opportunistisches NC für WSNs wird eingeführt. Weiters werden notwendige WSN Services und Tools vorgestellt um die beanspruchte eigenständige Methodologie zu vervollständigen.

Der wissenschaftliche Beitrag dieser Dissertation ist dreierlei. Zuerst und vorrangig werden Lösungen zur Energieeinsparung für WSNs vorgestellt, die auf der Implementierung informationstheoretischer Optimierungen der Kommunikation beruhen. Diese für WSNs neuen Methoden werden unter Berücksichtigung starker Ressourceneinschränkung und Energy Harvesting entworfen und evaluiert. Weiters werden vielseitig verwendbare, effiziente und energetisch robuste Energy Harvesting Prototypen entwickelt, getestet und mittels Middleware-Zugängen integriert. Schlussendlich werden Neuerungen und Verbesserungen beim Entwurf von WSN Systemen vorgestellt. Diese beinhalten Middleware Services zur Lokalisierung, Virtualisierung von Messumgebungen, Toolchain-Entwicklung und Batteriesimulation.

# Abstract

Wireless Sensor Network (WSN) motes are wireless embedded devices supposed to solve a given problem or implement an application in a cooperative manner. In many cases WSNs are used for applications where it is not feasible to send people due to safety or reasons of cost. This results in the need to develop cheap, autonomous and long lasting systems. The lifetime requirement, in particular, is a stringent demand because maintenance costs for replacing batteries can seldom be afforded. The need for minimizing system costs and maximizing the lifetime of WSN systems most often leads to heavy resource constraints such as limited energy and bandwidth.

There are two main ways for dealing with these issues. First, power-aware computing techniques are optimized and employed in order to mitigate the drain of energy reservoirs. These techniques concentrate heavily on the motes' radio transceiver modules and the amount of data that they are required to send, because the radio is usually the most power-hungry component. Second, the battery-replacement problem can be tackled with approaches utilizing an Energy Harvesting System (EHS). Enhancing motes with EHSs may allow the replenishment of energy reservoirs and the employment of Energy Neutral Operation (ENO).

In this thesis, the primary focus lies on establishing an ENO-capable EHS platform for enhancing motes and a WSN middleware for optimizing the traffic on the network, given the context of using EHS-enhancements. A hardware prototype is built and many versatile middleware services are designed, implemented and evaluated. The main optimization methods that are used are different types of Network Coding (NC). Therefore, localized, autonomous, scalable and opportunistic NC are presented. Furthermore, WSN system services, development tools and techniques are introduced for finally achieving a complete and efficient design methodology.

The contribution of this dissertation thesis is three-fold. First, the theory of information flows is used to optimize wireless communication systems with NC. This thesis successfully implements the approach for heavily constrained WSNs. Novel WSN optimization methods and their detailed evaluation are presented. Second, an inexpensive and robust EHS-enhanced mote platform, called RiverMote, is implemented with remarkable energy-efficiency and Black-Out Sustainability. It is integrated with NC by means of middleware services. Finally, further aspects necessary for WSN system development have been improved and tested with case studies. These include middleware services for localization and hardware measurement virtualization as well as toolchain development and battery-aware simulation experiences.

## Acknowledgements

*First of all, I would like to express my gratitude to my supervisor Prof. Dr. Reinhold Weiß for his encouraging guidance and sage advice throughout my thesis and while being a university assistant at his department, the Institute for Technical Informatics at Graz University of Technology. He has provided me with all necessary means and facilities for diving into teaching and conducting research visiting various different topics. I also want to thank my colleagues at the department, especially Prof. Dr. Eugen Brenner, Prof. Dr. Christian Steger, Dr. Christian Kreiner, Karima Hein and Leander Hörmann for inspiring me, regarding both teaching and research duties. Also, I am indebted to Dr. Damian Dalton for his kind support and insightful feedback. Furthermore, writing this thesis would not have been possible without the help of and the collaboration with the staff members from the department Silvia Reiter, Kurt Haller and Engelbert Meissl. In addition to that, invaluable help through discussion and by implementing their projects has been provided by a number of students and colleagues working as teaching assistant and doing project and thesis work: Thomas Aichholzer, Michael Aichner, Reinhard Berlach, Qiang Chen, Marcus Dietl, Christoph Feil, Julia Gebetsberger, Stefan Gruber, Andreas Hartl, Andrea Höller, Philipp Hölzl, Leander Hörmann, Michael Irauschek, Patrik Moravek, Andreas Neuhold, Dino Odobasic, Markus Quaritsch, Christian Rathgeb, Christian Ritter, Stefan Ruff, Patrick Schrey, Lukas Sternbach, Dr. Michael Thonhauser, Daniel Wittibschlager and Martin Wohlfahrt.*

*This thesis would not have been possible without the employment and financial support for presenting scientific dissemination at conferences by the Institute for Technical Informatics at Graz University of Technology, by Graz University of Technology itself and by Land Steiermark. I especially want to thank my parents, Helga and Walter Glatz, who have supported me all throughout my studies, the employment as a university assistant and when writing this thesis.*

*I thank my family, my friends and especially my girlfriend Elisabeth Wurglits for their understanding, time, patience, and continuous encouragement in my endeavor for pursueing my work. Otherwise, I would not have been able to complete this thesis.*

Graz, August 2011 Philipp Maria Glatz

## Extended Abstract

Wireless Sensor Networks (WSNs) are expected to provide solutions for a class of problems related to pervasive computing. WSN motes - small form factor devices - are expected to cooperatively solve problems working in the background without actually altering contextual foreground. This paradigm is particularly apt for the demands of remotely deployed systems that must operate autonomously. In such a context, it is important to foster longevity - and therefore energy-awareness - in order to ensure cost-efficient applications. Otherwise, maintenance cost for operating and fixing a system or, even worse, mission cost for going to a remote location to replace batteries, is - in general - not economically feasible. These preconditions lead to constraints that hold true for almost any domain in which the broad field of WSN technology is applied. Wireless means that neither energy infrastructure nor other hardware support will be available. This leads to the constraint of limited energy, and it results in the need for the mote platforms to implement all functionality on their own. It further demands the employment of system-on-chips in embedded system architectures, which is well known for providing a complex design space where many different aspects have to be considered. Especially in the case of wirelessly networked systems of such types, the design, simulation, implementation and deployment is a challenging task. The need for cooperative behavior and the reliable gathering of data or receiving control information may heavily impact the available energy of the system. This is indeed, above all, the case because the radio transceiver is the main source of power dissipation and the wireless channel poses severe constraints on applying accurate simulation techniques beforehand when preparing a deployment.

For implementing such systems, one needs to utilize design tools which support the complete life cycle of a system. A special focus has to be put on energy-efficient wireless networking and power-aware hardware platforms. Furthermore, the required ability to implement and control the latter two aspects autonomously necessitates the prerequisite of implementing a middleware system. A middleware systems takes care of the operation at runtime and supplies necessary services to the end-user application. Such a distributed system layer can further be utilized for easing the development process and abstracting away from unwanted complexity. Especially, frequently used services and implementation parts that are concomitantly required for a large number of applications can be integrated to speed up the development process and make it less prone to errors.

Two efficient ways for handling the issues described above are to be elaborated upon in greater detail and are then applied in this thesis' context. First, power-aware computing techniques are optimized and employed for mitigating the drain of energy reservoirs. These techniques concentrate heavily on the motes' radio transceiver modules and the amount of data they have to send, because the radio is usually the most power-hungry component. Also, unnecessary idle times of this and other modules have to be avoided. Second, the

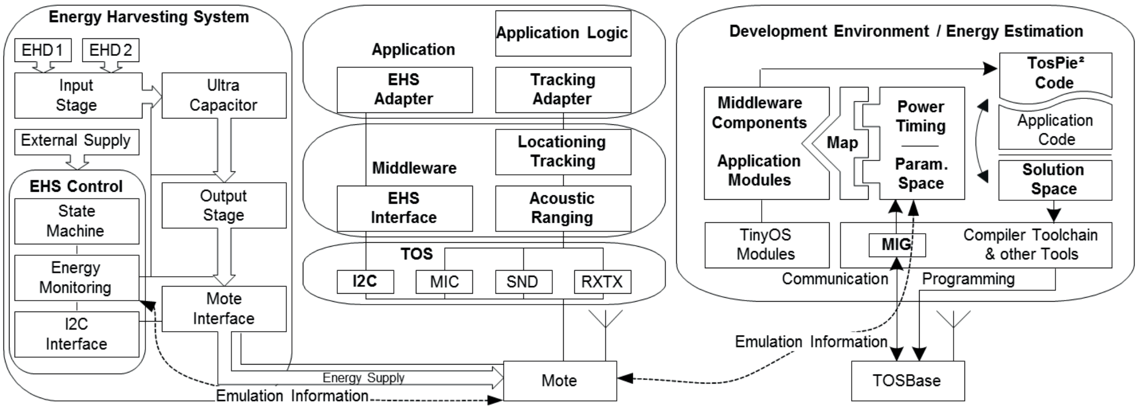



Figure 0.1: This figure depicts the main aspects of WSNs that are dealt with in this thesis. In general, the TOSPIE2 methodology focuses on Network Coding (NC) and energy harvesting middleware. It also includes hardware prototype implementations and related essentially needed WSN services as well as tools and methods for the WSN development life cycle.

battery-replacement problem can be tackled with approaches using an Energy Harvesting System (EHS). Using an EHS may allow the replenishment of energy reservoirs and the employment of Energy Neutral Operation (ENO) modes. In ENO mode, motes are able to support a constant end-user performance through harvesting environmental energy.

Tackling energy-aware design approaches with energy-efficient communication services and prototyping EHS-enhanced motes with a tightly coupled integration by means of middleware services, demands a broad perspective when choosing the development approach. Figure 0.1 outlines parts of the Tiny Operating System Plug-In with Energy Estimation (TOSPIE2) [1] project and it depicts what information is needed from which parts of a system at different times of the development life cycle. The necessity of integrating information that is relevant for the operation and optimization of a WSN deployment at a single instance requires a middleware system. A WSN middleware can be used for operating the actual hardware, implementing a networking stack and offering services to software running on motes and, all together, it may offer interfaces to an end-user or development host. WSN middleware systems might also be used for autonomously putting a mote or network in the loop and profiling a deployment or injecting knowledge from host-side modeling, simulation and emulation. While offline modeling and simulation are indispensable, the possible design space cannot completely be covered before the actual deployment. So, one of the main features of the toolchain and development setup that has been established by TOSPIE2 is emulation support. The profiling of system characteristics under given conditions and the evaluation of real hardware including the direct comparison to a database holding simulation characteristics can greatly reduce the complexity of the design space. It therefore supports design methodologies that are less prone to errors and misleading assumptions [2]. The TOSPIE2 methodology is concerned with making experiments repeatable for achieving more expressive results. This includes the separate test of non-deterministic parts like asynchronous protocols and channel behavior. The methodology also features component-wise testing of the system as well as the inclusion of interaction of parts that may be related to the same effects. For optimizing NC, energy harvesting, localization and Multi-Application MiddlewAre (MAMA) services, the expressiveness of simulation environments under use for given optimizations are regularly compared with results from hardware measurements.

Summing up what is discussed above, this thesis' contribution to answering research questions can be formulated. The main contribution claim is to answer research questions regarding NC services and energy harvesting hardware with their respective integration in a middleware system. Side aspects that are necessary for achieving energy-aware design methodologies, will be covered by additional contribution and evaluated with case studies. Additional contribution and the case studies are on (i) the design, implementation and evaluation of a multi-application Virtual Machine (VM) based middleware and a (ii) battery performance profiling and modeling approach with integration into a power-aware simulation environment.

TOSPIE2 is built around a power state database and distributed instances of an Eclipse plug-in including automated install routines for the plug-in and for WSN middleware and software development toolchains. It provides interfaces to a flexible, yet accurate, hardware measurement setup, a modified power profiling simulation environment and a proprietary EHS simulation in Matlab.

The power state database holds power and timing values that are necessary for describing a platform's power states and their transient effects. All data from the database can be written and read from all master and client components in the system. The simulation environment and the measurement setup are extended with debugging output for precise timing measurements of when power states are switched. This setup allows the exact determination of mote Power State Models (PSMs), Energy Harvesting System Efficiency Models (EEMs) and non-deterministic protocols' effects can be characterized.

The graphical user interface can be used for comparing simulation and measurement results as well as for tuning PSMs with the National Instruments setup that employs a digitizer for high speed measurements and a data acquisition card for flexible measurement setups.

Besides at least one master, there can be an arbitrary number of clients in TOSPIE2, utilizing the same PSMs and simulation or measurement setups. Consequently, TOSPIE2 offers the opportunity to virtualize development tools and it provides the means for accessing them via the internet. For doing so, one can choose to use an existing client where automated installation procedures are available or - as the system is designed modularly with the use of an open architecture - novel types of clients can be designed and implemented to communicate with the Simulation and Measurement Interfacer (SiMeInterfacer). Remotely operating virtualized measurement setups may be beneficial for cost-saving for its user. The setup need not be operated manually and interference on the wireless channel, caused by people being present, may be greatly reduced. The SiMeInterfacer has an open architecture that can further be extended with passive and active components.

The main contribution of implementing NC exploits the inherent overhearing of messages that are transmitted via the wireless channel. Different approaches for energy-efficiency from message conservation are implemented by NC means for WSNs. These methods include localized [3], autonomous [4], scalable [5] and opportunistic NC [6] taking into consideration side-effects to other networking approaches and interplay with other energy-efficient and low-power techniques as well. The techniques that are designed, implemented and tested with WSN hardware are novel for this kind of technology. Scalable Network Coding (SNC) extensions for implementing energy management [7] and embedding it into multi-path routing environments [8] are covered as well. NC service features include robust, seamless application integration and generic applicability characteristics. The contribution claim dealing with efficient EHS-enhanced WSNs, addresses low-power hardware platforms supplied by efficient and also low-power EHS architectures for enabling ENO. Different EHS architectures, their measurement and characterization [9] as well as testing distributed performance-aware tasking are implemented and evaluated. An efficient EHS-enhanced mote called RiverMote [10] is implemented with high Black-Out Sustainability (BOS) at low cost with the established design methodology [11]. It implements a river level monitoring service that can be used for flood warning systems. Increased cost-efficiency and robustness are achieved by utilizing back-up procedures with fail-safe states and efficiently designed energy storage architecture together with its BOS of up to more than three weeks [12].

The first case study implements a business model for WSN resource sharing by means of hardware virtualization. The implementation of a flood warning system together with other applications is taken as a use case. Scientific dissemination is found in a middleware offering suitable interfaces for running multiple applications concurrently [13]. For the sake of completeness further necessary system services are integrated as well. While RiverMote implements outdoor localization with Global Positioning System, [14] implements, evaluates and tunes an accurate power-aware indoor localization system for WSNs. Furthermore, the overhead of NC and EHS service integration in a single middleware system is elaborated in detail [15]. Back to the MAMA, it not only allows running multiple applications on a single mote and therefore the implementation of multiple Virtual Organizations in one single WSN. It is also extended with a multi-agent VM approach which is introduced for providing multi-hop Over-the-Air Programming and load balancing features. The load balancing power management console utilizes agent migration to respond to the load that has qualitatively been measured by the agents themselves. The optimization metric can include the number of messages that are sent or the time that an agent is active on a given mote.

This is further related to a case-study on battery performance profiling and modeling [16]. It allows modeling the actual power dissipation of a deployed system either by emulation means or by the simulation of models that cannot be emulated in real-time with hardware. Battery performance, in particular, cannot be emulated. Therefore, performance traces of different battery load modulation are taken with the hardware measurement setup [17] such that models can be built in ways, expressive enough, to describe a greater number of possible load patterns. These models are then trained and the most appropriate choices for given types of load characteristics (e.g. described by the coarse approximation of average power dissipation and intermittent relaxation times) are implemented with an interface to the power simulation environment. Results of these models and their transformations are applied for being implemented for battery-aware simulation and in-situ runtime battery-modeling for embedded platforms. This allows quickly profiling mote power dissipation compared to battery performance for determining the current status of a mote's energy budget and balancing the load across different motes in a WSN with the power management console.

# Contents


**1 Introducing Wireless Sensor Networks 1**

1.1 Motivation for Energy-Efficiency and Power-Awareness . . . 3
1.1.1 Maintenance Cost versus Hardware Cost . . . 3
1.1.2 Power-Aware Networking for Wireless Sensor Networks . . . 3
1.1.3 Energy Harvesting Systems for Energy Neutral Operation . . . 4
1.1.4 Design Techniques and Related Services . . . 4
1.2 Efficient WSNs: TOSPIE2 Project and Contribution Claim . . . 5
1.2.1 Contribution of this Thesis . . . 7
1.2.2 Thesis Organization . . . 8

**2 Related Work and State-of-the-Art 9**

2.1 Wireless Sensor Network Applications and Systems . . . 9
2.1.1 Wireless Sensor Network Middleware and Operating Systems . . . 10
2.1.2 Operating Systems for Wireless Sensor Networks . . . 10
2.1.3 Wireless Sensor Network Middleware . . . 11
2.2 Energy Harvesting Technology and Systems . . . 12
2.2.1 The Maximum Sustainable Duty Cycle . . . 12
2.2.2 Energy Harvesting Aware Routing . . . 14
2.2.3 Energy Harvesting System Storage and Component Efficiencies . . . 14
2.3 Network Coding for Wireless Sensor Network Communication . . . 15
2.3.1 Different Ways of Applying Network Coding . . . 16
2.3.2 Lowering Networking Complexity with Network Coding . . . 16
2.3.3 The Network Coding Advantage for Wireless Sensor Networks . . . 17
2.4 Methods and Tools Tackling WSN Technology, Design and Optimization . . 17
2.4.1 Industry Platforms for Modeling, Simulation and Implementation . . 18
2.4.2 Battery Models and Applications to Wireless Sensor Networks . . . 19
2.5 Summing up Related Work and State-of-the-Art . . . 20

**3 Concept and Interaction of Novel Comprehensive Methodologies in TOSPIE2 23**

3.1 Network Coding Middleware Services for Wireless Sensor Networks . . . 23
3.1.1 Localized Network Coding in Wireless Sensor Networks . . . 25
3.1.2 Autonomous Network Coding in Wireless Sensor Networks . . . 26
3.1.3 Scalable Network Coding in Wireless Sensor Networks . . . 26
3.1.4 Extensions for Multi-Path Routing and Energy Management . . . 26
3.1.5 An Integrated Opportunistic Network Coding based Middleware . . 27
3.2 Energy Harvesting for Wireless Sensor Networks . . . 27
3.2.1 Energy Harvesting Maximum Sustainable Duty Cycle . . . 27
3.2.2 RiverMote Prototype Implementation . . . 27

3.3 Application Scenarios for Virtual Organizations and Energy Harvesting . . 29
3.3.1 Virtual Organizations for Wireless Sensor Networks . . . . . . . . . 30
3.3.2 Power-Aware Indoor Localization and Simulation Expressiveness . . 30
3.4 The TOSPIE2 Toolchain and its Battery Awareness Case Study . . . . . . 31
3.5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

**4 Evaluation and Results of Case Studies and TOSPIE2 Methodologies 35**
4.1 A Multi-Agent Middleware for Wireless Sensor Networks . . . . . . . . . . . 35
4.2 The TOSPIE2 Setup and its Components . . . . . . . . . . . . . . . . . . 37
4.3 Extended Results Summary . . . . . . . . . . . . . . . . . . . . . . . . . 41

**5 Conclusion and Future Work 43**
5.1 Concluding Thesis Scope and Completeness . . . . . . . . . . . . . . . . . 43
5.2 Possible Future Directions . . . . . . . . . . . . . . . . . . . . . . . . . 44

**6 Contributed Publications 47**
6.1 Network Coding Middleware Services for Wireless Sensor Networks . . . . . 50
P.1.1 PM2HW2N'09 - LINDONCS: Localized In-Network Detection Of Network Coding Structures in Wireless Sensor Networks . . . . . . . 50
P.1.2 ICT'11 - Implementing Autonomous Network Coding for Wireless Sensor Network Applications . . . . . . . . . . . . . . . . . . . . . 58
P.1.3 ISPAN'09 - Energy Conservation with Network Coding for Wireless Sensor Networks with Multiple Crossed Information Flows . . . . . . 64
P.1.4 CNSR'11 - Opportunistic Network Coding for Energy Conservation in Wireless Sensor Networks . . . . . . . . . . . . . . . . . . . . . . 70
P.1.5 MICC'09 - A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks . . . . . . . . . . . . . . . . . 78
P.1.6 IJAACS - Efficient Network Coding Middleware for Energy Harvesting WSNs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84
6.2 Energy Harvesting for Wireless Sensor Networks . . . . . . . . . . . . . . . 103
P.2.1 WD'08 - A Measurement Platform for Energy Harvesting and Software Characterization in WSNs . . . . . . . . . . . . . . . . . . . . 103
P.2.2 WCNC'11 - Designing Sustainable Wireless Sensor Networks with Efficient Energy Harvesting Systems . . . . . . . . . . . . . . . . . 108
P.2.3 EJSE - Designing Perpetual Energy Harvesting Systems explained with RiverMote: A Wireless Sensor Network Platform for River Monitoring . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
P.2.4 EW'11 - Low-Cost Reliable Blackout Sustainability of Wireless Sensor Networks with Energy Harvesting Systems . . . . . . . . . . . . 124
6.3 Application Scenarios for Virtual Organizations and Energy Harvesting . . 132
P.3.1 ICT'11 - MAMA: Multi-Application MiddlewAre for Efficient Wireless Sensor Networks . . . . . . . . . . . . . . . . . . . . . . . . . . 132
P.3.2 PM2HW2N'10 - Design, Simulation and Measurement of an Accurate Wireless Sensor Network Localization System . . . . . . . . . . 140
P.3.3 ICT'11 - HANS: Harvesting Aware Networking Service for Energy Management in Wireless Sensor Networks . . . . . . . . . . . . . . 148

P.3.4 PerSeNS'10 - A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency . . . 154
6.4 The TOSPIE2 Toolchain and its Battery Awareness Case Study . . . 160
P.4.1 IPSN'10 - TOSPIE2: Tiny Operating System Plug-In for Energy Estimation . . . 160
P.4.2 IGI - Architecturing Resource Aware Sensor Grid Middleware: Avoiding Common Errors in Design, Simulation, Test and Measurement . 162
P.4.3 CRC - Towards Modeling Support for Low-Power and Harvesting Wireless Sensors for Realistic Simulation of Intelligent Energy-Aware Middleware . . . 182

**7 Acronmys 209**

**8 Bibliography 211**

# List of Figures

0.1 The Tiny Operating System Plug-In with Energy Estimation (TOSPIE2) . v

1.1 Thesis research context . . . 2
1.2 Design, hardware and maintenance cost . . . 3
1.3 TOSPIE2 project overview . . . 6

2.1 Related fields of research . . . 10

3.1 Detailed overall concept architecture . . . 24
3.2 Network Coding middleware components and services . . . 25
3.3 Energy harvesting prototypes . . . 28
3.4 Case study: RiverMote and Multi-Application MiddlewAre scenario . . . 29
3.5 Multi-Application MiddlewAre and services . . . 31
3.6 Design guidelines and battery simulation . . . 32

4.1 Power management console workload balancing . . . 36
4.2 TOSPIE2 simulation of an agent running on top of MAMA . . . 37
4.3 TOSPIE2 component connections . . . 38
4.4 The TOSPIE2 power state database . . . 38
4.5 The TOSPIE2 hardware measurement setup . . . 39
4.6 The TOSPIE2 Eclipse plug-in user interface . . . 39

6.1 Structure of contributed publications . . . 47

# 1 Introducing Wireless Sensor Networks

The concept of Wireless Sensor Networks (WSNs) stems from the vision of smart dust [18] being a platform for realizing pervasive and ubiquitous computing expected to autonomously be operated by middleware systems [19]. Working on achieving these characteristics since the advent of WSNs more than a decade ago in the early beginnings of the $21^{st}$ century, two main characteristics of WSNs and their developments can be observed.

First, this field of research has continuously been facing issues when it comes to actually building and deploying systems for real world scenarios. The difficulties that may arise when implementing these wireless, embedded, distributed systems - and the need for coping with these difficulties - have caused a large number of articles being written in a very technical flavor especially commenting on practical considerations of real world experiments and deployments. Therefore, when talking about WSN research, the scope is usually not limited to the optimization of systems and their operation including optimization of the hardware, software and networking in terms of performance, power-awareness, reliability and adaptiveness as well as more abstract considerations including mobility, application-level needs and pervasiveness [20]. Also, a great deal of effort is put into the development and optimization of modeling, simulation and measurement tools which support the above mentioned needs. Doing so demands taking a broad perspective when reasoning about WSNs as systems. Neglecting that demand, when optimizing these embedded platforms [21] which often mainly boils down to power-awareness and reliability measures, may either keep single findings and their discussion from being expressive and applicable, or it may render practical deployments inefficient or useless. Perhaps the most well-known example being used to discuss that need taking many aspects into account for fostering reliable and efficient deployments is presented in [22].

Second, in addition to implementing many monitoring and automation flavored applications, the research community dealing with WSNs has ever been searching for a killer application ensuring an explosive upsurge in the technology. However, despite enabling factors like energy harvesting technology and Network Coding (NC) for WSNs, the initial vision of smart dust has not yet become true to ambiently pervade our everyday lives. Of course, a number of applications that rely on WSNs exist today. But still, WSNs are more often used as a playground for exploring ideas to solve resource constrained settings in related technologies and scientific fields. A good example can be found in smartphone technologies that can be considered to be part of a sister field of WSN research, that is Mobile Ad-Hoc Networks (MANETs). MANETs are usually composed of larger and more complex platforms running more versatile services. The relaxed constraints and increased functionality is reflected by the complexity of MANET middleware systems, that are rather subject to service orchestration than to service integration and communication alone as it is most often the case for WSNs. However, smartphones, in particular, experience similar power-awareness issues like WSNs. And finally, after nearly a decade of WSN research utilizing low-power techniques for WSNs, they are also more often implemented

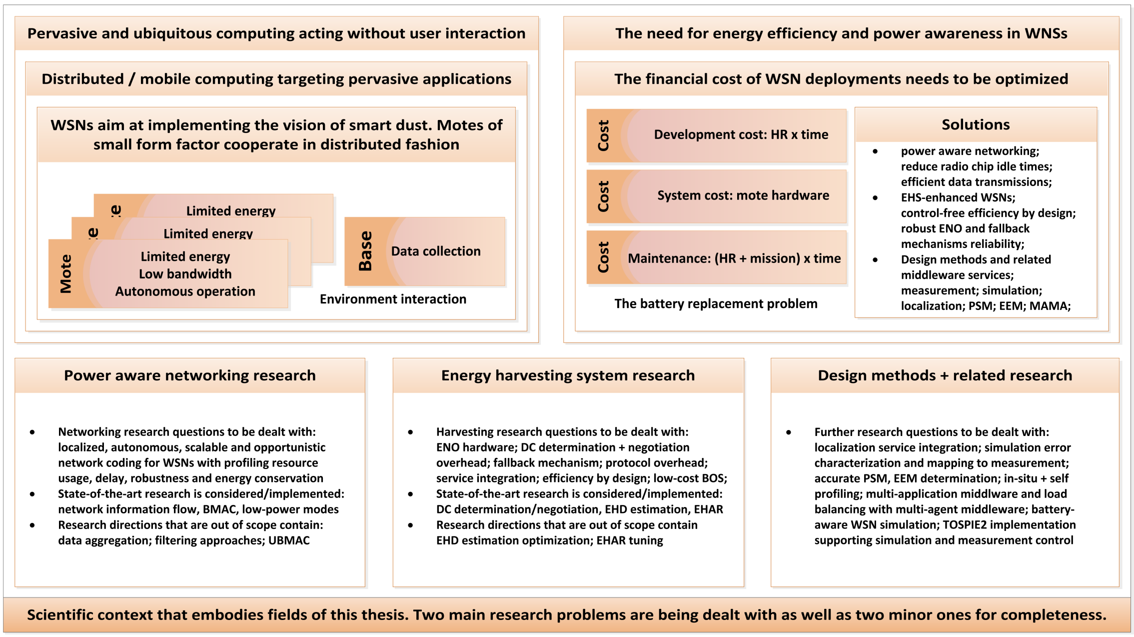



Figure 1.1: WSN motes suffer constraints in terms of limited energy and bandwidth with an inherent need to operate autonomously. Resulting research questions are embedded in and applicable for a wide scientific context.

in smartphones nowadays. Maybe, the situation can best be described with an excerpt of the *Coding for Life* speech of Jeffrey Sharkey at the Google I/O Developer Conference 2009:

*"The three most important considerations for mobile applications are, in order: battery life, battery life, and battery life."*

From that year on, many of in-WSN-explored technologies for energy conservation have become reality for smartphone users as well. So, despite the lack of a killer application within the field of WSNs, research in that area provides invaluable results, especially, because there is a field for researching optimization methods in the context of cost-efficiency in resource-constrained settings. In that sense, achievements that are made with WSNs can be beneficial for other domains as well.

The two characteristics mentioned above deserve consideration in different aspects of this thesis. First, completeness of different approaches and their applicability will be supported by always considering side-effects among them and specifically adding essential related WSN middleware services and measurement or simulation approaches including their characterization. Second, these different aspects that are to be discussed rely on similar levels of detail or abstraction. Their possible impact on other technologies and related future work needed for realizing that is discussed in this thesis' reminder in Chapter 5.


Figure 1.2: WSN application costs occur at different stages of the project life cycle.

## 1.1 Motivation for Energy-Efficiency and Power-Awareness

When designing WSN software and hardware, a number of different aspects and their relationships have to be taken into account. However, for following a structured methodology, this thesis employs a top-down approach, starting from overall system and development cost in Section 1.1.1, to explore what generic needs exist in virtually any WSN and deduce necessary measures that have to be taken to counteract them. So, in order to not get lost in the broad scientific context outlined above and depicted in Figure 1.1, this chapter clearly outlines how tuning application performance and keeping costs low necessitates the work shown in this thesis. It then deduces a contribution claim that covers these issues with its scientific dissemination.

### 1.1.1 Maintenance Cost versus Hardware Cost

Many optimizations can be related to the conservation of cost during development, for the physical system and its components themselves and costs of deployment and maintenance as depicted in Figure 1.2. While development costs can mainly be reduced with suitable design and simulation tools, the system, deployment and maintenance costs are dominated by the cost of human resources and technical equipment aside for WSN motes alone. Therefore, the actual cost of the usually not so expensive hardware is less of a concern compared to the mission cost for deploying a system or on-site fixing of problems. This leads directly to the need for (i) power-aware networking as radio transmissions are the main source of power dissipation and (ii) Energy Harvesting Systems (EHSs) to replenish drained energy budgets. Finally, for means of developing such systems and means of reasoning about them, expressive models have to be found and tested with suitable simulation and measurement tools. Middleware systems can be used to operate efficient networking and harvesting schemes according to relationships that are modeled and profiled prior to deployment or in offline environments.

### 1.1.2 Power-Aware Networking for Wireless Sensor Networks

For reducing the average power dissipation from networking, the radio transceiver activation times - just like any other component on a mote - have to be optimized subject to first reducing idle times and second being efficient when active. Reducing idle times of a radio module means utilizing efficient Media Access Control (MAC) protocols like Low-Power Listening (LPL), while efficiency may mean the conservation of energy by implementing information theoretical optimization for bandwidth optimization. The min-cut max-flow theorem in [23] dictates the maximum throughput that can be achieved with NC. It is called NC advantage. For wireless NC it is bounded by two. Novel methods for incorporating NC for WSNs are to be presented, considering side-effects with other optimizing protocols. LPL, that may heavily interact with NC, deserves detailed considerations. This includes configuration, evaluation and side-effect discussion, but it excludes research for options other than the Berkeley MAC protocol that is being considered here. Other options for power-aware networking, apart from information theory and LPL, would be approaches like data aggregation, filtering or compression in general [24] and Uncertainty Driven MAC or related complex clock drift estimation. Research and optimization of these and similar approaches goes beyond this thesis' scope.

### 1.1.3 Energy Harvesting Systems for Energy Neutral Operation

As a next step after power-aware operation alone, using EHS-enhanced WSNs may enable Energy Neutral Operation (ENO). Robust autonomous ENO may virtually solve any battery replacement problem and cut out maintenance cost. The main challenges for designing and operating EHSs are to achieve efficiency and robustness. This thesis contributes to efficiency-by-design approaches for EHS-enhanced WSNs and models concepts for describing virtual energy management and the interaction with other services and especially NC. The models and considerations incorporate and discuss state-of-the-art algorithms for Energy Harvesting Device (EHD) power profile estimation as well as distributed Duty Cycle (DC) negotiation. Furthermore, this thesis shall contribute to increasing robustness through providing means for low-cost high Black-Out Sustainability (BOS). Also, a novelty that can be found in this thesis lies in taking an integrated perspective when reasoning about NC and EHS-enhanced WSNs. A connection of these two topics is established with applying Opportunistic Network Coding (ONC) to WSNs that are negotiating DCs for ENO and Energy Harvesting Aware Routing (EHAR) mechanisms. State-of-the-art EHD estimation and DC negotiation options are considered as well. Characterization and measurement of an EHS are part of the case studies that are outlined in Section 1.1.4.

### 1.1.4 Design Techniques and Related Services

While the highest level of detail is to be given for NC and EHS elaborations, there is scientific contribution to other specific aspects that are vital for WSNs as well.

Localization methods are necessary for WSN deployments where motes do not a-priori know their position and manually programming each mote's position exceeds the acceptable workload. Most WSN data would be useless without position information, putting it into the right context. Furthermore, the opportunity of designing an indoor localization system is taken for getting the mapping of simulation results to theory right.

As outlined before, there may be inconsistencies between simulation and measurement results that make it difficult to tune applications without actually experimenting with hardware. This does not only hold for functional constraints as described for the localization problem. It is especially true for profiling energy conservation optimization methods.

Within the Tiny Operating System Plug-In with Energy Estimation (TOSPIE2) project, a design tool is developed including accurate hardware measurement setups and simulation environments. The combination in a single framework allows the determination of simulation and modeling errors along with possibilities to tune the expressiveness of single TOSPIE2 services by means of incorporating knowledge gained from experiments with other services.

For the hardware measurement itself, there are mainly two approaches available. First, a hardware platform built for in-situ and self-profiling of EHD profiles and power dissipation is discussed. Second, an accurate hardware measurement setup is implemented for profiling Energy Harvesting System Efficiency Models (EEMs) or Power State Models (PSMs) of WSN motes.

Two book chapters contribute to practical aspects of an energy-efficient design and life cycle of WSNs. One of them also gives an outline of how to design, evaluate and simulate battery models together with WSN applications for achieving more realistic results.

A multi-agent middleware built upon a Multi-Application MiddlewAre (MAMA) shall be the last service-oriented aspect and concludes with the last spectral line of this thesis' scientific contribution. First, it enables the mechanism of Virtual Organizations that is known from the field of grid computing. Multiple applications may share the same distributed hardware at the same time. The middleware takes care of mapping applications to actual hardware in the network. Also, there is usually some sort of migration technique for moving over different instances of the actual hardware in the network. Second, this very abstract way of cost and energy conservation by concurrently using virtualized hardware is also a possible case study for having its power management console control load distribution. This may incorporate information from battery-aware simulation into its metrics to allow for automated load balancing by the power management console.

## 1.2 Efficient WSNs: TOSPIE2 Project and Contribution Claim

Designing efficient WSNs and implementing means of optimization and interfaces for gathering needed information is accomplished under the TOSPIE2 project umbrella. This non-commercial project aims at answering research questions and overcoming technical hurdles related to hardware and software design. It provides support for integrated modeling, simulation and measurement environments. The separation of offline and runtime mechanisms, profiling and data integration as well as testing and model adaptation on harvesting platforms, motes alone or on host-based development systems is depicted in Figure 1.3.

The setup shall allow the addressing of several issues that are apparent in many current WSN design methodologies:

- Many frameworks exist that base their reasoning on simulation and theoretical results alone. However, there are a number of reports that show how unexpected changes in environmental conditions or hard-to-model channel variation may impact system performance when it comes to running applications on actual hardware instead of profiling it in simulation alone. In addition, modeling optimization strategies may give directions for improving a given mechanism, but current approaches may usually not reliably quantify actual effects. Here, as a design rule, all work

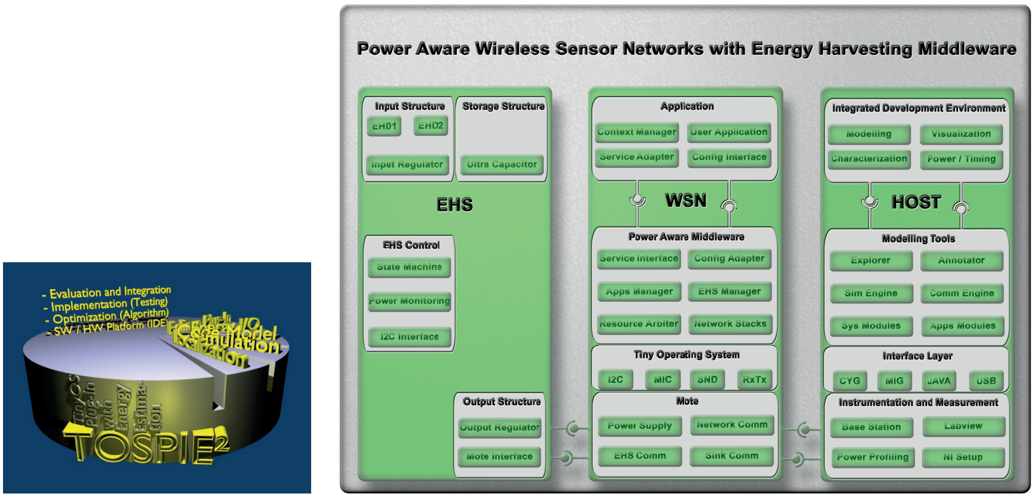



Figure 1.3: The TOSPIE2 project deals with different levels of abstraction at different stages of the WSN life cycle. Offline design tools feed design space exploration results back into the network controlled by optimized NC and EHS middleware services. EHS and hardware measurements return energy-efficiencies and power profiles to the host environment.

that is done within the context of TOSPIE2 tries to probe accuracy of modeling and simulation with comparison to experiments on actual hardware.

- WSNs may come in different types and the community lacks generically applicable cost and performance metrics. The idea behind optimizations in TOSPIE2 is to be as generically applicable as possible. Evaluation runs are set up in such a way that worst case scenarios can be run and tests can be repeated with different platforms and in different contexts. This is especially reflected by full overhearing in all networking tests allowing the congestion of the channel and by overhead-free options for middleware services and host-side tools.

- WSNs are wireless embedded systems that may need to be operated under given real-time and reliability constraints. Therefore, delay and packet loss traces from simulation and hardware runs need to be compared. It tells a lot about how applicable a simulation tool is for profiling WSN networking optimizations. Furthermore, embedded real-time systems should not be operated too close to their peak performance or full utilization of available memory. This may render the system unstable to unforeseen conditions and enhancements of implementations may be almost impossible. Therefore, traces of memory consumption of actual implementations with the Tiny Operating System (TinyOS) toolchain and profiles of timing behavior of hardware runs compared to simulation runs should be taken. TOSPIE2 provides the means to precisely annotate times when a switch in the PSM occurs. This is supported for simulation as well as for hardware experiments.

- Running cost-effective EHS-enhanced WSNs in ENO may result in very low DCs. Despite careful design approaches of such systems, changes in environmental conditions or programming errors may lead to an unstoppable decrease of energy and result in a reset. Therefore, more control-free approaches that are efficient by design are less prone to errors when an operating point is lost. Furthermore, long BOS times and defined reinitialization procedures after resets may relax dependability constraints of such systems. The RiverMote architecture, being designed under the TOSPIE2 umbrella, adheres to these rules and succeeds with stable ENO conditions in its hardware tests.

- Many optimizations at the networking layer are only applicable to a limited set of radio transceiver modules, end-user applications or specific use cases and settings. While the optimizations within the non-commercial TOSPIE2 project could have been chosen freely, because no specific constraints had been given with respect to how the to-be-developed middleware had to be optimized, a general information theoretical optimization has been chosen. The resulting NC approaches are tested under the design rules given above and show capabilities for providing energy conservation under a variety of different settings while being applicable to and portable across different systems.

### 1.2.1 Contribution of this Thesis

The primary contribution with power-aware networking and energy-efficient platforms is extended with additional contribution from case studies on middleware services and toolchain integration. All contribution of the thesis that is outlined here has undergone a scientific review process. It has either been published or it has been accepted for future publication.

1. An energy-efficient middleware is developed. A number of services are designed, implemented and tested. The main contribution consists of Localized Network Coding (LNC), Autonomous Network Coding (ANC), Scalable Network Coding (SNC), energy management with NC, harvesting and DC integration and ONC services being applicable to WSNs. For a variety of systems and communication paradigms the NC approaches can conserve energy from reducing the number of messages that have to be sent.

2. Energy-aware middleware services can only be useful if they are applicable to energy-efficient hardware platforms. An EHD-profiling EHS is discussed and an ENO-capable EHS-enhanced WSN platform - named RiverMote - is developed. The platform supports efficiency by design, high low-cost BOS and fall-back mechanisms applicable for implementing a river-level monitoring case study. Profiling EHS middleware service side-effects and NC interaction confirms interoperability.

- A power-aware indoor localization system service for WSNs is developed using localized power-aware algorithms. As another vital WSN middleware feature, MAMA allows implementing VOs on WSNs using a multi-agent system with load-balancing controlled by a power-management console. Implementing Virtual Organizations (VOs) on EHS-enabled WSNs with Over-the-Air Programming (OtAP) completes the river-level monitoring business case for resource and service sharing of EHS-based platforms.

- The TOSPIE2 Eclipse plug-in allows control of the power-aware and energy harvesting simulation and measurement setup. An accurate hardware measurement setup is established for accurate PSM and EEM characterization and provides a database for saving characteristics when using the virtualized environments. The latest feature includes battery performance characterization and modeling being interfaced to WSN simulation with applicability for the power management console.

### 1.2.2 Thesis Organization

This chapter introduces this thesis' scientific context, scope and contribution. Chapter 2 surveys the state-of-the-art of related methods, technologies and tools for establishing background to the topics and for providing opportunities to compare outcomes of this thesis' implementations. Figures 2.1 and 3.1 depict the fields of research related to this thesis and give a detailed overview on the subcomponents that make up these topics. Sections in Chapter 3 map publications to those topics and express their relation to the contribution claim given above. That chapter is dedicated to relating this thesis' novel methodologies and achievements to state-of-the-art related work and to discussing how it finally makes up the TOSPIE2 project. For providing more comprehensive results of the case studies than what has been published so far, Chapter 4 provides examples and results that go beyond what can be found in the publications in chapter 6. Finally, Chapter 5 identifies the significance of the approach presented in this thesis. After outlining its completeness possible future work is discussed.

# 2 Related Work and State-of-the-Art

During the process of publishing material for writing a dissertation thesis, a lot of new and exciting ideas are developed by the scientific community. Accounting for that fact, this chapter shall provide information on relevant material in two ways. First, all sections - dealing with their different topics - discuss scientific publications that the publications shown in Chapter 6 are based upon. This is necessary for understanding their contribution and how they work. Second, the latest results of research at the time of writing of this thesis are taken into account. This gives the opportunity to analyze concurrent developments in the field and to compare solutions from this thesis - albeit some time may have passed since their publication - with those that can be found in other publications represented by the latest state-of-the-art.

First, this chapter will give a very brief introduction to WSN technology and applications. Up to that point, discussion on development tools and methodologies will be left out of scope until the two main topics, NC and EHSs, have been covered in subsequent sections. This allows the introduction of related and state-of-the-art methods and tools, specifically considering aspects for the development of NC and EHS-enhanced WSNs.

Figure 2.1 gives an overview of the different topics that are covered and where they are situated in the WSN system architecture. Bottom-up and from left to right, energy harvesting hardware is the basis for middleware services and WSN applications. It also integrates hardware measurement support and the overall topic of WSN motes and systems is completed with tools and methods spanning different tiers of the architecture. The major topic of WSN middleware is split into different services that are being implemented. Only a brief, general overview of WSN Operating System (OS) and middleware architectures is to be given in Section 2.1. The main middleware parts, however, will be scattered across different sections on networking in Section 2.3 and related EHS topics in Section 2.2. The latter one especially considers the interplay and side-effects of using middleware services together with utilizing EHS optimizations. Targeting these two aspects in parallel is a novel methodology in this thesis making these two topics the most interesting fields of research in that context. Related services as well as design methodologies and tools, including simulation and measurement approaches that are inherently needed for tackling efficient WSN middleware implementations, are to be outlined in Section 2.4.

## 2.1 Wireless Sensor Network Applications and Systems

A recent book on WSNs [25] and an upcoming book chapter [16] give a good overview of past and current WSN technologies and issues. [25] presents an overview of several different mote architectures and is the follow-up publication to one of the most cited initial surveys [20] in the field. An overview of application domains and existing deployments is drawn, that targets the same type of WSNs that are thought of when using the term WSN in this thesis. The term is used in the sense of low data rate and static WSNs rather

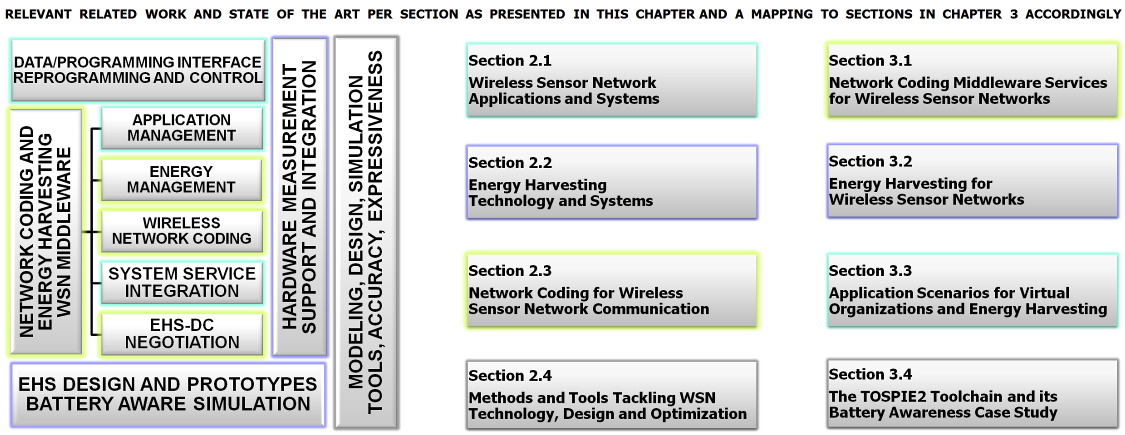



Figure 2.1: Middleware services span different levels from hardware to application across the networking stack. Simulation and measurement is necessary across application, NC and EHS layers as well. Interaction with low-level energy-efficiency approaches and effects for applications running in the WSN have to be considered.

than multimedia or mobile platforms. The type of WSNs considered here, can be found especially in environmental and structural health monitoring. But, it has also made its way into health systems, home automation, military applications and industrial settings. Recent standardization efforts [26], in particular, have pushed the field towards reliable industry-strength applications [27].

Most of the platform-dependent aspects in this thesis draw comparisons to two of the most common mote architectures of the last decade, the Mica2 [28] and the TelosB [29]. More recent developments that are used by the community include the CC430 [30] from Texas Instruments, the Atmel ATmega128RFA1 [31] and similar products. This continues a move towards single module platforms and it continues the trend of using these two suppliers as correctly pointed out by [32] which heavily impacts OS and middleware design.

### 2.1.1 Wireless Sensor Network Middleware and Operating Systems

Designing WSN software architectures with specific embedded system architectures in mind heavily influences them fitting especially these architectures' needs. But, saving cost and effort of making software portable across architectures and systems may later on inhibit the use of new architectures and other brands. For discussing this issue one needs to examine the different OSs and their architectures when taking a mote-centric perspective. From an end-user point of view, who is interested in the behavior of a network as a whole, one has to raise the abstraction and consider distributed middleware systems that provide networking, sensing and data-centric functionality.

### 2.1.2 Operating Systems for Wireless Sensor Networks

Probably the most well-known OS is TinyOS [33]. It is light-weight. Only components that are actually needed are compiled into the executable. Its event-driven programming paradigm with split-phase operations supporting low-power concurrent programming [34] can be implemented using nesC [35]. In version 2, it supports a three-level hardware architecture abstraction [36] which allows providing support for different hardware with no need to actually change the entire OS. Most software implementations in this thesis are based upon TinyOS.

A flexible extension that is utilized for implementing MAMA [13] is threading [37] technology since TinyOS2.1. [13] also includes references to other popular approaches like Contiki, SOS and a mechanism for updating TinyOS programs called Deluge where complete images to-be-run can be flooded into the network. The approach taken in [13] provides similar and extended functionality, but it employs principles similar to the light-weight middleware approach shown in [32] which establishes a virtual application programming interface as well. Different kinds of middleware architectures that can be thought of are discussed in the next section.

### 2.1.3 Wireless Sensor Network Middleware

The middleware survey in [38] categorizes systems into approaches using Virtual Machines (VMs), database approaches, modular programming, application driven and data centric approaches. For this thesis, especially the VM based approaches Mate [39] and Magnet [40] (also MagentOS) and modular programming and application-driven approaches in Impala [41] and Milan [42] are relevant. These approaches cover the VM aspect in [39] and [40], mobile agents in [41] as well as application demands in [42]. There is even a comparison regarding the power-awareness, but it turns out that middleware systems optimized towards different applications can rarely be compared that way. What makes comparisons easier is reasoning about different systems' support for characteristics that can be described with paradigms. Of course, there are power-aware features in these different systems, but there is no means of reliably quantifying them. This is probably due to a lack of benchmarking approaches for WSNs. But, for turning back to what can be quantified - namely support for different characteristics' or features' paradigms - a more current survey [43] tries to set up a metric for different communication and programming paradigms of WSN middleware abstractions. On one hand, they improve over [38] because many of their taxonomy's to-be-quantified characteristics and related issues have been experienced by the authors themselves. But, on the other hand, the take home message from that article might be that a general middleware classification system covering virtually any aspect non-negligibly impacting WSN application design would just be too complex and cannot be applied practically. In other words, it is difficult to compare systems dealing with matters as broad as WSNs. Therefore systems can, in general, be better compared if the possible application space can be limited, or even better, if benchmarks could be defined.

Summing up, the main impact for this thesis is twofold. First, the outline of the multicast communication paradigm in [43] is to be remembered regarding the possibly increased applicability of NC approaches that are to be presented. Second, for quantifying optimization results and making them comparable, one has to carefully select the criterion to be evaluated. This pretty much boils down to optimizing, what virtually any WSN scenario may benefit from. This includes EHS and NC applicability for energy-efficient hardware and communication which is discussed in the next two sections. Furthermore, it fosters the final case study that will be given on battery-aware simulation in Section 3.4 that is mainly concerned with discussion about how accurate an evaluation can be and therefore what optimizations can reliably be quantified.

## 2.2 Energy Harvesting Technology and Systems

An overview of the latest application technologies using energy harvesting for WSNs can be found on supplier websites like EnOcean Alliance or GreenPeak Technologies. Some enabling market pull is being responded, which has evolved from the desire for wireless home automation - fostered by the upcome of ZigBee Alliance - operated without batteries. In particular, case studies like equipping the 57-floor Torre Espacio building in Madrid are remarkable possibilities for advertising such environments. A well written overview of enabling state-of-the-art systems and their developments can be found in [44]. However, the principles of WSNs and energy harvesting have barely changed since their respective initial case studies. Therefore, some initial enabling work for different aspects of energy harvesting WSNs will be presented.

Maybe, the most well-known initial work implementing an EHS-enhanced WSN is the Heliomote plug-in [45]. It connects solar cells to the Heliomote board with 2 AA NiMH rechargeable batteries to supply a Mica2 mote that is connected to the board in a sandwich-on manner. The Heliomote demonstration has worked and can be considered an enabling factor for WSNs. However, some research questions come up immediately, when analyzing the design, if the goal was to build a perpetual system.

1. How can the capacity loss due to ageing effects of the battery be overcome?
2. How can one find the minimum-cost EHS configuration that supports application requirements?
3. What algorithms are needed for running an EHS-enhanced mote compared to one without?
4. How can one model and simulate such a system and how can errors be handled?

### 2.2.1 The Maximum Sustainable Duty Cycle

The maximum sustainable DC is the DC of the mote that still allows ENO. Models like the one in [46] can be used to select the DC right, given (i) proper prediction of future EHD output, (ii) correct estimation of the power consumption for a selected DC and (iii) knowledge about the size of the energy reservoir.

For being able to predict future EHD output - or being confident about being able to achieve a prediction error $\sigma$ below a given threshold - different algorithms might be used [47].

Accuracy problems that occur when trying to estimate the power consumption are discussed with the second case study outlined in Section 3.4. While in principle, it is possible to apply PSMs and calculate the expected energy beforehand or measure it at runtime, this approach may run into trouble regarding the fact that most prominent efficient LPL protocols for WSNs act asynchronously. The inter-mote timing may heavily influence the actual average power consumption of a DC modeling period. As the inter-mote timing often depends on environmental parameters like temperature, it is difficult to give exact estimates of the average power dissipation.

DC modeling periods should be chosen beforehand if possible, because online estimation unnecessarily complicates matters. Doing so, one can set up an energy model incorporating the (i) EHD input power and chosen (ii) maximum variance $\sigma$ for the prediction to lock in together with the (iii) worst case power dissipation or average value with maximum possible variation. This allows determination of the minimum size of an optimal monolithic energy storage as analyzed in detail by [48]. One remark has been made by [49]. They argue that online determination of the maximum sustainable DC introduces overhead at runtime from overly complex computational operation for evaluation of the DC. They propose to span a solution space for possible context settings offline and then just select the nearest-neighbor solution at runtime.

Having done modeling and dimensioning steps right beforehand with correctly setting the DC after a lock-in phase of the online EHD estimation process with a $\sigma$ falling below the chosen threshold, the system might run perpetually without interruption. But, what would happen if the EHD output suddenly changed with not enough energy to communicate a new DC as considered in [9], because this overhead is not incorporated in the state-of-the-art approach presented above. Furthermore, is the EHS efficient enough [10] for the ideal energy budget calculation to hold, or what could be done about a complete black-out [12]? In addition, the cheapest solution for building the hardware has not yet been determined, because it is not possible to incorporate such large errors from asynchronous MAC as outlined in [15] into a process for getting efficient low-cost designs with tight bounds of the energy reservoir size. Summing up all these problems that cannot yet be tackled with any approach, it can be recommended using an approach as outlined in [11] with setting the lower bounds on acceptable available energy a little higher then needed for the application at hand. Then a slotted design can be introduced for making the power dissipation estimation process more robust. Lower thresholds for the start-up energy can be utilized to ensure full initialization of a single platform and letting it join the network. This also allows deploying additional motes later on and having them join the network autonomously. This has impact on the energy storage design as well. This is true, especially, if Double-Layer Capacitors (DLCs) are chosen as in case of RiverMote. DLCs have lower capacity and suffer more leakage, but they can be recharged more often before they suffer a capacity loss of 20%. Typically, rechargeable batteries can be recharged 200 to 500 times, while DLCs can be recharged 500000 times or even more often with examples given in [50] and [51].

The last issue that has not yet been discussed is the possible need to adapt protocols that are known from WSNs with non-harvesting motes and answering the question of whether there are environments for simulation of a system including environmental conditions at design time.

Furthermore, mentioning it for reasons of completeness, totally different directions are taken in [52]. No energy is buffered at all. Therefore, all energy is used as it is available and the network is operated on a best effort basis.

### 2.2.2 Energy Harvesting Aware Routing

The initial EHAR mechanism [53] is basically an adaptation of an energy-aware routing protocol [54] to include the EHD power profile in its metric as well as the actual State of Charge (SoC) of the system, which reflects the energy that is available at any given time.

Another research problem in the area of EHAR is a lack of simulation environments that allow realistic testing of WSN routing algorithms for many different environmental conditions. [55] identifies a lack of research comparing different EHAR under realistic assumptions. They profile 4 different types of routing using a real MAC protocol, lossy wireless connections and protocol overhead including the information that needs to be exchanged for running an EHAR as a distributed algorithm. It finally turns out that - possibly due to overhead for considering the input of EHDs - the protocol with the energy budget without EHD profiles performs best. An energy-aware routing protocol without energy state information overhead [56] may therefore be a possible choice for EHSs as well. Furthermore, [55] points out that choosing the MAC protocol and its parameters correctly [57] may critically impact the overall system. Other important aspects include the initialization and synchronization overhead that has to be taken into account for harvesting solutions. Especially, for low data rate and low DC WSNs, this can be an issue. The initialization overhead as pointed out by [3] and exemplified in [5] may overlay the effects of a possible gain with routing optimizations in low data rate networks. The low DC that has to be selected for cost-efficient EHS-enhanced WSNs with energy reservoir dimensions fitting their respective tight lower bounds may keep desynchronized WSNs from re-establishing connectivity again [9] once it has been lost.

Tackling these issues in [58] provides us with an analysis of sleep-wakeup policies that allow implementing ENO according to [48]. Scheduling the wake-up event correctly and synchronizing it with effects of the environment that are of interest to the application is discussed in [59] and [60].

Finally, considering all these effects and implementing countermeasures for all possible issues, this approach, in itself, may introduce problems. Especially, for real-time applications, counter-measures that scale timing and activation policies or buffer data on its way to the sink may impact the system. [61] gives an analysis on communication delay in EHS-enhanced WSNs dealing with such issues.

### 2.2.3 Energy Harvesting System Storage and Component Efficiencies

Energy storage and component efficiencies deserve careful consideration when tuning an EHS architecture as well. For overcoming ageing effects of batteries it may be useful to buffer the high number of charge/discharge cycles with DLCs before draining and supplying a rechargeable battery. In that sense, Prometheus [62] provides an improved concept compared to Heliomote. The main variation of the actual energy budget is buffered by the DLC, while energy is only transferred to and from the battery when necessary. A comprehensive overview of several EHS architectures is given by [63] and different harvesting application domains are outlined in [64].

From related technologies - especially RFID tags - design space exploration is known for different energy storage architectures [65]. For WSNs, the RiverMote architecture [11] presented in this thesis is an architecture that matches solar panel output voltage, energy storage needs and power supply requirements by using two DLCs in series with an additional balancing circuit. It is exploiting experiences that are available from experiments with [9] and from [17]. This efficiency-by-design approach relaxes the need for applying Maximum Power Point Tracking (MPPT) or conversion component selection for bridging wide voltage differences depending on the application DC [66]. In the event that special control was needed for locking in an operating point, different approaches would be possible. AmbiMax [67] and Everlast [68] perform MPPT for improving efficiency. Everlast uses a software-based approach for control, while AmbiMax implements MPPT in hardware.

Concluding the section on energy harvesting for WSNs, there are several different issues that need to be considered as EHS characteristics and WSN constraints meet. Also, there are different dimensions along which this characteristic mapping has to be considered and there is, as yet, no self-contained design methodology available. In [69], they point out joint timing issues at different tiers of EHS and WSN models, while in [70], they point out the inherent need for cross-layer design approaches for wireless network design. This also needs to be considered when dealing with optimizing information flows in the following chapter on NC. Cross-layer design issues for WSNs are specifically dealt with in [71].

## 2.3 Network Coding for Wireless Sensor Network Communication

Developments of NC strategies have been initiated by the seminal article on network information flows [23]. They explore the maximum bandwidth utilization and how it can be achieved for multicasts of mutually independent information flows. The mutual independency has to be remembered when discussing the applicability of NC in combination with source coding in WSNs for gathering partly correlated data. The separation [72] and joint operation [73] is possible in principle. However, for the applicability in real world scenarios, especially the type of side-channel information carrying the correlation information has to be considered.

[23] points out, that for achieving min-cut max-flow like characteristics for the information flows, these should not be considered as fluids as in other work. They should rather be coded at forwarding nodes (what can be thought of as being a combination rather than a fluid pushing through) which they then call NC. Virtually any communication paradigm may possibly benefit from their approach in every occurrence of multi-casts experiencing crossings in their flow. Applicability has to be checked with min-cut max-flow based measures. Although initially meant to be applicable for switching mechanisms and wire-bound traffic, wireless communication also benefits from direct appliance of NC for butterfly structures and relay nodes. For the wireless case - on a networking level - this means that every occurrence of crossings of messages may be exploited with NC. This is especially applicable for multi-path routing [74]. Furthermore, the multicast communication paradigm in general may give opportunities [75], as do inter-session [76] and opportunistic [77] approaches. Related work discussing motivation and application for multi-sink multi-casts can be found in [78] and [79]. ONC is the most sophisticated networking approach that is going to be presented in this thesis. The author wants to point out that the novelty of applying this approach to WSNs is important. It can best be compared with COPE [80], but that approach is more appropriate for MANETs rather than

for WSNs. Similar as in [77], ONC for WSNs is to be evaluated for delay, performance and power, while there are no other ONC approaches published for WSNs doing so. A brief introduction on how WSNs might utilize NC approaches is discussed in [81] and later, but in greater detail, in [82].

### 2.3.1 Different Ways of Applying Network Coding

There are two different ways of how NC techniques can be beneficial.

- One can improve the network throughput with NC. The increased information capacity of the channel relaxes the networking load. Fewer messages have to be sent for the transmission of the same amount of information.

- Or, system complexity can be reduced. Determining the maximum achievable throughput and finding a routing solution is known to be NP-hard, while a NC solution can be found in polynomial time.

### 2.3.2 Lowering Networking Complexity with Network Coding

[83] argue that despite possible energy conservation from NC, it is more worth exploiting it for lowering complexity of building networking topologies. This has to be considered when applying NC for WSNs as well. From a network model point of view, they also deal with point-to-point connections, but they do not rely on directed models. [83] rather uses undirected network models, which they outline as being more general. In some sense, this brings the model closer to wireless applications, because the channel may provide the same capacity in both ways, but time variation of the channel and different transmission power leading to varying signal to noise ratio may apply to it.

[83] is built upon [23] and [84]. Although, they lack a complete description (only non-representative examples are given) for the Steiner tree packing problem [85] upon which their complexity comparison is based, they advance the view that NC gives a tool for finding optimal transmission schemes more easily, rather than gaining from the coding advantage. They explain that even for comprehensive testing of thousand network topologies they do not exceed a coding advantage of 1.125 in their tests which - given the overhead that comes with a new networking paradigm - might be no gain at all under realistic conditions. Surprisingly, they do not delve, in greater detail, into the fact that they were able to find examples manually, although they did not come close to the upper bound of the coding advantage of 2. Nevertheless, they discuss the upper bound for different types of networks. For acyclic networks they reference previous work [86] which shows that for directed acyclic networks with integral routing requirement the coding advantage is not upper bounded. It grows proportionally as $log\left(|V|\right)$ where $V$ are the network vertices. They then argue that for undirected networks it is bounded by 2 [87] and that even this value cannot be approached easily. One thing they did not mention is that the undirected network model can be used efficiently to model wireless scenarios [82]. Now, that is exactly where this thesis comes into play. The topology of WSNs need not be fixed and can easily be changed with different transmission power or may even change on its own due to mobility or due to channel noise although the focus of this thesis is put on static deployments. Also, this thesis does not control the power for reducing collisions. But,

instead of trying to adapt routing tables, all that is done is slotting [88] sending times [5] at the MAC and considering worst case scenarios in terms of congestion. The focus is more on virtually adjusting links in such a way that NC can be applied. Small structures are utilized to make them better controllable [3] though one may pack them together [5] and scale academic experiments for achieving the maximum gain and therefore testing upper bounds that are expected from theoretical results.

### 2.3.3 The Network Coding Advantage for Wireless Sensor Networks

Although, [83] motivates lowering complexity, because usually, there are few opportunities when the maximum NC advantage can be achieved referring to the point-to-point connection case, preconditions are somewhat different for this thesis. For wireless XOR-coding [89] the NC advantage is bounded by 2 [82]. Most algorithms in this thesis will be concerned with maximizing the opportunities to apply NC either with autonomous [4], localized [3], scalable [5], energy managing [7] or opportunistic [6] approaches. In all of these cases one waits for given scenarios to occur and then applies NC. The focus lies on trying to achieve representative settings, because either noisy and congested channels are assumed or localized and overhead-free integration into existing applications is needed or worst case MAC slots are assumed in combination with really distributed DC negotiation.

Other recent work, focusing on the interplay with lower parts of the networking stack, goes into similar directions. In [90] they are regarding NC that is aware of possible MAC slotting strategies for possibly interference-free communication. Other approaches at the physical layer, explicitly exploiting interference for NC are not considered in this thesis and are better dealt with in the field of cognitive radio. Further approaches like detailed elaboration of linear and random NC and the combination with source coding are out of the scope of this thesis as well. Either, the approach is not appropriate for implementation on resource constrained motes, or cognitive radio and physical layer considerations are too distinct issues compared to networking middleware aspects.

Considerations in this thesis will focus on the interaction with DCs, EHAR, slotting, and therefore, also LPL MAC has to be taken into account. Application of computationally different approaches utilizing different finite fields or random NC focusing on complexity reduction instead of the NC advantage are not considered in this thesis.

## 2.4 Methods and Tools Tackling WSN Technology, Design and Optimization

This section is split in two parts. First, tool chains and hardware platforms of some of the major players in the field of low-power and energy harvesting solutions are briefly surveyed. For these, the completeness of available simulation and measurement approaches related to WSNs are summed up and missing links are identified. Second, related work is introduced that provides background for and highlights related work of missing parts in the approaches presented previously. This will serve as a basis for the energy-aware considerations of battery and harvesting aspects of modeling and simulation.

### 2.4.1 Industry Platforms for Modeling, Simulation and Implementation

A number of state-of-the-art WSN platforms are built around the MSP430 and ATmega128 families of Central Processing Units (CPUs). Therefore, simulation engines for these types of platforms are of special interest for the WSN community. An early general power profiling simulation approach is presented in [91]. Avrora [92] with AEON [93] has been established as being an instruction set simulation environment and enabling power profiling for the ATmega128. At the time of the writing of this thesis MSP430 implementations are currently underway with promising results for adding power profiling support for this platform as well. However, what is missing for these simulation engines is support for modeling channel effects and environmental conditions.

This is less of an issue for another simulator at the level of basic blocks of instructions. TOSSIM [94] allows the incorporation of channel effects as well, which consumes quite a lot of time for setting up the channel characteristics at the beginning of a simulation. But in contrast to Avrora, the basic block level for simulation gives a performance increase when actually simulating the network. Power profiling extensions exist for TOSSIM as well. While the achievable power profile results cannot be as exact as those achieved with an instruction set simulation environment, similar expressiveness may be achieved. Results of the second case study of this thesis will show that severe issues can be related to the thought of to be more accurate approaches. Publications dealing with that issue are mapped to this thesis' topics in Section 3.4.

An obvious advantage of TOSSIM, that is not tailored towards a specific instruction set, is its portability across different platforms. The nesC program that is written using TinyOS components is compiled for the development host platform and can therefore be run as a native executable. The simulation engine takes care of interfaces to the outside world including Analog Digital Converters (ADCs) and the radio interface.

Despite the invaluable insights that can be gained from using these and other simulation tools [95], what is missing is a complete integration of simulation and emulation tools considering possible error sources of the simulation engine or models for functionality, power dissipation and timing. An example for quantifying possible errors in hardware measurements and software simulations is given in [96]. Well-known related hardware measurement approaches including error analysis are presented in [97] and [98]. However, what is also still not available, despite many claims of creating simulation approaches including environmental conditions like in PAWiS [99] or battery models as in [100], is a proven record of such approaches comparing simulation results with hardware measurements or emulation regarding battery models and energy harvesting. The TOSPIE2 [1] methodology with its integration of precise annotations of measurements of power profiles and timing in hardware as well as in simulation also connects to emulation and profiling of mote hardware including energy harvesting characteristics [17]. The approach called COOJA [101] is meant to connect approaches like Avrora and TOSSIM with NS-2 [102] and other integrating tool-connecting attempts can be found in environments like VM-Net [103] and EmStar [104] or EmPRO [105]. However, none of these approaches has made its way to a level, technically sound enough to be integrated or adapted by other research groups in the community. An example for an energy harvesting simulation engine still lacking some results with using a system-level experience is Castilia. Another hardware related approach is currently being developed using SystemC-AMS with first results being presented in [106].

Summing up, there are only few tools for EHS simulation. Furthermore, the scientific community lacks an integrated approach that combines system level simulation and networking. Approaches as shown in [107] are limited to single hardware instance considerations. Most environments for network level consideration of EHSs take a very abstract point of view with stochastic modeling as in [108]. The few integrated solutions as in [95] and Castalia in [109] lack versatile tools for accurate hardware measurements for validating simulation results. Another approach, presented as TOSPIE2 in [1], integrates accurate profiling of PSMs and EEMs, but the architecture is a very loosely coupled system.

### 2.4.2 Battery Models and Applications to Wireless Sensor Networks

When modeling and simulating low-power embedded system architectures' power consumption, one needs a profound basis that different power-aware optimizations can be compared to. For an often neglected issue for achieving accurate WSN lifetime modeling - battery effects modeling - related work for battery characterization setups, battery effects modeling and networking optimizations is outlined.

Different types of energy storage devices exist where each type implies special characteristics that have to be considered when tuning a system. However, as existing simulation environments even lack support for non-rechargeable batteries, rechargeable batteries and their effects are considered as being out of scope.

#### Automated Battery Characterization Setups

The battery performance at a given point in time depends on several factors that influence the system. In particular, batteries are systems with memory. The different possibilities of temperature history, the discharge rate of the load, the current SoC and possible combinations of what these values have been in the past result in a large number of different contexts that the battery may be in. Therefore, profiling performance characteristics - as shown for Alkaline Manganese types of Duracell primary cells in [110] - takes time and needs accurate measurements. In general, characterization may be based on SoC measurements that are chemical methods, current integration methods or voltage measurement methods. However, for finding a profiling methodology that can be integrated on the WSN mote later on, chemical methods and current integration are not practical.

For profiling the SoC, voltage-based measurements are promoted in [111]. They also discuss applicability to State of Health (SoH) measurements. SoH knowledge is especially useful for extending ideas to rechargeable systems. Setups for profiling different types of batteries, including charge and discharge performance, are presented by [112] and [113]. In [112] they provide simple Equivalent Circuit Diagrams (ECDs) of batteries using internal resistance and two low-passes with different time constants for modeling a battery. [113] uses ECDs as well and provides insights for online determination of a model's parameters.

#### Battery Effects Modeling

ECD modeling of battery effects is quite common in the fields of electrical and computer engineering. Distinguishing between theoretical capacity and actual capacity as of [114], the actual battery performance in terms of the amount of charge that is extracted, mainly depends on discharge rates, temperature conditions and ageing effects. Better actual performance means more extracted charge before the battery cut-off voltage is reached. For primary cells, recovery effects can be utilized for increasing their performance.

Different simple empirical models are given by the Thevenin battery model and a linear electrical model with their ECDs given in [115], Peukert's power law explained in [51], and the model of Pedram and Wu [116]. One thing that all these models have in common is that batteries are modeled as systems with memory. Despite the existence of model evaluation experiments with an error of one order of magnitude or below, there are a number of side effects that need to be considered for WSNs that cannot be handled entirely by using these standard models and variants.

First of all, a drain down to the cut-off voltage of batteries can often not be achieved due to a limited voltage range of components and needed converters' capabilities and efficiencies. Depending on the type of simulation integration, motes have to be removed from simulation when the lower threshold has first been hit, or they have to be shut off temporarily if proper functionality can be assumed after a relaxation period.

Furthermore, the evaluation of SoC dependent effects that could be used in simulation environments cannot easily be mapped to real-world deployments and real-time adaptation mechanisms. While the SoC models for simulations may utilize a lot of computational power, several difficulties arise at runtime where resources may be limited. The accuracy of voltage-based SoC determination methods depends on measurement capabilities and ADC accuracy. In addition to that, online SoC determination and appliance of battery models demands exact measurements of the load's characteristics and memorizing it as well as possibly evaluating computationally complex models. [11] depicts such in-situ measurement's accuracy for an EHS-enhanced WSN platform to be below 10% in relevant areas. Adding such an error to battery model input may drastically limit model expressiveness.

#### Battery Technology and Networking Optimizations

Knowledge of energy storage characteristics allows the application of power management techniques and optimizations. Transmission scheduling in [117] and [118] and traffic shaping techniques [119], in particular, apply. Dealing with energy storage performance modeling issues necessitates complex measurement, analysis and algorithmic efforts. This fact and the time it takes to prove novel energy storage technologies working makes battery modeling and optimizations lag behind current technology capabilities and application requirements. Therefore, despite novel technologies perhaps outperforming older technologies, the battery modeling considerations in this thesis are based upon technologies as exemplified by [120] and [121] for DLCs and [44], [122] and [123] for batteries and energy harvesting aspects. More recent developments would have included graphene-based ultracapacitors [124] and Lithium Ion Batteries with carbon nanofibers [125].

## 2.5 Summing up Related Work and State-of-the-Art

Chapter 3 will utilize related work findings outlined here for putting pieces of the concept together. The different system components and development aspects that are presented here can be used for completely modeling a perspective of WSNs that covers WSN project life cycles from modeling and design through to implementation and simulation up to the measurement and evaluation phase. Parts before and after these phases like requirements engineering and quality-of-experience measurements shall not be discussed. From a system approach perspective this thesis spans contents from hardware and the MAC layer up to distributed computing of applications upon middleware layers with possible inclusion of host-side and programming aspects. For the hardware that is utilized by these layers, especially the applicability of EHDs, the efficiency of EHSs and the expressiveness of harvesting and battery modeling have been considered in detail.

# 3 Concept and Interaction of Novel Comprehensive Methodologies in TOSPIE2

The overall conceptual architecture that is related to fields of research from Figure 2.1 is shown in Figure 3.1. The main contribution claim and how publications are mapped to the architecture are discussed in Section 3.1 for the NC middleware and in Section 3.2 for the EHS implementation. While networking and middleware components are depicted in Figure 3.2, topics that are related to energy harvesting are highlighted in Figure 3.3 accordingly. Sections 3.3 and 3.4 cover the case studies' contribution on related middleware services and development tools for WSNs including battery-aware simulation.

Different groups of topics will be dealt with separately. Publications of Chapter 6 will be grouped around these topics. This lets one follow the coverage of the overall thesis architecture more easily. While some publications could possibly be mapped to different topics, a one-to-one mapping according to the most important parts of a publication's significance is used here in terms of lucidity.

## 3.1 Network Coding Middleware Services for Wireless Sensor Networks

First of all, the middleware services' concepts shall respect what has been discussed in Chapter 2 regarding the generality and comparability of approaches. Comparability of different approaches as well as simulation and measurement results is assured by using full overhearing among all motes in all experiments. The virtual adjacency matrix for representing a given topology is then established directly above the TinyOS MAC layer. Messages that are not meant to be heard by a mote are discarded. The advantage of such hard-coded connectivity is twofold. First, measurements are repeatable and can therefore better be compared. Experiments can simply be rerun, while setting up specific topologies of hardware measurements by hand would be a very challenging and error-prone task. Second, the power dissipation that is directly affected by whether motes can hear each other or not can more deterministically be profiled. Therefore, simulation results will be compared more accurately to what is measured in hardware experiments. However, the functionality of the method that is to be profiled must not be changed by the virtual adjacency matrix in order to make such an approach applicable. Therefore, congestion-dependent effects must be neglected in the evaluation or the occurrence thereof must be avoided. The work in this thesis, in which such effects may impact evaluation results, is set up with using Time Division Multiple Access (TDMA)-based sending. The drawback of using concepts as described above is the overhead for unhinging the effects that are to be evaluated from additional effects that come from full overhearing. However, accepting

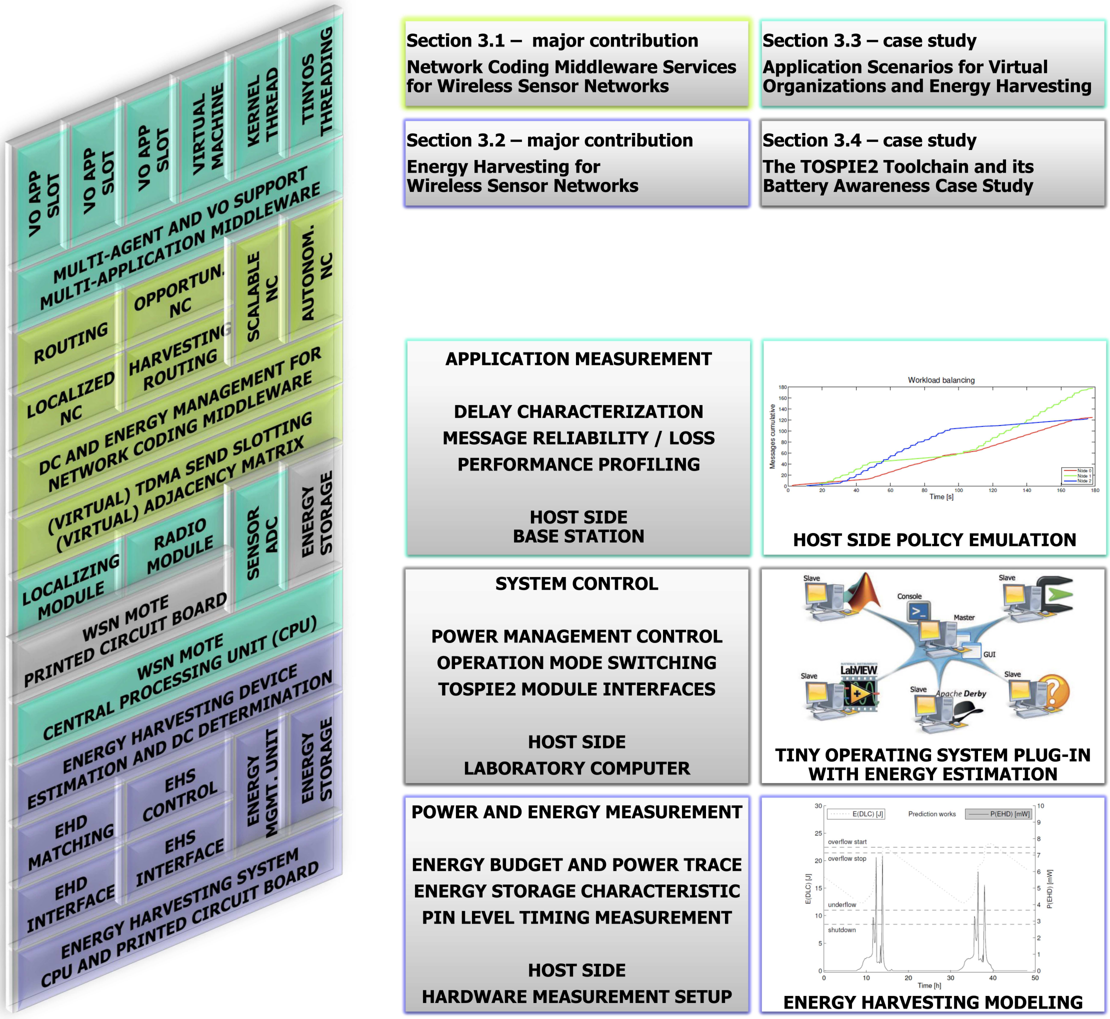



Figure 3.1: The conceptual structure of the overall system architecture provides details on different tiers of the coarse grain structure in Figure 2.1. It outlines where the main contribution and case studies are implemented in the EHS-enhanced WSN architecture. At host-side, tools and services enable novel development and evaluation techniques.

that overhead and accepting a slightly reduced design space (i.e. some parameters like maximum sending frequency depend on the number of motes participating in communication) allows for the application of technically sound evaluation processes. Another issue that shall be respected is related to effects from heterogeneous networking approaches that might interfere with each other. Where applicable, messaging structures are set up in such a way that different approaches can coexist besides each other. If unknown message types are delivered, messages are simply discarded.

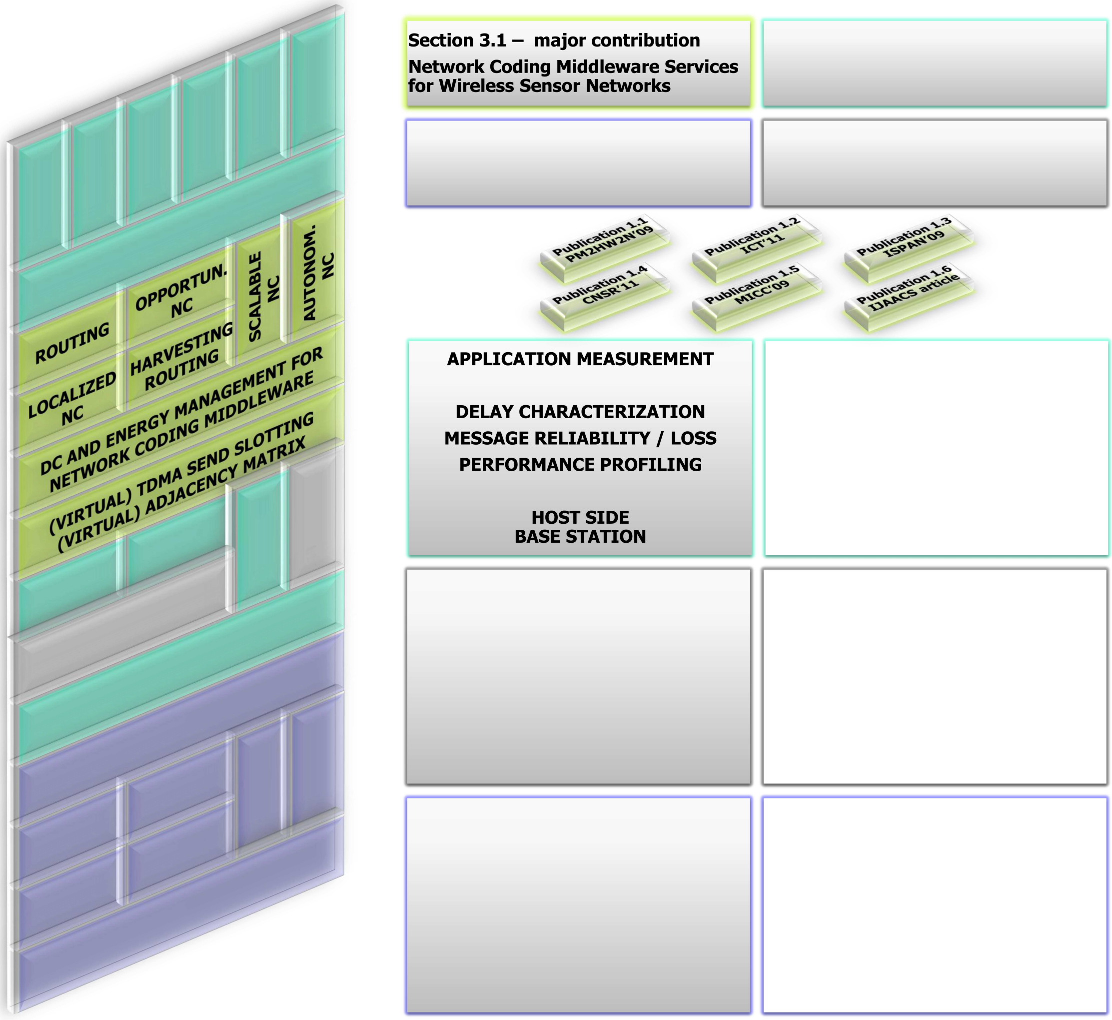



Figure 3.2: NC services include LNC [3] (publication 1.1 - PM2HW2N'09), ANC [4] (publication 1.2 - ICT'11) and SNC [5] (publication 1.3 - ISPAN'09). Energy management [7] (publication 1.5 - MICC'09) and SCN extensions, including multi-path routing considerations [8] (publication 1.6 - IJAACS) and ONC [6] (publication 1.4 - CNSR'11) are covered by the primary contribution as well.

### 3.1.1 Localized Network Coding in Wireless Sensor Networks

In particular, the embedding of an overhead-free and therefore necessarily control-free networking optimization necessitates applying localized algorithms. The LINDONCS approach [3] evaluates LNC for WSNs using energy-aware routing protocols. The approach makes use of state machines keeping track of neighboring motes' information flows. When an NC structure and applicability of LNC is detected, LINDONCS modifies meta-information that is sent for propagating data needed for evaluating the routing metric. In the case of being applied in an otherwise energy-aware routing WSN, information flows are attracted towards dedicated motes that are meant to be incoming motes for the NC structure by advertising a high level of energy budget at these motes. Results show the approach's applicability for conserving energy even in networks where other motes do not know about the LINDONCS approach.

### 3.1.2 Autonomous Network Coding in Wireless Sensor Networks

It is not only seamless integration to otherwise non-NC WSNs that is important. Also, the possible effects regarding the application utilizing the messaging interface has to be considered. [4] presents ANC for a given application scenario. Again, NC can be applied with no need for accepting any messaging overhead. State machines are used for checking applicability of this approach as well. However, this time, the focus is on conservation of application behavior rather than interaction with energy-aware routing protocols. Therefore, the routing modules - and of course the application modules as well - are kept completely intact. The interaction of ANC with the WSN happens when messages are passed from the routing layer to the application layer or vice versa. No special constraints are put on the routing layer, neither is there any need for a specific optimization metric to be used.

### 3.1.3 Scalable Network Coding in Wireless Sensor Networks

The idea of NC is applied for energy conservation from the NC advantage rather than because an optimal forwarding structure can be found more quickly than for routing alone. Given the knowledge about the bounded gain of wireless NC, it is much easier to evaluate for practical settings how well the NC advantage can be utilized compared to reasoning about the quality of methods for finding NC and routing schemes. Implementing SNC and evaluating the approach clearly shows that in areas, where NC is applicable, the NC advantage approaches its theoretical maximum of 2. Besides this scalability experiment in [5], an approach for coding multiple unicasts at the same time is introduced for mesh-structured WSNs providing a basis for energy management and multi-path routing integration.

### 3.1.4 Extensions for Multi-Path Routing and Energy Management

The approach from [5] is taken one step further in [8] by the concept of letting its multiple unicasts be made of multiple information flows from multi-path routing. Multi-path routing tries to overcome some of the inherent dependability issues due to message loss in wireless communication. Furthermore, [8] outlines the overhead for the middleware initialization phase until the protocol has settled down to a static mode of operation. After the initialization including the flooding phase for setting up the routing table is finished, a constant level of energy conservation can be achieved by applying SNC. In the event that the energy budget is not evenly distributed among motes (e.g. due to different EHD profiles in the past), the energy management approach in [7] can be applied for balancing it. The approach relies on the timing for SNC and the energy consumption of storing persistent data on the EHS architecture in [9]. For the sake of completeness [126] evaluates energy and performance issues with persistent memory for another embedded technology - SmartCards - which also contains the basic multi-application concept in their card manager that will be used for the MAMA approach discussed in Section 3.3. [7] trades opportunities to conserve energy and therefore also the need for using energy to persistently store data among motes over time. Virtually pushing these opportunities and needs back and forth through a WSN, where chaining is possible as well, the energy budget among different motes can be managed.

### 3.1.5 An Integrated Opportunistic Network Coding based Middleware

An approach relying on trying to establish opportunities alone is presented implementing ONC in [6]. The ONC approach is applied for different communication paradigms and other middleware services and settings for which it might be possible to experience side-effects with. These side-effects include detailed analysis of effects from ONC interaction with EHS DC determination, setting and effects as well as differently ordered TDMA-based sending, scalability, robustness to noise and traffic load. The approach is implemented upon EHAR and compared with EHAR alone as well as with SNC alone.

## 3.2 Energy Harvesting for Wireless Sensor Networks

Energy harvesting considerations in this section are mainly concerned with EHD profiling and estimation of the maximum sustainable DC as well as characterization of EHSs and EHS-enhanced mote hardware. Focus will be given to applicability, efficiency and cost per performance.

### 3.2.1 Energy Harvesting Maximum Sustainable Duty Cycle

Designing applications using energy harvesting for building cost-efficient perpetual systems still lacks sophisticated support for modeling, simulation and design space exploration. Therefore, for understanding different components' characteristics and interaction of components, usually, prototypes need to be built for being profiled. A first step towards such developments is presented by the EHS in [9]. Different EHDs can be profiled with the system, and they can be used to supply a Mica2 mote. This is then responsible for querying EHD profiles, the energy budget status and power dissipation information from the EHS. This information can be used to calculate the maximum sustainable DC, negotiate a network-wide one and set it at the EHSs accordingly. Possible DC problems - especially for EHSs dimensioned with cost-efficient small energy storage structures - can be tested with the platform. Optimizations regarding such issues can also be implemented. An example of using thresholding techniques for dealing with possible problems regarding small energy reservoirs is exemplified in [9]. What knowledge can be gained from EEM profiling experiments is outlined by the concept and contributed publications in Section 3.4 that is dealing with tools and services supporting the development of EHS-enhanced WSNs.

### 3.2.2 RiverMote Prototype Implementation

While the EHS in [9] is meant to be modular for enabling rapid prototyping with no need for completely redesigning the system, the RiverMote platform [10] with an extended version in [11] is implemented based on results and characterization of the first EHS.

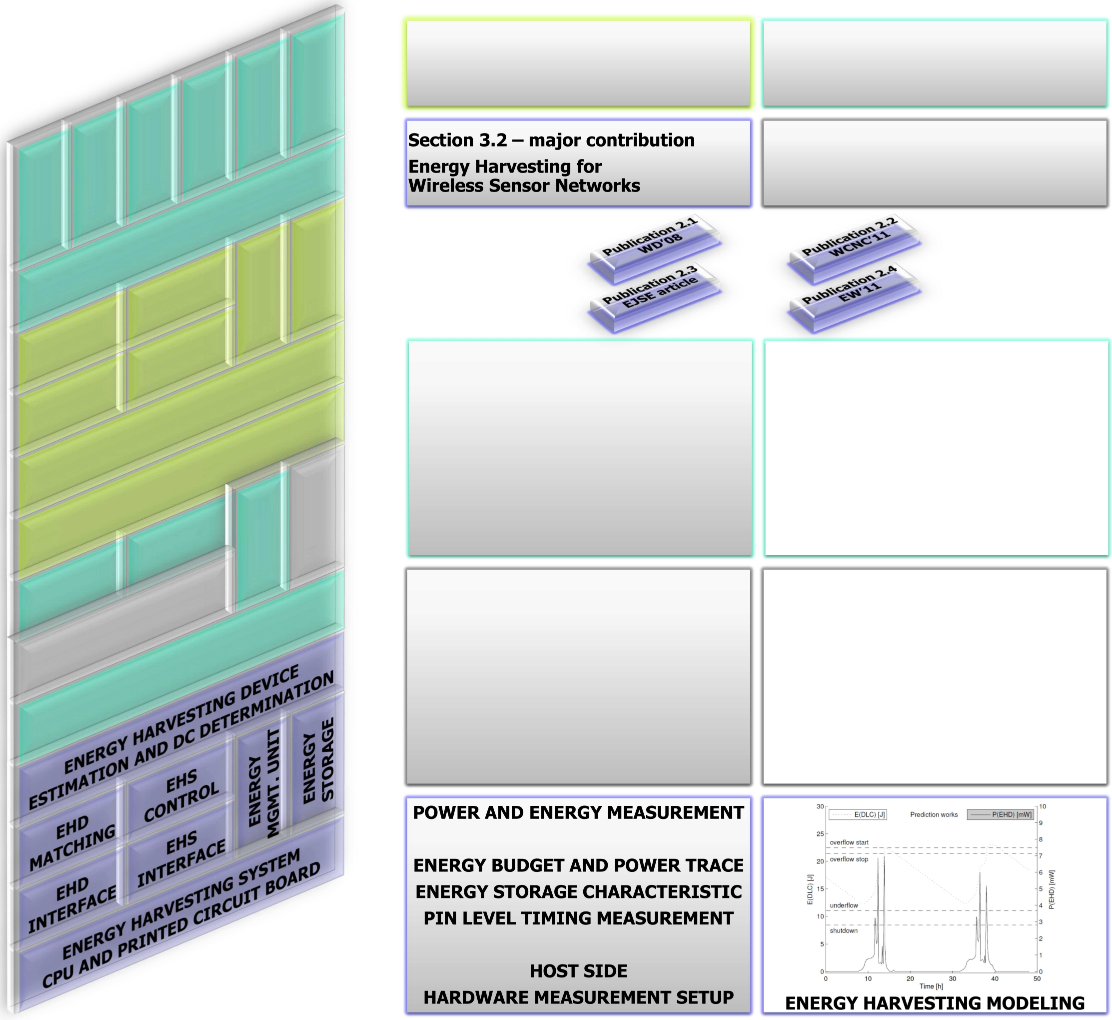



Figure 3.3: Two EHSs have been built. One EHS [9] (publication - 2.1 WD'08) is meant to be used for EHD and mote characterization and for reasoning about DC calculation and negotiation. Another platform called RiverMote [10] (publication 2.2 - WCNC'11) with an extended version in [11] (publication 2.3 - EJSE) is implemented based on results and characterization of the first EHS. Robust BOS [12] (publication 2.4 - EW'11) is achieved especially based on efficiently designed architectures [127] and trade-offs [128] as outlined in related work.

With a given application scenario in mind, discussed in Section 3.3, it implements an efficiency-by-design approach reducing power dissipation and efficiency-loss overhead from unnecessarily complex or control-based design. As discussed in related work in Chapter 2, it is reasonable to counteract possible unexpected scenarios with back-up strategies and fail-safe states. RiverMote is a platform which makes extensive use of that concept by implementing an energy hysteresis in hardware. It resets the mote part in the event that a

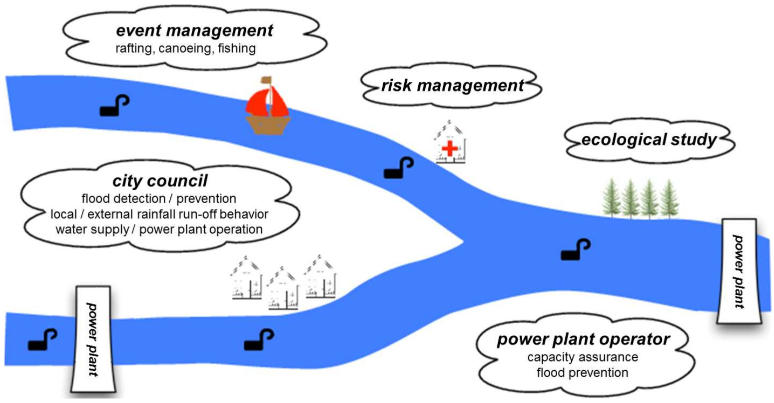



Figure 3.4: As a case study, a scenario has been developed for running VOs on a river-level monitoring platform. The network is more cost-efficient through having it shared among multiple users.

lower threshold, needed for sustaining the operation, is violated, and mote parts are only switched on when there is enough energy available to join a running network. Furthermore, the energy storage characteristics and its BOS prediction combined with a discussion on the low-cost reliable BOS of more than three weeks for fully charged DLCs can be found in [12].

