A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks

Jens Horneber and Anton Hergenröder

***Abstract*—Research results in wireless sensor networks are primarily gained from simulations and theoretical considerations. Currently, the community begins to realize that the results need to be validated in testbeds. Testbeds can be also used to directly gather knowledge with sensor network experiments on real hardware and a real environmental context. This survey gives an overview of different approaches to build testbeds and experimentation environments regarding different research foci. We discuss emerging testbed requirements and present existing solutions of the community. The overview is complemented with a discussion of common design decisions concerning architectures and experimentation support in current testbeds. A look on future trends and developments in wireless sensor network testbeds concludes this paper. This survey is intended to help researchers to attain own results in real-world experiments with wireless sensor networks. The reader gains a comprehensive overview on existing testbeds and practical knowledge documented in the referenced literature. The survey is laying the foundation for design decisions while developing an own testbed using the examples of described approaches.**

***Index Terms*—Survey, wireless sensor networks (WSNs), testbeds, testing and debugging, performance of systems.**

I. Introduction

WIRELESS sensor networks (WSNs) consist of small-sized devices with limited computing power, so called sensor nodes. Application scenarios for WSNs include autonomous and long term environment monitoring, wearable health care gadgets and unobtrusive home automation. In these scenarios the applications benefit from the decentralized and cooperative WSN capabilities. Small-sized sensor nodes and low per node costs enable large networks and make computation applications in the environments of our daily life possible.

Hence, sensor networks have sparked a lot of interest and have been extensively researched in the past. A lot of WSN platforms, protocols and applications have been developed. More and more WSNs have been realized and have become commercially available.

Common desired qualities of WSNs are autonomous operation, unobtrusive node sizes and also a sufficient performance and node lifetimes to suit the applications purposes. However, all these qualities are interrelated. Consequently, trade-offs have to be found for a feasible design of heavily resource constrained sensor nodes. The most important constraint is the limitation of the available energy, making energy efficient protocols a major research topic in WSNs.

Research and development of WSNs requires a comprehensive, reproducible and verifiable evaluation process. Early scientific research on WSNs mostly comprises theoretical analysis and evaluation of the contributions, caused by the lack of evaluation tools and sparse platform availability. With emerging WSN operating systems, development tools, simulators and platforms, the evaluation of real implementations in simulators or even on real hardware became feasible. In fact, significant and reliable theoretical or simulative evaluation is extremely challenging, because of the great effort involved in validating and verifying results. The distributed nature of many WSN applications adds another challenge to evaluation. Various aspects, introduced by distributed application logic, the environment and radio communication over a shared medium are hardly ever to simulate. Consequently, theoretical and simulative evaluation results can only be considered approximative. Even excellent approximations have to be confirmed by real world measurements. Due to this fact, real world experiments lately increased in popularity. Experimentation on real sensor nodes is usually done in so called testbeds.

Testbeds are special environments, designed for experimentation, ranging from simple systems with a few WSN nodes on a table up to large scale deployments with hundreds of nodes. Many different testbeds have been developed in the past motivated by the need to evaluate specific research works, e.g., novel mechanisms, protocols, applications or platforms. Accordingly, many existing testbeds focus on evaluation of certain WSN aspects. Examples for such foci are mobility, energy efficiency, the design of various application layers and protocols or the development of theoretical models for simulation. Some testbeds however, are designed to be more versatile and cover more than just one focus. Moreover, several general purpose testbeds have the design goal to provide a universal experimentation environment, to enable maximum scope for WSN research.

In this survey, we present our investigations of WSN testbeds, ranging from discontinued testbeds over existing work to current trends in WSN experimentation and testbed research. We only considered testbeds that deployed multiple interconnected hardware sensor nodes. That is why we don't cover middleware approaches on sensor nodes as a foundation for testbed experiments. Specialized approaches developed to evaluate a single use case, application or protocol are also not in the scope of this survey. Besides, we rather focused on pure sensor network testbeds than hybrid approaches combining different wireless or wired device classes.

Manuscript received January 8, 2013; revised June 24, 2013 and February 3, 2014; accepted March 21, 2014. Date of publication April 24, 2014; date of current version November 18, 2014. The associate editor coordinating the review of this paper and approving it for publication was E. Hossain.

The authors are with the Institute of Telematics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/COMST.2014.2320051