The RiverMote architecture mote part is implemented upon the TelosB CPU for possibly easing up future porting of TinyOS to the system. Attached to the sensor interface, it is meant to use pressure transducers, ultrasonic transducers or a Global Positioning System (GPS) receiver for measuring river levels. As a single industry strength pressure transducer is often up to an order of magnitude higher in cost than the RiverMote printed circuit board alone, this option has not been pursued any further. Furthermore, the use case scenario for autonomously monitoring a river's level for flood warning systems could be threatened by a cable spanning from a RiverMote at the surface down to the bottom of the river. Especially, for the flood warning use case, one has to expect flotsam tangling with the cable and breaking the system. Therefore, only the ultrasonic transducer and the GPS solution, in particular, are implemented and evaluated for their applicability.

## 3.3 Application Scenarios for Virtual Organizations and Energy Harvesting

Figure 3.4 depicts the scenario for the first case study. The cost-efficient EHS-enhanced WSN platform is combined with a scenario where multiple applications can run as VOs, further shaping the cost-efficiency of the overall system. This multi-application approach is also a perfect use-case for real power management and tuning such a system with battery-aware simulation technology. That simulation environment or its case study is one of the latest results that could be achieved with researching related services, tools and techniques.

Figure 3.5 depicts components that are part of or related to the first case study. Motivated by the cost-efficient and sustainable RiverMote platform outlined before, adding multi-application support and the capability to reprogram the resultant VOs with OtAP allows using the platform in even more cost-efficient ways. A small number of services and methods have to be considered as well for general applicability of the multi-application paradigm. But first, there is an inherent need for adding localization information to sensor data. While RiverMote provides GPS data for outdoor deployments, a solution for the not yet discussed issue of indoor localization is to be covered in this section. General integration issues and especially energy harvesting DC negotiation is considered and put into relation to the energy management service for SNC. Finally, a setup for accurate measurements of motes and EHSs for characterization of PSMs and EEMs and service characterization in an emulation environment where simulation cannot be applied is covered as well.

### 3.3.1 Virtual Organizations for Wireless Sensor Networks

MAMA [13] allows the running of multiple applications on a single WSN. The same assumptions as before are made for testing and profiling a middleware system. The middleware is responsible for implementing networking mechanisms and arbitration of accessing motes' system calls. This type of sharing of hardware resources can be tuned even further by implementing a VM based approach for enabling mobile agents for VOs. As yet unpublished material is presented in the corresponding case study in Chapter 4. The mobile multi-agent approach allows load-balancing controlled by a host-side power management console. As for the localization service [14], simulation results of MAMA are compared with measurement results of the system for the testing of their expressiveness. As depicted in Figure 3.5, a base station is used as a gateway between end-user interface and power management console at host side and the WSN. For long lasting deployments, the measurement approach allows first emulating adding of agents for determining whether power dissipation bounds are respected [17] before injecting them into the network. Implementing annotations to quantify the activation of different subsystem components or message transmissions by the agents therefore allows implementing power management and energy conservation at the host and configuring the network accordingly.

### 3.3.2 Power-Aware Indoor Localization and Simulation Expressiveness

Implementing and tuning a localization system gives the opportunity for profiling simulation results' expressiveness compared to measurement results [14] at different steps that are taken in the data model. A single step might be to calculate range estimates for time series of sensor readings or to combine range estimates to a position estimate. Both the Probability Density Function (PDF) of simulation results and the PDF of hardware measurement results are compared. If the same characteristics and especially the change of characteristics of error PDFs can be observed after tuning a computational step in simulation and also profiling it on hardware, simulation can be assumed to be appropriate for tuning optimization mechanisms. This process is repeated for all different design stages or parts of the data model and computational architecture where prototyping can be sped up by simulation means.

A discussion on how the measurement setup used for hardware measurements and the simulation setup is integrated into TOSPIE2 is given in Section 3.4.

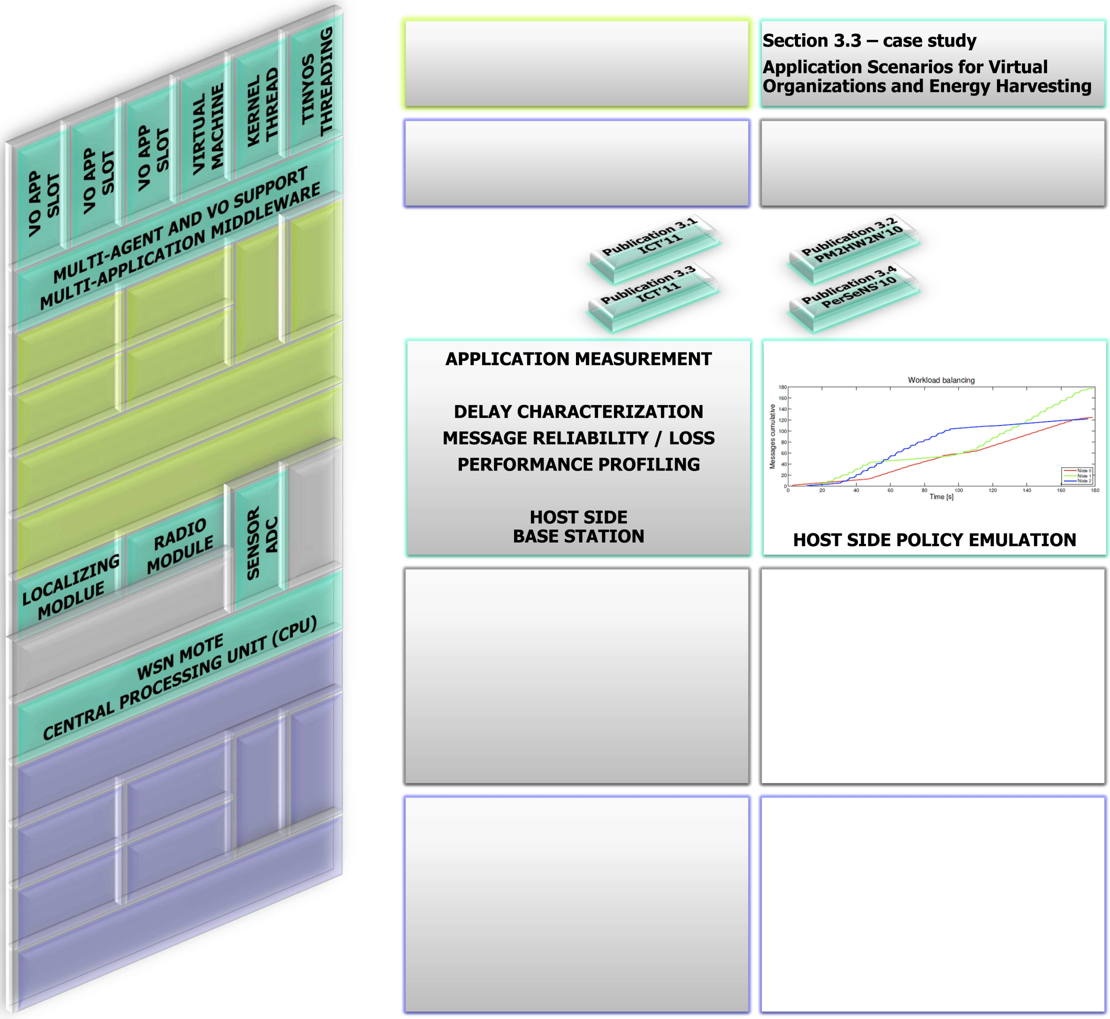



Figure 3.5: RiverMote is designed to be cost-efficient and sustainable. MAMA [13](publication 3.1 - ICT'11) allows sharing hardware among applications. Necessary localization [14] (publication 3.2 - PM2HW2N'10), harvesting integration [15] (publication 3.3 - ICT'11) and measurement [17] (publication 3.4 - PerSeNS'10) techniques are dealt with.

## 3.4 The TOSPIE2 Toolchain and its Battery Awareness Case Study

Figure 3.6 outlines where the TOSPIE2 [1] approach comes into play and profiles mote and EHD power dissipation, energy budgets, efficiencies and timing behavior. A power state database holds a PSM for the target architecture that can be profiled with the hardware measurement setup [17]. A modified Avrora can be used for power profiling and state switching simulation where the PSM database allows easy tuning erroneous parts with

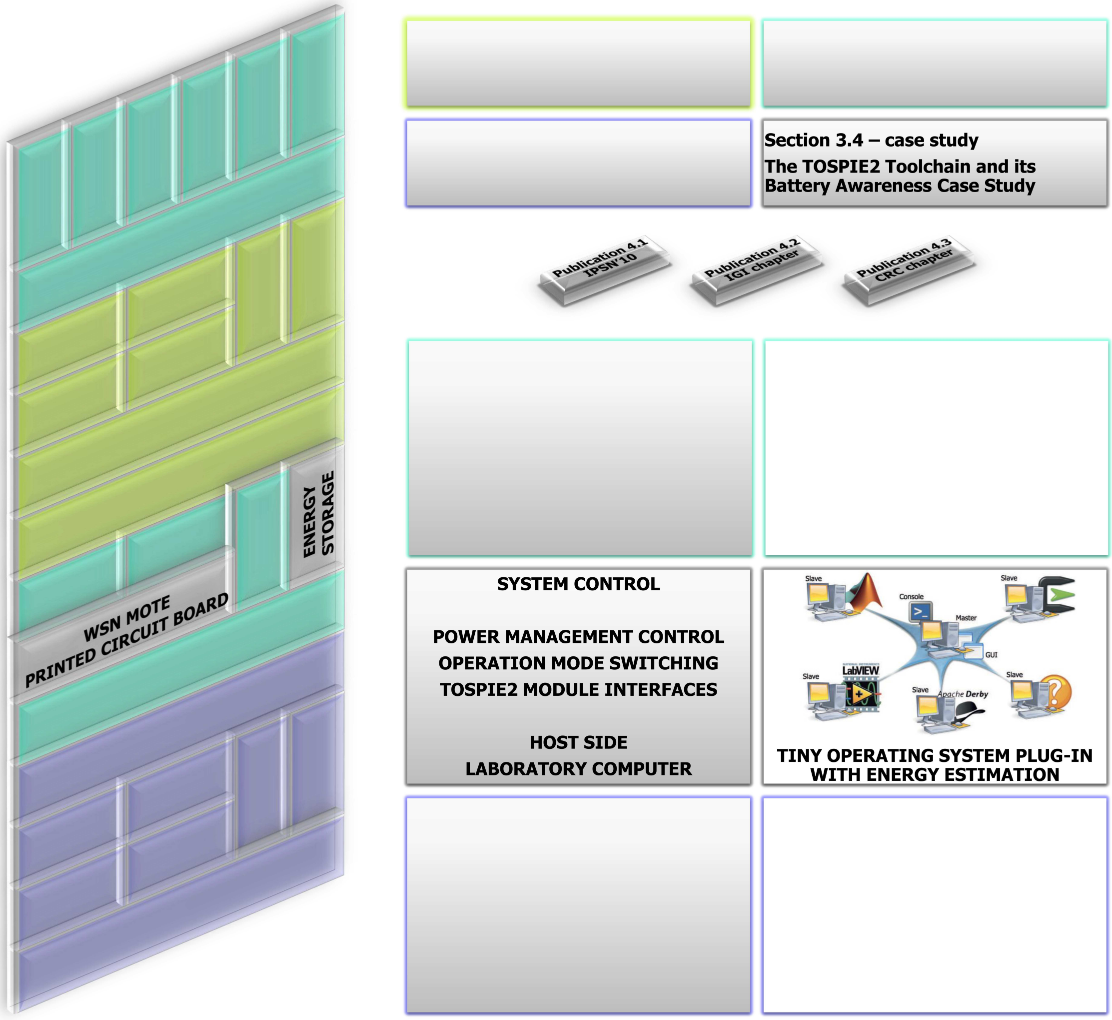



Figure 3.6: The TOSPIE2 environment [1] (publication 4.1 - IPSN'10) allows applying design rules discussed in [2] (publication 4.2 - IGI). Special focus is put on battery performance profiling and simulation [16] (publication 4.3 - CRC) using parts of the setup. Related approaches that have been taken into consideration where appropriate, include profiling energy supplies [129] and voltage scaling in [130] and [66] as well as simulation of energy harvesting in [106].

visual inspection of comparing simulation and measurement results of power profiles and state switching information. Experience from using the environment has led to the design rules applied for simulation and measurement of the publications referenced in preceding sections. Several possible issues and design rules are discussed in [2]. Analysis under the TOSPIE2 umbrella and results, profiled with the environment, allow deducing the accuracy and expressiveness of both simulation and measurement approaches. Especially the following questions are answered.

- What is the expected error of state-of-the-art power profiling simulation due to a lack of PSM expressiveness?
- What impact does missing the operating point have on EEMs and how accurately can the operating point be determined?
- How accurate are in-situ hardware measurement and laboratory setup measurement approaches?
- What is the actual variation of power dissipation in WSNs due to different energy storage structure clamp voltages?
- What is the variation of power dissipation in WSNs due to non-deterministic parts in MAC protocols?

The environment also includes an, as yet, unpublished upcoming energy harvesting simulation environment and battery performance modeling and simulation tools. The so far unfinished discrete event simulator for energy harvesting WSNs implements applications as state machines operating component-based EHS-enhanced motes. The networking simulation part of the EHS simulation includes timing and interference models of the radio and the audible channel.

Furthermore, the automated behavior and the possibility to remotely control hardware measurement setups via the internet makes TOSPIE2 an appropriate choice for being used for time-consuming battery performance characterization. First results outlined in [16] show applicability of the environment for (i) profiling battery performance characteristics, (ii) exploring battery models with Matlab SimScape and gradient descent methods and (iii) running the most appropriate models for WSNs with integration into Avrora.

## 3.5 Summary

This thesis' organizational structure comprises different mote and network tiers. Furthermore, different stages of the development life cycle and modeling environments are considered.

The major contribution of this thesis is built upon power-aware networking and energy-efficient harvesting. A number of first-authored papers are mapped into their sections accordingly. A smaller number of publications cover practical aspects with case studies on power-aware middleware services and support for the development process of WSNs. Selected extended results are presented in Chapter 4.

# 4 Evaluation and Results of Case Studies and TOSPIE2 Methodologies

This chapter extends results of already published material. A multi-agent middleware extending MAMA [13] and a detailed description of TOSPIE2, including a brief outlook on its extension by its battery model and simulation, extends contents in [1].

## 4.1 A Multi-Agent Middleware for Wireless Sensor Networks

MAMA [13] establishes the basis for running multiple applications within a single WSN. Utilizing TinyOS threading technology and managing networking and sensor resource arbitration, it enables the concept of VOs for WSNs.

An as yet unpublished extension of MAMA implements agents as the VOs' applications. By having their own instruction set, they are interpreted in a VM-like manner by the middleware. Different instructions are available for control flow operation, arithmetic operations and for accessing system calls.

1. Control flow

   **HALT:** halts the agent

   **JMP:** jumps to instruction given by first operand $PC = OP_1$

   **SLEEP:** pauses the agent for $OP_1\,ms$

   **LOAD:** loads the accumulator $A = M\{OP_1\}$

   **STORE:** stores the accumulator $M\{OP_1\} = A$

2. Arithmetic

   **INKR:** increments value of memory location $M\{OP_1\} = M\{OP_1\} + 1$

   **INKRA:** increments the accumulator $A = A + 1$

   **ADD:** $M\{OP_1\} = M\{OP_2\} + M\{OP_3\}$

   **ADDA:** $A = A + M\{OP_1\}$

3. System calls

   **SEND:** sends $M\{OP_1\}$ to $M\{OP_2\}$

   **SENDA:** sends $A$ to $M\{OP_1\}$

   **RECV:** receives and stores value $M\{OP_1\}$

   **RECVA:** receives and stores in the accumulator $A$

   **READ:** reads sensor $OP_1$ to memory $M\{OP_2\}$

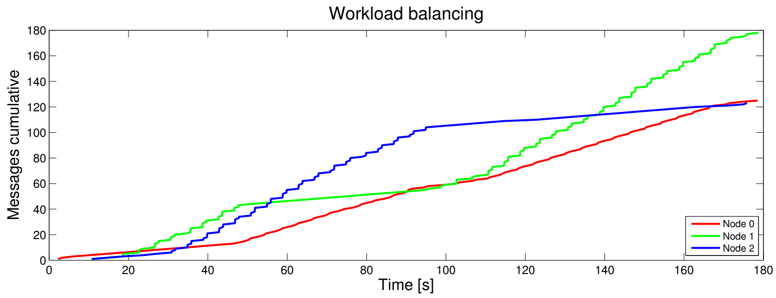



Figure 4.1: Agents are moved for balancing the workload.

**READA:** reads sensor $OP_1$ to accumulator $A$

**SETLED:** displays value $M\{OP_1\}$ with Light Emitting Diodes (LEDs)

**SETLEDA:** displays accumulator $A$ with LEDs

Different tiers of the approach consider the need for low-power and energy-efficient operation.

- The middleware implements means for EHS DC negotiation. The negotiation can either be done in-network or by a host-side instance controlling the network. Energy and activity reports can be queried for gathering the needed information via the same gateway that is used for injecting or commanding agents and gathering sensor data from the WSN.

- For accessing the channel, different LPL MAC configurations can be chosen from. Furthermore, the possible sending periods are slotted for TDMA for accessing available bandwidth. What still needs to be added is test scenarios with full overhearing among all motes with using LPL.

- Based on that timing behavior for the application level DC and for the networking layer, motes can run their agents by interpreting their instructions, and the host-side part of the middleware - operated by a power management console - may be utilized for load balancing among motes employing agent mobility.

- The workload of a mote is mainly determined by the number of active agents running on it. Either by pre-determined rules coded in the agents' memory space or controlled by the console, agents may move or may be cloned. The middleware message for migration can be as short as 7 byte agent header plus a 21 byte long data field. The data field can contain a complete agent with 1 byte VO address 4 byte parameters and a 16 byte instruction field containing up to 8 instructions.

Setting a DC and moving agents in the network, the power management console can balance the load among different motes.

Figure 4.1 depicts messaging load profiles for a part of the network. Two agents are injected with one sending 3 messages and another one sending 6 messages per DC period. In

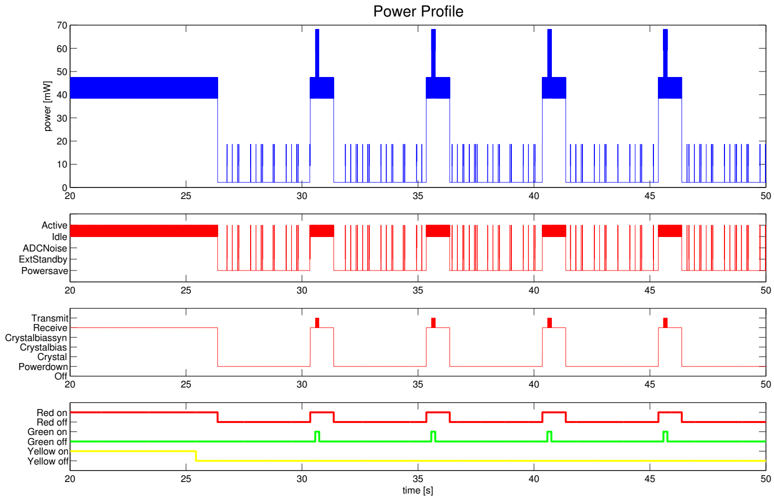



Figure 4.2: Traces of a Mica2 mote PSM are shown. An agent is running on top of the VM, hosted by MAMA, implementing a sensing application.

the event the network's energy resources are not evenly drained, the workload is balanced as shown by the cumulative function in Figure 4.1.

A typical profiling result of the agent-based approach by means of the TOSPIE2 environment is shown in Figure 4.2. The mote's PSM is accurately tracked. Furthermore, LEDs are toggled in this experiment to better analyze whether the timing behavior is correctly working or not by visual inspection of simulation results as well as for hardware runs.

## 4.2 The TOSPIE2 Setup and its Components

TOSPIE2 [1] is built around a power state database, its interconnecting interfaces, an accurate measurement setup, a modified AEON, a proprietary EHS simulation in Matlab and a front-end implemented as Eclipse plug-in. The Simulation and Measurement Interfacer (SiMeInterfacer) setup provides capabilities to remotely control TOSPIE2 components. The console and the user interface can communicate to remote or other local components through the SiMeInterfacer as shown in Figure 4.3.

The power state database holds power and timing values for a platform's power states. In the event a power profile is not flat during a phase, the timing values are used to describe how long these states last and annotate what their average power dissipation is. All data from the database can be written and read from all master and slave components in the system.

The Avrora environment and the National Instruments (NI) setup are extended with debugging output for exact timing measurements of when power states are switched. This setup allows automatically reading a PSM from Avrora and automatically tuning its values with hardware measurement results if measurements show that a value needs correction. The update is simple but powerful. It gets enabled by the PSM timing debug annotations

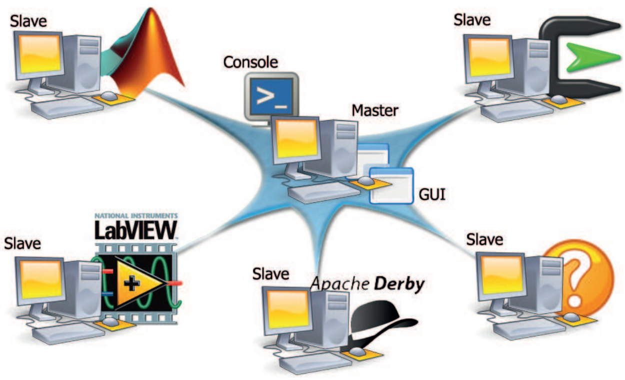



Figure 4.3: System architecture concept for connections of the SiMeInterfacer for connecting TOSPIE2 components.

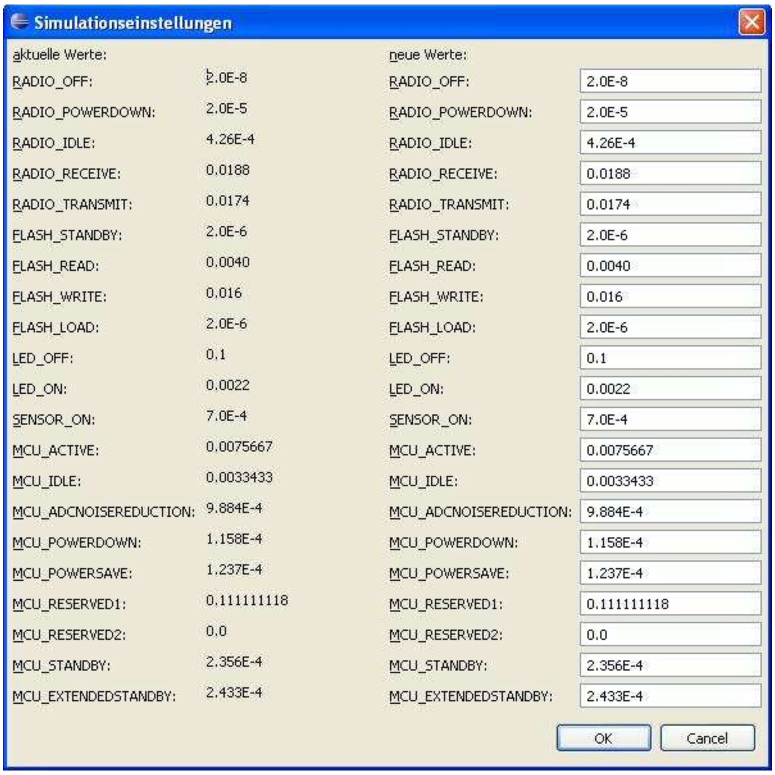



Figure 4.4: An overview of power state values needed for simulation of Mica2 motes.

that are used.

If the graphical user interface is used for comparing simulation and measurement results, the developer can alter the PSM after visual inspection of the traces as well. This can be useful for tuning the accuracy of power profiles of components that need fine grained timing resolution like timers being fired. Figure 4.4 shows the dialog box for altering PSM power values that is presented to the user.

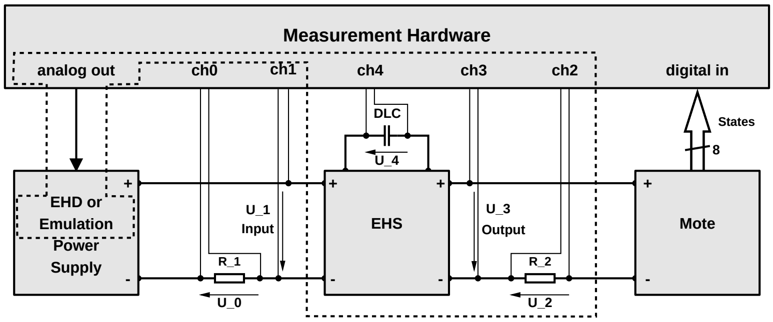



Figure 4.5: The automated PSM measurement setup and the extended setup (dotted area) for EHS efficiency profiling have been used with a laboratory power supply.

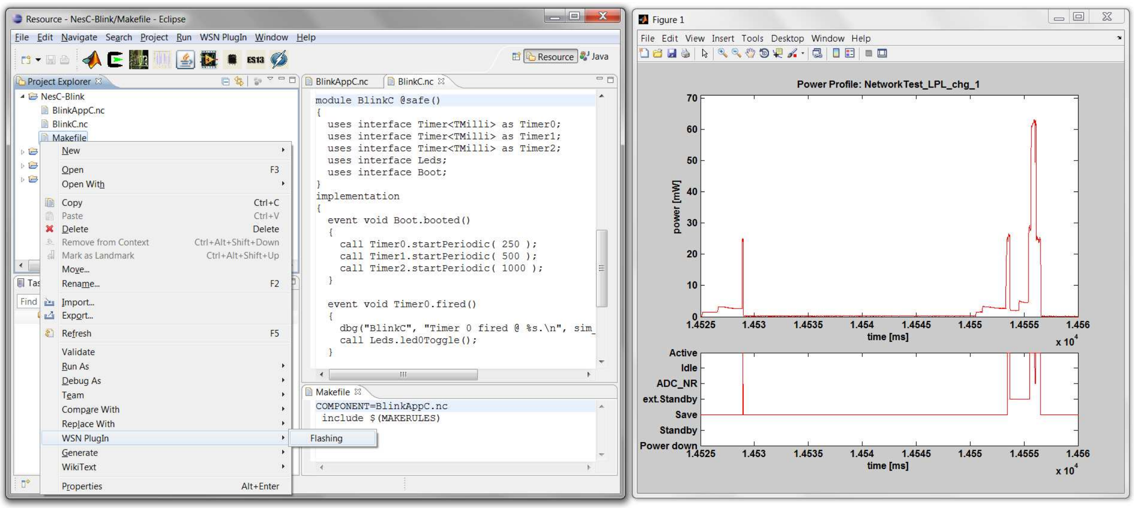



Figure 4.6: The TOSPIE2 Eclipse plug-in user interface

The NI setup employs a digitizer for high speed measurements and a data acquisition card for flexible measurement setups. It profiles up to two motes and one EHS at the same time including digital lines from CPUs for capturing PSM timing debug statements that set or reset single CPU pins with single-instruction statements. Figure 4.5 gives an overview of the measurement system.

While there needs to be at least one master, there can be an arbitrary number of slaves utilizing the same PSM and NI setup. Avrora and Matlab EHS simulation environments and a separate automated TinyOS2 and TinyOS1 installation can be kept locally. Traces of power profiles and all database access invocations can be sent via the network. This allows remote utilization of the measurement setup. This may also give more accurate results, because the measurement setup need not be altered manually. A team of developers designing and testing applications can utilize the TOSPIE2 setup with no need for physically accessing the measurement setup once the hardware has been attached.

The TOSPIE2 front-end allows applying a number of commands from context sensitive menus, from the menu bar and from menu icons as shown in Figure 4.6. As a precondition, an install script needs to be executed. At the initial time of first using it, master or slave roles have to be chosen and a preference page has to be filled out for customizing the plug-in for the developer and its environment.

The commands are as follows:

- The first icon evaluates a hardware power profile using Matlab. The power trace comes up with power state switches annotated. Even under worst case conditions its accuracy is below 2.1% relative error when measuring Mica2 motes. Other state-of-the-art motes can be measured as well.
- Cygwin starts up with the second icon. Registry entries for switching between TinyOS1 and TinyOS2 under WindowsXP can automatically be replaced.
- Simulate a TinyOS program using Avrora. A dialog box comes up that allows including monitors, sets applications to be simulated for single or networked motes and asks for a timing resolution of the power trace. The timing parameter is crucial, because the power trace sampling rate heavily impacts accuracy and performance.
- Evaluate an Avrora simulation run. A power trace comes up similar to that from the hardware measurement with annotated state switches. An example is shown in Figure 4.2. The accuracy of Avrora is limited, because the sampling rate cannot be increased without significant performance drawbacks. Therefore, the evaluation cannot usually be as accurate as a measurement with real hardware being given the same time for its evaluation.
- "Make Jar" allows the rebuilding of Avrora from scratch, e.g. when a new monitor is implemented or an existing one has to be refined.
- The next icon starts the hardware measurement setup. After a warm up period, the real measurement is performed and the power trace is sent to the TOSPIE2 instance that has asked for it.
- The small black icon allows the updating of the PSM manually by the developer using the dialog shown in Figure 4.4.
- ES13 is the discrete event simulator that has been included as a modeling tool for harvesting-aware high-level simulation. It communicates with TOSPIE2's PSM and provides power profiles similar to Avrora. A major advantage is that the timing resolution can easily be adjusted. The minimum sampling rate is that of top level application configuration interfaces' event rates, but it can easily be enhanced by increasing the EHS part's EHD sampling rate at specific times of the profile trading accuracy for performance [1].

Further results achieved with the TOSPIE2 environment regarding battery performance evaluation and modeling are presented and discussed in [16]. Profiling of battery performance characteristics is enabled by TOSPIE2 automation capabilities and high dynamic range of measurement and simulation components.

## 4.3 Extended Results Summary

While a large amount of data has been profiled, representing what can be analyzed, simulated and measured with the approaches discussed above or with other parts on NC in Section 3.1 and on energy harvesting in Section 3.2, this chapter does not focus on discussing measurement results. Rather, its focus lies on providing insights into work that has only been published to a certain extent or that has not yet been published at all. Therefore, the case studies of this thesis are outlined in greater detail here, because minor contributions have had less of a focus regarding scientific dissemination. The overall completeness of this thesis' approach - especially regarding the major contribution for achieving efficient WSNs with NC and EHSs with middleware integration - will be discussed in Chapter 5. These topics are comprehensively covered by publications presented in Chapter 6.

# 5 Conclusion and Future Work

This chapter concludes this thesis with a discussion on scope and completeness of scientific dissemination before contributed publications are included in Chapter 6. Future directions and possible implications for WSNs and related fields and technologies are to be outlined.

## 5.1 Concluding Thesis Scope and Completeness

Within the broad context of WSNs, this thesis establishes TOSPIE2 - a tool suite for WSN development with novel methodologies - and versatile middleware services as well as simulation- and hardware-based profiling approaches, while focusing on the design of efficient WSNs. Following the rule to not disregard side-effects and interplay of different optimizations, novel versatile NC services and robust and efficient EHS designs are established and integrated for WSNs. These parts of major contribution and their achievements that they stand for can well be quantified, while case studies' achievements are better explained by outlining their functional characteristics and concepts.

Discussing the overall concept with a tiered system architecture and different development phases in mind, the work focuses on applicable and generic achievements for fostering complete methodologies and expressive results. This is reflected by the structure of the thesis itself. First, it is embedded in a scientific context and connects power-awareness and energy-efficiency demands to related fields and applications. In particular, MANETs and mapping of WSN research results to smartphones are considered. Second, the necessity of efficient networking and hardware platforms is deduced from a project-oriented view of WSN development and deployment with top-down approaches. The resulting need and contribution claims are then worked on in a bottom-up manner. Energy harvesting concepts are briefly surveyed and efficient strategies are formulated, implemented and tested. Efficient networking using EHAR mechanisms are then implemented and used for comparison of NC approaches. Energy management and service integration approaches - all considering the necessary usage of LPL mechanisms - then further climb up the system architectures. A cross layer intersection for using indoor and outdoor localization services is included as well as measurement and simulation integration at different stages of the design and at different levels of abstraction of the architecture. Energy management and virtualization at the middleware layer further propagates energy-efficiency to the application layer together with a host-based power management console. Furthermore, the TOSPIE2 environment provides the means for offline emulation and simulation of power-aware and energy-efficient strategies. Especially, for exploring different strategies in a laboratory environment, before deploying a characteristic in a network by means of suitable configuration and agent injection, an invaluable tool is provided for running VOs on WSNs.

The major outcomes and achievements regarding the main contribution in the area of NC and EHS enhancements of WSNs are as follows. XOR-coding as NC is introduced for WSNs in different flavors. LNC and ANC allow seamless integration into routing optimization metrics, heterogeneous networks and readily working application scenarios. SNC and ONC show scalable behavior and general applicability with an achievable coding advantage leading to energy conservation between 15 % and 50 %. Integration of energy management schemes with EHS-enhanced WSNs and harvesting service integration has been proven to work in simulation and with hardware experiments. Approaches are of little computational overhead, they need hardly any additional space for code and only little overhead in terms of delay has to be accepted. Robustness to noise and different sending rates, more than sufficient for monitoring and control applications, are profiled to be working for different harvesting protocol configurations as well as for different communication paradigms.

Energy harvesting profiling and characterization along with prototype implementation have been pursued along two directions. A modular EHD measurement and software characterization environment allows gathering experience and serves as a blueprint for the proprietary EHS simulation environment within TOSPIE2. State-of-the-art DC implementations can be tested and implementing thresholds turns out to improve reliability. An application-tailored implementation called RiverMote is a case study for a low-cost efficiency-by-design approach with fault tolerance implemented including back-up mechanisms. Especially, BOS times of up to more than 3 weeks can be achieved with the efficiently implemented energy storage architecture. In-situ measurements of EHD profiles, energy budgets and mote power dissipation of little error in relevant areas are implemented.

Summing up, the following groups of contribution can be identified in this thesis.

- Different NC strategies are developed and applied to WSNs, enabling power-aware networking for WSN middleware systems.
- Energy-efficient EHS hardware prototypes are presented and integrated by means of middleware services.
- The concept of using VOs is realized for running multiple power-aware applications in a single WSN.
- Versatile simulation and measurement tools provide powerful features for developing WSNs when using the TOSPIE2 environment.

## 5.2 Possible Future Directions

While the focus of this thesis is on presenting a self-sustained approach, several issues considered out of scope are not included and several future directions with possible promising outcomes are not yet explored.

Different fields might benefit from NC capabilities. In particular, for implementations tackling the computationally hard problem of finding routing trees and coding rules, usage of different finite fields and random NC methods may be an enabling factor. This may in turn give freedom for achieving better coding advantage as well, although the approach may be less appropriate for being applied to WSNs compared to scenarios of peer-to-peer networking in dense wireless networks of platforms of higher computational power.

Further wireless applications that may benefit from the approach presented here may be smartphones, especially when it comes to streaming applications.

Novel promising approaches include physical layer NC, especially for cognitive radio systems. Instead of exploiting the decodable superposition of logical information flows, physical signals are synchronized and interfered. However, there are no working hardware prototypes available yet. Furthermore, possible integration of NC as presented in this thesis by means of physical layer NC and source coding or compression techniques may be of further benefit for WSN NC and wireless NC in general.

Within the presented types of NC for WSNs, questions remain, that are not yet covered in detail. Additional NC structures might be characterized for their applicability to LNC for WSNs allowing for a greater number of opportunities where energy can be conserved from sending fewer messages. ANC, currently described for quasi-stationary deployments, could be extended to mobile applications and therefore initialization overhead could be considered a promising optimization issue. SNC and ONC currently neglecting the possibility for inter-session NC could be tuned further towards taking that into account.

The connection of NC with harvesting in the form of EHAR may deserve further consideration as well. As current studies on the issue back up the need for novel EHAR considerations, the thesis at hand lets one expect it to be beneficial to incorporate overhearing costs of neighboring motes into a given mote's routing metric. Moreover, what has vastly contributed to the development of WSN routing metrics, publicly available testbeds, could be beneficial for harvesting considerations as well. Little work exists in that area. Extending that and further working towards introducing benchmarking approaches for WSNs, may be a necessary step for establishing large scale deployments and industry strength developments of WSNs.

# 6 Contributed Publications

Publications reprinted in this chapter are grouped into four categories as introduced in Chapter 3. A mapping of all publications included in this thesis is depicted in Figure 6.1. The main contribution regarding NC and harvesting for WSNs is covered by publications 1.1 through 1.6 and publications 2.1 through 2.4 respectively. Application considerations and the TOSPIE2 toolchain are represented by publications 3.1 through 3.4 and publications 4.1 through 4.3. As for other mappings in this thesis as well, sometimes they are ambigious. As an example, publication 3.2 can be considered a middleware service, but it could be considered an example for comparing simulation and measurement results as well. All publications abbreviated with their workshop or conference name as acronym and its year of publication are published through IEEE and ACM and contain copyright statements accordingly. Journal articles are published through EJSE and IJAACS as stated on their reprints. The publications published through IJAACS (publication 1.6) as well as book chapters through IGI Global (publication 4.2) and CRC Press (publication 4.3) are prior to entering the printing stage at the time of writing of this thesis. Preprint drafts are included as accepted by the editor.

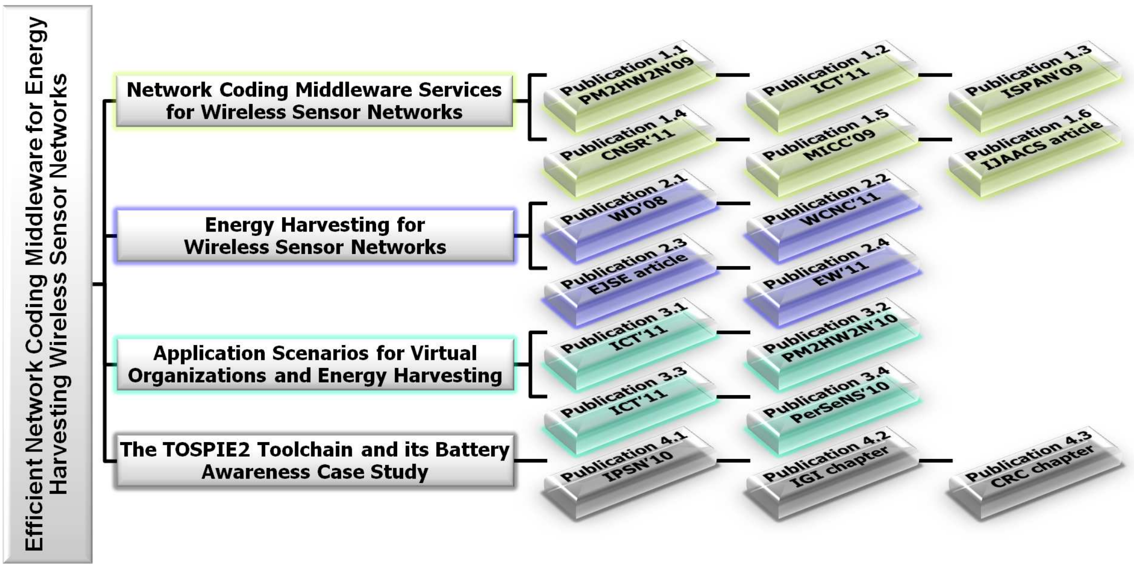



Figure 6.1: The structure of contributed publications is mapped to different aspects of this thesis.

The first group of publications makes up the dissemination regarding the power-aware networking approaches with NC for WSNs as discussed in Section 3.1.

Publication 1.1: P. M. Glatz and R. Weiss, “LINDONCS: Localized In-Network Detection Of Network Coding Structures in Wireless Sensor Networks,” in Proceedings of the 4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks, 2009, pp. 17 – 24.

Publication 1.2: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, “Implementing Autonomous Network Coding for Wireless Sensor Network Applications,” in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 9 – 14.

Publication 1.3: P. M. Glatz, K. B. Hein, and R. Weiss, “Energy Conservation with Network Coding for Wireless Sensor Networks with Multiple Crossed Information Flows,” in Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks, 2009, pp. 202 – 207.

Publication 1.4: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, “Opportunistic Network Coding for Energy Conservation in Wireless Sensor Networks,” in Proceedings of the Ninth Annual Communication Networks and Services Research Conference, 2011, pp. 239 – 246.

Publication 1.5: P. M. Glatz, J. Loinig, C. Steger, and R. Weiss, “A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks,” in Proceedings of the 9th IEEE Malaysia International Conference on Communications, 2009, pp. 905–910.

Publication 1.6: P. M. Glatz, K. B. Hein, and R. Weiss, “Scalable Network Coding for Wireless Sensor Network Energy Conservation,” International Journal of Autonomous and Adaptive Communications, accepted for publication.

Energy-efficient harvesting constitutes the topics of the second group of publications that are mapped to issues discussed in Section 3.2.

Publication 2.1: P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, “A Measurement Platform for Energy Harvesting and Software Characterization in WSNs,” in Proceedings of the IFIP/IEEE Wireless Days Conference, 2008, pp. 1 – 5.

Publication 2.2: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, “Designing Sustainable Wireless Sensor Networks with Efficient Energy Harvesting Systems,” in Proceedings of the IEEE Wireless Communications and Networking Conference - Service and Application Track, 2011, pp. 2018 – 2023.

Publication 2.3: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, “Designing Perpetual Energy Harvesting Systems explained with RiverMote: A Wireless Sensor Network Platform for River Monitoring,” Electronic Journal of Structural Engineering, Special Issue: Wireless Sensor Networks and Practical Applications, pp. 55 – 65, 2010.

Publication 2.4: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, “Low-Cost Reliable Blackout Sustainability of Wireless Sensor Networks with Energy Harvesting Systems,” in Proceedings of the 17th European Wireless Conference, 2011, pp. 155 – 162.

The next group of publications makes up the dissemination regarding the first case study with related middleware approaches and services for WSNs discussed in Section 3.3.

Publication 3.1: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "MAMA: Multi-Application MiddlewAre for Efficient Wireless Sensor Networks," in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 1 – 8.
Publication 3.2: P. M. Glatz, C. Steger, and R. Weiss, "Design, Simulation and Measurement of an Accurate Wireless Sensor Network Localization System," in Proceedings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, 2010, pp. 1 – 8.
Publication 3.3: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "HANS: Harvesting Aware Networking Service for Energy Management in Wireless Sensor Networks," in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 191 – 196.
Publication 3.4: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency," in Proceedings of the Sixth IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, 2010, pp. 468 – 473.

The last group of publications deals with topics concerned with the development process of WSNs supported by TOSPIE2. It makes up the second case study and it is discussed in Section 3.4.

Publication 4.1: P. M. Glatz, C. Steger, and R. Weiss, "TOSPIE2: Tiny Operating System Plug-In for Energy Estimation," in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010, pp. 410 – 411.
Publication 4.2: P. M. Glatz and R. Weiss, "Architecturing Resource Aware Sensor Grid Middleware: Avoiding Common Errors in Design, Simulation, Test and Measurement," in Computational and Data Grids: Principles, Designs, and Applications, N. Preve, Ed. IGI Global, 2011.
Publication 4.3: P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Towards Modeling Support for Low-Power and Harvesting Wireless Sensors for Realistic Simulation of Intelligent Energy-Aware Middleware," in Energy Optimization and Scavenging Techniques for Mobile Devices and Networks, H. Venkataraman and G.-M. Muntean, Eds. Accepted for publication by CRC Press, 2012.

# LINDONCS: Localized In-Network Detection Of Network Coding Structures in Wireless Sensor Networks

Philipp M. Glatz
Institute for Technical Informatics
University of Technology Graz
Inffeldgasse 16 / A-8010 Graz
Philipp.Glatz@TUGraz.at

Reinhold Weiss
Institute for Technical Informatics
University of Technology Graz
Inffeldgasse 16 / A-8010 Graz
RWeiss@TUGraz.at

## ABSTRACT

Wireless sensor network (WSN) motes of small form factor are operated in resource-constrained settings. This demands for low-power designs and energy-aware operation, but especially for optimization of costly wireless transmissions.
A promising approach is to find network structures where network coding can be applied to optimize energy efficiency of information flows. Schemes exist for global optimization of the load on WSN's resources.
Nevertheless, applying them in practical settings is often not feasible due to complex computations and the need for centralized knowledge. The gain from optimization methods fades away for scaled and autonomous operation of WSNs due to control overhead.
This paper presents LINDONCS: Localized In-Network Detection Of Network Coding Structures. Communication patterns are adapted autonomously without paying off impact on network scalability. We show how to detect structures and apply network coding to improve network lifetime over state-of-the-art solutions.

## Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*network management, network monitoring*

## General Terms

Algorithms, Management

## Keywords

Energy-Awareness, In-Network Detection, Network Coding, Network Information Flow, Power-Awareness, Wireless Sensor Networks

## 1. INTRODUCTION

Wireless sensor network (WSN) motes have to operate under tight constraints concerning their power supply and have to cope with limited bandwidth. Energy conservation and saving of bandwidth are tightly correlated. So, both aspects have to be considered: power-aware behavior on a per-mote basis and energy conservation on a network level. Methods have to be found to optimize both sides so that they do not impact one another negatively.

### 1.1 Contribution of this Paper

This paper follows a simulation based approach. We work under a setup that considers WSN constraints on a per-mote and a per-network basis. A simulation testbed is configured based on information taken from measurements of single motes' low-power methods. Novel network-wide measures are designed, simulated and analyzed with these low-power methods in mind and provide the following contribution.

- Piggybacking: energy-aware routed packets will be exploited to exchange information among motes that allow deducing the existence of network coding structures. A ground truth has to be chosen to compare against. We chose energy-aware routing and show how to improve network lifetime with network coding.
- In-network detection: a method will be given that can check if the evaluating mote is part of a network coding structure and decode its role that it has to implement. This will be done for the famous butterfly example.
- Converter functions: When a message enters an area with network coding from a routed area, the messages have to be converted accordingly. LINDONCS uses converter functions to translate from one communication paradigm into another one autonomously without impacting scalability.
- Implementation and evaluation: We show a lightweight implementation. So it can easily be implemented as part of a middleware on WSN motes with limited computational power. LINDONCS does not put constraints on system memory. For evaluation of the network coding approach we will compare network lifetime to simulation runs when energy-aware routing is used.

### 1.2 Outline

The design and implementation of networking abstractions have to be customized according to technical constraints of WSN hardware and the underlying media access protocol (MAC) on the one side and application needs and the support of a common, network-wide goal on the other side.

We will first discuss adjacent levels of abstraction under the aspect of a WSN's energy balance. We will then introduce network coding as a level of abstraction in between. For quantifying the effects of network coding we will use a second approach on the same level of abstraction that is optimized towards conserving a mote's energy balance too.
The second approach will be introduced by related work in the area of energy harvesting routing. Apart from deducing a state-of-the-art metric for energy-aware routing we will discuss related work for the theory of network coding.
The concept section shows the idea of how to detect a structure of crossing information flows. Furthermore it will be shown how to correctly attract these flows to enter the structure in a way such that the network coding approach is applicable.
The implementation will allow parameterizing the simulation in ways to evaluate it for realistic scenarios where results are applicable and can help find decisions for future designs. The configuration can be used to fit a given hardware's characteristic of how the usage of a mote's radio module impacts its energy balance.
The sections on results, conclusion and future work give a feeling on in how far the methods presented here are applicable for a given setting and how they will further be developed.

## 2. THE ENERGY BALANCE OF WIRELESS SENSOR NETWORKS

Starting from the vision of smart dust by Warneke et al. [16] WSNs have always been subject to deal with low-power issues. They run their applications and solve their problems for constraint settings that were set to further push technologies and develop optimization methods on top of it. Therefore the field of research that is concerned with WSNs spans a wide range of issues. Akyildiz et al. give a good overview [18] of different horizontal and vertical boundaries of these different fields. This paper will first introduce the technological constraints starting with the possibilities of WSN's power supplies. Based on that, we will develop a networking pattern for fitting the application's communication needs to what can be delivered in terms of performance per energy by the hardware. This implicitly defines our work to be situated at the middleware layer according to [18].

### 2.1 Limited Power Supply

Ubiquitous WSN motes are devices of small form factor. Their size mostly depends on their power supply. State of the art motes like the Mica and Telos family of motes are described by Hill et al. [6] and Polastre et al. [12]. They are typical examples for printed circuit boards and battery packs fitting the size of two AA batteries. Furthermore these are good examples for testing and implementing power-aware computing and communication hardware and furthermore energy-aware system design. A prominent example that uses energy harvesting for Mica 2 motes is the Heliomote [11] platform.
Using Energy Harvesting Systems (EHSs) takes the issue one step further from a power-aware method's view to a system view of energy-awareness. For that case it is especially important to consider aspects where measures taken at a given point in time may impact the system later on. An example has been shown for predicting forthcoming modeling periods [4] for energy harvesting duty cycling [8] and issues associated with that.

### 2.2 Per-Mote and Per-Communication Power Dissipation

On the one side power-awareness for a single-mote view can be considered with a mote's sleep periods to run a mote in a low-power mode or put it to sleep if there is nothing to do. On the other side it might be concerned with scheduling the usage of hardware components such that it works power-efficient in that it saves wake-up times of components in that it schedules different task's access to hardware drivers. If they are accessed by different tasks the requests might be buffered such that they are performed without the need of putting a component to sleep and wake it up again in between. One could also combine the requests to a compound one. What is possible or not also depends on the type of hardware or driver that is available. A detailed discussion on concurrency and energy for different types of driver implementation is given be Klues et al. [10]. It would pose no problem to combine the reading of a temperature value, but it might already give conflicts in the configuration of a microphone and it would most probably not be possible to combine requests of sending data via the radio.
The examples given before shall give the reader a feeling for what the authors try to explain when talking about single mote issues on the one side and network wide optimizations on the other. The first one is usually tackled with optimizing a driver implementation and the latter one is an issue of implementing functionality or behavior in a middleware system. The examples from above tried to take the view from a single resource to a system wide one for a single mote. While it seemed not possible to share the access to a sending radio - although it would be very beneficial in terms of energy savings - we will further raise the abstraction to a network level where it will become possible to share radio sending capabilities for different network information flows using network coding [1] as initially introduced by Ahlswede et al., but here we will consider the wireless case.

### 2.3 The Residual Energy Level and Network Lifetime

The energy consumption of sending a single bit of data is orders of magnitude more costly than processing it within a single mote. Therefore, strategies have been developed for running mote radios in low-power operation.
**Low-Power Listening**: The transmission of data among motes in a WSN can be even more costly at the receiver than what it is at the sender side. Although it takes a long time for sending a message at a high level of power dissipation, the reader has to periodically sense the channel. If there is a transmission being indicated by its preamble, then the receiver stays awake. Otherwise it will go back to sleep again after a short period of time to save energy. Listening causes a similar power dissipation level at the receiver as does the action of transmitting the preamble and the - compared to the preamble often very short - data at the sender. MAC protocols, network topologies and application scenarios may lead to a different additional mean contribution of the radio to the overall mote power dissipation. Furthermore, the ratio of mean sending power dissipation to mean receiving power dissipation can obviously not be seen as a

fixed value. Generally speaking, many techniques for power conservation rely on trading the activation time of different power states against each other. As far as realization of a method in hardware is concerned, this boils down to driver implementation and scheduling. An Insight into the topic for TinyOS is given in [10]. Low-Power Listening (LPL) is considered too.
Nevertheless, this paper does not put its focus on MAC design, but MAC can pose severe constraints to the scenario. So its main purpose has to be understood in principal is a lower-layer abstraction boundary. The upper layer will be the application layer, while we will be considered with networking and the possibilities of network coding.

## 3. NETWORK CODING

Network Coding relaxes the need for sending bits sequentially within a packet or several packets after each other. Messages are combined and can be coded instead of just being forwarded. The gain of coding over routing can go up to a factor of two for the general wireless case. In our case this will be used to achieve relaxation of constraints on connection bandwidth. One might think of a setting where two motes send their data to a relay mote. If routing is used, then the relay mote has to use twice the bandwidth for forwarding the packets. If this is not possible it will in turn reduce the rate at which its predecessors are allowed to send. For looking at the problem from a general point of view we will look at the case of motes that are driven by energy harvesting systems . These systems have been looked at in prior work [4]. They set a duty cycle rate to achieve the maximum sustainable duty cycle. As discussed by Kansal et al. [8] this is used to achieve the maximum end-user performance. In our case we will look at WSN duty cycling for using EHS as the application itself. On the one side it is the rate limiting factor, because motes are simply shut down between two periods of activation. On the other side, control messages have to be exchanged to negotiate the maximum sustainable duty cycle. So we get a communication pattern for free.
Under these generally declared settings we will try to identify structures which allow for energy conservation through efficient bandwidth usage with network coding.

### 3.1 Identifying Structures

Network coding cannot be applied in all parts of a network without careful consideration if it is (i) applicable at a given structure without impact on functionality, (ii) a gain for the metric - in our case it is lifetime - that is to be optimized can be achieved and (iii) an embedding of the method at the structure into the network is possible without effects on the rest of the network.
Furthermore, for tailoring a method to being applicable in practical settings one has to design it in a way such that it is scalable. Impacts on scalability can by prohibited with using an algorithm that works in a decentralized manner. We are going to present an approach that can be implemented as a middleware system which does not need centralized knowledge of the network topology.

### 3.2 Issues with Network Coding

Network Coding is a topic that has mostly been covered theoretically and mostly for unidirectional wireline networks.

**Bidirectional Communication**: Bidirectional communications links have to be considered for the wireless case. Although this is not always true in reality it is an acceptable simplification that is not too unrealistic - it holds for reliable MAC designs - on the one side and greatly eases deduction of a concept on the other side. The drawback is that bidirectional links have a bounded gain of 2 for network coding. It will be the task of this paper to translate this gain into energy conservation.
**Practically Feasible**: Although it has already been mentioned that a gain over routing can be achieved if coding is used and messages are combined, it has not yet been considered that a main reason for its importance are the polynomial algorithms for finding codes compared to the NP-hard problems for subgraph selection in routing. So we will present an approach that is feasible to be implemented on platforms of reduced computational power and limited bandwidth [6, 12].
**Mitigating Knowledge Centralizing**: For not introducing assumptions that have to hold for having the approach being applicable we do neither demand for centralized knowledge of the topology nor do we need prior initialization. We allow starting with energy-aware routing and piggyback information on routed message's semantics which allows to detect if network coding is applicable in a given region.

### 3.3 Characterizing Power-Aware Systems

It is difficult to make objective comparisons of characterization of middleware systems for their power-awareness as long as the application that they shall be used for is not being given. Studies that compare different middleware systems as from Hadim et al. [5] have shown that different approaches can be power-aware for their application domain but would be inefficient in other settings. For being able to cope with that problem we have found a generic metric in duty cycled WSNs. On the one side the energy balance of EHS driven WSNs gives a measure of how much an implementation drains the system's resources and on the other side a bandwidth is given by the rate of the duty cycling. Furthermore systems operated that way have special needs on communicating their sustainable duty cycles among all nodes. So a given communication pattern has to be applied anyway and we do not behave different for variable applications as they have to fit the EHS' needs and adapt to it. Furthermore we will be able to use energy-aware routing - as it is proposed for such systems [8] - as a ground truth for routing against which we will compare network coding.

## 4. COMPARING NETWORK CODING TO ENERGY-AWARE ROUTING

This section presents previous work in the field of energy saving mechanisms for WSN networking. Related work for EHS, MAC and middleware design will be kept out of this section. Parts of the LINDONCS concept that deal with these topics are out of scope of this work, where we are mainly concerned with identifying subgraphs in WSN topologies and evaluate the energy-awareness of network coding for that process.
Instead, we will discuss previous work in the field of network coding and see how it is related to WSNs. The concept of network information flow [1] first targets point-to-point connections. Based upon that, strategies have mainly been

developed for high data rate networks. For LINDONCS we have to consider other attributes.
For its evaluation we need some comparison. We will use state-of-the-art routing approaches. Therefore, we first discuss energy-aware routing in WSNs.

## 4.1 Energy-Aware Routing

Energy-aware routing is a technique for optimization of WSN motes' energy balance based on their current state that has evolutionary developed from past events and decisions.
As an example for energy-aware routing we will use an approach for harvesting-aware power management. This extends the approach from power-aware routing to including information on future conditions, namely energy harvesting device (EHD) output power traces.

## 4.2 Towards Energy Harvesting Aware Routing

Energy-aware routing is taken from Shah et al. [14]. This boils down to choosing a route not only dependent on which is the shortest multi-hop connection to the receiver, but also - e.g. if there is a tie - on which is the most resilient path based on a metric from Chang et al. [2]. They calculate link cost $c_{i,j}$ from mote $i$ to $j$ dependent on the expected unit flow transmission expenditure $e_{i,j}$, the initial Energy $E_i$ and the remaining transmitter energy $\underline{E_i}$ as shown in equation 1.

$$c_{ij} = e_{ij}^{x1}\, \underline{E_i}^{-x2} E_i^{x3} \qquad (1)$$

The vector $\{x1, x2, x3\}$ can be used for weighting the routing on top of the metric. For [14] the neighbors stored in a mote's forwarding table $FT$ are assigned values according to communication costs as can be seen in equation 2.

$$P_{N_j,N_i} = \frac{1/C_{N_j,N_i}}{\sum\limits_{k \in FT_j} 1/C_{N_j,N_k}} \qquad (2)$$

This time the path cost was defined as given in equation 3.

$$C_{N_j,N_i} = Cost(N_i) + Metric(N_j, N_i) \qquad (3)$$

As mentioned before the scope of a function's knowledge for choosing an energy efficient path is extended for energy harvesting power management by Kansal et al. [7]. The weighted function that is dependent on the expected average input power profile $\rho_i$ for node $i$ and the remaining battery level $B_i$ in equation 4.

$$E_i = \omega * \rho_i + (1 - \omega) * B_i \qquad (4)$$

The weighting factor $\omega$ can be set to 1 for solely considering power from EHDs and to 0 if we are just working with batteries. The edge cost can then be written as in equation 5.

$$c_{ki}(e_{ki}) = 1/E_i \;\forall k \in \{k | e_{ki} \in E_{comm}\} \qquad (5)$$

The contents of the references in [2, 14, 7] have been reviewed and equation 5 will be used as a cost metric in the concept section of this paper. In a first step presented here we will neglect EHD power traces and set $\omega$ to 0.

## 4.3 Network Coding

For discussing network information flow for WSNs we will start with the general case. This allows setting up the nomenclature and defining what coding and decoding is.

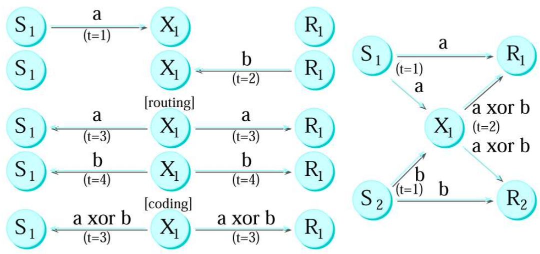



**Figure 1: S and R exchange information: The example on the left hand side shows mutual exchange for routing and network coding while a single coded information flow is depicted on the right. Network coding saves one communication time slot at $X_1$**

This will give us all means deducing the possible coding gain which we will later translate into energy efficiency.
Network coding has first been introduced and described for directed, acyclic graphs. Novel related work describes what it takes to approach the wireless case. We will also consider the aspects that cannot be covered with existing theory like incorporation of MAC behaviour.

## 4.4 Network Information Flow

We will consider network coding functions as they have been described in [1] and have got practical considerations by Fragouli et al. [3]. The method shall be explained with figure 1. There are two cases where $S$ and $R$ exchange information via $X$. The example on the right shows a single crossing of information flows in a butterfly structure and the example on the left shows a flow forth and back. In both cases we reduce the time needed for exchanging the messages and the number of messages that have to be sent.

### 4.4.1 Coding Gain

Dependent on the network model in use different gain can be achieved in terms of throughput advantage - which we will map to energy savings through a reduced number of messages to be sent. The advantage for lossless multicast network coding can be $\Omega = ((\log n / \log\log n)^2)$. Neither would it be possible for us to improve over the results that are summarized by Ho et al. [15] as it is out of scope and poses a hard problem nor do we claim to optimize the conditions under which we are working. We are - as has been indicated in the outline of this paper - much more concerned with how to cope with the constraints given.

### 4.4.2 Opportunistic Coding

It has been shown by Yunnan et al. [17] for large scale wired networks that there exist distributed schemes for network coding that do not need central authoring for running network coding. They deduce that selection of the message buffering schemes for opportunistic network coding [13] deserves decent consideration. This is especially important if the delay introduced for network coding to work under different conditions is an issue.

### 4.4.3 Wireless Sensor Networks

As pairwise xor coding in wireline networks xor coding is also applied in wireless networks for bidirectional links by

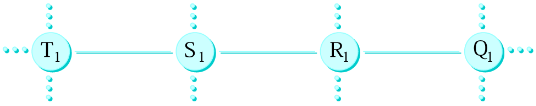



**Figure 2: A single communication channel.**

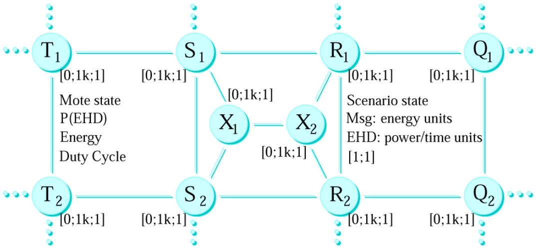



**Figure 3: A butterfly structure within a mesh network.**

Katti et al. [9]. One has to keep in mind that the channel capacity has to be modeled differently now that MAC protocols have other characteristics than in wireline networks.

## 5. CONCEPT OF LINDONCS

The previous section presented energy-aware routing and network coding for the general case. We will now show the setup of substructures for a simple case of nodes with a maximum degree of 2 in figure 1. Network coding can be applied as messages $a$ and $b$ meet at $X_1$. For forwarding the messages to the other side it takes two additional time slots with routing and a single one if coding is applied and bandwidth is fixed at one message per time step. For a butterfly structure as in figure 3 embedded into a mesh network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with information flows from left to right we have twice the structure of the simple case of a flow $\mathcal{G}_{channel}$ as depicted in figure 2 crossed in $\mathcal{G}_{butterfly}$. If they had not been crossed they had to be considered separately under the precondition that these two structures are to be recognized. This should point out that the amount of structures to be recognized has impact on the effort for recognizing these. In this work we will consider only the two structures pointed out above. For information flows from left to right as in our test case we will only evaluate the butterfly structure.
Although 3 nodes would suffice for capturing network coding considerations of $\mathcal{G}_{channel}$ and 5 would suffice for $\mathcal{G}_{butterfly}$ we have introduced an additional node to allow for independent duty cycle rates being run on both sides of the structure. This will allow extending $\mathcal{G}_{butterfly}$ in future developments to EHD rate adaptation and opportunistic coding while having both ends of the structure decoupled at $X_1$ and $X_2$. A mote's energy reservoir will be drained whenever it is involved in a transmission. We will first show how to setup and run energy harvesting aware routing built upon equation 5. We will then show how to setup and run the network coding approach.

### 5.1 Energy Harvesting Aware Routing

The cost metric will be implemented in a straight forward way - as it will be for network coding too - which on the one side will impact performance, but on the other side will make it more feasible for implementation on platforms of low computational power.
We will first collect the information that has to be exchanged between nodes. Then we will show state machines that explain how to setup and run the methods. So it will be possible to compare energy harvesting aware routing to network coding for energy harvesting WSNs for which we will do the same. Main evaluation criteria will be the setup time and the number of messages that need to be sent. Accordingly, the running system will be compared with the number of messages to be sent and the residual energy level and overall network lifetime.

**Table 1: Symbols for and descriptions of information to be exchanged for energy harvesting aware routing.**

| symbol | description |
|---|---|
| $N_i$ | mote numbered $i$ |
| $n$ | number of nodes |
| $B_i$ | $N_i$ remaining battery level in unit energy packets |
| $\rho_i$ | $N_i$ EHD power trace in unit energy per unit time |
| $\theta_i$ | $N_i$'s state (e.g. mote initializing) |
| $\mathcal{F}_{N_i,N_j}$ | information flow from $N_i$ to $N_j$ |

**Table 2: Possible network states $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$.**

| $\theta_i$ | name | purpose |
|---|---|---|
| 1 | mote initializing | build up routing table |
| 2 | harvesting-aware routing | apply energy conservation |
| 3 | network coding mode | attract flows to butterfly |

#### 5.1.1 *Transmission of Routing Information*

See table 1 for the information to be exchanged. Table 2 assigns meanings to different states that the network may transit into. The network starts in state $\Theta = \{\theta_i = 0 | i \in \mathcal{N}\}$. We will first consider the network $\mathcal{G}_{butterfly}$ which starts with flooding through broadcasting for $n$ time steps such that all nodes can be reached. Detailed settings are given in the implementation section.
When the network is in state $\Theta = \{\theta_i = 1 | i \in \mathcal{N}\}$ - which need not be communicated among nodes, because the network size is known to the nodes ($|\mathcal{N}| = n$) - we will forward messages to nodes so that we lower the distance to the sink or stay at the same distance and move the message to a lower-cost mote.
The shorter distance can easily be evaluated. While exchanging energy balance information the time-to-live is set such that a routing table to the sink can be built. We do not optimize this step as it does not influence the comparison of routing and coding. The information flow goes from left to right in figures 3 and 2.

### 5.2 Network Coding Concept

Plain, static xor coding will be used without buffering schemes. The knowledge to be assumed when the network is starting will be the number of nodes $n$ in the network and their identification number. Nothing more is needed for establishing a static MAC protocol where node $N_i$ is allowed to transmit during the $i$-th MAC slot if the mote is awake in that communication cycle. Our model assumes that sending has cost of an integer multiple of the receiving cost. The cost will be fixed for a single evaluation run. This does not exactly reflect reality, but the work presented here could be mapped to any scenario and would have to be configured accordingly then. The authors assume for the case given here that the EHS provides an average power supply that varies only slightly from the average duty cycled mote and LPL power dissipation. The variance can be small in reality compared to the additional average power dissipation due to radio usage. The authors are aware of the fact that the variance would scale with varying duty cycles of the motes, but incorporation of the effects into the model would complicate it without providing additional insights into this special field of research.

## 5.3 Detecting Network Coding Structures

For detecting network coding structures we search for crossing flows of information. This allows do apply localized methods. We build up a butterfly structure starting with its inner nodes ($X_1, X_2$ in figure 3). Energy-aware routing protocols (with their metrics based on cost calculation similar to equation 5) will switch routes to burden motes' energy budgets evenly distributed with minimal cost for the overall path. This leads to leaving paths unselected in dense parts of a WSN - due to increased per path cost for more hops. Nevertheless, especially these regions provide additional resources and will more probably allow for using them efficiently with network coding.

Our method is to check for flows that bypass the butterfly's inner nodes on both sides without intersecting them. In figure 3 there are two flows $\mathcal{F}_{T_1,Q_2}$ and $\mathcal{F}_{T_2,Q_1}$ which will trigger the initialization of network coding if both flows have been detected to pass by on both sides of $e_{X_1,X_2}$: $\theta_{X_1}(t) = 3$ and $\theta_{X_2}(t) = 3$ *iff* the nodes have overheard the flows prior to time $t$ and the routing initialization has finished before. Packets need not be translated when they leave routed areas into coded areas or vice versa. The routing metric is known throughout the network. So $X_1$ and $X_2$ instruct adjacent nodes to advertise for incoming connections. In our case they attract for being used by the information flows through pretending higher energy levels. Incoming converter motes ($S_1$, $S_2$) will then judge according to the real energy value which mote will be the next hop of an information flow and allow for an additional hop to be taken.

The case of energy harvesting as considered for routing in equation 4 will not be considered here (we set $\omega = 0$) due to limited space. It will be included in a follow-up paper. Furthermore we will examine the complexity of detecting other structures for network coding.

## 5.4 Evaluation

For evaluation of the structure finding mechanism and network coding we let the same test setup from figure 3 run once with energy harvesting routing alone and will then integrate network coding. The results section will then compare different factors of sending cost to receiving cost ratio settings for the fixed network coding method. This allows giving the reader a feeling for how to compare the novel approach to possibly already known methods - even if it is slightly generalized and then varied in this work. Probabilistic and opportunistic network coding methods would go beyond the scope of this paper so they will not be considered here. No more than one message per source is buffered at any time.

**Algorithm 1** Testbed Operation

```
function Run
    snd_rec_ratio ← 1...8
    initial_energy ← 1000
    sensing_interval ← 5
    stop_condition ← FIRST_NODE_DEAD
    eval_data ← Routing(ratio, energy, interval, cond)
    eval_data ← Coding(ratio, energy, interval, cond)
end function

function IterateTestbed
    for mote ≡ 1...Size(WSN) do
        PhysicalBroadcast(mote)
        time ← time + 1
    end for
end function
```

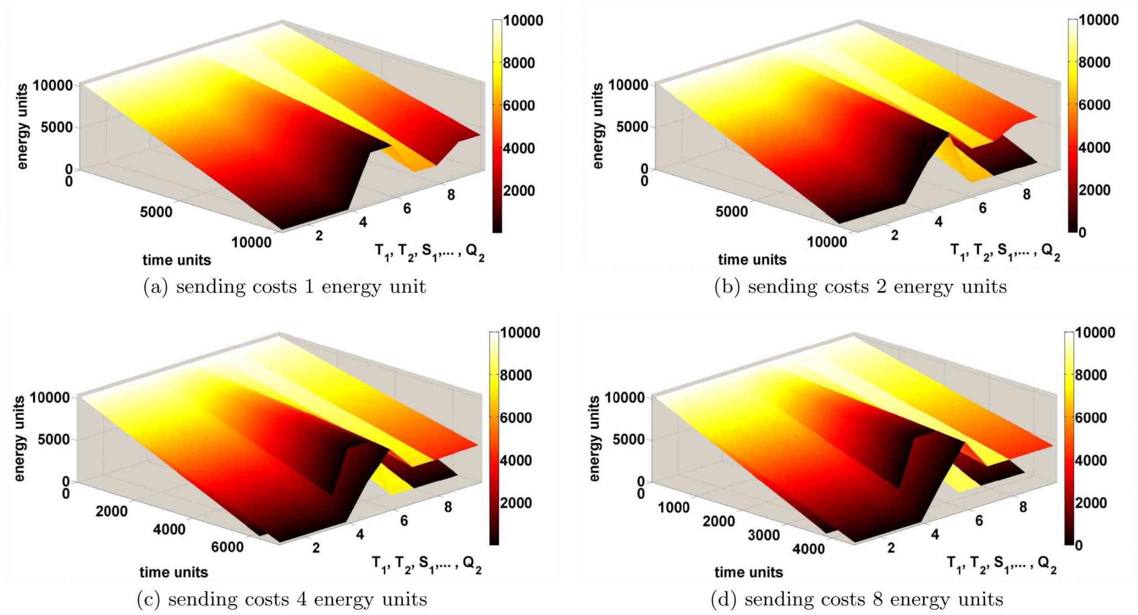
**Algorithm 2** Energy-Aware Routing

```
function Routing
    while Θ ≠ stop_condition do
        if time ≡ STARTUP then
            Θ(Q_1, Q_2) ← FLOOD_ENERGY_INFO
        end if
        if Size(WSN) ≥ time then
            Θ ← ROUTE_INITIALIZED
        end if
        if Θ ≡ ROUTE_INITIALIZED then
            if data then
                Forward(data, MIN_DIST, MAX_E)
                DeleteLocal(data)
            end if
        end if
        IterateTestbed(WSN)
    end while
end function
```

**Algorithm 3** Butterfly Network Coding

```
function Coding
    Routing
    if Θ ≡ ROUTE_INITIALIZED then
        if Via(F_{T_1,Q_2}, (S_1, R_1)) then
            Set(x_11)
        end if
        if Via(F_{T_1,Q_2}, (S_2, R_2)) then
            Set(x_12)
        end if
        if Via(F_{T_2,Q_1}, (S_1, R_1)) then
            Set(x_21)
        end if
        if Via(F_{T_2,Q_1}, (S_2, R_2)) then
            Set(x_22)
        end if
        if X then
            Θ ← CODE_INITIALIZED
        end if
    end if
    if Θ ≡ CODE_INITIALIZED then
        if mote ≡ X_1 then
            SetAttractor(S_1, S_2)
        end if
    end if
end function
```

**Figure 4: Different ratios of sending and receiving costs lead to different system behaviour due to rules implemented in the routing protocol. For the ratio of 2, 4 and 8 we see two separate planes. The simulation run with network coding stays at a higher residual energy level.**




(a) sending costs 1 energy unit
(b) sending costs 2 energy units
(c) sending costs 4 energy units
(d) sending costs 8 energy units

# 6. IMPLEMENTATION

The testbed is run with variable sending to receiving cost ratio and fixed initial energy and interval for sending sensed data as can be seen in algorithm 1. The result section will explain why different ratios have to be used. The testbed is then run with algorithm 2 which implements energy-aware routing and with algorithm 3 which implements network coding.

## 6.1 Implementation Remarks

For network coding we consider both algorithms to be run in parallel. This is indicated by calling routing from within the coding function. Further operations are self-contained. If a reader is interested in rerunning the tests it is important to consider the statically assigned MAC-slots in the order $T_1$, $T_2$, $S_1$, ..., $Q_1$.

Implementing the algorithms does neither put special constraints on computational power of a platform nor do we need to provide much memory for static or dynamic memory allocation.

When $\theta_{X_1} \equiv 3$ that mote will forward messages only if data of both information flows is pending. With *sensing_interval* set to 5 no more than one message per source has to be stored while old messages will always be processed before new ones arrive for steady information source rates.

## 6.2 Test Case Implementation

Tests have been implemented and evaluated such that the attracting behaviour of energy rich areas can be seen. Every *sensing_interval* time steps two messages are injected in the source nodes ($T_1$ and $T_2$) of the butterfly.

**Table 3: Network lifetime and messages sent.**

| cost factor | type | lifetime | msgs snd/rec |
|---|---|---|---|
| 1 | routing | 9955 | 19896/55708 |
| 1 | coding | 9955 | 19896/55708 |
| 2 | routing | 9035 | 18059/50566 |
| 2 | coding | 9035 | 14441/40684 |
| 4 | routing | 6166 | 12320/35576 |
| 4 | coding | 6865 | 10405/29782 |
| 8 | routing | 3780 | 7549/22194 |
| 8 | coding | 4415 | 6405/18729 |

# 7. RESULTS

Subfigures of figure 4 compare standalone energy-aware routing with the integration of network coding according to algorithm 3. Table 3 summarizes the final state of these figures in terms of network lifetime and messages sent and received. Table 5 summarizes the ratio of lifetime improvement and residual energy of network coding over routing. Table 4 shows that the network coding initialization time is a function of the surrounding motes' routing protocol, sending cost and power states values.

In subfigure 5(a) we do not see any difference in network lifetime. Both surfaces overlap completely. As we can see in table 4 network coding is not being initialized for such a setting where sending is of the same cost as receiving. The motes in the middle are never chosen, but they overhear the communication that passes by.

In subplot 5(b) we still have the same network lifetime for both approaches, but table 4 tells us that network coding has

**Table 4: Network coding starting time.**

| cost factor | initialization time |
|---|---|
| 1 | never |
| 2 | 110 |
| 4 | 45 |
| 8 | 25 |

**Table 5: Network coding gain over routing.**

| cost factor | lifetime | residual energy |
|---|---|---|
| 1 | none | 1 |
| 2 | none | 2.3357 |
| 4 | 11.34 % | 1.9315 |
| 8 | 16.80 % | 1.7734 |

been used. For doubling the cost for sending over the cost for receiving the approach starts to work. Table 3 shows that the overall number of messages that have to be sent and that are heard are reduced with network coding. Our definition of network lifetime does not consider the overall energy budget. We are rather testing for uniformly distributed strain of energy resources. Other metrics would most probably report a goodput for this case as it can be seen for motes in the information flow backend that a considerable amount of overall residual energy has been conserved.
Finally in subfigure 5(c) the positive effect of energy conservation starts to feed through to the source motes such that network lifetime is prolonged for 11 %. In the latter case of subplot 5(d) with a cost factor of 8 instead of 4 the lifetime gain increases to 16 %.
The different factors with their gain in lifetime and residual energy are shown in table 5.

# 8. CONCLUSION AND OUTLOOK

This paper has presented LINDONCS: Localized In-Network Detection Of Network Coding Structures. The proposed contribution has been demonstrated and some issues have been motivated that will be part of future work.

## 8.1 LINDONCS Contribution and Attributes

We have shown the feasibility of implementing network coding on small-form-factor platforms for large-scale networks. The performance of identifying and running network coding structures has been compared to energy-aware routing protocols and a lifetime improvement of up to 16 % has been shown. The approach has no negative impact on scalability and can be implemented on autonomously working motes. It can be run on top of energy-aware routing. Converting between routed and coded information flows is implicitly achieved with LINDONCS.

## 8.2 Future Work

The aspect of network coding for EHD rate adaptation has been mentioned to be part of future work for the setup presented in this paper. Furthermore a detailed analysis of different structures and their chances of being coded is part of ongoing work. Both issues will be covered in follow up papers to this one. Next steps will be to provide the basis for experimentation and simulation targeting large scale structures with opportunistic behavior in LINDONCS.

# 9. REFERENCES


[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, Jul 2000.

[2] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM 2000*, volume 1, pages 22–31 vol.1, 2000.

[3] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network coding: an instant primer. *SIGCOMM Comput. Commun. Rev.*, 36(1):63–68, 2006.

[4] P. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss. A measurement platform for energy harvesting and software characterization in WSNs. In *IFIP Wireless Days*, Dubai, UAE, Nov 2008.

[5] S. Hadim and N. Mohamed. Middleware for wireless sensor networks: A survey. In *Comsware 2006*, pages 1–7, 0-0 2006.

[6] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization, 2001.

[7] A. Kansal, J. Hsu, M. Srivastava, and V. Raqhunathan. Harvesting aware power management for sensor networks. In *43rd ACM/IEEE Design Automation Conference, 2006*, pages 651–656, 0-0 2006.

[8] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *ACM Trans. Embed. Comput. Syst.*, 6(4):32, 2007.

[9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3):497–510, June 2008.

[10] K. Klues, V. Handziski, C. Lu, A. Wolisz, D. Culler, D. Gay, and P. Levis. Integrating concurrency control and energy management in device drivers. *SIGOPS Oper. Syst. Rev.*, 41(6):251–264, 2007.

[11] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava. Heliomote: enabling long-lived sensor networks through solar energy harvesting. In *SenSys 2005*, pages 309–309, New York, NY, USA, 2005. ACM.

[12] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *IPSN 2005*, pages 364–369, April 2005.

[13] S. Katti, D. Katabi, W. Hu, H. Rahul and M. Medard. The importance of being opportunistic: Practical network coding for wireless environments, September 2005.

[14] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *WCNC 2002*, volume 1, pages 350–355 vol.1, Mar 2002.

[15] Tracey Ho and Desmond S. Lun. *Network coding: an introduction.* Cambridge, New York, Cambridge University Press, 2008.

[16] B. Warneke, M. Last, B. Liebowitz, and K. Pister. Smart dust: communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, Jan 2001.

[17] P. C. Yunnan, P. A. Chou, Y. Wu, and K. Jain. Practical network coding, 2003.

[18] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, 2002.

2011 18th International Conference on Telecommunications

# Implementing Autonomous Network Coding for Wireless Sensor Network Applications

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics, Graz University of Technology, Austria*
{*Philipp.Glatz, Leander.Hoermann, Steger, RWeiss*}*@TUGraz.at*

## Abstract

*Wireless sensor network (WSN) motes are resource constrained devices. Especially, bandwidth and energy are scarce resources. Therefore, lots of effort is put into the optimization of low-power networking protocols. While network control overhead is an issue for many to most of such protocols, we present an approach that is virtually overhead-free. We introduce the implementation of an autonomous network coding implementation that can be implemented for existing applications and added over existing routing mechanisms with no need to change the application or networking protocols. We discuss and carefully evaluate the approach for applicability and gain according to the most prominent optimization metrics for WSN networking protocols. Especially, we consider the comparison of autonomous network coding to state-of-the-art collection protocols and multipath routing. The completely autonomously acting network coding plug-in implementation is profiled to conserve up to* **29 %** *of the messages that need to be sent locally without the need for centralized network control.*

## 1. Introduction and Motivation

Wireless sensor network (WSN) [1] motes are resource-constrained devices. They suffer limited energy and bandwidth and need to act autonomously over long periods of time. Applying an approach like network coding (NC) - which conserves energy with reduced bandwidth usage with robust methods - in an autonomous and decentralized way, these issues can be tackled. We present such an promising approach and implement it on small WSN motes. We apply different metrics and tools to evaluate them to compare NC to other approaches.

There are situations where it is hard to tell [2] how to best compare two methods coming from different paradigms like network coding and routing. There are obvious constraints when using a regular mesh structure as in [2] and it is not trivial to generalize its comparison to state-of-the-art collection and multi-path routing. These are examples for the optimization of two commonly used metrics - namely robustness and power-awareness. One way of dealing with the problem is to not look at the problem from a global and scalability point of view [2], but to evaluate local characteristics of the behavior of such an approach. It will be modeled, implemented and tested without special assumptions on network control or topology.

We now implement and evaluate such a localized control-free approach for a WSN application. For being able to give more insights to the generalization capabilities of the approach, we leave an existing application and collection protocol unchanged. We then implement an autonomously acting NC approach between the existing networking layer and the application layer. There is no need to change the existing interfaces. The application commands are first linked to the NC layer now. These calls are forwarded to the existing networking layer then. For events from the lower layers this works vice versa. The approach is evaluated for different and random topologies. We discuss results for two networks with the chance to apply network coding and one where no chance exists.

While autonomous network coding stays passive if there are no chances to be applied but just forwarding commands and events, theory tells us that up to 50 % of messages can be conserved in optimal settings [3]. As no optimization comes for free, one has to carefully evaluate networking protocols according to common metrics for networking design. We evaluate the number of messages that needs to be sent against the number of messages that are lost due to link failure, the number of messages that are successfully injected and received, the average delay that occurs and initialization times. That way, the most important metrics and most well-known shortcomings of network coding can be tested. For the evaluation of its power conservation capabilities, we present modeling, simulation and measurement results from TOSPIE2, TOSSIM, Avrora and analysis of power profiling errors of simulation environments compared to accurate measurement results.

## 2. Related Work

A basis for network coding has been established by Ahlswede et al. [5]. Practical examples similar to the single crossing case of a network coding butterfly as shown in

978-1-4577-0024-8/11/$26.00 ©2011 IEEE

978-1-4577-0024-8/11/$26.00 ©2011 IEEE

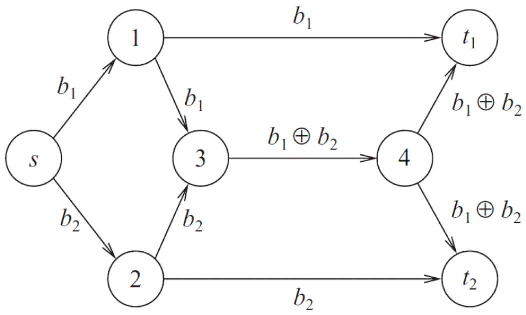



Figure 1. Applying NC to the standard butterfly structure in [4]

Figure 1 are explained by Fragouli et al. [6]. Ho et al. [4] discuss the maximum gain of network coding over routing. For the wireless case where xor-coding is used over galois fields ($2^1$) the gain is bounded by 2. We have shown that this applies to scalable WSNs [2]. For translating bandwidth conservation to energy conservation with an autonomously acting NC approach we extend these considerations. We provide a practical implementation with seamless integration into an existing application and evaluate the approach in detail on a mote platform. The evaluation and validation of the network coding plug-in is done by analysis, simulation and emulation using TOSSIM [7] and Avrora with AEON [8] and analytical modeling and profiling in TOSPIE2 [9]. This includes detailed elaboration of the needed components' power dissipation and analysis of the accuracy of power profile simulation.

For state-of-the-art WSN hardware of small form factor like the Mica and Telos family of motes by Polastre et al. [10] and Hill et al. [11] we will make use of low power-listening (LPL) as described by Moon et al. [12].

## 3. Network Coding System and Evaluation Design

We present the application scenario, autonomous network coding considerations and a concept for evaluation of the approach.

### 3.1. Application Scenario

As an application scenario, we consider a night guard alarm application where the network detects variation of continuously sampled light and temperature sensor values. For being capable of detecting dead motes, the motes continuously send their sensor readings to a base station and the night guard and do not only report if the values change. This also gives the nightguard and/or the base station the chance to decide whether a temperature change in a room is large enough to be classified as an alarming situation. An alarm is sent as long as it is not acknowledged with a single hop connection by the night guard. In other words: if an alarm occurs, the guard goes to the place where it occurred and can redeem it.

The networking system is implemented using a message pool that tries to send data packets on a best effort basis and drops packets after a time-to-live (TTL) threshold has been reached. Packet reception is acknowledged on a per-hop basis and packets are resent after a given timeout if no acknowledgement is received. For being able to detect faulty motes the information of whether acknowledgements can be received are implicitly transmitted to the base station and the guard as well in protocol packets. Protocol packets are sent on a regular basis for checking the availability of routes. Especially, a moving guard is given the chance to initiate route updates. In large networks the reach of such update requests can be regulated with the TTL field to only yield local updates.

### 3.2. Autonomous Network Coding

Earlier studies have elaborated NC to be capable of saving up to 50 % of the messages that need to be sent. The main limitation is that there is no detailed evaluation of a practical implementation for WSNs that solves the problem of autonomously performing NC. The approach presented here relaxes the need for network control by locally setting up adjacency matrices of other motes' connectivity from overheard acknowledgements. The authors want to point out that (i) there is no need to know global connectivity of motes and that (ii) WSNs should be deployed with overlapping regions of their radio range. Otherwise, the network might become too prone to errors. Given these conditions, the approach can notice the applicability of NC for passing flows of information by analyzing the local network structure from overheard communication and acknowledgements.

### 3.3. Evaluation

We provide a detailed evaluation of 20 runs that last for 30 seconds each. Although we do not provide accurate analysis of the tune-in time of the system, its capability to start working efficiently will show its applicability for dynamic environments as well as for static conditions. For static conditions we will compare the number of lost and successfully received messages for both, NC and routing alone. We evaluate trade-offs in terms of additional delay and provide results on the reduced channel capacity that is needed for sending less transmissions with NC.

For being able of accurately evaluating the power consumption of motes and putting savings from network coding into the right context, we perform accurate LPL evaluation and sensor activation power dissipation measurements. This allows deducing an accurate power state model and the application designer may easily give lifetime estimates based on elaborating the network's energy analysis.

## 4. Implementation of Autonomous Network Coding

The heart of the NC implementation is its rule-based engine that needs to decide where NC can be applied such that the network is still robust, of acceptable delay, messages can be decoded at receivers and that the actual number of transmissions can be reduced. The NC implementation is situated between application and networking interfaces. While the application has the right to decide where a message needs to be sent to and each intermediate multi-hop message is presented to upper layers by the networking interface as well, the NC layer may alter the messages. If data packets can be combined, the message frames are altered such that non-NC motes regard it as a not-to-identify message such that it is dropped. On the other side, NC motes know the protocol and know about when to code and decode data packets. This not only allows implementation separation from other layers and modules with no need to change them, this also allows not impacting other motes that maybe do not know about this NC protocol.

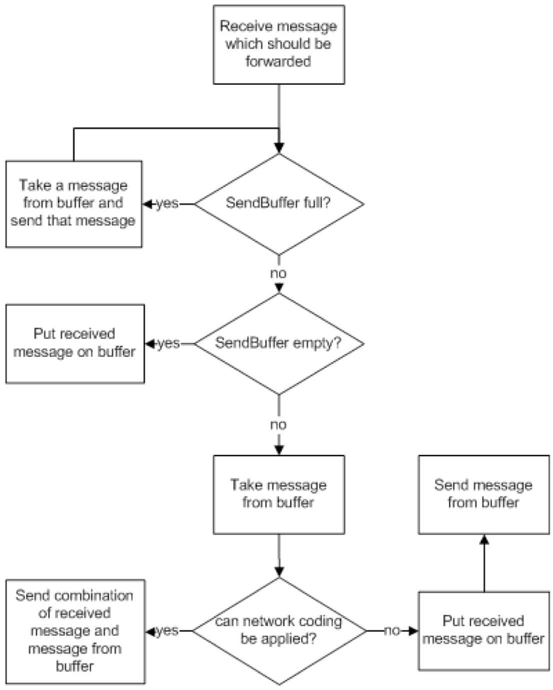



Figure 2. The forwarding procedure flowchart.


Figure 3. The NC decision procedure flowchart.

### 4.1. Message Buffering for Network Coding

Figure 2 shows how messages are buffered such that there can be messages available for being given chances of combing data. Whenever a message arrives, the buffer is checked for coding opportunities. In case that within a certain amount of time, there are no messages being sent, or in case the buffer gets full, messages are removed and forwarded according to routing information. This keeps the system from starving, yet it introduces some delay.

### 4.2. The Network Coding Decision Process

If the mote has identified itself being part of a NC structure and receiving messages that corresponds to it, it encodes and decodes messages according to the flow chart shown in Figure 3. NC can be enabled if flows bypass the node and are intersecting at it. If a node should redirect two messages to two different destinations and it can be ensured that each other destination already received the message intended for the other destination, NC can be applied. Furthermore, it has to be ensured that the sources of the messages are different. Possible losses can be considered with an acknowledgement mechanism by overhearing further forwarded messages.

## 5. Results

We present the results of autonomous NC performance based on the most common metrics and add a discussion on

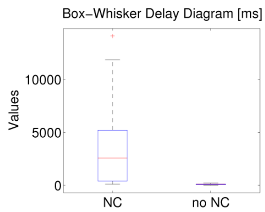
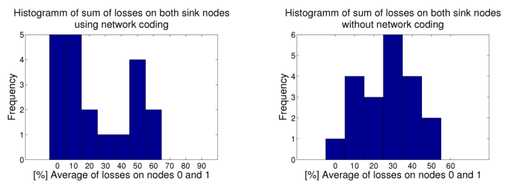



Figure 4. The delay increases when using NC.




(a) Message loss with NC (b) Message loss with routing

Figure 5. We could find no significant increase of message loss when using network coding.

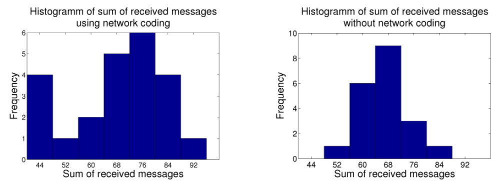
its possible impact on energy conservation.

## 5.1. Timeliness of Sensed Data

When NC is used, the motes may not instantly forward their data, but they have to buffer their messages in case they can be combined. Figure 4 shows the median, lower quartile, upper quartile, lowest values and highest values of delay. The average delay when using NC evaluates to be $3720\,ms$ while without NC it is only $92.5\,ms$. This value could be tuned with checking the message pool more frequently on whether there are messages that can be combined which is currently done each $1500\,ms$. Furthermore, the application samples the sensor with different sampling rates averaged over 5 seconds. These values have been chosen for achieving a robust application behavior. From that point of view, real-time (in terms of message delay) capabilities are mainly determined by the application itself. Furthermore, a message delay of a few seconds is usually acceptable for WSN data collection applications.

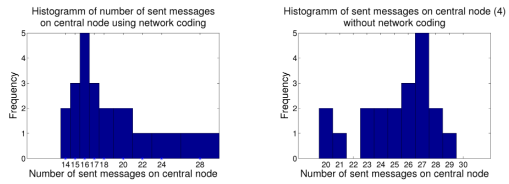
## 5.2. Impact on Robustness

NC does not alter system robustness as can be seen in Figure 5. Approximately the same loss of messages occurs. TOSSIM has been used for testing the setup with a simulation environment capable of path loss modeling. We want to add, that there is no direct quantitative mapping from path loss in simulation to real world path loss. Therefore, the actual message loss values may vary, but the relation of loss due to NC to the loss due to routing alone may




(a) Transmissions with NC (b) Transmissions with routing

Figure 6. There are approximately the same number of messages being transmitted when using NC as when using routing.




(a) Transmissions with NC (b) Transmissions with routing

Figure 7. $29.3\,\%$ of messages and the according power dissipation can be conserved when using autonomous network coding.

approximately stay the same.
With NC $25\,\%$ of the messages have been lost. Without NC, there was a message loss of $27\,\%$. The slight variation is expected to be caused from temporal path loss characteristics where the little more delayed NC approach may benefit. In general one can see that there is no significant difference of the average path loss. It has to be added that acknowledgment messages have been used for both NC and routing.

## 5.3. Comparable Settings

For being able to judge the impact of the actual number of messages being lost or conserved, one needs to compare the number of messages actually being sent in the network. From an application point of view one is interested in how many transmission have been received successfully. Therefore, we plot the number of messages that have been received during the different runs in Figure 6. On average (20 simulation runs) there are $75.5$ messages transmitted when using NC and $76$ messages are transmitted without.

## 5.4. The Network Coding Gain

Finally, Figure 7 shows histograms of the saving of messages when using autonomous localized NC. Results are plotted for the central mote of a butterfly structure. The approach conserves $29.3\,\%$ of the messages that need to be sent by the central mote and it reduces the overall number of messages that need to be sent in the local

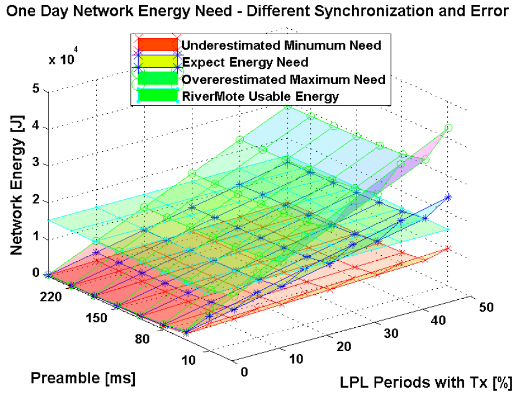



Figure 9. The subplots analyze the energy-budget and -needs of a 9-node network where all motes are in communication range of each other. We show results for different LPL settings, transmission rates, MAC synchronization and modeling error compared to the available energy budget.

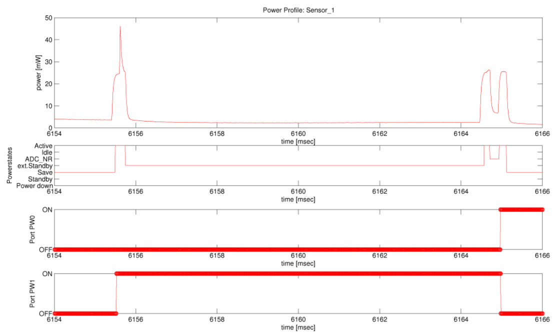



Figure 10. TinyOS Read - ReadDone for the photo sensor on the MTS300 sensorboard with a mica2 attached to the same ADC as the temperature sensor.

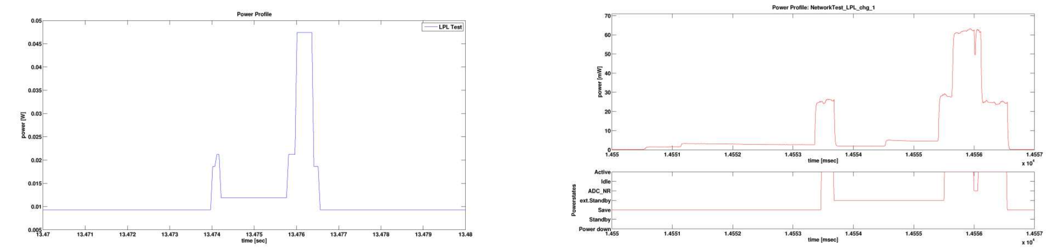
butterfly structure by $14.8\,\%$. Different types of bandwidth and energy conservation are achieved. First, this reduces the average power dissipation at motes that actually send data. Second, motes that listen for messages suffer less impact on their power dissipation as well. This includes neighboring nodes as well. Finally, the channel is far less utilized. This makes the overall system more robust or allows to increase sampling data and sending more messages at a higher rate with less delay if needed.

### 5.5. Network Coding Energy Conservation

After a detailed analysis of functional requirements and properties we now present an evaluation of energy conservation properties. Using Avrora is a good choice for accurately validating functionality and for giving a coarse grain estimate for energy conservation, but it fails to accurately model short time effects of LPL and resultant interaction of the radio, the bus system and the CPU. Figure 8(a) shows the results of an energy monitor that we have implemented on top of Avrora. Although it provides some valuable insights, it cannot replace the results from hardware measurements with automated power state annotation that we have implemented for the TOSPIE2 environment as can be seen in Figure 8(b). First, we have extended the LPL implementation such, that after LPL radio shutdown, we also turn off the SPI bus which lets the CPU go to power save mode as well. Avrora is not capable of modeling this fundamental difference. Second, although most important events are modeled by Avrora, not even when sampling the power profile at $20kS$ it is capable to provide detail insights for power dissipation of events as small as interrupts. Unfortunately, they have great impact on the overall average power consumption when using really low-power methods.
Leaving the detailed evaluation with TOSPIE2 out for brevity we can sum up the comparison of measurement and simulation/emulation results as follows: for ongoing LPL, when the machine model error (several times higher than the overall LPL energy needs) has already been removed and transient effects from switching on motes and initial LPL RSSI measurements are gone, there is still an error of more than $22\,\%$ on deterministic LPL parts' average power dissipation. These parts make up more up more than $70\,\%$ of the overall energy needs. Figure 9 shows trade-offs for LPL periods and data sending rates given a sustainable supply system as described in [13]. It shows the expected energy needs of a nine-node network with one sender. Worst case and best case LPL energy needs plus the simulation error have been included for visualization of the impact of the possible error of simulation environments described before. The acceptable preamble-length$\times$data-rate region below the available-energy surface vary significantly. Finally, Figure 10 depicts the power dissipation of a sensor reading for estimating its impact on the overall energy consumption for being able to put NC energy conservation into the right context. Although, taking the reading takes quite some time, the energy spent on the complete process is in the same magnitude of order as a single LPL event. However, LPL peaks occur very frequently (i.e. $255\,ms$ LPL periods result in $0.468\,mW$ measured as average power dissipation) and sending or receiving a message consumes energy 3 to 4 magnitudes of order higher than taking a sample.

## 6. Conclusion and Future Work

We have introduced, implemented and carefully evaluated an autonomously acting NC layer. It has been applied to an




(a) Avrora second peak

(b) NI second peak

Figure 8. Although simulation-based approaches (Avrora second LPL peak 8(a) and radio activation) can give good estimates, different problems with accuracy can be identified as far as hardware models and timing accuracy are concerned. Hardware measurements can give far more accurate results when it comes to short time radio activation as can be seen from the second LPL peak and radio activation 8(b).

existing application with no need to change the existing protocol or any other of the implemented components. It works completely control-free and has been profiled to save up to 29.3 % of messages on average. This approach for robust energy conservation through efficient bandwidth utilization can be applied in static and in dynamic environments. It considerably relaxes the overall energy constraints in the network, where all major power dissipating components have accurately been evaluated for sakes of completeness. Future directions may include to further tune the fade-in time of the process of establishing network information flows to further reduce response times and improve the gain from messaging conservation.

## References


[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] P. M. Glatz, K. B. Hein, and R. Weiss, "Scalable network coding for wireless sensor network energy conservation," *International Journal of Autonomous and Adaptive Communications*, accepted for publication.

[3] ——, "Energy conservation with network coding for wireless sensor networks with multiple crossed information flows," in *Proceedings of the 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, 2009, pp. 202–207.

[4] D. S. Lun and T. H. Ho, *Network Coding - An Introduction*. Cambridge University Press, 2008.

[5] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204 – 1216, jul. 2000.

[6] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63 – 68, 2006.

[7] E. Perla, A. O. Catháin, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "Powertossim z: realistic energy modelling for wireless sensor network environments," in *PM2HW2N*, 2008, pp. 35–42.

[8] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," in *EmNets*, 2005, pp. 37–44.

[9] P. M. Glatz, C. Steger, and R. Weiss, "Tospie2: tiny operating system plug-in for energy estimation," in *IPSN*, 2010, pp. 410–411.

[10] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN*, 2005.

[11] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," in *Technical report, Computer Science Department, University of California at Berkeley*, 2001.

[12] S. Moon, T. Kim, and H. Cha, "Enabling low power listening on ieee 802.15.4-based sensor nodes," in *IEEE Wireless Communications and Networking Conference*, 2007, pp. 2305 – 2310.

[13] L. B. Hörmann, "Master's thesis. design and implementation of a wirless sensor platform for river monitoring based on energy harvesting," 2010.

2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks

# Energy Conservation with Network Coding for Wireless Sensor Networks with Multiple Crossed Information Flows

Philipp M. Glatz, Karima B. Hein, Reinhold Weiss
*Institute for Technical Informatics*
*Universtiy of Technology*
*Graz, Austria*
$\{Philipp.Glatz,\ Hein,\ RWeiss\}$*@TUGraz.at*

## Abstract

*The theory of network coding is hardly ever used and cannot be mapped to general wireless sensor network (WSN) topologies without careful consideration of technology constraints. Severe energy constraints and low bandwidth are faced by platforms of low computational power.*
*We show how network coding methods can be implemented with low computational power. We discuss extensive experimentation in simulation of scalability for arbitrarily large networks. It is configured according to the results from mote hardware measurements.*
*The testbed implementation and the mote software implementation is kept completely generic. All phases, including the initialization, are implemented and work for mesh networks of any size without modification.*
*Our work explains important considerations for applying network coding to WSNs. The gain from applying the method over ad-hoc on-demand (AODV) like routing closely approaches the optimum value of 2. Thereby WSN resource constraints are relaxed and network lifetime is prolonged.*

## 1. Introduction and Related Work

Network coding, where crossing flows of information can be combined and be decoded later at the receiver provide large gain in network throughput. The idea of network coding with its algebraic description stems from the field of graph theory [1]. Using network coding in mobile ad hoc networks [2] allows to minimize the energy spent per bit. Furthermore the problem of finding optimal network codes can be related to the Steiner-Tree packing problem [3]. This results in lower complexity control overhead for network coding (polynomial time) than for finding the minimum-energy routing multicast tree (NP-hard).

### 1.1. Wireless Network Coding

While the gain for coding may grow further with growing wireline networks, the throughput gain for the wireless case is bounded by 2 [4]. Most work in the field of wireless network coding does not explicitly consider the energy gain in comparison to routing. Some even cut out power requirements and rely on technology with a form factor of laptops [5] with relaxed constraints compared to wireless sensor networks (WSNs) [6].

### 1.2. Wireless Sensor Networks

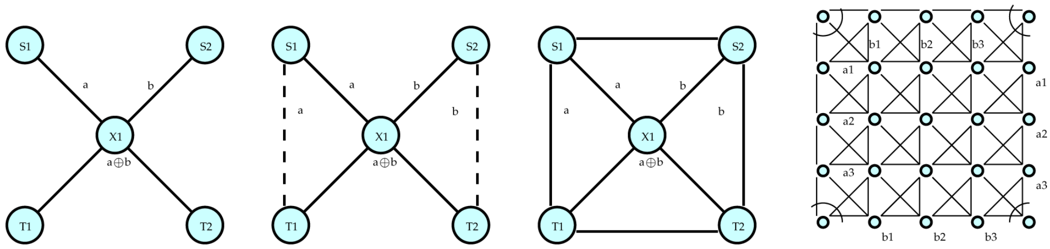
Few studies [7] deal with network coding for low data rate WSNs and consider the technology's powering [8] and bandwidth [9] constraints. The main issue is to map network coding to the technology's constraints where a first step is taken in [10]. Such approaches mostly gain through applying opportunistic behavior [11].

### 1.3. Contribution Claim

This paper provides the following contribution.

- It shows the generalization of butterfly structures in wireless scenarios to crossing perpendicular groups of virtual unicast information flows. All the information can be decoded at the receivers in the setup.
- The method can be implemented on WSN platforms of small form factor. The gain in bandwidth approaches the optimum at 2 already for relatively small sized networks. The implementation is described in detail and broken down to elementary algebra.
- The network coding approach can start after a routing initialization phase and can switch between network coding and routing. Motes in the network need no further information on the network except for confirmation that the network is of mesh type with diagonal communication links.
- The approach works fully generically for arbitrarily large networks. Both, the mote implementation and the testbed implementation, are fully scalable. The simulation time increases in $\Theta(n)$ with $n$ being the network size. The maximum delay increases with $\Theta(\sqrt{n})$ in the case of perpendicular information flows.
- Important considerations when implementing network coding on WSN motes are discussed. Implemented

978-0-7695-3908-9/09 $26.00 © 2009 IEEE
DOI 10.1109/I-SPAN.2009.90




(a) Butterfly (b) Mesh (c) Star (d) Network

Figure 1. The butterfly structure: standalone, in a mesh, star and net topology.

upon a state-of-the-art media-access-control (MAC) protocol - BMAC - in a standard mote operating system - TinyOS - we explain why the ratio of sending cost to receiving cost in low-power-listening (LPL) is an important factor to consider.
- For allowing for a detailed discussion of the sending to receiving cost ratio we have implemented and run power dissipation measurements on Mica2 motes. We compare the ratio's impact to the impact of different message lengths and sending power settings which have been profiled too in other work.

We will first describe the simulation setup and mention the measurement system. Then the mote implementation will be described in detail and settings for the simulation will be deduced from measurement results. Finally, the characteristic savings in residual energy will be discussed.

## 2. Concept

The well known butterfly structure is discussed. We will then show how to generalize the butterfly structure to multiple crossing information flows. We will explain what information is needed and the time when and how long it is needed. We will generalize in a way so that the method can be implemented for an arbitrary number of nodes in a quadratic star network.

For considering the energy conservation capabilities one may want to find a single optimum setting given a specific method like network coding. This cannot be achieved without careful consideration of the target technology. The following issues influence the achievable optimization:
- MAC protocol
- initialization times of hardware
- LPL preamble and data length
- sending power

These factors have influence due to the fact that they influence the sending to receiving cost ratio. For the given technology the preamble length can be easily set from software and the impact on the ratio is high - while the sending power only slightly impacts the average power consumption over time and the time for actually transmitting data is very small compared to the preamble.

In the remainder of this section we will explain how the system is set up, run and evaluated.

### 2.1. The Minimum Butterfly Structure

The minimum butterfly structure (see Figure 1(a)) can be viewed as a $1 \times 1$ part of a mesh network. Overhearing the information flow $\mathcal{F}_A$ at a sink node of flow $\mathcal{F}_B$ needs diagonal communication links - otherwise network coding ($\mathcal{F}_A \oplus \mathcal{F}_B$) is possible but the information cannot be decoded ($\mathcal{F}_{AB} \oplus ?$) at the sink. The diagonal links are missing in Figure 1(b), but they are present in Figure 1(c) which we view as being part of a star topology.

**2.1.1. The butterfly structure.** Figure 1(a) shows the standard network coding example with two multicasts $\mathcal{F}_A$ : $(S1) \rightarrow (T1, T2)$ and $\mathcal{F}_B$ : $(S2) \rightarrow (T1, T2)$ crossing each other in node $X1$. Each node is assigned a bandwidth for sending 1 message per time unit. Routing the messages will congest the channel at $X1$. It takes two time units to transmit both messages. This reduces the maximum rate $r_{S1}$ and $r_{S2}$ for the source processes at $S1$ and $S2$. For each edge $e \in \mathcal{E}$ in graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ let $o(e) \in \mathcal{N}$ be the edge source and $i(e) \in \mathcal{N}$ be the sink. Due to congestion the maximum source rate has to be reduced such that the set of incoming $\mathcal{I}(\mathsf{v})$ edges of a node (see equation 1) does not exceed its outgoing bandwidth for edges in $\mathcal{O}(\mathsf{v})$.

$$\mathcal{I}(\mathsf{v}) = \{e_{\upsilon\mathsf{v}} \in \mathcal{E} \forall \upsilon : \mathcal{O}(\upsilon) \cap \mathcal{I}(\mathsf{v}) \neq \{\}\} \quad (1)$$

For wireless networks $\mathcal{O}(\mathsf{v})$ can be viewed as a single edge for outgoing transmissions. So we have to satisfy the condition in equation 2.

$$\sum_{s \in \mathcal{N}: \mathcal{O}(s) \cap \mathcal{I}(t) \neq \{\}} r_{st} \leq SendingBandwidth \forall t \in \mathcal{N} \quad (2)$$

In other words, the limitation of crossing routes feeds back to the information source.

This can be avoided with network coding. With network

Table 1. Setup for an arbitrary number of crossings

| Topology | Quadratic star topology |
|---|---|
| Corner node | Node in the upper left of $S1$ and $S(k+1)$ |
| Receiver node | Node only connected to one node $Ti \in \mathcal{T}$ |
| $S1 \ldots Sk$ | Left hand side source of information flow |
| $S(k+1) \ldots S2k$ | Upper source of information flow |
| $T1 \ldots Tk$ | Lower sink of information flow |
| $T(k+1) \ldots T2k$ | Right hand side sink of information flow |
| $Xi$ | Repeater node inside the network |
| $n$ | Number of nodes |
| $k$ | Sources/sinks per side ($k = \sqrt{N} - 2$) |
| $m_{ij}(t)$ | message along edge $e_{ij}$ |

coding we view each message (and source process accordingly) as a vector of $|m|$ bits. The vector is represented in a Galois-Field $GF(2)$. Each node adds up all the incoming messages for which it is a designated receiver and forwards the result. For the butterfly case we have at a given point $t$ in time $m(\mathcal{O}(X1), t + \epsilon) = \mathcal{F}_{AB}(t) = \mathcal{F}_A(t) \oplus \mathcal{F}_B(t)$. The length of the message $|m|$ does not grow. Therefore we have the same bandwidth for sending available at all nodes. The feedback of bandwidth constraints is solved, but there are two other aspects to consider:

1) Which information is needed for being able to decode the packets at the receiver side $\mathcal{T} = \{T1, T2\}$?
2) Communication links and computations are considered being lossless and error-free, but for practical settings the network cannot be delay-free. Which buffering policy is needed to have the right information at hand at the right time under a given MAC setting?

The first one - decoding the packets - shows up in Figure 1(b). Overhearing $\mathcal{F}_A$ at $T1$ allows to decode $b = a \oplus a \oplus b$. In Figure 1(c) the star structure explicitly points out the transmission along $e_{S1,T1}$.
The latter one - timing issues - demands for MAC considerations. Consider evaluation of communication links in order $e_{S1,T1}(t)\ \ldots\ e_{X1,T1}(t+1)\ \ldots\ e_{S1,T1}(t+2)$. Dependent on whether $t$ and $t+1$ or $t+1$ and $t+2$ are assigned to the same session, different information will be decoded at target node $T1$.

## 2.2. Multiple Butterfly Structures

As a first step we discuss how information flows $\mathcal{F}_{A1}$ and $\mathcal{F}_{A2}$ cross $\mathcal{F}_{B1}$ perpendicular. This example with two crossings helps in understanding what it takes to raise it to an arbitrary number of crossings. For the packets which might contain the information of more than two flows it has to be reduced. This has to happen at nodes prior to entering areas in the network where the flow from network coding would contradict the direction of information flow that routing would have taken. This is important for keeping the network coding part embeddable in a routed network environment. In our setup, information flows from top to bottom and from left to right. So the direction of routing is pointing to the lower right in our case. In the following we will look at symmetric star networks which with $n$ nodes and $2 \cdot (\sqrt{n} - 2)$ sender and receivers accordingly.

Table 2. Number of messages for an arbitrary number of crossings

| Method | Number of Messages |
|---|---|
| Routing per flow | $k+2$ |
| Routing per network | $2 \cdot k \cdot (k+2)$ |
| Coding per network | $4 \cdot k + k \cdot k$ |

## 2.3. Arbitrary Number of Crossings in Star Networks

We now generalize to an arbitrary number of crossings in star networks. The setup for incorporating an arbitrary (but quadratic) number of crossings is explained in Table 1. We show the quadratic case only due to lack of space, but the method can be applied to any star-structured network with parallel information flows. Studying the impact of arbitrary multicasts will be part of other work.
For initialization of the system we will flood the network starting from the receiver side. In a real network, this allows to build the same state machine into each mote and initiate the network from receiver motes. The only precondition prior to deployment will be to deploy it in a way so that communication links form a star topology. We let the initialization phase last $N$ time units. This way, each node builds up a routing table to the $2 \cdot k$ receiver motes. Afterwards the motes start to sense and/or forward data.

**2.3.1. Rules for Combining Messages.** We add $m$-length messages in $GF(2)$. For generalizing to an arbitrary number of crossings we have to reduce the number of visible information flows per message to a maximum of two at every node. We achieve this - as already mentioned - by first combining and forwarding messages from left to right and top to bottom or vice versa until we end up in the lower right corner of the network. We code the receiver identification into each message. This way at the receiver side we will only combine messages from the left side, the upper left or above and keep other parts from being further forwarded. This perfectly reduces the content at each node to a maximum of two information flow's messages.

**2.3.2. The Gain over Routing.** Network coding reduces the number of messages that have to be sent compared to the case when routing is used. Table 2 gives a summary. We see from Table 2 that the gain for large networks converges to 2.

### 2.4. The Sending Cost to Receiving Cost Ratio

We will model with integer parameters, but from theory we should get a value somewhere around 2 anyway. Due to technology constraints like initialization times the ratio for the average sending energy over average reception energy needs can be assumed to be slightly above $sc = E(\frac{P_{snd}}{P_{rec}}) \approx 2$ (we call $sc$ the sending cost for the receiving cost set to 1), but the ratio is reduced for longer messages and lower sending power.

### 2.5. Evaluation

For evaluation we will put the intelligence for tracking the messages that are injected, time that passes by, the motes' energy balance and whether the channel is available into a testbed. The simulation's mote implementation is responsible for annotation of received packets. The testbed double-checks if packets are missing. The setup settings are summarized in Table 1.
The Mica2 LPL power dissipation is profiled accurately with a measurement setup based on a National Instruments 6221 PXI DAQ card.

## 3. Implementation

The implementation section describes the setup phase in an AODV manner without replies, the network coding phase and the routing phase for comparison all with pseudo code. Algorithm 1 shows the initialization phase that allows to run both - routing and network coding - afterwards. $rt\_d\_via(mote, via, target)$ contains the distance

**Algorithm 1** Network Initialization

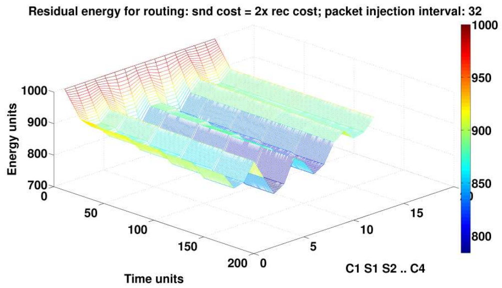
```
Require: FLOODING(ReceiverMotes)
Require: adj_matrix
Require: rt_d_via ← inf
Ensure: RoutingTableBuilt ≡ TRUE ∀ t ≥ N
function FLOOD                                   ▷ Matlab Notation
    for m ≡ 1...N do
        d ← ZEROS(N, N)
        for dd ≡ 1...N do
            for t ≡ 1...N do
                d(dd, t) ← MIN(rt_d_via(dd, :, t))
            end for
        end for
        for adj ≡ FIND(adj_matrix(m, :) ≥ 0, N) do
            for t ≡ 1...N do
                rt_d_via(adj, m, t) ← d(m, t) + 1
            end for
        end for
    end for
end function
```

**Algorithm 2** Network Routing

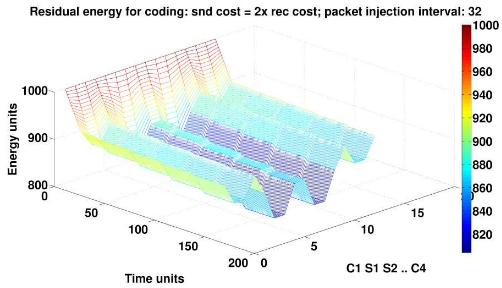
```
Require: FLOOD ≡ finished                        ▷ t ≥ n
Require: msgs(t) ← sensing(t)
Ensure: SentMessagesReceived ≡ TRUE ∀ messages
function ROUTING                                 ▷ Matlab Notation
    for mote ≡ mac_order do
        for rcvr ≡ 1...n do                      ▷ if(bwidth(mote))
            dist, ngb ← MIN(rt_d_via(mote, :, rcvr))
            msgs(ngb, rcvr) ← msgs(mote, rcvr)
            msgs(mote, rcvr) ← inf
            REMOVEENERGY(mote)
        end for
    end for
end function
```

**Algorithm 3** Network Coding

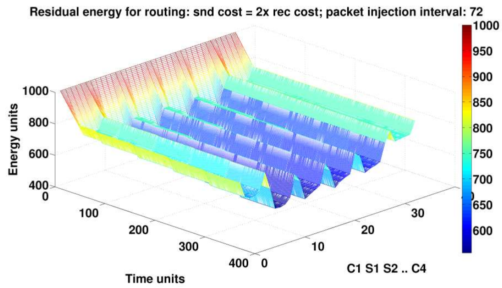
```
Require: FLOOD ≡ finished                        ▷ t ≥ n
Require: msgs(t) ← sensing(t)
Ensure: SentMessagesReceived ≡ TRUE ∀ messages
function CODING                                  ▷ Matlab Notation
    for mote ≡ mac_order do                      ▷ isStdTx(mote)
        for rcvr ≡ 1...n+1 do                    ▷ if(bwidth(mote))
            p, rec ← FIND(msgs(mote, :) < inf, n+1)
            p ← msgs(mote, rec)
            if LENGTH(p) ≡ 3 then
                packet ← GF2ADD(p(1), p(2))
                packet ← GF2ADD(packet, p(3))
                msgs(mote+1, rec(1)) ← packet
                msgs(mote+√n, rec(2)) ← packet
                msgs(mote+√n+1, rec(3)) ← packet
                msgs(mote, :) ← inf
                REMOVEENERGY(mote)
            end if
        end for
    end for
end function

function GF2ADD(i1,i2)                           ▷ Matlab Notation
    mpow ← CEIL(log2(i1, +i2))                   ▷ Truncate
    for pow ≡ mpow ↓ 0 do
            ▷ The power of two is a shift: little comp. load
        b ← 2^pow
            ▷ Dividing by a power of two: little comp. load
        v1 ← FLOOR(i1/b)
        v2 ← FLOOR(i2/b)
        i1 ← i1 − v1
        i2 ← i2 − v2
        o ← XOR(v1, v2), o
    end for
    for i ≡ 1 ↑ LENGTH(o) do
        GF2ADD ← GF2ADD + o(i) · 2^(i−1)
    end for
end function
```



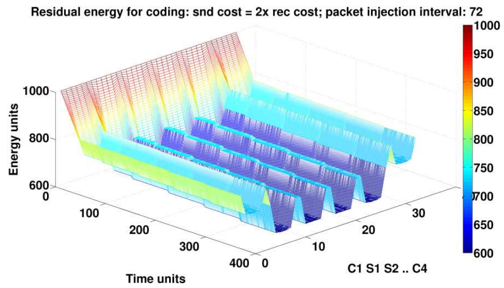

(a) 16 nodes routing




(b) 16 nodes coding




(c) 36 nodes routing




(d) 36 nodes coding

Figure 2. Subplots show the energy consumption of routing and network coding for differently sized networks. Initial flooding and $4$ rounds of packet injection are shown. Inner nodes have steeper residual energy drops, but increasing the network size does not further impact energy loss. The lower residual energy depression will not always be flat for routing due to multiple minimum-hop paths. There exists only one path for all transmissions if network coding is used. Therefore, the load among inner coding transmitters is inherently balanced by the approach that is presented.

from $mote$ to $target$ along $via$. The initialization starts with flooding from the receivers and ends up for any topology with a routing table at each node that leads to the receivers. The routing implementation chooses the minimum hop count and transfers messages to another place in array $msgs$ indicating it pending at the new mote and clearing it at the old one with $inf$ in Algorithm 2.

The network coding implementation has to implement crossing flows as depicted in the precious section. We show the case of a standard transmitter - that can be deduced from the messages received in the initialization phase - that activates if it has 3 messages pending. One can see how the receiver of physical broadcasts have to be set to not contradict the routing direction on the one side and always reducing the the number of flows from 3 to 2 for any mote on the other side. The second function in Algorithm 3 shows the simulation implementation and it becomes obvious that all computations can easily be broken down to be performed with shifting and adding and only little overhead in size of program code. The case of the standard transmitter can be extended to all the other nodes such that the implementation stays the same, but some lines are to be removed where following the routing direction fewer packets are to be

Table 3. Measuring the sending to receiving cost ratio with a NI PXI 6221 DAQ.

| Type | $\frac{P\_avg(snd)}{P\_avg(rec)}$ |
|---|---|
| 25 % Duty Cycle | 2.1191 |
| 10 % Duty Cycle | 2.2709 |
| 5 % Duty Cycle | 2.3028 |

awaited. As an example consider the upper left transmitter: it will get packets from the upper left and the leftmost top source nodes, but the upper left corner node will not send any messages in our setup.

## 4. Results

This section presents the results from measurements and simulation.

First of all we have evaluated the ratio of sending cost to receiving cost to set the testbed implementation to the nearest integer value accordingly. The values for sending short messages with maximum sending power results in a ratio close to 2 as can be seen in Table 3. For technologies

Table 4. Transmitter savings in Figure 3 for data delivery from network coding ($NC$) over routing ($RT$).

| | |
|---|---|
| Energy Loss ($RT16$) | $840 - 784$ |
| Energy Loss ($RT36$) | $640 - 556$ |
| Energy Loss ($NC16$) | $840 - 804$ |
| Energy Loss ($NC36$) | $640 - 600$ |
| Variance ($RT16$) | 20 |
| Variance ($RT36$) | 56 |
| Variance ($NC16$) | 0 |
| Variance ($NC36$) | 4 |

with a higher ratio, network coding has even larger benefits than shown by the results presented here.

In Figure 3 the residual energy is plotted over time for each mote. Motes are numbered top-down and from left to right: upper left corner, left transmitter nodes, the leftmost upper transmitter, leftmost repeater, leftmost lower receiver, ... until the lower right corner node. The highest residual energy can be observed at the corner nodes as they are not part of the crossing information flows. For network coding all sending and receiving nodes end up approximately at the same energy level (i.e. inner transmitter get penalties from overhearing others that outer ones do not get) for network coding and routing. The gain in energy conservation can best be seen from inner transmitter nodes. Furthermore the bottom of the residual energy depression is flat for network coding while it varies for routing. For prolonging a WSN's lifetime it is better to evenly distribute load among the group of nodes that will break down first. So it can clearly be seen that the proposed method is suitable for mapping it to WSNs. Table 4 gives a summary.

The number of messages that have to be sent for transmission of the packets approaches 2 very early as can be seen in Figure 3. In all experiments all messages could be decoded at any time even though the MAC order has been set to be the reveres of the mote numbering which represents the worst case for a statically assigned MAC as it was used in the testbed for deterministic results.

## 5. Conclusion

We have shown that a gain of up to 2 can be achieved with network coding and that the gain can be mapped to prolonged network lifetime under technology constraints of WSNs. We have set up a simplified network model in simulation to better understand what can be achieved and give a completely localized and scalable implementation that can be easily implemented on small motes.

Future work will be used to support power awareness of a middleware and will help to find energy conservation with adaptation to the efficiency of energy harvesting systems. Furthermore other parts of our work deal with automated recognition of structures in a network that are subject to network coding and switching forth and back between network coding and routing.

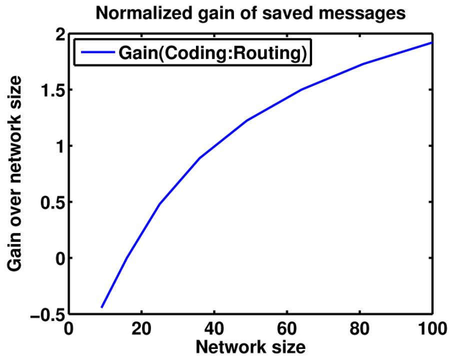



Figure 3. The normalized gain approaches $2$ for $k = 8$.

## References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, 2000.

[2] Y. Wu, P. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, 2005.

[3] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing steiner trees," in *ACM-SIAM SODA*, 2003.

[4] T. Ho and D. Lun, *Network Coding: An Introduction*. Cambridge University Press, 2008.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, 2008.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, 2002.

[7] R. Jacobsen, K. Jakobsen, P. Ingtoft, T. Madsen, and F. Fitzek, "Practical Evaluation of Partial Network Coding in Wireless Sensor Networks," in *MobiMedia*, 2008.

[8] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in WSNs," in *IFIP Wireless Days*, 2008.

[9] S. Moon, T. Kim, and H. Cha, "Enabling low power listening on ieee 802.15.4-based sensor nodes," in *WCNC*, 2007.

[10] P. Chou, Y. Wu, and K. Jain, "Practical network coding," 2003.

[11] W. Chen, K. Letaief, and Z. Cao, "Opportunistic network coding for wireless networks," in *ICC*, 2007.

2011 Ninth Annual Communication Networks and Services Research Conference

# Opportunistic Network Coding for Energy Conservation in Wireless Sensor Networks

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics, Graz University of Technology, Austria*
$\{Philipp.Glatz,\ Leander.Hoermann,\ Steger,\ RWeiss\}$*@TUGraz.at*

## Abstract

*Wireless sensor networks (WSN) continuously enhance processing capabilities and miniaturization. However, there exists a design gap to energy and bandwidth availability. Especially battery technology cannot keep pace with demands of novel versatile services. A common approach for conserving channel capacity and energy is optimizing power-aware routing and different kind of duty cycling (DC) and harvesting technology. While these optimizations are usually dealt with separately, we provide a novel framework with integrating and analyzing these different aspects in a practical TinyOS implementation at the same time. We implement the essential combination of energy harvesting aware routing (EHAR) together with radio and application DC and we add the novel approach of opportunistic network coding (ONC) for WSNs. We give detailed analysis of the applicability of application-level DC compared to low-power MAC and power save modes for state-of-the-art WSN and harvesting system hardware. We elaborate static and dynamic aspects of EHAR, scalable network coding (SNC) and ONC. Combining analytical models, comprehensive simulation and detailed highly accurate hardware power profiling measurement results, we demonstrate energy conservation from* $13\,\%$ *to* $50\,\%$ *when applying ONC and SNC.*

## 1. Introduction and Motivation

Wireless sensor networks (WSNs) [1] have found their way into many different application areas [2]. Especially for low data rate monitoring applications, they provide low-cost and maintenance-free autonomous solutions.

A range of examples exist where enabling technologies (especially solar energy harvesting device (EHD) and system (EHS) technology) and optimization methods (especially fault tolerance and energy conservation networking schemes) are used to improve WSN performance. However, hardly ever, different optimizations and technologies are accurately profiled in real-world settings considering all side-effects and interaction of different optimizations and layers.

Therefore, we provide a middleware approach, that is implemented for and thoroughly tested and measured with real hardware that takes many of the most promising and important energy conservation strategies into account. Firstly, we use state-of-the-art mote and EHS hardware with TinyOS. Secondly, we accurately profile mote power dissipation from hardware measurements and the error of power profiling simulation environments with the TOSPIE2 environment [3]. Thirdly, we implement energy harvesting aware routing (EHAR) and mote duty cycle (DC) approaches upon low power listening (LPL) with a robust slotted design. Next, we introduce network coding (NC) optimizations for two different communication paradigms: data gathering and data communication. These are optimized with a scalable NC (SNC) implementation that is well suited for static or topology-controlled WSNs and opportunistic NC (ONC) that can work in a completely control-free ad-hoc manner as well. Finally, thoroughly evaluating the combination and side-effects of these different techniques for complete runs always including all initialization overhead, we show that $13\,\%$ to $50\,\%$ radio power dissipation can be conserved. The practical implementation of little overhead and versatile communication and DC capabilities provide an easy-to-use networking and control interface. We remind, that all considerations are dealing with layers below the application layer. Approaches like data-aggregation are out of scope. Yet, they can be implemented interference-free upon the novel ONC approach due to its robust and single-hop-decodability nature.

## 2. Low-Power Wireless Sensor Networks

Networking energy conservation strategies can be divided into activity-based active and connectivity-based passive energy conservation protocols [4].

### 2.1. Energy Harvesting Aware Routing

Active energy conservation routing protocols and lifetime-maximization protocols try to minimize networking impact on power dissipation and energy budgets such that the overall application's requirements can better be fulfilled. This can be done according to different metrics.

The minimum energy spent per message at the hops along its route is a common metric and may include overhearing costs

978-0-7695-4393-2/11 $26.00 © 2011 IEEE
DOI 10.1109/CNSR.2011.42

as in our evaluation. In case the path is fixed, the transmission power could be adjusted such that the minimum power is used per hop for achieving minimum energy consumption per route as done by PARO [5]. However, such an approach might not be easily applicable to WSNs. For low-data rate communication in remote settings it is complicated to tune the transmission power down to an energy conserving level while preserving robust communication links. The protocol overhead for balancing such an approach over a scalable WSNs might soon exceed the possible energy conservation (especially, because of the often very low data rates for monitoring applications).

A more robust approach might be to use energy-aware routing with networking/radio load-balancing for optimizing longevity of a WSN. The approach presented in [6] takes into account the energy of a single mote and serves as a basis for EHAR in [7]. They consider the residual battery level $B$ and the harvesting rate $\rho$ at mote $i$:

$$E_i = \omega * \rho_i + (1 - \omega) * B_i \quad (1)$$

The weight parameter $\omega$ is used to set the ratio of the factor of influence of the EHD profile. The value range of $\omega$ is $0 \leq \omega \leq 1$. Using the energy level $E_i$ as cost of transmitting along an edge $e_{ki}$ to mote $i$ for all motes in range $E_{comm}$

$$c_{ki}(e_{ki}) = 1/E_i \ \forall k \in \{k|e_{ki} \in E_{comm}\} \quad (2)$$

the probability of transmitting along a possible route is evaluated as

$$P_{ij} = \frac{1/c_{ij}}{\sum_k 1/c_{ik}} \quad (3)$$

among all possible choices available. This energy harvesting aware routing will be used as comparison for NC. For scalability reasons and for upper bounding the delay, we limit the maximum number of different choices that are evaluated to 5. Only minimum hop routes and one hop detour (per end-to-end connection) routes are allowed.

### 2.2. Duty Cycling

Passive energy conservation strategies try to minimize the average power dissipation with DC approaches. They minimize unwanted idle times and unnecessary indirect activation.

As hardware platform we choose Mica2 motes [8] enhanced by EHS technology presented in [9]. The TinyOS2 implementation supports using radio DC for implementing LPL [10]. With implementing a small tweak in TinOS2, the Mica2 ATmega128 can be put to power save mode when the CC1k is shut down.

The application level DC - as presented as performance aware tasking in [11] and other related work - can be implemented upon energy budget information and software task power dissipation as provided by the EHS in [9]. For reasons of testability of scalable systems, we use TOSSIM [12] as simulation environment and annotate EHD profiles and software component power dissipation with highly accurate measurements - for actually the same executable as run on the motes - available from the TOSPIE2 setup [3]. For dealing with dynamic contexts (e.g. protocol changes lead to changing energy consumption) we implement a exponentially weighted moving average and thresholding as described in [13] and [9].

### 2.3. Network Coding

NC is introduced in [14], practical examples are given by [15] and a comprehensive summary can be found in [16]. Linear NC encoding and decoding are operations over galois fields which are finite fields $\mathbb{F}_{2^s}$ with $s$ being the symbol size in bits. The actual size of the field - their order - determines the degrees of freedom for NC. Especially, random schemes make use of higher order fields for more easily selecting coefficients. However, for the coding approaches in this paper $s = 1$ is sufficient. Longer messages can still easily be coded.

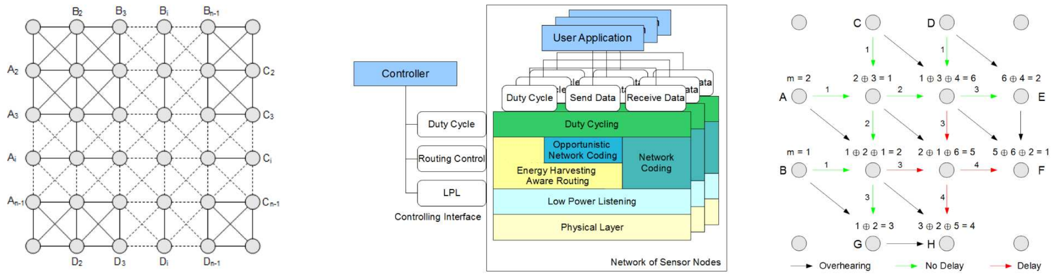
$$X_k = \sum_{i=1}^{n} g_i M_k^i \quad (4)$$

Packets $M^1, ..., M^i, ..., M^n$ are combined with coefficients $g_1, ..., g_i, ..., g_n$ in $\mathbb{F}_{2^s}$ on a per symbol basis. $X_k$ and $M_k^i$ are the $k$-th symbol of messages $X$ and $M^i$.

For wireless NC, the gain is limited to a maximum factor of 2. Often, this cannot be achieved, because there might be too few opportunities where delay and scalability constraints are not violated. ONC tries to maximize opportunities by combining messages from different sessions as well. [17] presents COPE as an 802.11 implementation for ONC. In contrast to that, we implement and practically evaluate ONC for low data rate EHS-enhanced WSNs together with DC and harvesting aware routing and power management for low-power operation with little complexity. While COPE uses reception reports for indicating which messages are available at which nodes, we exploit overhearing transmission and acknowledge information.

## 3. Harvesting Sensor Middleware with Energy Harvesting and Network Coding Service

We choose a mesh structured 8-neighbourhood topology of EHS-enhanced Mica2 motes with TinyOS2. Despite the fact that other EHS hardware [18] has been developed at the department, there is a deeper understanding of the chosen setup from comprehensive measurements that have partly been presented in [19]. Now, as the middleware is optimized subject to low power operation, it is easier to judge networking optimization and understand side-effects when being implemented on a commonly available platform




(a) Topology for $n \times n$ motes (b) Middleware Model (c) Scalable Network Coding

Figure 1. The networking middleware setup.

that has thoroughly been analyzed and tuned in all other aspects as well. E.g. we have tuned deterministically working LPL to an average continuous power dissipation as low as $0.468\,mW$ for the complete mote. Furthermore, we have identified the power profiling error of Avrora which profiles the same program to $9.4\,mW$ on average. After removing an error in its machine model, we could still identify an error of $22\,\%$ in regions that make up more than $70\,\%$ of LPL's energy needs. The mesh topology is chosen for ease of scalability tests and functional simulation is performed in TOSSIM with commonly used channel models. No setup may cover all possible application scenarios. Nevertheless, our setup allows covering several different scenarios and its modular design allows easily extending it for other networking models and application scenarios.

### 3.1. Middleware Model

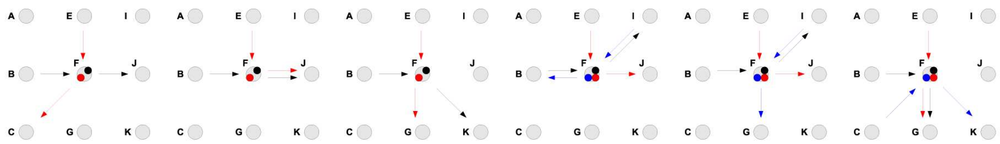
Figures 1(a) and 1(b) show the topology and the layer model. Two communication models are implemented. Firstly, motes connect through from motes $A_i$ to $C_i$ and $B_i$ to $D_i$. Secondly, all motes gather data and send it to the lower right corner node.
The networking model uses a tuned version of LPL as implemented in TinyOS2. Upon that, either SNC or EHAR are implemented. ONC is implemented as a service upon EHAR. The controller part can be implemented outside the framework at host-side or at a designated mote. We do not use any special collision avoidance, but we implement a simple - yet robust - slotted [20] networking model. For a sound comparison we apply a slot order that contradicts the flow of information that has to be sent. This allows modeling worst case scenarios more easily. We aim at setting up simulation scenarios in ways such that their characteristics can be understood and mapped to measurement results that are profiled in hardware measurement setups as done in [21]. We evaluate energy conservation, robustness and delay for the different approaches and discuss the results. A dedicated mote is used for controlling the WSN in all experiments.

### 3.2. Energy Harvesting Aware Routing

After a flooding phase initiated from the data sink(s), the network applies EHAR. Up to 5 different paths with a maximum per hop detour of one hop compared to the shortest path is allowed. Furthermore, we limit the maximum number of detours that may be taken. Thereby, for a $5 \times 5$ setup as shown in Figure 1(a) the maximum number of possible different paths for a single route is limited to 18. If packets were allowed to take one more additional hop, the number of different routes were increased to 108 possibilities and the protocol would not scale well any more.

### 3.3. Scalable Network Coding

SNC and ONC are implemented. For SNC, Figure 1(c) shows a $4 \times 4$ network setup where all motes are active and transmit data. SNC is used in the whole network to reduce the needed amount of packets. Coding with the information from overhearing motes allows conserving $50\,\%$ of inner motes' message transmissions. Only packets of the same session are combined and therefore, the delay continuously increases by 1 at each hop. Although the approach needs to have parallel flows of information overheard at each SNC mote, the approach can be applied more generally than what it might look like at the first glance. Other topologies may implement the same approach as well as long as they have their motes deployed within reasonable distances such that radio density is larger than an 8-neighborhood. Overlapping radio coverage will usually be used for reasons of fault tolerance anyway. E.g. for reasons of testability and comparability of simulation results with hardware power profiling measurements, we consider full overhearing among all motes. This power dissipation of overhearing or not overhearing a transmission would otherwise influence the overall result of the network too much. With full overhearing (and implementing the selection process of which messages may be used such that it comes down to an 8-neighbourhood) we can take it for granted that the profiling results can answer



(a) 2-way ONC (b) 2-way ONC (c) 2-way ONC (d) 3-way ONC (e) 3-way ONC (f) 3-way ONC

Figure 2. Different ONC scenarios with 2 messages and 3 messages.

the question of what energy can really be conserved when using SNC. It allows setting up repeatable hardware network experiments without the need to carefully tune transmission range of motes. Instead, each mote is fully utilizing the available sending power. This gives further worst case bounds for the energy consumption that is being profiled and worst case MAC conditions are achieved as well. Using the TOSPIE2 framework for comparing measurement and simulation results has shown, that this design methodology allows profiling real-world power dissipation (or what can be profiled in testbeds and self-monitoring deployments) more realistically with simulation-based approaches.

### 3.4. Opportunistic Network Coding

Although classical SNC can half the networking power dissipation of WSNs and therefore considerably leverage the sustainable end-user-performance, it cannot explicitly adapt to variation of EHD patterns among different motes. Actually, the theoretical considerations in [22] show that SNC even has the property of evenly distributing the load among motes. The only means of load balancing (on top of energy conservation) can therefore be implemented by using energy management techniques as shown in [22]. However, their applicability depends on whether it is efficient to completely shut down the mote compared to operating the hardware in power save mode. Giving a general solution to this question is not possible, because this applicability tradeoff changes with many application-dependent factors.

Therefore, we present ONC that can be applied to virtually any communication paradigm and any routing protocol. For showing its applicability to energy harvesting with flexibility and energy conservation capabilities, we implement it on top of EHAR and DC for extended low-power capabilities. While standard scalable NC needs to operate on pre-constructed (at least virtually constructed) mesh structured topologies and can only be operated after the second data session or after a negotiation phase, ONC is an ad-hoc-capable approach. The presented ONC approach works without any information on the underlying topology. It operates on the fly with incoming and overheard messages without control packets. It is solely based upon locally gathered information. ONC applicability depends on overhearing conditions of last hop and next hop motes. Figure 2 shows different cases with two and three incoming messages overheard at any receiver (2(a)), one message overheard at a common receiver (2(b)), and one message overheard at one receiver (2(c)). The first and the second example can have ONC applied, the third example can only be coded with help of motes $I$ and $J$. A short bitmap in the header of ONC messages tells next hop motes whether and which routes are combined. For increasing number of messages that could possibly be combined and opportunities that motes could be waiting for, the complexity of the problem increases. While the routes depicted by Figure 2(d) can be handled by decoding with overheard messages alone, it gets more complicated with situations in Figures 2(e) and 2(f) where there is always one receiver that is missing two messages. The proposed model does not implement such cases where messages need to be duplicated or timing of message forwarding is needed for presenting messages at receiver motes that need them for decoding.

### 3.5. Duty Cycling

Applying a DC to WSNs demands for determining EHD rates, the energy budget status and load (mote) power dissipation. The maximum per-mote DC has to be communicated to find a maximum sustainable DC (MSDC) among the WSN which in turn is adapted for the network in case the communication control overhead is below the expected energy conservation.

Motes periodically send their MSDCs piggy-backed with data messages to the controller instance and send requests for lowering the WSN DC in case they have reached their lower threshold [9]. Otherwise, it is the controller's duty to adapt the network or a network region to their MSDC.

## 4. Implementation

Figure 3 gives an overview of the implementation structure and ONC details. The *FrameworkControl* component is responsible for forwarding packets to the responsible component for DC, SNC, EHAR or ONC. Another always-on component is the *SendBuffer* which implements a time slotting such that no collisions occur even if all motes can overhear each other. The *DutyCycling* module always adapts the DC at all motes in the network if needed. The adaptation

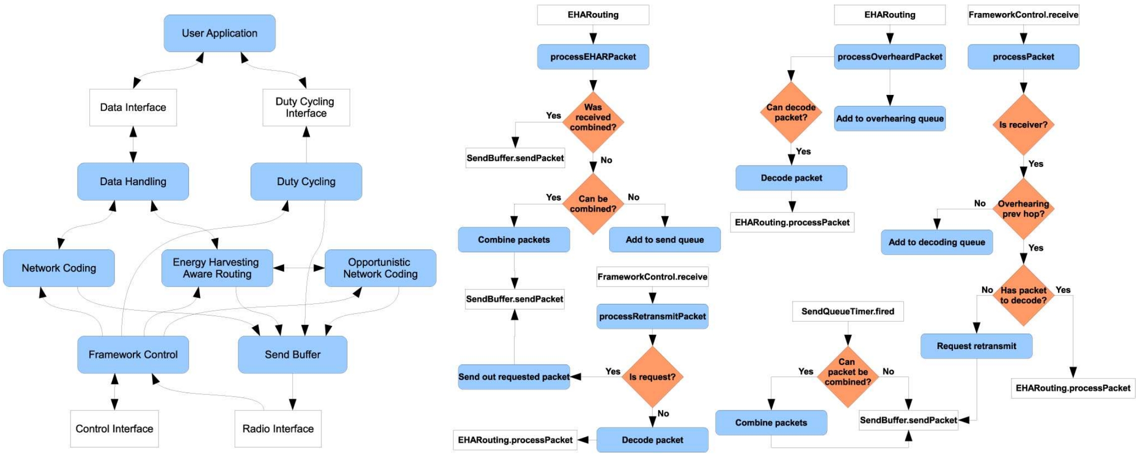
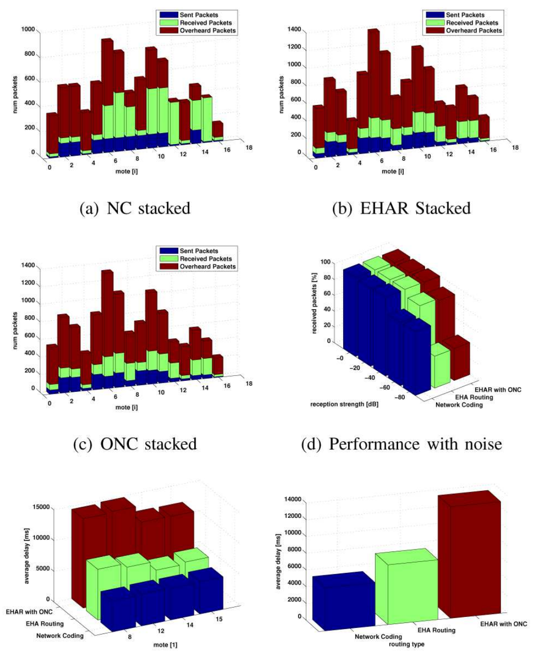



Figure 3. Main networking modules and detailed ONC program flow.

- that can only occur at the end of a modeling period but can be negotiated before - may be initiated by a network sink operated by a host computer or by a dedicated mote in the network. The DataHandling module offers networking interfaces to application level components. Here, the networking type may be chosen. If ONC is chosen, all messages passing through the *EnergyHarvestingAwareRouting* module will be offered to the *OpportunisticNetworkCoding* module.
The detailed ONC program flow is made up of three parts. Firstly, the module checks all EHAR packets whether they are combined. If not, it is checked whether packets can be combined. Therefore, the module has to take into account whether further motes along the route have overheard enough incoming packets and are not the destination if further forwarding is needed for other motes' decoding processes. Secondly, if packets cannot yet be combined with buffered overheard packets, incoming overheard packets are enqueued in the overhearing queue and send queue packets are checked for ONC applicability with the new enqueued packet. Thirdly, packets are not only sent out if ONC can be applied or their maximum waiting time is reached. They are also forwarded if the *SendBuffer* structure is full and a new packet arrives.

## 5. Evaluation of EHAR, NC, ONC and DC

All plots and packet counts include the overhead of protocol initialization. After initialization, 100 packets are sent out at each source mote. Different settings are evaluated that show the different configurations' behavior with different data rate, duty cycling, reception strength, MAC order, and different number of data sinks.




(a) NC stacked (b) EHAR Stacked

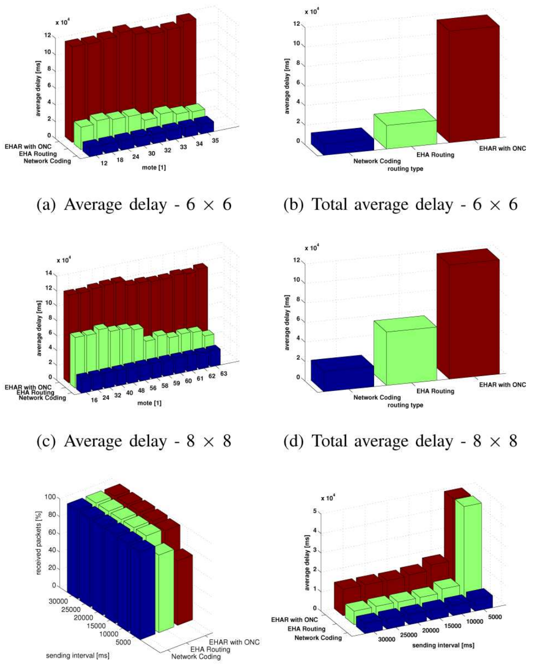
(c) ONC stacked (d) Performance with noise

(e) Average delay per destination (f) Total average delay

Figure 4. Networking performance with DC.

### 5.1. General Performance

Figure 4 shows significant savings when applying a NC approach. With SNC a gain of up to 50 % of messages is

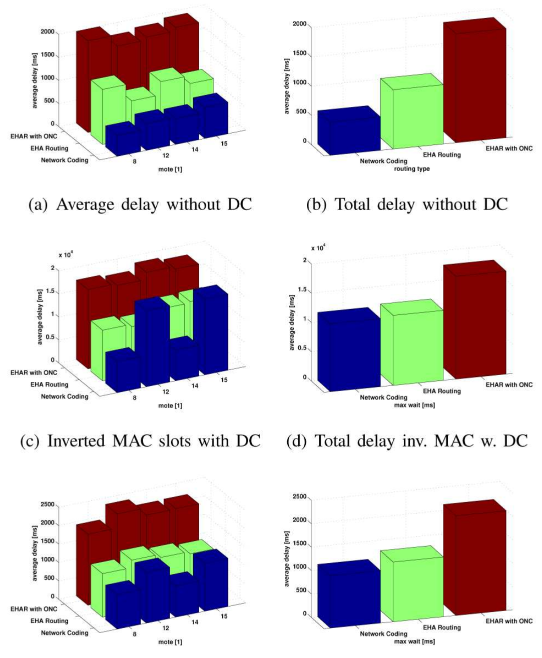


(a) Average delay - 6 × 6 (b) Total average delay - 6 × 6

(c) Average delay - 8 × 8 (d) Total average delay - 8 × 8

(e) Data sending rate performane (f) Data sending rate delay

Figure 5. Delay and throughput with DC.



(a) Average delay without DC (b) Total delay without DC

(c) Inverted MAC slots with DC (d) Total delay inv. MAC w. DC

(e) No DC - inverted MAC slots (f) Total delay inv. MAC no DC

Figure 6. Different source to destination delay.

achieved when initial and transient effects are gone. Even if initial and transient effects were considered in each and every 100 packet experiment, the effect would not go into reverse. The number of messages that make up the protocol overhead are a magnitude of order below the number of data messages that have to be sent. For the NC example that is shown in Figure 4(a), the overall number of packets that are sent as protocol overhead is between 9 and 13 at all motes with an average of 11 overhead packets per mote.

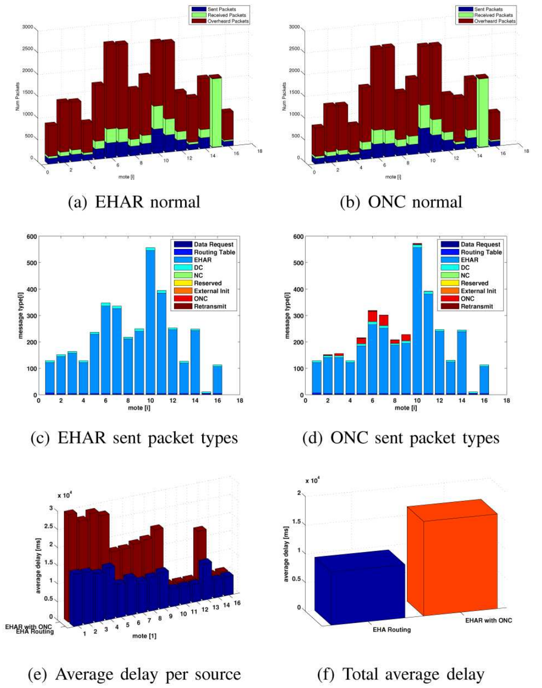
Simulation results may even show an overall gain of more than 50 % when transient effects are gone. This may be the case if an increased packet loss rate is applied. NC tends to be a little less prone to this effect due to their increased inter-packet time. However, this effect is negligibly small and could not be reproduced with hardware experiments.

In Figure 4(d) one can see that NC may be less affected by bandwidth problems. It shows that for bad reception strength - set in TOSSIM - the performance of NC stays at 80 % received packets, while the EHAR reception ratio drops to 40 %. These results also relate to the delay results in Figures 4(e) and 4(f). NC gives the best result in terms of delay. The more general approach of ONC trades time for energy conservation. In general ONC delay depends on the scenario and waiting time settings.

### 5.2. Networking Delay Characteristics

Figure 5 shows delay results for $6 \times 6$ and $8 \times 8$ networks. The delay of ONC reaches the maximum active time of one sending slot and has to wait for a subsequent one. Therefore, it is drastically increased to 120 seconds. The other two approaches stay one magnitude of order below that. Especially, the total average delay of 5.2 for SNC increases to 11.5 and 20.8 seconds.

Finally, the message delay of ONC may also congest message buffer memories of motes. The results from increasing the data rate (i.e. decreasing the time span between two periods of data injection) as shown in Figures 5(e) and 5(f). As expected, ONC still works, but in case memory for a message buffer is short and not extendable, the maximum waiting has to be reduced if the system cannot sustain the rate of a data stream anymore. However, ONC still works reliably when injecting messages every 10 seconds. This will be fast enough for most WSN applications. In case the demands of a high data rate stream could not be sustained that way any more, the concept would have to be reengineered anyway - especially, because other low-power mechanisms like LPL and using a low DC keep WSNs from fulfilling real-time constraints when sending at high data

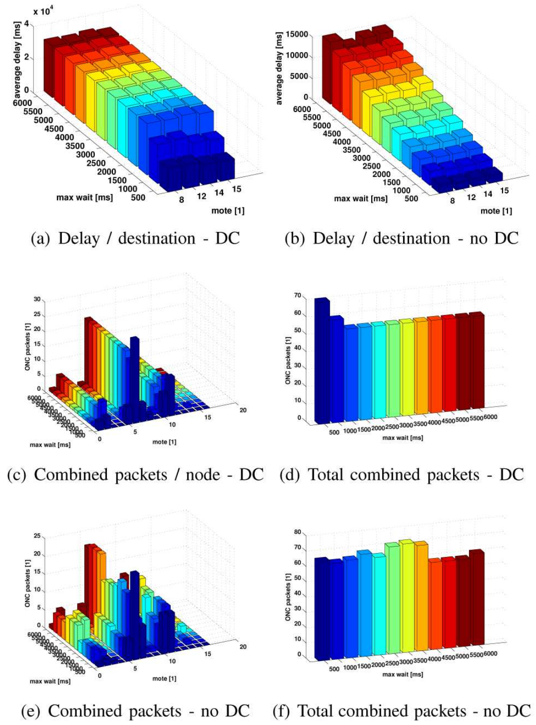


(a) EHAR normal (b) ONC normal

(c) EHAR sent packet types (d) ONC sent packet types

(e) Average delay per source (f) Total average delay

Figure 7. Single sink performance of ONC.



(a) Delay / destination - DC (b) Delay / destination - no DC

(c) Combined packets / node - DC (d) Total combined packets - DC

(e) Combined packets - no DC (f) Total combined packets - no DC

Figure 8. ONC opportunities and delay.

rates.
Delay elaborations continue with incorporating the effects of DC and MAC slot order into the results. Figure 6 gives a brief overview. While ONC still has the longest delay, the delay of NC and EHAR are pretty much the same when using an inverted MAC slot order. The reason is with a worst case MAC slot order, every type of protocol has to wait for another round of MAC slots to get its messages one step closer to a sink. Using a DC may amplify this effect.

### 5.3. Single Sink Performance

Figure 7 evaluates the performance of ONC and EHAR when sending data to a single sink. The energy conservation capabilities of ONC are reduced in case of a single sink. Figure 7(b) shows only little performance increase compared to results in Figure 7(a). However, Figures 7(c) and 7(d) show that up to 13 % of messages could be conserved locally. This implies that the implemented ONC mechanism can be applied in a quite autonomous way and for different patterns of communication topologies. In any case, a dense network structure will promote applicability of ONC. Again, the total average delay of ONC is approximately twice as much as for EHAR alone as shown in Figure 7(f). Figure 7(e) reveals the dependency of message delay from its source mote's hop distance to the destination mote.

### 5.4. Selecting ONC Waiting Time

Figures 8(a) and 8(b) evaluate the average delay of ONC with and without DC as a function of the maximum waiting time. While there is a waiting time dependent step function in case no DC is used, the DC - if applied - dominates the overall timing behavior. Figures 8(c) through 8(f) sum up results of when and where coding opportunities are taken. It can clearly be seen that applying a DC heavily interacts with ONC based on EHAR. Applying a DC makes the overall system a little less flexible in terms of variability of the networking behavior.

## 6. Conclusion and Future Work

Concluding the results, scalable and opportunistic network coding have been implemented for reducing the number of messages that have to be sent in a WSN.
They have been implemented in a realistic setting on a resource constrained WSN motes and have extensively been

studied with simulation runs and with measurements of test-wise deployments of hardware. Tests on real hardware are essentially needed when working with such resource constrained devices. Implementing an resource aware networking layer, but neglecting constraints at other levels would render the approach useless. A number of hardware measurements that have been used to validate that other layers' mechanisms have still been preserved.
All scenarios have thoroughly been evaluated including transient effects as well as static behavior. Results of detailed profiling of available power profiling simulation lead to implementing in-software annotations of results from accurate hardware measurements. Evaluating with this meaningful self-profiling approach, we show that the overall load on the radio may be reduced for $13\,\%$ to $50\,\%$.
Ongoing work elaborates ways for further generalizing the approach for even more easily testing different applications and environmental scenarios. Furthermore, novel directions of energy aware routing will be investigated that allow for better integration of the promising characteristics of network coding networking benefits.

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] P. M. Glatz, C. Steger, and R. Weiss, "Tospie2: tiny operating system plug-in for energy estimation," in *IPSN '10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 410–411.

[4] J. Li, D. Cordes, and J. Zhang, "Power-aware routing protocols in ad hoc wireless networks," *Wireless Communications, IEEE*, vol. 12, no. 6, pp. 69 – 81, dec. 2005.

[5] J. Gomez, A. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving transmission power in wireless ad hoc networks," nov. 2001, pp. 24 – 34.

[6] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, vol. 1, mar. 2002, pp. 350 – 355.

[7] A. Kansal, J. Hsu, M. Srivastava, and V. Raqhunathan, "Harvesting aware power management for sensor networks," in *43rd ACM/IEEE Design Automation Conference*, 2006, pp. 651 – 656.

[8] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," in *Technical report, Computer Science Department, University of California at Berkeley*, 2001.

[9] P. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in wsns," in *IFIP/IEEE Wireless Days*, nov. 2008, pp. 1 – 5.

[10] S. Moon, T. Kim, and H. Cha, "Enabling low power listening on ieee 802.15.4-based sensor nodes," in *IEEE Wireless Communications and Networking Conference*, 2007, pp. 2305 – 2310.

[11] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 223–234, 2004.

[12] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 126 – 137.

[13] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.

[14] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204 – 1216, jul. 2000.

[15] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63 – 68, 2006.

[16] D. S. Lun and T. H. Ho, *Network Coding - An Introduction*. Cambridge University Press, 2008.

[17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497 – 510, jun. 2008.

[18] P. M. Glatz, L. B. Hörmann, and R. Weiss, "Designing perpetual energy harvesting systems explained with rivermote: A wireless sensor network platform for river monitoring," *Electronic Journal of Structural Engineering, Special Issue: Wireless Sensor Networks and Practical Applications (2010)*, pp. 55 – 65, 2010.

[19] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency," in *IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing*, mar. 2010, pp. 468 – 473.

[20] R. Flury and R. Wattenhofer, "Slotted programming for sensor networks," in *IPSN '10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 24–34.

[21] S. Ruff, "Master's thesis. an energy harvesting networking service for wireless sensor networks using network coding," *Institute for Technical Informatics, Graz University of Technology, Austria*, 2010.

[22] P. Glatz, J. Loinig, C. Steger, and R. Weiss, "A first step towards energy management for network coding in wireless sensor networks," in *9th IEEE Malaysia International Conference on Communications*, dec. 2009, pp. 905 – 910.

Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications
15 -17 December 2009 Kuala Lumpur Malaysia

# A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks

Philipp M. Glatz*, Johannes Loinig†, Christian Steger‡ and Reinhold Weiss§
Institute for Technical Informatics, Graz University of Technology
Inffeldgasse 16/1, 8010 Graz, Austria
*Philipp.Glatz@TUGraz.at, †Johannes.Loinig@Student.TUGraz.at, ‡Steger@TUGraz.at, §RWeiss@TUGraz.at

***Abstract*—Network coding is a suitable mean to come up against the effects of crossing information flows in wireless sensor networks (WSNs) where these areas are heavily impacted in terms of channel bandwidth, message delay and energy balance.**
**While only few work exists that deduces general rules for resource conservation with network coding, no approach exists that formalizes the impact of scheduling network coding to optimize persistent storage efficiency.**
**We present a novel approach that can be used for network-wide energy-aware behavior. We consider the usage of persistent memory and discuss the access patterns' impact for state-of-the-art WSN technologies.**
**Our approach works for static mesh-structured WSNs. We show how to locally inhibit network information flows to rebalance the load on energy conservation among different motes. Our results from theory and simulation show how to scale the concept to arbitrarily large WSNs.**

## I. Introduction

Wireless sensor network (WSN) motes are resource-constrained devices. They suffer sparse bandwidth and limited power supply. Both issues - bandwidth limitation and bounded power supply - can be relaxed with network coding. Network coding allows to transmit separate flows of information in a single step. The information of both crossed flows can be decoded at the according sink node if the topology, coding function and other information sources are known. Most work in the area of network coding concentrates on gathering or controlling these settings. Translating bandwidth savings into energy savings for WSNs can be a promising approach. Few previous work has discussed the wireless scenario, but first steps have been taken towards energy conservation in WSNs through mappings from bandwidth gain recently.

### A. Contribution Claim

In this paper we will broaden the view on effects of network coding for WSNs. Not only that the data has to be forwarded in multi-hop networks in an energy-efficient manner, it also has to be stored persistently inside the network too. Numerous reasons exist: low data rates may force nodes into low power modes and to resetting volatile memory, data aggregation may demand for waiting and therefore storing data before forwarding it and many other opportunistic optimization methods may pose the need. One prominent example is network coding on duty cycled WSNs itself. We will show how to code packets over multiple crossings of information flows and keep them recoverable. In parallel to that we will consider the number of different power cycles of a mote in which persistent memory has to be accessed for network coding. This has not been done before to the best of our knowledge. We will then optimize the usage of persistent memory access patterns. Accessing persistent memory in two power cycles is way more costly than accessing it twice in one cycle due to initialization and completion overhead. We will show how to trade energy from motes with a higher residual energy level to reduce the energy drop at constrained motes and we therefore inherently balance the drop of residual energy among different parts of the network.

### B. Outline

The subsequent section will discuss related work for the field of network coding for WSNs. Next, we will discuss the concept and set up the simulation scenario. This will be implemented and results on residual energy balancing through optimization of memory access patterns will be shown. In the remainder of the work we will summarize what has been shown and give some outlook on future work.

## II. Related

We will introduce related work for network coding that describes how it works and what can be gained from it for WSNs. Next, we explain its relation to underlying protocols like the media access protocol (MAC) and energy harvesting duty cycling. There we will find the need for accessing persistent memory. This will finally be discussed for state-of-the-art WSN motes.

### A. Network Coding for Wireless Networks

A basis for network coding has been established by Ahlswede et al. [1]. Practical examples similar to the single crossing case of a network coding butterfly in Figure 3 are explained by Fragouli et al. [2].
Ho et al. [3] discuss the maximum gain of network coding over routing. For the wireless case where xor-coding is used over GF(2) as shown by Katti et al. [4] the gain is bounded by 2. An approach that translates savings of bandwidth to energy savings has been shown with LINDONCS: Localized In-Network Detection Of Network Coding Structures [5]. Now, in this work we will add localized algorithms for energy conservation from reducing time slots with memory access.

978-1-4244-5532-4/09/$26.00 ©2009 IEEE 

### B. Energy Conserving Access Protocols and Harvesting

The network coding approach cannot do any better than the underlying MAC-protocol. For state-of-the-art WSN hardware of small form factor like the Mica and Telos family of motes by Polastre et al. [6] and Hill et al. [7] TinyOS is the de-facto standard for research based upon these platforms. The implementation of TinyOS 2 supports BMAC for low-power-listening (LPL) as described by Moon et al. [8]. Ganeriwal et al. describe UBMAC [9] with uncertainty driven preamble length reduction. This improves over LPL with BMAC, but it is not so lightweight and may be less robust to unexpected timing errors so we will go for modeling standard LPL. These different approaches lead to different ratios for the additional average power dissipation of a sending mote compared to the receiving one. Obviously, one has to model this ratio to fit a given scenario's MAC. As has been shown in [5] this ratio heavily impacts the gain that can be achieved. We will fix the ratio to 2 first which comes close to the measurements of standard LPL on Mica 2 motes. For allowing to map the concept given in this paper to different scenarios we will keep the ratio of average additional power dissipation from persistent memory access overhead to listening variable.
We have initially mentioned the case of energy harvesting system (EHS)-enhanced WSNs. These try to negotiate the maximum sustainable duty cycle as described by Kansal et al. [10] where motes are completely shut off in between the power cycles. A system of that type has been implemented where data that could not be forwarded in the last power cycle has to be stored in the mote's flash memory or EHS [11] memory which is even more costly due to timing overhead.

### C. The Overhead for Accessing Persistent Memory

For state-of-the-art implementations of applications with TinyOS storing data into flash in different power cycles has stronger impact on the mote's residual energy level than having both writes in the same power cycle. In the latter case we save initialization time and have to ensure data persistency with the sync command only once. Detailed explanation of the interfaces in use and examples of using the flash on Mica 2 and other motes are given in the TinyOS tutorials based on TEP 103. Further strategies for concurrency control and energy management of device drivers have been published by Klues et al. [12].
Not only do we have to cope with the delay from the overhead of controlling hardware components, we also have to consider the impact of every networking protocol optimization on the application level timing. Therefore we will give a short analysis on the delay to energy conservation trade-off for implementing network coding in WSNs with multiple crossings of information flows compared to timely and energy conserving routing. Retransmitting messages implies buffering the messages to some extent which shows similar effects as shows opportunism in other work previously described by Katti et al. [13]. The longer we wait before data is being withdrawn, the more MAC slots we send redundant messages and the more we combine data to packets of larger size, the more we will get additional degrees of freedom for possible combinations of information flows, but we will pay with message delay and maybe also with drops in residual energy. The goal of the concept that will be presented next will be to inhibit residual energy drop while keeping the delay that is introduced below the application's real time constraints that uses network coding as a networking service.

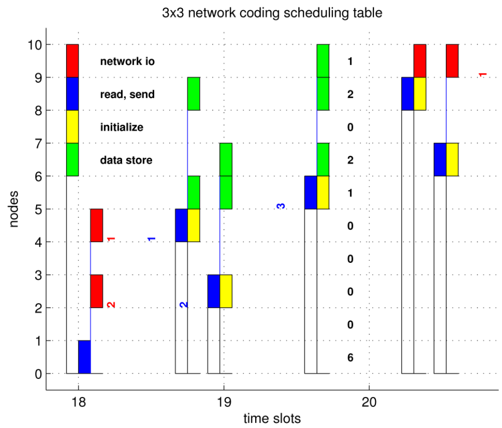



Fig. 1. Two time slots with the according worst case MAC slots being overdrawn are shown for a 9-node network. Packets are injected at source nodes 2 and 4 and cross at 5.

## III. Balancing Radio and Memory Load for Leveling Residual Energy Among Nodes

Power dissipation from using the radio module and accessing persistent memory drains the energy reservoir. Network-wide energy-awareness tries to balance load so that less energy is consumed in undersupplied areas by reducing the overhead from accessing persistent memory. As an acceptable trade-off more messages have to be sent in better supplied areas. For reasoning about this concept we will introduce a testbed for network coding with multiple crossings from crossing sets of unicasts. We will introduce examples with 2 and 6 sending nodes and show a radio and memory strain map for the usage of network coding under worst case MAC scenarios. Based on that we will present results for delay and energy loss and propose possible optimizations for leveling residual energy among motes.
This paper follows a simulation based approach. As examples we will use a $3 \times 3$ and a $5 \times 5$ scenario. The $3 \times 3$ network has its network coding scheduling table shown in Figure 1. Nodes from that network that perform network coding form a minimal butterfly structure and are depicted in Figure 3. Due to the fact that a 9-node mesh network does not allow to generalize to arbitraly large networks we consider a 25-node network too. The scheduling table in Figure 2 and network topology in Figure 3 allow to generalize in that one can see which messages are needed for keeping injected packets recoverable

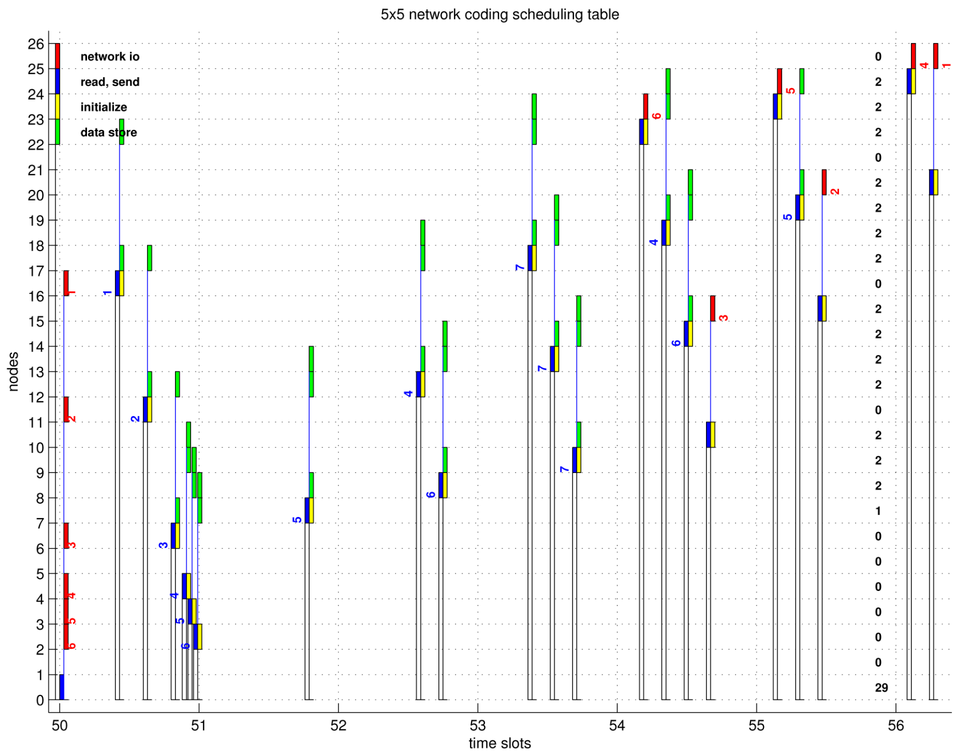



Fig. 2. Seven time slots with the according worst case MAC slots being overdrawn are shown for a 25-node network. Packets are injected at source nodes 2, 3, 4 and 6, 11, 16 and cross at 7, 8, 9, 12, 13, 14, 17, 18 and 19. More degrees of freedom for when to store messages exist if a retransmission protocol exist. Retransmissions are bounded by the maximum delay of the network that still fits its real time constraints.

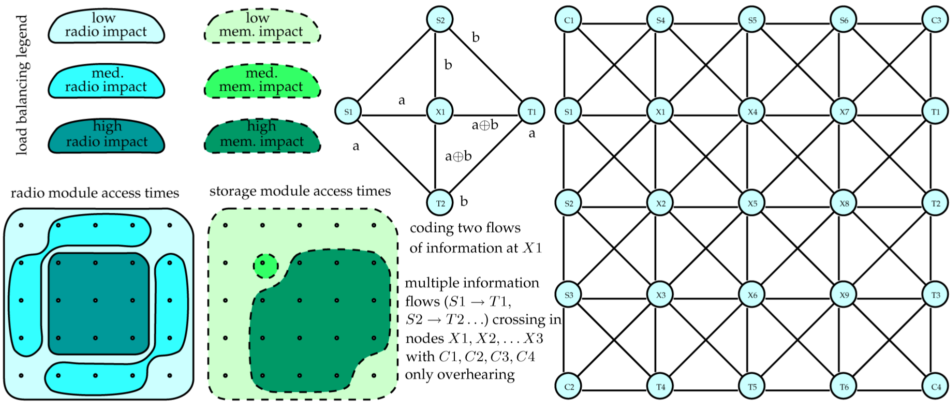



Fig. 3. The scenarios consider multiple crossings of information flows. We will trade additional messages for savings of memory access. The radio and memory strain map are shown for injecting a single packet for each of six unicasts in a $5 \times 5$ network. See the scheduling table in Figure 2. It shows where and when 16 messages are delivered and 29 power cycles with memory writes.

and how they are delayed propagating through the network. Messages are annotated with their payload accordingly.
In the results section we are going to present the residual energy level of different motes evolving from radio and flash module usage for a $5 \times 5$ network. Therefore we show the radio and memory strain map in Figure 3.
**Radio Strain Map**: corner nodes suffer the least impact from radio. They only overhear communication which does not reduce network lifetime, because due to their high residual energy level other nodes will run out of energy first. The reason to keep them awake and listening ist that they can help in the optimization steps proposed in our approach. The sending $[S1, S2, \ldots, S6]$ and receiving $[T1, T2, \ldots, T6]$ nodes suffer similar impact from radio activation. For sending motes less overhearing may occur while the target will be strained dependent on the application or a network surrounding the structure shown. Centre nodes are strained most and deserve most consideration in optimization steps.
**Memory Strain Map**: here we show that sending and corner nodes do not need to store data into persistent memory, because they either have no messages to send or send them immediately. The upper left of the inner nodes has all messages arriving in the same time slot which leads to only one power cycle with write access. All the other nodes will have two write access times to save data among different power cycles until their next MAC slot is available.

### A. Scheduling Table, Time Slots and MAC Slots

In our concept we consider a bandwidth of the wireless channel of one message per time slot for each node and a worst case order of MAC slots within each time slot. On the one side this could be optimized somewhat and therefore deviates from realistic scenarios - the bandwidth of arcs around $S2$ is independent from arc $S6$. On the other side a tuned MAC order cannot be mapped to arbitrary networks anymore. So we decided to use the worst case MAC slot order that contradicts the unicasts' directions within each power cycle. We will further call each activation period a time slot.
Figure 1 depicts incoming messages with sending data from node 0 as network input at nodes indicated by red boxes which leave the network with the same value at red boxes at the end of the simulation again. After messages have been sent (blue) to other motes the messages have to be stored locally (green) if they cannot yet be processed within the same time slot. Initialization of a node's memory indicates that its data need not be kept for future time slots any more. Before the last time slot starts we summarize the number of time slots where write access was necessary for every node. Furthermore we add up all writes among the network in the last line. In Figure 2 the same legend is used. There one can see that times when packets arrive that have to be combined can be far apart. Consider the message injected - or sensed - at node 16. In timeslot 50 the network has allowed enough time to flood the network from all the sink nodes and acknowledge for every type of topology. So the message is being sent to nodes 17 and 22 at that point in time. These have to store the message persistently, because node 22 has to combine the data with the one from messages received in time slot 53 from node 17 which in turn relies on one further message too. This leads to a memory access pattern that is shown in the memory strain map in Figure 3.

## IV. Network Coding Testbed Implementation

We have used Matlab to implement a simulation based approach where several parameters can easily be varied without the need for changing the implementation to adapt to that.

### A. Testbed Parameters

The following parameters are most important for the simulation leading to the results given in this paper.
**Network Size:** we implement all mote rules, adjacency matrices, packet injection, delivery and output with modulo arithmetic. This allows to scale the simulation runs to networks of arbitrary size and keeps the implementation viable for being mapped to implementation on real hardware.
**Initial Energy:** can be used to compare different approaches' lifetimes in that the simulation stops the first time a node's residual energy level has dropped to 0.
**Energy-Awareness:** we also provide a routing implementation for comparison. Energy-Awareness can be switched on so that messages are forwarded to nodes with a higher residual energy level. The maximum additional hop distance that may be used for selecting a better powered node can be set.

### B. Network Coding Rules

For implementing network coding for two sets of crossing unicasts a minimum of two and a maximum of three information sources has to be combined. In mesh networks where inner nodes have a degree of 8 this suffices to code all the data along the crossings. Consider unicast $\mathcal{F}_{S4}$ in the $5 \times 5$ setup from Figure 3. On its way there are three crossings at $X1$, $X2$ and $X3$. In every $Xi : i \in [1 \ldots 9]$ we combine the combined or source packets from nodes on the left and above. At $X1$ there are only two packets. So we forward $\mathcal{F}_{X1} = \oplus(\mathcal{F}_{S4}, \mathcal{F}_{S1})$ to $[X2, X4, X5]$. In the next time slot we will combine the three upper left packets that are sent to $X2$ and forward to $[X3, X5, X6]$ the combination $\mathcal{F}_{S2} = \oplus(\mathcal{F}_{X1}, \mathcal{F}_{S1}, \mathcal{F}_{S2})$. This implicitly cancels out $\mathcal{F}_{S1}$ that was included in $\mathcal{F}_{X1}$. In the next time slot $\mathcal{F}_{X3}$ will add $\mathcal{F}_{S3}$ and remove $\mathcal{F}_{S2}$ while $\mathcal{F}_{S3}$ will be cancelled out at the target receiver $T4$ so that the original information $\mathcal{F}_{S4}$ will be restored.

### C. Residual Energy Level Balancing

We can trade delay for balancing the residual energy level among different nodes. This will be used for simulation of EHS-enhanced motes with variance of input power patterns among different motes. Now we balance energy in that we save overhead from accessing persistent memory in that we inhibit messages from being stored in time slots where the receiver still has to wait for other sender's MAC slots in other time slots. Inhibition works where nodes of the crossing area

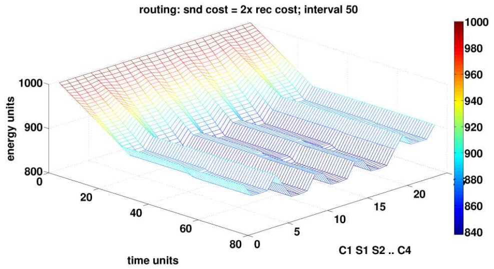



Fig. 4. The residual energy with routing.

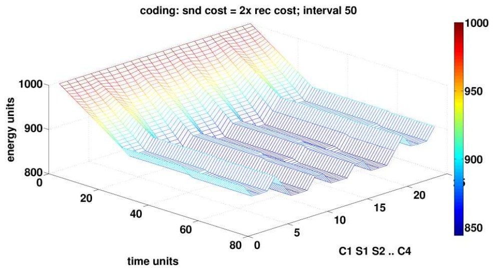



Fig. 5. The residual energy with coding.

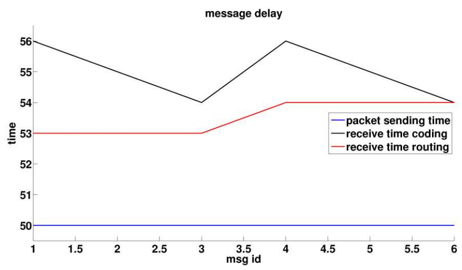



Fig. 6. The delay of routing and network coding.

boundary's nodes are receiver. Another option is to retransmit packets in time slots where other source nodes for a given receiver have their information available too. Retransmission can be performed by the node initially sending the data or overhearing nodes replicating the data.

## V. Residual Energy, Delay and Optimization

In this section we summarize residual energy, delay, retransmission patterns, their chaining, oscillation and how the method of energy balancing scales for large networks. Here chaining means to further propagate energy gain and oscillation is the case when chaining can be fed back again.

### A. Network Coding and Routing

In Figure 4 and 5 we show residual energy over time for routing and for network coding. Already for these short runs it can be seen that network coding translates savings in bandwidth to energy conservation. Furthermore, when network coding is used there is less variance in the residual energy level of motes that are strained most.

### B. Optimization Patterns, Chaining and Oscillation

For the $3 \times 3$ setup we can save the constrained mote number $5$ memory access cycles with replication of diagonal messages at nodes $6$ and $8$ or retransmission of packets at $2$ and $4$. The same effects can be used for the $5 \times 5$ setup and can be scaled arbitrarily with chaining.
For all the effects discussed we will mention type I approaches at the boundary and type II approaches at nodes that gain from type I for chaining the gain to nodes in the core network.

### C. Network Lifetime for Variable Memory Costs

In Figure 7 we show different retransmission and inhibition schemes for $5 \times 5$ networks. Inhibition to wait for delayed messages can only be seen from networks larger than $3 \times 3$ with more than one crossing.
**Retransmission:** the message retransmission type I at the sending nodes leads to a reduced number of persistent writes for $4$ transmitter and $2$ target nodes. In scenarios with storage overhead to listening cost ratio above $1$ we propose message retransmission type II where messages overheard along diagonal communication links from upper left nodes are retransmitted when the lower and right neighbors will have their MAC slot in the subsequent time slot. Type II can further propagate the balancing approach towards the lower right paying the local overhead cost after having gained from retransmission of type I.
**Replication:** at corner nodes we can replicate messages with type I which promises the best trade-off to be found. Type II replication of messages is performed by two nodes working together similar to nodes $6$ and $8$ in the $3 \times 3$ network. The advantage of this type of optimization is that it can evolve towards source nodes. The drawback is its cost that is higher per effect than for any other type.

### D. Message Delay

The maximum delay from routing for the scenarios used is bounded by $\sqrt{n} - 1$ due to path length and congestion at the inner nodes as can be seen from the delay plot of a $5 \times 5$ network in Figure 6. For network coding we depend on the depth of the data dependency graph. For the MAC structures used here this bounds the delay between $\sqrt{n}-1$ and $2 \cdot \sqrt{n}-4$.
**Single Message Delay:** Trading a single delay slot for energy balancing leads to one additional delay slot for all unicast targets that reside in the lower right of the inhibiting node.
**Accumulation of Delay:** Considering cases where the source dissipation rate fits the bandwidth of the network with network coding - or much more the messages have been designed to make the bandwidth fit the source rate - delay introduced at any part of the network will be accumulated. The delay will exist as long as no messages are dropped, the source dissipation rate is reduced or the MAC order has been changed. Detailed studies of these metrics are part of future work on networking service performance.

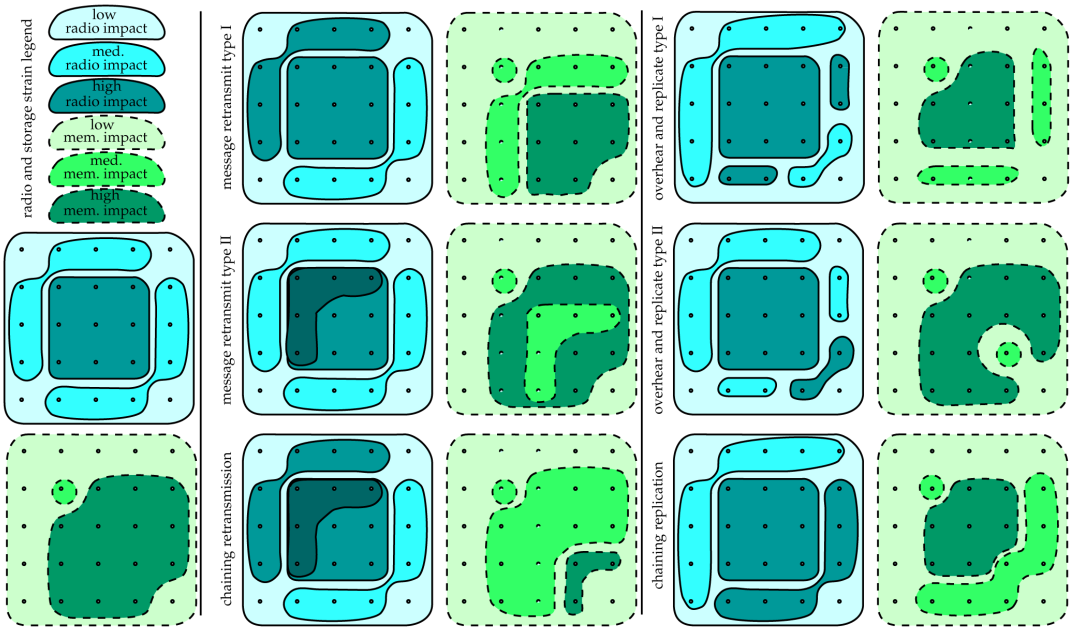



Fig. 7. This figure shows different types of optimizations for leveling the residual energy balance in the network. On the left side a legend codes the level of radio and memory access strain and shows radio and memory strain maps for the standard implementation of our concept. Different types of different optimizations with their effects are shown in the middle and on the right. We suggest to use type I optimizations initially and type II as follow ups. Chaining retransmission type I and retransmission type II shows how savings in terms of a reduced number of power cycles with memory access feed through the network to the lower right. This type of retransmission can be applied at all places in arbitrary large networks. A similar behavior can be seen for the replicate scheme and its chaining. Applying both complete chains after each other leads to oscillation of the optimization procedure forth and back through the network.

## VI. Conclusion and Outlook

This paper presents a method for trading message delay for balancing WSN motes' residual energy levels. The simulation based approach can be configured to fit different platforms and application scenarios. For the scenarios presented here a variance in the number of power cycle when persistent memory is accessed has been introduced that may save up to 50 % of access times at single motes. These savings of overhead directly translate into energy savings for WSN motes in undersupplied areas.

The approach fits the demands of energy-awareness on a network level very well. It allows to trade residual energy at well supplied nodes for reduced energy drop at undersupplied nodes. Applying sets of optimization methods one after the other allows to propagate such savings through the network.

Ongoing work is concerned with providing a formalism for energy-aware algorithms. The possibilities of energy balancing shall be deduced in closed form. These will then be applied in settings with EHS modules attached to balance different power supply patterns at different parts of a network.

## References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Information Theory*, 2000.

[2] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, 2006.

[3] D. S. L. Tracey Ho, *Network coding: an introduction*. Cambridge, New York, Cambridge University Press, 2008.

[4] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Trans. Networking*, 2008.

[5] P. M. Glatz and R. Weiss, "LINDONCS: localized In-Network detection of network coding structures in wireless sensor networks," in *ACM PM2HW2N*, 2009.

[6] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN*, 2005.

[7] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," 2001.

[8] S. Moon, T. Kim, and H. Cha, "Enabling lpl on ieee 802.15.4-based sensor nodes," in *WCNC*, 2007.

[9] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *SenSys*, 2005.

[10] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, 2007.

[11] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in WSNs," in *IFIP Wireless Days*, 2008.

[12] K. Klues, V. Handziski, C. Lu, A. Wolisz, D. Culler, D. Gay, and P. Levis, "Integrating concurrency control and energy management in device drivers," *SIGOPS Oper. Syst. Rev.*, 2007.

[13] W. H. H. R. S. Katti, D. Katabi and M. Medard, "The importance of being opportunistic: Practical network coding for wireless environments," 2005.

*Int. J. Autonomous and Adaptive Communications Systems, Vol. X, No. X, xxxx* 1

# Scalable network coding for wireless sensor network energy conservation

Philipp Maria Glatz*, Karima Barbara Hein and Reinhold Weiss

Institute for Technical Informatics,
Graz University of Technology,
Inffeldgasse 16/1, Graz 8010, Austria
E-mail: Philipp.Glatz@TUGraz.at
E-mail: Glatz.Philipp@gmail.com
E-mail: Hein@TUGraz.at
E-mail: RWeiss@TUGraz.at
*Corresponding author

**Abstract:** Wireless sensor network (WSN) nodes called motes are small wireless embedded devices. They suffer from limitations of their power supplies and available bandwidth as well as in their memory and computational power. As a promising approach, network coding (NC) provides means of relaxing the combined problem of limited supplies and bandwidth. We present experimentation in modelling, simulation and measurement of a novel approach for robust scalable NC in dynamic environments. It is compared to routing in mesh structured, low data rate WSNs. For keeping it from being purely academic, we discuss the scalable concept's applicability to different applications, systems and topologies. We evaluate its energy conservation capabilities for Mica2 motes and show savings of up to 50 %. This paper also highlights trade-offs and solutions that are applicable to other state-of-the-art systems as well.

**Keywords:** WSN; wireless sensor network; NC; network coding; energy conservation; scalability; modelling; simulation; measurement.

**Reference** to this paper should be made as follows: Glatz, P.M., Hein, K.B. and Weiss, R. (xxxx) 'Scalable network coding for wireless sensor network energy conservation', *Int. J. Autonomous and Adaptive Communications Systems*, Vol. x, No. x, pp.xx–xx.

**Biographical notes:** Philipp Maria Glatz received his BS and MS in Telematics specialising in System-on-Chip-Design and Computational Intelligence from Graz University of Technology, Austria, in 2005 and 2007, respectively. As a University Assistant with the Institute for Technical Informatics at Graz University of Technology, Austria, he is assigned research and teaching duties. Currently, he is a PhD candidate in Electrical and Computer Engineering. His research interests include wireless sensor networks and power awareness with a focus on network coding, energy harvesting, measurement and integration. He is a Member of the Institute of Electrical and Electronics Engineers.

Karima Barbara Hein is a Research and Teaching Assistant in the Institute for Technical Informatics at Graz University of Technology, Austria, where she is currently pursuing a PhD in Electrical and Computer Engineering. She received

her BS and MS in Telematics specialising in Technical Informatics and Pervasive Computing from Graz University of Technology, Austria, in 2004 and 2007, respectively. Her research focuses on wireless sensor networks and dependability with a special interest in fault tolerance.

Reinhold Weiss received the Dipl-Ing and Dr-Ing degrees in Electrical Engineering and the Dr-Inghabil degree in real-time systems from the Technical University of Munich, Germany, in 1968, 1972 and 1979, respectively. In 1981, he was as a Visiting Scientist with IBM Research Laboratories, San Jose, CA. From 1982 to 1986, he was a Professor of Computer Engineering with the University of Paderborn, Germany. Currently, he is a Professor of Electrical Engineering and the Head of the Institute for Technical Informatics, Graz University of Technology, Austria and the Author or a Coauthor of more than 170 publications. He is a Member of the International Editorial Board of the US journal Computers and Applications, the Institute of Electrical and Electronics Engineers, the Association for Computing Machinery, Gesellschaft für Informatik and the Austrian Association for Electrical Engineering.