TABLE I
ALL CONSIDERED SENSOR NETWORK TESTBEDS

Testbed / Project Name	Year of Publication	Institution	References	Related Sections
GNOMES	2003	Texas Instruments / Rice University	[1]	IV-A
SeNeTs	2004	University of Rostock	[2]	II-[C-D, H], III-A
MoteLab	2005	Havard University	[3]	II-A, III-C, IV-B
ExScal	2005	Multiple US-Universities / DARPA / Ohio State	[4]	II-F
TOSHILT	2005	CMU Pittsburgh	[5]	II-[D, G]
Tutornet	2005	University of South California	[6]	III-B
TrueMobile / Mobile Emulab	2005 - 2006	University of Utah	[7], [8]	II-[A-B, D]
TWIST	2005 - 2008	TU Berlin / Washington University in St. Louis	[9]–[11]	II-[A, I], III-[B-C], IV-A
SensorScope	2005 - 2008	École Polytechnique Fédérale de Lausanne	[12], [13]	III-A
Kansei	2006	Ohio State University	[14], [15]	II-C-F, III-B-C, IV-B
SignetLab	2006	University of Padova	[16], [17]	IV-B
S-WiNeTest / SensoNet	2006	University of Catania / GeorgiaTech	[18], [19]	II-I, III-A-B, IV-B
– without name –	2006	University of Ottawa	[20]	II-I
SWARMS	2007	University of Colorado	[21]	III-B
Motescope with iCount	2007-2008	UC Berkeley / MIT Cambridge	[22], [23]	II-[A, I], IV-B
Trio	2006	UC Berkeley	[24]	II-[C, F], III-A-C, IV-B
Omega / sMote	2006 - 2007	UC Berkeley	[25]	II-I
SenseNeT	2007	Athens Information Technology, Peania, Greece	[26]	II-F, III-A
FlockLab (DSN successor)	2009	ETH Zurich	[27], [28]	II-[A, H], III-B, IV-[A-B]
JAWS-DSN	2005 - 2009	ETH Zurich	[29]	II-H, III-A
wSCoop	2007	University of Darmstadt	[30]	II-B
Casino Lab	2007 - 2010	Colorado School of Mines	[31], [32]	II-[C-D], III-B
KonTest	2008	Vrije Universiteit Amsterdam	[33]	IV-B
WSNTB	2008	National Tsing Hua University, Taiwan	[34]	II-E
NetEye	2008	Wayne State University	[35]	II-E, IV-B
WINTeR	2008	Canadian Petroleum Applications of Wireless Systems (PAWS)	[36]	II-C
– without name –	2008	University of North Texas	[37]	IV-A
PowerBench	2008 - 2009	Delft University	[38], [39]	II-A
WiLab	2009	Ghent University	[40]	II-[A, C], III-B
MoteMaster	2009	RWTH Aachen	[41]	II-F
X-Sensor	2009	Osaka University / NICT Japan	[42]	III-B
– without name –	2009	Chungnam National University Daejon	[43]	II-[C, I]
– without name –	2009	Hamburg University of Technology	[44]	II-[A, H]
Sensei-UU	2009 - 2010	University of Uppsala / NEC / SAP et al.	[45]–[47]	II-[B-C], III-B, IV-B
SANDbed	2009 - 2014	Karlsruhe Institute of Technology	[48]–[51]	II-[A, C, E, H-I], III-[B-C]
EasiTest	2010	Chinese Academy of Science Beijing / Univ. of Hong Kong, China	[52], [53]	II-[C, E]
SensLAB	2010	INRIA Strasbourg / Lille / Grenoble	[54]	II-B
HINT	2010	Chinese Academy of Science Beijing / SFU Burnaby Vancouver	[55]	II-A
MOTEL	2011	SUPSI: University of Applied Sciences of Southern Switzerland	[56]	II-[B, H], III-A, IV-B
Indriya	2011	National University of Singapore	[57]	II-[E-F], III-C

In Table I we provide a chronological overview of existing work on WSN testbeds we consider in this survey. Testbed federations projects consisting of multiple interconnected sensor network testbeds are complemented in Table II. The following Table III lists additional testbed related projects that do not primary focus on sensor networks but on wireless networks in general. Nevertheless, the project contributions have important impact on the design of many current sensor network testbeds. At last, Table IV presents tools and projects that do not contribute entire testbeds but are still related and referenced

TABLE II
ALL CONSIDERED SENSOR NETWORK TESTBED FEDERATIONS

Testbed Federation / Project Name	Year of publication	Institution	References	Related Sections
WHYNET	2003 - 2007	Multiple US-Universities, Industries / DARPA	[58]	II-E
WISEBED	2008 - 2010	Multiple European Universities	[59]–[61]	II-E
KanseiGenie	2010	Ohio and Wayne State University	[62]	II-E

TABLE III
ALL CONSIDERED WIRELESS NETWORKING TESTBEDS

Testbed / Project Name	Year of publication	Institution	References	Related Sections
EmStar	2003	UC Los Angeles	[63]	II-D
ORBIT	2005	WINLAB, Rutgers University	[64], [65]	II-[B-C]
Roofnet	2005	MIT	[66]	II-I
MiNT	2005 - 2006	Stony Brook University	[67], [68]	II-B
DES-Testbed	2006 - 2010	FU Berlin	[69]–[75]	II-E, III-[B-C], IV-B
CitySense	2007	Harvard University	[76]	II-C

TABLE IV
ALL CONSIDERED TESTBED RELATED TOOLS

Tool / Project Name	Year of publication	Institution	References	Related Sections
Mirage	2005	Intel Research Berkeley	[77]	III-C
ATMA	2005	UC Santa Babara	[78]	III-C
Clamp-on current probe	2005	University of Alabama / Alaska	[79]	II-A
SPOT	2007	UC Berkeley	[80]	II-A
Hyperion	2007	University of Karlsruhe	[81]	II-G
TARWIS (former MARWIS)	2008 - 2010	University of Bern	[82]–[84]	III-C
SNMD	2009	Karlsruhe Institute of Technology	[51]	II-A

in the further sections. We added the year of publication to the reference overview to track the ideas and approaches of the different testbeds chronologically. To find the discussed ideas and individual testbed research topics quickly, we also reference the relevant survey sections in each table.

This survey is intended to give interested researchers, who want to build their own testbed or have to choose a suited experimentation approach, a guideline to identify their problem specific key requirements. It will confront them with testbed design decisions and challenges they have to solve in order to systematically experiment with real sensor networks.

We investigated previous studies on testbeds and experimentation environments for wireless sensor networks. An impressive amount of tools and testbeds is given in [85], but the article is structured more as a listing with little details, than as a comprehensive discussion of characteristics, benefits and drawbacks of the presented tools. The authors of [86] discuss benefits and drawbacks of simulators, emulators or testbeds, on the basis of various investigated tools for each approach. In [87] the authors classified the investigated testbeds by their type, mainly based on the testbed architecture. Some types identified by the authors don't really fit in this classification, so many testbeds don't clearly match a single type. Many surveys investigate tools and testbeds for evaluation of WSNs designed for a particular use case. In [88] the authors focus on wireless video sensor networks, discussing many existing platforms and also proposing their own solution. A survey on wireless multimedia sensor network hardware and testbeds is given in [89] and [90]. In [91] authors provide detailed description of NetEye [35] testbed design, covering testbed hardware, management and user tools. A short survey of several existing testbeds is also given. Most related work was done by the authors in [92], where a structured survey of testbed scopes, architectures, features and services is given. As to our knowledge we provide the most comprehensive overview of discontinued and existing testbeds for wireless sensor networks. We discuss in-depth the requirements behind testbed features and how they are met in the investigated solutions. Considering the increasing importance of energy measurements on real nodes in WSN related publications, we especially examine measurement methods of energy consumption in testbeds. Furthermore, we identify emerging trends in experimentation and thus new requirements to WSN testbeds.

The remainder of this survey is structured as follows. In Section II we identify the research foci of considered testbeds and describe how the challenges of each focus are met. Due to its importance in WSNs, we put special emphasis on the focus of energy efficiency. Furthermore, we consider the requirements and architectural aspects common to all WSN testbeds and present known approaches to realize them in Section III. Aspects concerning the actual experimentation in WSN testbed are dealt with in Section IV. Finally, we provide an overview of current trends in WSN testbed research in Section V and conclude with a brief summary of this work in Section VI.

TABLE V
WSN RESEARCH TOPICS INVESTIGATED IN TESTBEDS AND DERIVED REQUIREMENTS

WSN research topic	Testbed requirement
Energy efficiency	Energy consumption estimation and measurement II-A
Mobile sensor nodes	Mobile nodes and localization infrastructure II-B
Realistic WSN context	Deployment in target environments II-C
Scalability and performance in WSNs	Simulations and hybrid testbeds II-D, large network size II-F
Sensor node design	Development of sensor nodes, monitoring, measurement, debugging II-A, II-D, II-G
Interoperability and platform support	Support for different platforms II-D, II-H, heterogeneity II-E
Application and protocol design	Tool support for development, debugging and deployment II-D, II-I

II. WSN RESEARCH TOPICS AND DERIVED REQUIREMENTS FOR TESTBEDS

In this section, we discuss the different research intentions leading to the development of such a diversity of existing testbeds for wireless sensor networks. Table V presents each research topic and the required abilities and provisions for a specialized testbed in the corresponding field of study. Each requirement references the subsection where we briefly discuss the major challenges and possible solutions for the development of tailored testbeds. In addition, the subsections present existing testbeds and their suitable approaches to handle the requirements.

A. Energy Efficiency Evaluation

Outstanding characteristics of sensor networks are often described with autonomous long term operation, unobtrusive monitoring and large node numbers. These qualities dominate the WSN research in terms of resource constraints, node sizes and costs. The sensor nodes are desired to be small size and low cost. At the same time, sensor networks have to provide adequate lifetime and sufficient performance. Therefore, WSN applications have to be optimized for resource constraints given by the sensor node hardware.

The most important resource constraint in WSNs is the available energy for node operation. Many other resource constraints can be overcome by spending more energy and thus accepting less energy efficiency, e.g., in lack of hardware support doing software public key cryptography. Due to this fact, WSN applications have to be particularly optimized for energy efficiency.

Energy efficiency optimization of sensor networks can be done on various components:

a) *Energy storage devices*: Energy storage devices of sensor nodes are still dominating the node sizes of current WSN platforms. By developing a novel storage or reducing it's size, the energy storage have to become more energy efficient to still provide enough energy for adequate node lifetimes.
b) *Energy harvesting*: For autonomous WSNs, energy harvesters like solar cells are often used to reload exhausted batteries. In doing so, the harvesters have to provide a sufficient rate of yield and thus be efficient.
c) *WSN platforms*: Sensor nodes consist of micro controllers, memory, sensors, actuators etc. All these electronic components are energy consumers and so their efficiency accounts for nodes' overall energy efficiency.
d) *Communication transmitters*: Typically, transmitters for wireless communication in WSNs are energy-hungry. In fact, the radio transmitters are often first choice for optimizing a platform's energy efficiency. Optimization can be focused on energy consumption of electronic components and antennas, but it is also essential to use the transmitters efficiently. For example they can be turned off while not communicating.
e) *The application layer*: This sensor network layer is the most flexible component. Even though WSN deployments are always application specific and optimized for a corresponding scenario, the WSN hardware is rarely a special design. Usually, a generic and common platform with some scenario specific adaptations is used as a basis for the application written by developers. Typically, the application is the only modifiable component after the decision in favor of a specific platform. Thus, the application layer is the most scenario specific component. Furthermore, it can often be exchanged or at least adjusted during the WSN lifetime. Naturally, energy efficiency of a WSN highly depends on the application layer. The application layer consists of multiple components, e.g., OS, network layers, device drivers and application logic. Each of these component's energy efficiency and interaction has significant impact on the overall efficiency of a WSN.

Several different methodologies exist for energy efficiency evaluation in WSN research.

1) Theoretical Evaluation: In WSN related publications a prevalent method is theoretical evaluation. It's always based on a theoretical model of the system being evaluated. Systems represented by a model can be WSN platforms, mechanisms, protocols, applications, etc. The advantage of theoretical evaluation is the ability to evaluate unimplemented systems. Doing so, deliver at least a first proof of concept. Furthermore, it helps to evolve an expectation on behavior and characteristics of the system being developed. Validity of theoretical evaluation results highly depends on model quality and detail level. Highly detailed models are believed to deliver results closer to reality, but are commonly very complex and thus hard to understand and handle. Theoretical energy models are rarely complex and yield coarse grained results, which are mostly sufficient for approximative life time estimations. In fact, often just counters for send/received messages are used as metric for energy efficiency evaluation. However, in many cases due to MAC protocol optimizations sending several messages in burst does not require significant more energy than sending just one message. This conflicts with the prevalent assumption that the less messages sent/received, the less energy is consumed for communication purposes, thus prolonging node life time. Due to this fact, good models have to be built from real measurements and must be validated by extensive evaluations.

2) Simulative Evaluation: Simulation is another approach for energy efficiency evaluation in WSNs. By taking real application processes into account, simulation does a better job providing realistic results than pure theoretical evaluations. The application must be implemented to run the resulting code in simulator. There are two types of simulators used for evaluation of WSNs.

a) General purpose simulators: Most popular simulators, e.g. OMNeT++ [93] are general purpose simulation environments and thus are platform independent. This simulators abstract from a concrete WSN platform. They require the user to implement the application on an API provided by the simulator, strongly binding the implementation to the utilized simulation environment. Due to this fact, such implementations cannot be applied for deployment on a WSN platform. Given this characteristics the results of energy efficiency evaluations can only be considered as coarse estimations, because HW components like radio chips have significant impact on energy consumption of sensor nodes.

b) WSN specific simulators: Some simulation environments support WSN specific HW, e.g. radio chips or sensors. Furthermore they take wireless channel characteristics into account, which is important for energy efficiency evaluation. There are also simulators that support native platform code [94] or native binary images [95]. The former ones cross compile the WSN code for a platform where the simulation is supposed to run. The latter emulate the HW of WSN platforms and thus are even able to interpret the binary code compiled for a supported WSN platform. Several WSN specific simulators or emulators support energy consumption evaluation, e.g., Avrora [95] or PowerTOSSIM [96], [97].