## 1 Introduction: motivation and completeness

Sensing devices typically integrated into embedded platforms and networked wirelessly for cost reasons are referred to as wireless sensor network (WSN) nodes or motes as surveyed by Akyildiz et al. (2002). Different motes may exhibit different cost and performance constraints and different form factors ranging from the vision of smart dust by Warneke et al. (2001) and HVAC applications up to such as satellites. This paper considers state-of-the-art embedded platforms like the Mica2 or TelosB family of motes introduced by Hill and Culler (2001) and Polastre et al. (2005).

For reasons of cost-efficiency, motes are implemented using as few resources as possible. Irrespective of whether the overall architecture is targeting mobile multimedia application or low data rate applications for environmental monitoring, correct functionality and power dissipation pose the most stringent constraints in most cases. As sending and listening operation of the radio is often needed and may be very power consuming, it is common sense that networking and the number of messages that need to be sent is subject to optimisation for WSNs. Glatz et al. (2010b) give an overview of general power supply and dissipation issues and state-of-the-art solutions for tackling that problem.

A specific optimisation dealt with by Glatz et al. (2009a) applies NC to state-of-the-art WSN technology for reducing the number of messages that need to be sent for reasons of energy conservation. NC has first been introduced and analysed by Ahlswede et al. (2000) showing its capabilities for increasing throughput which can be described with the max-flow min-cut theorem. This paper provides concept, implementation, analysis and discussion of the completeness of the approach and answers questions that remain open in the work presented by Glatz et al. (2009a).

### *1.1 Contribution and further outline*

We show the generalisation of butterfly structures for wireless scenarios to cross perpendicular groups of virtual unicast information flows such that all the information can be decoded at all sink nodes accordingly. This especially fits the need of fault tolerant routing and novel paradigms like many-to-many communication presented by Mottla and Picco (2010).

### *1.1.1 Decentralised energy conservation*

The approach is scalable to arbitrarily large networks which allows for better insights to what can be gained compared to routing in terms of energy conservation. Generalisation issues will be discussed showing the possibility to locally apply the algorithms to virtually any kind of topology with crossing flows of information.

The method allows conserving energy by reducing the number of messages that needs to be sent by a factor of 2. Despite these substantial savings, the approach is also light weight. Hardly any code overhead occurs and it is of little computational complexity which allows implementing it on small motes.

NC can start after a routing initialization phase and it is possible to switch between NC and routing. Motes in the network need no further information in case of being implemented upon a mesh network topology provided diagonal overhearing.

### *1.1.2 Generically applicable scalable NC*

For giving an equitable comparison to routing, not only power awareness metrics will be considered, but also other main dimensions of dependability as well. Fault tolerance, initialisation time and message delay and cross-layer issues when interacting with media access control (MAC) and energy management schemes will also be discussed.

The approach is applicable to arbitrarily large networks. Both, mote and testbed implementation, are fully scalable. The maximum additional delay is bounded by $\Theta(\sqrt{n})$ and modelling time increases with $\Theta(n)$ where $n$ is the network size.

Important considerations when implementing NC on WSN motes are discussed. Implemented upon a state-of-the-art MAC protocol and operating system using TinyOS we explain why the ratio of sending cost to receiving cost in low power listening (LPL) is an important factor to consider. We discuss the sending to receiving cost ratio deduced from measurements of Mica2 hardware and other hardware dependent issues as well.

### *1.1.3 Article organisation*

After outlining related work we will first describe the conceptual setup. This includes short examples that also cover a generic implementation and measurement setup i.e. employed later on. The combination of setups will be discussed in two ways. Firstly, modelling, simulation and measurement results are used to provide a detailed discussion on what can be gained in terms of energy conservation when using the approach. Secondly, we discuss how the quite abstract implementation of NC in a proprietary Matlab modelling environment can be mapped to TinyOS applications, their simulation and finally, mote hardware experiments. A conclusion will show implications for future designs that can be deduced from that.

## 2 Related work impacting NC design

We briefly overview state-of-the-art WSNs and NC in Section 2.1. Next, as mentioned in the introduction, it is necessary to set up equitable settings for comparing NC to other approaches. Finally, Section 2.3 provides an introduction to MAC as well as power awareness in general and energy harvesting systems (EHSs). These issues affect the static design, rate control demands and the need for robust communication and the use of an acknowledgement (ACK) system.

### *2.1 WSNs and NC*

Starting with the vision of smart dust by Warneke et al. (2001). WSN as described by Akyildiz et al. (2002) have matured over the years. Though, especially the vision of smart dust has not been interweaved with the environment as pervasive as envisioned yet, several standard bodies have contributed to the field of WSNs covering a range of application issues as further outlined by Yick et al. (2008).

Currently, novel hardware and modelling techniques utilising EHS techniques start to mature. These allow envisioning energy neutral operation of networks which leads to new dependability aspects. For reasons of completeness this has to be considered when developing the energy conservation technique at hand. From a cross-layer optimisation perspective – as elaborated with TOSPIE2 presented by Glatz et al. (2010b) – constraints and opportunities have to be considered that stem from other levels of the networking stack or the hardware. Otherwise, the approach's general applicability could not be claimed. Therefore, constraints coming from indisputable goodputs of using EHSs as exemplified by Glatz et al. (2008) are discussed in Section 2.3.

The idea of NC with its algebraic description stems from the field of graph and information theory introduced by Ahlswede et al. (2000). Using NC in mobile *ad hoc* networks (MANETs) by Wu et al. (2005) allows minimising the energy spent per bit. Furthermore, the problem of finding optimal network codes can be related to the Steiner-Tree packing problem by Jain et al. (2003). This results in lower complexity control overhead for NC in polynomial time than for finding the minimum-energy routing multicast tree which is NP-hard.

While the NC advantage may grow further with growing wired networks, the throughput gain for the wireless case is bounded by 2 as described by Ho and Lun (2008). Most work in the field of wireless NC does not explicitly consider the energy gain in comparison to routing. Some even cut out power requirements completely and rely on technology with a form factor of laptops as Katti et al. (2006) with fairly relaxed constraints compared to WSNs. Combining these facts, we conclude that the idea of NC can be beneficial for WSNs. Only a few studies – as outlined by Jacobsen et al. (2008) – deal with NC for low data rate WSNs with considering the technology's powering constraints. A detailed discussion of state of the art power supplies and their implications can be deduced from Glatz et al. (2010a). The main issue is to map NC to the technology's constraints where a first step is taken in Chou et al. (2003). Often, the most beneficial approach is to use NC that applies opportunistic behaviour as the approach by Chen et al. (2007). Their approach is different from the approach presented here, where we outline a novel generic and scalable approach. It is a setup, such that the maximum gain that can be achieved with NC can clearly be identified. We stick to intra-session NC and will only consider inter-session aspects for examining delay constraints. In other words, different sessions' data will not be combined for energy conservation.

### *2.2 Issues with comparison completeness*

As discussed in Section 1, when providing a generic approach for optimising one metric, other metrics that might be important need to be considered as well. Apart from energy efficiency, wireless communication is always subject to deal with dependability issues. Lossy links and network dynamics demand for dealing with varying packet reception ratio (PRR) and changing topologies in mobile networks.

### *2.2.1 Mobile wireless technologies*

While WSNs are considered static most of the time this is different for a related field of research in MANETs as juxtaposed by Akyildiz et al. (2002). For MANETs where crossing flows of information might even be more dominating as it can be for WSNs when using multiple routes, the approach for scalable NC has to be analysed for its ability to adapt to changing environments. The most robust way of adapting to changing conditions is to locally reinitialise the approach. For NC, LINDONCS by Glatz and Weiss (2009) gives answers to questions regarding feasibility and from a quantitative point of view. For reasons of completeness, the approach presented by Glatz et al. (2009a) has its initialisation procedure accurately elaborated throughout this paper. While NC is applicable to MANETs as well, it might not be the number one issue for optimisation. Protocols that relax constraints of the *ad hoc* nature are not part of this paper's elaborations.

### *2.2.2 Dependable wireless networking by redundant schemes*

In case of a lossy channel where PRR may drop significantly, dependable WSN implementations are needed. For reasons of applicability, the NC approach must not interfere with optimisations concerned with dependability.

Avizienis et al. (2004) define dependability as an integrating concept that encompasses various attributes. The attributes that are important for our considerations are *availability* and *reliability*. A channel or a link is highly available if it has high PRR. Little energy overhead occurs because only few packets need to be retransmitted. Reliability means that transient errors and their energy overhead are accordingly tolerable. If the application cannot service requests as intended any more, it is not reliable any more.

WSNs are designed in the most cost-efficient way such that they are still reliable. A common means of doing so is using multiple paths for message transmission. Therefore, because there exists no self-contained evaluation of NC reliability yet to the best of the authors' knowledge, we discuss NC for multipath routing (MPR). MPR – or concurrent multipath routing (CMPR) (Lou et al., 2006) – tolerates more faults than standard routing techniques. For the simple case where two CMPR communication channels cross in a network, we arrive at a standard setting with multiple crossing where NC can be applied as previously elaborated by Glatz et al. (2009a).

## *2.3 Low power wireless sensing and communication systems*

Starting cross-layer considerations for WSNs at the hardware layer and its interaction with the environment, EHSs need to be taken into account. Energy harvesting as exemplified by Glatz et al. (2008) shows that even when using prominent approaches like duty cycling (Kansal et al., 2007) for Heliomote (Raghunathan et al., 2005) system robustness is an issue. All further EHS aspects that are considered throughout this paper assume state-of-the-art low power hardware and small-scale harvesting devices and energy storage as presented by Glatz et al. (2008).

MAC strategies are another very hardware-related issue. LPL by Moon et al. (2007) for standard TinyOS MAC or highly efficient UBMAC may introduce application specific performance constraints or introduce memory size and runtime overhead for implementing the strategies needed, where Raghunathan et al. (2006) gives an overview. Given these preconditions and for allowing for further integration of energy management upon scalable NC as introduced by Glatz et al. (2009b), the concept in this paper will be based on a slotted and static MAC design as fostered by Flury and Wattenhofer (2010) for novel designs as well. For further improving robustness of the system, the concept will utilise ACKs by means of overhearing along paths that are essentially needed for forwarding of the information flows themselves.

## 3 Conceptual modelling and analysis of NC

We generalise the well-known butterfly example with a single crossing to a network with $2 \times k$ unicasts crossing in a perpendicular fashion. This allows eliminating effects of neighbouring network regions with unknown and may be different NC or routing behaviour. This is not meant to model a single specific scenario. It aims at clearly showing what energy can be conserved if NC can be applied.

We assume a static slotted MAC protocol for ease of comparison and as discussed in Section 2.3. Varying different MAC approaches could arbitrarily bias the result without further helping in quantifying the gain of NC over routing. Each mote is assigned a single MAC slot per time slot. The ordering is such that the sequence of motes accessing the channel is inverse to the direction of the information flows for assuming worst case conditions.

For still capturing effects of state-of-the-art WSN MACs we annotate the sending to receiving cost ratio according to what is to be expected for the transmission of a single message when using LPL. These values are validated by using the measurement setup presented by Glatz et al. (2010a). LPL preamble length compared to effective data length and its overhead may influence what can be gained from NC as well as the sending power. An overview of related results can be found in Glatz et al. (2010b).

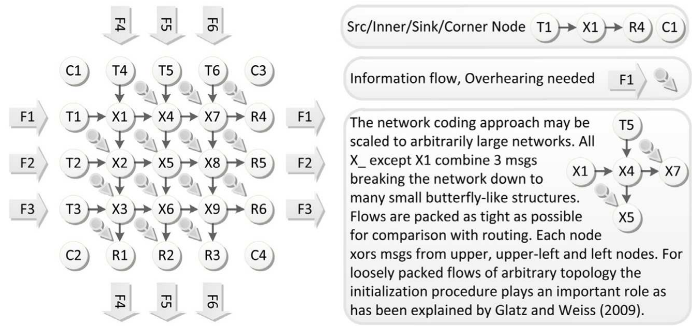
### *3.1 The butterfly structure*

The minimum butterfly structure can be seen as a $1 \times 1$ part of a mesh network. Consider information flows $\mathcal{F}_\text{A}$ injected at $X1$ and $\mathcal{F}_\text{B}$ at $T5$ in the subgraph on the right side of Figure 1. For successful delivery of two multicasts with two sinks at $X5$ and $X7$ routing will congest the channel at $X4$ where a unit rate of 1 is assumed for the MAC. Overhearing of messages transmitted from $X1$ and $T5$ allows applying wireless xor-coding as NC at $X4$. $X4$ conserves a single message. The question that comes up is whether this behaviour is robust and scalable or if special conditions need to apply.

Routing $\mathcal{F}_\text{A}: (X1) \rightarrow (X5, X7)$ and $\mathcal{F}_\text{B}: (T5) \rightarrow (X5, X7)$ along $X4$ will not only need one more message but also need one more time slot given the worst case setup. This reduces the maximum rate $r_{X1}$ and $r_{T5}$ for the source processes at $X1$ and $T5$. Now, for each edge $e \in \mathcal{E}$ in graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ let $o(e) \in \mathcal{N}$ be the edge source and $i(e) \in \mathcal{N}$ be the sink. Due to congestion the maximum source rate has to be reduced such that the set of incoming $\mathcal{I}(\nu)$ edges of a node (Equation (1)) does not exceed its outgoing bandwidth for edges in $\mathcal{O}(\nu)$ which is the set of outgoing edges.

$$\mathcal{I}(\nu) = \left\{ e_\upsilon \nu \in \mathcal{E} \forall \upsilon : \mathcal{O}(\upsilon) \cap \mathcal{I}(\nu) \neq \{\} \right\} \tag{1}$$

**Figure 1** Depicting the main concept of scalable NC for WSNs




As for wireless networks $\mathcal{O}(v)$ can be seen as a single edge for outgoing transmissions, we have to satisfy the condition in Equation (2).

$$\sum_{s \in \mathcal{N}:\ \mathcal{O}(s) \cap \mathcal{I}(t) \neq \{\}} r_{st} \leq \text{ Sending bandwidth } \forall t \in \mathcal{N} \tag{2}$$

In other words, the limitation of crossing routes feeds back to the information source. This can be avoided with NC where we view each message (and source process accordingly) as a vector of $|m|$ bits. It is represented in a Galois-Field GF(2). Each node adds up all the incoming messages for which it is a designated receiver and forwards the result. For the butterfly case and a given point in time $t$, we implement Equation (3).

$$m(\mathcal{O}(X1), t + \epsilon) = \mathcal{F}_{\text{AB}}(t) = \mathcal{F}_{\text{A}}(t) \oplus \mathcal{F}_{\text{B}}(t) \tag{3}$$

The length of the message $|m|$ does not grow, neither does the needed bandwidth, but there are two other aspects to be considered:

1 Which information is needed for packet decodability at the receiver?
2 How can delayed messages be handled given worst case MAC settings?

The first one – decoding the packets – shows up in Figure 1. Overhearing $\mathcal{F}_{\text{A}}$ at $X5$ allows decoding $\mathcal{F}_{\text{B}} = \mathcal{F}_{\text{A}} \oplus \mathcal{F}_{\text{A}} \oplus \mathcal{F}_{\text{B}}$.

The latter one – timing issues – demands for MAC considerations. Consider evaluation of communication links in order $e_{X1,X5}(t) \ldots e_{X4,X5}(t+1) \ldots e_{X1,X5}(t+2)$. Dependent on whether $t$ and $t+1$ or $t+1$ and $t+2$ are assigned to the same session, different information will be decoded at target node $X5$.

### *3.2 Multiple butterfly structures*

As a first step we discuss how information flows

1 $\mathcal{F}_{S1}$ that enters the network at $T1$
2 $\mathcal{F}_{S2}$ that enters the network at $T2$
3 $\mathcal{F}_{S(k+1)}$ i.e. sent along $T(k+1)$

cross perpendicular in Figure 1. In packets that contain information of more than two flows, information has to be reduced prior to entering areas where the flow of NC contradicts information available from overhearing. This is important for keeping the NC part embeddable in an otherwise routed or unknown network environment. In our setup, information flows from top to bottom and from left to right. So the direction of routing is pointing towards the lower right corner node. In the following, we look at symmetric star networks with $n$ nodes and $2(\sqrt{n}-2)$ sending and receiving entities accordingly.

### 3.2.1 *Arbitrary topologies and scalable network coding*

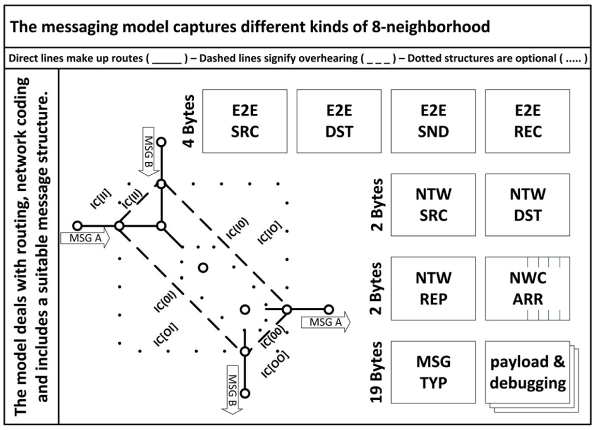
We now generalise to an arbitrary number of crossings in star structured networks. The setup and names are explained in Table 1 for quadratic networks as shown in the left side of Figure 1. The quadratic star structure case can be applied to other topologies as well given the WSN is dense enough such that overhearing that does not take place directly can be substituted by multihop transmission of motes in between. Consider the case depicted in the left of Figure 2 as a single crossing at any $X$ node of a tightly packed network. In case the network does not have its flows that narrow beneath each other, overhearing can be substituted by any IC[01] or IC[10]. The price one pays is that reducing the number of messages to be sent at crossings costs an additional number of messages to be sent for overhearing. So the approach will still work, but the gain will drop very fast for increasing number of hops in the substituted paths.

Now considering the case where IC[01] and IC[10] get closer similar to IC(01) and IC(10) and finally to a single non-disjoint path, this gives the very borderline case between NC and combining two routes from multiple routes. While still having zoomed into the network we now get a $Y$-like structure instead of an $X$-like one if $MSG\ A$ and $MSG\ B$ are decoded and forwarded in a disjoint manner again. Apart from that, a possible message structure is depicted on the right side of Figure 2 that allows full appliance of scalable NC with energy management, its flooding and initialisation and ACKs. There are four bytes for end-to-end information of a message providing a unique identifier using a session time stamp included in the message. Another four bytes are used for point-to-point transmission and finally 19 bytes are reserved for type information and the message payload.

**Table 1** Setup for an arbitrary number of crossings

| *Topology* | *Quadratic star topology* |
|---|---|
| Corner node | Node in the upper left of $T1$ and $T(k+1)$ |
| Receiver node | Node only connected to one node $Ri \in \mathcal{R}$ |
| $T1 \cdots Tk$ | Left-hand side source of information flow |
| $T(k+1) \cdots T2k$ | Upper source of information flow |
| $R1 \cdots Rk$ | Lower sink of information flow |
| $R(k+1) \cdots R2k$ | Right-hand side sink of information flow |
| $Xi$ | Repeater node inside the network |
| $n$ | Number of nodes |
| $k$ | Sources/sinks per side ($k = \sqrt{N} - 2$) |
| $m_{ij}(t)$ | Message along edge $e_{ij}$ |

**Figure 2** A message structure for applying scalable NC to WSNs of arbitrary topologies




*Note*: It indicates possible ways of how messages may be overheard.

For initialisation of the system, we will flood the network starting from the receiver side. In a real network, this allows building the same state machine into each mote and initialisation the network from receiver motes. During that phase the nodes are gathering neighbourhood information from flooding messages that are being forwarded and overheard. We let the initialisation phase last $n$ time units. This way, each node builds up a routing table to the $2k$ receiver motes. Afterwards, the motes start to sense and/or forward data.

### *3.2.2 Rules for combining messages*

We add $m$-length messages in GF(2). Generalisation to an arbitrary number of crossings is achieved by first combining and forwarding messages from left to right and top to bottom or vice versa until we end up in the lower right corner of the network. We code the receiver identification into each message. This way at the receiver side we will only combine messages from the left side, the upper left or above and keep other parts from further being forwarded. This perfectly reduces the content at each node to a maximum of two information flow's messages.

### *3.2.3 The gain over routing when using LPL*

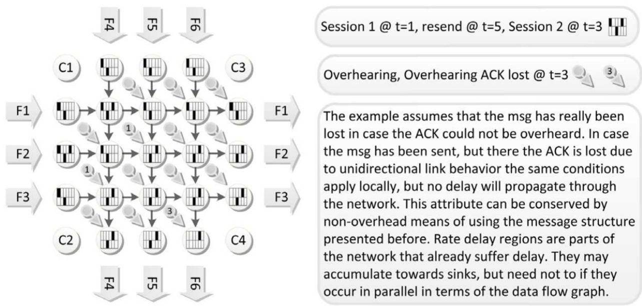
NC reduces the number of messages that have to be sent compared to the case when routing is used. Table 2 gives a summary. We see from Table 2 that the gain for large networks converges to 2.

For modelling with integer parameters we set the average receiver to listener cost ratio for LPL to 2. Due to technology constraints like initialisation times of the radio, the average sending energy over average reception energy needs can be assumed to be slightly above $\text{sc} = E(P_{\text{snd}}/P_{\text{rec}}) \approx 2$ for Mica2, but the ratio is reduced for longer messages and lower sending power. Nevertheless, translating the approach to other hardware architectures with possibly different power dissipation characteristics will demand evaluation of their ratio as well. For this paper it has been measured using the setup presented by Glatz et al. (2010a). The setup settings are summarised in Table 1.

**Table 2** Number of messages for an arbitrary number of crossings

| *Method* | *Routing per flow* | *Routing per net* | *NC per net* |
|---|---|---|---|
| No. of messages | $k+2$ | $2k(k+2)$ | $4k+kk$ |

**Figure 3** In case receivers' messages are not overheard to be forwarded they are resent the next time the channel is free




*Note*: Each odd time step can be used for sending in the example.

### *3.3 Resending policy for lost packets*

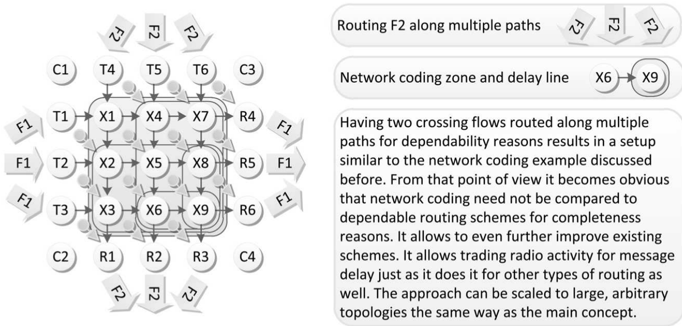
Packets that are not overheard and therefore cannot be further forwarded as depicted in Figure 3 are assumed to be lost. They will be resent up to five times as the channel is available in case no ACK has been received. Figure 3 shows that resending may introduce delay in case the channel is fully utilised. If high data rates and a lossy channel have to be combined with many hops per path it has to be taken into account to reserve enough space in memory for buffering the messages. Though, NC introduces more overhead than routing due to backpressure on two flows instead of one for routing, conditions are relaxed again as each coded message i.e. buffered also contains more information. This is why less motes will need to buffer messages. Furthermore, it must be noted that not each and every occurrence of a resend leads to accumulation of delay as it is also shown in Figure 3.

### *3.4 A hybrid scheme: combining MPR and NC*

Figure 4 depicts a sample scenario employing MPR: for each of the information flows $F1$ and $F2$ there are three redundant paths available. As only one path has to stay operational per flow, this scheme supports robustness and fault tolerance.

For the hybrid scheme, we define the so-called *network coding zone* (NCZ) which is defined by the area overlapped by two perpendicular information flows. As paths of the information flows cross each other per definition at the corners of the NCZ, any of the two original messages or the xored message can always be computed when needed. By crossing, we refer to two perpendicular paths crossing in one node, not an edge. NC is only used within the NCZ. Outside the NCZ, routing is employed. The exceptions are the right and lower edges of the NCZ, where original messages are forwarded instead of xored messages. This is necessary to compute the other original message at the lower right corner of the NCZ.

**Figure 4** MPR with three redundant flows for each routing




*Note*: Within the NCZ messages can be combined just as it can be done for any other type of routing. Paths that cross a delay line may take more than one time slot due to the worst case slotted MAC setup.

Using NC, the same networking optimisation and therefore energy conservation scheme can be employed as discussed before. Trade-offs that need to be considered here are reliability and delay. For reliability reasons, for both protocols – MPR and the hybrid scheme with NC – ACKs and retransmissions are implemented using the message structure shown in Table 1.

### *3.4.1 Improving dependability by using ACKs*

The dependability of either of the discussed approaches can be improved by using ACKs. An ACK informs the sender whether the last message sent was good or bad. A *bad message* is a message that either does not arrive within a certain time frame or that arrives damaged. Late arrival can be identified by setting a timeout and a damaged message can be detected by checking message IDs for MPR, by checking the neighbourhood history for NC.

ACKs are coded into the further forwarded messages' bodies by all entities of the NCZ. None of the entities is an end-to-end sender or receiver, for each combination of incoming messages, there will always be a follow-up message. This implies that there is always the opportunity for the previous sender of overhearing the next message containing the ACK. We only consider single hop ACKs. This relaxes conditions as no central knowledge or control is needed. The number of retransmissions is limited to five, which overcomes transient errors. The time that passes by while performing them allows time-varying channels and noise to relax again.

Apart from missing ACKs, the $NTW\,REP$ and $NWC\,ARR$ fields (of the message structure shown in Figure 2) can be used for energy management requests. They allow identifying time slots – where a 2 byte $NTW\,REP$ covers a sufficiently large number of activation periods of small-scale EHSs – and actions. $NWC\,ARR$ allows pointing to IC[00] through IC(11) and to set a flag for energy request, energy supply, for NC (in-)activation or a notification that the actual duty cycle cannot be sustained any more. This allows implementing the energy management scheme presented by Glatz et al. (2009b) on top of it.

## 4 Implementing scalable network coding for modelling in Matlab, simulation in TOSSIM and measurement of motes

We describe a grid setup initialisation using flooding and routing as well as NC. For simplicity reasons, the pseudo code shows the Matlab model implementation i.e. being used. In Section 5, results are shown from Matlab model evaluation, simulation runs in TOSSIM and measurement results with TOSPIE2.

**Algorithm 1** Network initialisation

```
Require: FLOODING(ReceiverMotes)
Require: adj_matrix
Require: rt_d_via ← inf
Ensure: RoutingTableBuilt ≡ TRUE ∀ t ≥ n
function FLOOD                                          ▷ Matlab Notation
    for m ≡ 1...N do
        d ← ZEROS(N, N)
        for dd ≡ 1...N do
            for t ≡ 1...N do
                d(dd, t) ← MIN(rt_d_via(dd, :, t))
            end for
        end for
        for adj ≡ FIND(adj_matrix(m, :) ≥ 0, N) do
            for t ≡ 1...N do
                rt_d_via(adj, m, t) ← d(m, t) + 1
            end for
        end for
    end for
end function
```

**Algorithm 2** Network routing

```
Require: FLOOD ≡ finished                               ▷ t ≥ n
Require: msgs(t) ← sensing(t)
Ensure: SentMessagesReceived ≡ TRUE ∀ messages
function ROUTING                                        ▷ Matlab Notation
    for mote ≡ mac_order do
        for rcvr ≡ 1...n do                             ▷ if(bwidth(mote))
            dist, ngb ← MIN(rt_d_via(mote, :, rcvr))
            msgs(ngb, rcvr) ← msgs(mote, rcvr)
            msgs(mote, rcvr) ← inf
            REMOVEENERGY(mote)
        end for
    end for
end function
```

Algorithm 1 shows the initialisation phase. It allows running both – routing and NC – afterwards. Variable rt_d_via(mote, via, target) contains the hop distance from mote to target along via. The initialisation starts with flooding from the receivers and ends up with a complete routing table for any connected topology at each node that leads to the receivers.

The routing implementation chooses the minimum hop count and transfers messages to another place in array msgs indicating it pending at the new mote and clearing it at the old one with inf in Algorithm 2.

**Algorithm 3** Network coding

```
Require: FLOOD ≡ finished                                        ▷ t ≥ n
Require: msgs(t) ← sensing(t)
Ensure: SentMessagesReceived ≡ TRUE ∀ messages
function CODING                                                  ▷ Matlab Notation
    for mote ≡ mac_order do                                      ▷ isStdTx(mote)
        for rcvr ≡ 1 . . . n + 1 do                              ▷ if(bwidth(mote))
            p, rec ← FIND(msgs(mote, :) < inf, n + 1)
            p ← msgs(mote, rec)
            if LENGTH(p) ≡ 3 then
                packet ← GF2ADD(p(1), p(2))
                packet ← GF2ADD(packet, p(3))
                msgs(mote + 1, rec(1)) ← packet
                msgs(mote + √n, rec(2)) ← packet
                msgs(mote + √n + 1, rec(3)) ← packet
                msgs(mote, :) ← inf
                REMOVEENERGY(mote)
            end if
        end for
    end for
end function
function GF2ADD(i1,i2)                                           ▷ Matlab Notation
    mpow ← CEIL(log_2(i1, +i2))                                  ▷ Truncate
    for pow ≡ mpow ↓ 0 do          ▷ The power of two is a shift: little comp. load
        b ← 2^pow                  ▷ Dividing by a power of two: little comp. load
        v1 ← FLOOR(i1/b)
        v2 ← FLOOR(i2/b)
        i1 ← i1 − v1
        i2 ← i2 − v2
        o ← XOR(v1, v2), o
    end for
    for i ≡ 1 ↑ LENGTH(o) do
        GF2ADD ← GF2ADD + o(i) · 2^(i−1)
    end for
end function
```

The NC implementation has to implement crossing flows as depicted in the previous section. We show the case of a standard transmitter (motes deduce their role from messages received in the initialisation phase) that activates if there are three messages pending. One can see how the logical receiver of physical broadcasts has to be set to not contradict the routing direction on the one side and to always reduce the number of flows from three to two for any mote on the other side. The second function in Algorithm 3 shows the simulation implementation and it becomes obvious that all computations can easily be broken down to be performed with shifting and adding and therefore only little overhead in code size and computational load. The case of the standard transmitter can be extended to all the other nodes such that the implementation stays the same. The only difference is that some lines are to be removed if fewer packets are to be awaited at a given position in the network. As an example, consider the upper left transmitter: it will get packets from the upper left and the leftmost top source nodes, but the upper left corner node will not send any messages in our setup. The modelling setup has been designed such that it can be easily translated to a TinyOS implementation using low complexity state machines based on mote identifier alone after initially determined network positions. Similarly, the TinyOS implementation is such that it can be tested in a hardware testbed such that it can be profiled on real hardware in principle. Nevertheless, it will turn out that the overall problem's complexity does not allow accurate large-scale investigation at runtime on hardware. Therefore, we limit the discussion of experimental setups to Matlab pseudo code. It can be explained in a more compact way. Nevertheless, results will also examine TOSSIM simulation results and experiments with Mica2 hardware measurement setups.

## 5 Evaluation and results

The ratio of sending cost to receiving cost values for sending short messages with maximum sending power on Mica2 motes results in a ratio close to 2 if using LPL as it can be seen in Table 3. For technologies with a higher ratio, NC has even larger benefits than what is shown by the results presented here.

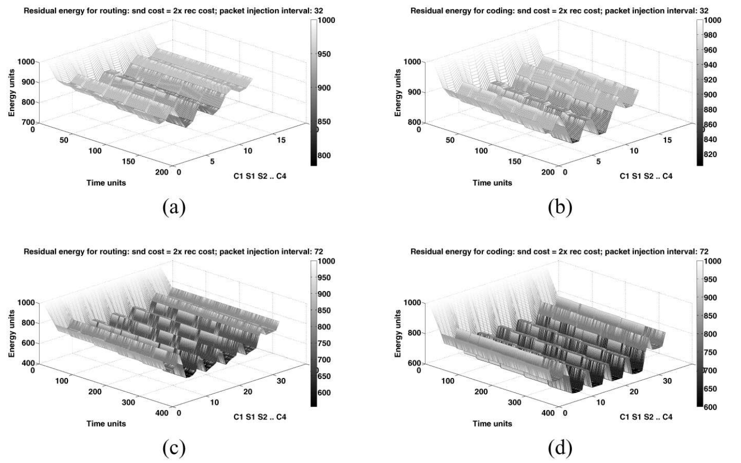
In Figure 5, the residual energy is plotted overtime for each mote when modelling NC compared to routing. Motes are numbered top–down and from left to right: upper left corner, left source nodes and lower left corner, the leftmost upper source node, leftmost inner nodes, leftmost lower receiver until the lower right corner node. The highest residual energy can be observed at the corner nodes as they are not part of the crossing information flows. For NC, all sending and receiving nodes end up approximately at the same energy level for NC and routing. The gain in energy conservation can best be seen from inner transmitter nodes. Furthermore, the bottom of the residual energy depression is flat for NC while it varies for routing. This implicit load balancing is a nice feature for WSNs that prolongs network lifetime as well. Table 4 gives a summary on NC energy conservation.

While all messages can be transmitted in all cases when using modelling alone without applying noise to the transmission channel, things get more complicated when doing simulations of power-annotated TinyOS implementations. Figure 6 shows plots of differently complete flooding procedures that are used for initially setting up the routing tables. There are several lessons to be learned from that. First, a lot of messages can be seen impacting the motes' energy reservoirs for flooding – still, it results in similar energy reservoir depression among motes as does modelling. While it helps in validating the modelling experience it is far too complex to use the simulation approach for complete, generic and scalable characterisation. Next, subplots in Figure 7 include energy input from EHDs and power dissipation for persistent memory write operations as well. Finally, injecting two sessions for only four motes lets the information i.e. provided by the plots explode. The main result is that allowing up to five resends per message was sufficient to finally get all messages delivered using meyer-heavy.txt as provided with TinyOS for all simulation runs. However, the number of resends reaches the same order as the number of messages that can be conserved using the NC optimisation at hand. In other words, applying a scheme for improving robustness to reduced PRR is worth implementing as could also be validated for hardware experiments of size $n = 16$.

**Table 3** Measuring the sending to receiving cost ratio with a NI PXI 6221 DAQ

| *Type* | *25% duty cycle* | *10% duty cycle* | *5% duty cycle* |
|---|---|---|---|
| $\frac{P\_avg(snd)}{P\_avg(rec)}$ | 2.1191 | 2.2709 | 2.3028 |

**Figure 5** Subplots show the energy consumption of routing and NC for differently sized networks: (a) 16 nodes routing; (b) 16 nodes NC; (c) 36 nodes routing and (d) 36 nodes NC



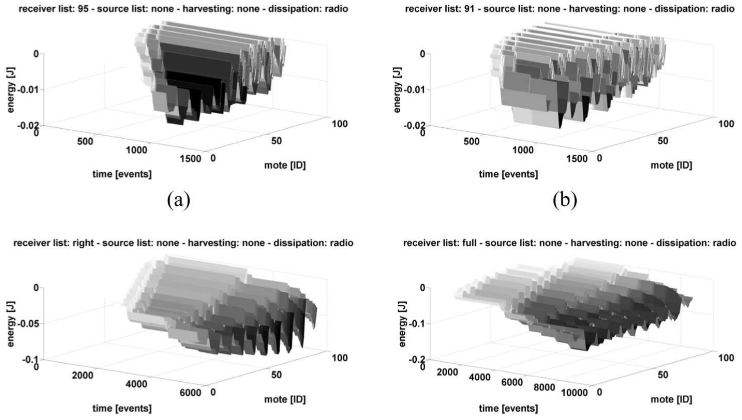

*Note*: Initial flooding and four rounds of packet injection are shown. Inner nodes have steeper residual energy drops, but increasing the network size does not further impact energy loss. The lower residual energy depression will not always be flat for routing due to multiple minimum-hop paths. There exists only one path for all transmissions if NC is used. Therefore, the load among inner NC transmitters is inherently balanced by that approach.

**Table 4** Transmitter savings in Figure 5 for data delivery from NC ($N$) over routing ($R$) in terms of energy loss ($L$) and variance ($V$)

| *R16L* | *R36L* | *N16L* | *N36L* | *R16V* | *R36V* | *N16V* | *N36V* |
|---|---|---|---|---|---|---|---|
| 840→784 | 640→556 | 840→804 | 640→600 | 20 | 56 | 0 | 4 |

A benefit of the hybrid scheme compared to MPR is that fewer messages have to be sent, especially when employing ACKs. This is because when using NC, every node sends only one message to multiple addressees. With MPR, every node has to send one message per path the node is part of. For a mesh topology like in Figure 4, nodes in the NCZ have to send twice as many messages when using MPR compared to the hybrid scheme. That means the more redundant paths are available the larger the NCZ becomes and, consequently, the more messages can be saved.

**Figure 6** Subplots show the energy consumption of flooding for different sink configurations: (a) receiver 95; (b) receiver 91; (c) receivers right and (d) receivers all

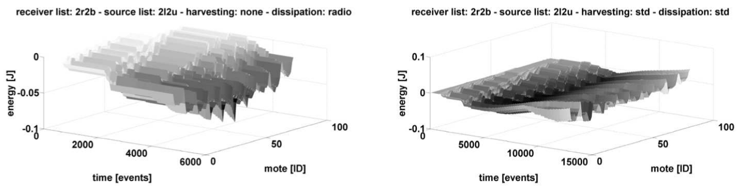



(a) (b)

(c) (d)

*Note*: Each node i.e. a designated sink sends out a flooding broadcast. The nodes' energy reservoirs are impacted by sending and listening costs. Results have been profiled from a fully functional TinyOS implementation. Firstly, it can be seen that the energy trend fits modelling results for routing quite accurately. Secondly, it is hard to exactly track down the actual differences among different cases from visual inspection.

**Figure 7** Subplots show the energy consumption of flooding and sending two sessions of four flows through the network with including up to five resends if ACKs are missed: (a) receiver 212u and (b) 212u full energy




(a) (b)

*Note*: The second subplot adds full energy modelling support. Extending over considerations in the last figure, a small scale EHD is assumed as power supply and power dissipation of writing to persistent storage is considered as well. This leads to more events being reported by TOSSIM which scales the timeline of the second subplot.

The earliest time of delivery when using MPR is equal to the number of hops through the NCZ. The latest time of delivery is twice that delay. For NC, the same delay plus a delay equal to the number of hops applies. Applying noise to the simulation setup introduces delay for both MPR and NC. Finally, it turns out that the scalable NC scheme can be run with MPR as well. While the same delay is to be expected, the reliability constraint can be relaxed with straight forward single-hop ACKs.

There is no perfect approach as a scheme's suitability depends on the optimisation criteria of the application, but enhancing scalability, power awareness and reliability is usually a trade-off. The scalable NC scheme turns out to conserve energy and to be applicable to reliable MPR networking systems as well. So, it allows optimising the most prominent cost metrics of WSN at the same time.

## 6 Conclusion

We have shown that a gain of up to two can be achieved with NC and that the gain can be mapped to prolonged network lifetime under technology constraints of WSNs. We have set up a simplified network model in simulation to better understand what can be achieved and give a completely localised and scalable implementation that can be easily implemented on motes of small form factor.

Furthermore, the concept has been refined to work with arbitrary topologies. Its use with MPR has been elaborated as well. It turns out that the decentralised and scalable NC approach all applicable to be built into reliable systems as well.

For future work, the networking middleware may be used to support power awareness for very low power systems. It is suitable to be integrated into a middleware implementing energy management schemes and will help to implement energy conservation with adaptation to the efficiency of EHSs. Furthermore, its combination with local and automated recognition of structures in a network that are applicable to NC techniques may allow to optimise messaging in higher throughput MANETs as well.

## References

Ahlswede, R., Cai, N., Li, S-Y. and Yeung, R. (2000) 'Network information flow', *IEEE Transactions on Information Theory*, Vol. 46, pp.1204–1216.

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'Wireless sensor networks: a survey', *Computer Networks*, Vol. 38, pp.393–422.

Avizienis, A., Laprie, J.C., Randell, B. and Landwehr, C. (2004) 'Basic concepts and taxonomy of dependable and secure computing', *IEEE Transactions in Dependable and Secure Computing*, Vol. 1, pp.11–33.

Chen, W., Letaief, K. and Cao, Z. (2007) 'Opportunistic network coding for wireless networks', *Proceedings of the IEEE International Conference on Communications*, Glasgow, Scotland, pp.4634–4639.

Chou, P., Wu, Y. and Jain, K. (2003) 'Practical network coding', *Proceedings of the 51st Allerton Conference on Communications, Control and Computing*, Monticello, USA.

Flury, R. and Wattenhofer, R. (2010) 'Slotted programming for sensor networks', *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Stockholm, Schweden, pp.24–35.

Glatz, P.M., Meyer, P., Janek, A., Trathnigg, T., Steger, C. and Weiss, R. (2008) 'A measurement platform for energy harvesting and software characterization in WSNs', *Proceedings of the 1st IFIP/IEEE Wireless Days*, Dubai, UAE, pp.1–5.

Glatz, P.M. and Weiss, R. (2009) 'LINDONCS: localized in-network detection of network coding structures in wireless sensor networks', *Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, Tenerife, Canary Islands, Spain, pp.17–24.

Glatz, P.M., Hein, K.B. and Weiss, R. (2009a) 'Energy conservation with network coding for wireless sensor networks with multiple crossed information flows', *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, Kaohsiung, Taiwan, pp.202–207.

Glatz, P.M., Loinig, J., Steger, C. and Weiss, R. (2009b) 'A first step towards energy management for network coding in wireless sensor networks', *Proceedings of the 9th Malaysia International Conference on Communications*, Kualua Lumpur, Malaysia.

Glatz, P.M., Hörmann, L.B., Steger, C. and Weiss, R. (2010a) 'A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency', *Proceedings of the 6th International Workshop on Sensor Networks and Systems for Pervasive Computing*, Mannheim, Germany.

Glatz, P.M., Steger, C. and Weiss, R. (2010b) 'Poster abstract: TOSPIE2: tiny operating system plug-in for energy estimation', *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Stockholm, Schweden, pp.410–411.

Hill, J. and Culler, D. (2001) 'A wireless embedded sensor architecture for system-level optimization', Technical Report *UCLA*.

Ho, T. and Lun, D. (2008) *Network Coding: An Introduction*. Cambridge University Press.

Jacobsen, R., Jakobsen, K., Ingtoft, P., Madsen, T. and Fitzek, F. (2008) 'Practical evaluation of partial network coding in wireless sensor networks', *Proceedings of the 4th ICTS/ACM International Mobile Multimedia Communications Conference*, Oulu, Finland.

Jain, K., Mahdian, M. and Salavatipour, M.R. (2003) 'Packing steiner trees', *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, Baltimore, Maryland, USA, pp.266–274.

Kansal, A., Hsu, J., Zahedi, S. and Srivastava, M. (2007) 'Power management in energy harvesting sensor networks', *ACM Transactions on Embedded Computing Systems*, Vol. 6, No. 4, p.38. DOI = 10.1145/1274858.1274870. Available at: http://doi.acm.org/10.1145/1274858.1274870.

Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M. and Crowcroft, J. (2006) 'Xors in the air: practical wireless network coding', *Proceedings of ACM SIGCOMM*, Pisa, Italy, pp.243–254.

Lou, W., Liu, W. and Zhang, Y. (2006) 'Performance optimization using multipath routing in mobile ad hoc and wireless sensor networks', *Combinatorial Optimization in Communication Networks*, Vol. 18, pp.117–146.

Mottla, L. and Picco, G.P. (2010) 'Tutorial: programming wireless sensor networks beyond operating systems: from theory to practice', *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Stockholm, Schweden.

Moon, S., Kim, T. and Cha, H. (2007) 'Enabling low power listening on ieee 802.15.4-based sensor nodes', *Proceedings of the IEEE Wireless Communication and Networking Conference*, Hong Kong, pp.2305–2310.

Polastre, J., Szewczyk, R. and Culler, D. (2005) 'Telos: enabling ultra-low power wireless research', *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, Los Angeles, California, USA.

Raghunathan, V., Ganeriwal, S. and Srivastava, M. (2006) 'Emerging techniques for long lived wireless sensor networks', *IEEE Communications Magazine*, Mag. 44, No. 4, pp.108–114.

Raghunathan, V., Kansal, A., Hsu, J., Friedmann, J. and Srivastava, M. (2005) 'Design considerations for solar energy harvesting wireless embedded systems', *Proceedings of the 5th ACM/IEEE International Symposium on Information Processing in Sensor Networks*, Los Angeles, California, USA, pp.457–462.

Warneke, B., Last, M., Liebowitz, B. and Pister, K.S. (2001) 'Smart dust: communicating with a cubic-millimeter computer', *Computer*, Vol. 34, No. 1, pp.44–51.

Wu, Y., Chou, P. and Kung, S-Y. (2005) 'Minimum-energy multicast in mobile ad hoc networks using network coding', *IEEE Transactions on Communications*, Vol. 53, pp.1906–1918.

Yick, J., Mukherjee, B. and Ghosal, D. (2008) 'Wireless sensor network survey', *Computer Networks*, Vol. 52, No. 12, pp.2292–2330.

# A Measurement Platform for Energy Harvesting and Software Characterization in WSNs

Philipp M. Glatz*, Philipp Meyer†, Alex Janek‡, Thomas Trathnigg§, Christian Steger¶ and Reinhold Weiss‖
Institute for Technical Informatics, Graz University of Technology, Inffeldgasse 16/1, 8010 Graz, Austria
*Philipp.Glatz@TUGraz.at, †Philipp.Tirol@GMail.com, ‡Alex.Janek@TUGraz.at,
§Thomas.Trathnigg@TUGraz.at, ¶Steger@TUGraz.at, ‖RWeiss@TUGraz.at

***Abstract*—Wireless Sensor Network (WSN) nodes are resource-constrained computing devices. Adaptive behavior of autonomously working WSNs tries to maximize the cost efficiency of deployments. This includes maximizing the lifetime through power consumption optimization and recharging energy reservoirs with the use of energy harvesting.**

**The adaptive behavior that leads to efficient resource usage needs information about the WSNs energy balance for decision making. We present a novel platform to measure the harvested, stored and dissipated energy. For being applicable to different environments it allows to attach different energy harvesting devices (EHDs). EHDs do not provide power continuously. Power availability patterns are used to determine how these sources can be used efficiently. Models from harvesting theory try to adapt to it.**

**We implement a model that targets energy neutrality on our platform. It is used to evaluate the model and improve it. Our novel platform can be used to evaluate theories that model different sources. It can utilize and characterize thermoelectric, piezoelectric and magnetic induction generators and solar cells. The measurement platform tracks energy dissipation too. Mote software is implemented to establish communication to the platform. A sample application on top of it shows that the system can be used for software characterization.**

**This paper contributes a novel modular and low-power design for measurement platforms for WSNs. It shows utilization of different energy sources and the ability to supply different mote types. Our work shows how theories for energy harvesting can be evaluated and improved. Our work also contributes to the field of simulation and emulation through online software characterization. The approach improves in accuracy and completeness over the capabilities of offline simulation.**

## I. INTRODUCTION

WSNs have experienced a lot of improvements over the years. The field of WSNs [1] has established as a research area. Starting from the vision of smart dust [2] many developments have been made to achieve different goals. A main issue are resource constraints due to the limited form factor. Furthermore, WSNs have to be capable of ad-hoc communication. Measures in software (SW) and in hardware (HW) deal with these issues.

In SW the implementation of network stacks demands for a layered design on the one hand and battery economy may raise the need for cross-layer design on the other hand. Another issue might be to allow motes to move in a mobile ad-hoc network. Next, the network has to be adaptive to best cope with environmental conditions. Recent developments in operating systems (e.g. TinyOS 2) for WSNs offer power management capabilities. Progress in HW has led to more memory, faster processing units and more effective radio. Especially the processing unit and the radio component need orders of magnitudes less power to be operated.

Different directions are being looked at in active research. They are concerned with intelligent power management strategies in SW and with reduction of energy dissipation in HW. Despite some progress in the field there are still problems that demand for further improvement of existing solutions. There are three different areas that are directly related to the power management problem in WSNs:

*a) Simulation:* A framework simulates the execution of the SW that runs on a mote and integrates the power consumption over different states of these exemplary runs. This way it is possible to give lifetime estimation dependent on the available battery level. Recent developments like PAWiS [3] also try to simulate power consumption using OMNeT++ with additional modelling of environmental influences.

*b) Batteries:* Another approach to enable long lived WSN nodes of small form factor would be to offer better power supplies to motes. Unfortunately, despite great success in development of electronic circuitry the power supplies for electronics evolve only slowly. Shortcomings in power supplies with batteries can partly be overcome with energy harvesting.

*c) Energy Harvesting:* Different types [4] of ambient energy can be used with energy harvesting. The challenges that energy harvesting system (EHS) design faces are twofold. First the available energy has to be consumed at a suitable power level to transfer it efficiently into a storage structure. This is done with maximum power point tracking (MPPT) with the usage of conditioning circuitry. Next a suitable energy model (e.g. energy neutrality in [5]) has to be implemented in SW and an output structure for the system has to be provided by the EHS HW. The components that are needed for an EHS are depicted in Fig. 1.

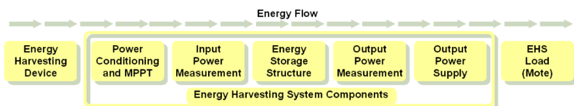



Fig. 1. Generic EHS Structure

Exemplary projects that target EHS or similar designs are

978-1-4244-2829-8/08/$25.00 ©2008 IEEE

grouped into different categories in section II. Improvements of our work over these solutions will be discussed.
We target several different issues of EHS for WSNs in this work where we could not find existing work that incorporates these in a single platform while some features have not been considered or implemented in other work at all:

- We allow for attaching multiple EHDs at the same time. They may be of different types: thermoelectric, piezoelectric and magnetic induction generators and solar cells.
- Through the use of a modular design, different HW components can be exchanged easily. Different mote types can be supplied by the system.
- The use of super capacitors allows for long lived platforms with at least up to 500000 recharge cycles [6].
- Exact measurement of input, stored, and output energy gives a detailed basis for modelling. The power supply and communication interface for attaching the EHS to a WSN mote makes runtime measurement with a high resolution possible.
- A sample evaluation of a model on energy neutrality shows how the prototype can be used to evaluate power management theories and help to improve them. Driver and SW interfaces have been implemented in TinyOS 1 and 2. Applications on top of it can sense and react using the EHS with an ease of programming similar to communication with sensor boards.

## II. Energy Harvesting Methodology Revised

Research projects that deal with energy harvesting are manifold. On the one hand they focus on the implementation of suitable HW to perform energy harvesting. On the other hand harvesting aware power management algorithms are developed and implemented in SW. In the following both aspects will be reviewed briefly.

### A. Energy Harvesting Systems

The spectrum of implemented EHS ranges from systems without any energy storage [7] over very simple systems with an energy storage [8] to simulation verification frameworks [9] and advanced EHS:
*Heliomote* [10] is a plug-in for Mica2 nodes. It is powered by solar panels and 2 AA-type NiMH batteries. A special battery monitoring IC is used to measure the energy balance. The stored energy is determined by the clamp voltage of the batteries.
The approaches in *Everlast* [11], and *AmbiMax* [12] store energy that is harvested by solar cells into double layer capacitors (DLCs). The voltage across the capacitor is used to acquire the energy balance.
*Trio* [13] uses the combination of DLCs and rechargeable lithium batteries to perform long and short term solar energy storage. The energy balance is acquired by the clamp voltages of the energy storage elements.
*AmbiMax* and *Everlast* perform MPPT in order to maximize efficiency. This load matching is done with (*Everlast*) and without (*AmbiMax*) CPU intervention.

### B. Harvesting Aware Power Management

Since environmental energy sources are subject to great fluctuations, it is necessary to acquire the exact behavior of the source and the consumer to perform harvesting aware power management.
The consumer is characterized offline. Before the system is deployed the consumer's powering needs are estimated. For the energy source and the energy flow to the consumer it is a similar approach. A model is stated that is expressive enough to capture possible environmental conditions (ambient energy) and execution paths (WSN workload).
Possible ways of source and consumer analysis will be considered in the following.

*1) Energy Consumption Modelling:* A WSN application's powering needs can be thought of as the sum of all its active components needs. All components (e.g. processor, radio, sensors) can be assigned a power value which is needed to drive a given system operation level. Different projects try to model WSNs that way with different level of detail and completeness. Prominent candidates are outlined below.
*PowerTOSSIM* [14] keeps track of a mote's peripherals activity. For a time span in which the combination of activity of different components did not change one can assign a power value. The length of the time span is calculated from a CPU cycle estimation. This allows calculating the energy dissipation.
*PAWiS* [3] is a simulation environment that is based upon the OMNeT++ network simulator. It increases in level of detail and completeness over approaches likes PowerTOSSIM [14] and aims at easing WSN SW development. The programmer is supported with information that also takes inter-node communication and environmental conditions into account.
Contrary to the two approaches given above *SPOT* [15] provides a platform that allows for measurement of power consumption patterns. Attaching *SPOT* to a mote improves in completeness over simulation based approaches. It even incorporates variations in HW among different samples of the same mote type.
In [16] a similar approach is taken. A low-power platform is being developed. This setup can additionally deal with variable supply voltage and uses a different method for measurement. This takes the approach one step further towards unattended usage of a measurement platform at runtime.

*2) Energy Source and Flow Modelling:* Complex analytical models are not viable to be implemented on wireless sensor nodes. Therefore, rules for the behavior of WSN nodes (e.g. duty cycles) are modelled offline to fit a given model. Then the rules are implemented and they are operated on at runtime. This is much cheaper in terms of needed computational power and code size. Evaluation of a model during runtime to do power management is more costly.
[17] proposes a simple analytical model for abstracting real environmental sources and consumers. The paper introduces

the parameters $\rho$ and $\sigma$ to describe the behavior of the energy source and the consumer. Energy neutrality can be achieved if the average harvested power $\rho_1$ is greater or equal the average consumed power $\rho_2$:

$$\rho_1 \geq \rho_2 \qquad (1)$$

Furthermore, the system must be able to store the energy of

$$\sigma_1 + \sigma_2 + \sigma_3. \qquad (2)$$

These parameters are the fluctuations of the energy source below and above the average harvested power and the fluctuation of the average consumed power above its average.
In order to calculate the exact parameters $\rho$ and $\sigma$ the source respectively the consumer must be known from $t = 0$ to $\infty$. In most cases, neither this knowledge is necessary nor can it be obtained. Solar energy has a certain daily respectively annual repetition. Wind shows returning patterns too. For this reason the parameters are determined within a certain modelling period, which is chosen according to the expected repetition. [18] introduces a framework that is able to determine the modelling period itself and does a prediction of the energy harvested in future.

## III. Energy Harvesting Prototype

This section presents the design of the EHS HW prototype, the EHS power management algorithm and the mote application programming interface (API).

### A. System Overview

This section presents the implemented EHS and the power management algorithm.
The main objectives for the EHS are to be modular, low-power and suitable for different harvesting devices. We built a prototype that can handle piezoelectric and thermoelectric generators as well as solar cells and magnetic induction generators. The system can be attached to a Mica2 mote and could be extended to supply other types without modification in HW as well. The main system components are shown in Fig. 2.

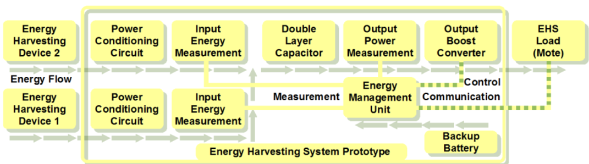



Fig. 2. Block diagram of the implemented EHS.

The system is capable to utilize the energy input from two EHDs separately. The harvested energy is stored into a DLC. The dissipated energy is measured with a separate circuit. A boost converter generates an appropriate voltage for the Mica2 sensor node. It can drive other types of sensor nodes too. They are allowed to operate at a supply level up to that of the Mica2. The energy management unit (implemented based on a $\mu$C i.e. MSP430 [19]) acquires the energy balance of the system, determines the formerly discussed model parameters $\rho$ and $\sigma$ and allows the communication with the wireless sensor node over $I^2$C-Bus. In order to save the energy required for the standby mode of the Mica2 it is switched off in periods when no operation in the wireless sensor network is needed. For this reason the energy management unit controls the duty cycle of the wireless sensor node as well.

*a) Input Energy Measurement and Power conditioning:* The use of commercial off-the-shelf energy measurement solutions is not found to be reasonable because of their relatively high current consumption compared to low-power generators like the piezoelectric generator. The power conditioning and energy measurement functions (shown in Fig. 2) are combined and implemented as a charge pump. The schematic representation is shown in Fig. 3. The charge pump starts in state 1. The EHDs charges the buffer capacitor $C_{buffer}$ and the transfer capacitors $C_1$ and $C_2$ to a certain voltage. At this threshold voltage the transfer capacitors are connected in series to the DLC (state 2). In succession they are discharged until a certain voltage level is reached and the operation starts from the beginning. The harvested energy can be determined by counting the number of operations (which transfer $14\,\mu J$ each). The main advantage of this setup is its low-power consumption of about $20\,\mu W$.

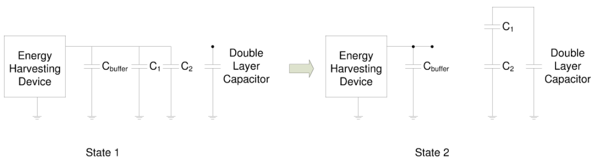



Fig. 3. Input energy measurement by a self-clocked charge pump.

*b) Stored Energy measurement:* The state of charge of a capacitor can be determined by its clamp voltage if the capacity is known. The capacity of the DLC depends on its electrode potential. The dependence of C($U_{cap}$) of the used 10 F DLCs from [6] is found to be nearly linearly dependent on the clamp voltage. This model is considered when calculating the stored energy.

*c) Output Energy Measurement:* The output current is measured by the voltage drop over a high side shunt resistor. This voltage is amplified and converted to a proportional current by a high side current amplifier (INA138, [19]). The amplifier output current is used to charge a capacitor to a certain threshold voltage. At this voltage it gets discharged and operation starts from the beginning. Multiplying the number of pulses with DLC voltage results in the dissipated energy. It can be measured with a high resolution of $0.281\,mA$ steps.

### B. EHS Software

The low-power constraints for the HW development apply to the firmware development as well. *Hurry up and sleep* is a key phrase to describe the general outline of the SW. After basic initialization of the $\mu$C it enters a very low-power

operation state and services all requests and measurement jobs by interrupt service routines.
The EHS firmware is responsible for implementing rules that are derived from the energy model in such a way that energy neutrality is preserved. This means that $\sigma$ and $\rho$ values are calculated to predict expected energy harvesting input. A suitable duty cycle to work under that condition is then suggested to the mote.
For building the model the $\mu$C counts the pulses generated by the input and output energy measurement circuits to determine the harvested and the dissipated energy. It acquires the clamp voltage of the storage capacitor to determine the amount of the stored energy using the capacity-voltage model.
Negotiation on the duty cycle between the Mica2 sensor node and the EHS is done via an $I^2C$ implementation (both $\mu$Cs offer $I^2C$ support). In succession the wireless sensor node can decide on the duty cycle respectively the time at which it will be woken up again. Coming to a decision at the mote rather than in the EHS allows for network level power management. To allow cooperation with a wireless sensor application driver for TinyOS 1 and 2 were developed. The commands `GetDutyCycle` and `SetDutyCycle` offer the basic interface between the EHS and the wireless sensor network operation. Subsection III-C will further describe the exemplary SW framework. It provides the Mote API.

### *C. Mote API*

Development of CPU-based measurement platforms and development of SW from scratch is a tedious task. Therefore we present a thin SW abstraction layer that allows controlling the EHS from the mote application layer. Two functions `GetDutyCycle` and `SetDutyCycle` have already been mentioned. Further self-explanatory functions are `StartWakeupCount` and `MotePowerOff`. These have been used for prototype evaluation within TinyOs 1. To keep up with current developments an API is implemented in TinyOS 2.
All communication is done via $I^2C$ register read/write with the mote in master mode. The API offers interfaces to upper SW layers that are as follows.
The API provides two interfaces. First `SplitControl` is a well known TinyOS 2 interface. Second the `EhsData` interface provides two sets of commands. On the one hand there are functions to read and write $I^2C$ registers and to loop them through to a data sink or host computer. On the other hand there are commands for powering on and off or requesting and freeing shared mote resources. This set of function allows a programmer to apply power conserving techniques on the SW that runs the prototype communication itself.
The used interface `Params2D` has to be implemented by SW components if they wish to have their power consumption characterized at runtime. This allows to reconfigure and test the powering needs of a SW configuration. Such a service can be used to provide information for network level power management.

## IV. Results

Since solar cells were used for the evaluation of the EHS a modelling period of 24 hours was chosen. The system acquires energy management information every 7.5 minutes and after 24 hours it calculates the energy balance modelling parameters.
To test the model checking capability and overall function of the EHS and the TinyOs drivers, a TinyOs application was implemented, that operates the wireless sensor node at the suggested duty cycle. The achieved result is shown in Fig. 4.

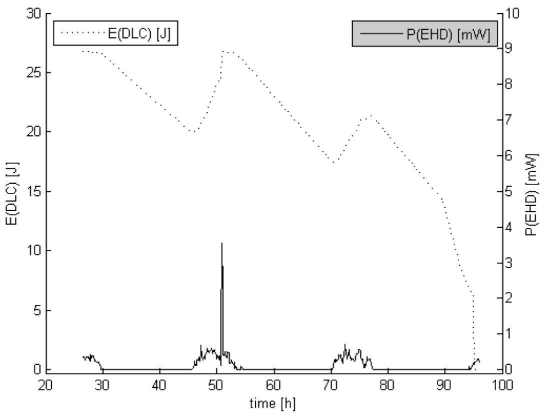



Fig. 4. Energy neutrality model evaluation over a 4 days time period.

It shows the harvested and the stored energy in the DLC during a 4 day evaluation period. As it can be seen, the system "died" after 4 days of operation. The reason for this behavior was that the system calculates the suggested duty cycle from the elapsed modelling period and assumes the repetition of the energy pattern in the future. A slightly counterintuitive effect occurs. The spike after 50 hours kills the system later on, because the model assumes the same spike in the next model period again. This effect is amplified by the small energy reserve in the storage capacitor whereby small variations of the daily energy pattern have a significant effect. The harvesting powered WSN setup gets desynchronized. In such a setup there is no chance to re-establish a connected network due to extreme mote duty cycles.
In order to face this problem the energy management proposed in [17] was enhanced as shown in Fig. 5.

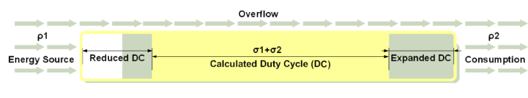



Fig. 5. Enhanced energy management splits operation range into 3 parts.

Three operation states were defined: normal operation (in case the energy pattern repeats as predicted), underflow operation (if less energy is harvested than expected) and overflow operation (more energy is harvested). If the stored energy drops out of the normal operation region the system tries to return to the normal region by automatically altering (increasing or decreasing) the suggested duty cycle. The results of the evaluation of this enhanced energy management (Fig. 6) show that the system works for 4 days in an energy neutral mode. The stored energy never exceeds the limits that were set. This way the network stays connected even though the EHD power output provides high variation and therefore makes energy prediction a hard challenge.

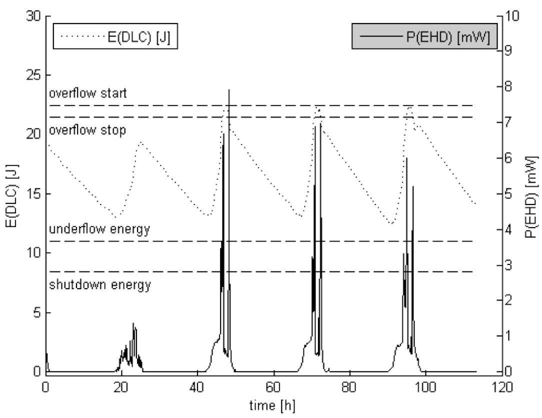



Fig. 6. Evaluation with the enhanced model using 3 operation ranges.

## V. Conclusion and Future Work

This paper presented the design and implementation of an EHS that allows the operation of wireless sensor nodes with different types of EHDs (solar cells, magnetic induction, piezoelectric and thermoelectric generators). This makes the prototype more flexible than other approaches.

A modular approach allows to attach different kinds of WSN nodes without modification of the HW in the future. The low-power design can lead to a platform with no backup battery to get solely environmental powered mote and EHS.

The prototype can be used efficiently for energy harvesting model checking. A theoretical algorithm for energy harvesting aware power management has been implemented and evaluated in the prototype. It acquires the energy balance of the overall system and models the energy source and the consumer to operate the wireless sensor node in an energy neutral mode. It has been shown how existing theory can be improved. The platform can detect and counteract energy pattern variations in real-time.

Further work might include the implementation of a model checking framework. New harvesting theories and power management strategies could be developed and tested on real HW. This simplifies the development of simulation environments.

An API has been implemented for communication abstraction with the EHS from a mote programmer's point of view. Implementation of the offered interfaces enables the platform for basic power profile characterization of SW components.

The presented novel EHS enables characterization of the energy budget of WSN nodes and thus provides the basis for network-wide energy and power management.

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, no. 1, pp. 44–51, Jan. 2001.

[3] "Pawis (power aware wireless sensors) project website," http://sourceforge.net/projects/pawis/ last visited: 16.09.2008.

[4] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," *Lecture Notes in Computer Science*, vol. 2920, pp. 1–17, Feb. 2004.

[5] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *Trans. on Embedded Computing Sys.*, vol. 6, no. 4, p. 32, Sept. 2007.

[6] "Maxwell Technologies," http://www.maxwell.com/ last visited: 16.09.2008.

[7] J. Paradiso and M. Feldmeier, "A compact, wireless, self-powered pushbutton controller," *ACM Ubicomp*, pp. 299–304, 2001.

[8] E. Lai, A. Redfern, and P. Wright, "Vibration powered battery-assisted passive rfid tag," *Lecture Notes in Computer Science*, vol. 3823, pp. 1058–1068, Nov. 2005.

[9] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer, "Simulation based verification of energy storage architectures for higher class tags supported by energy harvesting devices," *Microprocessors and Microsystems*, vol. 32, pp. 330–339, Aug. 2008.

[10] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *4th Int. Symp. on Information Processing in Sensor Networks (IPSN)*, April 2005, pp. 457–462.

[11] F. Simjee and P. H. Chou, "Everlast: long-life, supercapacitor-operated wireless sensor node," *Proc. of the Int. Symp. on Low Power Electronics and Design (ISLPED)*, pp. 197–202, 2006.

[12] P. Chulsun and H. C. Pai, "Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes," *3rd Annu. IEEE Commun. Soc. on Sensor and Ad Hoc Communications and Networks (SECON)*, vol. 1, pp. 168–177, 2006.

[13] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: enabling sustainable and scalable outdoor wireless sensor network deployments," in *5th Int. Conference on Information Processing in Sensor Networks (IPSN)*, April 2006, pp. 407–415.

[14] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. of the 2nd Int. Conference on Embedded Networked Sensor Systems*. ACM, 2004, pp. 188–200.

[15] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro power meter for energy monitoring of wireless sensor networks at scale," in *Proc. of the 6th Int. Conference on Information Processing in Sensor Networks*. ACM, 2007, pp. 186–195.

[16] T. Trathnigg and R. Weiss, "A runtime energy monitoring system for wireless sensor networks," in *Proc. of the 3rd Int. Symp. on Wireless Pervasive Computing*, May 2008, pp. 21–25.

[17] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," *Proc. of the Joint Int. Conference on Measurement and Modeling of Computer Systems*, pp. 223–234, 2004.

[18] A. Kansal and M. B. Srivastava, "An environmental energy harvesting framework for sensor networks," in *Proc. of the Int. Symp. on Low Power Electronics and Design (ISLPED)*. ACM Press, August 2003, pp. 481–486.

[19] "Texas Instruments Incorporated," http://www.ti.com/ last visited: 16.09.2008.

IEEE WCNC 2011 - Service and Application

# Designing Sustainable Wireless Sensor Networks with Efficient Energy Harvesting Systems

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics*
*Graz University of Technology*
*Austria*
{*Philipp.Glatz*, *Leander.Hoermann*, *Steger*, *RWeiss*}*@TUGraz.at*

## Abstract

*Wireless sensor networks (WSNs) are resource-constrained devices. Especially energy is scarce. Recent advances with energy harvesting system (EHS) technology now offer new opportunities for long-lived WSNs. Several prototypes of EHS-enhanced WSN platforms have recently been designed in a low-power and power-efficient way. While the field starts to mature there are still a few lessons to be learned for building low-cost and robust systems. Especially, perpetual systems pose new constraints that have not yet been formalized in well-established metrics. Several studies exist, that try to tune systems and algorithms offline for being able to cope with different situations that might occur. The prediction of solar traces is a well-known example for EHS-enhanced WSNs. However, it is a cumbersome task trying to be prepared for all possible extreme conditions especially when prediction is needed. Therefore, we introduce RiverMote which is a low-cost highly efficient design with extended black-out sustainability (BOS) capabilities. We suggest to put special focus on BOS as a mechanism that can even react to hard-to-model events in a general way without the need to rely on error-prone prediction mechanisms. RiverMote is a low-cost WSN platform with integrated EHS. It outperforms state-of-the-art EHS technology efficiency for WSNs and supports BOS of more than three weeks.*

## 1. Introduction and Related Work

Recent advances and prototyping experience in the field of energy harvesting for networked embedded systems [1] have shown that there are mainly two aspects to be considered when designing an energy harvesting system (EHS) that shall be used for supporting a wireless sensor network (WSN) platform.

Firstly, the platform needs to support very-low-power operation and therefore needs to be efficient and low on power dissipation itself. This can lead to energy neutral operation (ENO) [2] designs from a purely conceptual energetic point of view. However, the long mission time that the ENO-enabled motes have to expect will bring a huge variation of different states that the mote will be in that will be hard to be completely modeled and forethought before a deployment which fosters the need for online adaptation.

Therefore, secondly, robustness in terms of coping with extreme and unexpected situations - dependability and especially fault tolerance - comes into play. For systems that are as much energy constraint as EHS-enhanced WSNs, this usually boils down to black-out sustainability (BOS) an maybe combined with hardware resets. Situations where long-term BOS may be important include misleading energy harvesting device (EHD) prediction, dirt covering or blocking EHDs and detection of erroneous hardware or software components. Continuous checks at runtime for how long the system could sustain an immediately starting black-out can considerably improve system reliability. This can be implemented in parallel to performance-aware tasking [3] or duty cycling in general [2]. Several studies exist that are dealing with different prediction algorithms [4]. Although up to five years of solar traces have been used for offline evaluation of tasking algorithms in [5] as reported in [6], already a single exceptionally bad day can cause troubles up to rendering the network useless due to getting desynchronized as described in [7]. Despite the fact that several groups are applying that standard way of modeling with sleep/wake-up policies [8] for achieving a duty cycle (DC) of the load, we suggest to make systems more reliable through robustness in terms of BOS. We showcase a robust and low-cost solution with the perpetual RiverMote design for river level monitoring. It targets a similar application scenario as other well-known water monitoring applications that are implemented by [9] and [10]. Compared to these solutions, RiverMote is a low-cost device with costs in the range of the TelosB and Mica2 motes while RiverMote already implements an EHS and an accurate energy management unit(EMU) in hardware similar to [11]. While we could not find acccurate characterizations of other EHS and measurement systems, RiverMote has been evaluated to be as accurate as $5\,\%$ error or less for magnitudes of power significantly impacting the overall energy budget. A power state model (PSM) and energy

978-1-61284-254-7/11/$26.00 ©2011 IEEE 

efficiency model (EEM) have been profiled with a setup as presented in [12]. EEM shows EHD to end-user performance energy efficiency of more than $80\%$ which outperforms other state of the art approaches. ENO, BOS of more than 3 weeks and an energy hysteresis for system initialization ensure reliably working deployments. As a comparison, the system presented in [13] claims only 5 days of BOS.
RiverMote's accurate on-board power and energy measurement facilities allow to accurately estimate software and hardware components' energy needs similar to the design presented in [7]. This allows to easily subtract the energy needed for important (maybe error reporting) tasks from the latest energy level before BOS evaluation. This allows to implement intelligent behavior on-board on top of information from the EMU. In case the BOS comes close to the minimum acceptable down time, we suggest to send an error message. This message reports to the base station the fact that the RiverMote will deactivate until BOS exceeds the minimum down time again and will then in turn send an activation message.

## 2. RiverMote Design

RiverMote has mainly been designed for water level monitoring of rivers. There are several design constraints that have to be met in terms of sensing quality, low-power operation and ENO. For brevity, we limit detailed discussion to aspects of energy constraints and their impact on BOS. GPS-heightening and ultra-sonic transceiver distance measurement are implemented and pressure transducer interfaces are included in RiverMote's design, but they are excluded from the paper. The main constraints stem from a minimum water level sampling rate of $5 - 15\,min$, the $10 - 15\,min$ long initialization time of the GPS and the need for reliable wireless communication. For being capable of dealing with extreme conditions and repair the network online if needed, a BOS of 2 weeks is targeted. All motes, except for the base station, are to be deployed for in-situ measurements in buoys directly in the river. The network is connected wirelessly (1km+ ranging) several kilometers along the river with at least one mote deployed outside the river for injection of differential GPS correction values. This mote or a second node outside the river can be connected to a PC via USB for programming and data collection easily.

### 2.1. Power Supply and Energy Storage

RiverMote is supplied by solar cells and DLCs are used for energy storage. We supersede the need for additional hardware or software for MPPT by fulfilling the following constraints:

**Solar cells** must be large enough to provide enough energy per day for a mote DC that is capable of providing the end-user performance that is needed by the application. Their maximum output voltage must be considered when designing protection circuits and their MPP must be considered when selecting suitable voltage thresholds and hysteresis for the storage structure and its charging circuits. Cost is an issue as well.

**Input and output charging**, load matching and protection circuits must be designed with efficiency high enough that do not annul working power state and energy efficiency models (PSM and EEM).

**The storage structure** itself must be capable of storing enough energy to bridge a full modeling period and EHD pattern variation. Storage structure and - if needed - balancing circuit leakage currents and efficiencies have to be considered for PSM and EEM. We design a serial DLC structure with balancing for achieving a long lasting hardware with a suitable input voltage range for meeting the solar cells' MPP.

**Finally, the mote part** may vary DC ratio and period length for varying end-user performance and delay accordingly. We go for a slotted networking design where motes' radios are activated $2\,s$ each for sending and $28\,s$ each for listening. The closer a mote is to the base station the earlier its 2 seconds transmit slot will be. The slotted design makes the system more robust due to its simplicity [14] and there is enough space in time to counteract timing misalignments due to temperature drifts and other issues [13].

Although, batteries have more capacity compared to the DLCs they have limited lifetime and are temperature dependent. During evaluation of RiverMote its temperature varies for approximately 50 degrees each day. Furthermore, DLCs have more than 500000 charge-discharge cycles and a nominal lifetime of 10 years where NiMH batteries have a lifetime of only 500 cycles before serious capacity loss.

Energy harvested during the day must be stored in the DLCs for continuous operation at night as well. The needed capacity depends on the power consumption of the overall hardware architecture. Furthermore, a hysteresis is implemented in hardware that will only switch on the mote if enough energy can be offered for full hardware and network initialization. From that point the DLCs can constantly be drained by a fully operating mote before the lower bound of the DLC's voltage operating range is reached for a duration of worst case initialization time of the system. The time is lower bounded by the time for initial GPS configuration - afterwards the configuration is stored persistently without further significant power dissipation until a possible hardware reset. We have selected the lower bound of initialization time such that it is above the motes' DC period to ensure proper initialization network joins via its ZigBee module. This is a key idea of RiverMote's robust design. Even though we cannot imagine a scenario where the storage structure would be trained below its operating range after being initialized once, severe programming errors might do the job of completely draining the system. In such a case

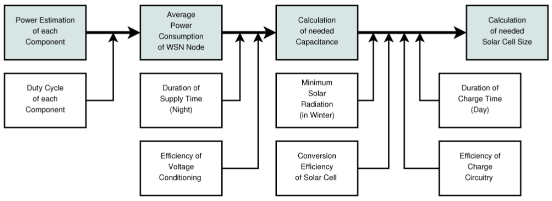



Figure 1. Calculation methodology of the DLC capacitance and the size of the solar cell.

the hardware back-off solution with the hysteresis would allow to re-initialize a RiverMote and restart its operation by joining the network with the process described above again. Back to dimensioning the storage architecture, the energy stored in $n$ capacitors can be calculated with:

$$E = \frac{1}{2} \cdot C \cdot U^2 = \frac{1}{2} \cdot C \cdot n \cdot U_C^2.$$

Additional energy of each additional DLC can be traded for power lost and cost of a further balancing circuit.

## 2.2. Power Estimation

Table 1 shows a list of RiverMote's components and their expected power dissipation for implementing the application scenario. Furthermore, Figure 1 gives an overview of the overall power dissipation calculation process presented here that incorporates the values from Table 1. The components

| Component | $t_a$ [s] | $P_a$ [mW] | $P_{sl}$ [mW] | $P_{avg}$ [mW] |
|---|---|---|---|---|
| MCU | 30 | 15.0 | 0.1 | 0.60 |
| Send | 2 | 429.0 | 0.0 | 0.95 |
| Receive | 28 | 82.5 | 0.0 | 2.57 |
| RxTx Sleep | 30 | 0.0 | 0.1 | 0.10 |
| GPS | 30 | 120.0 | 0.1 | 4.10 |
| Pressure | 1 | 250.0 | 0.0 | 0.28 |
| Ultrasonic | 1 | 10.0 | 0.0 | 0.01 |
| Other HW | 30 | 30.0 | 0.1 | 1.10 |
| Total $P_{tot}$ | | | | 9.70 |

Table 1. The MSP430F1611 and MRF24J40MB support TinyOS compatibility and ZigBee for more than $1\,km$. The GPS Fastrax UP300 supersedes repeated costy setups by an internal backup.

are active for $t_a$ seconds each activation interval $T$ which leads to their duty cycle $D$ with an average power dissipation of $P_{avg}$. It can be summed up to 9.7 $mW$. For further calculation a higher value of $P_{tot} = 12\,mW$ will be assumed. During full operation mode, all components are active and the EHS must be capable of supplying $P_{max,full}$ which in turn allows to determine the needed capacity $C_{real}$ and total average power $P_{tot,max}$ for a given efficiency

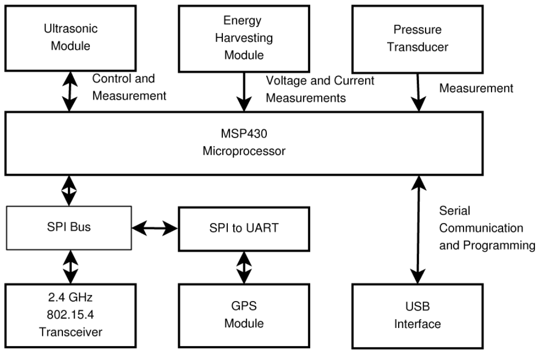



Figure 2. Hardware modules of the mote and the communication between them.

$\eta_{convert}$:

$$I_{max,full} = \frac{P_{max,full}}{U_{sup}} = \frac{854\,mW}{3.3\,V} = 258.8\,mA.$$

$$E_{supply} = P_{tot} \cdot t_{night} = 12\,mW \cdot 16\,h = 691.2\,J$$

$$E_{needed} = \frac{E_{supply}}{\eta_{convert}} = \frac{691.2\,J}{0.8} = 864\,J$$

$$C_{real} = \frac{2 \cdot E_{needed}}{U_{start}^2 - U_{stop}^2} = \frac{2 \cdot 864\,J}{(5\,V)^2 - (1.8\,V)^2} = 79.41\,F$$

$$E_{usable} = \frac{1}{2} \cdot C \cdot \left(U_{start}^2 - U_{stop}^2\right) = 1686.4\,J$$

$$P_{max} = \frac{E_{usable} \cdot \eta_{convert}}{t} = \frac{1686.4J \cdot 0.8}{16h} = 23.42\,mW$$

$$P_{leakage} \leq I_{leakage} \cdot U_n = 0.45\,mA \cdot 5\,V = 2.25\,mW$$

$$P_{tot,max} \geq P_{max} - P_{leakage} = 23.42 - 2.25 = 21.17\,mW$$

## 3. System Implementation and Setup

The system setup consists of mote and EHS implementation. Figure 2 depicts the mote's components. The main component of the mote is the microcontroller: the MSP430F1611 low-power MCU. The GPS receiver needs an extra serial interface, so an SPI-to-UART adapter is used. The 2.4GHz 802.15.4 transceiver module is connected to the SPI as well for ease of communication. Ultrasonic module and EHS are directly connected to the MSP430. The USB module is implemented for programming and communication features as known from other mote platforms as well.

Figure 3 shows a block diagram of the EHS part. The solar cell is a voltage limited current source. No voltage conditioning is needed for the DLC storage structure due to appropriately selected components for high efficiency at low cost. The nominal solar cell voltage is a little higher than that of the dual-DLC struture. Overcharge protection

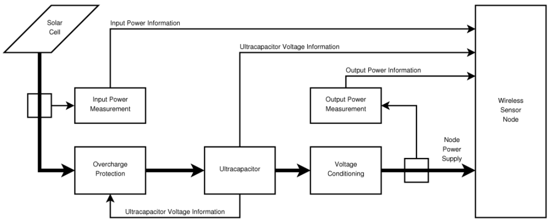



Figure 3. Design of the energy harvesting hardware. The thick arrows illustrate the flow of power and the thin ones illustrate the flow of information.

guarantees a maximum voltage at the DLCs. Other power will be bypassed. Voltage conditioning is used to generate a stable output voltage at $3.3\,V$.
For selecting proper DCs, current and voltage measurement lines have been implemented in the EMU on-board. The energy budget and input/output power can accurately be determined.

## 4. Results

For the evaluation of BOS characteristics and perpetuity constraints, we first evaluate single DLCs' leakage. Then, we introduce balancing circuitry and characterize self-measurement capabilities. Next, the overall leakage is characterized and a long-term test will be used for validating the system concept and hardware functionality.

### 4.1. Capacitor Leakage and Balancing

The National Instruments®DAQ NI PXI-6221 from [12] is used for measuring the leakage current of single DLCs. Two measurements are performed that show how a DLCs leakage depends on its clamp voltage history. First, a DLC is charged to its maximum capacity. Then the current source is detached and the leakage measurements start instantly. The second measurement keeps the maximum voltage for one day before performing leakage measurements. The DLC is charged with $E_{supply} = 1246.62\,J$ with an average current of $89.25\,mA$ for $2.8\,h$. Figure 4 shows the DLC leakage current after charging the DLC and disconnecting the power supply:

$$I_{leakage} = C\cdot\frac{\Delta U_{cap}}{\Delta t} = C\cdot(U_{cap,n+1} - U_{cap,n})\cdot samp\ rate.$$

It can be seen that a lot of energy is lost in the beginning and leakage reduces over time to an average of $0.487\,mA$ which is slightly above $0.45\,mA$ from the datasheet. However, the significant value is to be determined over the first 16 hours after detaching the power supply as this time is to be bridged with the DLC. The average leakage current during this interval was $1.398\,mA$.

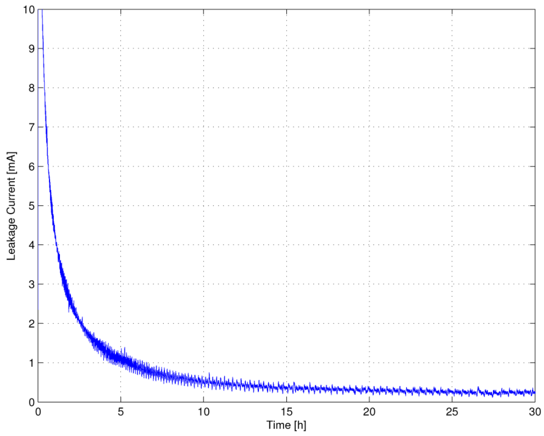



Figure 4. Leakage current of the DLC.

The second experiment shows that the DLC is more stable after keeping the clamp voltage constant for one day. The average leakage current during the first 16 hours is $0.121\,mA$ which is below a third of the value of the data sheet. While the real value will be somewhere between $1.398\,mA$ and $0.121\,mA$, we conclude that the process of BOS measurements cannot completely be done offline. We rather suggest and evaluate the process of offline characterization of significant scenarios and then use online curve fitting algorithms of low computational complexity. A self-complete offline characterization with highly nonlinear EHD patterns, single DLC leakage characteristics, capacitor balancing energy loss, other hardware components' leakage and not to forget variable component's activities would be too complex.
The balancing circuit is needed to keep both DLCs at the same voltage. This is important for not overcharging the DLCs due to unbalanced charging with only one combined overcharge protection for both of them. The average balancing current increases nearly linearly with $I_{bal,meas,0.2V} = 10\,mA$ at a voltage difference of $0.2\,V$. For up to 20% misalignment of the charging process among DLCs the minimum charging time and maximum charging current can be calculated with:

$$\Delta Q = \Delta U_{max}\cdot C_{DLC,1} = 0.3V\cdot 310F\cdot 0.8 = 74.4\,As.$$

$$t_{charge} = \frac{\Delta Q}{I_{bal,0.4V}} = \frac{\Delta Q}{I_{bal,meas,0.2V}\cdot 2} = \frac{74.4As}{20mA} = 3720\,s.$$

$$I_{charge} = \frac{Q_{tot}}{t_{charge}} = \frac{744As}{3720} = 200\,mA.$$

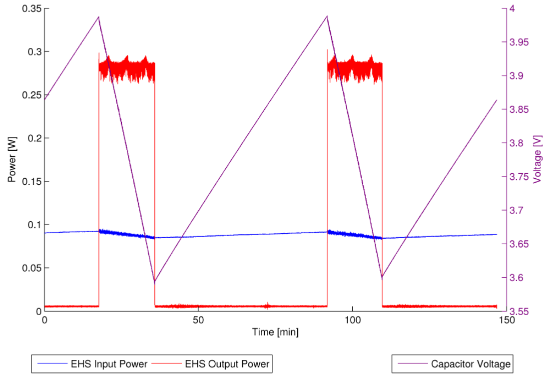



Figure 5. EHS efficiency measurement.

## 4.2. EHS Efficiency Evaluation

EHS efficiency and stored energy are:

$$\eta_{EHS} = \frac{P_{output}}{P_{input}}, E_{stored} = \frac{1}{2} \cdot C \cdot U_C^2.$$

Two thresholds are set for measuring EHS efficiency. DLCs are constantly supplied by a laboratory power supply and the GPS is used for draining them. The voltage thresholds are set in the middle of the operating voltage range: $4\,V$ and $3.5\,V$. Figure 5 shows two charge-discharge cycles of the DLCs and key results are listed in Table 2. It can be seen

| Description | Value |
|---|---|
| Maximum stored energy in the capacitor | $1233.10\,J$ |
| Minimum stored energy in the capacitor | $998.52\,J$ |
| Dynamic energy of the capacitor | $234.57\,J$ |
| Maximum capacitor voltage | $3.99\,V$ |
| Minimum capacitor voltage | $3.59\,V$ |
| Dynamic voltage of the capacitor | $0.40\,V$ |
| Duration of a high-power period | $18.02\,min$ |
| Duration of a low-power period | $55.41\,min$ |
| Duty-Cycle | 0.33 |
| Average input current | $21.57\,mA$ |
| Average input power | $88.07\,mW$ |
| Average output power | $73.24\,mW$ |
| Total Efficiency of the EHS | $83.16\,\%$ |

Table 2. Key results of the measurement of the EHS efficiency at middle voltages of the capacitor.

that the true voltage difference of the DLCs is only $0.40\,V$. The reason is the inaccuracy of the voltage measurement of the microcontroller at high mote currents. These high currents cause a higher voltage drop at the shunt resistor for the measurement and the shunt resistor of the EHS.
The resulting overall efficiency (in the less efficient lower voltage range) of $\eta_{EHS,low} = 83.16\,\%$ is better than expected (64% for $\eta_{in} = 80\%$ and $\eta_{out} = 80\%$). To the best

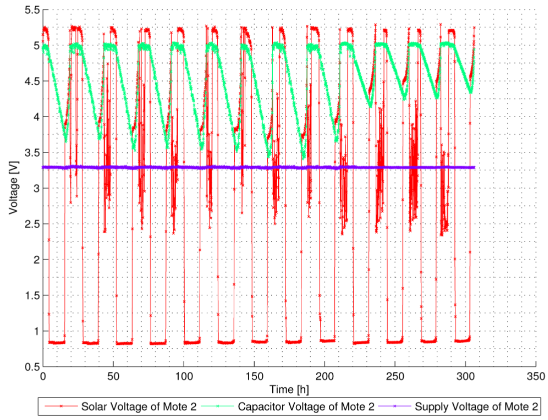



Figure 6. Measured voltages of mote 2 during the long-term test.

of the authors knowledge there exists no such an efficient EHS for WSNs without MPPT techniques. Even in case costly MPPT is used, the efficiency of a single stage could easily drop below $80\%$. Compared to that, already the output efficiency alone of the EHS in [7] has been characterized in [12] to vary for up to $50\,\%$.

## 4.3. On-board and Long-Term Measurement

The error of the solar current counter is better than $5\,\%$ during daytime. Mote current measurements' accuracy have evaluated to an accuracy of $5\%$ or better for currents of $3\,mA$ or more. Despite the fact that the relative errors of on-board runtime measurements may increase for smaller currents their impact on the energy budget are negligibly small and especially magnitudes of order below DLC leakage. Figure 6 shows self-test debug information (sent every $10\,min$) of RiverMote2 during moderate to really bad weather winter days in the middle of Europe in Austria. Other motes' and other measurements' plots are not shown for brevity, but all long-term tests have validated ENO capabilities. The GPS has always been switched on when the mote was activated. Ultrasonic transceivers have only been used for tests in the water. The results on DLC leakage, EHS efficiency, power measurement accuracy and long-term experience have shown a low cost platform. What is left open is the overall system leakage and standby and error-reporting strategies when it comes to blackouts. GPS and over sensors' characterization will be left out for brevity.
RiverMote is of similar cost as TelosB nodes without EHS-support: PCB ($45.80\,EUR$) + HW ($92.79\,EUR$) + DLCs ($30.84\,EUR$) + solar cell ($\approx 50.00\,EUR$) + GPS module ($46.86\,EUR$) = Total cost $266.29\,EUR$.

eJSE International

*EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)*

# Designing Perpetual Energy Harvesting Systems explained with RiverMote: A Wireless Sensor Network Platform for River Monitoring

Philipp Maria Glatz*
Leander Bernd Hörmann
Christian Steger
Reinhold Weiss
*Institute for Technical Informatics, Graz University of Technology, Inffeldgasse 16/1, 8010 Graz, Austria*
**Corresponding author*
*Email: Philipp.Glatz@TUGraz.at, Glatz.Philipp@GMail.com*

ABSTRACT: Wireless Sensor Network (WSN) motes are devices of small form factor. They are tailored to cost-effectiveness for monitoring and control applications. Different optimizations exist for the robust lifetime improvement of such devices, but the community still lacks a clear approach of how to achieve a robust system design that is of low cost and implements low-power optimizations. In particular, it is a demanding task to efficiently utilize energy harvesting system (EHS) technology for WSNs. However, that is the only way for implementing battery-free mote devices for achieving perpetual operation of WSNs. We demonstrate the design methodology that let us implement RiverMote. RiverMote is a case study for designing a low cost hardware system architecture combining low-power mote and a highly efficient EHS architecture. First, we provide a detailed explanation for how to deduce the prototype dimensioning parameters given the application requirements of an energy-autarkic river level monitoring system. Then we show how to select proper energy harvesting device (EHD) technologies and design all EHS stages for fitting the application-depending power state model of the integrated mote. Finally, we implement and evaluate tests of all stages for their energy efficiency as well as RiverMote's self-measurement accuracy which is crucial for robust perpetual designs.

## 1 INTRODUCTION AND MOTIVATION

Wireless sensor network (WSN) motes [1] are small networked embedded devices offering low-cost solutions to various applications. Though WSNs offer diverse functionality and a range of platforms, protocols and optimizations for dependability and powering issues [2], several lessons have been learned for how WSN deployments may fail partly or completely when it comes to real world deployments and issues with dynamic environments.

The Great Duck Island experience [3] is a prominent example for problems being experienced due to misconceptions of how dependability issues and power dissipation may interact. More recent deployments keep on experiencing different issues over and over again when it comes to mapping theory to deployments [4]. WSN deployments for environmental monitoring have to cope with changing environmental conditions. A well-known example discussing WSN deployment issues is given by [5].

The majority of optimization techniques for WSNs boil down to either (i) reducing the average power dissipation per end-user performance, (ii) dealing with dependability issues, or (iii) to handle energy more efficiently from an electrical engineering point of view.

Using energy harvesting systems (EHSs) is the state of the art approach of providing more energy and longer lifetime or even energy neutral operation (ENO) until technical breakdown. When designing EHS-enhanced WSNs, different types of optimization (i)-(iii) need to be considered at the same time.

However, conditions for developing EHSs turn out to be more complex [6] than designing WSN motes alone. First, this is due to more complex hardware and software. Second, when it comes to ENO dependability issues, all possible scenarios and faults that may occur need to be taken into account. Especially the power profiles of energy harvesting devices (EHDs) and their effects are not precisely predictable. Therefore, protocols and policies are difficult to validate with simulation-based approaches. Furthermore, EHS and EHD selection and design are subject to trading hardware system cost against robustness and computational complexity when profiling and estimating EHD profiles and dealing with their effects. EHS design is concerned with introducing different kinds of thresholds for energy budgets, synchronization and back-off times while preserving end-user performance. The problem is that considering all these aspects results in EHS-enhanced WSNs being designed for ENO, but suffering efficiency loss due to dependability protocol overhead or energy safety bounds.

Although EHS technology is in the process of maturing and several important aspects have been considered and optimized, the authors of this article could not find design methodologies for robust and efficient EHS-enhanced WSN platforms. Here, we contribute with describing the design methodology of a highly efficient EHS and a robust low-power mote platform.

### 1.1 *Contribution Claim*

We state that there is a lack of self-contained design methodologies for EHS architectures for WSNs. Therefore, we present such a methodology combining the definition of application requirements, power estimation and power state models (PSMs), EHS design and EHD selection as well as designing energy storage requirements for achieving ENO. Tests will be run given worst case settings that are known to be challenging for existing EHS technology. Furthermore, the integration of accurate self-measurement as well as safety bounds and fallback mechanisms into RiverMote will be explained in detail. Hardware components have been selected such that the overall system cost (including water proof and UV resistant housing and all hardware) is similar to state of the art motes like TelosB [7] and Mica2 [8] platforms with a sensor-board attached.

RiverMote is a perpetual energy autarkic battery-free design and can be deployed with empty energy reservoirs. A robust design is established through using a dual DLC architecture with a balancing circuit. Efficient load matching is implemented with no need for maximum power point tracking (MPPT). It outperforms state of the art approaches in terms of input to output energy efficiency. Accurate characterization of efficiency plus accurate on-board measurement circuitry allows for intelligent behaviour. The approach is complete, extensible and easy to use.

## 2 RELATED WORK

Different EHSs have been developed so far with different issues and end-user performance in mind. For achieving an EHS design tailored to an application at hand, it is important to set up a proper PSM or to have a profiling tool of suitable accuracy available. Therefore, we present in-situ measurements and tools for characterization and simulation that are of similar accuracy as well. Finally, we will give a brief overview on EHD pattern prediction and ENO modelling.

### 2.1 *EHS Enhanced WSNs*

Different types of EHDs [9] can be used and different kind of EHS architectures can be implemented. Starting with Heliomote [10] different EHSs have made use of solar energy [11], [12], [13], [14]. It is the most convenient energy source for environmental monitoring applications [15]. While Heliomote uses batteries for its energy storage, there are prominent examples of EHS architectures using DLCs like Everlast [16] and AmbiMax [17] and even specific approaches exploring different DLC storage architectures [18].

Designing an EHS consists of two main tasks. First, the load's energy demands are modelled with a mote PSM with regard to application requirements Second, possible EHS designs have to be evaluated using an expressive EHS efficiency model (EEM). Accurately measured PSM and EEM parameters are discussed in [19]. Furthermore, offline analysis results (usually by simulation) need to be validated by runtime measurements.

Different techniques impact the EEM usually trading energy efficiency for measurement accuracy or hardware cost and robustness. As an example, the simple trade-off of whether a shunt-based measurement system should be implemented in the supply path or at ground side is discussed for an embedded energy monitor [20]. A number of decisions of different complexity need to be made when designing an EHS and tuning an EEM.

When designing an EHS, a substantial trade-off is whether MPPT should be implemented or not. It may improve system efficiency, but it leads to more complex hardware or software and will increase platform cost. Everlast and AmbiMax both implement MPPT, where [16] uses CPU intervention and [17] implements a hardware mechanism. MPPT might be used if EHS efficiency is increased at reasonable hardware cost leading to more end-user performance while achieving ENO. MPPT need not be used if load-matching can be implemented efficiently by other means.

### 2.2 *Runtime Power Dissipation Measurement*

Different direct and indirect debugging and profiling techniques exist. While passive software based approaches exist – consider passive network inspection in [21] – we will concentrate on more direct and hardware-related approaches for power profiling.

PowerBench [22], SPOT [20] and the energy measurement board in [23] are well-known examples for sandwich-on in-situ measurement boards. These systems are meant to be used for WSN motes that are of similar form factor as RiverMote is intended to be. SPOT is designed with subject to meeting accuracy requirements similar to what RiverMote current measurements need to accomplish.

eJSE International

EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)

They define a power dissipation spectrum that needs to be measured with a given accuracy. However, what is missing is a clear definition of the maximum error that is still acceptable. The sum of the integral of weighted power states' errors could be upper bounded and tested for worst case examples. That is what will be done for RiverMote. A given power state measurement's error impact on the final application and its Duty Cycle (DC) will be considered.

A neat approach for deducing the worst case error as presented in [19] helps in identifying and in optimizing single sources of error. With integrating the information into suitable simulation environments, [24] allows to model the maximum error that may occur when measuring energy from the EHDs and supplying it to the mote. Combining this information with the maximum variation of EHDs' patterns that can be expected and PSM error bounds allows introducing overall thresholds and single stages' safety bounds for not violating the ENO condition.

### 2.3 *ENO and Energy Thresholds*

The most common type of modelling ENO for EHS-enhanced WSNs is $\rho/\sigma$-modelling as introduced and discussed for this type of technology in [25] and [26]. However, an evaluation on hardware in [6] shows sensitivity of such systems to variable EHD patterns. This may quickly lead to an unstable network given the fact that hardware is being designed to be cost efficient combined with changing environments and inherently unstable embedded devices' wireless communication. Summing up the power loss at different stages of the EHS - and adding errors of EEM and PSM runtime measurements where necessary – as well as introducing suitable thresholds for worst case solar irradiation given long-term meteorological data of the deployment region has not been done so far.

From a conceptual point of view the approach presented by HydroWatch [27] is similar to what is presented here. We neatly describe the methodology behind designing and testing RiverMote. The RiverMote target application will be similar to [28], but measurements will be performed with in-situ techniques placed directly into the water with components that are lightweight in terms of hardware cost and power consumption. This allows designing a more efficient architecture still being capable of ENO at less cost compared to allowing power dissipation of up to 3W in [28].

## 3 RIVERMOTE DESIGN

The design phase starts with determining the application requirements and outlining possible sensor technologies. Then, the PSD and EEM requirements can be deduced and the system can be designed according to their characteristics.

### 3.1 *Application Requirements and Sensor Choices*

RiverMote shall be capable of real-time river level monitoring. The required accuracy of the water level measurements depends on the specific application. Three possibilities for measuring the level are considered important: using a pressure transducer, GPS leveling and ultrasonic transceivers. All three are outlined briefly. The latter two are integrated and tested with RiverMote and hardware interfaces are implemented for the first. Firstly, with a pressure transducer one can get very accurate results, but accurate waterproof sensors are quite expensive. Furthermore, the sensor needs to be placed on ground of the river which may be problematic. Secondly, using a GPS receiver offers a solution where cost of the GPS device and complexity of the correction algorithm can be traded for accuracy improvements. It also has the advantage that it can be placed inside the mote's housing and can be used for a number of other application scenarios as well. Thirdly, ultrasonic transceivers can be attached to the mote's housing and are easy to use. Though they deliver inherently noisy measurements, the sensor readings can be post processed which can give accurate measurements with cheap hardware. A related project in [29] shows how embedded ranging technologies can be optimized with expressive simulations so that results hold for deployments as well.

Finally, one has to define the sampling frequency that is needed to fulfil the given task. Therefore we first define the radio to have a range in the order of several hundred meter. So a large scale deployment is still possible without the need for adding GSM or 3G support which will in turn save costs. Furthermore, the water level measurement principles demand for a fixed base station outside the water for differential measurements like changing barometric pressure anyway. This base station - which is more or less the same RiverMote hardware, but is supplied via USB and not from a solar cell - allows reading out the measurement results from a RiverMote network in range. For being able to capture river level effects with motes deployed 1km apart from each other a maximum sampling period of 5−15min allows to accurately track river flood waves. Given these application characteristics, we design and dimension different stages of the mote and its EHS according to the resulting energy needs. We add energy thresholds where appropriate to achieve a robust design. In addition, we implement hardware fallback mechanisms that stabilizes a RiverMote's energy budget in case erroneous application behaviour or software or network policies drain the energy reservoir.

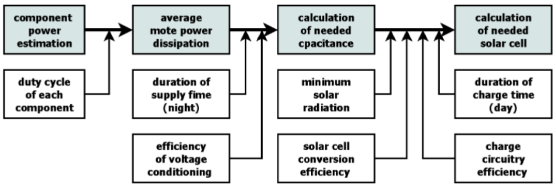



Figure 1. Calculation methodology of the DLC capacitance and the size of the solar cell.

## 3.2 *Power Supply*

The battery-less mote must harvest sufficient energy from the environment. It operates outside on the surface of a river and is exposed to the daylight. Therefore, a solar EHD is used to supply RiverMote. A little generator using the flow of the river has been considered too, but the problem of a mechanical system is that the generator has to be placed outside the housing and therefore is prone to errors. RiverMote is designed maintenance-free with solar cells. The needed size and consequently the output power depend on the power consumption of the mote including leakage and errors that occur when profiling power dissipation online. While explaining the power estimation process later we start with outlining the storage structure first.

## 3.3 *Energy Storage*

There are mainly two possibilities to store the electrical energy. First, the energy can be stored in a rechargeable battery and second it can be stored in a DLC. Batteries have much more capacity compared to the DLCs, but the disadvantages of batteries are their much more limited lifetime and temperature dependencies.

Neglecting temperature dependencies of embedded systems may lead to hard-to-debug errors when the system is deployed. Therefore, we evaluate RiverMote during a Winter in Austria where hardware temperature is measured to vary for approximately 50 degrees each day.

A DLC has typically 500.000 charge-discharge cycles and a nominal lifetime of 10 years. In contrast to that, a NiMH battery has lifetime of only 500 charge-discharge cycles before experiencing a capacity loss of down to 80% [10]. Energy harvested during day must be stored in the energy storage device for continuous monitoring during night as well. This is only a short time and can be bridged with a DLC which is not impacted by the high number of recharge cycles. The needed capacity depends on the power consumption of the overall hardware architecture. Furthermore, a hysteresis is implemented in hardware that will only switch on the mote if enough energy can be offered for full hardware and network initialization. This results in a design, such that even completely wrong software configurations (activating all components all the time) do not drain RiverMote below its lower energy threshold before a network join and software reconfiguration is possible.

Table 1. The MSP430F1611 MCU has been selected for TinyOS compatibility reasons. The MRF24J40MB has been selected as the ZigBee radio due to its range and versatile capabilities including printed antenna. The pressure transducer and the ultrasonic sensor will be switched off completely. Therefore, they have a power consumption of zero during sleep state. An activation interval of 15min is assumed here. A GPS Fastrax UP300 has been chosen due to its internal backup capabilities for its initially found configuration.

| Component | $t_a$ [s] | $P_a$ [mW] | $P_{sl}$ [mW] | $P_{avg}$ [mW] |
|---|---|---|---|---|
| MCU | 30 | 5.0 | 0.1 | 0.60 |
| Send | 2 | 429.0 | 0.0 | 0.95 |
| Receive | 28 | 82.5 | 0.0 | 2.57 |
| RxTx Sleep | 30 | 0.0 | 0.1 | 0.10 |
| GPS | 30 | 120.0 | 0.1 | 4.10 |
| Pressure | 1 | 250.0 | 0.0 | 0.28 |
| Ultrasonic | 1 | 10.0 | 0.0 | 0.01 |
| Other HW | 30 | 30.0 | 0.1 | 1.10 |
| Total $P_{tot}$ | | | | 9.70 |

The design is such that the time of being fully drained is lower bounded by the application DC period and the maximum time for component initialization. For RiverMote, this bound clearly depends on the time for downloading an initial GPS configuration and storing it persistently after a full reset. Designing RiverMote this way will also lead to a blackout sustainability of several weeks despite the fact that the design is battery-free. Although not explained in further detail, we mention here that the back-off energy reservoir easily fits into the energy reservoir of the storage architecture range that will be deduced.

Back to dimensioning the storage architecture, the energy stored in capacitors can be calculated with:

$$E = \frac{C \cdot U^2}{2}, E = \frac{C \cdot (U_C \cdot n)^2}{2 \cdot n}, E = \frac{C \cdot n \cdot U_C^{\ 2}}{2} \qquad (1)$$

The voltage U has a quadratic effect and DLCs usually have a very low nominal voltage. Therefore, two DLCs have been connected in series. The resulting capacity is only half the capacity of a single capacitor, but the nominal voltage doubles and this has a higher effect on the overall storable energy.

Unfortunately, the gain in storable energy does not come for free. Additional hardware is needed to balance the DLCs and keep them from exceeding their operating voltage. So, each additional DLC can be traded for power lost and cost of a further balancing circuitry between each of the DLCs. It can be seen that the number of capacitors connected in series directly impacts the energy budget calculation and RiverMote's system cost.

eJSE International

EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)

## 3.4 *Power Estimation and Energy Storage Device*

Table 1 shows a list of RiverMote's components and their expected power dissipation for implementing the application scenario. Their values make up a coarse grain PSM that is used for the calculations shown in Figure 1. The components are active for $t_a$ seconds during each period time T which leads to a DC and average power dissipation of 9.7 mW.

$$D = \frac{t_a}{T}, P_{avg} = D \cdot P_a + (1 - D) \cdot P_{sl} \quad (2)$$

For further calculation a higher value of 12mW will be used. The maximum current that the EHS must be capable of supplying can be calculated from summing up all power dissipation over supply voltage.

$$I_{\max, full} = \frac{P_{\max, full}}{U_{\sup}} = \frac{854mW}{3.3V} = 258.8mA \quad (3)$$

Now, one can design the energy storage structure according to the longest night during midwinter. Austria is south of the 49th latitude. The longest night lasts for about 15 h and 40 min. Here, we assume 16 h:

$$E_{blackout} = P_{tot} \cdot t_{night} = 12mW \cdot 16h = 691.2J \quad (4)$$

The energy stored in the capacitor must be converted to supply the mote. For lossy conversion an efficiency of 80% is assumed. Now, the energy that must be stored in the capacitor can be calculated:

$$E_{needed} = \frac{E_{blackout}}{\eta_{convert}} = \frac{691.2J}{0.8} = 864J \quad (5)$$

As mentioned before, it is better to connect capacitors in series to increase the storable energy. In this project, two capacitors will be connected in series. A typical nominal voltage of a DLC is 2.5V. Therefore, the resulting nominal voltage is $U_n$ = 5V. Using a converter operating from 1.8V capacity gives:

$$C_{real} = \frac{2 \cdot E_{needed}}{U_{start}^2 - U_{stop}^2} = \frac{2 \cdot 864J}{(5V)^2 - (1.8V)^2} = 79.41F \quad (6)$$

In this project, two Boostcap® DLCs with a nominal voltage of 2.5V and a capacity of 310F are used. Connected in series they provide a usable energy of

$$E_{usable} = \frac{C \cdot \left(U_{start}^2 - U_{stop}^2\right)}{2} = 1686.4J \quad (7)$$

and a maximum power consumption of the mote of

$$P_{\max} = \frac{E_{usable} \cdot \eta_{convert}}{t} = \frac{1686.4J \cdot 0.8}{16h} = 23.42mW \quad (8)$$

The maximum DLC leakage current is 0.45mA.

$$P_{leakage} \le I_{leakage} \cdot U_n = 0.45mA \cdot 5V = 2.25mW \quad (9)$$

The highest tractable mote power dissipation is.

$$P_{tot, \max} \ge P_{\max} - P_{leakage} = 23.42 - 2.25 = 21.17mW \quad (10)$$

In theory, RiverMote can easily be supplied by the structure. Detailed practical evaluation will be given.

## 3.5 *Solar Cell Calculation*

Finally, the power needed and consequently the size of the solar cell will be calculated. Storing energy from the EHD is not lossless. With 80% input efficiency the total energy needed per day is 1036.8J.

$$E_{24h} = P_{tot} \cdot \left(t_{day} + t_{night}\right) = 1036.8J \quad (11)$$

Adding leakage, input and conversion efficiencies

$$E_{24h, leakage} = P_{leakage} \cdot \left(t_{day} + t_{night}\right) = 194.4J \quad (12)$$

$$E_{24h, solar} = \frac{E_{24h}}{\eta_{charge} \cdot \eta_{convert}} + \frac{E_{24h, leakage}}{\eta_{charge}} = 1863J \quad (13)$$

the total solar power can be calculated with:

$$P_{solar} = \frac{E_{24h, solar}}{t_{day}} = \frac{1863J}{8h} = 64.69mW \quad (14)$$

This power must be provided by the solar cell also during days with bad weather conditions. A monthly-averaged horizontal daily extra-terrestrial irradiation of $P_{D,avg}$ = 2120W/m² at a latitude of 50° in December is mentioned in [30]. The converting coefficient of solar light into electrical power is about 10%. Therefore, the needed solar radiation and solar cell size can be calculated with:

$$P_{radiation} = \frac{P_{solar}}{\eta_{day}} = \frac{64.69mW}{0.1} = 646.9mW \quad (15)$$

$$A_{cell, sun} = \frac{P_{radiation}}{P_{D, avg}} = \frac{646.9mW}{212mW / cm^2} = 3.05cm^2 \quad (16)$$

A worst-case scenario must be considered, where the DLC can only store the energy needed for one night of continuous operation. The minimum solar irradiation (power density) at a very cloudy and rainy day is about $P_{D,min}$ = 3mW/cm² [31]. This is equal to a radiation of 30W/m². The value is coherent with the diagrams on page 37 in [30]. Now, the needed area of the solar cell can finally be calculated as follows.

$$A_{cell} = \frac{P_{radiation}}{P_{D, \min}} = \frac{649.9mW}{3mW / cm^2} = 215.6cm^2 \quad (17)$$

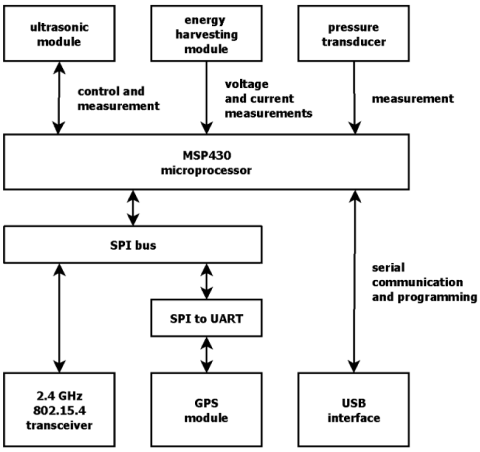



Figure 2. RiverMote hardware modules and interfaces.

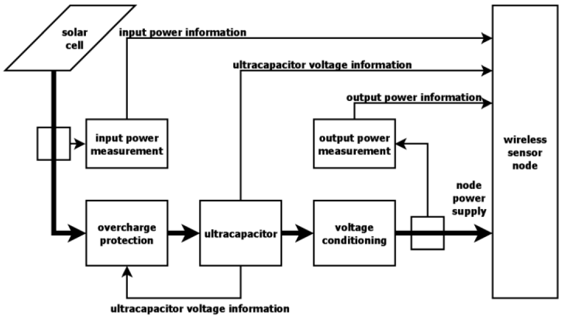



Figure 3. RiverMote control and power design.

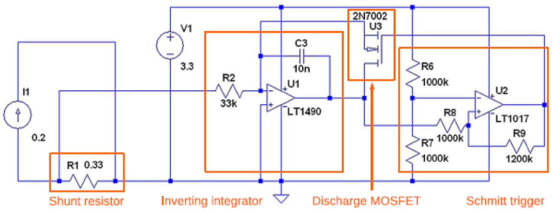



Figure 4. Circuit diagram of the current counter with its inverting integrator and Schmitt trigger[3] as main parts.

The resulting size of the solar cell is suitable to be integrated into the housing. Therefore, a compact and robust system can be built. If it would not match needs, model parameters would have to be tuned.

## 4 SYSTEM IMPLEMENTATION AND SETUP

The system setup consists of mote and EHS implementation.

### 4.1 *Mote Hardware Design*

Figure 2 depicts RiverMote's components built around the MSP430F1611 low-power microcontroller from Texas Instruments®. The GPS receiver needs an extra serial interface. An SPI-to-UART adapter is used. It receives data from the SPI and transmits it to the GPS receiver and vice versa. No direct connection to the microcontroller is needed. The 2.4GHz 802.15.4 transceiver module is connected to the SPI as well for fast and easy communication. The ultrasonic module and the EHS module are directly connected. Ultrasonic control lines are directly operated by the MSP430 whereas the EHS can operate completely control-free so that software errors cannot alter system robustness. Current and voltage measurements can be done directly by the MSP430. Programming RiverMote via a bootstrap loader and runtime communication for debugging and configuration from a PC-connected base station RiverMote can be done directly via USB.

### 4.2 *Energy Harvesting System Design*

Figure 3 shows a block diagram of the EHS part. The solar cell transforms the solar energy into electrical energy. It is a voltage limited current source. A short is no problem for a solar cell. The electrical power will be converted into heat in the solar cell and the connecting leads. As can be seen in Figure 2, no voltage conditioning is done between the solar cell and the DLCs. This can only be done if the solar cell is selected appropriately and connected in line with the DLCs. Section 5 will show results on the EHS's input efficiency that results from this design. High efficiency is achieved without any need for costly additional hardware or control for load matching mechanisms. The voltage of the solar cell at nominal power output should be a little bit higher than the nominal voltage of the DLCs. This is because some losses and a lower output voltage of the solar cell at dark light conditions must be considered. The solar cell ASI3Oo05/162/192FAmod from Schott Solar has been selected. An overcharge protection guarantees a maximum voltage at the DLCs. Voltage conditioning is used to generate a stable output voltage at 3.3V, because most of the components are supplied with this voltage. The voltage at the DLCs changes during operation and must be stabilized with a buck-boost converter. The stored energy and the input and output power must be measured, because the system needs to know the current energy state. Then the system is able to select a proper DC to ensure continuous operation. Figure 4 shows the circuit diagram. The voltage drop over shunt resistor R1 is input to the integrator. This voltage is accumulated by the integrator until a certain threshold is reached. This threshold is given by the upper threshold of the Schmitt trigger. Next, the integrating capacitor C3 is discharged via MOSFET U3 down to the lower threshold. Then, a pulse used as current counter is generated and the process starts again. Figure 5 shows a RiverMote prototype implementation without a solar cell or housing.

eJSE International

EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)

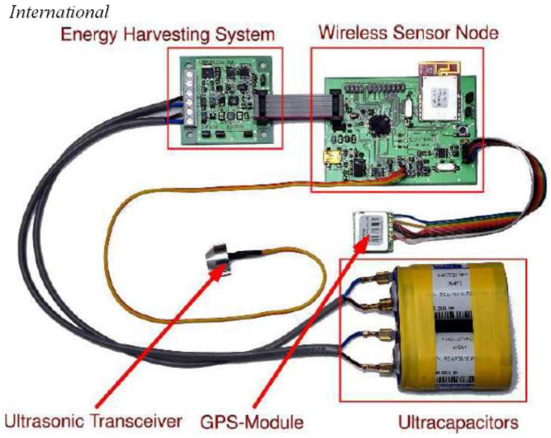



Figure 5. Photograph of a RiverMote prototype.

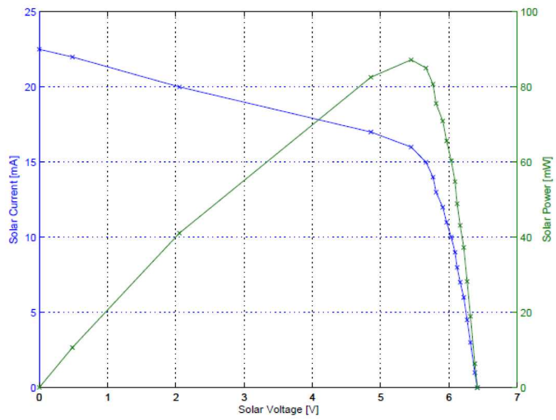



Figure 7. Profiling solar cell characteristics.

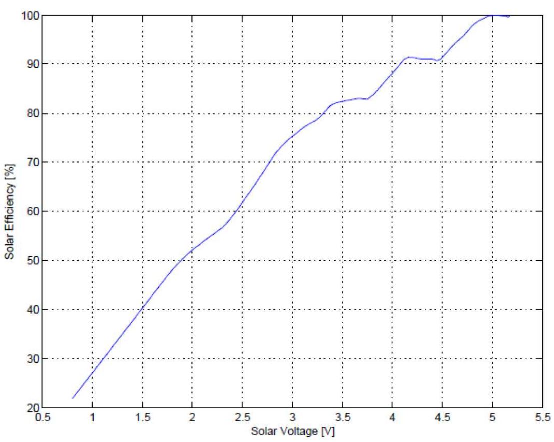



Figure 6. Solar efficiency depending on input clamp voltage.

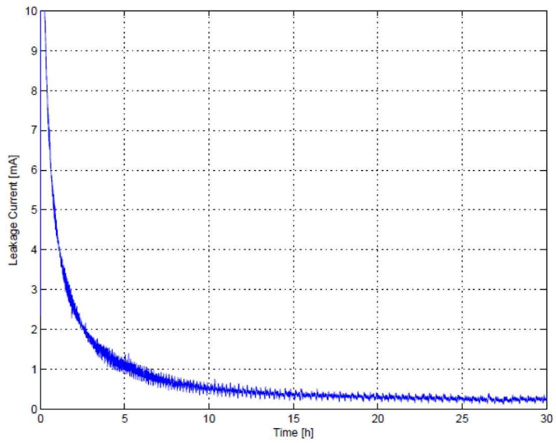



Figure 8. The DLC leakage current is settling down.

## 5 PLATFORM EVALUATION

Evaluation includes input efficiency of the solar cells, effects of DLC loading and its leakage, overall EEM, measurement circuitry evaluation, long-term measurements, sensor, radio and cost issues.

### 5.1 *Solar Cell Evaluation*

Figure 6 shows V-I characteristics of the solar cell for bad light conditions on a cloudy December day with its maximum power point at 5.5V which will be lower for better conditions. This perfectly suits the design of RiverMote. Figure 7 shows the solar efficiency during the charging process for solar efficiency for the maximum power point being 100%:

$$\eta_{solar} = \frac{P_{solar}}{\max(P_{solar})} \cdot 100\% \tag{18}$$

It can be seen that the solar efficiency is better than 80% if the voltage of the DLCs is higher than 3V. In this case, the solar voltage is about $U_{solar}$ = 3.4V, because of the forward voltage of the Schottky diode. Therefore, the voltage of the DLCs should be kept as high as possible.

### 5.2 *Capacitor Leakage*

DLC leakage current has been measured after charging the DLC and disconnecting the power supply.

$$I_{leak} = \frac{C \cdot \Delta U_{cap}}{\Delta t} = C \cdot \left( U_{cap,n+1} - U_{cap,n} \right) \cdot rate \tag{19}$$

The average leakage current during the whole measurement was 0.487mA. This is slightly above the

Table 2. Key results of EHS efficiency measurements.

| Description | Value |
|---|---|
| Maximum stored energy in the capacitor | $1233.10J$ |
| Minimum stored energy in the capacitor | $998.52J$ |
| Dynamic energy of the capacitor | $234.57J$ |
| Maximum capacitor voltage | $3.99V$ |
| Minimum capacitor voltage | $3.59V$ |
| Dynamic voltage of the capacitor | $0.40V$ |
| Duration of a high-power period | $18.02min$ |
| Duration of a low-power period | $55.41min$ |
| Duty-Cycle | 0.33 |
| Average input current | $21.57mA$ |
| Average input power | $88.07mW$ |
| Average output power | $73.24mW$ |
| Total Efficiency of the EHS | 83.16% |

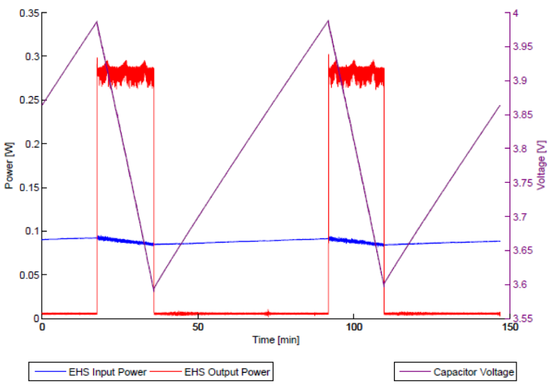



Figure 9. EHS efficiency measurements.

expected leakage current according to the data sheet which is 0.45mA [32] with an initial peak of more than 10mA. However, the significant value of the leakage current is the average leakage current of the first 16 hours after disconnecting the power supply. This time span must be bridged with the DLC. The average leakage current during this interval was 1.398mA. Conditions are relaxed again, becaused it must be considered that the DLC was disconnected immediately after reaching the maximum voltage of 2.5V. A second measurement was done. The maximum voltage was kept constant for one day. After this time, the DLC was disconnected and the leakage current was measured. Figure 8 shows that there is no initial peak any more. The average leakage current during the first 16 hours is 0.121mA. So, we can get down below a third of the value of the data sheet.

### 5.3 *Capacitor Balancing*

The balancing circuit is needed for not overcharging the DLCs due to unbalanced charging with only one combined overcharge protection for both. The balancing current at a difference of 0.2V is 10mA. The maximum charge current of the DLCs can be calculated with the results of the previous measurement. The tolerance is ±20%(d = 0.2) for the capacitors. The worst-case scenario and total capacity is:

$$C_1 = C\cdot(1-d), C_2 = C\cdot(1+d), C_{tot} = C\frac{1-d^2}{2} \tag{20}$$

The total charge is equal to that of the single DLCs:

$$Q_{tot} = Q_{uc1} = Q_{uc2} = Q_{full,ch}\cdot C\cdot\frac{1-d^2}{2} = 744As \tag{21}$$

The smaller capacitor's voltage after charging is:

$$U_{uc1} = \frac{Q_{uc1}}{C_{DLC,1}} = U_{full,ch}\cdot\frac{1+d}{2} = 5V\cdot\frac{1.2}{2} = 3V \tag{22}$$

The difference to the surge voltage of the DLC is:

$$\Delta U_{max} = U_{uc1} - U_{max} = 0.3V \Rightarrow \Delta Q = \Delta U_{max}\cdot C_1 = 74.4As \tag{23}$$

Now, the minimum charging time can be calculated:

$$t_{ch} = \frac{\Delta Q}{I_{bal,0.4V}} = \frac{\Delta Q}{I_{bal,meas,0.2V}\cdot 2} = \frac{74.4As}{20mA} = 3720s \tag{24}$$

Thus the maximum charging current is:

$$I_{ch} = \frac{\Delta Q}{t_{ch}} = \frac{74.4As}{3720s} = 200mA \tag{25}$$

The maximum charging current constraint is greatly relaxed, because the discharge threshold is set to be 1.8V. The selected solar cells can be used.

### 5.4 *EHS Efficiency Evaluation*

The measurement setup presented in [19] is used for the measurements. The input power and the output power can be calculated with:

$$P_{input} = U_2\frac{U_{R_2}}{R_2}, P_{output} = U_1\frac{U_{R_1}}{R_1} \tag{26}$$

EHS efficiency and the stored energy are as follows.

$$\eta_{EHS} = P_{output}/P_{input}, E_{stored} = \frac{1}{2}\cdot C\cdot U_C^2 \tag{27}$$

Figure 9 shows two charge-discharge cycles of the DLCs. The key results of this measurement are listed in Table 2. The input current during the whole measurement was $I_{in}$ = 21.57mA. The small variation of the input power is caused by the changing voltage of the DLCs. The resulting efficiency of 83.16% is better than expected. The efficiency consists of the efficiency of input circuit and the efficiency of the output circuit. Both have been assumed to be 80%. The resulting efficiency of the EHS would have been 64%. Therefore, the real circuit is much better than expected and no additional MPPT hardware needs to be implemented. RiverMote's efficiency outperforms other EHS approaches.

eJSE International

*EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)*

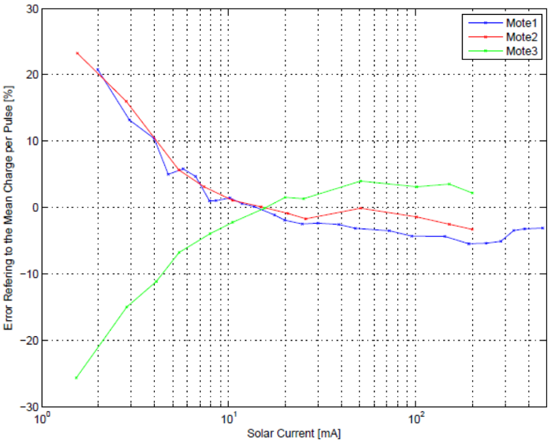



Figure 10. Relative error of the solar current counter.

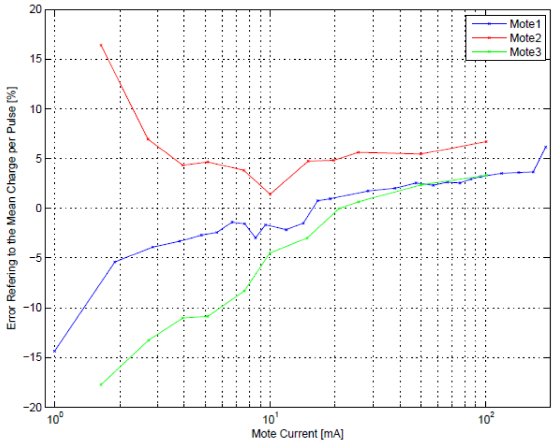



Figure 11. Relative error of the mote current counter.

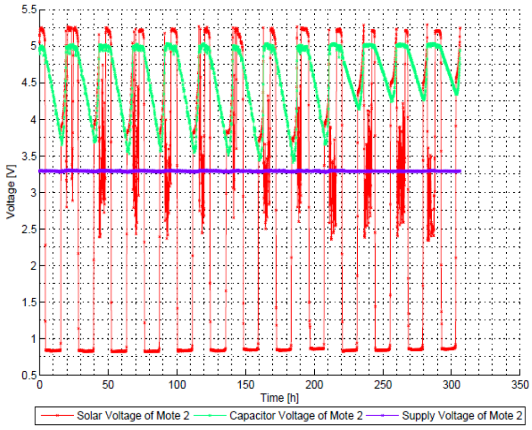



Figure 12. Voltage measurements of RiverMote 2.

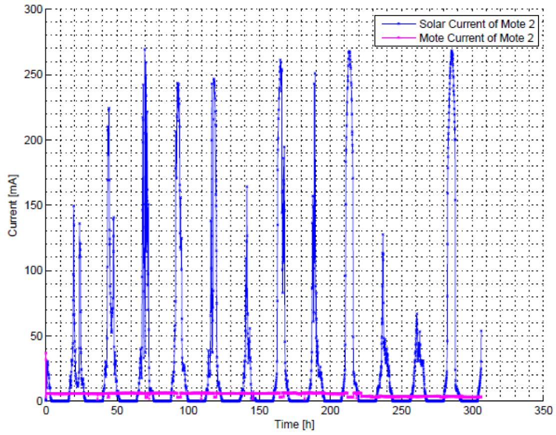



Figure 13. Current measurements of RiverMote 2.

Already the output efficiency alone of the EHS in [6] has been characterized in [19] to vary for up to 50%.

## 5.5 *Current Measurement Error Evaluation*

Figure 10 shows that the error can increase to 20% at low solar currents. However, it works properly at high enough solar currents and the error at lower values does not violate ENO conditions due to thresholds introduced in the system. The solar current is higher than 10mA during daytime even at bad weather conditions. Therefore, the error of the solar current counter is better than 10% during daytime. The error of the mote current measurement depending on the mote current is shown in Figure 11. It can be seen that the error is up to 20% at low mote currents. However, similar conditions apply as for the mote current measurement. Due to the fact that the current of the mote is about 1.5mA in LPM 3, the impact of an error of the current measurement of 20% is still below the DLC leakage as defined in its data sheet.

## 5.6 *Long-Term Evaluation*

Here we present the results of the long-term measurements using RiverMotes 2 and 3. Both motes are initially configured to send their data every 10min. Mote 2 is placed under a clear-transparent cover. Mote 3 is mounted in a water-proof housing with a greyish transparent cover. Both motes measure their energy state and the temperature of the environment. Figure 12 shows the measured voltages of mote 2. Mote 3 has shown similar results. The RiverMotes operate continuously without manually recharging the DLCs even during days of bad weather. Figure 13 shows the current of mote 2 – again mote 3 has shown similar results. The traces of the input and output power have been similar to the traces of the solar and mote current and are left out for brevity. Tests show that the temperature in the housing varies up to 50°C per day as can be seen in Figure 14.

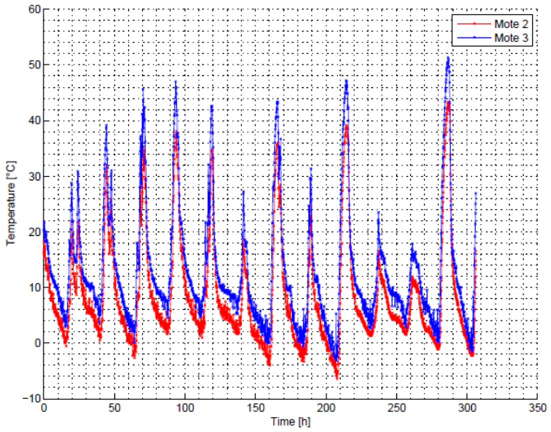



Figure 14. Temperatures measured at RiverMotes 2 and 3.

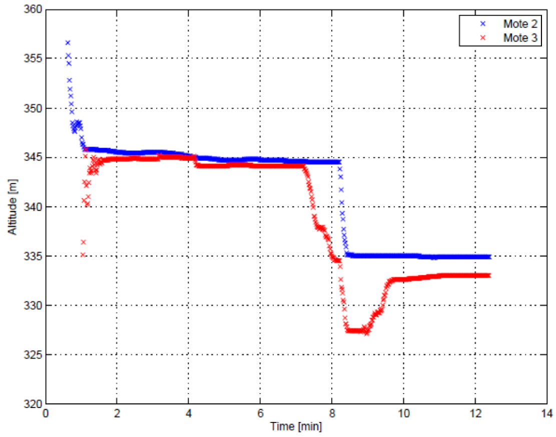



Figure 16. GPS leveling measurements.

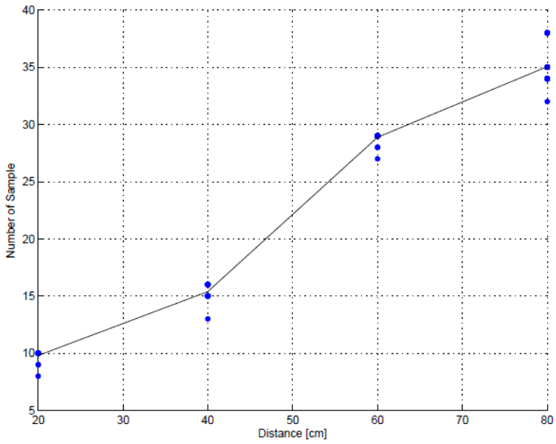



Figure 15. Ultrasonic distance measurements.

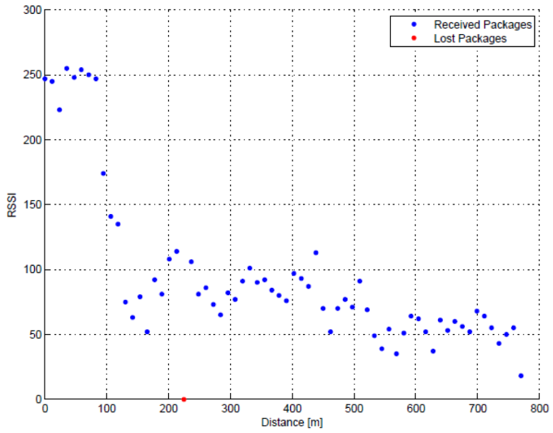



Figure 17. RSSI value over different distances.

### 5.7 *Sensor Accuracy*

The ultrasonic transceiver and the GPS have been profiled for their accuracy. As expected, the GPS - without further post processing or applying differential GPS correction - gives only coarse results. The results for initialization after reset and movement along the shore of Schwarzlsee near Graz is shown in Figure 16. Mote 3 is carried down the shore and back again. The basics of the measurement principle are working, but further post processing will be needed for accurate measurements. Another option for achieving accurate results is using the ultrasonic transceiver which can be quite accurate as shown in Figure 15.

### 5.8 *Communication Range Evaluation*

The range measurements are taken at Schwarzlsee near Graz as well. The receiver is placed on the dashboard of a car that moves at 35 km/h. Figure 17 shows that the system is working correctly even for large distances. It can be seen that most of the packets are received successfully. So, the radio can efficiently span large distances along river deployments. For an even more robust deployment we suggest to put 2 RiverMotes per km.

## 6 CONCLUSION AND FUTURE WORK

RiverMote – an EHS-enhanced WSN platform – is characterized along with a neat description of its design process. This includes setting up the application requirements, examining environmental conditions, EHD selection, EHS design, explicit and implicit measurement setups, component selection and mote design as well as testing and evaluation.

The design is battery-free. Solar cells are used for harvesting and DLCs are used for storing the energy. The setup can be reconfigured at runtime. This allows running the system as testbed as well. Long-term measurements' energy balance and power dissipation profiles are shown. The whole design process is performed in a way such that it leads to an energy autarkic platform and allows ENO. Dynamic reconfiguration allows adapting the end-user per-

eJSE International

EJSE Special Issue: Wireless Sensor Networks and Practical Applications (2010)

formance and the energy drain. Erroneous conditions for draining the energy reservoir (e.g. with over-sampling the GPS) are tested. In case such software errors occur there are cheap hardware mechanisms that allow fully restarting and initializing a RiverMote and let it join the network again in a robust way. Additional hardware for protection, balancing, conversion and matching is analysed and implemented where appropriate. The highly efficient design is achieved with hardware of reasonably low cost. It is similar to state of the art motes without EHS. It offers different types of bus systems and connectors that allow easily attaching new sensor hardware, programming the system, and connecting a RiverMote that is working as a base station to a PC. A GPS receiver along with GPS leveling measurements and ultrasonic transducers have been integrated and profiled with the system.

Future directions are to add support for design decisions that are made when designing RiverMote or similar EHS-enhanced WSN platform to the TOSPIE2 environment [24]. Considering their design trade-offs and effects in its tools may contribute to the modelling and simulation environment. It could automatically be adapted to future EHS design descriptions and their PSMs and EEMs and be used for design space exploration when designing novel energy harvestings architectures.

## 7 REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Comput. Netw., 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. Comput. Netw., 52(12), 2008.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In WSNA'02, 2002.

[4] J. Beutel and M. R. K. Römer, M. Woehrle. Deployment techniques for wireless sensor networks. Sensor Networks: Where Theory Meets Practice, 2009.

[5] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. IPDPS'06, 2006.

[6] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss. A measurement platform for energy harvesting and software characterization in WSNs. In Wireless Days 08, 2008.

[7] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In IPSN'05, 2005.

[8] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization, 2001.

[9] S. Chalasani and J. M. Conrad. A survey of energy harvesting sources for embedded systems. IEEE Southeast Conference, 2008.

[10] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. Design considerations for solar energy harvesting wireless embedded systems. In IPSN'05, 2005.

[11] J. Jeong, X. Jiang, and D. Culler. Design and analysis of micro-solar power systems for wireless sensor networks. In INSS'08, 2008.

[12] D. Brunelli, L. Benini, C. Moser, and L. Thiele. An efficient solar energy harvester for wireless sensor nodes. In DATE'08, 2008.

[13] P. Stanley-Marbell and D. Marculescu. An 0.9 x 1.2", low power, energy-harvesting system with custom multi-channel communication interface. In DATE '07, 2007.

[14] P. Corke, P. Valencia, P. Sikka, T. Wark, and L. Overs. Long-duration solar-powered wireless sensor networks. In EmNets'07, 2007.

[15] C. Park, J. Liu, and P. H. Chou. Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring. In IPSN'05, 2005.

[16] F. Simjee and P. H. Chou. Everlast: long-life, supercapacitor-operated wireless sensor node. In ISLPED'06, 2006.

[17] C. Park and P. H. Chou. Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes. SECON'06, 1, 2006.

[18] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer. Simulation based verification of energy storage architectures for higher class tags supported by energy harvesting devices. Microprocess. Microsyst., 32(5-6), 2008.

[19] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss. A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency. In PerSeNS'10, 2010.

[20] X. Jiang, P. Dutta, D. Culler, and I. Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In IPSN'07, 2007.

[21] K. Römer and J.Ma. Pda: Passive distributed assertions for sensor networks. In IPSN'09, 2009.

[22] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen. PowerBench: A Scalable Testbed Infrastructure for Benchmarking Power Consumption. In IWSNE'08, 2008.

[23] T. Trathnigg and R. Weiss. Towards runtime support for energy awareness in wireless sensor networks. In Notere'07, pages 49–54, 2007.

[24] P. M. Glatz, C. Steger, and R. Weiss. Poster abstract: Tospie2: Tiny operating system plug-in for energy estimation. In IPSN'10, 2010.

[25] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. ACM Trans. Embed. Comput. Syst., 2007.

[26] N. Jaggi, K. Kar, and A. Krishnamurthy. Near-optimal activation policies in rechargeable sensor networks under spatial correlations. ACM Trans. Sen. Netw., 4(3), 2008.

[27] J. Taneja, J. Jeong, and D. Culler. Design, modeling, and capacity planning for micro-solar power sensor networks. In IPSN'08, 2008.

[28] D. Hughes, G. Blair, G. Coulson, P. Greenwood, B. Porter, P. Smith, and K. Beven. An adaptable wsnbased flood monitoring system. 2008.

[29] P. M Glatz, C. Steger, R. Weiss. Design, simulation and measurement of an accurate wireless sensor network localization system. In PM2HW2N'10, 2010.

[30] T. Muneer. Solar Radiation and Daylight Models. Elsevier Butterworth-Heinemann, 2004.

[31] SchottSolar. Asi oem outdoor solar modules.

[32] MaxwellTechnologies. Datasheet bcap0310.

European Wireless 2011, April 27-29, 2011, Vienna, Austria ISBN 978-3-8007-3343-9 © VDE VERLAG GMBH

# Low-Cost Reliable Blackout Sustainability of Wireless Sensor Networks with Energy Harvesting Systems

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics, Graz University of Technology, Graz, Austria*
$\{Philipp.Glatz,\ Leander.Hoermann,\ Steger,\ RWeiss\}$*@TUGraz.at*

## Abstract

*Wireless sensor network (WSN) motes are typically small and networked embedded systems. They are of limited computational power and suffer severe resource constraints - especially energy is scarce. With the advent of energy harvesting system (EHS) technology new opportunities open up for long-lived WSNs. Several prototypes of EHS-enhanced WSN platforms have recently been designed in a low-power and energy-efficient way. However, while the field starts to mature there are still a few lessons to be learned for building low-cost and robust systems. While the term low-cost may easily be defined by means of calculating what hardware and maintenance cost per end-user performance for a given business model, it is more complicated to define what is meant by the term robust. For perpetual systems one has to find a formal way of validating whether a system implementation - including all low power hardware and networking approaches - can sustain a given quality of service despite harsh environmental conditions or faulty operation. For the presented case study with the RiverMote hardware platform, we model its black-out sustainability (BOS). We can validate that with its robust low-power design and BOS of more than three weeks can be achieved. Perpetual operation with error detection, fall-back mechanisms, error reporting and self-healing of the system is possible. The system relies on more easy-to-validate system-level mechanisms than other approaches that concentrate on energy harvesting device pattern prediction and therefore have to assume robust forecasting of environmental conditions.*

## 1. Introduction and Related Work

Recent energy harvesting system (EHS) architectures for wireless sensor network (WSN) [1] motes are discussed in [2] with their components' and methods' characteristics accordingly. Common design goals for EHS-enhanced WSN systems are low cost, lifetime improvement, low power, reliable performance, coverage, scalability, ease-of-operation and many others. Improving one part of a metric should not inhibit performance in another part. So, one should group metrics with similar issues and strong dependencies among each other. This makes the problem more tractable than optimizing each metric on its own. Mainly, two groups can be identified.

Firstly, systems should maximize performance for fulfilling application needs while being of low cost given certain environmental conditions. These demands lead to the need for low-power and energy-efficient platforms. [3] provides a mapping from predicted ambient energy that is expected to be available to dimensioning EHS energy storage components. An adaptation mechanism is proposed for maximizing end-user performance while not violating the constraint of energy neutral operation (ENO). However, the long mission time that the ENO-enabled motes have to expect will bring a huge variation of different states that the mote will be in. These will be hard to completely be modeled and forethought before a deployment which fosters the need for online adaptation. This brings us to the next group of considerations, namely robustness. [4] provides a simple example for an eventuality where an unexpectedly low energy harvesting device (EHD) [5] pattern drains an EHS below its energy threshold.

So, secondly, robustness in terms of coping with extreme and unexpected situations - dependability and especially fault tolerance but also scalability - has to be considered. Especially autonomously acting EHS-enhanced WSNs - that are running at a low duty cycle (DC) - usually need a considerable amount of time for communicating system-wide adaptation or to apply self-healing techniques. Furthermore, localized algorithms - especially, such that rely on autonomous topology exploration - may significantly impact motes' energy budgets and considerably tighten ENO constraints. A robust but hardly ever implemented way of relaxing constraints is to improve black-out sustainability (BOS) of an EHS combined with hardware resets. Situations where long-term BOS may be important include misleading EHD pattern prediction, dirt covering or blocking EHDs and detection of erroneous hardware or software components.

In parallel to performance-aware tasking [6] or duty cycling in general [3] the system could continuously check its BOS time - how long it could sustain an immediately starting black-out period. This can considerably improve reliability and performance of a system at the same time, because it relaxes the need for incorporating possible prediction errors into DC calculation and sleep/wake-up policies [7]

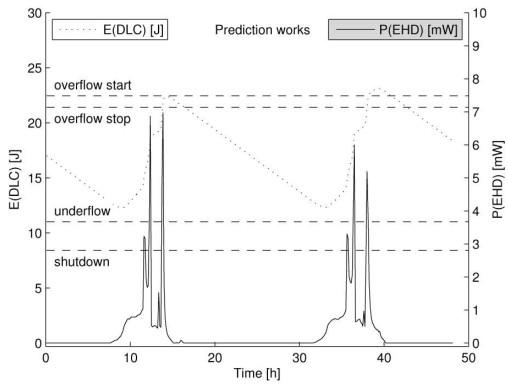



Figure 1. The EHS setup from [4] is analyzed. Plots show two days of operation. Outer thresholds are indicating the system's operating range. The DC is set for enabling ENO according to the first day's EHD pattern. For ease-of-understanding load adaptation or adaptive duty cycling is not applied.

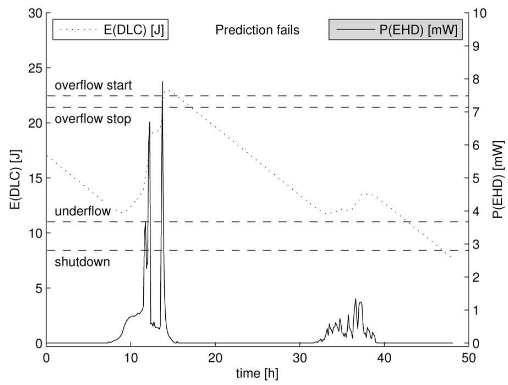



Figure 2. Environmental conditions limit EHD output on the second day. ENO conditions cannot be fulfilled with the learned DC. Additionally draining the DLC with negotiation of a network wide adaptive DC could drain it below its operating range.

for achieving ENO beforehand. Instead, continuous checks for situations that are threatening reliable operation - e.g. unexpected performance of EHD pattern prediction algorithms [8] and [9] or underestimated power consumption of components or timing of tasks - can be used to trigger fallback mechanisms. Although up to five years of solar traces have been used for offline evaluation of tasking algorithms in [10] as reported in [9], already a single exceptionally bad day can cause troubles up to rendering the network useless due to getting desynchronized as described in [4]. Fallback mechanisms may apply hardware resets, power hungry components' DC adaptation, or reporting error messages to a control instance at the sink or host computer.

### 1.1. A Motivating Example

As a motivating example and to show what problems might occur, Figure 1 shows traces of the double layer capacitor (DLC) energy budget and the EHD output pattern of an existing EHS platform used for supplying a Mica2 mote. The DC is set in a way such that ENO can be achieved. Running the the same DC on the second day still allows performing ENO within the thresholds where reasonable efficiency can be guaranteed.

A more critical example is shown in Figure 2. Here, on the second day a reduced EHD power output is observed due to bad conditions at this mote's solar panel. Consider the case where adaptive duty cycling is used to adapt the end-user performance in a network. We argue that the control overhead for negotiation of a DC over the whole network may critically impact this mote's and other motes' energy budgets. A passive approach - e.g. just sending a single error message to a sink node and waiting for the next modeling period where ENO conditions can be satisfied - could be applied. This way, motes need not decide locally whether a mechanism for network wide DC adaptation would scale to demands of the current topology or not. The time that a mote can play this passive role depends on the time that it can sustain periods without EHD output, possibly DLC leakage, and the energy needed to join the network again. For this reason, RiverMote, the platform presented in this paper, claims to provide improved robustness at low cost, especially by means of high efficiency of the EHS and its high BOS.

### 1.2. RiverMote Characteristics

In this paper, we showcase the design and implementation of a robust and low-cost solution with the perpetual RiverMote design for river level monitoring. It targets a similar application scenario as other well-known water monitoring applications that are implemented by [11] and [12]. Compared to these solutions, RiverMote is a low-cost device with costs in the range of the TelosB and Mica2 motes while RiverMote already implements an EHS and an accurate energy management unit (EMU) in hardware similar to [13]. While we could not find accurate characterizations of other EHS and measurement systems, RiverMote has been evaluated to be as accurate as $5\,\%$ error or less for magnitudes of power significantly impacting the overall energy budget. A power state model (PSM) and energy efficiency model (EEM) have been profiled with a setup as presented in [14]. EEM shows EHD to end-user performance energy efficiency

of more than 80% which outperforms other state of the art approaches. ENO, BOS of more than 3 weeks and an energy hysteresis for system initialization ensure reliably working deployments. As a comparison, the system presented in [15] claims only 5 days of BOS.
RiverMote's accurate on-board power and energy measurement facilities allow accurately estimating software and hardware components' energy needs similar to the design presented in [4]. This allows easily subtracting the energy needed for important (maybe error reporting) tasks from the latest energy level before BOS evaluation. This allows implementing intelligent behavior on-board on top of information from the EMU. In case the BOS comes close to the minimum acceptable down time, we suggest to send an error message. This message reports to the base station the fact that the RiverMote will deactivate until BOS exceeds the minimum down time again and will then in turn send an activation message.

## 2. RiverMote Design

RiverMote has mainly been designed for water level monitoring of rivers. There are several design constraints that have to be met in terms of sensing quality, low-power operation and ENO. For brevity, we limit detailed discussion to aspects of energy constraints and their impact on BOS. GPS-heightening and ultra-sonic transceiver distance measurement are implemented and pressure transducer interfaces are include in RiverMote's design, but they are excluded from the paper. The main constraints stem from a minimum water level sampling rate of $5 - 15\,min$, the $10 - 15\,min$ long initialization time of the GPS and the need for reliable wireless communication. For being capable of dealing with extreme conditions and repair the network online if needed, we target 2 weeks BOS. All motes, except for the base station, are to be deployed for in-situ measurements in buoys directly in the river. The network is connected wirelessly (1km+ ranging) several kilometers along the river with at least one mote deployed outside the river for injection of differential GPS correction values. This mote or a second node outside the river can be connected to a PC via USB for programming and data collection easily.

### 2.1. Power Supply and Energy Storage

RiverMote is supplied by solar cells and DLCs are used for energy storage. We supersede the need for additional hardware or software for MPPT by fulfilling the following constraints:

**2.1.1. Solar Cells.** Solar cells must be large enough to provide enough energy per day for a mote DC that is capable of providing the end-user performance that is needed by the application. Their maximum output voltage must be considered when designing protection circuits and their MPP must be considered when selecting suitable voltage thresholds and hysteresis for the storage structure and its charging circuits. Cost is an issue as well.

**2.1.2. Charging.** Input and output charging, load matching and protection circuits must be designed with efficiency high enough that do not annul working power state and energy efficiency models (PSM and EEM).

**2.1.3. Energy Storage.** The storage structure itself must be capable of storing enough energy to bridge a full modeling period and EHD pattern variation. Storage structure and - if needed - balancing circuit leakage currents and efficiencies have to be considered for PSM and EEM. We design a serial DLC structure with balancing for achieving a long lasting hardware with a suitable input voltage range for meeting the solar cells MPP.

**2.1.4. The Load.** Finally, the mote part may vary DC ratio and period length for varying end-user performance and delay accordingly. We go for a slotted networking design where motes' radios are activated $2\,s$ each for sending and $28\,s$ each for listening. The closer a mote is to the base station the earlier its 2 seconds transmit slot will be. The slotted design makes the system more robust due to its simplicity [16] and there is enough space in time to counteract timing misalignments due to temperature drifts and other issues [15].
Although, batteries have more capacity compared to the DLCs they have limited lifetime and are temperature dependent. During evaluation of RiverMote its temperature varies for approximately 50 degrees each day. Furthermore, DLCs have more than 500000 charge-discharge cycles and a nominal lifetime of 10 years where NiMH batteries have a lifetime of only 500 cycles before serious capacity loss. In addition to that, EHS-enhanced WSNs rely on accurate state of charge measurements. Using batteries would make the state of charge measurement overly complex.
Energy harvested during the day must be stored in the DLCs for continuous operation at night as well. The needed capacity depends on the power consumption of the overall hardware architecture. Furthermore, a hysteresis is implemented in hardware that will only switch on the mote if enough energy can be offered for full hardware and network initialization. From that point on, the DLCs can constantly be drained by a fully operating mote before the lower bound of the DLC's voltage operating range is reached for a duration of worst case initialization time of the system. The time is lower bounded by the time for initial GPS configuration - afterwards the configuration is stored persistently without further significant power dissipation until a possible hardware reset. We have selected the lower bound of initialization time such that it is above the motes' DC

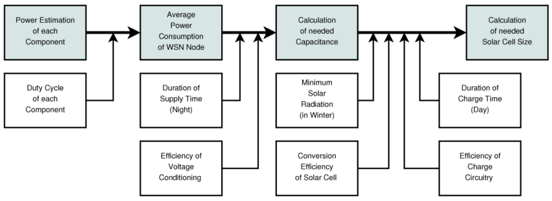



Figure 3. Calculation methodology of the DLC capacitance and the size of the solar cell.

period to ensure proper initialization and network joins via its ZigBee module. This is a key idea of RiverMote's robust design. Even though we cannot imagine a scenario where the storage structure would be trained below its operating range after being initialized once, severe programming errors might do the job of completely draining the system. In such a case the hardware back-off solution with the hysteresis would allow to re-initialize a RiverMote and restart its operation by joining the network with the process described above again. Back to dimensioning the storage architecture, the energy stored in $n$ capacitors can be calculated with:

$$E = \frac{1}{2} \cdot C \cdot U^2 = \frac{1}{2} \cdot C \cdot n \cdot U_C^2.$$

Additional energy of each additional DLC can be traded for power lost and cost of a further balancing circuit.

### 2.2. Power Estimation

Table 1 shows a list of RiverMote's components and their expected power dissipation for implementing the application scenario. Furthermore, Figure 3 gives an overview of the overall power dissipation calculation process presented here that incorporates the values from Table 1. The components

| Component | $t_a$ [s] | $P_a$ [mW] | $P_{sl}$ [mW] | $P_{avg}$ [mW] |
|---|---|---|---|---|
| MCU | 30 | 15.0 | 0.1 | 0.60 |
| Send | 2 | 429.0 | 0.0 | 0.95 |
| Receive | 28 | 82.5 | 0.0 | 2.57 |
| RxTx Sleep | 30 | 0.0 | 0.1 | 0.10 |
| GPS | 30 | 120.0 | 0.1 | 4.10 |
| Pressure | 1 | 250.0 | 0.0 | 0.28 |
| Ultrasonic | 1 | 10.0 | 0.0 | 0.01 |
| Other HW | 30 | 30.0 | 0.1 | 1.10 |
| Total $P_{tot}$ | | | | 9.70 |

Table 1. The MSP430F1611 and MRF24J40MB support TinyOS compatibility and ZigBee for more than $1\,km$. The GPS Fastrax UP300 supersedes repeated costly setups by an internal backup.

are active for $t_a$ seconds each activation interval $T$ which leads to their duty cycle $D$ with an average power dissipation of $P_{avg}$. It can be summed up to $9.7\ mW$. For further

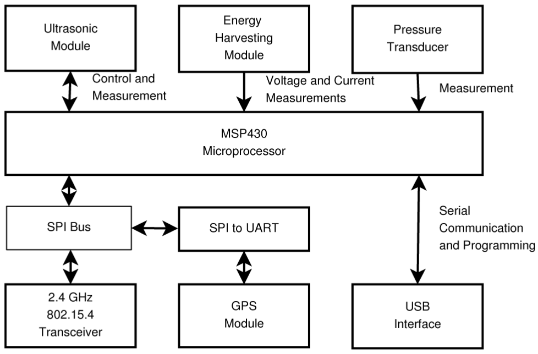



Figure 4. Hardware modules of the mote and the communication between them.

calculation a higher value of $P_{tot} = 12\,mW$ will be assumed. During full operation mode, all components are active and the EHS must be capable of supplying $P_{max,full}$ which in turn allows to determine the needed capacity $C_{real}$ and total average power $P_{tot,max}$ for a given efficiency $\eta_{convert}$:

$$I_{max,full} = \frac{P_{max,full}}{U_{sup}} = \frac{854\,mW}{3.3\,V} = 258.8\,mA.$$

$$E_{supply} = P_{tot} \cdot t_{night} = 12\,mW \cdot 16\,h = 691.2\,J$$

$$E_{needed} = \frac{E_{supply}}{\eta_{convert}} = \frac{691.2\,J}{0.8} = 864\,J$$

$$C_{real} = \frac{2 \cdot E_{needed}}{U_{start}^2 - U_{stop}^2} = \frac{2 \cdot 864\,J}{(5\,V)^2 - (1.8\,V)^2} = 79.41\,F$$

$$E_{usable} = \frac{1}{2} \cdot C \cdot \left(U_{start}^2 - U_{stop}^2\right) = 1686.4\,J$$

$$P_{max} = \frac{E_{usable} \cdot \eta_{convert}}{t} = \frac{1686.4J \cdot 0.8}{16h} = 23.42\,mW$$

$$P_{leakage} \leq I_{leakage} \cdot U_n = 0.45\,mA \cdot 5\,V = 2.25\,mW$$

$$P_{tot,max} \geq P_{max} - P_{leakage} = 23.42 - 2.25 = 21.17\,mW$$

## 3. System Implementation and Setup

The system setup consists of mote and EHS implementation. Figure 4 depicts the mote's components. The main component of the mote is the microcontroller: the MSP430F1611 low-power MCU. The GPS receiver needs an extra serial interface, so an SPI-to-UART adapter is used. The 2.4GHz 802.15.4 transceiver module is connected to the SPI as well for ease of communication. Ultrasonic module and EHS are directly connected to the MSP430. The USB module is implemented for programming and communication features as known from other mote platforms as well.

Figure 5 shows a block diagram of the EHS part. The

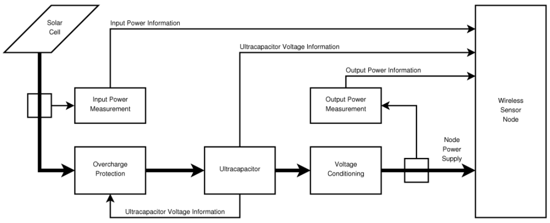



Figure 5. Design of the energy harvesting hardware. The thick arrows illustrate the flow of power and the thin ones illustrate the flow of information.

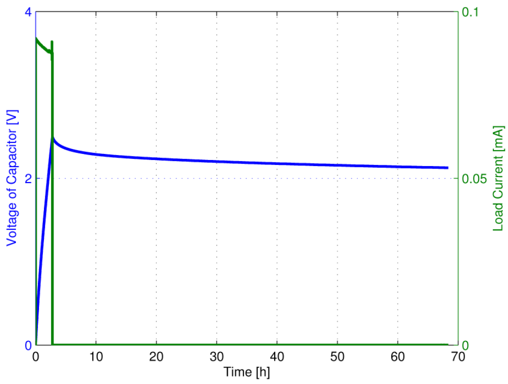



Figure 6. Long-term measurement of a single BC310 P250 T10 from Maxwell with a capacitance of $C = 310\,F$. The voltage of the DLC and the charging current have been measured for nearly 3 days. After charging the DLC to a voltage of $2.5\,V$, the power supply has been disconnected.

solar cell is a voltage limited current source. No voltage conditioning is needed for the DLC storage structure due to appropriately selected components for high efficiency at low cost. The nominal solar cell voltage is a little higher than that of the dual-DLC structure. Overcharge protection guarantees a maximum voltage at the DLCs. Other power will be bypassed. Voltage conditioning is used to generate a stable output voltage at $3.3\,V$. BC310 P250 T10 DLCs from Maxwell have been chosen for the energy storage structure. From experiments with the platform from [4] we know that self-discharge of DLCs can be a limiting factor. Therefore we have performed an initial test of the DLCs self-discharge. Results that are depicted in Figure 6 show that despite an initial drop of its voltage the DLC performs well. For selecting proper DCs, current and voltage measurement lines have been implemented in the EMU on-board. The energy budget and input/output power can accurately be determined.

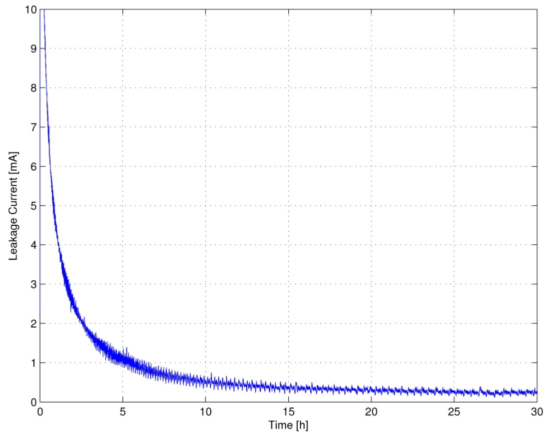



Figure 7. Leakage current of the DLC.

## 4. Results

For the evaluation of BOS characteristics and perpetuity constraints, we first evaluate single DLCs' leakage. Then, we introduce balancing circuitry and characterize self-measurement capabilities. Next, the overall leakage is characterized and a long-term test will be used for validating the system concept and hardware functionality.

### 4.1. Capacitor Leakage and Balancing

The National Instruments®DAQ NI PXI-6221 from [14] is used for measuring the leakage current of single DLCs. Two measurements are performed that show how a DLCs leakage depends on its clamp voltage history. First, a DLC is charged to its maximum capacity. Then the current source is detached and the leakage measurements start instantly. The second measurement keeps the maximum voltage for one day before performing leakage measurements. The DLC is charged with $E_{supply} = 1246.62\,J$ with an average current of $89.25\,mA$ for $2.8\,h$. Figure 7 shows the DLC leakage current after charging the DLC and disconnecting the power supply:

$$I_{leakage} = C \cdot \frac{\Delta U_{cap}}{\Delta t} = C \cdot (U_{cap,n+1} - U_{cap,n}) \cdot samp\ rate.$$

It can be seen that a lot of energy is lost in the beginning and leakage reduces over time to an average of $0.487\,mA$ which is slightly above $0.45\,mA$ from the datasheet. However, the significant value is to be determined over the first 16 hours after detaching the power supply as this time is to be bridged with the DLC. The average leakage current during this interval was $1.398\,mA$.
The second experiment shows that the DLC is more stable after keeping the clamp voltage constant for one day.

The average leakage current during the first 16 hours is $0.121\,mA$ which is below a third of the value of the data sheet. While the real value will be somewhere between $1.398\,mA$ and $0.121\,mA$, we conclude that the process of BOS measurements cannot completely be done offline. We rather suggest and evaluate the process of offline characterization of significant scenarios and then use online curve fitting algorithms of low computational complexity. A self-complete offline characterization with highly nonlinear EHD patterns, single DLC leakage characteristics, capacitor balancing energy loss, other hardware components' leakage and not to forget variable component's activities would be too complex.

The balancing circuit is needed to keep both DLCs at the same voltage. This is important for not overcharging the DLCs due to unbalanced charging with only one combined overcharge protection for both of them. The average balancing current increases nearly linearly with $I_{bal,meas,0.2V} = 10\,mA$ at a voltage difference of $0.2\,V$. For up to 20% misalignment of the charging process among DLCs the minimum charging time and maximum charging current can be calculated with:

$$\Delta Q = \Delta U_{max} \cdot C_{DLC,1} = 0.3V \cdot 310F \cdot 0.8 = 74.4\,As.$$

$$t_{charge} = \frac{\Delta Q}{I_{bal,0.4V}} = \frac{\Delta Q}{I_{bal,meas,0.2V} \cdot 2} = \frac{74.4As}{20mA} = 3720\,s.$$

$$I_{charge} = \frac{Q_{tot}}{t_{charge}} = \frac{744As}{3720} = 200\,mA.$$

## 4.2. EHS Efficiency Evaluation

EHS efficiency and stored energy are:

$$\eta_{EHS} = \frac{P_{output}}{P_{input}}, E_{stored} = \frac{1}{2} \cdot C \cdot U_C^2.$$

Two thresholds are set for measuring EHS efficiency. DLCs are constantly supplied by a laboratory power supply and the GPS is used for draining them. The voltage thresholds are set in the middle of the operating voltage range: $4\,V$ and $3.5\,V$. Figure 8 shows two charge-discharge cycles of the EHS. Key results of efficiency measurements are listed in Table 2. It can be seen that the true voltage difference of the DLCs is only $0.40\,V$. The reason is the inaccuracy of the voltage measurement of the microcontroller at high mote currents. These high currents cause a higher voltage drop at the shunt resistor for the measurement and the shunt resistor of the EHS.

The resulting overall efficiency of $\eta_{EHS}$ between $79.11\,\%$ and $83.16\,\%$ is better than expected $(64\% for 80\% in\ 80\% out)$. To the best of the authors knowledge there exists no such an efficient EHS for WSNs without MPPT techniques. Even in case costly MPPT is used, the efficiency of a single stage could easily drop below 80%. Compared to that, already the output efficiency alone of the EHS in [4] has been characterized in [14] to vary for up to $50\,\%$.

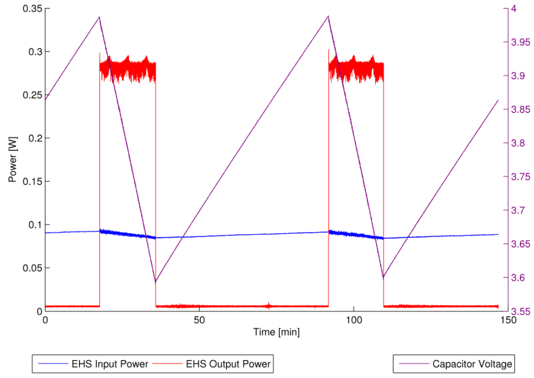



Figure 8. EHS efficiency measurement.

| Voltage level | low | middle | high |
|---|---|---|---|
| Maximum energy | $690.76\,J$ | $1233.10\,J$ | $1775.01\,J$ |
| Minimum energy | $520.91\,J$ | $998.52\,J$ | $1520.90\,J$ |
| Dynamic energy | $169.85\,J$ | $234.57\,J$ | $254.11\,J$ |
| Maximum voltage | $2.99\,V$ | $3.99\,V$ | $4.79\,V$ |
| Minimum voltage | $2.59\,V$ | $3.59\,V$ | $4.43\,V$ |
| Dynamic voltage | $0.39\,V$ | $0.40\,V$ | $0.36\,V$ |
| High-power period | $11.21\,min$ | $18.02\,min$ | $20.55\,min$ |
| Low-power period | $47.79\,min$ | $55.41\,min$ | $52.53\,min$ |
| Duty-Cycle | 0.23 | 0.33 | 0.39 |
| Average input current | $22.79\,mA$ | $21.57\,mA$ | $21.67\,mA$ |
| Average input power | $70.35\,mW$ | $88.07\,mW$ | $106.17\,mW$ |
| Average output power | $57.42\,mW$ | $73.24\,mW$ | $84.00\,mW$ |
| Total EHS efficiency | $81.62\,\%$ | $83.16\,\%$ | $79.11\,\%$ |

Table 2. Key results of EHS efficiency measurements. Results at middle voltages represent the results from Figure 8.

## 4.3. On-board and Long-Term Measurement

The error of the solar current counter is better than $5\,\%$ during daytime. Mote current measurements accuracy have evaluated to an accuracy of 5% or better for currents of $3\,mA$ or more. Despite the fact that the relative errors of on-board runtime measurements may increase for smaller currents their impact on the energy budget are negligible small and especially magnitudes of order below DLC leakage. Figure 9 shows self-test debug information (sent every $10\,min$) of RiverMote2 during moderate to really bad weather days during an Austrian winter. Other motes' and other measurements' plots are not shown for brevity, but all long-term tests have validated ENO capabilities. The GPS has always been switched on when the mote was activated. Ultrasonic transceivers have only been used for tests in the water. The

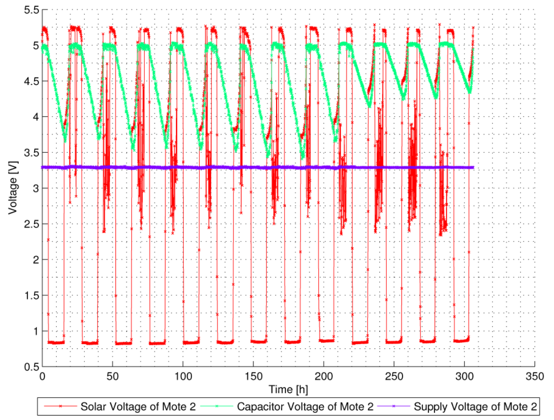



Figure 9. Measured voltages of RiverMote2 during the long-term test.

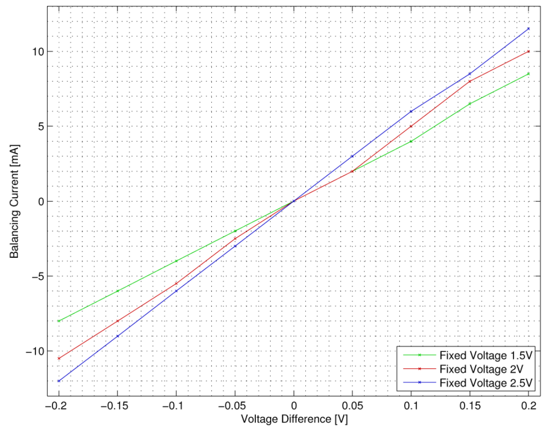



Figure 10. Measurement of the balancing current depending on the voltage difference of the DLCs.

results on DLC leakage, EHS efficiency, power measurement accuracy and long-term experience have shown a low cost platform. What is left open is the overall system leakage and standby and error-reporting strategies when it comes to blackouts. GPS and over sensors' characterization will be left out for brevity.

RiverMote is of similar cost as TelosB nodes without EHS support: PCB ($45.80\ EUR$) + HW ($92.79\ EUR$) + DLCs ($30.84\ EUR$) + solar cell ($\approx 50.00\ EUR$) + GPS module ($46.86\ EUR$) = Total cost $266.29\ EUR$.

### 4.4. Blackout Sustainability

BOS characteristics depend on DLC leakage and the EHS components' efficiencies. Especially, the performance of the DLC balancing mechanism has to be taken into account. Figure 10 shows results of the balancing current evaluation. The balancing current limits the maximum charging current of the system. For a charging process that is starting with empty DLCs, the average charging current should not exceed $200\,mA$. However, this value is the limit for a complete charging process while assuming a pretty high voltage difference of $0.2\,V$ of the DLCs. Therefore, the maximum charging current increases and the selected solar cells can be used.

Although DLC leakage and the balancing circuitry have carefully been evaluated, results cannot directly be mapped for completely describing RiverMote's overall leakage and BOS. Therefore, the system has been evaluated for its overall leakage and BOS after a charging process with solar cells attached. RiverMote 2 has been charged to $4.40\,V$ by a solar cell and RiverMote 3 has been charged to $3.55\,V$. For assuming harsh conditions, the system leakage measurements are started immediately after the charging process with no

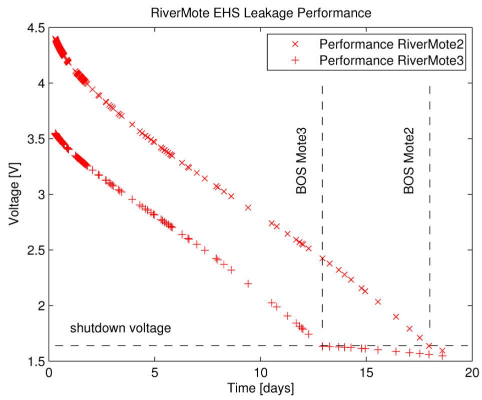



Figure 11. System Voltage over time during a complete blackout.

stabilization of the DLCs.

We use curve fitting techniques and extrapolation of achievable BOS. Figure 11 shows results of the overall EHS performance including DLC leakage, EHS leakage and circuit efficiency. The EHS performs well with several days of BOS in both cases. At a shutdown voltage the EHS completely shuts off the drain that comes from the EHS itself. It then waits until the energy budget has relaxed for supplying a new system initialization process including GPS initialization and to then join the network again.

While Figure 11 shows the result of the two tests alone, Figure 12 depicts the complete BOS characteristic curve of the platform. It is still based on the measurements with harsh conditions with starting the black-out experiment immediately after the charging process. Results of the experiments

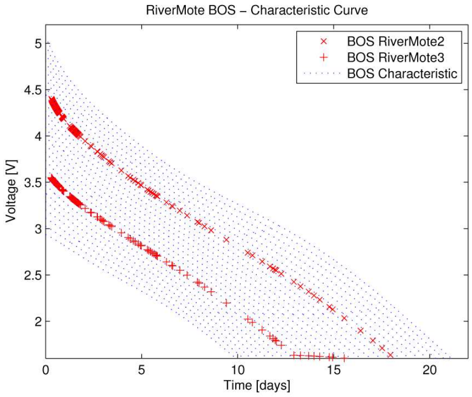



Figure 12. The RiverMote platform BOS characteristic.

have been plotted as a reference as well. It can be seen that the BOS can be up to 3 weeks for a fully charged system. In case the system might know about an approaching black-out period, the load could be reduced for stabilization of the DLCs and even longer BOS. The local information on how long RiverMote can sustain a black-out can locally be computed with locally evaluating the characteristic curve generating function. This relaxes the amount of energy that is needed for determining the BOS and possibly sending an error message to a negligible small amount of the overall energy budget.

## 5. Conclusion and Future Work

We have presented RiverMote: a perpetual EHS-enhanced WSN platform for river level monitoring. It is a highly cost- and energy-efficient platform designed with an MSP430, EHS and accurate on-board measurement capabilities and its extendable design has been evaluated with integrated GPS and ultrasonic transducers. The paper concentrates on evaluation of the energy storage structure with respect to ENO, EEM and BOS capabilities. It improves over state-of-the-art solutions in that it can operate perpetually at more than 80% efficiency and more than 3 weeks of sustainable energy black-outs. Several findings in terms of energy management deserve further analysis and they enforce the need for a commonly accepted metric for reliable EHS design.

## References


[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *Tech. Rep. IITB/CSE/2008*, dez. 2008.

[3] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, p. 32, 2007.

[4] P. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in wsns," in *IFIP/IEEE Wireless Days*, nov. 2008, pp. 1 –5.

[5] C. Trummer, "Device evaluation for energy harvesting devices for higher class tags," *Graz University of Technology, Institute for Technical Informatics, project thesis*, 2006.

[6] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 223–234, 2004.

[7] N. Jaggi, K. Kar, and A. Krishnamurthy, "Near-optimal activation policies in rechargeable sensor networks under spatial correlations," *ACM Trans. Sen. Netw.*, vol. 4, no. 3, pp. 1–36, 2008.

[8] C. Bergonzini, D. Brunelli, and L. Benini, "Algorithms for harvested energy prediction in batteryless wireless sensor networks," in *IEEE International Workshop on Advances in Sensor and Interfaces*, jun. 2009, pp. 144 –149.

[9] C. Moser, J.-J. Chen, and L. Thiele, "An energy management framework for energy harvesting embedded systems," *J. Emerg. Technol. Comput. Syst.*, vol. 6, no. 2, pp. 1–21, 2010.

[10] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time scheduling for energy harvesting sensor nodes," *Real-Time Syst.*, vol. 37, no. 3, pp. 233–260, 2007.

[11] F. Regan, A. Lawlor, B. Flynn, J. Torres, R. Martinez-Catala, C. O'Mathuna, and J. Wallace, "A demonstration of wireless sensing for long term monitoring of water quality," in *IEEE Conference on Local Computer Networks*, oct. 2009, pp. 819 –825.

[12] J. Zhou and D. De Roure, "Floodnet: coupling adaptive sampling with energy aware routing in a flood warning system," *J. Comput. Sci. Technol.*, vol. 22, no. 1, pp. 121–130, 2007.

[13] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, and S. Shenker, "An architecture for energy management in wireless sensor networks," *SIGBED Rev.*, vol. 4, no. 3, pp. 31–36, 2007.

[14] P. Glatz, L. Hormann, C. Steger, and R. Weiss, "A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency," in *IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing*, mar. 2010, pp. 468 –473.

[15] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 1–32, 2010.

[16] R. Flury and R. Wattenhofer, "Slotted programming for sensor networks," in *ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 24–34.

2011 18th International Conference on Telecommunications

# MAMA: Multi-ApplicationMiddlewAre for Efficient Wireless Sensor Networks

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics, Graz University of Technology, Austria*
$\{Philipp.Glatz,\ Leander.Hoermann,\ Steger,\ RWeiss\}$@*TUGraz.at*

## Abstract

*Wireless sensor network (WSN) motes are resource constrained devices. This is due to optimizations tailoring them towards application-specific and cost-efficient scenarios and setups. Many to most of these optimizations are built upon power aware aspects and dependability measures due to the fact that battery technology still evolves quite slowly and WSN's inherent redundancy is suitable for fault-tolerant schemes. What we do here is implementing a programming paradigm where several applications may be distributed heterogeneously among different subsets of a WSN's motes. Multiple applications can be run side by side on the same mote. This is far more efficient than deploying hardware for each and every application. The Multi-Application MiddlewAre (MAMA) layer allows running different TinyOS applications at the same time easily. Concurrent sensor usage and messaging is handled by MAMA. Results from simulation runs, hardware deployments, power profiling and scope measurements show good scalability, only little timing and memory overhead and power efficient properties.*

## 1. Wireless Sensor Network Programming Paradigms

We give a short outline of what wireless sensor network (WSN) programming paradigms are, what multi-application approaches exist for WSN platforms, and how the novel Multi-Application MiddlewAre (MAMA) can disburden the programmer. It utilizes resources efficiently on a per-mote basis and on a per-network basis. Radio and sensor access times are shared locally while end-to-end communication channels and mote hardware instances are shared among applications globally. MAMA implements a concept which is called virtual organizations (VOs) in grid computing [1]. WSNs [2] consist of an often scalable number of devices of small form factor that are called motes. A recent survey proves that the field of WSNs has matured in several aspects and has made its way into a number of technical committees' agendas and is used in different applications [3]. Though, the opportunities offered by WSNs may often be unused due to the diversity of different hardware and software approaches as outlined in [4]. To not necessitate having an application developer learn a novel dedicated approach each time a new problem has to be solved, programming paradigms can be used which are usually implemented by middleware layers abstracting away unwanted complexity with several examples given in [5]. Novel paradigms are often tailored towards simplicity and ease-of-use with slotted designs [6], especially when it comes to robustness issues where single faults may heavily impact end-user performance [7].

Another important aspect is that many issues with mote design are well understood today. Firstly, this allows to come up with industry-strength reliable systems [8]. Secondly, hardware architectures are likely to not change too rapidly anymore. Different directions have been explored since the idea of smart dust [9]. Provided that situation, it is valid to pursue the idea of multi-application environments where VOs can coexist in virtualized environments with no need to resolve conflicts (i.e. when accessing resources) on their own.

This is what MAMA does. It is an environment where multiple VOs can run their applications concurrently. So, available bandwidth, coverage, sensor systems and computational power may better be utilized. There is no need to per se redevelop and replace WSN hardware or to deploy additional motes for deploying yet another single application.

Though MAMA behaves similar to the card manager in smart cards for running with multiple VOs, we still have to take into account that we are dealing with WSNs. Therefore, we discuss related operating systems and methods for reprogramming WSNs for comparing MAMA to other middleware approaches in Section 2. We also give an overview of techniques and operating systems (OSs) that are capable of running multiple applications in WSNs. In Section 3 we present the concept behind MAMA itself and ways of using it as well as implementation issues including setups for evaluation. The evaluation using Avrora [10], Mica2 motes and the TOSPIE2 [12] measurement setup is discussed in Section 4. Finally, Section 5 will conclude from the results how MAMA can impact WSNs and its applications running on TinyOS 2 in general. It will also

978-1-4577-0024-8/11/$26.00 ©2011 IEEE

978-1-4577-0024-8/11/$26.00 ©2011 IEEE 

give an outlook on plans for an energy- and power-aware management console for efficient task scheduling which is part of ongoing work as a follow-up of the project [11] implementing the basis of MAMA.

## 2. Operating Systems with Multi-Application Support

There are many approaches to implement multi-application support at OS-Level. In this paper we briefly outline OSs with binary loadable modules like Contiki [13] and SOS [14]. We discuss TinyOS [15] and some work to implement multi-application support for this popular OS, like Deluge [16] and TinyOS threads [17].
Contiki [13], SOS [14] and TinyOS threads [17] are using cooperative threads to implement multi-application support.

### 2.1. Contiki

Contiki [13] is an event-driven OS. It implements dynamically loadable binary modules and protothreads for its multi-application support. However, this thread implementation is unable to support local variables. This produces an overhead, when it comes to using variables in an application. Also, all variables of the applications are accessible from each application [18]. MAMA overcomes these limits.

### 2.2. SOS

SOS [14] consists of a statically built kernel. The rest of the system and applications are implemented as modules. These modules can dynamically by installed on a mote. The kernel provides services such as dynamic memory allocation, software timers, sensor manager and high level I/O interfaces [19].

### 2.3. TinyOS

TinyOS [15] is a light-weight and event-driven OS for WSNs. It uses statically built images running on motes. In its latest version - TinyOS 2.x - it supports a three-level hardware abstraction. This brings a simple way to use the same application in a heterogeneous network. The component-based architecture of TinyOS and the programing language nesC [20] give programmers a new paradigm to implement their applications.

### 2.4. Deluge

Deluge [16] is based on TinyOS and provides a system to load programs for a mote over the network. Since TinyOS only supports static images to run on the $\mu C$, it must send large data (up to 30 or 40kB) through the network. Deluge implements a small bootloader for TinyOS to load and start different images stored on the mote. A limitation is that loading and starting an image in Deluge, disrupts ongoing applications and reboots the mote [19]. Deluge also provides a networking protocol to transmit its large program images over the network.

### 2.5. TinyOS threads

TinyOS threads [17] are an extension to the concurrency of TinyOS. It is implemented since TinyOS 2.1 and provides interfaces to start, stop, create and suspend threads. The mote is split into a kernel-thread, that works in an event-driven manner, and user-threads. These are implementing the applications and working with blocking I/Os. Each thread has its own memory-space, its own set of registers and its own stack. All that is saved at a context switch. Using a blocking I/O invokes a system call. System calls are TinyOS components that implement one specific I/O like radio access, sensors and timers.

As can be seen in Figure 1 TinyOS threads can be in five


Figure 1. States and possible transitions of TinyOS threads.

different states. When creating a thread it will come to INIT. It will stay in this state as long as it takes to register it in the scheduler. After this is done, the scheduler transitions the thread into the READY state. Here, the thread waits until it is its time to be executed. Then it transitions into the ACTIVE state. In this state, all its registers are loaded into the $\mu C$, and the thread runs in its own memoryspace. If the thread calls a blocking I/O or if it is waiting for an event to occur, it is in the SUSPEND state. The STOP state is entered when it has finished its work.

Figure 2 shows the timing of a system call. Two VOs are

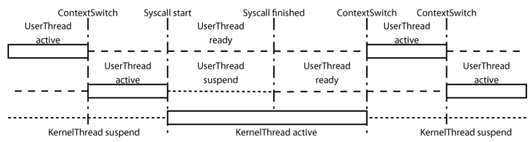



Figure 2. Timeline of a system call. Dashed lines are the time when a thread waits in the ready-queue. Dottet lines are the time when a thread is suspended.

running on a mote. A time-sliced scheduler, like the one shown, is implemented in TinyOS threads. The first context switch occurs when the time for thread A is used up, then thread B is active and A waits in the ready-queue. Thread

B uses a blocking I/O. So it calls the specific system call and is put into the SUSPEND state by this system call. From here on, the systems works in kernel-space. The kernel starts reading the sensor and after a while the sensor signals that it has finished. Then the kernel puts thread B back on the ready-queue and into the READY state. Now, both threads are on the ready-queue and waiting for execution. In this example the kernel has something more to do, so it stays active. After it has finished, the kernel suspends it and the next thread on the queue gets active. After its execution time the scheduler switches threads again and thread B returns from reading the sensor.

## 3. Multi-Application Middleware Setup

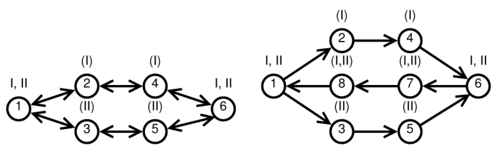
Applications running on MAMA have to implement the interface shown in Listing 1 and wire it to the Application-Wrapper component.

**Listing 1** The application interface

```
interface Application{
  event error_t init();
  event error_t run();
  event error_t exit();
}
```

These three events are signaled from a function that is embedded in a TinyOS thread. `init` is the initialization of the application. This thread is created with a command from MAMA. `run` is the body of the infinite loop, and `exit` is the event signaled, when the application has to stop. The application can be stopped with a command from MAMA.

### 3.1. Function and Scalability Test




(a) The first test case (b) The second test case

Figure 3. Two throughput test setups have been implemented where both applications ping node 6 from node 1 with increasing packet injection rates for bandwidth characterization.

Figure 3 shows two different test cases that will be used for bandwidth and delay characterization of MAMA. In both test cases nodes 2 and 4 (route of application I) only have a message buffer of size 2. All other nodes have a buffer size of 10 messages.

In the first test case MAMA sends packets from application I from node 1 via node 2 and 4 to node 6 and on the same route back to node 1. Application II sends from node 1 via node 3 and 5 to node 6 and on the same route back to node 1. This architecture and a second one will be profiled for bandwidth utilization.

In the second test case MAMA uses nodes 7 and 8 to route the packets back from node 6 to node 1 for both applications. Nodes 2 to 5 (and in test case 2 also 7 and 8) are programed with MAMA without any applications.

Since this test will be used with real hardware (Mica2 motes) and used in simulation (Avrora instruction set simulator [10]) complete overhearing is applied. This means that every node is in radio-range of each node, but MAMA only sends messages that are addressed according to the setup described above. This delays the sending process, but tests with real hardware can easily be repeated with the same setting.

### 3.2. Networking Implementation

Figure 4 depicts the Network component that is instantiated once in the kernel thread (solid box) including send and receive queue built upon a common message pool. All applications are allowed to access the simple interface concurrently. Incoming messages are queued in a first-in first-out manner in the sending- or receiving-queue. Messages overwrite the oldest message in case all places are taken.

The router decides where to put the incoming message.

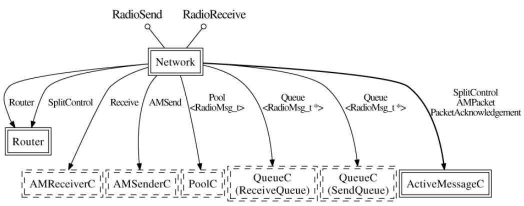



Figure 4. The Network component implemented in kernel space offers a simplified send and receive interface using TinyOS ActiveMessage components.

In case the message is for this mote, it will put it on the receiving-queue. In the other case, it is for another mote, it will put the message to the sending-queue.

Messages on the receiving-queue are signaled to the RadioReceive component, which looks for a registered application to receive this message. If it finds one, the system call will be finished, if not, the message will be discarded.

Messages on the sending-queue are processed by a task. This sending task checks if the first message ist for this node. If that is the case, the message will be enqueued on the receiving-queue. If not, the router is asked for the next hop to the destination of this packet and the message is sent via the ActiveMessage components of TinyOS.

The Routing component implements the whole routing

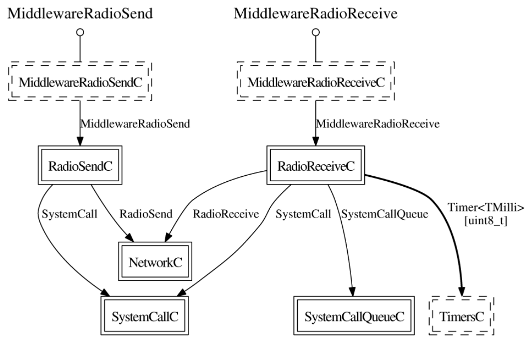



Figure 5. The middleware arbitrates system calls and the Network component.

algorithm. It only has to implement two services for the Network component, `isForMe` and `nextHop`.
Figure 5 shows the middleware networking API offering an intuitive receive and send interface. The generic component (dashed box) is virtualized including the according system calls on a per-instance basis. While only one system call for sending can be run at a time, all applications running on a mote can register for listening to incoming messages at the same time. The sending system call only builds and enqueues the message and therefore returns instantly.

**Listing 2** Application I for the stress test setup

```
19 event error_t Application.run(){
20   time = 500 - (msg.data / (25)) %
       480;
21   //every 25 pkgs 1ms less time down
       to 20ms
22   msg.data++;
23   call MiddlewareRadioSend.send(6, 1,
       &msg, sizeof(msg));
24   //call Leds.led0Toggle();
25   call GeneralIO.toggle();
26   return call System.sleep(time);
27 }
```

### 3.3. Test Case Implementation

Listing 2 shows the application that is used for profiling the stress test set up on Node 1 as shown in Figure 3. This application sends a 32bit counter from node 1 to node 6. The value of the counter defines the time this application waits until it sends the next message. Lines 20 and 22 calculate the sending delay decrement for the stress test application. Line 23 utilizes the middleware send interface queuing a message to be sent to application I on node 6. When the call returns PW0 is toggled with the call in line 25 and put to sleep for the calculated time. The status of PW0 is used for debugging and analyzing as seen in Section 4. Detailed timing information will be deduced from this debug information from hardware experiments.

**Listing 3** Appliction on the receiving node in stress test

```
19 event error_t Application.run(){
20
21   if (call
       MiddlewareRadioReceive.receive(&msg,
       0)== SUCCESS){
22     call MiddlewareRadioSend.send(1,
         1, call
         RadioPacket.getPayload(&msg),
         call
         RadioHeader.getlength(call
         RadioPacket.getHeader(&msg)));
23     call GeneralIO.toggle();
24   }
25   return SUCCESS;
26 }
```

Listing 3 shows application I on the receiving node (node 6 in Figure 3). This part of the stress test receives the messages sent by node 1. It takes the length of the message and the payload from the received message and sends it back to node 1. Then it toggles an output pin for debugging purpose.
Line 21 calls a receive with timeout 0. This registers a system call for the instance that will last until it returns due to a decision of the Routing component accordingly. Specifying a timeout other than 0 in ms will discard the request after a specified time or the call returns if a packet arrives. A GPIO is toggled as debug output that can be profiled with the measurement setup from [12].

## 4. Results

We compare different applications programmed with MAMA to similar applications that are implemented in TinyOS alone and TinyOS with using TinyOS- threads. We also present the results from simulations of the stress tests and the same tests on mote hardware.