Energy consumption estimation in simulators is mostly based on energy models of simulated platforms, so the significance of simulation results depends on the accuracy of these models. Often, the models used are not fine grained enough and neglect for example state transitions of a node for simplification reasons. In addition, the energy models in simulators are usually parametrized with energy consumption information provided by manufacturers in platform hardware specifications. Due to hardware tolerances, the energy consumption of real hardware is subject to strong variations. Furthermore, environmental factors like temperature cause measurable fluctuation of current draw. For these reasons, simulation results may significantly differ from real world experiment results. In [98], the authors identified such issues in the WSN simulation tool Avrora by comparing simulation results to measurements in SANDbed [48]–[51]. They adjusted the energy model in Avrora to reflect the current draw in their real world experiment setup and introduced state transitions. Moreover, they implemented a model for hardware tolerances in the simulation tool to get results comparable to real world measurements.

3) Real world experiments: The most precise and realistic approach to evaluate energy efficiency in WSNs are real world experiments with energy measurements. Unfortunately, this is also the most challenging approach. Due to the distributed fashion of sensor network applications, it is important to evaluate the distribution of energy consumption in the whole network. Measurements on just one single node are insufficient. Depending on actual topology and node role in the network, energy consumption varies significantly from node to node. This means energy consumption of every single node have to be investigated.

Evaluation of energy consumption of any electrical device requires in-depth understanding of physical effects related to electric potential energy. The energy E needed for operation of an electrical device depends on the voltage U of the power source, the current draw I and the time of measurement t. While U is quite constant, I varies much over time, depending on the actual node operation mode. This relation can be described mathematically as: $E = \int_{t_0}^{t_1} u(t) \cdot i(t) \cdot \mathrm{d}t$. Measuring energy consumption of an electrical device generally means measuring $u(t)$, $i(t)$ over the evaluation period and then determining E. While it is possible to measure $u(t)$ and t accurately, there is no direct way to measure the current draw $i(t)$. Consequently current draw is measured indirectly, by determining its effects on other variables more suited for direct measurements. Since this is the case for voltage and time, it is convenient to use them to determine the current draw and thus the energy E.

Several approaches for energy measurements exist. In-depth description and discussion of known and appropriate measurement methods in wireless networks is given in [99]. In a nutshell, there are four measurement methods known in literature. First, there is the *shunt resistor* approach, where the voltage drop on a shunt resistor is measured by an analogue to digital converter to determine the current drain. This approach suffers from imprecision when measuring very low currents at the bottom of the measurement range. To improve the precision for very low currents, the *Voltage to Frequency Conversion* method was proposed by Jiang *et al.* [80] as SPOT, a scalable power observation tool. In this approach, the voltage drop on the shunt is converted to a signal with voltage dependent frequency. This frequency can be measured very precisely; So, the current drain can be determined. Next, there is the *current probe* [79] approach, where the measurement instrument and the circuitry of the device under test are galvanically isolated. This approach aims to be side-effect free. A very popular measurement method is using so called *Coulomb counters*, e.g., described as *Energy Bucket* in [100]. Doing so, the energy flow is measured with help of charging and discharging capacitors, which in their turn supply the measured device with energy. Last but not least, it is often possible to use electrical components that are already integrated into the devices whose energy consumption is supposed to be measured. Since Coulomb counters are often a part of charging controllers on devices, it is appropriate to use them for energy measurements.

Considering the qualities of the described measurement methods there emerges a stress field. Designing special energy measurement hardware demands for trade-offs between the dimensions of this stress field, as depicted in Fig. 1. Whereas it is out of scope of this survey to investigate mathematical interrelations between the qualities, we briefly illuminate the interdependency of the qualities.

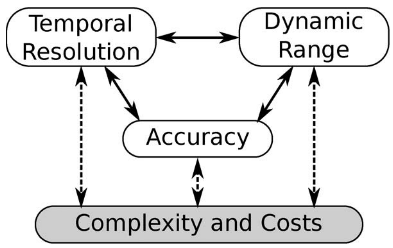

Fig. 1. Stress field of qualities in energy measurement methods.

Methods with high temporal resolution like the shunt resistor approach suffer from low accuracy for very low currents, thus featuring a smaller dynamic range than Coulomb counters. On the other hand, Coulomb counters are characterized by a relatively low temporal resolution, compared to shunt resistors, but gain in accuracy capturing very low currents. The measurement approach with Voltage to Frequency Conversion demonstrates a trade-off between accuracy, temporal resolution and dynamic range. Last but not least, cost and complexity make some methods unsuitable for integration into embedded devices. Furthermore, in many cases high temporal resolution of measurements is not needed, so simplifying the measurement setup may save time and costs.

Consequently, choosing the right measurement method highly depends on the requirements and goals for energy measurements. Identifying these goals is essential for proper and adequate measurement results. As a matter of course, the knowledge about specific characteristics of available methods is very important. The criteria to consider are whether a measurement technique is prone to disturbances, whether it offers high temporal resolution and whether it can cope with high dynamics. Further, it is judged by the question whether they can be realized by electrical components that already are part of the devices under test. Finally, the effort needed to instrument a device and calibrate the setup has to be considered.

Not only the strengths and weaknesses of the different measurement techniques, but also the question when and where the measurements should be taken, impact the choice for a measurement approach. If energy measurements are supposed to be performed only during the development and testing of a device or application within a laboratory setup, the shunt resistor approach is the best choice [101]. The same applies to testbeds of wireless sensor networks that are used for application profiling, e.g., SANDbed [48]–[51]. Profiling applications and protocols profits from high temporal resolution as a detailed insight into the energy characteristics of the software under test is given. However, if the measurements are taken at run-time, for example within energy-aware wireless sensor networks, the measurement tools must be integrated into the devices. This speaks in favor of Coulomb counters. Many embedded devices already contain the electrical components needed for Coulomb counters. Measurements without the laborious instrumenting of dozens or hundreds of devices is a compelling feature of this technique. Using integrated switching regulators as described by Dutta *et al.* [23] is perfectly suitable for in-network energy measurements and energy-aware applications. Fonseca *et al.* [102] describe such a design of a network-wide time and energy profiler, called *Quanto*, which is based on iCount [23].