Empty
: is taken from the TinyOS and threading examples and runs an empty loop for the middleware implementation where significant overhead can be seen

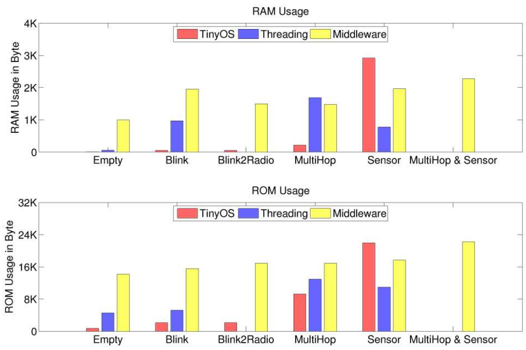



Figure 6. Overhead of MAMA seen in RAM and ROM of a Mica2 mote.

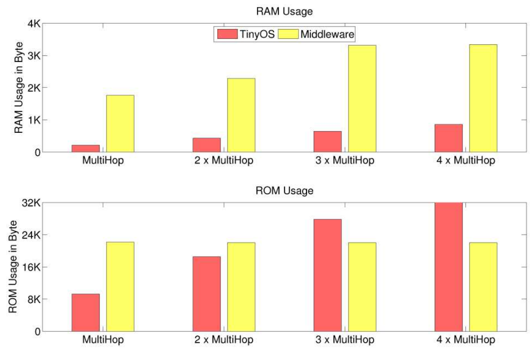



Figure 7. Comparison of MAMA and TinyOS for more than on instance of MultiHop on a mote.

for the empty middleware-based application. The routing module that is implicitly set up in kernel space causes this overhead. This overhead is to be considered every time MAMA is used.

Blink
: runs three timer components for TinyOS, three threads with another LED each and a middleware with three application instances with a single LED each. The RAM usage is impacted by the number of instances.

Blink2Radio
: has been used as the Count2Radio application from the TinyOS source tree and Blink2Radio with a dynamic shortest-hop routing for the middleware.

MultiHop
: is the same as Blink2Radio but for multiple hops. Threading has been implemented here as well. Here, we see the advantage of MAMA. The multi-hop variant needs hardly any more RAM and ROM than the single-hop application.

Sensor
: uses the TinyOS Oscilloscope multi-hop application, the threading Oscilloscope and a middleware application sensing and sending in a multi-hop fashion each 300 ms. Here the middleware needs less resources. Overhead can be conserved. The heavy memory footprint of TinyOS in this example is caused by its bulky routing protocol.

Blink2Radio & Sensor
: combines both applications on one node for the middleware which is not possible without using MAMA. This example shows that only little overhead occurs for adding both instances together compared to the deployment of 2 nodes in case of TinyOS or the sum of memory used in case of using TinyOS threads natively.

The overhead in RAM usage comes from the static allocation of memory for the stack used by the applications in MAMA and the allocation of the memory for the message buffer (10 x am_message_t). This can be up to 1.5 kByte.
As seen in Figure 7 this overhead comes with every application. But, when running more applications (or instances of on application as in Figure 7) we can see the benefit of MAMA. In the last case in Figure 7 (4 x MultiHop) on one single mote there are running 8 applications (4 x Sender and 4 x Receiver) independently from each other.
All examples have been run on Mica2 mote hardware. The number of VOs that can make use of one single mote is only limited by the available memory. The Blink and Blink2Radio applications have been tested on Mica2dot as well which are quite limited in their memory. Still, several VOs can be run concurrently. So, there is no need for additional motes and finally, this saves money for hardware and reduces collisions on the channel or reduces the need for synchronization.
Other than in SOS [14], a sensor application in MAMA only needs less than 20 lines of code to implement a periodic reading of a sensor and send it to a base station. On one side, this is convenient for the programmer. On the other side, errors are less likely to occur. In MAMA dynamic loading of programs or modules is not implemented yet, but it provides interfaces to start and stop applications at runtime. Since SOS has been implemented before TinyOS 2.x and before TinyOS threads have been implemented, much of the HPL and HIL of the Mica2 implementation from TinyOS 1.x is being used.
Since TinyOS 2.x has its 3 layer abstraction of the hardware, MAMA could easily be ported to other platforms as well. Only the multi-threading capability of TinyOS has to be implemented. MAMA has been compiled and tested for Mica2 and Mica2dot motes and compiled and simulated for TelosB motes as well.

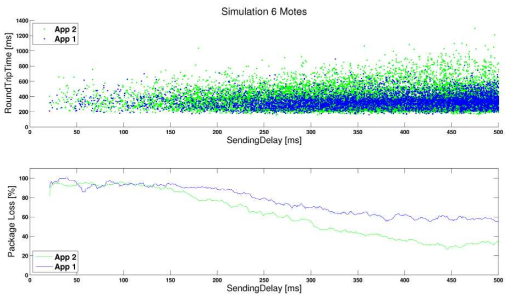



Figure 8. Simulation of package loss and round trip time with different delay of message injection for 6 motes. RTT and package loss are being profiled.

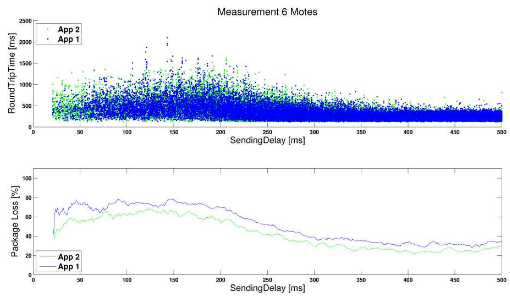



Figure 10. Profiling package loss and round trip time on hardware for 6 motes. Simulation results are being validated.

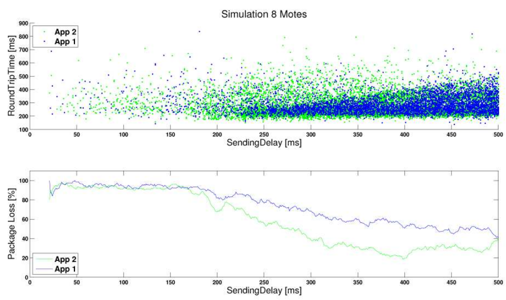



Figure 9. Simulation of package loss and round trip time with different delay of message injection for 8 motes.

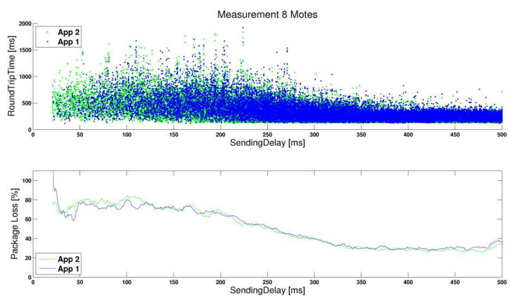



Figure 11. Profiling package loss and round trip time on hardware for 8 motes.

### 4.1. Stress Test Simulation Results

The 6 node setup in Figure 3 has been used to profile the results in Figure 8. Profiling results from Avrora show significant increase in package loss as the sending interval drops below 350 ms due to limited buffer size and full overhearing among all motes. Especially, application I instances - with a buffer size of 2 elements - suffer these problems. Still the overall results validate proper functionality of the MAMA layer and even utilizes the bandwidth more efficiently than needed for typical WSN monitoring applications. The round trip time (RTT) is the time from sending a counter value on node 1 to the receive event on node 1 of the message with the same counter value. The RTT is measured for each application running in the network. The RTT increases due to queuing more messages with less frequent message injection from reduced package loss rate. The rate is calculated on a per-message basis which means that packets that are not lost have traveled all the way from node 1 to 6 and back without failure.

Figure 9 shows the same behavior with scaled drop rate and delay characteristics due to more motes being involved which leads to more overhearing and more conflicts at the MAC layer. This effect validates the initial idea of overall resource awareness by putting more than one instance on the hardware instead of using multiple motes.

### 4.2. Stress Test Profiling Results on Mica2 Motes

Figure 10 shows results on motes similar to these from Avrora profiling. The RTT starts to increase at 300 ms as well as the drop rate. For injection intervals above 300 ms the performance allows to implement robust networks with reasonable effort. Figure 11 confirms the results as well.

### 4.3. Power Dissipation and Debug Profiling

Figure 12 shows profiling results of node 1 of a 6 node setup. The upper plot shows power dissipation and the lower plots depict debugging information from GPIOs. PW0 toggles if App1 has sent and PW1 for App2 accordingly. PW2 is ON between 'send' and 'sendDone'.

It turns out that visual inspection of the profiles tells a lot which would be hard to be modeled analytically or to be


Figure 12. Power dissipation and port debug measurement of mote 1 for test case 1.

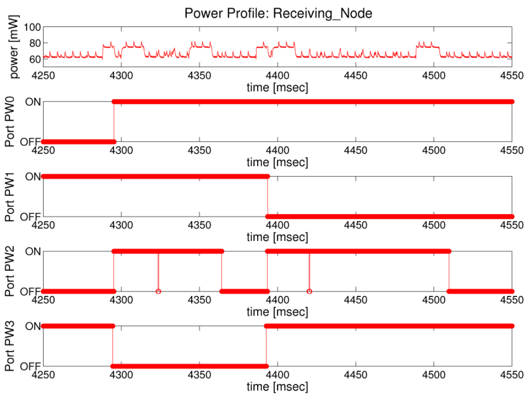



Figure 13. Power dissipation and port debug measurement of mote 6 for test case 1.

profiled with simulation and emulation environments. At 1990 ms a higher peak signifies a context switch and the PW0 and PW2 transition accordingly. The transmission is operated between 2012 and 2027 ms. Since no ACK arrives (no PW2 transition to OFF) until 2035 a resend happens from 2055 to 2070 ms. At 2075 ms the context is switched to App2 with the send until 2077 ms. Then, the same procedure is run for App2. From 2095 to 2105 a message is received. For node 6 the same type of profiling and hardware level debugging has been done as depicted in Figure 13. PW0 toggles for App1, PW1 for App2 and PW2 marks the sending system call being active again. PW3 toggles at the receive event. From 4290 ms at reception for App1, unpacking, repacking and transmitting takes until 4325 ms (with a resend afterwards) and gives an idea of the minimum delay that has to be accepted. Processing the packet on the processor and passing it from kernel space to user space and back takes 0.7 ms when using the MAMA layer.
As we see in Figure 12, a node needs about 50 mW when it is idle and up to 80 mW when it is sending or receiving a message. If we would build the same tests with native TinyOS applications, we need at least 2 more motes (1 sender and 1 receiver). The amount of messages sent will be the same, but there will be 2 more motes running idle. So, one would need more energy in total to operate these motes. This is another benefit of using MAMA.

## 5. Concluding Remarks and Outlook

This paper presents a multi-application middleware abstraction layer. It allows running multiple applications on a single mote. This allows efficiently utilizing available resources. The middleware encapsulates sensor resource arbitration and networking in a scalable way by means of a light-weight threading implementation. Networking capabilities have been profiled in detail.
Several sample applications from the TinyOS source tree have been implemented and shown to be working with only little or no overhead. Efficient memory usage, little delay and power efficient operation have been profiled with simulation/emulation, real Mica2 hardware and an accurate power dissipation and hardware level debugging setup.
Further steps include setting up a feedback for visualization of slotted programming capabilities and real-time feedback for the environment that runs the simulation/emulation and measurement setups as well. Also a detailed power analysis will be provided with policies to reduce the power consumption on motes that are operated in too power hungry idle modes.
A power management console is scheduled for the near future. This console allows an administrator to start and stop applications remotely depending on the energy available on the mote. This will make the system more applicable to adapt to varying energy harvesting devices' power profiles.

## References

[1] Foster, I., Kesselman, C., and Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int. J. High Perform. Comput. Appl. 15, (2001).

[2] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y.,Cayirci, E.: Wireless Sensor Networks: A Survey. IEEE Comp., 38(4), 393–422 (2002).

[3] Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw., 52(12), 2292–2330 (2008).

[4] Mottola, L., Picco, G. P.: Programming wireless sensor networks: Fundamental concepts and state of the art. ACM Computing Surveys, 2010.

[5] Molla, M. M., Ahamed. S. I.: A survey of middleware for sensor network and challenges. International Conference on Parallel Processing Workshops (2006).

[6] Flury, R., Wattenhofer, R.: Slotted Programming for Sensor Networks. IPSN (2010).

[7] Glatz, P. M., Loinig, J., Steger, C., Weiss, R.: A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks. MICC (2009).

[8] Gungor, VC and Hancke, GP: Industrial wireless sensor networks : challenges, design principles, and technical approaches IEEE Transactions on Industrial Electronics, (2009).

[9] Warneke, B., Last, M., Liebowitz, B., and Pister, K. S.: Smart Dust: Communicating with a Cubic-Millimeter Computer. Computer 34, (2001).

[10] Titzer, B. L., Lee, D. K., and Palsberg, J.: Avrora: scalable sensor network simulation with precise timing. IPSN (2005).

[11] Berlach, R.: Multiapplication Middleware for Wireless Sensor Networks. Project thesis, Institute for Technical Informatics, Graz University of Technology, (2010).

[12] Glatz, P. M., Steger, C., Weiss R.: Poster Abstract: TOSPIE2: Tiny Operating System Plug-In for Energy Estimation. IPSN (2010).

[13] Dunkels, A., Gronvall, B., and Voigt, T.: Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. LCN. IEEE Computer Society (2004)

[14] Han, C.-C., Kumar, R., Shea, R., Kohler, E., Srivastava, M.: A dynamic operating system for sensor nodes. MobiSys (2005).

[15] Levis, P.: TinyOS Programming, (2006).

[16] Deng, J., Han, R., and Mishra, S.: Secure code distribution in dynamically programmable wireless sensor networks. IPSN (2006).

[17] Klues, K., Liangy, C.-J. M., Paekz, J., Musaloiu-Ey, R., Govindanz, R., Levis, P., Terzisy, A.: TOSThreads: Extending the TinyOS Concurrency Model to Support Preemption. SenSys (2009).

[18] McCartney, W. P. and Sridhar, N.: Abstractions for safe concurrent programming in networked embedded systems. SenSys (2006).

[19] Balani, R., Han, C., Rengaswamy, R. K., Tsigkogiannis, I., and Srivastava, M.: Multi-level software reconfiguration for sensor networks. EMSOFT (2006).

[20] Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., and Culler, D.: The nesC language: A holistic approach to networked embedded systems. PLDI (2003).

# Design, Simulation and Measurement of an Accurate Wireless Sensor Network Localization System


Philipp M. Glatz
Institute for
Technical Informatics
Graz University of Technology
Austria
Philipp.Glatz@TUGraz.at

Christian Steger
Institute for
Technical Informatics
Graz University of Technology
Austria
Steger@TUGraz.at

Reinhold Weiss
Institute for
Technical Informatics
Graz University of Technology
Austria
RWeiss@TUGraz.at



## ABSTRACT

Wireless sensor network (WSN) motes are devices of small form factor. Tailoring WSNs towards a specific application domain leads to resource constraints due to limited use of hardware and especially bandwidth and energy. Furthermore, system aspects of interaction of different components and services lead to a number of further non-functional constraints and especially timing and memory issues.
Though several resources of related work exist that are dealing with localization for WSNs, there are no suitable approaches yet for how to optimize 3D localization accuracy with subject to conserving real-time aspects, memory footprint and power dissipation at the same time. It has also not yet been described how 1D, 2D and 3D errors each relate to one another in experimentation.
We present an approach that allows designing, prototyping and improving scalable localization systems for mote-class devices. We implement a localization system on networks of Mica2 motes for which the 3D position accuracy error has been validated to be below 30 cm in real world settings.
The paper presents ways to model, simulate, implement and measure the system model, used components' power dissipation and energy balance impact, timing behavior, memory footprint, robustness and accuracy. Statistical means of simulation results' accuracy are compared to real-world measurements' statistics.


## Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*network management, network monitoring*

## General Terms

Algorithms, Design, Experimentation, Management, Measurement, Performance




## Keywords

Energy-Awareness, Design, Localization, Measurement, Power-Awareness, Ranging, Simulation, Wireless Sensor Networks


## 1. DESIGNING WIRELESS SENSOR NETWORK LOCALIZATION SYSTEMS

Localization and positioning are useful to many types of networked applications. Application scenarios for localization systems are inherently given for wireless sensor network (WSN) applications. These may range from environmental monitoring and technical process control over user tracking up to ubiquitous mobile ad-hoc network (MANET) scenarios. The location of where WSN motes gather their sensor data needs to be remembered and MANET's entities' (maybe logical) position is needed for resource- and self-localization as well.
While these different application domains of localization systems may demand for different characteristics, we will develop and discuss an approach for designing and evaluating such a system when being implemented on WSN mote-class devices. Despite the fact that we will stick to this example throughout the paper, general ideas will be evaluated such that the approach can be applied to other technologies and methods as well.
The system that is to be presented is not just about being accurate, power-aware and bandwidth-efficient and of little overhead when being implemented as a 3-dimensional po-

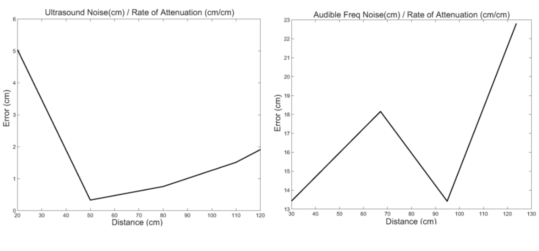



**Figure 1: Comparing the ranging error for different distances between the two acoustic frequency domains in [7]. Ultrasonic pulses may be more accurate, but a complete comparison including other properties is difficult. We analyze a novel power-aware localization system in detail.**

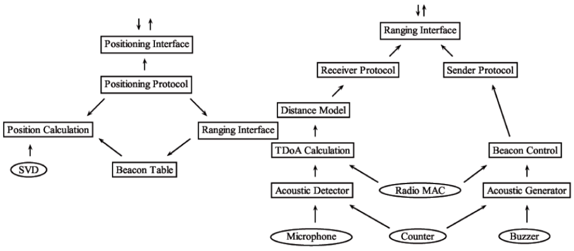



**Figure 2: Component diagram of the positioning and ranging process showing the interactions of the main components.**

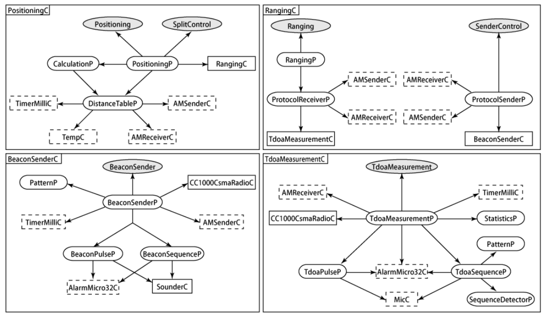



**Figure 3: Main nesC Components, apart from messaging services.**

sitioning service in a WSN middleware or application. It shall also explain and evaluate what it takes to develop such a system in a robust way with different constraints in mind.

## 2. RELATED WORK

Different types of WSN localization systems have been implemented and tested at the department. GPS-based localization gives a general purpose outdoor solution for the RiverMote [3] platform. RSSI-fingerprinting is used in [2] for indoor localization which turns out to result in quite noisy readings. And finally, a range based method using audible sound in [5] gives fine-grained indoor localization which will be discussed here in detail.

Different related technologies are possible to be used for localization. Thunder [6] claims to implement a zero-cost system for localization with audible sound, but this is valid only as long as the application scenario fits the setup that is used. Cricket [1] makes use of ultrasonic transceivers for ranging which can improve accuracy according to traces taken in [7] as shown in Figure 1. While this pays off with improved accuracy in several scenarios, there are also situations where transceiver directionality, power consumption, transceiver fade-in and fade-out effects and implementation issues may render the system cumbersome. The Bat system [8] even claims to be as accurate as up to 3 cm using ultrasonic distance measurements. We will find out later that it is difficult to estimate a location system's quality that way as well as defining a ranging method's accuracy or error bounds.

While the projects like the ones presented so far are all based on time-of-flight or time-difference-of-arrival (TDoA) based trilateration or multilateration methods, other technologies demanding for other computational models are possible as well. While a FM radio based system [9] is heavily concerned with channel estimation for noise reduction, radio tomography methods perform ongoing RSSI measurements and map radio attenuation to the position estimation procedure as in [10]. Proximity based methods using smart cards or different classes of tags up to smart dust may demand for a middleware layer moving the computational load to a powerful host due to complexity reasons. A well-known example of a completely different computational model with lots of computational resources needed is anchor-free localization [11]. Taking many ranging measurements of different motes into account at the same time for many different motes and then running iterative optimization procedures would overload platforms of small computational power.

A common problem to different fields of computer science is the question of how to select the best technology or approach. Different metrics may apply: accuracy, time to market, system cost, overhead for existing systems, usability, real-time constraints and many others. A single dimension that deserves consideration for all range-based methods is the accuracy of a single ranging operation. [7] compares two different technologies and their accuracy accordingly over different ranging distances. Though, a general conclusion can be drawn in that the ultrasonic measurements tend to be more accurate than the one with audible sound, other effects might apply as well. For instance, the different methods' power dissipation and real-time capabilities has not been taken into account, the different technologies suffer different constraints in directionality and distance and the error seems to vary for different distance in a nonlinear way. Different noise will apply and all these different aspects might differently impact a localization or positioning system based upon these range measurements.

What we are going to present in this paper is a possible way of how to model and test an approach on different levels of abstraction for different features and at different phases of the design. Incorporating different technologies' and methods' characteristics will be possible at distinct parts of the methodology as explained in the following section on conceptual considerations.

## 3. LOCALIZATION SYSTEM CONCEPT

The most accurate localization system in terms of its error is not necessarily the optimal implementation of a localization system when it comes to WSNs. First, there are several constraints to be considered in terms of the motes' hardware capabilities. Second, localization systems that can be used as a middleware service by another application - built upon it - must not interfere with that application or other services. Finally, it is not useful to implement such systems right away on the target platform. One will be better off, if suitable analysis, modeling, simulation, measurement and profiling tools are used first. When all conceptual aspects have thoroughly been elaborated, final design decisions can be made and one can finally map all models to target platform implementations. We will discuss, test and evaluate the most important aspects that need to be considered when implementing a resource-aware indoor localization system in a WSN with Mica2 motes. As these WSN motes are devices of low computational power, computational complexity has

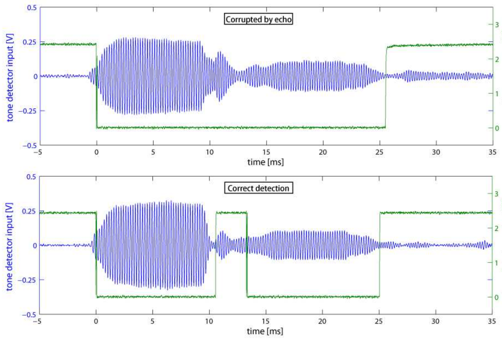



**Figure 4: The tone detector may not always detect the correct pulse length due to echoes or other noise.**

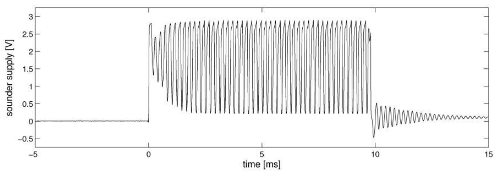



**Figure 5: The voltage across sensor board sounder pins M and G reveals tune-in and fade-out effects.**

to be taken into account. Most of the load stems from least squares (LS) calculations needed due to the multilateration residuum from multiple TDoA rangings of types as shown in Equation 1.

$$d_i^2 = (u_x - x_i)^2 + (u_y - y_i)^2 \quad i = 1 \ldots 3 \qquad (1)$$

The resulting over-determined equation system can be solved with an LS approach according to the L2-norm as given in Equation 2 after using a linearization tool like the one in [12].

$$\underset{x \in \Re^n}{Minimize} \; \|Ax - b\|^2 \;\rightarrow\; (A^T A) \cdot x = A^T b \qquad (2)$$

The system can be solved directly by matrix inversion if $A^T A$ is well populated and non-singular. If it is not, QR-decomposition can be used. In case this is not possible either, singular value decomposition (SVD) always leads to a result [12]. For robustness reasons, this approach will be employed, despite the fact that this is the most complex way of dealing with the problem solution.

## 3.1 Ranging and Position Calculation

The localization system is designed in a modular way such that ranging and position calculation are independent. A coarse grain abstraction is shown in Figure 2. The most know-how and computational effort resides on the receiver side, but this approach makes the process of ranging scalable with the network size due to its computations' localized nature. There are typically few anchor nodes and many more unknown nodes which have to find their position. Figure 2 shows an overview of the ranging concept where hardware components are depicted as ellipsoids. Both, sender and receiver behavior, can be implemented using that design as depicted in Figure 3.

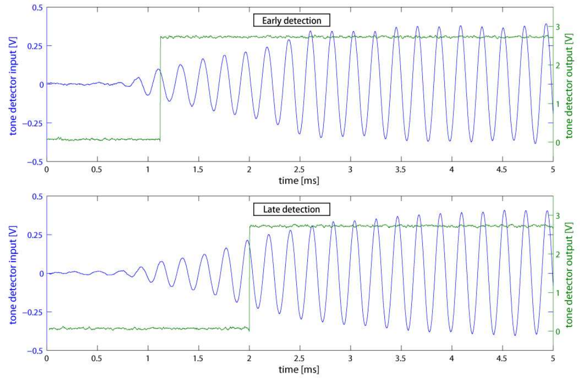



**Figure 6: Tone detector output signal variance.**

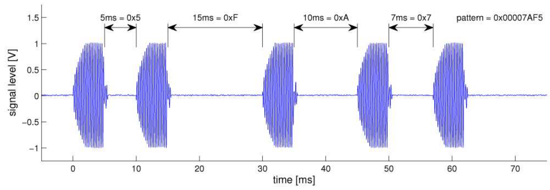



**Figure 7: Coding a chirp with the least significant nibbles first.**

## 3.2 Acoustic Pulse Generation and Detection

The sounder that is used for TDoA measurements heavily impacts power dissipation and timeliness. Therefore, the minimum activation time is found experimentally with tests on hardware. The minimum time of a single pulse is determined with a test program. It turns out that the tune-in effect of the buzzer and the detection latency at the microphone are the main parameters to consider. They are profiled with oscilloscopes and their evaluation is shown in Figures 4, 5 and 6. In case of direct line of sight the first detection of an incoming pulse will give the most accurate results, but multi-path effects, attenuation and noise need to be dealt with. With a known pulse length (of beacon messages) the rising edge and the falling edge of a pulse may still suffer interference from echoes as shown in Figure 4, but it helps improving the performance in terms of accuracy and removing false detections on average. Reducing the microphone gain would remove some echoes, but it would decrease the overall signal to noise ratio - which was profiled with hardware experiments - and it would heavily impact the maximum ranging distance. Summing up, the first time a microphone interrupt fires will be considered signal reception. Interrupts arriving later in time from radio beacons than the maximum assumed distance will be assumed noise or echoes. Furthermore, we implement a robust ranging mechanism with chirps as shown in Figure 7.

## 4. IMPLEMENTATION ISSUES

The core components implementing the process shown in Figure 2 are shown in Figure 3. For achieving more reliable results for noisy audible channels without additional

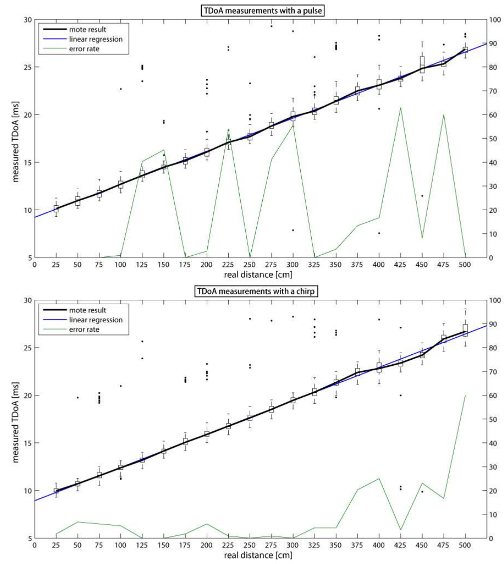
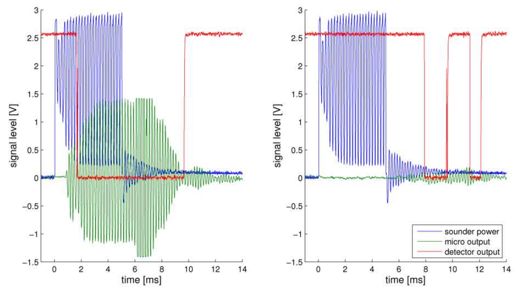



**Figure 8: Using chirps makes the ranging robust.**




(a) Measurement for 10 cm. (b) Measurement for 2 m.

**Figure 9: Both rangings are detected correctly.**

radio overhead chirps are used. Using 5 pulses per chirps turns out to suit the environment's needs when coding the chirps as shown in Figure 7. As WSN motes suffer resource constraints, the localization system overhead has to be minimized. For reducing the time and average power consumption needed the minimum pulse duration is examined. Transient effects of the MTS-310 sensor board piezoelectric hardware last for typically 2 ms before the maximum output power is achieved (Figure 5) and the tone detector needs to lock the internal PLL to the measured signal where measurements have shown that this setup time can be as long as 1 ms (Figure 6). Therefore, the minimum pulse length is 3 ms. Furthermore, two consecutive pulses should be at least 5 ms apart. Otherwise the second pulse will seem to be delayed due to lower amplitude, because the maximum sounder output power is reduced if it is activated again too fast after a preceding activation. The position calculation is implemented in the `CalculateP` module and uses the data provided by the distance table as shown in the component graph of Figure 3. When a calculation is started, the selected beacon data is converted by the linearizing tool, then the SVD is started and finally the LS solution is calculated. If high errors can be detected in the solution, the process is repeated with other parameters. After some rewriting, the Meschach framework was best suitable for doing the calculation. An offline C implementation was used to test the localization algorithm with different types of noise on the range measurements and was then ported to nesC for execution on the nodes. The code size increases by 20 kB. Nevertheless, the library is significantly smaller than other choices available.

The calculation process is initiated by the `PositioningP` module. After beacon election, trust and confidence threshold definition, the SVD is calculated for rangings to a configurable number of motes. For the smallest possible memory footprint and reasonable runtime the minimum number of 4 beacons has been selected for experiments presented here. After calculating the confidence value from special cases of singularity - trust values are used for different roles of beacons, mobile and unknown nodes - the pseudo inverse is calculated according to Formula 3. If the accumulated error grows above 1 m the measurement is dropped and repeated.

$$A \xrightarrow{SVD} U \cdot \Sigma \cdot V^T A^+ \quad = V \cdot \Sigma^+ \cdot U^T x \quad = A^+ \cdot b \qquad (3)$$

System robustness and accuracy heavily depend on the memory footprint that can be sustained on a given hardware platform. The number of rangings that need to be held and evaluated make up most of the system's memory needs. In Mica2 motes one could access external flash memory for relaxing these constraints, but writing to external flash is quite costly in terms of time and power dissipation. Therefore, it was omitted from this work. Still, evaluation shows that the system is quite robust and very accurate.

## 5. RESULTS

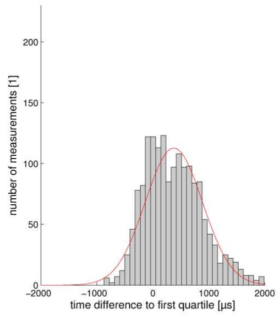
Ranging, position calculation, timing, power dissipation and memory footprint have been profiled. First, results are shown for the hardware accuracy as well as for TDoA range measurements and comparison of single pulses and chirps. Then, model parameters are evaluated and used for simulated localization and real hardware measurements. Both simulated and real-world experiments are discussed in 2D and 3D in terms of their accuracy, trustworthiness and error distribution. Finally, the timing overhead in terms of computational load with its energy needs accordingly, radio module and audible systems' power dissipation and additional memory footprint are evaluated.

**Acoustic Signal Detection and Ranging**

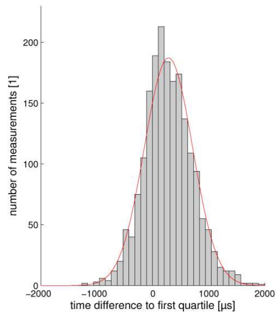
Tracing the buzzer power supply pin, the microphone amplifier and tone detector pin in Figures 9(a) and 9(b) proves assumptions from concept and implementation considerations. While sounder initialization times have already been profiled the receiver delay for generating an interrupt is approximately 0.75 ms here. For the 2 meters measurement the signal is considerably attenuated to remarkably low amplitude.

**TDoA Measurement Analysis**

Nodes are deployed on the floor facing towards each other for comparing single pulses with chirps at distances from 25 centimeters up to 5 meters. Each ranging process consists of 100 TDoA measurements for reasons of statistical signifi-

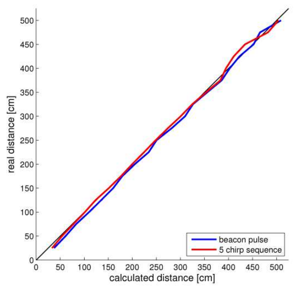



(a) Measurements with separate beacon pulses.

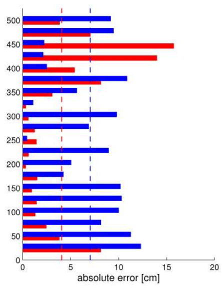



(b) Measurements with pulse sequences (5 chirps).

**Figure 10: As expected the TDoA measurement error is nearly Gaussian distributed and is lower for the sequence approach.**




(a) Full featured ranging process.




(b) Ranging error.

**Figure 11: Two plots to compare chirps and pulses.**

cance. Figure 8 shows the results of the test for both beacon types. The graph shows a box-plot for every measurement to give information about the precision. A high precision is represented by a small inter-quartile-distance. The accuracy slightly drops at high distances because of the lower acoustic signal level whilst the precision nearly stays the same. Figure 10 shows the error distribution of the distance measurements and proves the concept of the implemented statistical data analysis. The data for the histograms is filtered for outliers and is relative to the first quartile of the single dis-

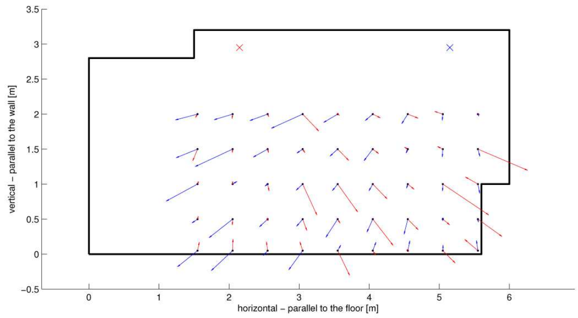



**Figure 12: 2D test on hardware with error force vectors in a $3\,m \times 6\,m$ plane.**

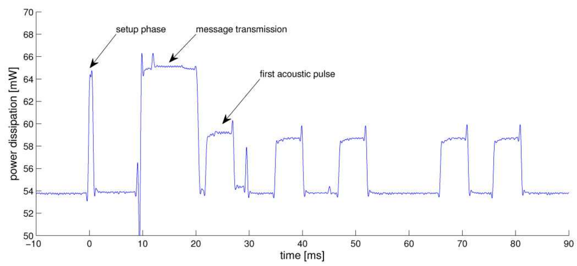
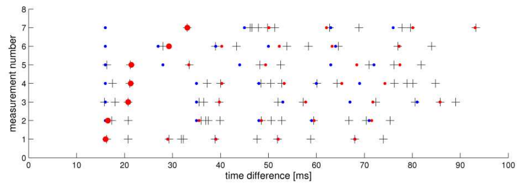
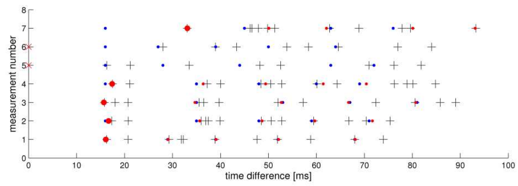



**Figure 13: A 5 chirp and setup after the radio message.**



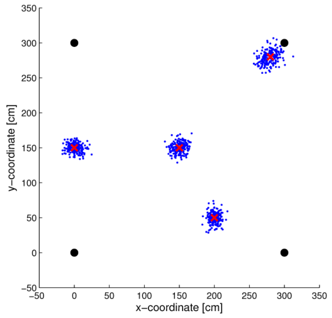

(a) Algorithm with standard deviation correction alone.



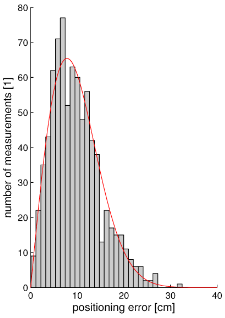

(b) Advanced fault tolerant PSD algorithm.

**Figure 14: PSD performance with interrupts as black crosses horizontally per chirp, blue points as pattern and red ones as PSD results.**

tance measurements. Gaussian nature of the chirp results' distribution allows improving accuracy with averaging. The mean value is only slightly above the first quartile and allows efficient detection of outliers, reducing the impact of echo measurements. The increase of SNR when using pulse sequences from 511 $\mu$s in the beacon mode to 431 $\mu$s when using 5 pulses proves better accuracy of the chirps.

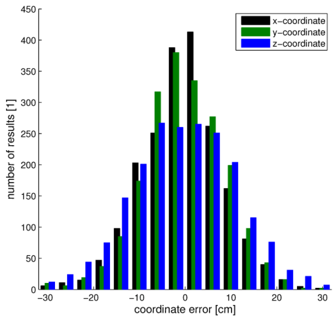



(a) 2D results and mean.

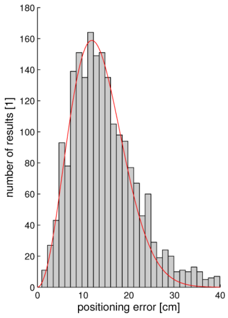



(b) Rayleigh error.

**Figure 15: Simulation results of the positioning algorithm for two dimensions.**




(a) Error per coordinate.




(b) 3rd order Chi error.

**Figure 16: Simulation results of the positioning algorithm for three dimensions.**

**Pulse Sequence Detection (PSD) Algorithm**
Poor debugging features of the Mica2 hardware force us to shape PSD performance with an offline plain-C implementation with real ranging results as input. Figure 14(a) shows an analysis plot of the algorithm performance. The first measurement shows nearly exact pattern matching. The second one shows impact of the standard deviation to correct the too late detected first pulse. The next two calculations are both missing one detection. However, the echo signal was detected better and thus taken as a result. The more drastic measurement number 5 has three out of five pulses undetected. Though, the result would still be correct, because the first quartile eliminates outliers. Finally, earlier detections are preferred and single missing pulses are tolerated. The optimized PSD algorithm succeeds better results as can be seen in Figure 14(b). Measurements 3 and 4 are now detected correctly. The next two pulse sequences were detected badly so the algorithm aborted the operation. This behavior is more useful because invalid detections cannot corrupt the quartile calculation.

**Distance Model Specification**
The environment mainly impacts range estimations with varying temperature. Linear regression over traces taken with different temperature averaged from 100 measurements allows to calculate distance model parameters: the detection time offset at 0 meters ($d$), the gradient at $0°C$ ($k_0$) and the gradient variation per degree ($k_T$). Equations (4) to (7) show regression results and parameter values.

$$regr.: \ for\ 23^{\circ}C : d_{23} = 8798\ \mu s\ ,\ k_{23} = 3549\ \mu s/m \quad (4)$$

$$regr.: \ for\ 11^{\circ}C : d_{11} = 8738\ \mu s\ ,\ k_{11} = 3803\ \mu s/m \quad (5)$$

$$param.: d = 8798\ \mu s \quad (6)$$

$$param.: k_0 = 4013.6\ \mu s/m\ ,\ k_T = -20.20\ \mu s/m/^{\circ}C \quad (7)$$

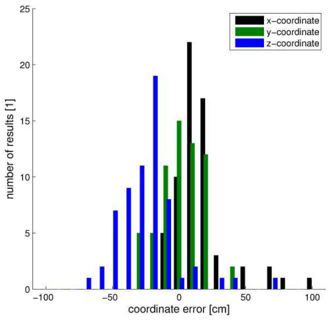
**Ranging Accuracy**
Figure 11 shows the accuracy to be $8.75cm$ absolute or 2.75% relative error. 2D Measurement results in a laboratory at the department are shown in Figure 12 when using 5 pulse chirps for overall 20 TDoA measurements with 250 ms intervals. The error prone points mostly are in the area outside the signal cone of the sounder where the received acoustic signal strength is much lower with a standard deviation of 20 cm and overestimation of 14 cm.

**Ranging Power Measurements**
Evaluation of the approach's power dissipation has been done with an accurate NI setup [4]. Table 1 lists the most important power states and Figure 13 shows the power dissipation of a sender during the generation of a 5 chirp pulse sequence.

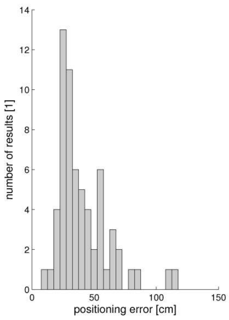
**Position Calculation**
The first step is a simulation based approach. Suitable noise is generated and 2D multilateration is evaluated as can be seen in Figure 15. With a distance error deviation of 8.75 centimeters, the two resulting coordinates are distributed with a standard deviation of approximately 7.8 centimeters

| State | Power | Task | Energy | Time |
|---|---|---|---|---|
| radio active | 53.83 mW | send a 5 chirp | 6.98 mWs | 120 ms |
| radio send | 65.07 mW | send additional chirp | 0.83 mWs | 15 ms |
| radio receive | 53.98 mW | send a beacon | 2.85 mWs | 50 ms |
| send a beacon | 57.06 mW | sender protocol | 9.83 mWs | 150 ms |
| send a 5 chirp | 55.88 mW | microphone activation | 63.5 mWs | 1100 ms |
| receive beacon | 56.50 mW | receive chirp | 7.06 mWs | 120 ms |
| receive chirps | 56.47 mW | receive additional chirp | 0.85 mWs | 15 ms |
| buzzer active | 58.63 mW | receive beacon | 2.82 mWs | 50 ms |
| microphone active | 56.56 mW | receiver protocol | 26.7 mWs | 450 ms |

**Table 1: Different power states necessary for the ranging program.**




(a) Error per coordinate.




(b) 3rd order Chi error.

**Figure 17: Results with beacons in 1 - 3.5 m distance.**

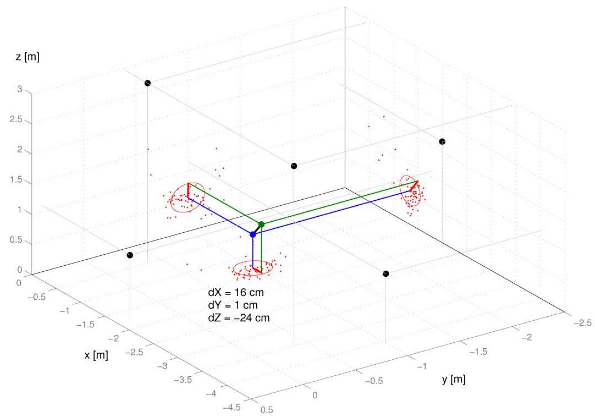



**Figure 18: A 3D experiment with 64 measurements for testing the system.**

each. As the mean of the Rayleigh distribution is defined as $1.256 \cdot \sigma$ the resulting positioning error is $9.8\,cm$ on average. The same simulation runs as described in the last section are now extended to the three dimensional space. The random distance values are again normal distributed with a standard deviation of 8.75 centimeters. As in the latter simulation the results' coordinates are also normal distributed. This leads to the assumption that the positioning error will be Chi distributed this time with three degrees of freedom. Figure 16(a) shows the dependence of the overall accuracy on single poorly occupied axes: the z-coordinate introduces a 20 percent higher variance. As expected, the error is similar to the Chi distribution.

**Test in a Laboratory Environment**

A final test of the system is made to prove the functionality and performance of the overall system. The experiment was performed in the same room as the 2D real world experiment where 5 nodes span a cube of approximately 3 meters edge length. One ranging consists of 16 single TDoA measurements with 5 chirp pulse sequences. Figure 17 shows the combined results of all measurement points. The overall accuracy of the experiment is a mean error of 40 centimeters. In other words 50 percent of the calculation results have an error of less than 33 centimeters as well as 90 percent are within an error of 66 centimeters. The final experiment with its results plotted in Figure 18 achieves 3D accuracy with an error of less then 30 centimeters.

Not yet discussed, Figure 19 shows the real computation phase and the announcement of the node as assistant with the new coordinates.

**Memory Footprint**

The ATmega128 provides 128 kB of memory where 42 kB are used by the system. Still, some of the components are not exclusively for localization as is summarized in Figure 20. For example, a program using almost only the radio stack needs 9.5 kB of program memory, while the ranging system does not need more than only 24 kB as well. There are tough constraints for the RAM of 4 kB of memory from which 1790 bytes are used with a radio buffer size of 8, a positioning table for 6 nodes and an 32 element buffer for

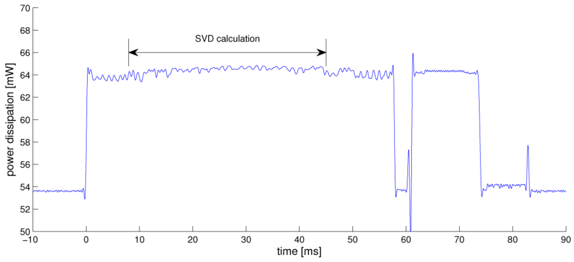



**Figure 19: Position calculation power profile including coordinate transmission.**

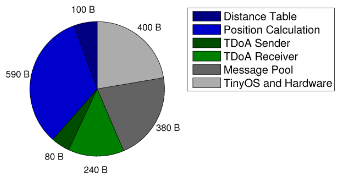



**Figure 20: Localization system memory footprint.**

the PSD. Small configurations can run with 1090 bytes but are less robust.

## 6. CONCLUSION AND OUTLOOK

Concluding, we sum up the localization system properties and findings when evaluating the approach.

### 6.1 Localization System Properties

TDoA measurements from audible sound allow for a success rate of above 50 % for up to 3 $m$ - especially, when using a PSD. Acoustic ranging has an accuracy of 8.75 centimeters or 2.75 percent at distances of up to 5 meters. A typical ranging takes approximately 5 seconds with a required energy of 200 mWs for the sender and 300 mWs for the receiver respectively.
The positioning error in this work is characterized by the mean Euclidean distance of a result set. A higher number of beacons does not necessarily minimize the positioning error, but beacon arrangement has a high impact on the precision of the results. The worst case positioning error is 40 centimeters on average with a ranging precision of approximately 20 centimeters. The energy needed for a positioning process with 5 beacons varies between 1.3 and 6.9 Ws. The framework is light-weight and scalable and therefore perfectly suits conditions under which WSNs need to be operated. The operational distance of the approach with up to 5 meters ranging distance or more allow to apply the approach in industrial settings for process automation or at indoor building sites as well.

### 6.2 Future Work

Ongoing work, based upon the project, is considering the goodput from simultaneous reference measurements for calibration for cancelling out even more errors. Furthermore, looking into finding weighted filters over time for which samples from which times should be considered deserves fine grained analysis.
Another promising option might be to set up an estimation unit for fitting results' bins into error curves that are to be expected from statistical models. Results that have been presented here show that the mapping of simulation results' statistics to experimental results holds for small data sets already.

## 7. REFERENCES

[1] Priyantha, N. B., Chakraborty, A., Balakrishnan, H.: The Cricket location-support system. MobiCom, (2000).
[2] Pletzer, F.: Location Based Services for Wireless Sensor Networks. Master's thesis, ITI, TUG, (2008).
[3] Hörmann, L. B.: Design and Implementation of a Wireless Sensor Platform for River Monitoring Based on Energy Harvesting. Master's thesis, ITI, TUG, (2010).
[4] Glatz, P. M., Steger, C., Weiss R.: Poster Abstract: TOSPIE2: Tiny Operating System Plug-In for Energy Estimation. In IPSN, (2010).
[5] Rathgeb, C.: Implementation of an Acoustic Indoor Localization System for Wireless Sensor Networks. Master's thesis, ITI, TUG, (2010).
[6] Zhang, J., Yan, T., Stankovi, J. A., Son, S. H.: Thunder: towards practical, zero cost acoustic localization for outdoor wireless sensor networks. SIGMOBILE Mob. Comput. Commun. Rev., 11(1), 15–28, 2007.
[7] Whitehouse, C. D.: The Design of Calamari: an Ad-hoc Localization System for Sensor Networks. Master's thesis, University of California at Berkeley (2002).
[8] Harter, A., Hopper, A., Steggles, P., Ward, A., Webster. P., The anatomy of a Context-Aware Application. Wireless Networks, 8, 187–197, (2002).
[9] Matic, A., Papliatseyeu, A.: Tuning to Your Position: FM radio based Indoor Localization with Spontaneous Recalibration. PerCom, (2010).
[10] Wilson, J., Patwari, N.: Radio Tomographic Imaging with Wireless Networks. IEEE Trans. Mob. Comp., (2009).
[11] Youssef, A., Agrawala, A., Younis, M.: Accurate anchor-free node localization in wireless sensor networks. 24th IPCCC, (2005).
[12] W. S. Murphy W.S.: Determination of a position using approximate distances and trilateration. Master's thesis, Colorado School of Mines, (2007).

2011 18th International Conference on Telecommunications

# HANS: Harvesting Aware Networking Service for Energy Management in Wireless Sensor Networks

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics, Graz University of Technology, Austria*
$\{Philipp.Glatz,\ Leander.Hoermann,\ Steger,\ RWeiss\}$*@TUGraz.at*

## Abstract

*Large scale deployments of small, wireless, networked, embedded systems demand for cost reduction in development and maintenance. Most often, this translates into the need for reliable methods for energy conservation as it is the case for wireless sensor networks (WSNs).*
*Our work considers energy harvesting system (EHS)-enhanced WSN technology which is the state-of-the-art technology for perpetual systems supplied from ambient environmental energy. Therefore several aspects have been considered in literature so far: EHS design, energy prediction modeling, harvesting aware media access control (MAC) and routing and finally power management.*
*This paper postulates that identifying and optimizing these aspects on their own does not necessarily lead to feasible solutions. We take a cross-layer perspective and provide a networking protocol for implicit or explicit EHS policy negotiation under the constraint of two prototypical communication patterns.*
*The methods presented show how to combine energy aware routing or conservation from network coding with EHS power management policies under the constraint of state-of-the-art MAC. Evaluation of different end-user communication patterns lets the reader interpret the results for an application at hand.*

## 1. Resource Constrained Networking

On one side, wireless sensor networks (WSNs) [1] constitute a state-of-the-art technology. On the other side, methods from WSN research and the technology itself are found in many applications nowadays. Especially the diversity of possible applications make it difficult to develop generically applicable optimization methods. Firstly, subsequent paragraphs discuss optimization measures for energy conservation in WSNs. Secondly, we will show that different related optimization methods heavily impact one another: harvesting aware routing provides information on a path's energy budget but not yet a protocol for duty cycle (DC) negotiation. Thirdly, we show how different patterns of communication may demand for contradicting types of optimization.

978-1-4577-0024-8/11/$26.00 ©2011 IEEE

### 1.1. Energy Harvesting and Power Management

Perpetual operation of WSNs can be achieved by attaching an energy harvesting system (EHS). Especially solar energy harvesting has been modeled and implemented for WSNs. Borrowing concepts from artificial intelligence and machine learning, different prediction methods like supervised neural networks, spectral power density analysis and autonomous echo state networks and many others can be applied. Prediction results based on such methods are then used for adaptive power management. Sensor nodes set a limit for the workload that can be assigned to them according to the expected energy budget. This can translate into a maximum end-user performance in terms of communication throughput and delay or information accuracy and resolution.
Networks that make use of EHS hardware will most probably also want to integrate harvesting aware methods for media access control (MAC) and routing. Many aspects have been considered in WSN research that profile the powering needs of different MAC and routing approaches and their interaction given a specific application.

### 1.2. Cross-Layer Energy Conservation Perspectives

Despite the fact that profiling a routing protocol for a single application does not imply its behavior for other applications it is still very useful for comparing similar applications. Though, we also have to set preamble lengths when using low-power-listening (LPL) accordingly for avoiding overhead in powering needs.
Similar consideration can be made for the interaction of networking optimization and EHS optimization. Motes may want to use network coding instead of routing for energy conservation in communication-oriented applications. Setting up different routing or coding structures requires proper and maybe repeated updates of their networking policies to conserve energy with reducing the number of messages that need to be sent. The other side of the picture is that the best optimization will not be of any use if the EHS-computed maximum sustainable duty cycle (DC) can only

978-1-4577-0024-8/11/$26.00 ©2011 IEEE

be negotiated at high overhead costs. A reason for that could be that it is assumed to piggyback control information on data messages, but there are only messages sent towards a single sink node. So the networking abstraction has cut out communication into the opposite direction as an optimization step and no DC can be negotiated.

### 1.3. The Need for Application Metrics

Although WSNs are of great interest to the communications society we still lack a common metric for characterization of communication patterns and their impact on application needs. We consider the impact of choosing between synchronous and asynchronous duty cycling and between piggybacking control information and sending separate control packets.
While metrics for WSN middleware exist, it is out of scope of this work to provide an evaluation of many different quality of service levels, network topologies, data rates or variation of other attributes. We conceptually consider two different types of application as depicted in Figure 1:

- information or monitoring applications have their network traffic directed towards one single data sink
- communication networks are used for forwarding data in the style of unicasts without modification

### 1.4. Contribution Claim

This paper provides the following contribution.

- We provide a unified view on energy conservation in WSN networking that is especially applicable for EHS-enhanced WSNs.
- A discussion on application metrics shows that not all approaches can be developed in a generic and optimal way. We compare extra control messages with piggybacking the information on application-data messages.
- We deduce which form of networking service for harvesting policy negotiation is applicable to which types of application.
- We give pseudo code for protocol rules of HANS: Harvesting Aware Networking Service.
- Accurate hardware measurements of duty cycled, EHS-enhanced WSN motes and MAC establish the basis for comparison of routing and network coding which in turn gives the precondition of available information flows for setting up HANS. So all the way from hardware up to application needs can be considered from a cross-layer perspective.

## 2. Related Work

This section presents related work for harvesting systems and modeling, LPL, routing and wireless network coding.

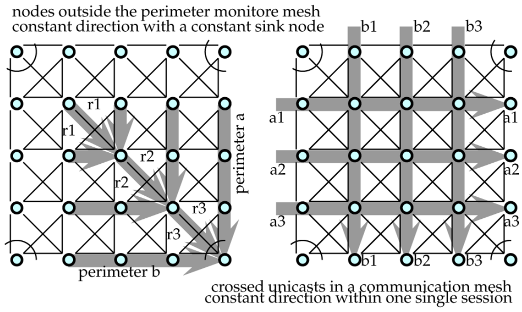



Figure 1. We choose a mesh topology with an 8-neighborhood at inner nodes. A monitoring application and a communication network will be considered.

### 2.1. Energy Harvesting

EHS reliability can be increased with hardware dimensioning as with Heliomote that is referenced from a theoretically modeled in [2] or with a policy update. A policy update is given with introducing thresholds at the upper and lower limits of an energy storage efficiency interval selected as is shown in [3]. Although this solves the problem of pairwise adaptation, it is a non-trivial problem to decide whether a scalable protocol for adaptation negotiation can exist. The demands in robustness needed for reliable power management heavily depend on the system model error which in turn comes from simulation and prediction errors. A range of algorithms is evaluated in [4] where we will select the one used in [5] and [3]. The modeling period is chosen as the spectral density's main lobe frequency.

### 2.2. Media Access Control and Networking

The readily applicable MAC in version 2 of the most far spread WSN operating system TinyOS provides LPL which is discussed in [6]. The preamble of a message being sent is long enough to be heard by any receiver that is duty cycling its radio module.
Different energy and harvesting aware routing methods are briefly reviewed in [7] that are then juxtaposed to network coding. Network coding initially stems from the theory of network information flows [8]. The relation to solving the Steiner Tree [9] problem shows that network codes can be established in polynomial time while construction of a optimal routing scheme is np-hard. These relations are summarized in [10]. The application of network coding in xor-coding [11] has been used in previous work to introduce network wide energy management for network coding WSNs [12] for the case of implementing multiple crossed unicasts [13]. The type of topology used in [12] and [13] is shown by Figure 1 for 25 nodes. A maximum

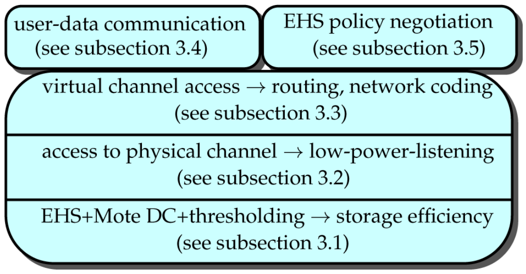



Figure 2. All three different layers offer different types of optimization. A single layer's optimization potential depends on optimization measures taken at other layers. Application data and control data efficiency may rely on different types of optimization depending on their communication patterns.

gain of 2 can be achieved for wireless network coding if all transmissions provide coding opportunities.

## 3. Design Methodology for the Harvesting Aware Networking Service (HANS)

This section explains the duty cycling, MAC and networking approach that is used for HANS as summarized in Figure 2. A brief outline of all methods in use is given.

### 3.1. Adaptive Duty Cycling

The energy harvesting system in use is capable of ($\rho/\sigma$)-modeling. The working point is chosen in an area where the EHS storage efficiency is above 60 %. The impact of choosing a working point will be calculated from profiling measurements of mote and EHS hardware for a fixed DC. While it is not possible to characterize all possible DCs in measurements we will consider the existence of mutually independent energy harvesting device (EHD) patterns and therefore different DCs for theoretical considerations.

### 3.2. Media Access Control

LPL is considered, because it is one of the main features used for energy conservation in WSNs and its implementation also has significant impact on other possible optimizations at the networking layer. We again rely on measurements conducted in earlier experiments. The main characteristic that will be used in this paper is the ratio of sending to receiving cost of 2 that will be used for simulation annotation.
We will stick to the standard TinyOS2 LPL implementation, because it is light-weight and easy to instrument. The impact of radio overhead to persistent storage overhead will be measured in hardware to quantitatively describe what has qualitatively been modeled earlier.

### 3.3. Network Topology, Routing, Network Coding

Figure 1 shows a mesh topology with horizontal, vertical and diagonal links. For variable network size we will consider energy aware routing, ad-hoc on-demand distance vector routing and xor network coding.
Different HANS modes are compared in Table 1. It turns out that the optimization of running the per-mote maximum sustainable DC comes at high overhead costs for DC negotiation and data communication. We assume the same worst case scenarios that have been used in [13] and [2]. Therefore, we will use synchronous DCs. Furthermore, we go for separate control messages instead of piggybacking data. The main reason for that is conservation of scalability. If we switch the radio activation to polling mode instead of using MAC slots, we can use synchronous duty cycling and control messages with a constant overhead of 2 time slots. This will be explained in the implementation section. Different options are evaluated in the results section.

### 3.4. Application Metric

Topologies, communication patterns and data rate vary among different applications. No generic approach exists that satisfies all constraints at the same time while being optimal. So we look at two cases: a monitoring application where each node has to send its data to the perimeter of the network and the case of a network serving as communication infrastructure. As there are no means to capture all possible types of application demands we use these two cases with very different flow topology characteristics. Using examples from two ends of the communication paradigm spectrum shall be a step towards classifying different application characteristics.

### 3.5. Harvesting Aware Networking Service

HANS: Harvesting Aware Networking Service allows to adapt the protocol for duty cycle negotiation to an application's needs with respect to environmental conditions. The main questions that lead to a decision are as follows.

- What type of duty cycling shall be used - synchronous or asynchronous?
  - This will result in different delays - heavily impacted by the EHS and the EHD's supply power - and hard to predict efficiency.
- Which type of communication is best - piggybacking or control messages?
  - This depends on data rate and direction of the top-level application, while separate messages are more robust than piggybacking.

All protocols under consideration will set their duty cycles such that they start with the same offset in a given modeling period. The authors are aware of the fact that the modeling period has to be negotiated initially, but providing an approach for that is out of scope of this work. So, our considerations are restricted to long term - at best, perpetual - deployments where initial effects decay. Work in progress is currently considering the initialization procedure.

## 4. HANS System Setup

The system is set up such that the DC operates the EHS in an efficiency range above 60 %. The efficiency measurement setup will be explained.
The MAC has been measured in hardware on one side and measurement results are used for configuring a simulation to compare networking approaches on the other side.
The networking approaches' implementation is explained. Different MAC implementations lead to different results of the comparison, so they will be compared qualitatively.
Then, as the tradeoffs on lower layers are explained, possible types of DC negotiation for perpetual operation are discussed. HANS rules provide a possible implementation. Different modes' overhead complexity will be evaluated in theory.

**Algorithm 1** HANS Negotiation Rules

**Require:** INITIALIZATION $\equiv$ $finished$ ▷ $t \geq n$
**Require:** $(\rho/\sigma)$ $capable\ EHS$ ▷ $PeriodStartKnown$
**Ensure:** $ModelingPeriodSynchronicity \equiv TRUE\ \forall\ periods$

**function** NEGORCOMMDUTYCYCLE ▷ Matlab Notation
  **for** $mote \equiv mac\_order$ **do**
    **for** $rcvr \equiv 1 \ldots n$ **do** ▷ $if(bwidth(mote))$
      **if** ASYNCPIGGYBACKING **then**
        NEGCOMM ▷ $DC + AllNbgsAwake$
        DATACOMM ▷ $DC + AllNbgsAwake$
        TRACENGBDCS ▷ $PersMem$
      **else if** ASYNCCONTROLMESSAGE **then**
        NEGCOMM ▷ $DC + NextDiagsAwake$
        DATACOMM ▷ $DC + NextHopAwake$
        TRACENGBDCS ▷ $PersMem$
      **else if** SYNCPIGGYBACKING **then**
        NEGCOMM ▷ $InitDC$
        DATACOMM ▷ $DC$
      **else if** SYNCCONTROLMESSAGE **then**
        NEGCOMM ▷ $InitDC$
        DATACOMM ▷ $DC$
      **end if**
    **end for**
  **end for**
**end function**

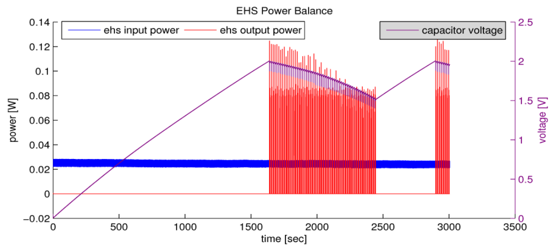



Figure 3. The EHS efficiency is characterized with an application kept constant over a duty cycle of 5 seconds power-on and 10 seconds power-off.

### 4.1. Energy Harvesting System Measurements

Measurements are conducted with a National Instruments data acquisition card PXI-6221. The setup consists of Mica2 motes from Crossbow attached to the EHS described in [3]. One of the two EHD ports is connected to a Voltcraft PS 405 Pro laboratory power supply. The DLC used in the setup has been characterized and it is operated at a level above a storage efficiency of 60 %. Figure 3 shows the measurements. Efficiency within that interval has been evaluated from the time that is needed for reloading the DLC over the interval that has been lost over one DC while the mote is running a simple BlinkToRadio application.

### 4.2. MAC and Networking

As explained before, all experiments with hardware have been conducted with TinyOS2 LPL. The factor of 2 has been used for annotation of a simulation where this factor has an important impact on the possible optimization of network coding over routing as has been shown in [7]. The value can vary depending on the MAC implementation. Despite these facts, we use a worst case MAC slot ordering for networks with the structure as shown in Figure 1 where each node gets only one MAC slot per time slot / DC due to rigorous reliability constraints that have to be met for energy harvesting WSNs.

### 4.3. HANS Rules

As it is the case for data messages, also DC negotiation messages can only be propagated through the network if all nodes participating in communication are awake at the same time. All nodes synchronize their modeling periods such that all nodes are awake at the same time in the initial DC. For synchronous DCs all nodes are driven with the maximum sustainable DC and standard communication procedures as shown in [13]. For the asynchronous approach

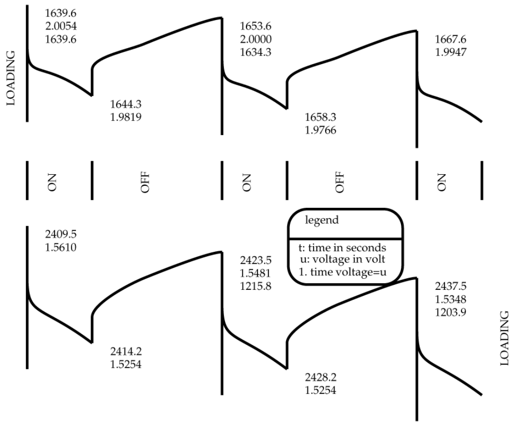



Figure 4. 10 points of interest from Figure 3 are depicted qualitatively. For 4 points the time is given when the EHS dual layer capacitor (DLC) clamp voltage reached the same voltage level while loading. Voltage drops occur when the EHS stops charging the DLC and supplies the mote. A voltage step occurs when switching back. Switching occurs at a DC of 33 %.

one has to send data at times only when two adjacent nodes are awake at the same time. We describe this with the least common multiple ($lcm(DC(i), DC(j))$) of different motes' DCs. Using control messages, this can suffice for policy negotiation communication. For piggybacking negotiation data on user data, we have to restrict communication to times when adjacent nodes are awake. Algorithm 1 gives a pseudo code listing for DC negotiation.

## 5. Results

This section presents the results from measurements, simulation and theory.

### 5.1. EHS Efficiency

The ratio of input-output efficiency is profiled with comparing the times needed for reloading the voltage drop of a single DC. Figure 4 depicts the first and last 3 DCs of measurements shown in Figure 3. Reloading the first DC's voltage drop takes 5.3 seconds, while it takes 14.9 seconds for the last one. In other words, one gets 281 % efficiency out of the first DC compared to the last one within the DLC storage efficiency interval of 60 % and above.
With regard to these results we suggest to send control messages - if they are used instead of piggybacking - at the beginning of each modeling period. This comes at the drawback that one has to predict the next but one modeling period, but related work discussed in this paper already considers initial modeling periods of several days and the use of thresholds anyway.

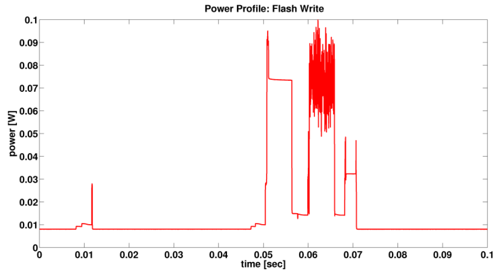



Figure 5. Accessing flash storage heavily impacts the power dissipation Profile. A single byte block write shows overhead costs for clearing a full page, writing and syncing for persistent storage.

### 5.2. MAC and Persistent Storage Cost

Previous work has characterized sending power dissipation cost based on MAC settings [13]. Extending the view to network wide energy management as shown in [12] demands for characterization of persistent storage energy cost. Figure 5 shows that significant overhead has to be handled. This especially impacts EHS duty cycled networks where distributing storage access among several time slots activates the overhead. We draw the conclusion that HANS rules have to be set such that negotiation occurs in a single time slot per direction with MAC slots reordered accordingly.

### 5.3. The Gain of Network Coding over Routing

Figure 6 shows the gain of network coding over routing for a sending cost of 2 and listening cost of 1 energy unit for 4 rounds of packet injection. Comparing these results to the maximum number of duty cycles (54) running a simple TinyOS impacts HANS design decisions twofold.
On one side the control message overhead presented in the next subsection keeps asynchronous approaches from being applicable due to the limited number of overall activation times.
On the other side we even have to keep the number of time slots used below the network edge length to not lose gain from network coding with policy negotiation. AODV-like routing implementations with reliable networking settings can send the maximum sustainable DC back into the network - which is non-trivial for coding - with acknowledgements. Though, the delay of piggybacked control information increases at least with the network edge length which makes it infeasible for piggybacking to conserve scalability.

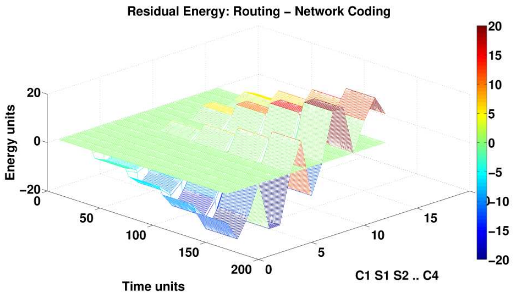



Figure 6. The per mote difference of messages that need to be sent for routing and network coding is shown for a $4 \times 4$ network. The topology is the same as in Figure 1. Motes are numbered from the upper left down (corner $1$, sender $1$, $\ldots$) and to the right. Otherwise stressed inner node's energy is conserved with coding.

### 5.4. Control Messages Overhead

Table 1 shows that the least overhead from policy negotiation alone occurs for synchronous DCs with extra control rounds. Though, it is obvious that per-mote maximum sustainable duty cycle and piggybacking control information can be the only choice for maximum end-user performance. Nevertheless, intelligent network behavior can still perform better. The reason can be found in higher energy loss due to the need for additional persistent memory access as commented in algorithm 1. Increasing $\rho_{Mote}$ will in turn reduce the maximum sustainable duty cycle.
The authors are aware of the fact, that different types of optimizations are best suitable for different types of systems. The decisions made try to support low data rate, EHS-enhanced WSNs of small form factor which includes limitations in size of its energy storage.

## 6. Conclusion

This paper presents HANS: Harvesting Aware Networking Service. A cross-layer perspective is taken to compare possible optimization of different levels of abstraction.
We suggest the use of ($\rho$/$\sigma$)-modeling and evaluate the efficiency of an EHS-enhanced mote prototype. State-of-the-art LPL is used for MAC, which provides the basis for the gain that can be achieved and has been evaluated with network coding. Based on two exemplary application communication patterns we show that a synchronous DC policy negotiation with separate control messages performs best for scalable WSNs.
Future work is concerned with building an accurate EHS simulation environment for studying the interaction of HANS with network wide energy management for WSNs at scale.

Table 1. The order of time slots needed for given HANS modes for different DC types.

| modes | async | sync |
|---|---|---|
| control msgs | $\sum_{i=diag(1)}^{diag(k-1)} lcm(DC(i), DC(i+1))$ | $2$ |
| piggyback | $\sum_{i=diag(1)}^{diag(k-2)} lcm(DC(i) \ldots DC(i+2))$ | $2k$ |

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, 2002.

[2] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, 2007.

[3] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in WSNs," in *IFIP/IEEE WD*, 2008.

[4] C. Bergonzini, D. Brunelli, and L. Benini, "Algorithms for harvested energy prediction in batteryless wireless sensor networks," in *IWASI*, 2009.

[5] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," *SIGMETRICS Perform. Eval. Rev.*, 2004.

[6] S. Moon, T. Kim, and H. Cha, "Enabling low power listening on ieee 802.15.4-based sensor nodes," in *WCNC*, 2007.

[7] P. M. Glatz and R. Weiss, "LINDONCS: localized in-network detection of network coding structures in wireless sensor networks," in *The 4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks*, 10 2009.

[8] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, 2000.

[9] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing steiner trees," in *ACM-SIAM SODA*, 2003.

[10] T. Ho and D. Lun, *Network Coding: An Introduction*. Cambridge University Press, 2008.

[11] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, 2008.

[12] P. M. Glatz, J. Loinig, C. Steger, and R. Weiss, "A first step towards energy management for network coding in wireless sensor networks," in *MICC*, 2009.

[13] P. M. Glatz, K. B. Hein, and R. Weiss, "Energy conservation with network coding for wireless sensor networks with multiple crossed information flows," in *I-SPAN*, 2009.

# A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
*Institute for Technical Informatics*
*Graz University of Technology*
*Graz, Austria*
{*Philipp.Glatz@, Leander.Hoermann@Student., Steger@, RWeiss@*}*TUGraz.at*

***Abstract*—Wireless sensor network (WSN) nodes have to cope with severe power supply constraints. Energy harvesting system (EHS) technology is used for prolonging network lifetime. Robust operation of such systems heavily relies on accurate models of EHS efficiency and node power dissipation for calculating sustainable operation modes.**
**A node's energy balance can be described with a power state model (PSM). While for battery operated WSNs PSM measurement errors and battery effects have to be considered, this paper widens the point of view to EHS properties. We analyze varying PSMs at runtime, EHS efficiency and measurement error's impacts on duty cycled WSNs.**
**We show a measurement system setup and results from profiling a PSM for Mica2 nodes with an EHS. We explain important considerations for such systems with showing the variance of a node's energy consumption depending on its supply and we profile EHS efficiency for deducing an energy storage level operating point.**
**The results contribute to problems in the field of low power WSNs and especially EHS supported ones. Robustness of duty cycled WSNs heavily depends on measurement error bounds and is impacted by EHS efficiency. The reader is provided with a methodology for measuring and modeling robust WSNs.**

***Keywords*-efficiency; energy harvesting system; measurement; power dissipation; power state model; wireless sensor networks;**

## I. Introduction and Motivation

The limited lifetime of a wireless sensor network (WSN) node can be prolonged by attaching it to an energy harvesting system (EHS). Now, while EHS driven duty cycled WSNs allow for longevity they are inherently hindered from large scale networking due to the pairing of costly radio activation and energy harvesting policy negotiation overhead at the networking layer. The question is how to optimize the negotiation times and their frequency subject to maximize end-user performance.

For providing a robust networking service on one side and still support high end-user performance on the other side, the errors of power profiles that low-power policies are based upon deserve careful consideration. Furthermore, a working point with high system efficiency has to be found for working off necessary protocols.

We contribute to the field in that we present an automated measurement setup. Firstly, it is capable of profiling WSN motes for their power profiles and to deduce a power state model (PSM). We show how to deduce it and give bounds for the maximum measurement error that can occur. Secondly, it allows to measure motes that are driven by an EHS. Thereby, the EHS - which usually is a measurement system itself - is profiled for its accuracy and efficiency. We deduce operating points and analyze the range and impact of PSM and EHS errors. We state our results with explanations for WSNs with EHS duty cycling policies, but other low power and harvesting methods are affected as well.

## II. Related Work

Several simulation environments exist for WSNs where some allow summing up PSMs over time for tracing an application's power profile. Prominent examples are AEON [1], EMStar [2] and VMNet [3]. Unfortunately, simulation results are of limited accuracy, especially for EHSs. Other approaches go for online profiling which can be separated into testbeds and online measurements. [4] presents a 24 node testbed with hardware measurements that are compared to simulation results. They report a few percent ($1.03\,\% - 13.48\,\%$) error compared to their reference for which the maximum error was assumed to be below one percent. Accurate on-board and sandwich-on measurements are implemented by the BTNode platform [5], SPOT [6] and an energy measurement board in [7] which reports inaccuracies of PowerTOSSIM [8] of up to 31 percent.

As explained in the introduction, low-power techniques and their optimization are the main reason for all the effort being spent on driving measurement and simulation errors lower if possible. Modeling of an EHS is exceptionally heavily impacted. $\rho/\sigma$-models [9] measure energy harvesting device (EHD) power, an attached node's power consumption and the actual energy budget. A maximum sustainable duty cycle is predicted based upon that information. Robust operation depends on introducing safe bounds [10] and taking account of measurement errors. So, self-contained measurement systems as implemented in an EHS have to be profiled prior to deployment for reliable and accurate operation.

A solution for measurement automation to preserve re-

978-1-4244-5328-3/10/$26.00 ©2010 IEEE 

peatable measurements and to save time - the profiling of harvesting policies can take hours to several days - is provided by National Instruments (NI) LabVIEW. LabVIEW is a purely graphical development environment implementing a graphical, general-purpose programming language called G [11]. A real instrument can be transformed into a virtual software-based instrument with extended functionality. This improves the flexibility of such a measurement system. We use a similar measurement setup as used for [12].

## III. Characterization of WSN Power Dissipation under EHS Supply

Offline simulation and characterization needs the appliance of power models and power state simulations. Our concept allows to profile EHS supported WSNs. One has to keep in mind that the same mote hardware with the same powering needs might drain with a higher variance in case an EHS supplies a mote compared to the case of the more constant supply with a battery or a power supply.
The latter is easier to model in that a mote's power dissipation profile can more easily be captured if it is supplied by a near-monotonic energy reservoir of a battery and not from a pulsed EHS output stage. We did not find any prior work that explicitly pointed this out for characterization of a mote's power state. For reducing errors we use a laboratory power supply of type Voltcraft PS 405 Pro instead of batteries to operate the mote and to drive the EHS.

### A. Power Dissipation Profiling with Laboratory Power Supplies

Figure 1 gives a block diagram of the measurement setup. The measurement hardware needs to measure the power consumption of the mote. For an easy analysis of the different power states it is also necessary to capture the digital representations of the port state. Therefore, eight pins of the mote are combined and connected to a digital port of the measurement hardware. Each power state gets a unique number. For being able of doing so we provide a software interface that can be used by a programmer for the annotation of power states.
The measurement of the power consumption and of the states should be done with the maximum available sampling frequency for achieving accurate results. However, a trade-off must be found between the accuracy and the resulting data-rate of the measurements. Furthermore, different error sources may impact measurements.

1) The absolute accuracy of the measurement hardware is an important error source that cannot be compensated without changing hardware.
2) The tolerance of the components in use must not be neglected. This error can be reduced by an exact measurement of the components.
3) The total noise of a measurement consists of noise produced by the measurement hardware and noise produced by the device under test (DUT). Noise of the measurement hardware is integrated in the absolute error. The DUT noise must be considered separately.
4) The finite accuracy of the measurement hardware input resistor influences measurement results too.

A later section shows detailed calculations for all of these error sources. Other sources like electromagnetic radiation and temperature can be neglected for the laboratory conditions at hand. Using a laboratory power supply allows to keep measurements repeatable with the possibility to compare different instances of the same type of hardware.

### B. Power Dissipation Profiling with EHS Supply

In this subsection we describe the concept of the power dissipation profiling of motes supplied with an EHS. The measurement setup is similar to the previous one, but the power supply is replaced by the EHS. Here, the same assumptions about the measurement concept can be made. Again, the same error sources are to be expected.
The power consumption of a mote depends on the supply voltage. The EHS cannot stabilize this voltage as a power supply can do it. So, there will be a voltage drop. Also, the output voltage depends on the charge of the capacitor. For achieving a good EHS characterization these dependencies can be measured with the setup.

### C. EHS Efficiency Measurement

Figure 1 shows the setup for measuring EHS input and output power and its energy balance. The digital representation of the power state of the mote can be captured too. Now, more information has to be processed. Therefore, the data rate can not be as high as in the previous measurement setups. However, a high data rate is not that important here. The measurement time is usually much longer and single samples can be traced as a moving average with variable resolution. This relaxes timing constraints when writing measurement data to the NI setup's hard disk. Again the same error sources are to be considered.
This measurement setup is important for measuring the efficiency of the EHS. Different thresholds [10] can be tested too. Thresholds are limits that bound the range of energy storage levels where the mote can be operated. This is a main design criteria and heavily impacts robustness. So the best options of the EHS policy negotiation can be determined. This can greatly enhance reliability of a WSN. Furthermore, the efficiency of the input charge pump and output charge pump can be calculated with tracing the energy stored in the double layer capacitor (DLC).

### D. Test Environment

For testing the measurement setups the same Mica2 mote with the same program (TinyOS2 BlinkToRadio) will be used in all cases. Mote communication is the most important part of wireless sensor networks. Therefore, it is integrated

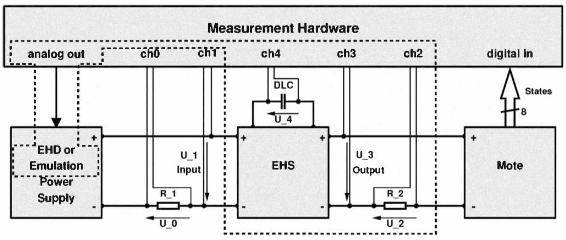



Figure 1. The automated PSM measurement setup and the extended setup (dotted area) for EHS efficiency profiling have been used with a laboratory power supply.

in the test program. LED activation gives easy access to measuring the impact of changing a power supply on constant PSM parts. Due to the fact that only one program is used, single measurements can be compared easily with each other. In the following list the main measurement targets will be discussed:

1) Full EHS charge / discharge cycles will be measured. The laboratory power supply, DLC charge and mote dissipation will be tracked. This allows calculating EHS efficiency values as they change over different duty cycles.
2) The evaluation of the total energy consumption of the used program will be compared with different supplies of the mote. These are battery, laboratory power supply and EHS.

## IV. Power State Characterization of a Sample Application

For the BlinkToRadio application the following different power states can be identified:

1) Power down mode: a mote's power consumption can be neglected when it is switched off. The self-discharge of the EHS hardware in use has a considerably higher leakage in its DLC. Its self-discharge will half its energy reservoir within a few weeks.
2) Low power modes: have not been used. Under the given constraints it would have been too complex to implement low power configurations.
3) Mote Initialization: when the mote is switched on it has a characteristic current profile. EHS supplied motes will show this behavior in every cycle of their operation.
4) LEDs: different LEDs have different powering needs. Their power consumption cannot be neglected. They should not provide debug output if not needed when they are deployed due to their power consumption.
5) Radio: radio characteristics are influenced by setup times, transmission power, preamble length and inter-mote communication effects. These are not explicitly considered by this measurement setup.

### A. Power Dissipation Profiling Setup

A shunt resistor is used to measure the current of the mote. The voltage drop at the shunt resistor and the supply voltage of the mote are used to determine the total power consumption. These voltages are measured differentially for common mode rejection.
The measurement is performed with a sampling rate of $100kS$. Analog lines and digital lines for tracking power state are synchronized with an external clock signal. The light-weight software component for power state annotation allows trapping events as short as timer fired events.

*1) Shunt Resistor Value:* Accurate results need an accurate measurement. We select a PXI-6221 data acquisition card (DAQ) and a shunt resistor for an expected maximum current of $35\,mA$ being drawn by the Mica2 mote. The lowest possible DAQ input range is $-200\,mV$ to $200\,mV$. Therefore, the maximum voltage drop at the shunt resistor can be $200mV$:

$$R_{shunt} = \frac{U_{shunt}}{I_{max}} = \frac{200mV}{35mA} = 5.714\Omega.$$

$R_{shunt}$ should be lower than the calculated value to guarantee correct functionality. So the value of the shunt resistor is set to

$$R_{shunt} = 4\Omega$$

for being able to capture peaks above $35\,mA$.

*2) Power Consumption:* The DAQ measures mote supply voltage as $U_1$ and its current as the voltage drop $U0$ at the shunt:

$$P_{mote} = U_{mote} \cdot I_{mote} = U_1 \cdot I_{shunt} = U_1 \cdot \frac{U_0}{R}.$$

*3) Error Analysis:* We outline all expected errors:

1) **Measurement Error Propagation:** Power profile

$$P = U_1 \cdot \frac{U_0}{R} = f(U_0, U_1, R)$$

errors from DAQ and shunt propagate as follows:

$$\Delta P_{er} \approx \pm\left(\left|P \cdot \frac{\Delta U_1}{U_1}\right| + \left|\frac{U_1}{R} \cdot \Delta U_0\right| + \left|-P \cdot \frac{\Delta R}{R}\right|\right).$$

For an absolute accuracy of the supply voltage ($3\,V$) measurement of $\Delta U_1 = \pm 1620\mu V$, the voltage drop accuracy at the shunt of $\Delta U_0 = \pm 112\mu V$ and its tolerance of $2\,\%$ we get:

$$\Delta P_{er} \quad \left( \left| \quad 1.62mV \right| \left| \frac{3V}{} \qquad \right| \qquad \right)$$

The worst case offset of measurements from the true value mainly depends on resistor tolerance:

$$P_{true} = P + \Delta P_{er} \approx P \pm (P \cdot 0.02054 + 84\mu W).$$

2) **Noise:** The temperature is assumed to be $T = 25°C = 298.15°K$ and the maximum sampling rate (and thus the bandwidth) is assumed to be $B = 100kHz$:

$$U_{noise} = \sqrt{4 \cdot k_B \cdot T \cdot B \cdot R_{shunt}} = 81.14nV.$$

470

The noise error is far below DAQ sensitivity at lowest input range $U_{sens} = 5.2\mu V$ and can be neglected.

3) **Internal Resistance of Supply Voltage Measurement:** The internal resistance of the DAQ ($1M\,\Omega$) and the battery ($0.5\,\Omega$) lead to

$$\Delta U = R_{i,batt} \cdot \frac{U_{batt}}{R_{i,meascard}} = 0.5\Omega \cdot \frac{3V}{1M\Omega} = 1.5\mu V$$

which is below DAQ accuracy for the used input range of $48.8\,\mu V$ and can be neglected.

4) **Internal Resistance of Shunt Voltage Measurement** For $I_{max} = 50\,mA$ we get

$$\Delta R = R_{shunt} - \frac{R_{shunt} \cdot R_{i,meascard}}{R_{shunt} + R_{i,meascard}} = 16\mu\Omega$$

$$\Delta U_{max} = \Delta R \cdot I_{max} = 16\mu\Omega \cdot 50mA = 0.8\mu V$$

which is below DAQ sensitivity too.

*4) Measurement and Analysis Tools:* The measurements are run by a measurement program created with LabView 8. This program is accessing the measurement card of the measurement computer. It also generates the clock signal which is connected to the triggering input of the measurement. The data is read, processed and stored in a measurement file. The stored data is evaluated with scripts written in Matlab 2009a. Therefore, the data must be read from the file, analyzed and displayed properly.

This guarantees a maximum of flexibility, because a Matlab script can be changed very fast and easily compared to a LabView program and measurement setup. Once the data has been read, it can be processed very easily. Many different operations can be used, e.g. an FFT analysis to determine the main frequency of the power trace. The visualization can be used in a flexible way too.

### B. Energy Efficiency Measurement Setup

Differential measurements are performed at $50kS$. Blocks of 50 samples are averaged at a rate of $1kS$ for reducing the amount of data while preventing artifacts. Again, LabVIEW 8 is used for measurement automation and Matlab scripts are used for evaluation. EHS efficiency is calculated in addition to that.

*1) Efficiency:* We will evaluate the EHS efficiency

$$P_{input} = U_1 \cdot \frac{U_0}{R_1} \quad P_{ouput} = U_3 \cdot \frac{U_2}{R_2} \quad \eta = \frac{P_{output}}{P_{input}}$$

that partly depends on the voltage level of the DLC

$$W_{stored} = \frac{1}{2} \cdot C \cdot U_C^2.$$

Furthermore, it has to be mentioned that $\eta$ is affected by the EHS input efficiency, energy storage technology parameters and EHS output efficiency. While there is a close to linear increase in the DLC's input circuitry efficiency, we will discuss output efficiency separately. Output efficiency can easily be influenced by mote protocols while input efficiency is mainly determined by the EHS hardware setup.

Table I

DIFFERENT MOTES VARY SLIGHTLY IN THEIR POWER DISSIPATION. THE $null$ PROGRAM HAS BEEN SUBTRACTED FROM MEASUREMENTS OF OTHER POWER STATES. IN $t_{10}L_r$ A TIMER IS FIRED EVERY $10\,msec$ AND THE RED LED IS TOGGLED EVERY TIME THE TIMER FIRES. FINALLY, THE DIFFERENCE FROM SUMMING UP DISCRETE POWER STATES AND THE MEASUREMENT IS AT MOST $0.051\,mW$ FOR THAT PROGRAM.

| program | mote1 | mote2 | mote3 | mote4 |
|---|---|---|---|---|
| $null$ | $9.189\,mW$ | $9.219\,mW$ | $9.222\,mW$ | $9.146\,mW$ |
| $LED_g$ | $6.758\,mW$ | $6.695\,mW$ | $6.726\,mW$ | $6.671\,mW$ |
| $LED_r$ | $7.525\,mW$ | $7.586\,mW$ | $7.580\,mW$ | $7.565\,mW$ |
| $LED_y$ | $6.965\,mW$ | $7.062\,mW$ | $6.924\,mW$ | $6.878\,mW$ |
| $t_1$ | $3.422\,mW$ | $3.288\,mW$ | $3.391\,mW$ | $3.264\,mW$ |
| $t_{10}$ | $0.910\,mW$ | $0.883\,mW$ | $0.899\,mW$ | $0.853\,mW$ |
| $t_{100}$ | $0.124\,mW$ | $0.121\,mW$ | $0.117\,mW$ | $0.115\,mW$ |
| $t_{10}L_r$ | $3.715\,mW$ | $3.742\,mW$ | $3.742\,mW$ | $3.734\,mW$ |
| $Error$ | $0.048\,mW$ | $0.051\,mW$ | $0.048\,mW$ | $0.049\,mW$ |

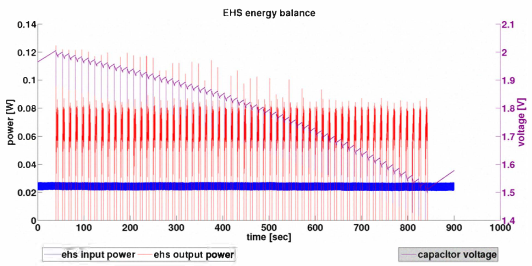



Figure 2. An EHS powered mote is activated 58 times in a single DLC discharge cycle. The overall system shows capacitive characteristics. This results in better efficiency of the EHS [10] when being operated at the upper voltage limits which can be influenced by software policies. DLC overcharge protection should be implemented in hardware to avoid damaging the system.

## V. RESULTS

Firstly, Mica2 PSM values have been measured. They will be discussed shortly. Secondly, EHS efficiency measurements will be analyzed in detail.

### A. Power State Model Measurements

Table I lists different power states that have been measured for four Mica2 motes. It turns out that different LEDs show different power dissipation. Furthermore, frequently fired timer events lead to an overhead significantly above linear increase. Measurements of combined usage of LEDs and timer show the applicability of PSMs for such simple components. Furthermore, power dissipation varies among different motes. Results for the $null$ program vary up to $0.36\%$ among motes and up to $4.84\%$ when running a program that fires its timer every $1\,msec$. The PSM measurement setup's applicability has been shown by the error below $1.37\%$ when combining PSM values analytically.

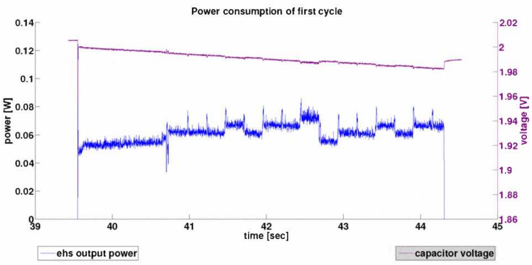



Figure 3. Supplying WSN motes by an EHS leads to peaky load profiles due to a pulsed EHS output stage. Furthermore, the measurement setup can only sample the EHS and mote with finite accuracy. Sampling frequencies have to be chosen for capturing the maximum DUT frequency and not exceeding the measurement computer's hard disk access speed accordingly.

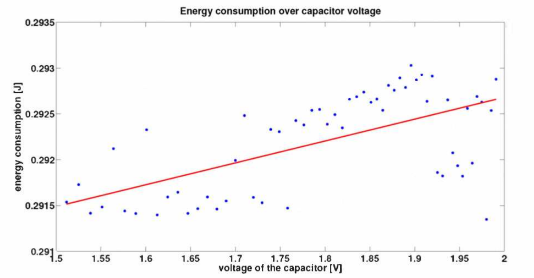



Figure 4. The mote's power dissipation profile is integrated over each activation cycle and a linear regression is plotted. Two effects have to be mentioned. Firstly, there is a slight increase in energy consumption, but this can be neglected compared to variable efficiency at the DLC. Secondly, the measurement results vary for up to 0.5%. This might prohibit optimizations from taking effect if they demand for very accurate measurements.

### *B. Energy Harvesting System Measurements*

Figure 2 shows a full discharge cycle of an EHS [10]. It can be seen that the voltage drop at the capacitor during mote activation (activation period) increases with decreasing capacitor voltage. This is caused by the EHS. The output efficiency is higher for a higher voltage of the capacitor. Measurements have shown that the voltage drop of the first activation period is $5.2\,mV$ and it is $13.1\,mV$ during the last activation period. The decrease of stored energy in the capacitor can be calculated with:

$$\Delta E_{DLC} = \frac{1}{2} \cdot C \cdot \left(U_{DLC,1}^2 - U_{DLC,2}^2\right).$$

So the energy is $104\,mJ$ and $202\,mJ$ respectively. The overall system's capacitive behavior leads to the first activation period's $\eta_{DLC,out}$ being $194\,\%$ as efficient as in the last one. Concluding from this output efficiency decrease over time, important networking protocols for negotiating duty cycle settings among different motes should be worked off at the very beginning of a discharge cycle. Additionally, this effect is amplified by the close to linear input efficiency increase for increasing capacitor voltage which is apparent in the EHS in [10] and in similar systems.

Figure 3 shows a single activation period of the mote. After powering on, the mote needs approximately 1 second to start the user application. Therefore, we suggest putting the mote to sleep instead of shutting it down. Otherwise, the initialization time's power consumption will significantly impact end-user performance for short activation times. Mote power profiles are then integrated over each single activation period which gives results that are plotted in Figure 4. Adding a regression line clearly shows the dependence of $U_{DLC}$. Furthermore, it shows that runtime PSM measurements are unreliable. Concluding, offline mote characterization is indispensable for systems that need to make predictions based upon accurate measurements including the mote's power dissipation.

Combing EHS measurement results from above, one can see that the different period's $\Delta E_{DLC}$ values result in a fraction different from that one for energy values per activation accordingly. The difference comes from the power supply attached to the system and from other hardware parts' efficiencies. The power supply provides an average power level of $24.19\,mW$ throughout the experiment which equals $\Delta E_{supply} = 343.0\,mJ$ per activation cycle approximately. We note that we do not give error bars, neither evaluate EHS efficiency results' variance or confidence intervals. So, this work is concerned with discussing efficiency trends that hold for other systems as well. Continuing the evaluation, the energy measured at the mote $\Delta E_{mote}$ using $ch2$ and $ch3$ is $291.0\,mJ$ in the first activation period and $292.9\,mJ$ in the last one. EHS output stage energy loss can now be calculated as

$$\Delta E_{loss,out} = \Delta E_{DLC} + \Delta E_{supply} - \Delta E_{mote}.$$

This evaluates to $156\,mJ$ for the first activation and $254\,mJ$ for the last one.

Summing up the findings when evaluating EHS efficiency for different voltage levels, all EHS components benefit from operating the system at the upper limit. We get:

- better efficiency for the DLC input charge pump,
- a better DLC output efficiency when supplying motes,
- reduced EHS output stage energy loss and
- only slightly increased mote power consumption on average.

Other effects might be more important for other types of EHSs, but for this type with capacitive characteristics the upper limit provides more efficient operation. Furthermore, this leads to more robust policies, because there is more room for energy thresholds.