Energy measurement tools were implemented in several WSN testbeds. MoteLab [3] is a general purpose sensor network testbed, where a single node was instrumented with a network-connected digital multimeter, providing a sampling rate of 250 Hz. In Mobile Emulab [8] a prototype of a low cost ($\approx$35$) energy measurement device was build and tested on one node, with the plan to equip all testbed nodes in the future. TWIST (TKN Wireless Indoor Sensor network Testbed) [9], [10] allows protocol evaluation in terms of energy consumption. For this purpose, the nodes are equipped with power measurement circuitry [11] and the measurement results are available to the node itself at any time. The shunt resistor based circuit suffers from inaccuracy measuring very low currents and the measured node have to process the results. In WiLab [40], an environment emulation device is used to control and measure nodes power consumption with a sample rate of 10 kHz. In FlockLab [28] dedicated monitoring hardware implements energy measurement support with jumper selectable sampling rate of 28 kHz for better SNR or 56 kHz in high-speed mode and 24 bit resolution. Based on calibration parameters determined for every measurement device, the test management server corrects the results before delivering them to the user. The authors of [44] also target to implement hardware support for energy measurements. In PowerBench [38], the sensor node programmer boards attached to each testbed node were extended with a power monitoring circuit to provide energy measurements. The HINT [55] testbed incorporates special monitoring boards, that passively probe chip-level signals of the micro controller, transceiver and other components. HINT test units are able to switch node power on and off and measure the current draw of the sensor nodes as well. In [50], [51] a *Sensor Network Management Device* (SNMD) is proposed as dedicated monitoring and simulation hardware for WSN testbeds. It provides software configurable energy sampling rates up to 500 kHz, a resolution of 16 bit and evaluated measurement error $< 1\%$. The SNMD is integrated in SANDbed [49], which currently consists of 28 nodes.

Many dedicated energy monitoring solutions provide more extended features, e.g., triggering the measurement start and stop from the measured device. This allows selective measuring of certain sections in the program flow. Further extended features for evaluating energy efficiency are programmable power supplies and debugging interfaces. Such features are provided by the devices in FlockLab and in SANDbed.

B. Mobility

Mobility of users and their devices is one of the main reasons why wireless communication networks are needed. A sensor network is a specialized type of wireless communication network, that usually has a very small number of distinct users. Nevertheless, mobility is a major concern because WSNs often rely on movable nodes quickly and easily deployed at new locations. Furthermore, it can be necessary to attach nodes to mobile hosts to cover the locations of moving phenomena and information sources. For special application setups, it is imaginable that the nodes are realized as self-moving platforms. Exemplary applications for self-moving platforms are oil spill surveys or coverage area optimization of a monitored woodland. WSN testbeds have focused on mobility as a research topic in several manners. Most of testbeds using mobile nodes are currently very restricted by the size of their coverage area. Indoor environments are usually preferred for such testbed deployments.

Mobile sinks and nodes can be realized in combination with mobile host devices like gadgets or robots. Popular approaches use human experimenters carrying devices like in Sensei [45]–[47], model railroads like in SensLAB [54], toy or domestic robots like [e-puck, romba] [TrueMobile/Mobile Emulab] [7], [8] and MOTEL [56] or even flying quad copters like in wsCoop [30]. Robots like in MiNT [67], [68] are used in testbeds for research of wireless networks with a broader scope, not only limited on WSNs. Humans carrying nodes like notebooks or mobile phones and even public transportation were used in ORBIT [64], [65] to carry some nodes.

An additional challenge designing a testbed for mobile nodes is the reproducibility of locations and trajectories of the mobile nodes through the network between different experiments. In Sensei [45]–[47] the authors observed high variance in experiment results because it was impossible for humans to carry mobile nodes each time exactly in the same pattern. For tracking indoor localization techniques were used and GPS in outdoor experiments. Mobile Emulab [7], [8] uses markings and cameras mounted on the ceiling to monitor and guide mobile robot nodes to reach predefined locations as precisely as possible. To set up experiments with mobile nodes, usually a scripted configuration is used to conduct mobility experiments.

Another approach to handle mobility in WSN experiments is the construction of a completely or partly movable testbed infrastructure. Movable infrastructure allows to cover experiments in places where static testbeds cannot be installed due to reasons of accessibility, security or effort. These testbeds are called *nomadic* according to their ability to move the testbed to an interesting spot for an experiment. Usually, nomadic testbed designs use wireless backbones to support experiments in outdoor environments with low effort.

C. Realistic and Target Environments

Sensor networks are dedicated systems intended for monitoring of physical or environmental conditions, e.g temperature or movement of objects. Less usual network components are actuators, e.g., lamps, for environment manipulation.

In many cases, the behavior of a sensor network is directly affected by the sensor values. The sensor values are influenced by the environmental conditions of the network. For example, in fire detection scenarios, each node monitors ambient temperature and sends an alarm message if this temperature exceeds a predefined threshold. When temperature is below the threshold, no fire alarm is triggered. The number of messages sent in a fire detecting sensor network depends on the temperature in the monitored area. In such cases, the environment has significant impact on a sensor network application, affecting its behavior characteristics and its power consumption. The evaluation of a fire detection application in a testbed can lead to unrealistic and thus meaningless results if the testbed environment differs from the network target environment. Some existing testbeds are designed with the intention to meet this challenge and provide realistic evaluation results, in spite of differing experiment and target environments.

The testbeds can be categorized into two groups depending on the quality of their experimentation environment and the data able to extract.

1) Simulated and Emulated Environments: The first group consists of all testbeds simulating or emulating the environment of the sensor network as it would prevail in a real deployment.

The Kansei [14], [15] testbed can not only be used to measure real world data, it also offers three additional methods to inject artificial sensor data into running experiments. Sensor nodes can either use prerecorded sensor traces from preceding experiments, use synthetic data generated from models based on the sensor physics, or data from probabilistic models exploiting correlations of sensor nodes in the neighborhood with Gaussian Markov Fields.

SANDbed [48]–[51] also provides the option to feed real sensor nodes with simulated or previously measured sensor readings. SANDbed's management model allows to extend and adjust environment models to the experimenters need. The environment model can be adjusted by management PCs in the testbed backbone to generate sensor readings that can be feed via IO-pins or a serial interface to the nodes. Predefined simulated environments for sensor reading generation can be reused in different experiments. Nevertheless, WSN applications must support feeding sensor readings into the application explicitly.

EasiTest [52], [53] focuses on the emulation of the radio communication environment. Therefore, it utilizes software defined radios to record and playback the radio noise in testbed experiments.

A similar approach is researched in ORBIT [64], [65] which offers a RF interference generator and traffic generator to emulate different wireless media conditions.

WINTeR [36] is an industrial testbed for the deployment of sensor nodes on offshore oil platforms. To rebuild the target environment as closely as possible, an oil platform mockup was used as experimentation environment. The harsh radio environment is emulated with an electromagnetic interference generator and the sensor nodes are fed data generated by a dedicated server.

For environment emulation in SeNeTs [2], the testbed supports a mapping of real nodes to virtual node positions in the emulated environment. Based on the virtual position of the nodes, the testbed can filter the sensor data from predefined sets for air pressure and temperature. SeNeTs is not only able to supply the virtual nodes with individual sensor values, but also with incoming transmissions of their virtual neighborhood.

Additional hardware environment emulators are connected to the nodes in WiLab [40]. The hardware is able to simulate node failures, battery depletion, energy harvesting and offer various analog and digital interfaces to supply the nodes with artificial sensor values.

2) Real Environments: The second group consist of testbeds that are already deployed in their target environment and use real world sensor data for experiments.

The testbed discussed in [43] is deployed partly indoors and outdoors to analyze the different environment effects on high speed data sensing and monitoring with a sampling frequency up to 18.5 kHz. Trio [24] is another testbed deployed in a real outdoor environment with solar powered nodes to complement the existing Berkeley indoor deployments.

In [31], [32] the authors deployed Casino Lab in an engineering building. A second testbed is deployed in the real target environment, an underground mineral mine. Especially multipath and signal degradation was observed in the underground environment.

To gather data like weather conditions, air and water pollutants and other biochemical measurement values, CitySense [76] has been deployed in urban streetlights. The continuously powered nodes allow to gather high quality sensor values but also limit the testbed in experiments concerning sensor lifetime and low power applications and communication.

Sensei [45]–[47] is designed to carry the whole *nomadic* testbed into a target environment to work with realistic sensor values. We will discuss this ability in more details in Section II-B and V.

D. Simulator Integration

The size of a testbed deployment is usually limited by the amount of money available for sensor node hardware and infrastructure. Therefore, many projects try to expand the network size by adding virtual, emulated or simulated node entities.

Working with a combination of real and simulated hardware leads to distinct challenges. Especially, it has to be considered, how to handle communication between nodes crossing the real and virtual domains. Problematic research topics in this context are e.g., latency, energy consumption, media access control, transmission reliability, but also differences in sensor readings between spacial related real and virtual nodes.

The Kansei [14], [15] testbed offers two simulation interfaces. On one hand, it is able to work with simulated sensor readings on real hardware as described in Section II-C. On the other hand, Kansei supports hybrid simulation with virtual nodes using real sensor readings and channel characteristics with their simulation tool *MULE* [103]. Similar to Kansei, CasinoLab [31], [32] is primarily a testbed of deployed real Tmote Sky nodes, but also offers a TOSSIM Live extension to connect with a simulation.

SeNeTs [2] realizes a distributed simulation environment on real hardware. Therefore it uses a middleware adaptation layer for hardware abstraction to allow test applications to run either on simulators or real nodes. The adaption layer includes components for logging, debugging, controlling and encapsulated environment simulations.

The reasons for simulation and testbed co-usage include easier and faster protocol development cycles utilizing a combination of evaluation and verification tools available. Usually, the improvement of simulations with results gathered from real or testbed deployments is one of the intended targets in mind. Accordingly, a well-designed hybrid simulation and testbed environment can e.g., support researches by using the same application code in the entire development process. Such hybrid experimentation environments rather often emulate the sensor node hardware of individual nodes instead of simulating an entire sensor network. Nevertheless, the wireless channel between virtual and real node entities still has to be simulated to research communication behavior. More or less automatically, hybrid experimentation environments allow to compare the evaluation results from simulation, emulation and test on real hardware by using compatible data formats and integrated evaluation tools.

The emulation of sensor nodes is used by EmStar [63], which combines a sensor network emulator with a real Berkeley Motes deployment. EmStar can work in different modes, from pure simulations over hybrid modes using a channel simulator, up to pure hardware deployments. Even a data replay mode can be used to playback recorded data traces for repeatability of experiments or simulation of specific sensor modules.

In TOSHILT [5] a *Hardware-In-the-Loop Emulator* is connected with a number of embedded processors to analyze design flaws and bugs in applications with hardware specific code. The emulator simulates the environment of a test scenario and also emulates sensors and A/D converters of the sensor network to generate sensor readings for the processors in-the-loop.

TrueMobile/Mobile Emulab [7], [8] is based on the experimentation framework for network testbeds named *Emulab*. It is able to emulate wireless network links and supports distributed events during experiment runs. The authors extended the existing framework to WSN testbeds. They added tools to control and localize mobile robots within the sensor network with the intention to improve their simulation abilities concerning mobile nodes.

E. Heterogeneity

Several testbed approaches focus on either heterogeneous sensor networks or heterogeneous wireless networks in general, integrating WSN components in a larger network testbed. Heterogeneity in testbeds is always useful to enable an experimenter to choose the desired components and adapt the environment for the experiment.

The term *heterogeneity* is used with different intentions in the testbed research context.

1) Sensor Heterogeneity: Sensor heterogeneity is given in a testbed deployment if individual nodes are equipped with different types of sensors. For example, in Indriya [57] are deployed heterogeneous nodes. One type of nodes is equipped with infrared sensors only. Another type of deployed nodes is also equipped with visual light and acoustic sensors and a third type adds temperature and acceleration sensors and magnetometers into the testbed. Sensor heterogeneity allows to detect a large variety of phenomena in an experimentation environment while covering a large experimentation area for lower cost than equipping every node with the entire sensor equipment. Some experiments can also benefit from the reduced amount of sensor data which can be used for efficient distributed processing.