## VI. Conclusion and Future Work

We present the design and implementation of an accurate measurement setup for WSN PSMs and EHS efficiency and discuss novel aspects of their impact on energy harvesting policies.

### A. Conclusion

The setup has been evaluated with laboratory power supplies, batteries and EHSs. PSM measurements have been shown to be feasible with the measurement setup. The worst case error of the accurate setup is below 2.1%. It turns out that the variation in power supply impacts the maximum accuracy of EHSs' measurements that they are not accounted for. Furthermore, we show how the system can be applied to EHSs and be used for characterization of their efficiency. Based on profiles from this automated measurement setup, we suggest to perform application parts with power peaks in the beginning of each discharge cycle due to its high efficiency and to send motes to sleep instead of shutting them off for duty cycling due to hardware setup times.

### B. Future Work

The system will be used to configure a WSN simulation and EHS modeling environment. An energy harvesting device emulation will widen the system's characterization capabilities even further. This will allow running variable simulation scenarios for EHS supplied WSNs in the future. Furthermore, complex energy harvesting test patterns can be generated for testing energy harvesting policies.

## References


[1] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," in *EmNets '05: Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 37–44.

[2] L. Girod, N. Ramanathan, J. Elson, T. Stathopoulos, M. Lukac, and D. Estrin, "Emstar: A software environment for developing and deploying heterogeneous sensor-actuator networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 3, p. 13, 2007.

[3] H. Wu, Q. Luo, P. Zheng, and L. Ni, "Vmnet: Realistic emulation of wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 277–288, Feb. 2007.

[4] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, "PowerBench: A Scalable Testbed Infrastructure for Benchmarking Power Consumption," in *International Workshop on Sensor Network Engineering (IWSNE)*, Santorini Island, Greece, jun 2008, pp. 37–44.

[5] J. Beutel, O. Kasten, F. Mattern, K. Römer, F. Siegemund, and L. Thiele, "Prototyping wireless sensor network applications with btnodes," in *First European Workshop on Wireless Sensor Networks (EWSN)*, 2004, pp. 323–338.

[6] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro power meter for energy monitoring of wireless sensor networks at scale," in *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2007, pp. 186–195.

[7] T. Trathnigg and R. Weiss, "Towards runtime support for energy awareness in wireless sensor networks," in *Notere - International Workshop on Wireless Sensor Networks*, 2007, pp. 49–54.

[8] E. Perla, A. O. Catháin, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "Powertossim z: realistic energy modelling for wireless sensor network environments," in *PM2HW2N '08: Proceedings of the 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*. New York, NY, USA: ACM, 2008, pp. 35–42.

[9] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, p. 32, 2007.

[10] P. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A measurement platform for energy harvesting and software characterization in wsns," in *WD '08: 1st IFIP Wireless Days*, Nov. 2008, pp. 1–5.

[11] G. W. Johnson and R. Jennings, *LabVIEW Graphical Programming*, 4th ed. McGraw-Hill Professional, 2006.

[12] M. Wendt, M. Grumer, C. Steger, R. Weiss, U. Neffe, and A. Muehlberger, "Tool for automated instruction set characterization for software power estimation," *on Instrumentation and Measurements*, vol. 59, no. 1, pp. 84–91, 2010.

# Poster Abstract: TOSPIE2: Tiny Operating System Plug-In for Energy Estimation

Philipp M. Glatz
Institute for Technical Informatics, Graz University of Technology, Austria
Philipp.Glatz@TUGraz.at

Christian Steger
Institute for Technical Informatics, Graz University of Technology, Austria
Steger@TUGraz.at

Reinhold Weiss
Institute for Technical Informatics, Graz University of Technology, Austria
RWeiss@TUGraz.at

## ABSTRACT

TinyOS2 (TOS2) is the state of the art operating system for wireless sensor network (WSN) programming. There are a number of testbeds and simulation tools for checking functional correctness and a few integrated development environments (IDEs) that support graphical user interface (GUI) and power profiling of WSN simulation as well. All these systems and environments are tailored towards design optimization of WSNs subject to functional correctness and energy conservation. Unfortunately, all these approaches lack self-contained support for energy harvesting systems (EHSs) which is the state of the art technique for tackling the energy conservation problem. Here, we present the design and implementation of a self-contained X-in-the-loop approach. TOSPIE$^2$ is an Eclipse plug-in that integrates TOS2 installation and compilation support, power profiles and EHS efficiency in simulation and measurement.

## Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Network Operations

## General Terms

Algorithms, Design, Experimentation, Measurement

## Keywords

Design, Development, Efficiency, Energy Harvesting, Plug-In, Power States, Simulation, Wireless Sensor Networks

## 1. ISSUES TACKLED BY TOSPIE2

Wireless Sensor Networks (WSNs) are applicable to a wide range of applications [13]. Though, they suffer dependability issues in that it is an error prone task to implement functional correctness and power aware behavior at the same time. While there is some support available for power profiling of WSNs there is no approach taken yet for integrating energy harvesting system (EHS) and device (EHD) simulation or even emulation. We provide a fully self-contained development environment with profiling capabilities in simulation and through measurements in hardware. The integration using an Eclipse plug-in allows using an X-in-the-loop approach for WSNs which is tested exhaustively with different types of hardware and software that is subject to power-optimization at different levels of abstraction.

TOSPIE$^2$ is an Eclipse plug-in and provides the following contribution to the field of WSNs. As a novel type of integrated development environment, it supports different versions of TinyOS2 (TOS2). Integrating state-of-the-art simulators and an accurate measurement setup allows for power aware WSN design and X-in-the-loop design: power state models (PSMs) and EHS efficiency models (EEMs) can be traced, tested and varied using a PSM/EEM database, EHS and mote hardware can be traced and partly be reconfigured on the fly and EHDs can be traced. Furthermore, we support automatic network scaling and reconfiguration in simulation and in hardware implicitly including effects like realistic packet loss in hardware. We improve over existing solutions in that we integrate all these capabilities into one single plug-in which can even be virtualized due to its remote interface. This allows a team of developers for independent testing using one single setup.


*IPSN'10*, April 12–16, 2010, Stockholm, Sweden.
ACM 978-1-60558-988-6/10/04.

## 2. RELATED WORK AND COARSE GRAIN CONCEPT

For giving a coarse grain overview of related work we consider hardware and software case studies as well as tools for development, simulation and profiling and also measurement and emulation setups including software, hardware and system in the loop design.

We use Mica2 motes [4], and motes similar to the TelosB architecture [9] and two EHS [3] approaches for evaluation. First evaluation runs have been finished for low power listening (LPL) optimization and PSM and EEM measurements. Different simulation environments for power aware design have been proposed and designed. We set up an environment similar to the one in [1] but including energy harvesting and many opportunities for measurements of hardware testbeds. Furthermore, we interface with the state of the art TOS simulator [7] and Avrora [10]. While [10] improves over [7] with [6] with cycle accurate simulation and power profiling, we improve over that with adding another simulation environment including EHS simulation and trade it for a higher level of abstraction. Similar to other IDEs for TOS we provide a user interface for using the TOS tool-chain. Furthermore, we add a database for capturing PSM and EEM characteristics. Simulations can automatically be scaled in size in the simulation environments and on hardware with adjacency matrices automatically hardcoded to nesC using Matlab. One of the common pitfalls when designing power

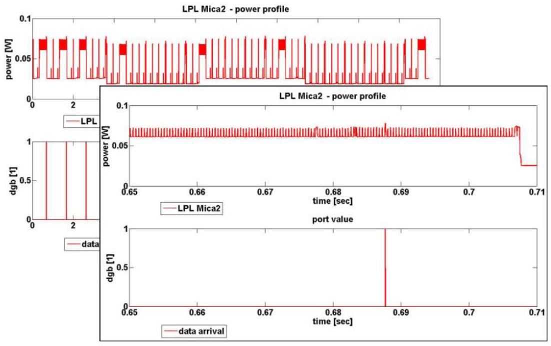



**Figure 1: The plot shows a high resolution Mica2 power profile when using LPL and toggling an LED.**

aware WSNs is that of inaccurate measurements and power profiling in simulation. Especially the radio component is difficult to model with its different configurations and possible interference. Therefore, we integrate an accurate and high speed measurement setup [2] for PSM and EEM measurements. Updating the database with these values and configuring and even validating power dissipation profiles from simulation gives us the chance for system in the loop (SIL) methods. The SIL concept is similar to the approach taken in [12] and uses a similar hardware setup as well. It allows testing and configuring simulation environments, mote applications and EHS protocols. Different operating points for the EHS can be tested through EHS and mote configuration. Multiple EHDs of different types can be used with the setup where a lot of traces exist. These and EHD modeling in previous work [11] are being used for work in progress that deals with construction of a configurable EHD emulation. It is being implemented on the same measurement setup again using LabVIEW [5] as the setup for measuring PSM and EEM. It follows an approach similar to the one by [8], but it uses the irradiation modeling from [11]. Implementing EHD emulation is given priority over battery emulation due to the high impact of correctly operating an EHS [2].

## 3. RESULTS AND CONCLUSION

TOSPIE[2] is an Eclipse plug-in for developing power aware WSNs. TOS2, Avrora and TOSSIM Support and front-ends are integrated as well as a novel simulation environment for EHS enhanced WSNs and a PSM and EEM database and measurement setup. Motes and EHSs can be measured very accurately where the EHS itself can be used for profiling EHDs in the field. Using these profiles and concept from previous work builds a basis for EHD emulation. The simulation and measurement setup allows to automatically scale for differently sized networks where the testbed size is only limited by the number of available hardware entities. The measurement setup has a worst case error of 2.1% using measurement cards with $10^5$ and $10^6$ samples per second respectively. Different levels of abstraction in simulation face different sampling rates from traces of fine grained LPL measurements (Figure 1) to coarse grained EHS efficiency (Figure 2). The main result from using the approach and setup is that TOSPIE[2] automation can greatly reduce development time and allows for a novel approach in testing

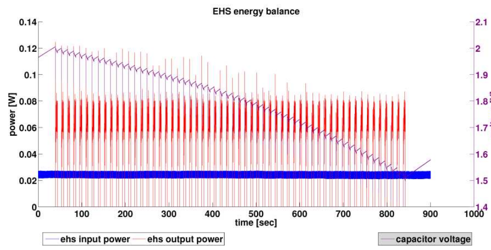



**Figure 2: Sampling the EHS with a Mica2 mote attached at a lower resolution does not significantly impact EHS efficiency calculation accuracy.**

of EHS networking policies in simulation and measurement to evaluate dependability and energy conservation at once.

## 4. REFERENCES

[1] J. Glaser, D. Weber, S. A. Madani, and S. Mahlknecht. Power aware simulation framework for wireless sensor networks and nodes. *EURASIP J. Embedded Syst.*, 2008:1–16, 2008.

[2] P. M. Glatz, L. Hoermann, C. Steger, and R. Weiss. A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency. In *PerSeNS*, 2010.

[3] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss. A measurement platform for energy harvesting and software characterization in WSNs. In *Wireless Days*, 2008.

[4] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization, 2001.

[5] G. W. Johnson and R. Jennings. *LabVIEW Graphical Programming*. McGraw-Hill Professional, 2006.

[6] O. Landsiedel, K. Wehrle, and S. Gotz. Accurate prediction of power consumption in sensor networks. *EmNetS-II.*, 2005.

[7] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *SenSys*, pages 126–137. ACM, 2003.

[8] C. Park, J. Liu, and P. Chou. B♯: a battery emulator and power-profiling instrument. *Design & Test of Computers, IEEE*, 22(2):150–159, March-April 2005.

[9] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *IPSN*, 2005.

[10] B. L. Titzer, D. K. Lee, and J. Palsberg. Avrora: scalable sensor network simulation with precise timing. In *IPSN*, 2005.

[11] C. Trummer. Device evaluation for energy harvesting devices for higher class tags. Technical report, Institute for Technical Informatics, TUG, 2006.

[12] M. Wendt, M. Grumer, C. Steger, R. Weiss, U. Neffe, and A. Muehlberger. Tool for automated instruction set characterization for software power estimation. *IEEE Transactions on Instrumentation and Measurements*, 59(1):84–91, Jan. 2010.

[13] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12), 2008.

# Architecturing Resource Aware Sensor Grid Middleware: Avoiding Common Errors in Design, Simulation, Test and Measurement

**Philipp M. Glatz, Reinhold Weiss**
*Institute for Technical Informatics, Graz University of Technology, Austria*

## ABSTRACT

Resource aware sensor grid middleware is subject to optimization of services and performance on one side and has to deal with non-functional requirements and hardware constraints on the other side. Implementing different applications and systems on different types of hardware and architectures demands for sophisticated techniques for modeling and testing. This chapter highlights common misconceptions in design, simulation, test and measurement that need to be overcome or at least be considered for successfully building a system. Rules of thumb are given for how to design sensor grids such that they can easily be simulated and tested. Errors that are to be expected are highlighted. Several practical issues will be discussed using real world examples. A sensor grid utilizing network coding and duty cycling services serves as an example as well as a multi-application middleware and a localization system. The approach shows how to implement performance optimizations and resource awareness with a minimum of negative impact from mutual side effects. This type of view on system development of sensor grids has not been looked at before in detail. Therefore the reader will get valuable insights to state of the art and novel techniques of networking and energy management for sensor grids, power profile optimization, simulation and measurement and on how to translate designs from one stage to another.

## INTRODUCING THE COMBINATION OF GRID MIDDLEWARE AND WIRELESS SENSOR NETWORKS FOR PERVASIVE SERVICES

Computing anything, anywhere at any time as envisioned by Weiser M. (1991) gives a good specification for what is targeted by ambient computing technologies. The challenge when setting up such ubiquitous system is twofold. First, functional constraints need to be satisfied for providing services at a given quality of service (QoS) or for achieving a given end-user performance in terms of throughput or sampling rate. Second, a suitable technology needs to be selected and integrated for setting up the system satisfying non-functional constraints like ambient integration or low power operation.

For achieving a cost efficient solution one needs to apply suitable modeling techniques, simulation environments and pre-deployment characterization using modeling, simulation and testbeds for the chosen technology. Yick et al. (2008) and Akyildiz et al. (2002) give surveys on wireless sensor network (WSN) technology. Using WSNs as an enabling technology for pervasive computing, services can be provided by means of sensor grids (Lim H.B. et al., 2005). Tham and Buyya (2005) give an example of a hierarchically organized sensor grid using Mica2 motes. These motes have been introduced by Hill and Culler (2001) and will serve as target technology in this paper as well. Though novel platforms have been introduced like TelosB and others Mica2 motes provide a valid platform for planning and implementing development and deployment techniques and different kind of optimization. An example for how to implement the technique called low power listening on the different platforms has been shown by Moon et al. (2007).

Similar to the twofold structure of the challenge, the approach presented in this paper will take two perspectives as well.
On one side, functional constraints for sensor grid services will be discussed. A middleware layer will be presented considering two often used communication paradigms. It will be designed and optimized suitable for worst case assumptions in the middleware, the networking and the media access control (MAC) layer. Functionality will be tested using modeling and simulation-based approaches.
On the other side WSN hardware will be discussed. Its implications for the maximum end-user performance that can be achieved will be related to the MAC layer and power state models (PSM) as well as higher level protocols impact on the energy balance. Instead of only running simulations, the main aspect will be on how measurement systems and testbeds can be set up.

## Chapter Objectives

Both ways mentioned need to be combined for arriving at a complete analysis of a sensor grid prior to deployment. Apart from introducing and discussing novel and state of the art solutions for different levels of abstraction a main focus will be given on modeling, simulation, measurement and test errors.
Functional and non-functional properties of the system may experience different kinds of errors. We give a novel description of how to avoid or at least recognize these errors and compare different errors' impacts.
Especially for WSNs the modeling of the wireless channel is a tough challenge. Usually, accurate characterization of wireless scenarios can only be done using in-network measures when the system has been deployed. We will introduce and discuss means of modeling scale dependent issues in laboratory sized environment and discuss its implications on errors when profiling testbeds compared to online characterization of deployments.
Figure 1 gives an overview on different levels of abstraction that are used in the chapter. Environmental Conditions will be will be considered as far as energy harvesting is considered. Power supply issues are the main issue then as are how the energy reservoirs are impacted by power dissipation given some MAC order, routing functionality and middleware services which will be simulated and measured.

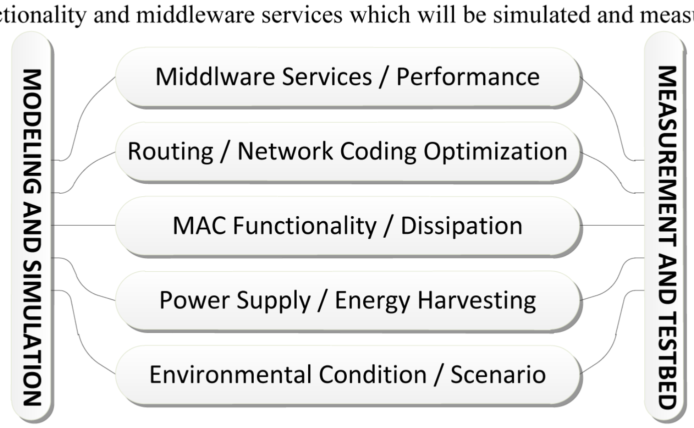



*Figure 1. During the different phases of development the design has to validated at different levels of abstraction. Modeling, simulation, measurement and profiling in testbeds will be considered. The maximum end-user performance that can be achieved depends on different levels of abstraction of the network stack. A cross-layer perspective will be taken in that interaction among different levels of abstraction will be considered.*

### Chapter Organization

Related work overviews state of the art methods and technologies for setting up mesh structured sensor grids briefly. Optimization issues for hardware components and software implementations will be discussed and simulation and measurement environments will be overviewed as well. Next, the design and implementation tradeoffs will be discussed. Results will then compare different optimizations' benefit to inaccuracies that may result from modeling, simulation, measurement and test errors. These aspects will be discussed on a per topic basis, because it is assumed to be more convenient for the reader to have a brief introduction to the different topics as it is needed instead of summing up everything at once. Finally, we conclude giving guidelines for which aspects are the most valuable ones when being given decent consideration.

## RELATED WORK FOR LOW POWER WIRELESS SENSOR NETWORKS

First, energy harvesting systems (EHSs) for WSNs will be explained. Networking protocols based upon state of the art MAC protocols are explained. Network Coding will serve as an example for optimization at the networking layer given mesh structured WSNs. Simulation environments and measurement setups are discussed as well as other work discussing errors in simulation and measurement frameworks as well.

### Energy Harvestings Systems for Wireless Sensor Networks

EHSs are the state of the art enabling technology for long lived WSNs. Furthermore, recent developments for energy management for advanced resource awareness are based upon properties and effects of using EHSs.

Different EHSs for WSNs have been presented. A well-known example is Heliomote (Lin et al. 2005) which attaches solar cells to Mica2 motes. Based upon such platforms the term energy neutral operation has been introduced by Kansal et al. (2004) and has been refined in Kansal et al. (2007). The idea is to estimate future EHD power profiles based on past modeling periods. As soon as the variation of these values stays below a certain value compared to estimated values the estimation process of the average energy that is to be expected per modeling period is stopped. From that point in time the system can continue operation in energy neutral mode. Despite the fact that these enabling results for EHSs may allow for long-lived operation, this type of EHS-enhanced WSNs is not yet error free.

Coarse grain models for energy storage architectures, harvesting devices, converter circuitry and other hardware related issues are already available in literature, though we are still missing an unified approach for optimizations of existing architectures. But, what is more impacting usability of such systems is that the community still lacks a pool of protocols for actually dealing with the different situations that the system might transit into.

Energy neutral operation can only continue to deliver a given end-user performance if EHS protocol overhead – especially for synchronization – does not demand for too much additional energy to be used and especially if environmental conditions do not change too fast. Glatz et al. (2008) give an example that common harvesting theory cannot cope with. If conditions change too fast the network may become unstable due to leaving the duty cycle operating point that has been calculated for energy neutral operation (Kansal et al., 2004). So, threshold are introduced when an EHS double layer capacitor (DLC) approaches the limits of its operating range. Glatz, Hörmann, Steger and Weiss (2010) give insights into EHS efficiency on a more fine grained level. Especially for EHS with DLCs it holds that the system's efficiency increases when being operated on a higher values of its energy storage.

While there is no suitable approach yet for giving applicable bounds of environmental or other conditions, there are still other aspects from where we can start from. So we start introducing a special sort of networking optimization applicable to EHS-enhanced WSNs such that it is even applicable to further extend the view from only energy conservation to energy management given the constraint that EHS technology is being used. As we have no complete bounds for exactness or reliability for EHSs, what we can do is to describe the optimization exploiting the context set up by EHS-enhanced WSNs. Describing the optimization itself as well as the ways for how to simulate it will allow us to at least give estimates on how exact a simulation or measurement can be which will allow to further develop the issue throughout this chapter.

### Network Coding for Energy Conservation Management in Sensor Communication

As the enabling technologies of EHSs and state of the art motes have been introduced the next level of abstraction according to Figure 1 must be the MAC layer.
The latest version of TinyOS2 when writing this chapter is capable of power aware communication in that a low power listening technique can be supported (Moon et al. 2007). Given such an approach the ratio of average power dissipation for sending a message compared to listening to the message can be estimated to be approximately 2. Measurements that have been used by Glatz et al. (2009) validate this assumption. The first step of modeling worst case conditions on the MAC layer for setting up a cross layer view based upon that starts here as well.
The LINDONCS approach by Glatz and Weiss (2009). Allow for switching on and off a network coding approach that has initially been introduced by Ahlswede et al. (2000). Following this approach, for communication middleware network coding may be treated as a service that can be switched on and off as a service that is being requested. One way of doing so might be to use a high level state machine which offers network coding as one of its services. It might be used to improve dependability. Another way would be to have network coding implemented on sensor nodes as a means of energy usage optimization as mentioned by Glatz, Hein and Weiss (2009). The grid structure used in that approach will serve as an example for sensor grid architecture. This approach that will serve as an example here can also be used to have energy management implemented upon it as presented by Glatz, Loinig, Steger and Weiss (2010).

### Wireless Sensor Networks Simulation, Testbeds and Measurement Experiences

There are prominent examples for WSN deployments failing due to false assumption being made ahead of the deployment. The most well-known example may be the Great Duck Island Experience as reported by Mainwaring et al. (2002). As of 2010 novel sniffing techniques for debugging WSNs start deserving more and more interest as mentioned by Kay Römer in his keynote speech at PerCom 2010. Many different approaches for modeling, sniffing and debugging in the context of BTNode style of motes (Beutel et al. 2004) are under active development.
Other teams are currently dealing with the issue as well. Especially the group related to the PowerBench setup (Haratcherev et al. 2008) gives insights to what can be gained in terms of visibility when using real world testbeds. Based upon that Glatz, Hörmann and Weiss (2010) discuss the accuracy that is possible with measurement based approaches compared to simulation based ones. Apart from power state models the invaluable information of EHS efficiency models is discussed for different operating points as well. The main outcome is that for EHS-enhanced WSNs it must not only be considered to draining different motes' energy reservoir such that they all run out of energy nearly at the same. Also for all other operating points of the energy reservoir the efficiency varies that drastically – it doubles for the EHS presented by Glatz et al. (2008) – such that it may impose severe constraints on the maximum end-user performance that can be achieved if not being considered carefully.

The two most commonly used tools for profiling WSN motes' power dissipation ahead from deploying it may be AEON based on Avrora by Landsiedel et al. (2005) and PowerTOSSIM based on TOSSIM or PowerTOSSIM-Z for version 2 of TinyOS as presented by Perla et al. (2008). The problem with these simulation-based approaches is that though they have an accuracy usually below 20% for the overall energy consumption of the motes and below maybe 5% for shorter parts they still do not capture the dynamic behavior well. Unfortunately, short term variation in the power profile may completely render power profiles useless just because the hardware components may behave differently due to peaky power profiles that are not considered when averaging over time.

## ESTIMATING AND PROFILING POWER-AWARE SENSOR GRID PROTCOLS

Related work has shown several examples of different hardware and software platforms for WSNs and means of modeling, developing and testing them. Furthermore different experiences show lessons that have to be learned with their outcomes considered when rolling out WSN applications. Related work has been discussed that is concerned with important steps when developing and profiling energy conservation optimization measures.

For explaining what can be expected from applying optimization measures for energy conservation compared to how much different errors impact profiling of these measures the concept splits in two parts. First, we will define a setup of a WSN middleware for mesh structured sensor grids. State of the art networking methods will be discussed for low power WSNs with energy management. Next, we discuss a methodology for modeling, simulating, estimating and measuring. A detailed step-by-step analysis will be given for translating an application design to a running deployed system. A multi-application middleware and a localization system – both implemented on Mica2 WNSs – serve as real world examples as well for keeping the approach from being academic. Figure 1 gives an overview of different issues that will be considered when translating an application design to a deployed system. At different stages of development different errors in terms of traced functionality of power profiles have to be considered.

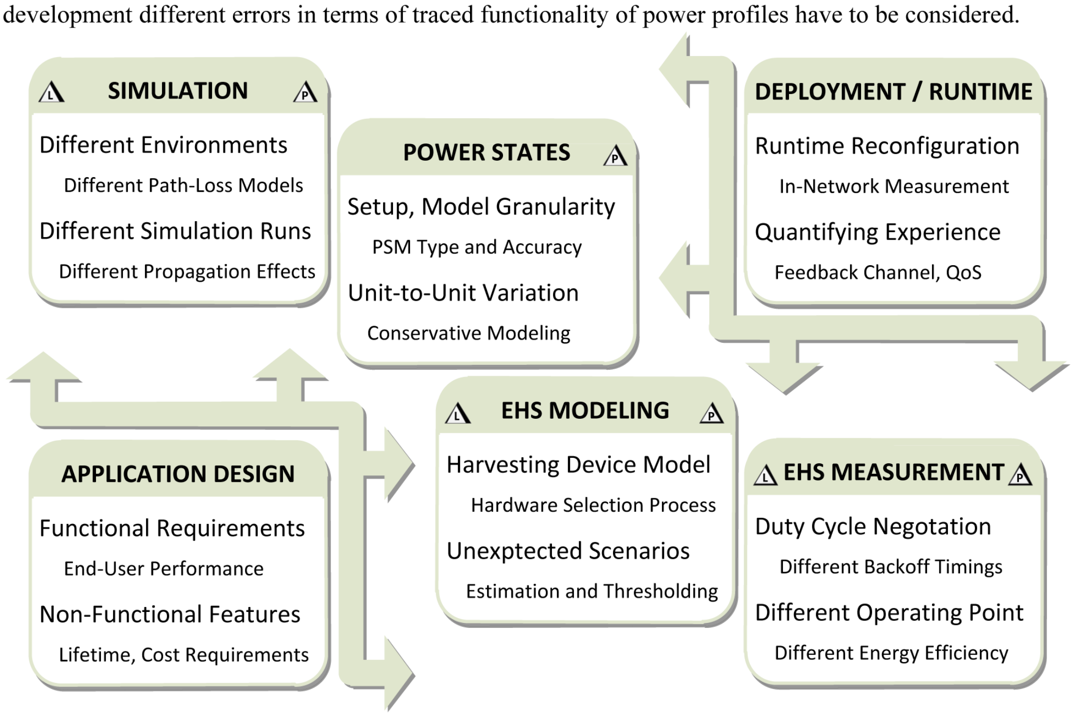



*Figure 2. We start with application design phase. Along the way that makes us arriving at a running application that has been deployed different phases of development may introduce logical and profiling errors. That is what will be neatly narrowed down in this chapter.*

We set up a concept for guiding developers through the process of finding which parts of the problem should be considered based on their impact. For the example at hand - a sensor grid middleware with network coding - we will consider the following issues: The networking itself and how messages are being forwarded, the optimization that is possible with applying network coding, the errors that are to be expected from simulation and testbed experience and ways for applying a robust development method.

**Networking in Mesh Structured Sensor Grids**

We set up a 10x10 mesh structured WSN using Mica2 motes similar to the structures introduced by Glatz, Hein and Weiss (2009). Figure 3 depicts the topology, possible flows of information and the information that is needed for coding information flows in the network. Transmitter nodes T1-T6 send information flows F1-F6 to sink nodes F1-F6. All inner nodes Xk of the network need to send two messages if routing is used but only need to send one message if XOR-coding is applied to all three incoming messages except for X1 where only two messages need to be combined. R1-R6 can combine three incoming messages using XOR-coding as well for arriving at the messages initially injected into the network.

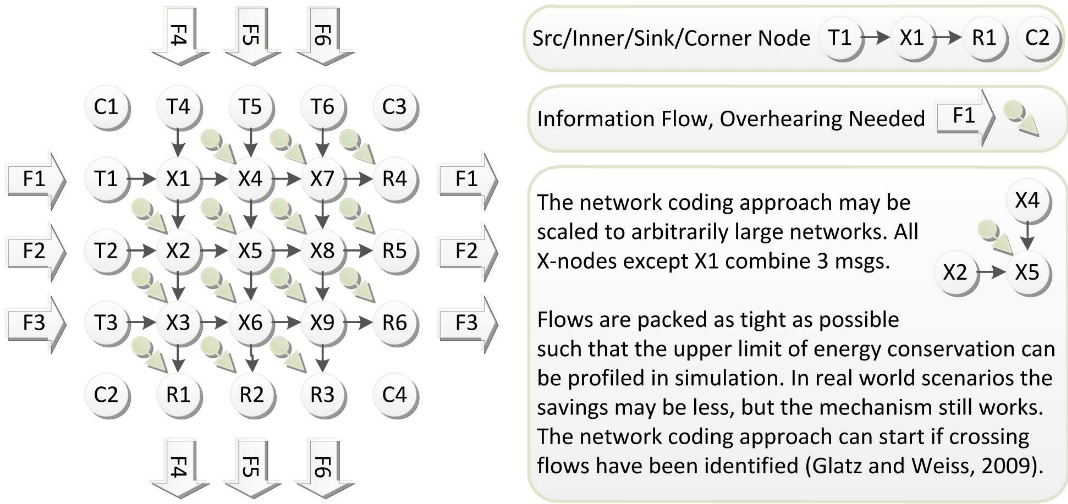



*Figure 3. For mesh structured WSNs the energy for up to 50% percent of inner nodes' messages' transmissions can be conserved with network coding compared to routing.*

From an analytic point of view one could evaluate the performance of network coding compared to routing using the lifetime of the grid before batteries need to be replaced. A common criterion is to check the number of messages that need to be sent as the radio module is even considered the most energy hungry component of a mote sometimes (Rincón et al., 2007). Plotting the energy that is conserved due to the reduced number of messages to be sent averaged over the network for differently sized networks one arrives at the results plotted in Figure 4.

The problem with that type of modeling is that we do not consider what happens if single nodes run out of energy. Taking EHSs into account – which serves as a basis for the energy management scheme presented by Glatz, Loinig, Steger and Weiss (2009) – different nodes will be supplied and drained differently. Shutting off a single node due to energy shortage influences neighbouring nodes as well. First, the node cannot participate in message transmission any more. For routing as well as for network coding all messages of one session that are to be sent along a route using that node might be dropped. Next, energy management schemes or other types of higher level protocols including energy harvesting duty cycling might get stalled.

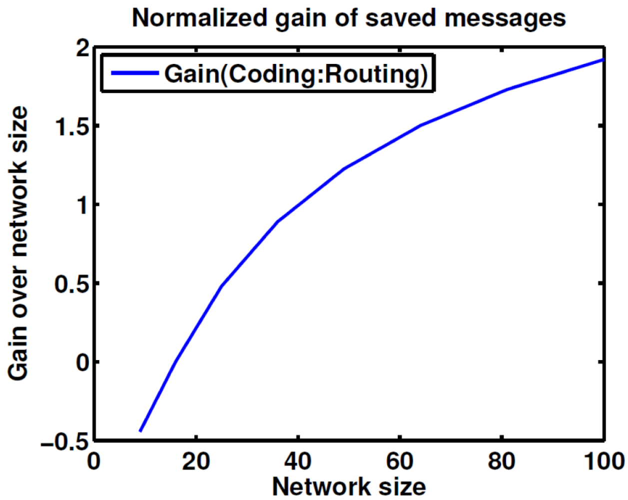



*Figure 4. For differently sized quadratic grid networks we plot the gain from network coding over routing as the ratio of messages that need to be sent when injecting flows of information as shown in Figure 3. The gain converges at 2 as expected.*

For considering these effects as well we need to analyze how the spatio-temporal variation of mote's residual energy is affected. For doing so we apply an analytic model and combine it with measurement results from Mica2 motes using low power listening to arrive at a ratio of sending cost to receiving cost of approximately 2. While this factor is technology and protocol dependent it suits well to characterize state of the art WSN technology and MAC protocols. For test purposes we have set up a simplified 10x10 network with a grid structure as introduced in Figure 5. It is simplified in that only 4 flows are injected in only two sessions and no network coding is used yet. This allows concentrating on other issues in more detail. The MAC order is assigned statically such that each node is assigned a MAC slot in each time slot. So, all nodes can transmit virtually collision free without the need for difficult synchronization protocols. While allowing full overhearing, where each node's radio may impact each other node's radio, degrades performance per energy spent, it opens possibilities for validating performance results with translating them from simulations to testbeds. MAC slots are ordered starting with the lower right corner node C4, continuing right receiver nodes, the upper right corner node C3 and all other columns bottom up and from right to left. Motes are numbered the other way round starting the upper left corner node C1, left sending nodes, C2, the leftmost upper sending node and continuing columns top down from left to right. Following this definition and starting with node identifier 1 we define source nodes 3, 6, 21 and 51 and sink nodes 93, 96, 30 and 60. This setup suffices to set up routes during a flooding phase that do not intersect for parallel information flows. Flooding is being initiated from sink nodes and all flooding messages are broadcasted to neighboring nodes in an up to 8 nodes neighborhood from a logical point of view.

Arriving at the simplified model, we have more room to consider effects that are hard to track with more detailed scenarios. Therefore, we include the initialization phase and a power model as well. While the TOSSIM simulation environment by Levis et al. (2003) provides full and fast functional simulation we annotate events for the most important components impacting a node's power dissipation profile. According to Mica2 hardware measurements using the setup provided by Glatz, Hörmann, Steger and Weiss (2010) we annotate 1.88 mJ for sending and 0.94 mJ for listening to an incoming message that are removed from the simulated energy level. Assuming small scale EHDs we add 2mW on average.

One of the main issues when comparing different approaches is to provide conditions that stress both - routing and network coding - equally strong. One issue has already been discussed that makes comparisons among approaches that complicated. It is packet loss. Packets that are not acknowledged by overhearing forwarding of unique identifiers marking messages are resent. The maximum number of resends is limited by 5. In Figure 5 the change of residual energy for all motes is plotted according to a TOSSIM simulation that is annotated with the power model. All nodes at the perimeter of the network remain at the positive side of the energy balance. They receive less impact from overhearing in up to 8 node neighborhoods. The residual energy depression for transceiver nodes in the network becomes perfectly flat if flows are packed tightly as shown in Figure 3 if network coding is applied. In other words the messaging workload can perfectly be balanced among nodes given that no messages need to be resent. Further experiments that incorporate the effects of accessing persistent memory as well have been implemented, but results are not considered here. Though they provide the basis for the energy management approach by Glatz, Loinig, Steger and Weiss (2010), they do not further provide to the issues targeted by this chapter.

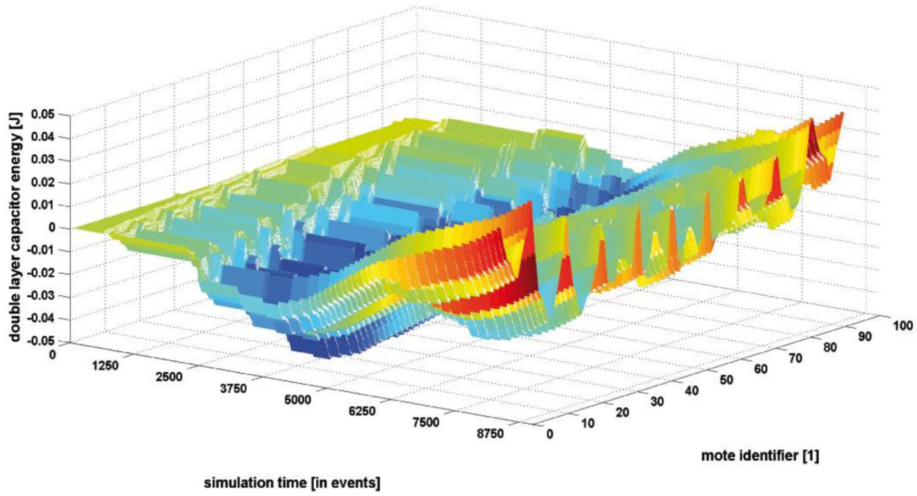



*Figure 5. The variance of the EHS-enhanced mote's energy balance is plotted differentially for the DLCs residual energy according to the power model which has been annotated for the TOSSIM simulation environment. Residual energy is plotted for different motes of a 10x10 grid over time where TOSSIM events serve as a timeline. The initial depression in of residual energy comes from flooding the network from the sink nodes 30, 60, 93 and 96. After that the surface starts to increase due to power from EHDs with another short depression for sending 2 sessions of 4 messages each.*

## Preparing Realistic Simulation Scenarios and Resulting Trade Offs

While introducing the simplified model in the previous section some assumption have been made that make life easier when it comes to translating simulation results to hardware testbeds. We have allowed for full overhearing for being able to run the same software in simulations and directly on hardware as well which is invaluable for realistic testing. It has also been mentioned that for being able to do that the price we pay is that we do not measure the exact power profile, but we at least we know exactly what we measure. That is what will be explained in this section. We start with the system setup and will then present as special situation where one can clearly see the differences that comes from sending or listening to single messages. For the sending cost we will consider a setup-specific question while listening cost will be related to that by exploring a technology dependent factor. The factor has been profiled on real hardware to be in the range of 2.1 to 2.3 for reasonable modes of low power listening as long as single messages are considered with the additional power for transitioning the radio module to receiving or sending averaged over time.

Starting with the setup of testing the multi-application middleware, Figure 6 shows a setup that has been chosen for profiling the congestion characteristics of the middleware. Such structures can be used to support an aspect of grids that has not been considered for sensor grids before. Handling several applications in grid systems at the same time is called virtual organization as introduced by Thonhauser et al. (2010). The idea is to allow switching forth and back between different types of behavior of grid nodes. It is similar with the multi-application middleware for WSNs. Different routing behavior can be achieved by different needs that are posed by different applications. Different sensing can be accomplished and applications may even decide to yield control to other motes as well. Due to the fact that the development is in its early stages of being characterized it has not been set up with the same structure as the network coding approach that has been presented. Instead, we present a setup that can be used for bandwidth and delay characterization of a network. Future use for implementing sensor grids based on that architecture will obviously be possible similar to what is proposed in terms of virtual organizations.

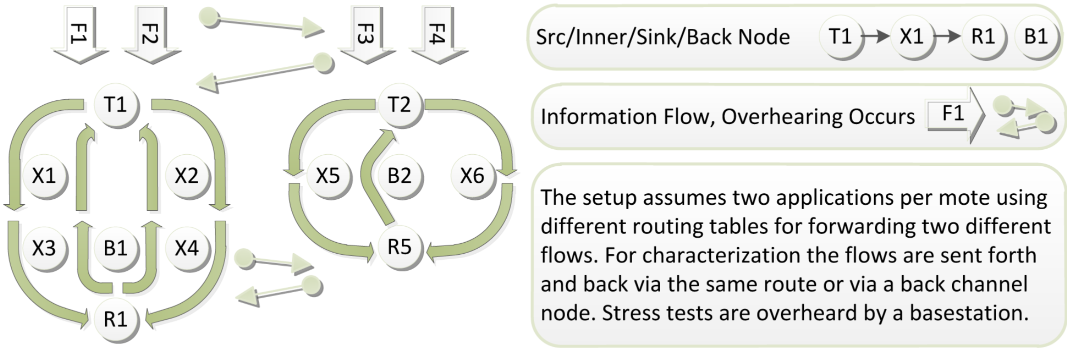
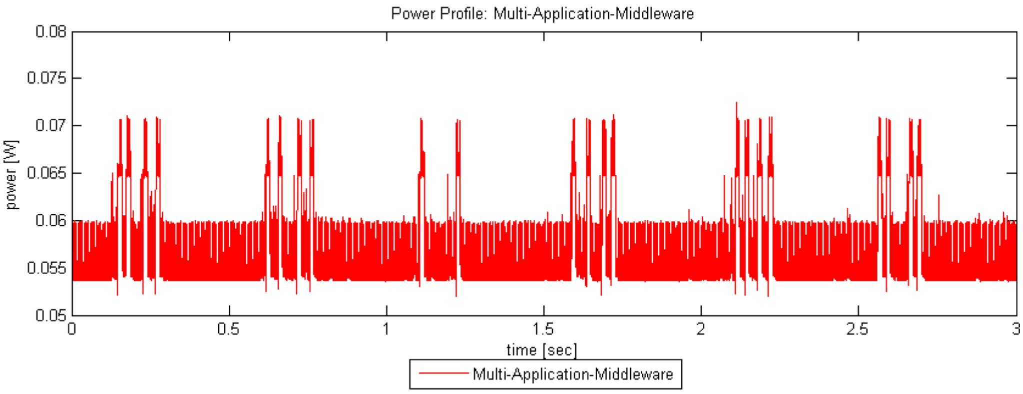



*Figure 6. The multi-application middleware has been simulated using Avrora with AEON (Landsiedel et al. 2005) and has been implemented on Mica2 motes. Motes have been profiled using an accurate measurement setup as well. Running different experiments in parallel leads to overhearing which allows gathering insights into errors of estimating and measuring power dissipation and networking protocols.*

Again these experiments have been run using full adjacency matrices as far as the physical channel has been concerned. The middleware has been configured to allow full communication as well, but the applications create different logical channels. In the 8-node setup – which is 7 nodes and a sniffing base station – Flows F1 and F2 are injected at T1 and the applications that reside on the motes are configured such, that F1 is routed T1-X1-X3-R1 and back via the same path again. The 6-node setup is configured such that R5 sends back the flow via B2. These two choices have been considered for being able to compare the effect of sending back an acknowledgement (ACK) via the same channel compared to the cost of all nodes overhearing each other for reasons explained before. It turns out that the setup does not improve in terms of the data rates that are still acceptable before packet loss starts to increase though nodes B1 and B2 are selected as the logical back channel while the delay does not change significantly as well. This holds for both sides of the routes where different routes are used by applications with different message buffer size. X2, X4 and X6 have a buffer size of only 2 while all other nodes are using a message buffer with 10 elements. What can be observed is that there are strange effects in the experiment due to other reasons.

## A Power Profile Analysis of the Multi-Application Middleware

Two laboratories have been used in parallel for profiling the middleware in terms of throughput and delay and in terms of power dissipation. Putting lots of effort into designing and characterizing robust middleware, test and measurement design and making simulation and measurement comparable, we finally missed to have all motes switched off and locked away that have not been expected to be included in a given set of experiments. Tracking down some strange behavior that occurred from time to time we have found out that conditions may exist where communication between motes situated in the two different labs is possible. A NACK-based protocol has been implemented that resends messages if it does not hear messages that it has sent being forwarded further on. From time to time messages did not get resent despite the fact that there was no designated receiver that could have signaled an ACK. For this special case, when motes from the other laboratory got overheard by transmitting nodes that have been connected to the National Instruments setup by Glatz et al. (2010), will be used for depicting the difference in actually dissipated power if a single message need not be transmitted



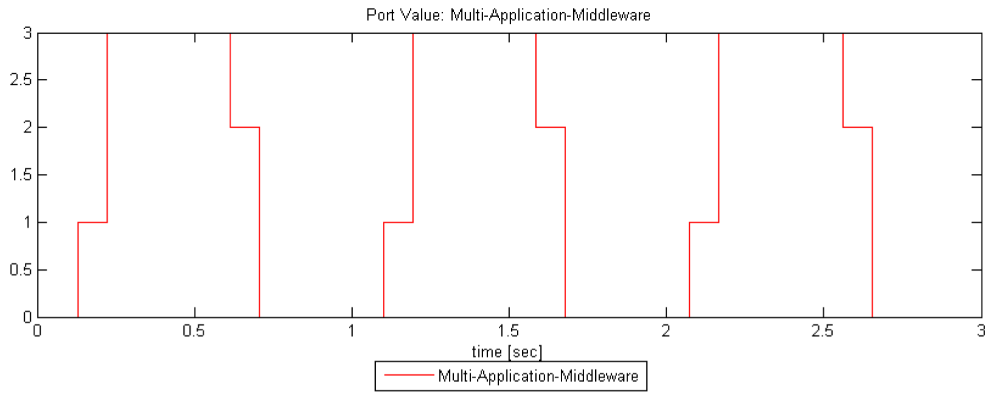




*Figure 7. The multi-application middleware sends counter values and resends messages once if no ACK is received. In the sending period between 1.0 and 1.5 seconds the applications send their messages to the middleware layer which can be seen by toggling the LSB at the digital debug output for application 1 and LSB+1 for application 2. In that special interval no resending occurs though resends would be expected.*

## Wireless Sensor Network Power State Models Looked at in Detail

Figure 7 shows the Mica2 power profile and the digital debug output that marks time stamps when applications pass their messages to the middleware. So, switches from 0 to 1 and 3 to 2 indicate that application 1 has sent a message and switches from 1 to 3 and 3 to 2 indicate that application 2 has sent a message. The overall trace has lasted for 60 seconds where the power profile has been sampled with 100 kS with a National Instruments PXI-6221 data acquisition card.

| *Program Part* | *Mean Power* | *Std. Dev.* | *Length* | *Abs. Error* | *Rel. Error* |
|---|---|---|---|---|---|
| *Full Program* | *56.089 mW* | *3.588 mW* | *59500 ms* | *1.236 mW* | *2.20 %* |
| *3 Second Segment* | *56.089 mW* | *3.588 mW* | *3000 ms* | *1.236 mW* | *2.20 %* |
| *1. 4 Msg Block* | *56.200 mW* | *3.713 mW* | *500 ms* | *1.238 mW* | *2.20 %* |
| *1. 2 Msg Subblock* | *58.572 mW* | *4.958 mW* | *92.6 ms* | *1.287 mW* | *2.20 %* |
| *1. 2 Msg Block* | *55.505 mW* | *2.879 mW* | *500 ms* | *1.224 mW* | *2.21 %* |
| *1. 1 Msg Subblock* | *56.686 mW* | *4.083 mW* | *92.8 ms* | *1.248 mW* | *2.20 %* |
| *No Radio Send* | *54.744 mW* | *1.382 mW* | *9.7 ms* | *1.208 mW* | *2.21 %* |
| *Full Radio Send* | *65.806 mW* | *1.223 mW* | *9.7 ms* | *1.436 mW* | *2.18 %* |

*Table 1. The 3 seconds block starts after 10 seconds of the full program. The first messages' power profiles are averaged over 500 ms. The first two radio activations, marked by the debug transition 0 to 1, is a subsequence of these 500 ms. The first block of 2 activations without resend after 1 second is profiled as well as is the subsequence of the first single radio activation. Finally, a two scheduler switch length has been profiled where the radio is active but not sending and full activation of the radio is profiled as well. The latter two traces can be found in Figure 8.*

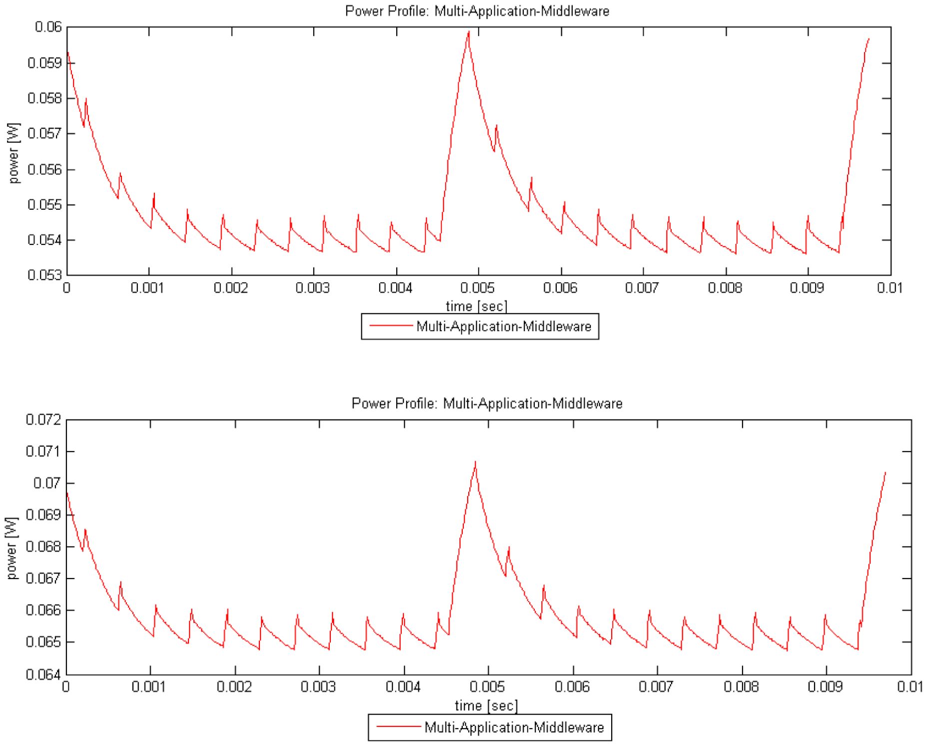



*Figure 8. Zooming into the power trace reveals the timed scheduler behavior for a typical low behavior at 1.00206 secoands of Figure 7 and the lowerplot shows the behavior when sending at 1.01179 seconds of Figure 7. Approximately every 5 ms it checks if there is a task pending that has to be dealt with.*

In Table 1 we provide characteristics of different parts of the power profile using their mean dissipation, the standard deviation, absolute error and relative error. First, the 3 seconds subsequence of the overall 59.5 seconds shows the same behavior as the full length trace. For depicting the difference of having a resend or not, suitable parts of the trace are chosen. As the power trace shows quite some variation as can be seen in Figure 8 we have chosen specific bounds for all parts to keep them comparable. Figure 8 depicts the traces that have been used for profiling the power dissipation of the radio send operation. Two peaks of the scheduler timer that fires periodically every 5 ms are chosen. In case the subsequence that is chosen for evaluation may be chosen arbitrarily short anywhere in the trace, the error can obviously rise above 5mW or 50 % of the average additional power dissipation that occurs when the radio is switched to sending mode. Furthermore, Figure 8 shows that both plots match very well and it looks like the only difference in the two power traces is the offset that is due to transmission power dissipation. So, the circuitry of the radio, that is switched on additionally in this case, seems to have no further side effects. Though, respecting the periodicity is still important as mentioned before. The approach presented by Glatz et al. (2010) allows for applying frequency analysis for automated selection of suitable subsequences. Table 2 summarizes measurements of a null program, different programs activating LEDs and programs firing timer events at variable frequencies. The empty TinyOS2 program has been measured and its power dissipation is subtracted from all other measurements. The different LEDs on the Mica2 mote consume a different and significant amount of power compared to approximately 11 mW that can be calculated from Table 1 for starting the send process.

| Program | Mote1 | Mote2 | Mote3 | Mote4 |
|---|---|---|---|---|
| Empty TinyOS2 | 9.189 mW | 9.219 mW | 9.222 mW | 9.146 mW |
| Green LED | 6.758 mW | 6.695 mW | 6.726 mW | 6.671 mW |
| Red LED | 7.525 mW | 7.586 mW | 7.580 mW | 7.565 mW |
| Yellow LED | 6.965 mW | 7.062 mW | 6.924 mW | 6.878 mW |
| Timer 1 ms | 3.422 mW | 3.288 mW | 3.391 mW | 3.264 mW |
| Timer 10 ms | 0.919 mW | 0.883 mW | 0.899 mW | 0.853 mW |
| Timer 50 ms | 0.255 mW | 0.248 mW | 0.249 mW | 0.239 mW |
| Timer 100 ms | 0.124 mW | 0.121 mW | 0.117 mW | 0.115 mW |
| Timer 500 ms | 0.056 mW | 0.056 mW | 0.054 mW | 0.054 mW |
| Timer 1000 ms | 0.042 mW | 0.044 mW | 0.039 mW | 0.042 mW |
| Red LED + 10 ms | 3.715 mW | 3.742 mW | 3.742 mW | 3.734 mW |
| Superpostion Error | 0.048 mW | 0.051 mW | 0.048 mW | 0.049 mW |

*Table 2. Different programs running on a Mica2 mote have been profiled using a measurement setup based upon a National Instruments data acquisition card NI PXI-6221 DAQ.*

Another aspect that is considered by Table 2 is the superposition of different power states. First of all it has to be kept in mind that superposition can only be applied if startup times or other timing issues do not interact with each other. Consider the example of two components sharing the same bus system where one cannot start and stop both components at the same time if they need to be under permanent control via the same bus system. For that reason we have chosen non-interacting components: timer and LEDs. The combined and averaged power consumption of toggling the red LED every time a 10 ms timer generates an event is compared to 50 % the power dissipation of the LED plus the timer. The last line in Table 2 shows that the error is tolerable, especially compared to unit variation among different motes it is quite small.

Another effect that has its results presented in Table 2 is scaling the timeline. Frequency analysis has been applied similar to what is shown in Figure 8 for accurate results. The average additional power dissipation of the timer decreases monotonically with increased time periods as expected, but it is nonlinearly depending on the period length. This effect occurs due to finite range of registers being used for counting ticks until the timer has to be fired in software. Visual inspection of traces for 500 ms and 1000 ms periods shows additional events in between the events that are signaled to the application.

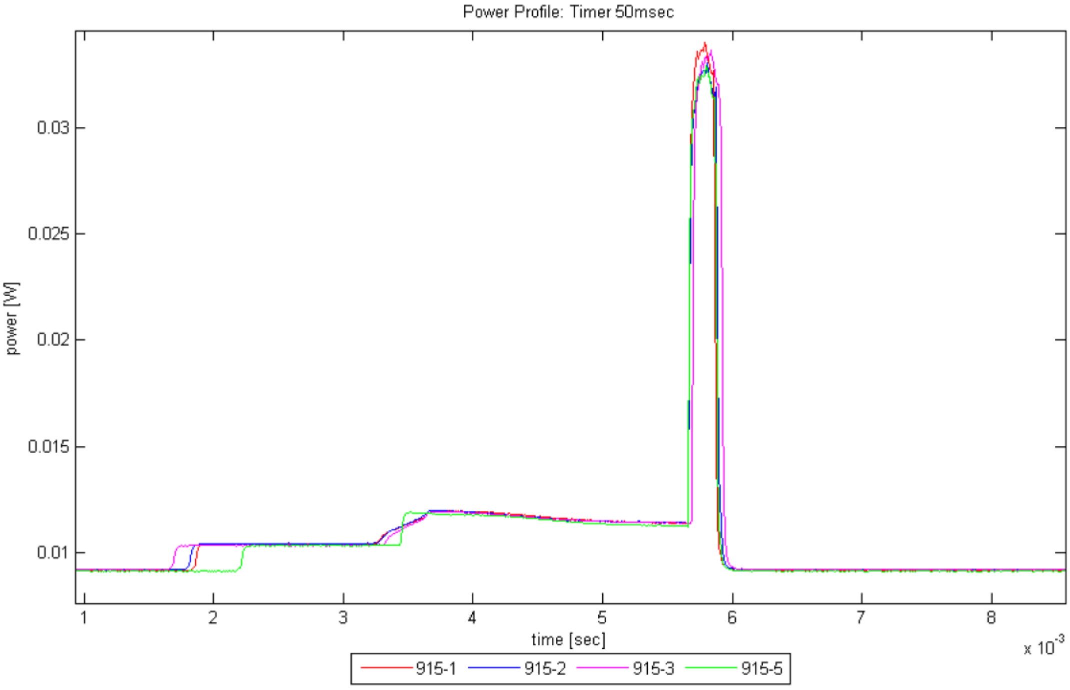



*Figure 9. Four different motes have been profiled running a timer fired periodically every 50 ms. All four motes' power profiles have been traced and are plotted above.*

Apart from concrete values that have been traced we have experienced that a lot of debugging capabilities lies within visual inspection of power profiles as has also been mentioned by Haratcherev, Halkes, Parker, Visser and Langendoen (2008). Figure 9 provides an example for the combination of register access, interrupt handling and forwarding through software layers which makes up the significance of a timer. Timers are needed for switching state machines, switching hardware and software components, generation of time stamps and for synchronization – they are virtually used everywhere. Fortunately, timers have a very characteristic profile as can be seen in the plot. Though the different motes result in different traces, they all share the same structure. Therefore they are good candidates for automated recognition, especially as timers are usually used before other hardware components are switched on which leads to transitioning into another power state. Furthermore, we have already seen that scheduler events, task switches, a running radio module and a transmitting radio module can be identified directly as well. Therefore, future directions might be to use this knowledge for including automated recognition of power state transitions from power profiles.

Another important factor for sensor grids is the transmission power. Throughout this article we have suggested using full transmission power for sending messages via the radio. We will now present results on power dissipation for sending differently sized messages at different transmission power. Table 3 gives an overview of profiles that have been traced for sending messages every 250 ms.

First, Table 3 shows that mote 2 seems to have its radio hardware altered somehow. Furthermore, we do not get significant results for differently sized messages. Using batteries for supplying the motes for these experiments may have influenced the setup's accuracy, but we conclude, that the additional power dissipation or longer messages using BMAC as of Tinyos 2.1 does not significantly impact power dissipation on average. So, especially if network coding or other message-size related optimizations are being applied, we conclude that the maximum message size that is supported by the network stack in use should be selected. Spending only a very small fraction of time and power dissipation for sending longer messages we can conserve energy and make networking more reliable by reducing the pressure on MAC protocols. Splitting up a given payload among a little more messages is far less efficient and reliable than sending a little longer payload.

| Program | Mote1 | Mote2 | Mote3 | Mote4 |
|---|---|---|---|---|
| 1 Byte | 46.719 mW | 44.914 mW | 46.209 mW | 45.681 mW |
| 2 Bytes | 46.606 mW | 44.651 mW | 45.985 mW | 45.704 mW |
| 5 Bytes | 46.565 mW | 45.038 mW | 45.949 mW | 45.488 mW |
| 10 Bytes | 46.822 mW | 44.924 mW | 45.976 mW | 45.554 mW |
| 25 Bytes | 46.729 mW | 44.487 mW | 46.195 mW | 46.206 mW |
| 10 Bytes -20 dBm | 45.385 mW | 43.990 mW | 44.792 mW | 44.250 mW |
| 10 Bytes -10 dBm | 45.639 mW | 44.140 mW | 44.794 mW | 44.320 mW |
| 10 Bytes 0 dBm | 45.517 mW | 44.113 mW | 44.935 mW | 44.974 mW |
| 10 Bytes 1 dBm | 45.751 mW | 44.239 mW | 45.180 mW | 44.751 mW |
| 10 Bytes 2 dBm | 45.357 mW | 44.182 mW | 45.212 mW | 44.815 mW |
| 10 Bytes 10 dBm | 47.445 mW | 44.510 mW | 46.611 mW | 46.409 mW |

*Table 3. Different programs running on a Mica2 mote have been profiled using a measurement setup based upon a National Instruments data acquisition card NI PXI-6221 DAQ.*

Furthermore, Table 3 shows that increasing radio sending power does not increase power dissipation too much for Mica2 motes. So it is a similar situation as with message length. Investing a little more power dissipation on average pays off for more robust transmissions and ease of networking. This is obvious when comparing Table 3 entries for increased transmission power with what can be conserved from saving a full message in Table 1.
As a last component the sounder will be characterized and then be profiled when being used as a component for ranging in a localization system based on acoustic ranging.

| Program | Mote1 | Mote2 | Mote3 | Mote4 |
|---|---|---|---|---|
| SB1 Null | 9.132 mW | 9.163 mW | 9.168 mW | 9.095 mW |
| SB2 Null | 9.131 mW | 9.159 mW | 9.163 mW | 9.093 mW |
| SB3 Null | 9.132 mW | 9.162 mW | 9.163 mW | 9.092 mW |
| Sounder Programs | SB1 | SB2 | SB3 | Mean |
| Snd 100-10 ms | 10.315 mW | 10.284 mW | 10.299 mW | 10.300 |
| Snd 100-50 ms | 14.396 mW | 14.246 mW | 14.353 mW | 14.332 mW |
| Snd 1000-10 ms | 9.303 mW | 9.305 mW | 9.305 mW | 9.304 mW |
| Snd 1000 50 ms | 9.703 mW | 9.716 mW | 9.728 mW | 9.716 mW |
| Snd 1000 100 ms | 10.157 mW | 10.177 mW | 10.147 mW | 10.160 mW |

*Table 3. Three different Mica2 sensor boards have been profiled. Sounder measurements are named with the period length first and then the sounder active period length. Both values are given in microseconds. Mote1 has been used for the sounder measurements not subtracting the null program here.*

Table 3 lists sensorboard measurements for using the sounder. Though the sounder is suitable for building time-difference-of-arrival (TDoA) measurement based indoor localization systems it consumes power in the order of power that is dissipated by the CPU alone. While the CPU is the last component that has not been considered in detail so far, we will only give a short example. Different low power modes are not explicitly considered by this work. Their characterization can be neglected compared to operational modes. The only thing that has to be considered here are the possible wake up events and hardware wake up time.

### On the CPU Load when Performing Calculations in a Localization System

For computing the position on a mote from an over determined system the least squares solution can be calculated making use of a technique called singular value decomposition (SVD). Figure 10 shows the power profile of a mote calculating SVD and sending a chirp signal from the sounder after sending a beacon for allowing to perform TDoA measurements at the receiver.

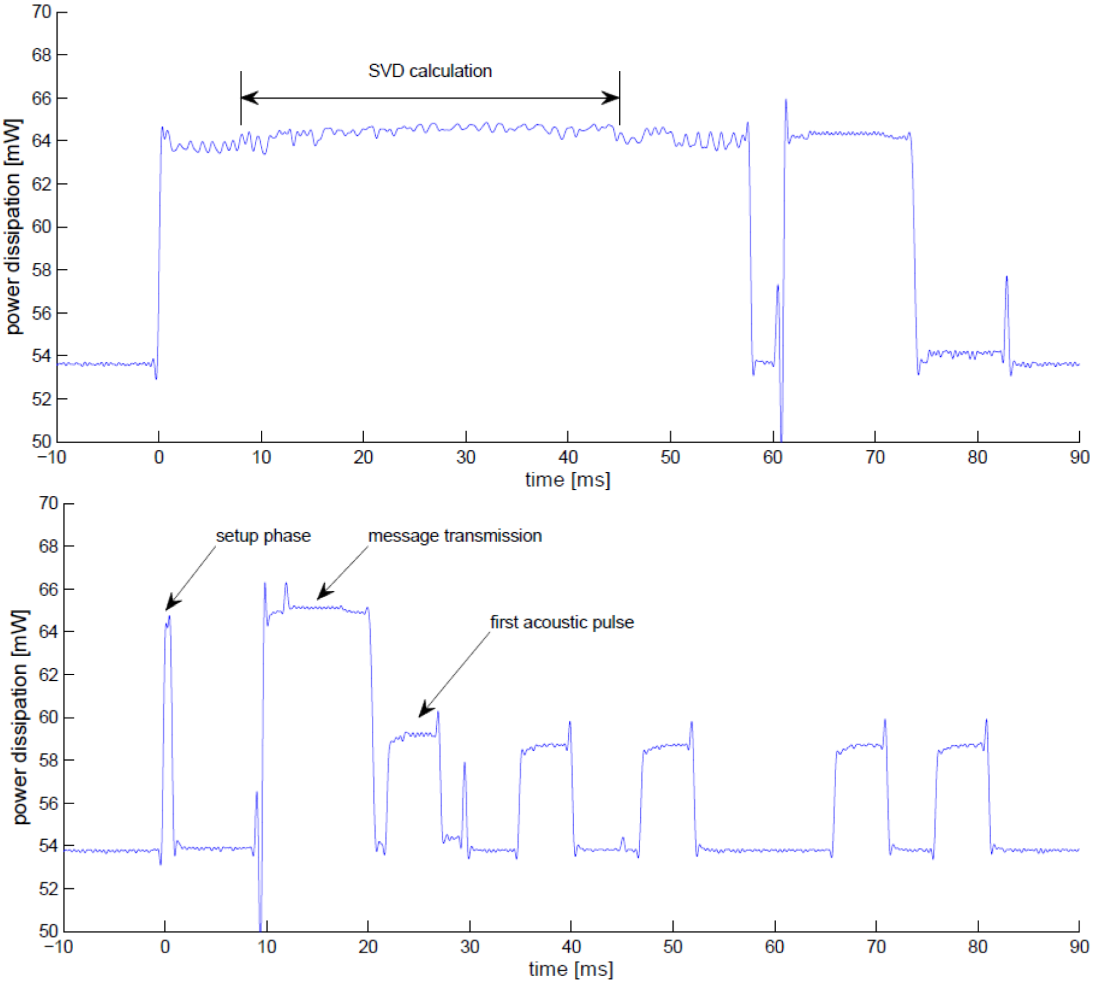
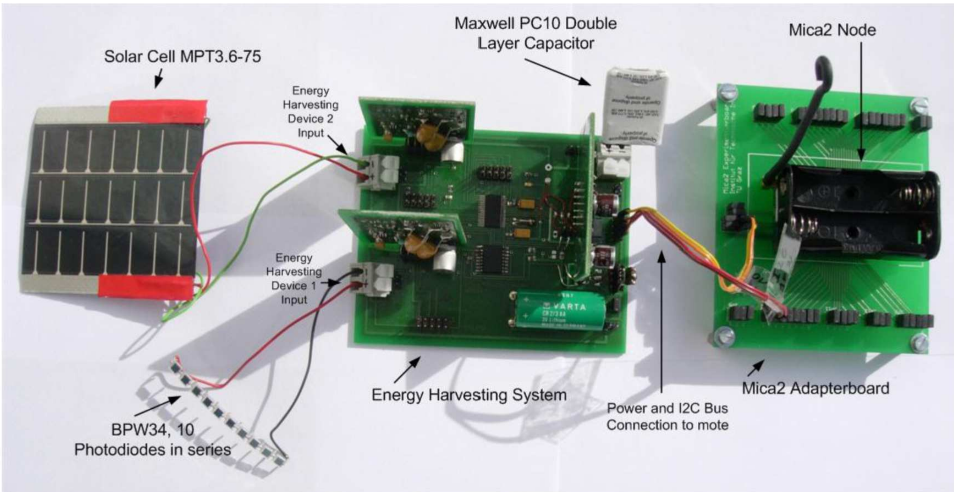



*Figure 10. Computational resources are scarce as well. Therefore the computationally demanding parts of applications like the SVD calculation when performing localization significantly contribute to the power dissipation profile. The upper plot shows the SVD calculation. The lower plot compares the impact of message transmission and acoustic pulse generation.*

## Simulation and Testbed Accuracy and Different EHSs and their Efficiencies

From a hardware technology point of view all results presented so far have been traced using Mica2 motes. This section will extend the view to EHSs as well. They usually have an energy measurement system on board or may even be such a system and have sensory hardware and a radio module as well. So, they may primarily be measurement devices and secondly be wireless sensors as well.



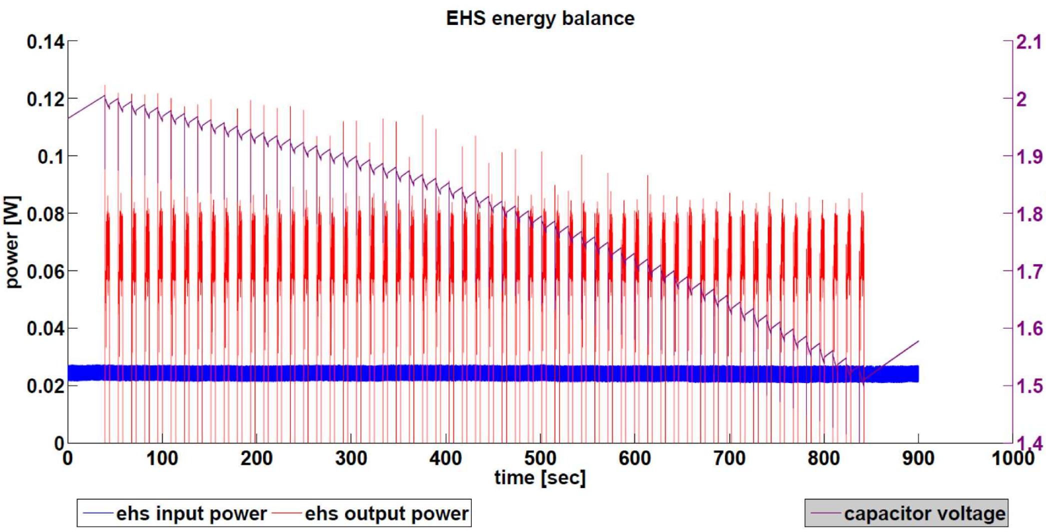




*Figure 11. The EHS presented by Glatz et al. 2008 and the measurement of the EHS efficiency model with a laboratory power supply attached with the measurement setup as presented by Glatz, Hörmann, Steger and Weiss 2010.*

The measurements may be based on counting energy packets or from integrating a shunt based power measurement over time. An energy packet counting approach has been implemented by the EHS by Glatz

et al. (2008) which can be seen in Figure 11 including results from an EHS efficiency model measurement setup as well.
Furthermore, an EHS using shunt-based measurement approaches is depicted in Figure 12 including a plot for its dynamic range and accuracy of the shunt-based measurement approach. The second approach called RiverMote – due to the fact that it is tailored towards in-river water level measurements – is built around the same microprocessor as the TelosB architecture.

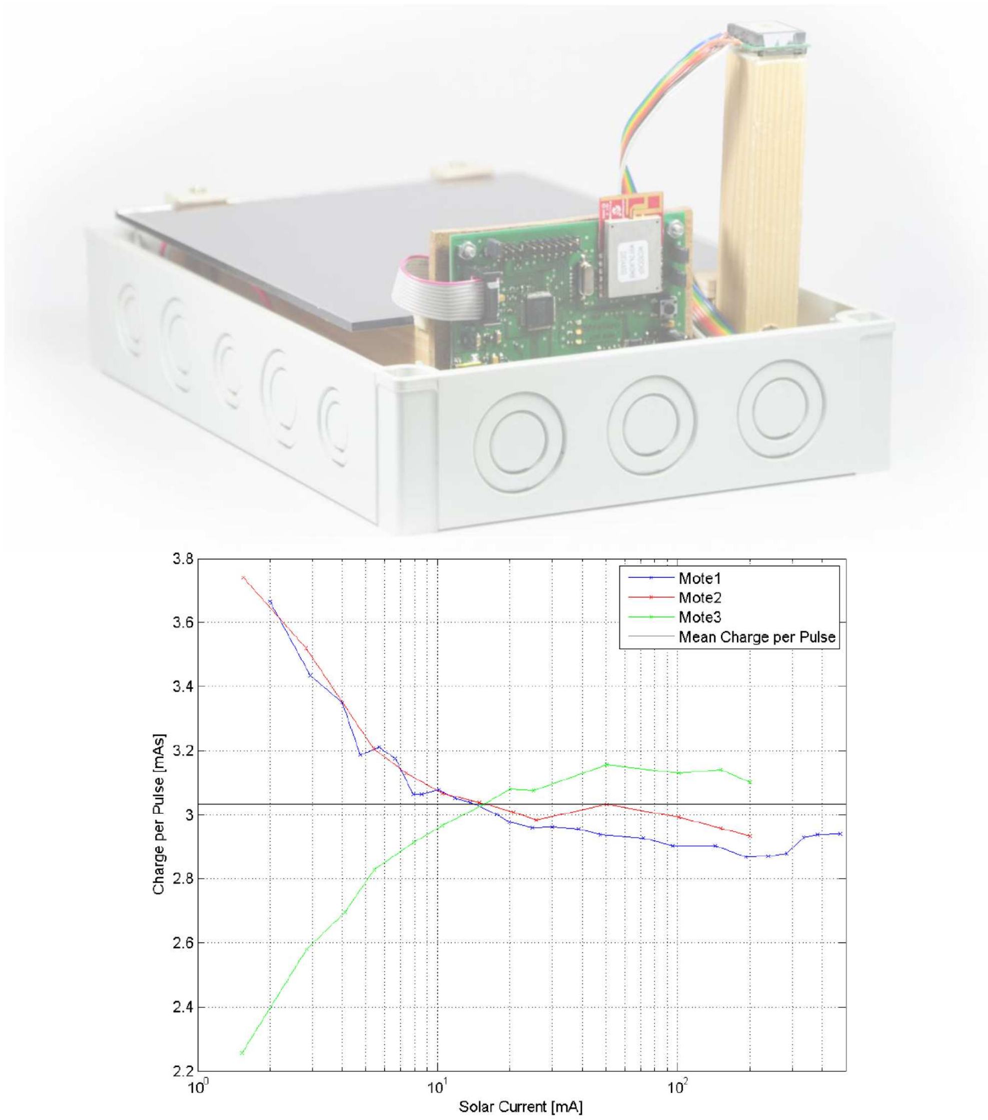



*Figure 12. RiverMote is shown without the top of its water-tight housing. The lower plot shows test measurements that have been profiled from different RiverMote's input measurement circuitry. The accuracy varies among different motes, but it stays below the accuracy of typical simulation-based approaches for regions that significantly contribute to the overall motes' energy consumption.*

Though the measurement results using a cheap approach – in terms of money and power – are less accurate than what can be measured with the other EHS, it is much more efficient. Still, the accuracy is in the range of what can be simulated with state of the art power profiling tools in terms of power dissipation averaged over one activation period. Furthermore, the results from the efficiency measurements from Figure 11 show that output efficiency alone drops to half the value at the lower threshold (approximately 60% of what can be stored in the DLC) compared the upper threshold with a full DLC. Both systems are utilizing photovoltaic cells as energy harvesting devices, and DLCs as energy storage. Additionally both systems are of approximately the same price if both the EHS and the mote are being considered. So, we can compare both approaches easily. The RiverMote design results in an overall input to output efficiency that will always be above 60 %. This is the final example of showing that the tradeoff between using low-power or low-cost hardware, simulation and testbed setups may lead to significantly different results. In this hardware related example it suffices to have quite low accuracy due to the fact that the information is available at runtime and other components are efficient enough.

## FUTURE RESEARCH DIRECTIONS

Ongoing work at the department dealing with the EHSs and the measurement setup by Glatz et al. is concerned with further automating the measurement setup and working towards providing a validation suite for power profiling simulation. A first step into that direction is being taken by the presentation of the TOSPIE2 (Tiny Operating System Plug-In for Energy Estimation) approach by Glatz, Steger and Weiss (2010). The Eclipse plug-in operates a power state model database, allows for automated hardware and software measurement runs and evaluation and can annotate the information gathered back into the system. Different research groups around the world are currently dealing with that issue of interconnecting and therefore validating measurement and simulation based approaches.

## CONCLUSION

This chapter has presented the different steps that have to be taken when designing and running wireless sensor networks. Energy harvesting systems and mote hardware have been profiled accurately in terms of power dissipation and energy conservation characteristics.

The often neglected problem of translating a model from simulation-based approaches to hardware measurements has been discussed and a methodology has been outlined for how to deal with that issue given the task of implementing a network stack optimization. Here, network coding serves as an example for outlining of what measures are appropriate for dealing with the issue.

Other hardware-implemented methods and applications have been profiled as well with their pitfalls explained for keeping the approach from being academic. A novel multi-application middleware and an acoustic range-based localization system have served as an example.

This way the reader has been guided through state of the art approaches for designing WSN hardware, dealing with channel access, networking in sensor grids and their middleware aspects as well as application development issues.

## REFERENCES

Ahlswede R., Cai N., Li S.-Y. & Yeung R. (2000). Network information Flow. *IEEE Transactions on Information Theory, 46*(4), 1204-1216.

Akyildiz I.F., Su W., Sankarasubramaniam Y. & Cayici E. (2002). Wireless Sensor Networks: A Survey. *Elsevier Computer Networks 38*(4), 343-422.

Beutel J., Kasten O., Mattern F., Römer K., Siegmund F. & Thiele L. (2004). Prototyping Wireless Sensor Network Applications with BTNodes. First European Workshop on Wireless Sensor Networks: EWSN'04. Berlin, Germany, 2004, 01, 19.

Glatz P.M. (2010). *A Scalable Middleware with Energy Management for Environmentally Powered Sensor Networks.* Unpublished doctoral dissertation. Graz University of Technology, Austria.

Glatz P.M., Hein K.B. & Weiss R. (2009). Energy Conservation with Network Coding for Wireless Sensor Networks with Multiple Crossed Information Flows. *The 10th International Symposium on Pervasive Systems, Algorithms and Networks: I-SPAN'09.* Kaohsiung, Taiwan. 2009, 12, 14.

Glatz P.M., Hoermann L.B., Steger C. & Weiss R. (2010). A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency. *The Sixth IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing: PerSeNS'10.* Mannheim, Germany. 2010, 3, 29.

Glatz P.M., Loinig J., Steger C. & Weiss R. (2009). A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks. *The 9th IEEE Malaysia International Conference on Communications 2009: MICC'09.* Kuala Lumpur, Malaysia, 2009, 12, 14.

Glatz P.M., Meyer P, Janek A., Trathnigg T. Steger C. & Weiss R. (2008). A Measurement Platform for Energy Harvesting and Software Characterization in WSNs. *IFIP/IEEE Wireless Days Conference: WD'2008.* Dubai, UAE. 2009, 11, 24.

Glatz P.M, Steger C. & Weiss R. (2010). Poster Abstract: TOSPIE2: Tiny Operating System Plug-In for Energy Estimation. *The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks 2010: IPSN'10.* Stockholm, Sweden. 2010, 04, 12.

Glatz P.M. & Weiss R. (2009). LINDONCS: Localized In-Network Detection Of Network Coding Structures in Wireless Sensor Networks. *The 4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks: PM2HW2N'09.* Teneriffa, Canary Islands, Spain. 2009, 10, 29.

Haratcherev I, Halkes G., Parker T., Visser O. & Langendoen K. (2008). *PowerBench: A Scalable Testbed Infrastructure for Benchmarking Power Consumption*. International Workshop on Sensor Network Engineering: IWSNE'08. Santorini Island, Greece, 2008, 06, 11.

Hill J. & Culler D. (2001). A Wireless Embedded Sensor Architecture for System Level Optmization. Technical Report, U. C. Berkeley, 2001.

Kansal A., Potter D. & Srivastava M (2004). Performance Aware Tasking for Environmentally Powered Sensor Networks. *SIGMETRICS Performance Evaluation Review, 32*(1), 223-234.

Kansal A., Hsu J., Zahedi S. & Srivastava M. (2007). Power Management in Energy Harvesting Sensor Networks. *ACM Transactions on Embedded Computing Systems, 6*(4), 32.

Landsiedel O., Wehrle K. & Gotz S. (2005). *Accurate Prediction of Power Consumption in Sensor Networks*. The 2nd IEEE Workshop on Embedded Networked Sensors: EmNets'05. Sydney, Australia, 2005, 05, 30.

Levis P., Lee N., Welsh M. & Culler D. (2003). *TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications*. 1st International Conference on Embedded Networked Sensor Systems: SenSys'03. Los Angeles, CA, 2003, 11, 5.

Lim H.B., Teo Y.M., Mukherjee P., Lam V.T., Wong W.F. & See S. (2005). Sensor Grid: Integration of Wireless Sensor Networks and the Grid. *Annual IEEE Conference on Local Computer Networks: LCN'05.* Sydney, Australia. 2005, 11, 15

Lin K., Yu J., Hsu J., Zahedi S., Lee D., Friedmann J., Kansal A., Raghunathan V. & Srivastava M. (2005). Poster Abstract: Heliomote: Enabling Long – Lived Sensor Networks Through Solar Energy

Harvesting. Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems: SenSys'05. San Diego, CA. 2005, 11, 2.
Mainwaring A., Culler D., Polastre J., Szewczyk R. & Anderson J (2002). Wireless Sensor Networks for Habitat Monitoring. The 1st International Workshop on Wireless Sensor Networks and Applications: WSNA'05. Atlanta, Georgia, USA, 2002, 09, 28.
Moon S., Kim T. & Cha H. (2007). Enabling Low Power Listening on IEEE 802.15.4-based Sensor nodes. The IEEE Wireless Communications and Networking Conference: WCNC'10. Hong Kong, 2007, 03, 11.
Perla E., Catháin A.Ó., Carbajo R.S., Huggard M. & Goldrick C. (2008). *PowerTOSSIM z: Realistic Energy Modelling for Wireless Sensor Network Environments*. The 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks: PM2HW2N'08. Vancouver, Canada 2008, 10, 27.
Rincón F.J., Susu A.E., Sánchez-Élez M., Atienza D & Micheli G. (2007). *A Simulation Model for Wireless Sensor Networks Based on TOSSIM.* 22nd Conference on Design of Circuits and Integrated Systems: DCIS'07. Barcelona, Spain, 2007, 11, 21.
Tham C. & Buyya R. (2005). SensorGrid: Integrating Sensor Networks and Grid Computing. *CSI Communications*, Special Issue on Grid Computing. Computer Society of India, July 2005
Thonhauser M., Kreiner C. & Leitner A (2010). *A Model-Based Architecture supporting Virtual Organizations in Pervasive Systems*. Submitted to 15th IEEE International Conference on Engineering of Complex Computer Systems: ICECCS'10. Oxford, United Kingdom, 2010, 03, 22.
Weiser M. (1991). The computer for the 21st century. *Scientific American. 265*(3), 94-101.
Yick J., Mukherjee B. & Ghosal B. (2008). Wireless Sensor Networks Survey. *Elsevier Computer Networks 52*(12), 2292-2330.

## KEY TERMS & DEFINITIONS

Keyword: Wireless Sensor Networks, Sensor Grids, Modeling, Simulation, Measurement, Power State Model, Energy Harvesting System Efficiency

**Towards Modeling Support for Low-Power and Harvesting Wireless Sensors for Realistic Simulation of Intelligent Energy-Aware Middleware**

Philipp M. Glatz, Leander B. Hörmann, Christian Steger, Reinhold Weiss
Institute for Technical Informatics, Graz University of Technology, Austria
{Philipp.Glatz, Leander.Hoermann, Steger, RWeiss}@TUGraz.at
Contact author 2nd email: Glatz.Philipp@Gmail.com

Index terms: battery modeling, battery performance, energy harvesting, low-power, measurement, middleware, modeling, simulation, networking, wireless sensor networks

Abstract: Low-power embedded system architectures with integrated standardized wireless transceivers and versatile CPUs enable previously impossible wireless and mobile devices. The quality of mobile entertainment platforms, sensing system, automation and control constantly increase with the advent of new architectures. While the field of high performance and low-power embedded architecture has matured and offers many trade-offs today, the field of energy harvesting wireless sensor networks has evolved from power aware ones.

Today, a number of optimizations, modeling approaches and tools exist for various wireless embedded devices. However, there are still issues where there is a strong need for deepening the understanding of power aware and energy harvesting technology. First, there is a lack of battery performance aware low-power modeling approaches. Second, there is only little support available for cost-efficient dimensioning of energy harvesting architectures. Reason for still having open issues and very limited support for design space exploration in these fields is the complexity of the system and the time it takes to validate models and assumptions in real world experiments.

This chapter provides an overview of existing devices and solutions for wireless embedded systems with a special emphasis on low-power solutions and energy harvesting opportunities. Modeling and simulation environments including wireless communication simulation will be briefly surveyed. A lack of integrated low-power chips, radios and energy harvesting with suitable tool-support will be identified. Discussion will then be narrowed down to state-of-the-art solutions and environments in the area of wireless sensor networks. As a case study, an energy harvesting and battery plug-in for a power profiling simulation and emulation environment will be developed. It will then be shown how the environment can be used for profiling the optimization of an energy-storage-aware middleware for wireless sensor networks. Power profiling and energy simulation results accuracy will be determined by comparison with accurate energy harvesting and mote hardware measurements.

## I. ENERGY CONSTRAINTS FOR MOBILE AND WIRELESS EMBEDDED SYSTEMS

Since the advent of pervasive computing [1] and the vision of smart dust [2], several different scientific fields and technical solutions for various application domains have positioned themselves in our everyday's life.

Wireless sensor networks (WSNs) as surveyed in [3] and [4] and mobile ad-hoc networks (MANETs) as of [2] and [5] have found their ways into scientific, technical and consumer applications. Especially, findings from experiments in related scientific fields pave the way for innovations when the acquired knowledge is used to implement novel optimizations in end-user applications. A striking example for how knowledge from scientific fields may foster innovation is given by a speech of Jeffrey Sharkey at the Google I/O Developer Conference

2009. The major topics of his speech - *Coding for Life* - can best be depicted with citing parts of its introductory outline:
*"The three most important considerations for mobile applications are, in order: battery life, battery life, and battery life."*
This reflects the generic nature of energy conservation issues. Unlike any other it can critically impact mobile and distributed systems in general. The speech also shows how energy conservation schemes that have first been researched for WSNs and MANETs are implemented for end-user applications after some time. Some of the issues like (i) using low-power modes, (ii) estimating energy consumption from processing and transmission power, and (iii) conserving energy with opportunistic behavior or with (iv) using event-driven paradigms seem pretty obvious. However, for the application domain under consideration the developers paradigm had not been shifted to incorporating these issues up to that time.
A similar situation can be found for the battery performance awareness of WSNs. Although there exist simulation environments that claim to be battery aware, and despite the fact that a number of battery performance models are around that are far better than linear ones, there is a lack of evidence that power profile from simulation have been compared to results from actual hardware measurements of battery performance. The situation for modeling energy harvesting and dimensioning cost-efficient hardware is similar. [WAN] surveys different WSN simulation environments. However, there is no implementation available that incorporates voltage dependencies of power profiles, rate discharge and relaxation effects or temperature dependencies [27]. The modeling errors that come from neglecting these effects and neglecting the impact of energy harvesting efficiency models complicates matters when measuring energy conservation optimization methods [34].

*A. Outline of the Chapter*

The authors of this chapter argue that mapping applicable means of optimization from one field or domain to another one (e.g. from WSN/MANET research to smartphone implementations) and developments within single fields of research are mainly inhibited if there is something missing at the basis of theoretical frameworks or tools that are supporting their implementation and simulation. For this reason, the chapter will start with outlining the state-of-the-art of low-power, energy-efficiency and harvesting support in hardware, its measurement and its simulation techniques, and tool-chain availability of major players in the field. Summing up related work we will identify a lack of battery modeling support in simulation as well as a lack of tools for harvesting hardware dimensioning in system modeling environments. Especially, seamless integration of simulation and profiling tools at different levels of abstraction is missing. There is no such tool that lets the system architect choose energy harvesting devices (EHDs) with suitable characteristics and e.g. explore different low drop out (LDO) regulator analogue designs at the same design step when designing an energy harvesting system (EHS) at system level for supplying a WSN mote. However, the efficiency of differently designed systems may vary drastically for different kinds of systems running or simulating different kind of application settings even if as basic as setting a duty cycle (DC) as evaluated in [6]. As WSNs and MANETs have many similar characteristics, but WSNs tend to be a little more resource constrained, we choose WSN platforms as examples for showcasing novel tools and techniques. These will include battery model evaluation and its integration for WSN simulation as well as EHS and EHD characterization with energy efficiency modeling. Results of these energy storage focused considerations will then be applied for simulating the appliance of novel optimization techniques for networking middleware design. WSNs provide good examples for constrained devices - especially when it comes to low-power, energy-efficiency, battery performance, and energy harvesting. Section III will discuss results of power profiling simulation and hardware measurements. Furthermore, the accuracy of these results will be discussed. This provides the basis for the two subsequent sections on the implementation and evaluation of battery models for WSNs and embedded systems in general. Finally, the chapter will be concluded and an outline of promising future directions will be discussed.

## II. INDUSTRY TOOL CHAINS FOR ENERGY HARVESTING AND RELATED WORK

This section is split in two parts. First, tool chains and hardware platforms of some of the major players in the field of low-power and energy harvesting solutions are briefly surveyed. For these, the completeness of available approaches will be summed up. Missing links will be identified. Second, related work will be introduced that provides some background and highlights related work for missing parts in the approaches presented before. They will serve as a basis for the energy-aware considerations of battery and harvesting modeling and simulation in this chapter

### *A. Industry Platforms for Modeling, Simulation and Implementation*

The scope of this short outline on industry energy energy-aware tool chains and hardware is limited to technologies that are relevant for WSNs.

*1) Texas Instruments:* A state-of-the-art WSN platform - namely the TelosB [7] - is implemented using a product from the MSP430 family of CPUs. Instruction-level simulation and emulation engines are available. Furthermore, for WSNs, simulation tools exist that let one integrate embedded system (ES) platforms like the TelosB in simulations that span different levels of abstractions. TOSSIM [8] is a simulation environment that allows simulation with incorporating channel effects as well. For this simulation environment, there exist extensions like PowerTOSSIM that enable power profiling features. Power profiling is available for different versions of TOSSIM where one example is shown [9]. Other options exist as well. If disregarding channel modeling, Avrora [10] and its power profiling extension in AEON [11] even enable power profiling at the instruction-level.

*2) Atmel:* For the ARM-based Mica2 [12] platform there are several well-integrated and hybrid simulation approaches available, that consider different levels of abstraction. A well-known example is given by ATEMU [13] - the ATmel EMUlator.

For both types of platforms - TI and ARM - and their ES accordingly, although power simulation is a well understood aspect of these platforms, there is a lack of integrated expressive battery models. Extension points for possible integration of battery models exist as exemplified by the PAWiS environment [14], but a proper implementation is missing.

There are even existing approaches that try to combine several different techniques. COOJA [15] integrates and interconnects Avrora with TOSSIM and NS-2 to cover instruction-level aspects as well as operating system considerations and network-level issues.

*3) Energy Harvesting Modeling and Simulation:* With EHS-enhanced WSNs, an even more energy-constrained field of scenarios is being targeted. For both, ARM and TI platforms, examples exist that integrate their most-well-known motes in an EHS solution. While Heliomote and Prometheus are solely or mainly based on rechargeable batteries, other EHS solutions with ARM and TI platforms exist that are only supplied by energy storage structures made of double layer capacitors (DLCs). The EHS in [16] supplies a Mica2 mote and the RiverMote [17] platform with integrated EHS is implemented using the same processor as the TelosB platform.

Compared to power profiling simulation alone, things get a lot more complicated when it comes to designing systems with energy harvesting in mind. One gets a feeling for that when looking at the fact, that quite close in time after when TI and Jennic - that joined NXP - made available their EHS design kits like the ez430-rf2500 since 2008, one to two years later, integrated low-power chips and low-power radio embedded platforms appear in 2009 and 2010. The TI CC430 and ATmega128RFA1 are very-well designed for meeting high-performance and low-power for EHS requirements.

Only few tools exist for EHS simulation. Furthermore, the scientific community lacks an integrated approach with the combination of system level simulation and networking.

Approaches as shown in [18] are limited to single hardware instance considerations. Most environments for network level consideration of harvesting systems take a very abstract point of view with stochastic modeling as in [19]. The few integrated solutions as in [20] and Castalia in [21] lack versatile tools for accurate hardware measurements for validating simulation results. Another approach, presented as Tospie2 in [22], integrates accurate profiling of power state models (PSMs) and EHS efficiency models (EEMs), but the architecture is a very loosely coupled system.

*B. Battery Models and Applications to Wireless Sensor Networks*

When modeling and simulating low-power ES architectures power consumption, one needs a profound basis that different power aware optimizations can be compared to. For an often neglected issue for achieving accurate WSN power dissipation profiles - namely battery effects modeling - we outline related work for battery characterization setups, battery effects modeling and networking optimizations that are applied upon that.

Different types of energy storage devices exist where each type implies special characteristics that have to be considered when tuning a system. However, as existing simulation environments are even lacking support for non-rechargeable batteries, we will limit our concept to these types. Rechargeable batteries and their effects will not be considered, but for extending the scope to battery-free systems, systems with DLCs will be considered.

*1) Automated Battery Characterization Setups:* The battery performance at a given point in time depends on several factors that are influencing the system, but it is also a system with memory. The different possibilities of temperature histroy, the discharge rate of the load, the current state-of-charge (SOC) and possible combinations of what these values have been in the past result in a large number of different contexts that the battery may be in. Therefore, profiling performance characteristics - as shown for Alkaline Manganese types of Duracell primary cells in [23] - may take some time and needs accurate measurements. In general, characterization may be based on SOC measurements that are chemical methods, current integration methods or voltage measurement methods. However, for finding a profiling methodology that can be integrated on the WSN mote later on, chemical methods and current integration are not practical.

For profiling the SOC, voltage-based measurements are promoted in [24]. They also discuss applicability to state-of-health (SOH) measurements. SOH are especially useful for extending ideas to rechargeable systems. Setups for profiling different types of batteries (charge and discharge performance) are presented by [25] and [26]. In [25] they provide simple equivalent circuit diagrams (ECDs) of batteries using internal resistance and two low-passes with different time constants for modeling a battery.

*2) Battery Effects Modeling:* ECD modeling of battery effects is quite common in the fields of electrical and computer engineering. Distinguishing between theoretical capacity and actual capacity as of [27], the actual battery performance in terms of the amount of charge that is extracted mainly depends on discharge rates, temperature conditions and ageing effects. Better actual performance means more extracted charge before the battery cut-off voltage is reached. For primary cells, recovery effects can be utilized for increasing the performance.

Different simple empirical models are given by the Thevenin Battery model and linear electrical model with their ECDs given in [28], Peukert's power law explained in [29], and the model of Pedram and Wu [30]. All these models have in common that batteries are modeled as nonlinear systems with memory. Despite the existence of model evaluation experiments with an error of one order of magnitude or below, there are a number of side effects that need to be considered for WSNs that cannot completely be coped with by using these standard models and variants.

First of all, a drain down to the cut-off voltage of batteries can often not be achieved because of a limited voltage range of components, needed converters' capabilities and efficiencies. Depending on the type of simulation integration, motes have to be removed from simulation when the lower threshold has first been hit, or they have to be shut off temporarily if proper functionality can be assumed after a relaxation period.

Furthermore, the evaluation of SOC dependent effects that can be used to in simulation environments cannot directly be mapped to real-world deployments. While the SOC models for simulations may utilize a lot of computational power, several difficulties arise at runtime where resources may be very limited. The accuracy of voltage-based SOC determination methods depends on measurement capabilities and ADC accuracy. In addition to that, online SOC determination and appliance of battery models demands for exact measurement of the load's characteristics and memorizing it as well as possibly evaluating computationally complex models. [17] depicts such measurement's accuracy for an EHS-enhanced WSN platform to be below 10% in relevant areas. Adding such a bias to battery model input may drastically limit model performance. Finally, timing measurements and memorizing their results may heavily impact PSMs of applications using sophisticated platforms.