2) Platform Heterogeneity: If different types of sensor node platforms like MICAz and TelosB can be used together in a single testbed, platform heterogeneity is realized. WSNTB [34] for example, mixes self developed *Octopus I* and *Octopus II* node based on Atmel AVR and Texas Instruments MSP430 microcontrollers to incorporate platform heterogeneity in their testbed. Both node types use ZigBee compliant radios reducing the heterogeneity on the microcontroller and the used operating system, in this case TinyOS and a self developed OS named *LOS*.

Experiments can benefit by platform heterogeneity support if the power consumption of a new communication protocol is heavily dependent on the underlying hardware and operation system. Therefore, a very efficient communication mechanism on a specific platform might be unfeasible on different hardware even if not predicted on prior simulations. In that case, creating a testbed that supports different node hardware platforms or operating systems, enables the users to draw more general results from experiments.

SANDbed [48]–[51] leaves the software side of platform-interoperability to the experimenter. It supports platform heterogeneity with reconfigurable *Sensor Node Management Devices* (SNMDs). SNMDs integrate all kinds of sensor nodes into the testbed that are programmed via USB or are based on Atmel AVR microcontrollers. MICAz, IRIS and SunSPOT nodes have already been successfully integrated with this self developed hardware.

3) Device and Protocol Heterogeneity: Heterogeneous devices and protocols are often combined in testbeds to support experiments with WSN integration in larger ad-hoc or mesh networks. In these networks not only WSN nodes are deployed, but also less resource constrained nodes with more computation power. Capable devices in the network can also feature WLAN or Ethernet interfaces to access larger LANs or the Internet. An example for such a testbed is NetEye [35] which consists of a network with mixed IEEE 802.15.4 and 802.11 nodes. A similar approach is realized with EasiTest [52], [53]. The two deployed node types *EZ271* and *EZ521* provide different network interfaces, but both provide radios for IEEE 802.15.4 and 802.11 compliant wireless access, allowing to develop protocols to exploit multi-channel transmissions.

The DES-Testbed [72]–[75] integrates the WSN nodes into a wireless mesh network by pairing sensor nodes with wireless mesh routers. Additionally, the DES-Testbed can be used to conduct experiments with WPAN, TETRA and GSM devices.

The Kansei [14], [15] testbed is another example for device heterogeneity realized with Extreme Scale Nodes (XSM), Extreme Scale Stargates (XSS), Trio Motes and robotic mobile nodes.

4) Testbed Heterogeneity: Testbed heterogeneity is given in some projects with interconnected testbeds like WISEBED [59], [60], KanseiGenie [62] or WHYNET [58]. A larger testbed is built as a *federation* of different smaller testbeds from multiple institutions, each with its own hardware and network topology. Combining different hardware, infrastructures and communication technologies, they contribute to a larger heterogeneous sensor network. Such testbed federations usually need special software components for testbed abstractions in the sensor network like WISELIB [61]. An adaptive management system for experiment scheduling and user accounting on administration level is mandatory for testbed federations. In Section V we describe special requirements of testbed federations in detail.

F. Size and Scalability

If a WSN application working as expected on a single node, it doesn't mean that the application shows the desired behavior in a network with hundreds or thousands of nodes. Simulations can be used as a first indication to find out if an application is scalable. However, simulations often simplify the media access or channel model. Unlike real sensor networks, simulations process concurrently triggered events in a sequential and quasi-parallel order [2]. While these simulation properties might be negligible for small networks, they can have major impact on results for networks with thousands of nodes. For example, researching congestion in dense networks and coexistence of WSNs with existing WLAN networks or other radio devices operating in the same frequency range can lead to unexpected behaviors in large networks.

Testbeds focusing on scalability of WSN networks try to validate the widespread simulation studies. They usually apply a hybrid approach between a classical testbed with a few nodes and outdoor field tests with many nodes and high hardware cost. Realizing large testbed network sizes requires an elaborate system for management and maintenance, a pre-planned network topology and large deployment space. That is why warehouse or outdoor deployments with solar panels are preferred over usual battery-driven setups in office buildings.

SenseNeT [26] is the smallest testbed researching scalability of WSNs with only eight MICAz nodes. It focuses on accessibility mechanisms for larger testbeds, i.e., on multi-user support, parallelism of experiments and over-the-air programming.

ExScal [4] has been the largest known testbed studying wireless sensor networks with more than 1000 nodes deployed in a forest for aerial intruder detection. The nodes were battery powered because the experiments took only two weeks until the deployment was taken down.

A similar approach to ExScale was taken with the 557 nodes in Trio [24] during its 4 months of operation. Trio was developed to evaluate multi-target tracking algorithms in an outdoor environment with solar powered Trio nodes. The SNMS [104] management system, already used in ExScal, was improved to track the dissemination of program images through the network.

To support small- and large-scale setups, the MoteMaster [41] testbed developed a *deployment site* concept to build groups of nodes and manage the hardware. On each deployment site, a mote manager is responsible for querying a central dispatcher and database for new tasks. During an experiment, a deployment site can work independently and gather results locally. This distributed approach simplifies multi-user resource access and parallel experiment execution.

Kansei [14], [15] was developed to analyze sensing at scale with heterogeneous hardware. It provides 210 XSM nodes, 150 Tmote Sky, 50 Trio motes and 5 mobile robots. The sensor nodes topology is build out of array groups. While the largest part of the testbed is deployed in a warehouse, some of the arrays are equipped with solar panels for mobile or outdoor experiments.

With Indriya [57], especially the scalability behavior of geographic routing protocols were researched. For this reason 127 TelosB nodes have been deployed, organized in clusters with Mac minis as cluster-heads and a central server running a modified version of Motelab for management and access to the testbed.

G. Development of Sensor Nodes

In some research works, WSN testbeds and testing environments are used to develop new sensor network systems and node hardware platforms. These testbeds are specialized to evaluate system or hardware design decisions. They offer a dedicated set of testing, measurement and monitoring tools to support the design and evaluation process.