Similar as for approximation of battery performance alone, EHS energy storage structures can also usually not be modeled with curve fitting techniques alone. Therefore, EEMs and leakage are used for dynamic and static energy budget calculation. This chapter will mainly focus on battery modeling considerations and static EHS effects.

*3) Battery Technology and Networking Optimizations:* Knowledge of energy storage characteristics allow applying power management techniques and optimizations. Especially transmission scheduling in [31] and [32] and traffic shaping techniques [33] apply. Dealing with energy storage performance modeling issues necessitates complex measurement, analysis and algorithmic efforts. This fact – and the time it takes to prove novel energy storage technologies working – makes battery modeling and optimizations to lag behind current technology capabilities and application requirements. Therefore, despite novel technologies may outperform older technologies, the considerations in this chapter's modeling section are based upon technologies as they can be compared based on examples of [MAX] and [EPCOS] for ultracapacitors and [Neil], [Tartagni] and [Chu] for batteries and energy harvesting aspects. More recent developments include graphene-based ultracapacitors [Stoller] and Lithium Ion Batteries with carbon nanofibers [Fan].

## III. POWER PROFILING ACCURACY AND CONCEPTUAL CONSIDERATIONS

The conceptual part of this chapter starts with experimenting with related power profiling approaches. Expressiveness and possible exactness are quantified. Different aspects of power profiling, that can still be optimized deserve consideration.

### *A. Expressiveness of Wireless Sensor Network Power Profiling Based on Power State Models*

Mica2 motes, the EHS in [16], RiverMote and the measurement setup from [34] are used for dimensioning, modeling and measuring systems.

Figure 1 can be used for comparing simulation-based power profiles with results from hardware measurements. Although a modified Avrora in Tospie2 [22] allows performing decent profiling of power traces and the main components' power state switches, there are some issues that need to be considered when applying simulation based approaches.

*1) Issues with Simulation Environments:* First, there are several aspects that are hard to be modeled exactly when it comes to using the radio. This is mainly due to different output power and complex signal propagation issues and MAC interaction among motes. Second, it is a non-trivial task to deduce correction values from these comparisons for the simulation-based approach or vice-versa. This is what a developer using Tospie2 can try to do with the PSM database. It holds the power values of all different power states and allows setting a time span for transient effects.

*2) Simulation Evaluation Results:* Energy, power and timing results that are shown in Figure 1 are summarized in Table I. Radio activation and its power dissipation needs careful considerations on nearly all WSN platforms. Targeting low-power or energy neutral operation

(ENO)-capable WSNs, we evaluate LPL, because we expect long-term deployments' applications to be asleep most of the time unlike communication or multimedia networks such that LPL plays an important role in their PSMs. The main focus is less on the sending process but rather on the listening cost. This is also, because sending power can be varied on most platforms while listening power dissipation is pretty much fixed most of the time. Summing up, LPL is a significantly important and good PSM part to compare quality of simulation-based, measurement-based, analytical and hybrid power profiling models with annotations.
The test application wakes up every 4 s for 2 s where LPL is used to check for incoming transmissions. During the first two activation periods the noise floor needs to be estimated. Therefore, the radio is constantly listening. Exactly profiled state switches from simulation are shown in Figure 1(c) and their hardware measurement counterpart in Figure 1(g) accordingly. An erroneous behavior of TinyOS2.1 is corrected. The latest community implementation keeps the SPI bus active while shutting down the radio for LPL. This keeps the CPU from being put to sleep.
**Problems with Hardware Models:** This is the first big problem of using simulation-based approaches alone: wrong machine or hardware models. Although both tests run exactly the same (SPI-corrected LPL) executable binary, it is a matter of fact that Avrora cannot correctly profile power dissipation, because SPI activity is out of its scope. While this over-estimation of power dissipation can be corrected by subtracting approximately 10mW power dissipation on average, it makes the power profiling process more error prone. A possible solution can be to use an approach as it is supported by Tospie2: do design space exploration with simulation-based approaches and validate important results with accurate measurements.
**Limited Resolution of Simulated Time:** The second issue is related to timing. While hardware measurements need to take into account Shannon's Theorem, simulation environments have issues with sampling rates as well. A slightly extended Avrora that is used in Tospie2 takes 1 h time on an otherwise idling dual-core processor to simulate 10 s of the LPL test at 100kS to achieve the same sampling rate as with the hardware measurement setup. To get to a more acceptable performance simulation profiles in this chapter use 20kS and comparing Figures 1(e) and 1(f) shows how accurate the simulation can get. However, the achievable accuracy is not accurate enough to accurately profile low-power networking optimization measures.
**Profiling LPL Radio Activation:** Figure 1(b) shows the process of LPL activation. This includes two timer interrupts and radio activation with the states 'power down', 'crystal activation', 'crystal bias activation', 'crystal bias synchronization' and 'receive', whereas 'receive' turns out to be an RSSI measurement. Table I tells us that simulation results are way off from being accurate enough to trace a timer event's power dissipation. The time span between the two timer interrupts cannot easily be compared either due to the error in the machine model (time span between t = 14.53 s and t = 14.55 s in Figure 1(b)). Therefore, the discussion concentrates on the second timer interrupt and radio activation periods that are shown in Figures 1(e) and 1(f) for simulation and measurement. Their evaluation in Table I (2nd peak) shows that both timing and power values vary a lot and result in the simulation having an error of more than 22% under-estimation.
**The Average Power Dissipation of LPL:** We argue, that in a setting where already LPL alone with no incoming messages makes up more than 70% of the overall energy needs, errors of more than 22% cannot be acceptable. It is of no use - from a power profiling point of view - to use such an approach for profiling power-optimization capabilities when developing new protocols. Albeit the simulation environment is an invaluable tool for characterization, measurements, analytical measures or hybrid approaches are needed as well. Further calculations of how much energy is needed will be based on the average power dissipation during a full LPL period if no message is received which has been measured to be approximately $P_{LPL,period} = 468\ \mu W$.

**Average Transceiver Power Dissipation:** The energy budget that is needed for the transmission of a single message depends on the preamble length, the time needed for transmission of the message with the payload itself and the transmission power. For actually deciding whether the message could be received by another party, their antenna models, channel models and MAC considerations have to be taken into account as well. Therefore, we do not evaluate a power profiling simulation environment's accuracy here, because for virtually any application one might find a setting that leads to an accurate value of overall energy needs, but only with simulation settings tuned towards a specific application that might completely fail for another one. The authors want to point out the fact that it is out of scope of this chapter and remind the reader that not only there is a significant error of simulated power dissipation to be expected, but also that it is pretty uncertain of what the error will be. Several environmental conditions and especially water, human beings and crop may render an accurate offline error analysis useless when it comes to real deployments. The authors believe, that it is worth to set up design rules and best-practice guidelines for deployments (especially mote placement, temperature dependencies and activation and reset policies). Then, still, one might be better off to use maximum transmission power and simple slotted networking design with worst-case assumptions on power dissipation for offline analysis instead of power-optimization for coverage efficiency and uncertainty and synchronization-based MAC protocols for possibly a little less energy needs. Characteristics of low-power embedded systems for long-term operation impose the need for error-aware and deterministic-driven approaches for their design process. So, these less deterministic approaches - despite their possible savings - are not considered here.

Worst case analysis is based on results from measurement-based power dissipation profiling. Although, there is a significant mote-to-mote variation, it is valid to assume the power dissipation to be approximately 65mW when the mote is sending. The actual average power dissipation during the transmission process varies, also if the transmission power is kept constant. For a preamble length as in the LPL experiments above, it is further assumed that a message transmission takes 260ms until the power state can be left again. Therefore, on average, one transmission will need 16.900mJ. Motes that are listening to incoming messages have been measured to consume the 2.2309th part of the sending process on average which is approximately 7.575mJ then. Depending on the time offset of when the sender starts sending the preamble and when the receiver samples the RSSI, the energy needed at the receiver may vary from 0.119mJ over the expected value of 7.575mJ up to 15.032 when receiving a message. The authors therefore suggest to rather introducing a random variable and evaluate networks for upper and lower bounds of how much energy may be needed. This may give a feeling for how much can be gained from tuning the networking mechanism and especially motes' synchronization.

**Choosing a Suitable Preamble Length:** Depending on how dense a network has been deployed, different LPL preamble lengths may be optimal when optimizing with subject to overall network energy conservation. This assumes that an energy-aware or energy management protocol is applied where motes balance their energy or their workload accordingly.

We compare two cases and show trade-offs for a 9-node network with full adjacency matrix in Figure 2: 8 motes are uploading their data to a root or cluster head in Figure 2(a) and 1 mote is downloading information into the network (e.g. for network control) in Figure 2(b). Plots assume the energy budget of RiverMote's DLCs and further assume that load-balancing or energy management is available, such that the overall network's energy budget can be modeled instead of single motes' energy. Modeling single motes' energy budgets can be done the same way, but it only provides little more insights while it would complicate the discussion at hand. The modeling results show that network connectivity or density heavily impact the energy model. Figure 2(c) shows different possible network settings for worst case

MAC synchronization and the maximum simulation error profiled in Figure 1 in greater detail. We show that choosing a LPL preamble length - which may be constrained by scalability, real-time or channel capacity constraints - impacts the overall network energy needs if short LPL periods have to be chosen. The network consumes over 50% more energy when using small preambles. E.g. 10100 J instead of 2500 J, 13310 J instead of 23940 J and 28570 J instead of 42620 J for the underestimated, expected and overestimated energy needs for 5ms and 255ms LPL periods in Figure 2(b). The maximum sustainable data rate increases linearly with the available energy. 0−50% of the LPL periods are used for transmitting data. In case 8 out of 9 motes are transmitting (Figure 2(a)) the maximum sustainable data rate for an usable energy of 1686 J is bounded between 2% (overestimated) and 5% (underestimated).

**Energy Needs of CPU, Sensors and Data Logging:** Figure 3 shows the process of sampling the photo sensor on a Mica2 Mote. It is shown that the time span for reading the sensor with TinyOS2 takes 9ms, while the actual conversion of the value at the ADC only needs 260 µs in the end. The ADC is given the long time span to let its value settle down. The maximum value of power dissipation that is to be expected is approximately 48mW at initialization time when the CPU is active. It has to be taken into account that the time measurements include the software with hardware abstraction and access times as well.

Figure 4 shows power profiles taken with the hardware measurement setup similar to these taken for the radio before in Figure 1. What is left out is the plot of the second peak of activation that actually operates the sensor. It is shown separately in Figure 3. Finally, Figure 5 shows corrected measurements. Summing up the results in Table II it turns out that sampling a sensor is of similar cost (43 µJ) like sensing the channel (83 µJ) for LPL. The authors want to point out that it is pretty hard to model sensor readings' power consumption that way, while it might actually be even harder to give accurate estimates with simulation environments. It is presented here, because current literature lacks detailed comparison of exemplary sensors. Other platforms and other sensors might behave in other ways, but the example presented here perfectly shows what it takes to accurately model and quantitatively discuss and evaluate WSN optimization measures. Other exact sensor and actuator profiles for the same platform can be found in [35] where they are put in the context of being optimized subject to implementing a low-cost and accurate ranging solution.

Other sensors exist that need significantly more time and power than what is needed for sampling sensors directly connected to the ADC with no special control circuitry. E.g. in the datasheet of the accelerometer of the MTS300 sensor board one can see that the setup time is 16.5ms. TinyOS sets itself asleep for 17ms while the module comes up. Next, there exist modules where it depends on QoS demands to decide how long and which way a component should be used to find an optimal power dissipation to QoS trade-off. [35] gives an overview of power awareness aspects and how they can be optimized when implementing localization with the MTS300 sounder and microphone.

*C. The Impact of Variable Battery Voltage - From Power State Models to Resistance Models*

Unfortunately, PSMs alone are not sufficiently expressive for describing WSN motes' energy budgets. While errors from measurements, simulation and modeling have to be coped with when dealing with PSMs, the concept of PSMs has to be rethought when it comes to different supply voltages [6]. Figure 6 shows experiments taken every 30 minutes. A battery pack with 2 Duracell PLUS batteries has constantly been drained. Figure 6 plots average power, current and voltage values that are normalized to one over the first experiments' values. Within 6 hours, the average power dissipation drops for more than 8%. This trend continues down to voltages where system components cannot be operated any more. Different components of the Mica2 mote - processor, radio, LEDs, sensor - have been profiled for their power dissipation and functionality at different voltage levels. Converting circuits allow operating these components down to battery voltage below their specified voltage range. Figure 7 shows results for profiling the same application as in Figure 6. This time, a laboratory power supply is used for supplying the mote. Although, no complete tests on component functionality could be performed, radio communication was still working in a small laboratory setup. Finally, it can be concluded that the error in measurement, simulation and modeling can easily be exceeded by the error of PSMs if their variation at different supply voltages is not taken into account.

*D. A Battery Modeling Concept*

For implementing battery-awareness in a power profiling extension of Avrora in Tospie2, extended standard ECD models have been implemented in Matlab with SimScape and a large number of hardware measurements have been performed for training these models. Figure 8 shows the overall performance of a Duracell PLUS battery pack. A Mica2 mote is used as load. It is running a modified RadioCount2Leds application with LPL and a 50% application DC. Figure 9 outlines core parts of the main modeling concept. Subcomponents are depicted in Figure 10. The idea is to set up SimScape models of different complexities. These implement battery and load models that can be compared with measurements as shown in Figure 8. The load is to be represented as impedance value and the battery is represented by partly pre-charged capacitors. Their capacitance values have to be learned with gradient decent methods together with related resistor values to achieve correct time constants. Upper parallel capacitors are pre-charged while lower parallel capacitors contain no charge at all when the simulation starts similar to the capacitors that are in series with the main capacitance and mote. The mote implements model parameters that can be profiled from hardware measurements. Parallel first order capacitors are mainly responsible for long-term rate discharge and relaxation effects. Second order capacitors allow modeling voltage overshoots. Series capacitors can conveniently be used to model short term deviation with different time constants.

All states of the mote are automatically captured by a profiling system. These include exact measurements of the timing of processor states, LED states and radio activation. An example is given by Figure 11. The ability to track down which component is responsible for a load change is essential for setting up correct impedance models. This will also reveal why PSMs cannot be as expressive and correct as impedance models. The power dissipation trace that is shown in Figure 11 is representative for the test applications being used here. The modified TinyOS2 implementation is used with a 50% DC. LPL is activated all the time. Half the time the LEDs are on and a message is being sent.

*1) Impedance Models for Energy Budget Design:* Figure 12 reveals the direct dependency of impedance values of processor and LEDs power states from the supply voltage. The impedance of the power-intensive sending process stays the same over 25 days.

*E. Harvesting Modeling Concept*

EHSs are best described by their efficiencies, their black-out sustainability (BOS) [36] and voltage, current and energy thresholds that have to be met. A detailed description of the basics of the RiverMote platform that have to be considered for BOS-aware system design can be found in [36].

BOS characteristics depend on DLC leakage and the EHS components' efficiencies. Especially, the performance of the energy storage structure balancing mechanism that prevents the system from being physically damaged has to be taken into account. Figure 13 shows results of the balancing current evaluation. The balancing current limits the maximum charging current of the system. For a charging process that is starting with empty DLCs, the average charging current should not exceed 200mA. However, this value is the limit for a complete charging process while assuming a pretty high voltage difference of 0.2 V of the DLCs. This way, the maximum current that may come from the EHDs can be modeled.

Although DLC leakage and the balancing circuitry have carefully been evaluated, results cannot directly be mapped for completely describing RiverMote's overall leakage and BOS. Therefore, the system has been evaluated for its overall leakage and BOS after a charging process with solar cells attached. RiverMote 2 has been charged to 4.40 V by a solar cell and RiverMote 3 has been charged to 3.55 V. For assuming harsh conditions, the system leakage measurements are started immediately after the charging process with no stabilization of the DLCs. Figure 14 shows results of the overall EHS performance including DLC leakage, EHS leakage and circuit efficiency. The EHS performs well with several days of BOS in both cases. At a shutdown voltage the EHS completely shuts off the drain that comes from the EHS itself.

While Figure 14 shows the result of the two tests alone, Figure 15 depicts the complete BOS characteristic curve of the platform. It is still based on the measurements with harsh conditions with starting the blackout experiment immediately after the charging process. Results of the experiments have been plotted as a reference as well. It can be seen that the BOS can be up to 3 weeks for a fully charged system. In case the system might know about an approaching black-out period, the load could be reduced for stabilization of the DLCs and even longer BOS. The local information on how long RiverMote can sustain a black-out can locally be computed with locally evaluating the characteristic curve generating function. This relaxes the amount of energy that is needed for determining the BOS online and possibly sending an error message with a negligible small amount of the overall energy budget.

## IV. IMPLEMENTATION

For being able of modeling partly pre-charged capacitors in the ECD, we have replaced the parallel capacitors in our model with capacitors plus a voltage source for making the model applicable to using a Laplace transform.

Figure 16 outlines one implementation option that can be feasible for approximating the battery performance traces that have been profiled in experiments. For arriving at analytical expressions of the model in different domains, as has been indicated in Figure 9, we need to set up current equations at each of the nodes and voltage equations for each mesh. Equations (1) to (11) show how results can be obtained.

### *A. Analytical Derivation of Model Equations*

Kirchhoff's current law for Node 1:

$$I_{CC}(t) - I_{CAP}(t) - I_{CPL1}(t) - I_{CPU1}(t) = 0 \tag{1}$$

Kirchhoff's voltage law for mesh I:

$$V_{CPU1}(0) - V_{CAP}(0) + V_{CAP}(t) - V_{RPU1}(t) - V_{CPU1}(t) = 0 \tag{2}$$

Kirchhoff's voltage law for mesh II:

$$V_{CAP}(0) - V_{CPL1}(0) + V_{CPL1}(t) + V_{RPL1}(t) - V_{CAP}(t) = 0 \tag{3}$$

These equations can be directly transformed with the help of Laplace Transformation and the number of unknowns can be reduced by using the Ohmic Law.

Kirchhoff's current law for Node 1:

$$I_{CC}(s) - I_{CAP}(s) - I_{CPL1}(s) - I_{CPU1}(s) = 0 \tag{4}$$

Kirchhoff's voltage law for mesh I:

$$\frac{V_{CPU1}(0)}{s} - \frac{V_{CAP}(0)}{s} + V_{CAP}(s) - I_{CPU1}(s) \cdot R_{PU1} - V_{CPU1}(s) = 0 \quad (5)$$

Kirchhoff's voltage law for mesh II:

$$\frac{V_{CAP}(0)}{s} - \frac{V_{CPL1}(0)}{s} + V_{CPL1}(s) + I_{CPL1}(s) \cdot R_{PL1} - V_{CAP}(s) = 0 \quad (6)$$

Now one can use the Laplace transformed current-voltage relation for the capacitor to simplify the mesh equations.
Kirchhoff's voltage law for mesh I:

$$\frac{V_{CPU1}(0)}{s} - \frac{V_{CAP}(0)}{s} + \frac{I_{CAP}(s)}{C_{CAP} \cdot s} - I_{CPU1}(s)\left(R_{PU1} + \frac{1}{C_{PU1} \cdot s}\right) = 0 \quad (7)$$

Kirchhoff's voltage law for mesh II:

$$\frac{V_{CAP}(0)}{s} - \frac{V_{CPL1}(0)}{s} + I_{CPL1}(s)\left(R_{PL1} + \frac{1}{C_{PL1} \cdot s}\right) - \frac{I_{CAP}(s)}{C_{CAP} \cdot s} = 0 \quad (8)$$

With the help of Equations 4, 7, 8 and the Ohmic law it's possible to derive a solution for $V_{CC}$ which only depends on $R_{Sim}$.

$$V_{CC}(s) = f(R_{Sim}(s), s) \quad (9)$$

The resulting solution for our Laplace model:

$$\begin{aligned} V_{CC}(s) \;=\; & \Bigg(R_{Sim}(s)\Bigg(\frac{C_{PL1}V_{CAP}(0)}{C_{CAP}s(C_{PL1}R_{PL1}s+1)\left(\frac{C_{PL1}}{C_{CAP}(C_{PL1}R_{PL1}s+1)} + \frac{C_{PU1}}{C_{CAP}(C_{PU1}R_{PU1}s+1)} + 1\right)} + \\ & + \frac{C_{PU1}V_{CAP}(0)}{C_{CAP}s(C_{PU1}R_{PU1}s+1)\left(\frac{C_{PL1}}{C_{CAP}(C_{PL1}R_{PL1}s+1)} + \frac{C_{PU1}}{C_{CAP}(C_{PU1}R_{PU1}s+1)} + 1\right)} - \\ & - \frac{C_{PL1}V_{CPL1}(0)}{C_{CAP}s(C_{PL1}R_{PL1}s+1)\left(\frac{C_{PL1}}{C_{CAP}(C_{PL1}R_{PL1}s+1)} + \frac{C_{PU1}}{C_{CAP}(C_{PU1}R_{PU1}s+1)} + 1\right)} - \\ & - \frac{C_{PU1}V_{CPU1}(0)}{C_{CAP}s(C_{PU1}R_{PU1}s+1)\left(\frac{C_{PL1}}{C_{CAP}(C_{PL1}R_{PL1}s+1)} + \frac{C_{PU1}}{C_{CAP}(C_{PU1}R_{PU1}s+1)} + 1\right)} - \frac{V_{CAP}(0)}{s}\Bigg)\Bigg) / \\ & / \left(-\frac{1}{C_{CAP}s\left(\frac{C_{PL1}}{C_{CAP}(C_{PL1}R_{PL1}s+1)} + \frac{C_{PU1}}{C_{CAP}(C_{PU1}R_{PU1}s+1)} + 1\right)} - R_{Sim}(s)\right) \end{aligned} \quad (10)$$

With the help if the inverse Laplace transform $V_{CC}$ can be solved.

$$V_{CC}(t) = \mathcal{L}^{-1}(V_{CC}(s)) \quad (11)$$

For completely solving the model, $V_{CPU1}$ and $V_{CPL1}$ have to be calculated the same way. For solving systems of differential equations in analytical form, one can use Mathematica, which also allows providing code output. Bash scripts for post-processing of this output leads to C-code that is automatically included by the environment that builds the license-free executable for running the model.

Although, it is a cumbersome task to set up all equations and solve them analytically with Mathematica and then implementing root-solving procedures in C-code, it finally pays off if an arbitrary number of instances can be created license-free to perform gradient descent on the model parameters for optimizing their fit to measured data traces.

## V. APPLICATION OF BATTERY-AWARE SIMULATION TO LOAD BALANCING

First of all, training results and ECD settings of the battery model are shown that could best capture the measurement results' characteristics. Battery models that are trained for fitting battery performance with Mica2 loads at given DC periods and average discharge rates are then used for simulating programs that show the same basic DC and discharge characteristics.

### *A. Training a Battery Model with Gradient Descent*

The best model error that could be achieved with a model mapped to all stages of the design flow is shown in Figure 17 including its model parameter settings. The gradient decent method tries optimizing the RMSE of the quite peaky simulation curve (blue curve) that contains the application level DC effects as well compared to the averaged measurement curve (green curve).

### *B. Evaluation of a Wireless Sensor Network Program*

WSN goals usually include maximizing the lifetime by evenly draining the motes energy reservoirs. A standard way of doing this is to evenly balance the load, that comes from the burden to fulfill tasks for achieving a given end-user performance, among the motes in the whole network or in clusters. With battery effects in mind, the question has to be answered if load balancing schemes still work as expected with evenly draining energy reservoirs for maximizing the network lifetime.

An example for the TinyOS implementation of a workload balancing WSN middleware is the agent-based multi-application middleware MAMA that is presented in [37]. Figures 18 and 19 show how MAMA is dealing with several applications at the same time by using the concept of virtual organizations (VOs) and where the agents are implemented in the TinyOS protocol stack. It turns out that load balancing and state-of-the-art power profiling simulation alone is not sufficient for achieving the most efficient solution to the question of how the agents shall be grouped and how they shall migrate among motes. A sample evaluation of the on-going project for energy-storage-aware simulation by implementing Avrora extensions and interfaces to the modeling tools described above is performed with a simple load-balancing example where battery effects take place and impact power aware optimizations.

The application scenario assumes a linear-topology (for ease of discussion) WSN that has load balancing applied. Possible application scenarios include applications where a small cluster of motes is responsible for monitoring a given area with a given spatial sampling rate per cluster and sending back its results to a sink. Figure 20 shows the burden at each node from radio activation (sending commands and sending back results) with green bars and sensor and processor activation with black bars. Two different types of load balancing are applied to simplified nesC programs for sample evaluation with the modified Avrora including a battery model.

First (prog 1), each mote splits its work per 100min time slot into 4 smaller packets. 50% energy drain from the radio and 50% energy drain from an energy-intensive sensor or algorithm are assumed which is reasonable for using a GPS sensor as it comes with RiverMote or the Cyclops camera [38] that can be used with WSNs as well. The second type of load balancing (prog 2) has the same load applied on average. The difference in battery performance, that might occur, can come from different time spans of loads and times that are given for battery relaxation. We apply the same balancing among the motes, but this time, the load is split into single packets per 100min time slot instead of 4.

Results in Figure 20 show that the strategy with a single activation per 100min time slot achieves better results than splitting it up in a larger number of work load packets. Intuitively interpreting the results, one can say that the load is short enough such that it applies the same close to linear decrease every time the mote is active, but it pays off if more time is given for relaxation effects.

VI. CONCLUSION AND FUTURE WORK

This chapter has been written with the intention to shed light on modeling aspect as well as energy budget simulation and power profiling hardware measurements of low-power and energy harvesting wireless embedded systems. It aims at surveying existing technology and modeling methodologies with providing novel insights – especially to issues related to battery and harvesting modeling.

Several case studies in the field of WSNs have been outlined. The scope of the chapter spans different aspects from different levels of abstraction for power profiling and networking with related aspects. As a main result, BOS modeling of a state-of-the-art EHS-WSN platform is shown and a common instruction-level power profiling simulation environment is extended with a novel battery effect modeling and simulation environment. A special focus is put on the expressiveness, possible level of detail and accuracy of LPL and networking optimization at the middleware level.

Future work will include further automation of the measurement setup that is being used for profiling measurements of energy storage characteristics. In addition to that, a state-of-the-art high performance and low-power multi-core DSP-FPGA board is planned to have its PSM described. The PSM shall then be transformed to an impedance model as has been shown in this chapter for WSNs. This shall allow implementing an adaptive battery modeling environment in-situ on board of a platform gaining benefit from it.

REFERENCES

[1] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *Computer*, vol. 36, no. 3, pp. 25 – 31, Mar. 2003.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "smart dust"," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 271–278. [Online]. Available: http://doi.acm.org/10.1145/313451.313558

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[5] S. Giordano, *Mobile Ad Hoc Networks, in Handbook of Wireless Networks and Mobile Computing*. John Wiley & Sons, Inc., New York, USA, 2002, ch. 15.

[6] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Energy efficient supply of WSN nodes using Component-Aware dynamic voltage scaling," in *European Wireless 2011 (EW2011)*, pages 147 – 154, Vienna, Austria, 2011.

[7] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN*, 2005.

[8] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003, pp. 126 – 137.

[9] 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 *PM2HW2N '08: Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*. New York, NY, USA: ACM, 2008, pp. 35–42.

[10] B. Titzer, D. Lee, and J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 477 – 482.

[11] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," in *EmNets '05: Proceedings of the 2nd IEEE workshop on Embedded Networked Sensors*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 37–44.

[12] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," in *Technical report, Computer Science Department, University of California at Berkeley*, 2001.

[13] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J. Baras, “Atemu: a fine-grained sensor network simulator,” in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, 2004, pp. 145 – 152.
[14] 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 *in Proceedings of 6th IFAC International Conference on Fieldbus Systems and their Applications*, 2005, pp. 129–134.
[15] F. Osterlind, 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*, 2006, pp. 641 –648.
[16] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, “A measurement platform for energy harvesting and software characterization in wsns,” in *IFIP/IEEE Wireless Days*, nov. 2008, pp. 1 – 5.
[17] P. M. Glatz, L. B. Hörmann, and R. Weiss, “Designing perpetual energy harvesting systems explained with rivermote: A wireless sensor network platform for river monitoring,” *Electronic Journal of Structural Engineering, Special Issue: Wireless Sensor Networks and Practical Applications (2010)*, pp. 55 – 65, 2010.
[18] L. Mateu and F. Moll, “System-level simulation of a self-powered sensor with piezoelectric energy harvesting,” *Sensor Technologies and Applications, International Conference on*, vol. 0, pp. 399–404, 2007.
[19] A. Seyedi and B. Sikdar, “Modeling and analysis of energy harvesting nodes in wireless sensor networks,” in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, 2008, pp. 67 –71.
[20] G. Merrett, N. White, N. Harris, and B. Al-Hashimi, “Energy-aware simulation for wireless sensor networks,” in *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON ’09. 6th Annual IEEE Communications Society Conference on*, 2009, pp. 1 –8.
[21] P. De Mil, B. Jooris, L. Tytgat, R. Catteeuw, I. Moerman, P. Demeester, and A. Kamerman, “Design and implementation of a generic energy-harvesting framework applied to the evaluation of a large-scale electronic shelf-labeling wireless sensor network,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, pp. 7:1–7:14, February 2010. [Online]. Available: http://dx.doi.org/10.1155/2010/343690
[22] P. M. Glatz, C. Steger, and R. Weiss, “Tospie2: tiny operating system plug-in for energy estimation,” in *IPSN ’10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. New York, NY, USA: ACM, 2010, pp. 410–411.
[23] “Duracell - entire mno2 technical bulletin collection.” [Online]. Available: http://www1.duracell.com/oem/Pdf/others/ATB-full.pdf
[24] V. Pop, H. Bergveld, P. Notten, and P. Regtien, “State-of-the-art of battery state-of-charge determination,” *Measurement Science and Technology*, vol. 16, no. 12, pp. R93–R110, 2005. [Online]. Available: http://doc.utwente.nl/62192/
[25] B. Schweighofer, K. Raab, and G. Brasseur, “Modeling of high power automotive batteries by the use of an automated test system,”*Instrumentation and Measurement, IEEE Transactions on*, vol. 52, no. 4, pp. 1087 – 1091, 2003.
[26] S. Abu-Sharkh and D. Doerffel, “Rapid test and non-linear model characterisation of solid-state lithium-ion batteries,” *Journal of Power Sources*, vol. 130, no. 1-2, pp. 266 – 274, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/B6TH1-4BK2FX3-1/2/4938d1b3771bb71f4989b7010f4d160a
[27] R. Rao, S. Vrudhula, and D. Rakhmatov, “Battery modeling for energy aware system design,” *Computer*, vol. 36, no. 12, pp. 77 – 87, 2003.
[28] Z. Salameh, M. Casacca, and W. Lynch, “A mathematical model for lead-acid batteries,” *Energy Conversion, IEEE Transactions on*, vol. 7, no. 1, pp. 93 –98, Mar. 1992.
[29] D. Linden, *Handbook of Batteries, 3rd ed.* McGraw-Hill, 2002.
[30] M. Pedram and Q. Wu, “Design considerations for battery-powered electronics,” in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, ser. DAC ’99. New York, NY, USA: ACM, 1999, pp. 861–866. [Online]. Available: http://doi.acm.org/10.1145/309847.310089
[31] P. Nuggehalli, V. Srinivasan, and R. Rao, “Energy efficient transmission scheduling for delay constrained wireless networks,” *Wireless Communications, IEEE Transactions on*, vol. 5, no. 3, pp. 531 – 539, mar 2006.
[32] C. Ma and Y. Yang, “Battery-aware routing for streaming data transmissions in wireless sensor networks,” *Mob. Netw. Appl.*, vol. 11, pp. 757–767, October 2006. [Online]. Available: http://dx.doi.org/10.1007/s11036-006-7800-2
[33] C.-F. Chiasserini and R. Rao, “Improving battery performance by using traffic shaping techniques,” *Selected Areas in Communications, IEEE Journal on*, vol. 19, no. 7, pp. 1385 –1394, Jul. 2001.
[34] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, “A system for accurate characterization of wireless sensor networks with power states and energy harvesting system efficiency,” in *IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing*, mar. 2010, pp. 468 – 473.
[35] P. M. Glatz, C. Steger, and R. Weiss, “Design, simulation and measurement of an accurate wireless sensor network localization system,” in *The 5th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks (PM2HW2N 2010)*, Bodrum, Turkey, 10 2010.

[36] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Designing sustainable wireless sensor networks with efficient energy harvesting systems," in *IEEE WCNC 2011 - Service and Application (IEEE WCNC 2011 - Service and Application)*, Cancun, Mexico, 3 2011.
[37] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "MAMA: Multi-Application MiddlewAre for efficient wireless sensor networks," in *2011 18th International Conference on Telecommunications (ICT 2011)*, pp. 1 – 8, Ayia Napa, Cyprus, 5 2011.
[38] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks," in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05. New York, NY, USA: ACM, 2005, pp. 192–204. [Online]. Available: http://doi.acm.org/10.1145/1098918.1098939
[WAN] Wan Du, David Navarro, Fabien Mieyeville, and Frédéric Gaffiot. 2010. Towards a taxonomy of simulation tools for wireless sensor networks. In Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools '10). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, , Article 52 , 7 pages. DOI=10.4108/ICST.SIMUTOOLS2010.8659 http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8659
[MAX] YEG Components, "Hc power series ultracapacitors datasheet," Maxwell Technologies
[EPCOS] EPCOS, "Ultracap single cell 5 F / 2.3V B49100A1503Q000 data sheet," [Online] http://www.epcos.com/inf/20/35/ds/B49100A1503Q000.pdf.
[Neil] N. White, and S. Beeby, "Energy Harvesting for Autonomous Systmes," Artech House, 292 pages, ISBN 9781596937185, June 2010.
[Tartagni] M. Belleville, E. Cantatore, H. Fanet, P. Fiorini, P. Nicole, M. Pelgrom, C. Piguet, R. Hahn, C. V. Hoof, R. Vullers and M. Tartagni, "Energy autonomous systems: future trends in devices, technology, and systems" 2009 - CATRENE - Cluster for Application and Technology Researchin Europe on Nanoelectronics, ISBN 9788890439902, 84 pages.
[Chu] B. Chu, "Selecting the Right Battery System For Cost-SensitivePortable Applications While Maintaining Excellent Quality," Microchip Technology Inc., 12 pages.
[Stoller] M. D. Stoller, S. Park, Y. Zhu, J. An, and R. S. Ruoff, "Graphene-Based Ultracapacitors," Nano Letters 2008 8 (10), 3498-3502.
[Fan] Z.-J. Fan, J. Yan, T. Wei, G.-Q. Ning, L.-J. Zhi, J.-C. Liu, D.-X. Cao, G.-L. Wang, and F. Wei, "Nanographene-Constructed Carbon Nanofibers Grown on Graphene Sheets by Chemical Vapor Deposition: High-Performance Anode Materials for Lithium Ion Batteries," ACS Nano 2011 5 (4), 2787-2794.

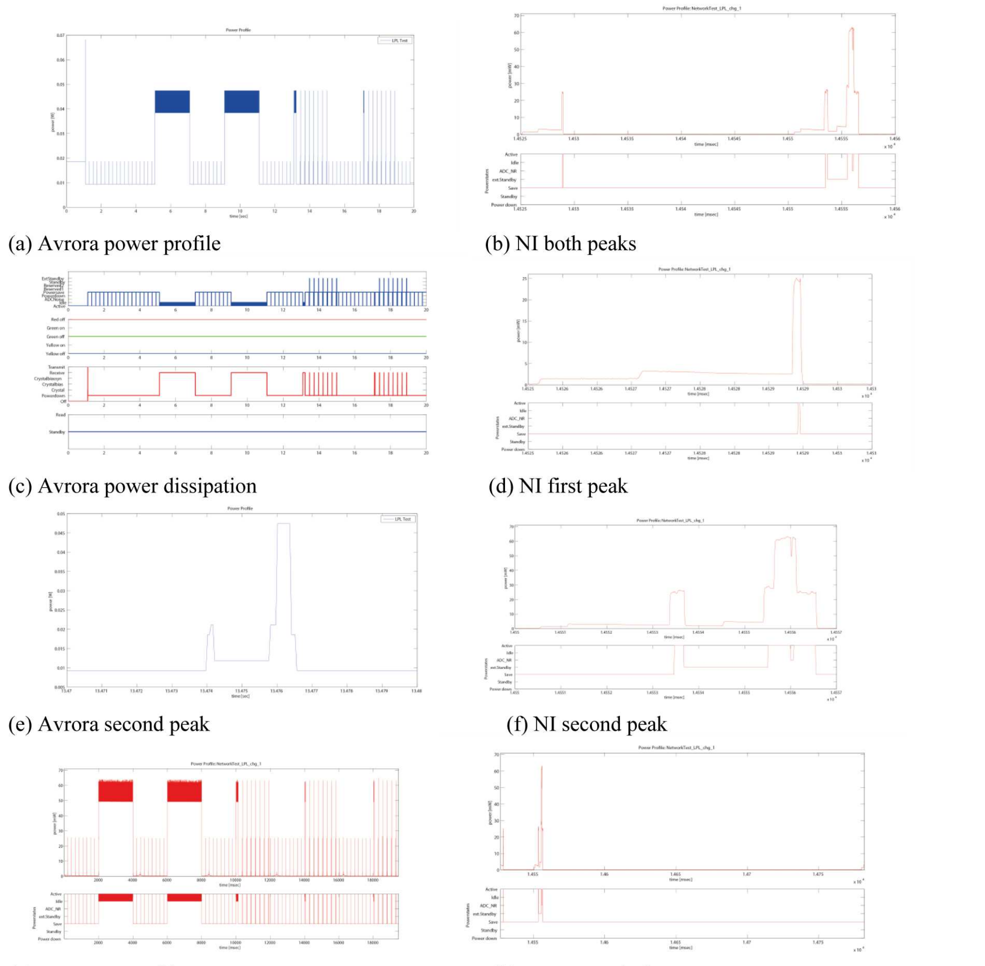
TABLES AND FIGURES

| Type of trace | Avg. power $[mW]$ | Time $[ms]$ | Energy $[\mu J]$ |
|---|---|---|---|
| NI 1st peak | 2.3322 | 5.0100 | 11.680 |
| NI 2nd peak | 9.6604 | 7.0100 | 67.720 |
| NI both peaks | 2.6162 | 32.0100 | 83.746 |
| NI full period | 0.4678 | 255.0100 | 119.286 |
| Sim 1st peak | 12.4000 | 0.1497 | 1.900 |
| Sim 2nd peak | 17.9000 | 2.9448 | 52.600 |
| Sim both peaks | 10.2000 | 28.2006 | 288.100 |
| Sim full period | 9.4000 | 252.1088 | 2372.400 |

TABLE I
Key results of comparing power profiles from measurement-based approaches with those that have been gathered from simulations using a standard PSM. LPL is used with a sleep interval of a quarter second.

| Type of trace | Avg. power $[mW]$ | Time $[ms]$ | Energy $[\mu J]$ |
|---|---|---|---|
| NI 1st peak | 4.0189 | 3.9500 | 15.875 |
| NI 2nd peak | 4.0471 | 13.5200 | 54.717 |
| NI both peaks | 1.4617 | 115.0500 | 168.164 |
| NI full period | 1.0727 | 1012.7500 | 1086.331 |
| NI 1st peak | 2.0930 | 4.3900 | 9.1881 |
| NI 2nd peak | 1.7959 | 26.0555 | 14.5100 |
| NI both peaks | 0.3701 | 116.0400 | 42.9520 |
| NI full period | 0.1362 | 1013.7100 | 138.0976 |

TABLE II
Key results of power profiles from a sensor that is directly connected to an ADC. The sensor is sampled every second.



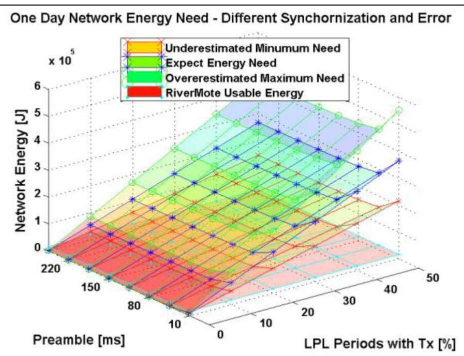
(a) Avrora power profile (b) NI both peaks

(c) Avrora power dissipation (d) NI first peak

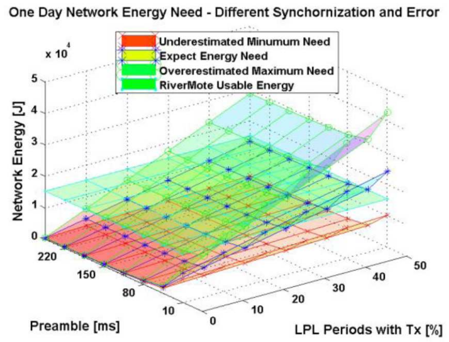
(e) Avrora second peak (f) NI second peak

(g) NI power profile (h) NI LPL period

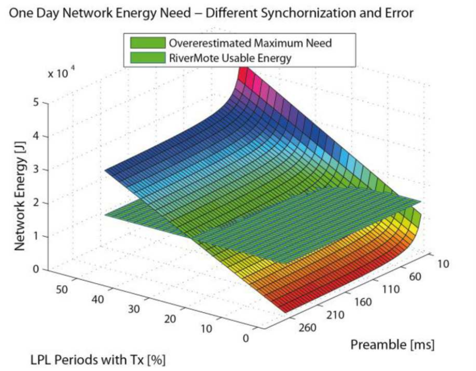
Fig. 1. Although simulation-based approaches (Avrora power profile 1(a), power states 1(c) and LPL peak 1(e)) can give good estimates, different problems can be identified as far as hardware models and timing accuracy are concerned. Hardware measurements give far more accurate results when it comes to short time radio activation as can be seen from the power profile and state switches 1(g), the complete radio activation process 1(b), its first peak from an interrupt 1(d), its second peak including full radio activation 1(f) and measuring a full LPL period 1(h) of 255ms.




(a) 8 out of 9 motes are sending.




(b) 1 out of 9 nodes is sending messages




(c) Possible network settings for a given error.

Fig. 2.The subplots analyze the energy-budget and -needs of a 9-node network where all motes are in communication range of each other. Results are shown for different LPL settings, transmission rates, MAC synchronization, modeling error and number of senders compared to the available energy budget.

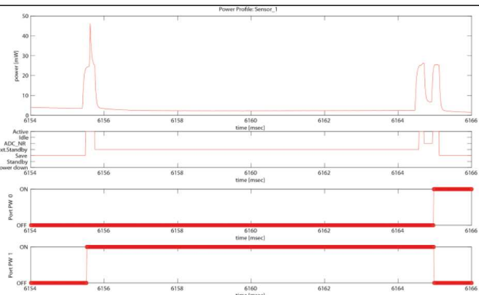
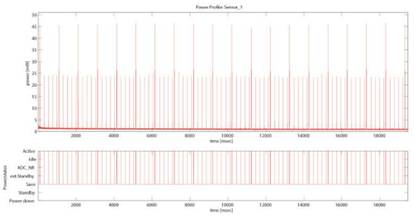
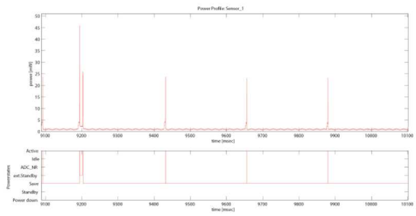


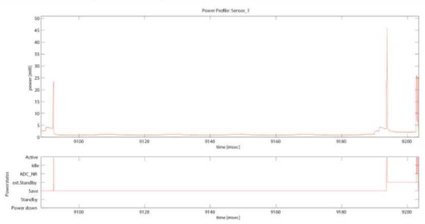
Fig. 3. Read - ReadDone for the photo sensor on the MTS300 sensor board with a Mica2 mote.



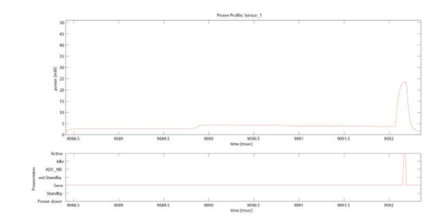
(a) Sensor power profile



(b) Sensor readout period

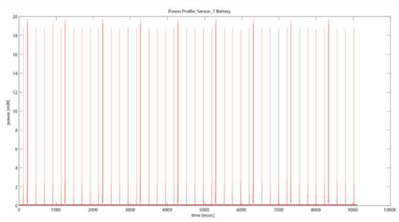


(c) Sensor – both peaks

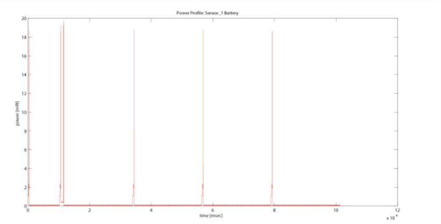


(d) Sensor – first peak

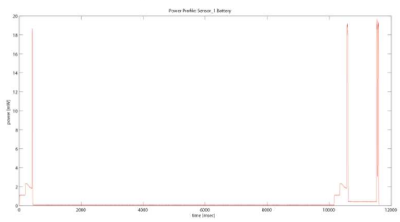
Fig. 4. The plots in 4(a) show a full 20 s 100kS measurement with Tospie2 for characterizing the power consumption of reading sensors that are directly attached to the ADC of the processor. It is an example for showing issues when running hardware measurements.



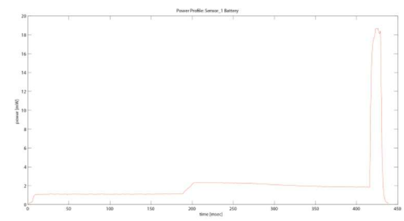
(a) Sensor power profile



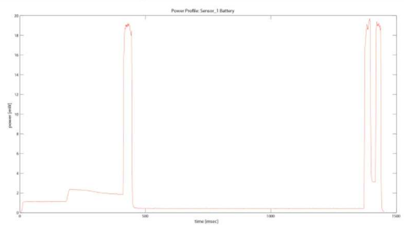
(b) Sensor Period



(c) Sensor – both peaks



(d) Sensor – first peak



(e) Sensor – second peak

Fig. 5. Corrected sensor profiles.

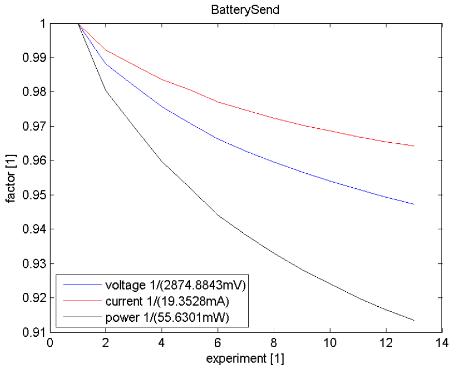



Fig. 6. The RadioCount2Leds application from TinyOS2 is profiled at different voltages of a battery pack supply.

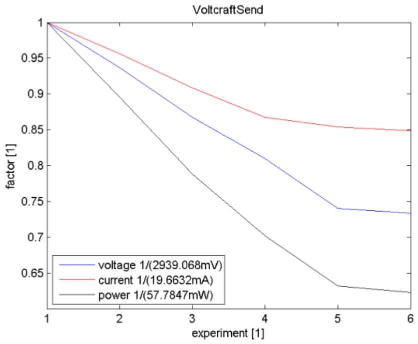



Fig. 7. The RadioCount2Leds application from TinyOS2 is profiled at different voltage settings of a laboratory power supply.

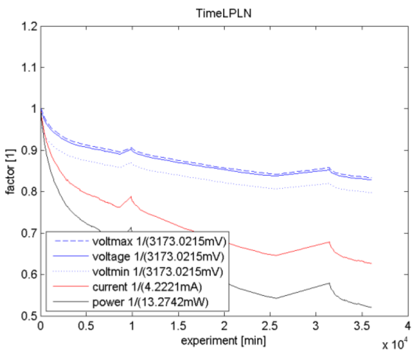



Fig. 8. The battery performance has been profiled over 25 days with two periods of relaxation in between.

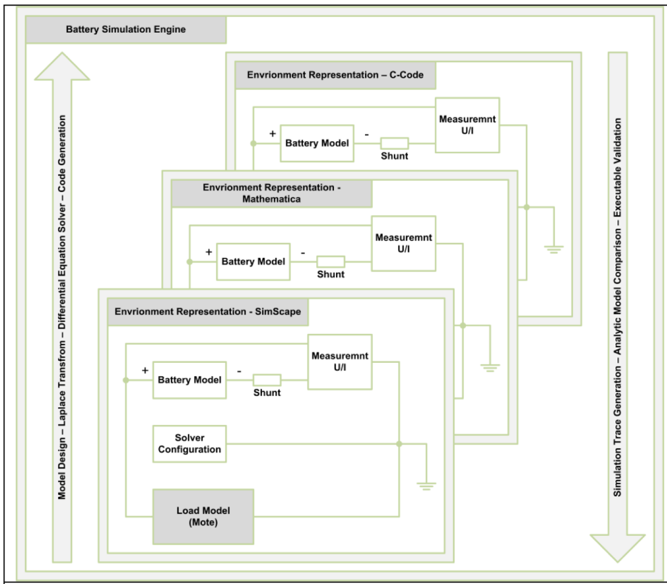



Fig. 9. Setting up a flow for battery model compilation and testing.

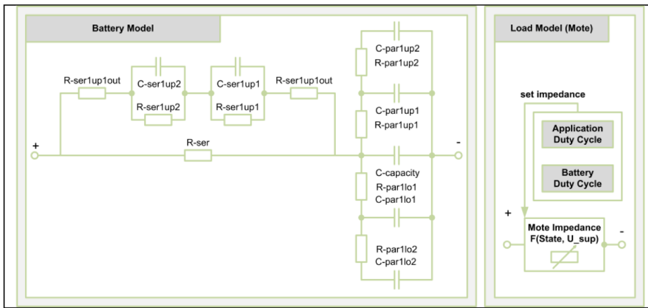



Fig. 10. Inner parts of the battery model and the load model.

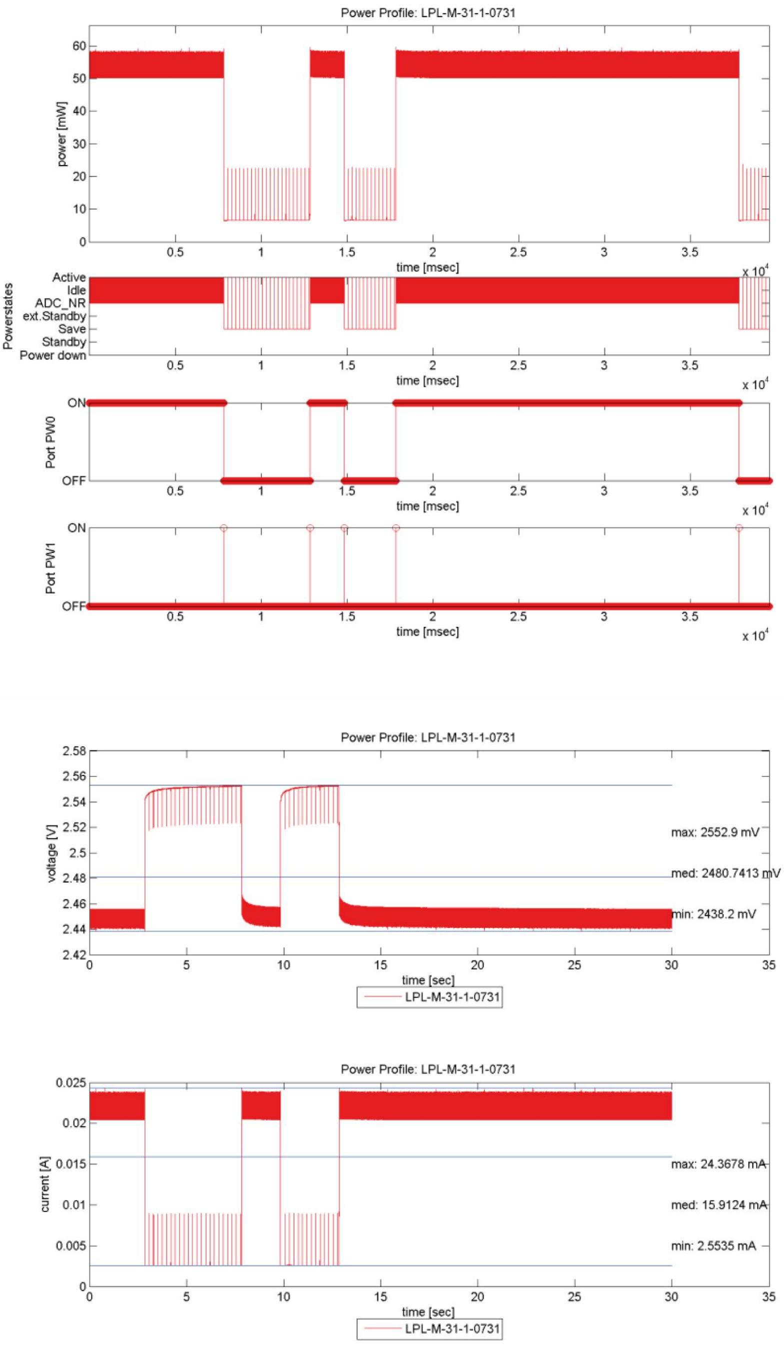



Fig. 11. Detailed output of the measurement unit for tracking power states and resulting load.

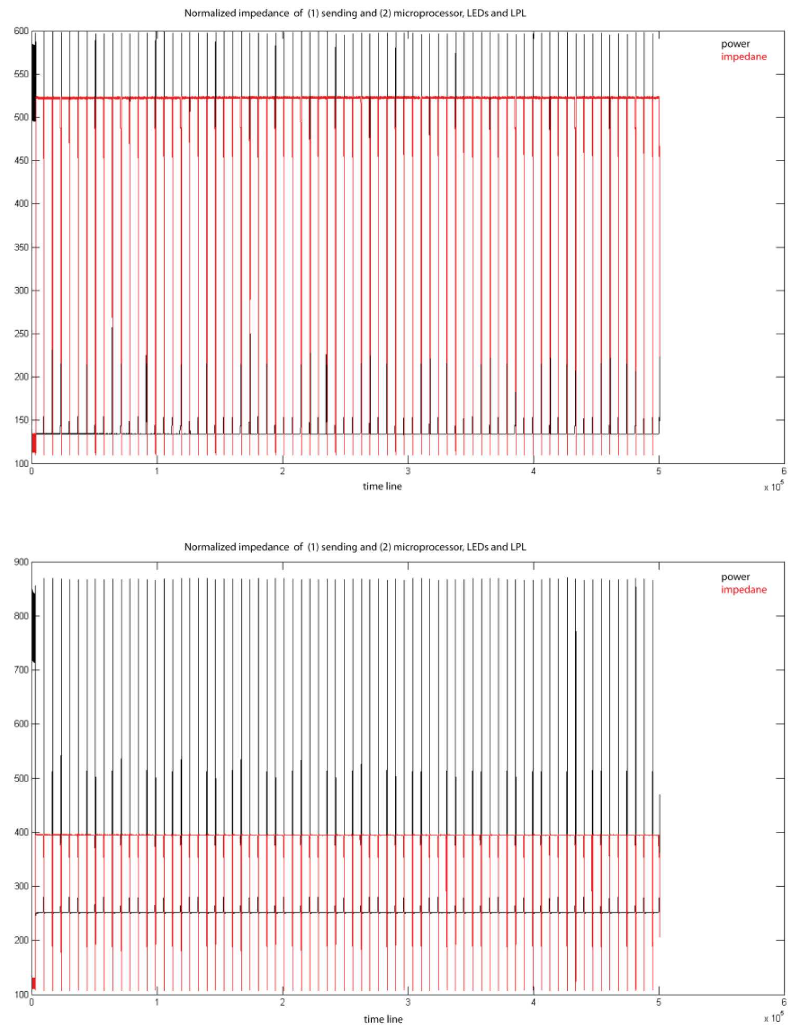



Fig. 12. The normalized resistance values are shown at the beginning of the 25-days battery performance measurement and at the end. It can clearly be seen that the normalized calculated impedance values of the LEDs and the processor change with the supply voltage, while the according values of the CC1k stays constant.

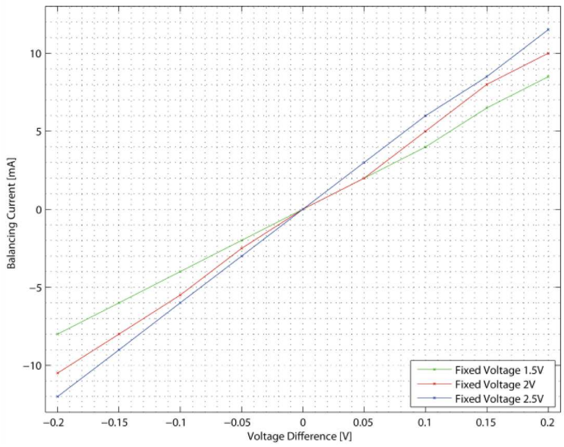



Fig. 13. Measurement of the balancing current depending on the voltage difference of the DLCs.

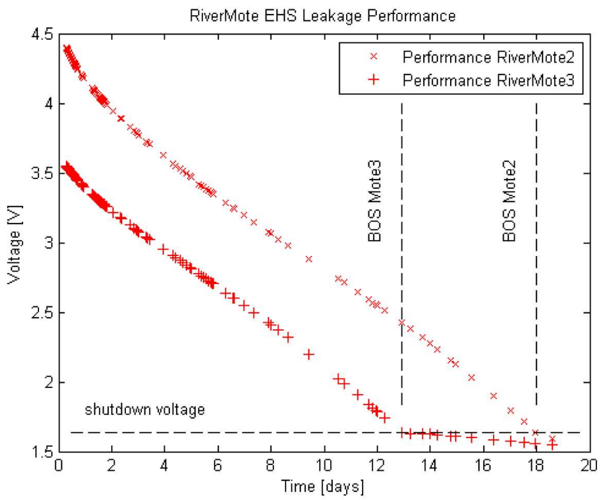



Fig. 14. System voltage over time during a complete blackout.

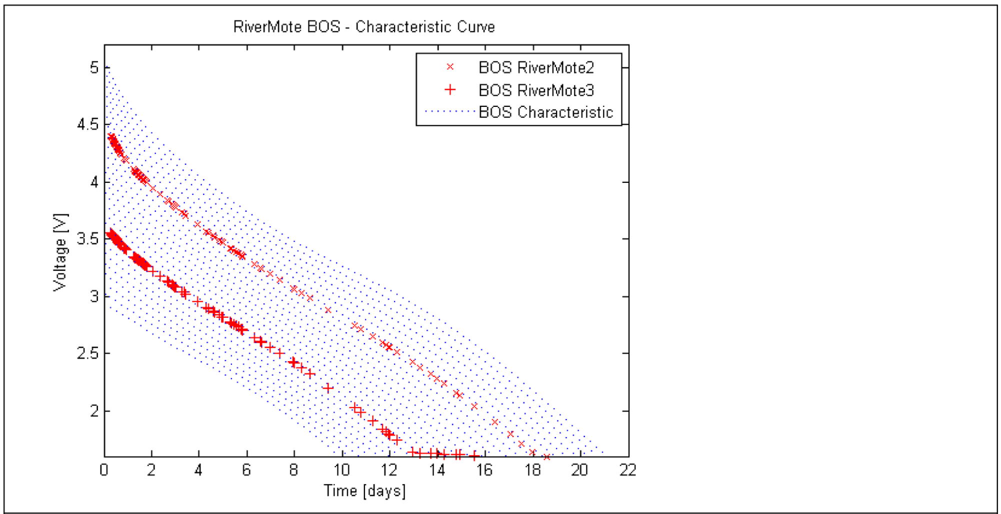



Fig. 15. The RiverMote platform BOS characteristic.

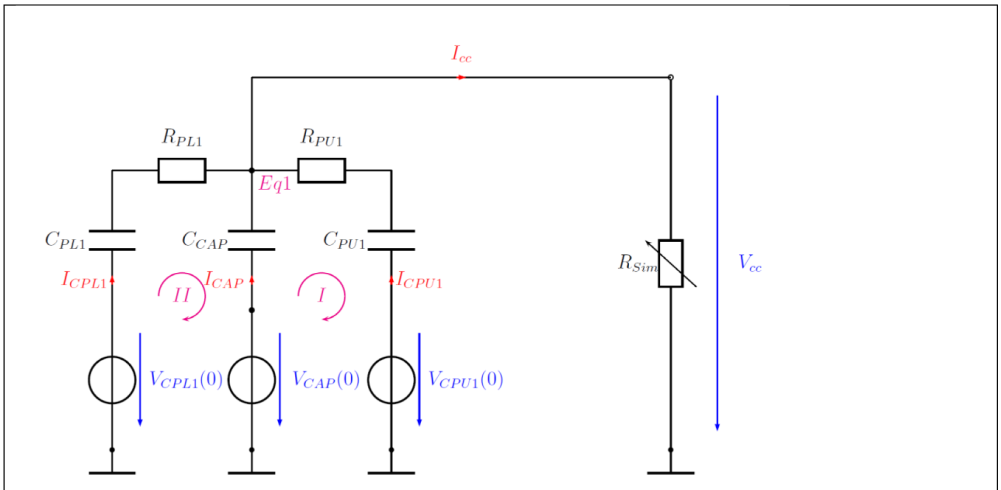



Fig. 16. Circuit of the battery model.

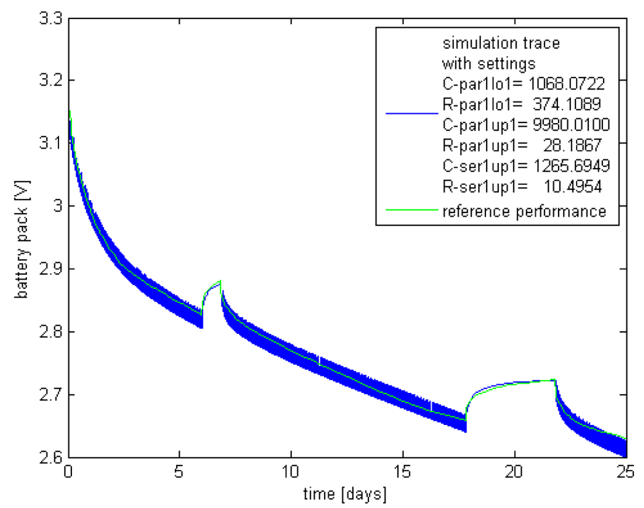



Fig. 17. The model approximates the measurement trace, that is taken as the battery pack's reference performance, very well.

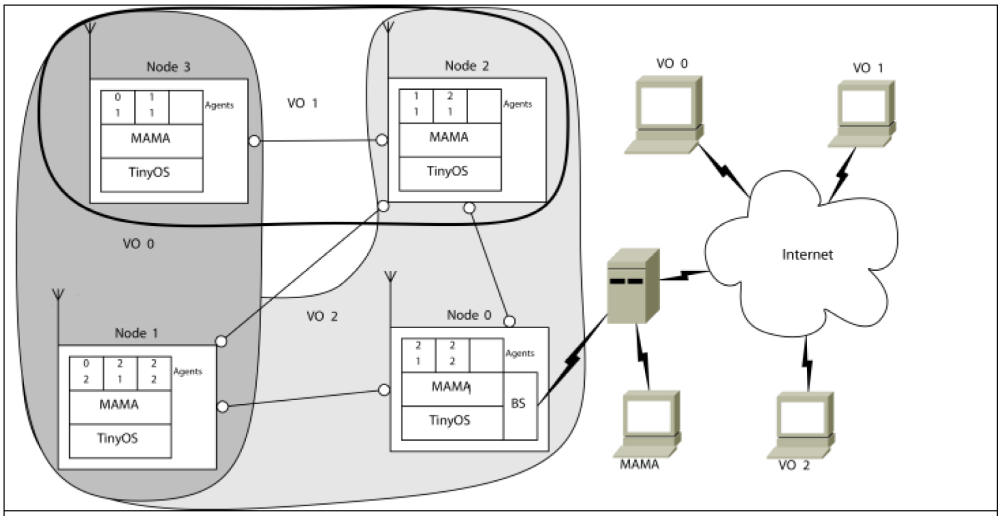



Fig. 18. Overview of the concept of the MAMA middleware.

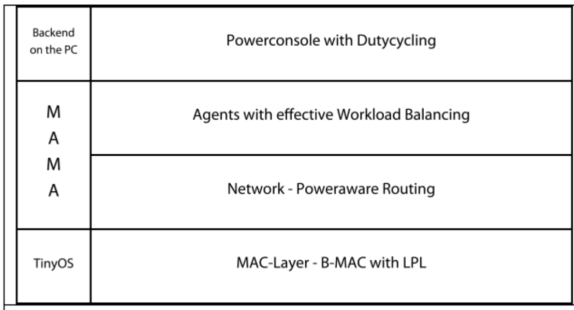



Fig. 19. Layers of power management in MAMA.

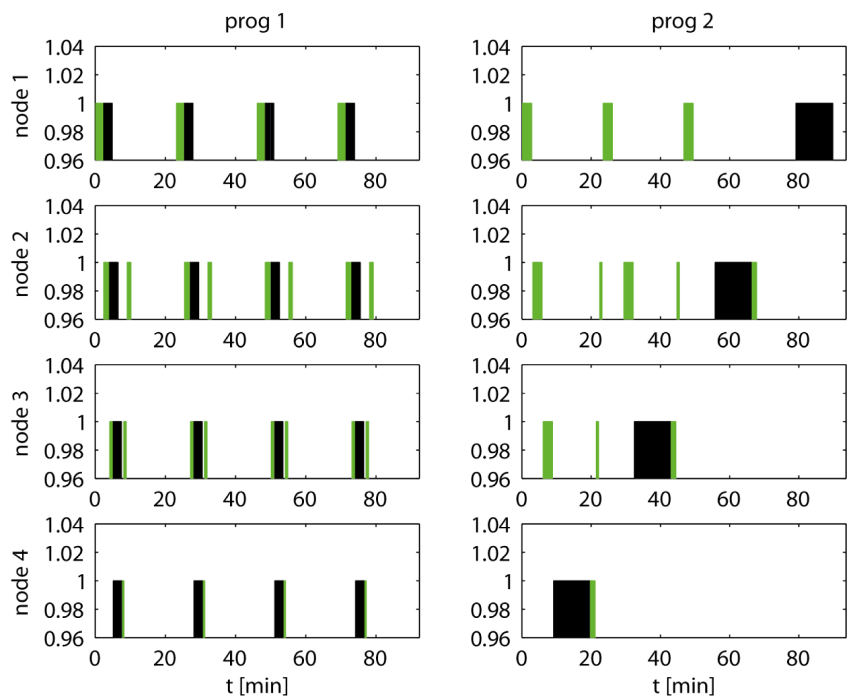



Fig. 20. Plots show that the more 'coarse-grain' resolution of the load balancing scheme achieves better results on average.

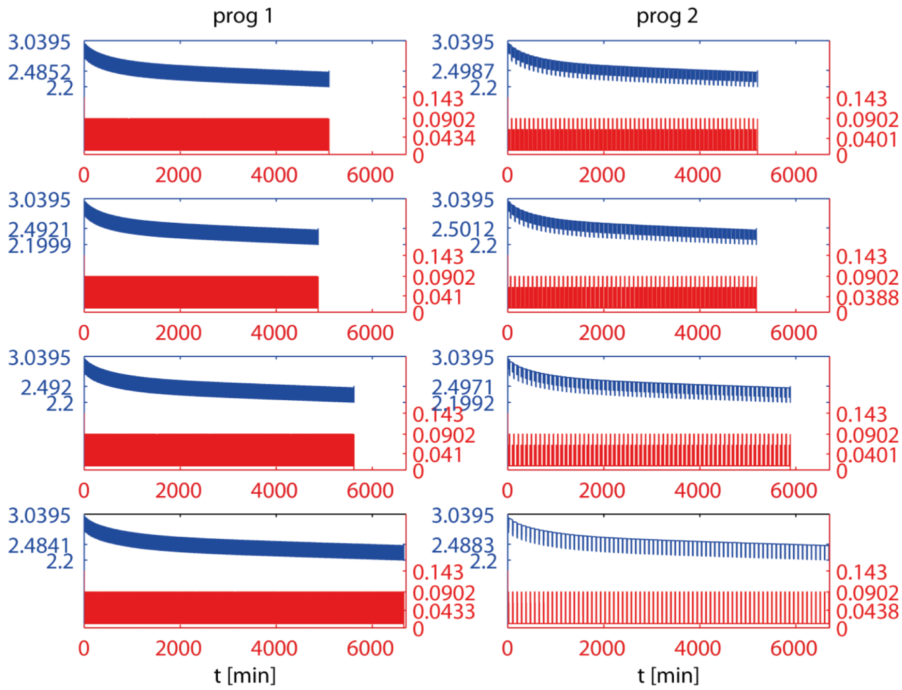



Fig. 21. For a theoretical 4-motes-in-a-line network, the burden at each mote is identified. The burden comes from balancing the activation periods for achieving the needed end-user performance, and it comes from multi-hop forwarding of results from motes that are further away from the sink mote.

# 7 Acronmys

| | |
|---|---|
| **ADC** | Analog Digital Converter |
| **ANC** | Autonomous Network Coding |
| **BOS** | Black-Out Sustainability |
| **CPU** | Central Processing Unit |
| **DC** | Duty Cycle |
| **DLC** | Double-Layer Capacitor |
| **ECD** | Equivalent Circuit Diagram |
| **EEM** | Energy Harvesting System Efficiency Model |
| **EHAR** | Energy Harvesting Aware Routing |
| **EHD** | Energy Harvesting Device |
| **EHS** | Energy Harvesting System |
| **ENO** | Energy Neutral Operation |
| **GPS** | Global Positioning System |
| **LED** | Light Emitting Diode |
| **LNC** | Localized Network Coding |
| **LPL** | Low-Power Listening |
| **MAC** | Media Access Control |
| **MAMA** | Multi-Application MiddlewAre |
| **MANET** | Mobile Ad-Hoc Network |
| **MPPT** | Maximum Power Point Tracking |
| **NC** | Network Coding |
| **NI** | National Instruments |
| **ONC** | Opportunistic Network Coding |
| **OS** | Operating System |

**OtAP** Over-the-Air Programming

**PDF** Probability Density Function

**PSM** Power State Model

**SiMeInterfacer** Simulation and Measurement Interfacer

**SNC** Scalable Network Coding

**SoC** State of Charge

**SoH** State of Health

**TDMA** Time Division Multiple Access

**TinyOS** Tiny Operating System

**TOSPIE2** Tiny Operating System Plug-In with Energy Estimation

**VM** Virtual Machine

**VO** Virtual Organization

**WSN** Wireless Sensor Network

# Bibliography


[1] P. M. Glatz, C. Steger, and R. Weiss, "TOSPIE2: Tiny Operating System Plug-In for Energy Estimation," in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010, pp. 410 – 411.

[2] P. M. Glatz and R. Weiss, "Architecturing Resource Aware Sensor Grid Middleware: Avoiding Common Errors in Design, Simulation, Test and Measurement," in Computational and Data Grids: Principles, Designs, and Applications, N. Preve, Ed. IGI Global, 2011.

[3] ——, "LINDONCS: Localized In-Network Detection Of Network Coding Structures in Wireless Sensor Networks," in Proceedings of the 4th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks, 2009, pp. 17 – 24.

[4] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Implementing Autonomous Network Coding for Wireless Sensor Network Applications," in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 9 – 14.

[5] P. M. Glatz, K. B. Hein, and R. Weiss, "Energy Conservation with Network Coding for Wireless Sensor Networks with Multiple Crossed Information Flows," in Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks, 2009, pp. 202 – 207.

[6] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Opportunistic Network Coding for Energy Conservation in Wireless Sensor Networks," in Proceedings of the Ninth Annual Communication Networks and Services Research Conference, 2011, pp. 239 – 246.

[7] P. M. Glatz, J. Loinig, C. Steger, and R. Weiss, "A First Step Towards Energy Management for Network Coding in Wireless Sensor Networks," in Proceedings of the 9th IEEE Malaysia International Conference on Communications, 2009, pp. 905–910.

[8] P. M. Glatz, K. B. Hein, and R. Weiss, "Scalable Network Coding for Wireless Sensor Network Energy Conservation," International Journal of Autonomous and Adaptive Communications, accepted for publication.

[9] P. M. Glatz, P. Meyer, A. Janek, T. Trathnigg, C. Steger, and R. Weiss, "A Measurement Platform for Energy Harvesting and Software Characterization in WSNs," in Proceedings of the IFIP/IEEE Wireless Days Conference, 2008, pp. 1 – 5.

[10] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "Designing Sustainable Wireless Sensor Networks with Efficient Energy Harvesting Systems," in Proceedings of the IEEE Wireless Communications and Networking Conference - Service and Application Track, 2011, pp. 2018 – 2023.

[11] ——, "Designing Perpetual Energy Harvesting Systems explained with RiverMote: A Wireless Sensor Network Platform for River Monitoring," Electronic Journal of Structural Engineering, Special Issue: Wireless Sensor Networks and Practical Applications, pp. 55 – 65, 2010.

[12] ——, "Low-Cost Reliable Blackout Sustainability of Wireless Sensor Networks with Energy Harvesting Systems," in Proceedings of the 17th European Wireless Conference, 2011, pp. 155 – 162.

[13] ——, "MAMA: Multi-Application MiddlewAre for Efficient Wireless Sensor Networks," in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 1 – 8.

[14] P. M. Glatz, C. Steger, and R. Weiss, "Design, Simulation and Measurement of an Accurate Wireless Sensor Network Localization System," in Proceedings of the 5th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, 2010, pp. 1 – 8.

[15] P. M. Glatz, L. B. Hörmann, C. Steger, and R. Weiss, "HANS: Harvesting Aware Networking Service for Energy Management in Wireless Sensor Networks," in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 191 – 196.

[16] ——, "Towards Modeling Support for Low-Power and Harvesting Wireless Sensors for Realistic Simulation of Intelligent Energy-Aware Middleware," in Energy Optimization and Scavenging Techniques for Mobile Devices and Networks, H. Venkataraman and G.-M. Muntean, Eds. Accepted for publication by CRC Press, 2012.

[17] ——, "A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency," in Proceedings of the Sixth IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing, 2010, pp. 468 – 473.

[18] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next Century Challenges: Mobile Networking for "Smart Dust"," in Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999, pp. 271 – 278.

[19] D. Hoareau and Y. Maheo, "Middleware support for the deployment of ubiquitous software components," Personal and Ubiquitous Computing - Special Issue: Selected Papers of the ARCS06 Conference, vol. 12, no. 2, pp. 167 – 178, 2008.

[20] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine, vol. 40, no. 8, pp. 102 – 114, 2002.

[21] L. Doherty, B. A. Warneke, B. E. Boser, and K. S. J. Pister, "Energy and Performance Considerations for Smart Dust," International Journal on Parallel and Distributed Sensor Networks, 2001.

[22] R. Szewczyk, J. Polastre, A. M. Mainwaring, and D. E. Culler, "Lessons From A Sensor Network Expedition," in Proceedings of the First European Workshop on Wireless Sensor Networks, 2004, pp. 307 – 322.

[23] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network Information Flow," IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1204 – 1216, 2000.

[24] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in Proceedings of the International Conference on Information Technology: Coding and Computing, vol. 2, 2005, pp. 8–13.

[25] I. F. Akyildiz and M. Can Vuran, Wireless Sensor Networks. John Wiley & Sons Ltd., 2010.

[26] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Elsevier Computer Networks, 2008.

[27] V. Gungor and G. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pp. 4258 – 4265, 2009.

[28] J. Hill and D. Culler, "A wireless embedded sensor architecture for system-level optimization," Technical report, U.C. Berkeley, 2001.

[29] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, 2005, pp. 364 – 369.

[30] Texas Instruments, Inc., The All New CC430 Combines Leading MSP430 MCU and Low Power RF Technology, 2009.

[31] A. Corporation, ATmega128RFA1 Preliminary Summary, www.atmel.com, 2011, 8266BS-MCU Wireless-03/11.

[32] Y.-T. Chen, T.-C. Chien, and P. H. Chou, "Enix: A Lightweight Dynamic Operating System for Tightly Constrained Wireless Sensor Platforms," in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, 2010, pp. 183 – 196.

[33] P. Levis, S. Madden, J. Polastre, R. Szewczyk, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "TinyOS: An Operating System for Sensor Networks," in Ambient Intelligence, 2004.

[34] K. Klues, V. Handziski, C. Lu, A. Wolisz, D. Culler, D. Gay, and P. Levis, "Integrating Concurrency Control and Energy Management in Device Drivers," in Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, 2007, pp. 251 – 264.

[35] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC Language: A Holistic Approach to Networked Embedded Systems," in Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation, 2003, pp. 1 – 11.

[36] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible Hardware Abstraction for Wireless Sensor Networks," in Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005, pp. 145 – 157.

[37] K. Klues, C.-J. M. Liang, J. Paek, R. Musăloiu-E, P. Levis, A. Terzis, and R. Govindan, "TOSThreads: Thread-safe and Non-Invasive Preemption in TinyOS," in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, 2009, pp. 127 – 140.

[38] S. Hadim and N. Mohamed, "Middleware for Wireless Sensor Networks: A Survey," in Proceedings of the First International Conference on Communication System Software and Middleware, 2006, pp. 1 – 7.

[39] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks," ACM SIGPLAN Notices, vol. 37, pp. 85 – 95, 2002.

[40] R. Barr, J. C. Bicket, D. S. Dantas, B. Du, T. W. D. Kim, B. Zhou, and E. G. Sirer, "On the Need for System-Level Support for Ad hoc and Sensor Networks," ACM SIGOPS Operating Systems Review, vol. 36, pp. 1 – 5, 2002.

[41] T. Liu and M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," ACM SIGPLAN Notices, vol. 38, pp. 107 – 118, 2003.

[42] W. Heinzelman, A. Murphy, H. Carvalho, and M. Perillo, "Middleware to Support Sensor Network Applications," IEEE Network, vol. 18, no. 1, pp. 6 – 14, 2004.

[43] L. Mottola and G. P. Picco, "Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art," ACM Computing Surveys, vol. 43, pp. 19:1–19:51, 2011.

[44] S. Beeby and N. White, Energy Harvesting for Autonomous Systems. Artech House, 2010.

[45] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava, "Heliomote: Enabling Long-Lived Sensor Nnetworks Through Solar Energy Harvesting," in Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, 2005, pp. 309 – 309.

[46] A. Kansal, D. Potter, and S. M. B., "Performance Aware Tasking for Environmentally Powered Sensor Networks," ACM SIGMETRICS Performance Evaluation Review, vol. 32, 2004.

[47] C. Bergonzini, D. Brunelli, and L. Benini, "Algorithms for Harvested Energy Prediction in Batteryless Wireless Sensor Setworks," in Proceedings of the 3rd International Workshop on Advances in sensors and Interfaces, 2009, pp. 144 –149.

[48] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," ACM Transactions on Embedded Computing Systems - Special Section LCTES'05, vol. 6, no. 4, p. 32, 2007.

[49] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Robust and Low Complexity Rate Control for Solar Powered Sensors," in Proceedings of the Conference on Design, Automation and Test in Europe, 2008, pp. 230 – 235.

[50] C. Kompis and S. Aliwell, "Energy Harvesting Ttechnologies to Enable Remote and Wireless Sensing,," Sensors and Instrumentation, 2008.

[51] D. Linden and T. B. Reddy, Handbook of Batteries, 3rd ed. McGraw-Hill, 2002.

[52] C. Vigorito, D. Ganesan, and A. Barto, "Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks," in Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2007, pp. 21 – 30.

[53] A. Kansal, J. Hsu, M. Srivastava, and V. Raqhunathan, "Harvesting Aware Power Management for Sensor Networks," in 43rd ACM/IEEE Design Automation Conference, 2006, pp. 651 – 656.

[54] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in Proceedings of the IEEE Wireless Communications and Networking Conference, 2002, pp. 350 – 355 vol.1.

[55] D. Hasenfratz, A. Meier, C. Moser, J.-J. Chen, and L. Thiele, "Analysis, Comparison, and Optimization of Routing Protocols for Energy Harvesting Wireless Sensor Networks," in Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2010, pp. 19 – 26.

[56] L. Lin, N. B. Shroff, and R. Srikant, "Energy-Aware Routing in Sensor Networks: A Large System Approach," Elsevier Ad Hoc Networks, vol. 5, pp. 818 – 831, 2007.

[57] K. G. Langendoen, "Medium access control in wireless sensor networks," in Medium Access Control in Wireless Networks, H. Wu and Y. Pan, Eds. Nova Science Publishers, Inc., 2008, pp. 535 – 560.

[58] V. Joseph, V. Sharma, and U. Mukherji, "Optimal Sleep-Wake Policies for an Energy Harvesting Sensor Node," in Proceedings of the IEEE International Conference on Communications, 2009, pp. 52 – 57.

[59] N. Jaggi, K. Kar, and A. Krishnamurthy, "Near-Optimal Activation Policies in Rechargeable Sensor Networks under Spatial Correlations," ACM Transactions on Sensor Networks, vol. 4, no. 3, pp. 1 – 36, 2008.

[60] ——, "Rechargeable Sensor Activation under Temporally Correlated Events," Wireless Networks - Special Issue: Selected papers from WiOpt'07, vol. 15, pp. 619 – 635, 2009.

[61] Y. Gu and T. He, "Bounding Communication Delay in Energy Harvesting Sensor Networks," in Proceedings of the International Conference on Distributed Computing Systems, 2010, pp. 837 – 847.

[62] X. Jiang, J. Polastre, and D. Culler, "Perpetual Environmentally Powered Sensor Networks," in Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, 2005, pp. 463 – 468.

[63] S. Sudevalayam and P. Kulkarni, "Energy Harvesting Sensor Nodes: Survey and Implications," IEEE Communications Surveys & Tutorials, vol. PP, no. 99, pp. 1 – 19, 2010.

[64] F. B. James M. Gilbert, "Comparison of Energy Harvesting Systems for Wireless Sensor Networks," International Journal of Automation and Computing, vol. 5, pp. 334 – 347, 2008.

[65] A. Janek, C. Trummer, C. Steger, R. Weiss, J. Preishuber-Pfluegl, and M. Pistauer, "Simulation Based Verification of Energy Storage Architectures for Higher Class Tags supported by Energy Harvesting Devices," Microprocessors & Microsystems, vol. 32, pp. 330 – 339, 2008.

[66] L. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Energy Efficient Supply of WSN Nodes using Component-Aware Dynamic Voltage Scaling," in Proceedings of the 17th European Wireless Conference, 2011, pp. 147 – 154.

[67] C. Park and P. H. Chou, "Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes," in Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006.

[68] F. Simjee and P. Chou, "Everlast: Long-life, Supercapacitor-operated Wireless Sensor Node," in Proceedings of the International Symposium on Low Power Electronics and Design, 2006, pp. 197 – 202.

[69] V. Joseph, V. Sharma, and U. Mukherji, "Joint Power Control, Scheduling and Routing for Multihop Energy Harvesting Sensor Networks," in Proceedings of the 4th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, 2009, pp. 128 – 136.

[70] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks," Foundations and Trends in Networking, vol. 1, pp. 1 – 144, 2006.

[71] X. Lin, N. Shroff, and R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," IEEE Journal on Selected Areas in Communications, vol. 24, no. 8, pp. 1452 – 1463, 2006.

[72] A. Ramamoorthy, K. Jain, P. A. Chou, and M. Effros, "Separating Distributed Source Coding From Network Coding," IEEE/ACM Transactions on Networking, vol. 14, pp. 2785 – 2795, 2006.

[73] S. E. Tajbksh and A. Movaghar, “Joint Distributed Source and Network Coding for Multiple Wireless Unicast Sessions,” in Proceedings of the 2009 IEEE conference on Wireless Communications and Networking Conference, 2009, pp. 302 – 306.

[74] S. Dulman, T. Nieberg, J. Wu, and P. Havinga, “Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks,” in Proceedings of the IEEE Wireless Communications and Networking Conference, 2003, pp. 1918–1922 vol.3.

[75] A. Das and D. Dutta, “Data Acquisition in Multiple-sink Sensor Networks,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 9, no. 3, pp. 82–85, 2005.

[76] A. Eryilmaz, “Control for Inter-Session Network Coding,” in Proceedings of the Workshop on Network Coding, Theory & Applications, 2007.

[77] W. Chen, K. Letaief, and Z. Cao, “Opportunistic Network Coding for Wireless Networks,” in Proceedings of the IEEE International Conference on Communications, 2007, pp. 4634 – 4639.

[78] I. Stojmenovic, “Geocasting with Guaranteed Delivery in Sensor Networks,” IEEE Wireless Communications, vol. 11, no. 6, pp. 29 – 37, 2004.

[79] Q. Cao, T. He, and T. Abdelzaher, “uCast: Unified Connectionless Multicast for Energy Efficient Content Distribution in Sensor Networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 2, pp. 240 – 250, 2007.

[80] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, “The Importance of Being Opportunistic: Practical Network Coding for Wireless Environments,” in In Proceedings of 43rd Allerton Conference on Communication, Control, and Computing, 2005.

[81] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, “Network Coding: An Instant Primer,” ACM SIGCOM Computer Communication Review, 2006.

[82] T. Ho and D. Lun, Network Coding: An Introduction. Cambridge University Press, 2008.

[83] Z. Li, B. Li, D. Jiang, and L. C. Lau, “On Achieving Optimal Throughput with Network Coding,” in Proceedings of the 24th Annual Joint Conference of the IEEE Communications Societies, 2005, pp. 2184 – 2194 vol. 3.

[84] R. Koetter and M. Médard, “An algebraic approach to network coding,” IEEE/ACM Transactions on Networking, vol. 11, no. 5, pp. 782 – 795, 2003.

[85] M. . S. M. R. Jain, Kamal; Mahdian, “Packing Steiner trees,” in Proceedings of the Fourteenth Annual Acm-Siam Symposium on Discrete Algorithms, 2003.

[86] P. Sanders, S. Egner, and L. Tolhuizen, “Polynomial Time Algorithms for Network Information Flow,” in Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures, 2003, pp. 286 – 294.

[87] Z. Li and B. Li, "Network coding in undirected networks," 2004.

[88] R. Flury and R. Wattenhofer, "Slotted Programming for Sensor Networks," in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010, pp. 24 – 34.

[89] S. Katti, H. Rahul, H. Wenjun, K. Dina, M. Mé dard, and C. Jon, "XORs in the Air: Practical Wireless Network Coding," IEEE/ACM Transactions on Networking, 2008.

[90] R. Niati, T. Kunz, and A. H. Banihashemi, "The Effect of Transmission Capacities on Interference-free Scheduling in Wireless Networks," in Proceedings of the Ninth Annual Communication Networks and Services Research Conference, 2011, pp. 257 – 262.

[91] H. Park, W. Liao, K. H. Tam, M. B. Srivastava, and L. He, "A Unified Network and Node Level Simulation Framework for Wireless Sensor Networks," University of California, Los Angeles, Tech. Rep., 2003.

[92] 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, 2005, pp. 477 – 482.

[93] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate Prediction of Power Consumption in Sensor Networks," in Proceedings of the Second IEEE Workshop on Embedded Networked Sensors, 2005, pp. 37 – 44.

[94] 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, 2003, pp. 126–137.

[95] G. Merrett, N. White, N. Harris, and B. Al-Hashimi, "Energy-Aware Simulation for Wireless Sensor Networks," in Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009, pp. 1 – 8.

[96] T. Trathnigg and R. Weiss, "Towards Runtime Support for Energy Awareness in Wireless Sensor Networks," in Proceedings of the First International Workshop on Wireless Sensor Networks, 2007, pp. 49 – 54.

[97] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, "PowerBench: A Scalable Testbed Infrastructure for Benchmarking Power Consumption," in Proceedings of the International Workshop on Sensor Network Engineering, 2008.

[98] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro Power Meter for Energy Monitoring of Wireless Sensor Networks at Scale," in Proceedings of the 6th International Symposium on Information Processing in Sensor Networks, 2007, pp. 186 – 195.

[99] 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.

[100] G. Chen, J. Branch, M. J. Pflug, L. Zhu, and B. K. Szymanski, "Sense: A Wireless Sensor Network Simulator," in Proceedings of the Advances in Pervasive Computing and Networking, B. Szymanski and B. Yener, Eds. Springer, New York, NY, USA, 2005, p. 249267.

[101] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in Proceedings of the 31st IEEE Conference on Local Computer Networks, 2006, pp. 641 – 648.

[102] T. Issariyakul and E. Hossain, Introduction to Network Simulator NS2. Springer, 2009.

[103] H. Wu, Q. Luo, P. Zheng, and L. M. Ni, "VMNet: Realistic Emulation of Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 2, pp. 277 – 288, 2007.

[104] J. Elson, L. Girod, and D. Estrin, "EmStar: Development with High System Visibility," IEEE Wireless Communications, vol. 11, no. 6, pp. 70 – 77, 2004.

[105] C. Park and P. Chou, "AmbiMax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes," in Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2006, pp. 168 – 177.

[106] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "A SystemC-AMS Simulation Environment for the Evaluation of Energy Harvesting Wireless Sensor Networks," in Proceedings of the 2011 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2011.

[107] L. Mateu and F. Moll, "System-level simulation of a self-powered sensor with piezoelectric energy harvesting," in Proceedings of the International Conference on Sensor Technologies and Applications, 2007, pp. 399 – 404.

[108] A. Seyedi and B. Sikdar, "Modeling and Analysis of Energy Harvesting Nodes in Wireless Sensor Networks," in Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing, 2008, pp. 67 – 71.

[109] P. De Mil, B. Jooris, L. Tytgat, R. Catteeuw, I. Moerman, P. Demeester, and A. Kamerman, "Design and Implementation of a Generic Energy-Harvesting Framework Applied to the Evaluation of a Large-Scale Electronic Shelf-Labeling Wireless Sensor Network," EURASIP Journal onWireless Communications and Networking, vol. 2010, pp. 7:1–7:14, 2010.

[110] "Duracell, Entire MnO2 Technical Bulletin Collection.".

[111] V. Pop, H. J. Bergveld, P. H. L. Notten, and P. P. L. Regtien, "REVIEW ARTICLE: State-of-the-art of battery state-of-charge determination," Measurement Science and Technology, vol. 16, pp. R93 – R110, 2005.

[112] B. Schweighofer, K. Raab, and G. Brasseur, “Modeling of High Power Automotive Batteries by the Use of an Automated Test System,” IEEE Transactions on Instrumentation and Measurement, vol. 52, no. 4, pp. 1087 – 1091, 2003.

[113] S. Abu-Sharkh and D. Doerffel, “Rapid test and non-linear model characterisation of solid-state lithium-ion batteries,” Journal of Power Sources, vol. 130, no. 1-2, pp. 266 – 274, 2004.

[114] R. Rao, S. Vrudhula, and D. Rakhmatov, “Battery Modeling for Energy Aware System Design,” IEEE Computer, vol. 36, no. 12, pp. 77 – 87, 2003.

[115] Z. Salameh, M. Casacca, and W. Lynch, “A Mathematical Model for Lead-Acid Batteries,” IEEE Transactions on Energy Conversion, vol. 7, no. 1, pp. 93 – 98, 1992.

[116] M. Pedram and Q. Wu, “Design Considerations for Battery-Powered Electronics,” in Proceedings of the 36th annual ACM/IEEE Design Automation Conference, 1999, pp. 861 – 866.

[117] P. Nuggehalli, V. Srinivasan, and R. Rao, “Energy Efficient Transmission Scheduling for Delay Constrained Wireless Networks,” IEEE Transactions on Wireless Communications, vol. 5, no. 3, pp. 531 – 539, 2006.

[118] C. Ma and Y. Yang, “Battery-Aware Routing for Streaming Data Transmissions in Wireless Sensor Networks,” Kluwer Mobile Networks and Applications, vol. 11, pp. 757 – 767, 2006.

[119] C.-F. Chiasserini and R. Rao, “Improving Battery Performance by Using Traffic Shaping Techniques,” IEEE Journal on Selected Areas in Communications, vol. 19, no. 7, pp. 1385 – 1394, 2001.

[120] M. Technologies, YEG Components, hc power series ultracapacitors datasheet.

[121] EPCOS, ”Ultracap single cell 5 F / 2.3V B49100A1503Q000 data sheet,”.

[122] M. Belleville, E. Cantatore, H. Fanet, P. Fiorini, P. Nicole, M. Pelgrom, C. Piguet, R. Hahn, C. V. Hoof, R. Vullers, and M. Tartagni, “Energy Autonomous Systems: Future Trends in Devices, Technology, and Systems,” CATRENE - Cluster for Application and Technology Researchin Europe on Nanoelectronics, 2009.

[123] B. Chu, ”Selecting the Right Battery System For Cost-SensitivePortable Applications While Maintaining Excellent Quality,” Microchip Technology Inc., 12 pages.

[124] M. D. Stoller, S. Park, Y. Zhu, J. An, and R. S. Ruoff, “Graphene-Based Ultracapacitors,” ACS Nano Letters, vol. 8, pp. 3498 – 3502, 2008.

[125] Z.-J. Fan, J. Yan, T. Wei, G.-Q. Ning, L.-J. Zhi, J.-C. Liu, D.-X. Cao, G.-L. Wang, and F. Wei, “Nanographene-Constructed Carbon Nanofibers Grown on Graphene Sheets by Chemical Vapor Deposition: High-Performance Anode Materials for Lithium Ion Batteries,” ACS Nano, vol. 5, no. 4, pp. 2787 – 2794, 2011.

[126] J. Loinig, P. M. Glatz, C. Steger, and W. R., "Performance Improvement and Energy Saving based on Increasing Locality of Persistent Data in Embedded Systems," in Proceedings of the Fifth International Conference on Systems, 2010, pp. 175 – 180.

[127] L. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Designing of Efficient Energy Harvesting Systems for Autonomous WSNs Using a Tier Model," in Proceedings of the 18th International Conference on Telecommunications, 2011, pp. 9 – 14.

[128] ——, "A Wireless Sensor Node for River Monitoring Using MSP430 and Energy Harvesting," in Proceedings of the 4th European DSP in Education and Research Conference, 2010, pp. 140 – 144.

[129] L. B. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Energy Efficient Supply of Mobile Devices," in Energy Optimization and Scavenging Techniques for Mobile Devices and Networks, H. Venkataraman and G.-M. Muntean, Eds. Accepted for publication by CRC Press, 2012.

[130] L. Hörmann, P. M. Glatz, C. Steger, and R. Weiss, "Evaluation of Component-Aware Dynamic Voltage Scaling for Mobile Devices and Wireless Sensor Networks," in Proceedings of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks - Industry Track, 2011.