The Hyperion [81] testbed is focused on single sensor node hardware rather than a sensor network. It uses a FPGA to build and easily replace sensor node prototype CPUs. The CPU implementation is analyzed in terms of energy consumption on an instruction-cycle accurate basis.

TOSHILT [5] has been developed as a dedicated tool for hardware-in-the-loop (HIL) testing. In this testing methodology the sensor network hardware and a designated software is tested as a system. In difference to field tests, a HIL test uses a smaller network with predefined use-cases making a setup change much simpler. Therefore, HIL tests are faster and much cheaper in a controlled testbed environment.

Evaluation of timing, concurrency behavior and hardware-specific optimization in computation and communication is still possible, because the same hardware is used as in a real deployment. TOSHILT is not realized in hardware. Instead, the environment is emulated with a middleware framework running on the sensor nodes. A PC controlled emulator is used to generate scenario data and sensor readings for virtual sensors in the emulated environment.

H. Platform Independence

Simulation results are usually independent of any sensor node platform specific hardware issues. Unlike simulations, deploying applications in testbeds or real sensor networks demands hardware specific adjustments and optimizations. To make statements about protocol or algorithm characteristics for WSNs in general, comparing different sensor node platforms is necessary.

To draw generalized conclusions from experiments, some testbeds focus on platform independence as a design goal. In heterogeneous wireless sensor, mesh and ad-hoc networks, supporting platform independence can also simplify experimentation. Supporting multiple node platforms leads to a higher level of abstraction in the testbed architecture and toolchain. Dedicated hardware functionality of a specific platform might not be supported in a platform generic sensor testbed.

Sensor nodes in SANDbed [48]–[51] are attached to a dedicated SNMD. This piece of hardware is used to provide interfaces and functionality for experiments like node power connection or serial IO interfaces independent of the used sensor node platform or operating system. To connect nodes to Sensor Network Management Devices, a plug adapter is used. However, for some platforms simple adjustments of the adapter might be necessary.

Flocklab and its predecessor JAWS-DSN [27], [29] providing platform independence by connecting sensor nodes to additional testbed hardware devices. While JAWS-DSN connects each sensor node to a second sensor node which is used only for experiment control and result collection, Flocklab enhanced the concept and connects the sensor node to small embedded observer PCs with an Ethernet and WLAN backbone. This allows Flocklab to separate both networks physically and to minimize the intrusive effect of control and management network traffic on the experiment. Like SANDbed, Flocklab needs adapter boards to interconnect the sensor nodes with the observer nodes.

MOTEL [56] realizes platform independence with a modular software framework for run-time management called *FLEXOR*. FLEXOR defines exchangeable software modules based on a set of interfaces and rules how to connect them. Running experiments means preloading the entire FLEXOR framework on the nodes while a *ModuleManager* component decides which modules to activate. However, using nodes with operating systems other than TinyOS, requires porting multiple FLEXOR modules.

SeNeTs [2] uses a similar approach for platform independence. To provide functionality for logging, debugging and controlling during experiments, SeNeTs encapsulates the applications running on the nodes and adds its own middleware components and platform specific sensor drivers. As a result, the application running on the nodes can run independently of an observing experiment control system. However, the application has to share the node hardware with the SeNeTs adaption. Due to this middleware, the SeNeTs applications can also be used for simulation.

Many testbeds can be used with various types of sensor nodes as long as the nodes are compatible with TinyOS. In [44], such a testbed is presented based on the TinyOS toolchain with hardware and software support for the development of energy-efficient applications. The testbed uses features of the TinyOS toolchain to extract the application state information and associate it with the measured energy consumption. Nodes with other operating systems cannot be supported with such an approach.

I. Applications and Protocols

The capabilities of testbeds to debug and analyze network protocols in details, are very promising for communication research in wireless sensor networks. WLAN testbeds like Roofnet [66] already were used successfully to evaluate link characteristics, network metrics and protocol design. Research on routing or MAC layer protocols is currently the most frequent use case for WSN testbeds. WSNs are generally optimized for a special use case, so that the applications with the used algorithms and protocols are attuned to each other. That is why some researchers like [43] also

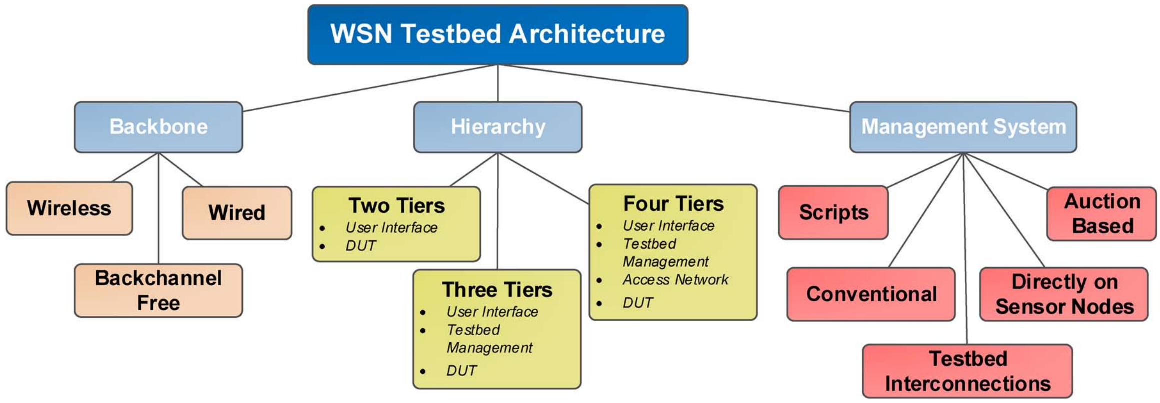

Fig. 2. Taxonomy of architecture components for sensor network testbeds.

evaluate signal processing algorithms with similar testbed setups.

A common requirement for testbeds, designed for protocol and algorithm development, is an expedient toolkit supporting software and network analysis. For debugging purposes a developer might be interested in tools to evaluate output log-files of sensor nodes, tools capable of triggering inputs on the sensor nodes during experiment runs, tools for application profiling or tools to monitor and visualize the network and application state.

A typical example for a testbed providing a general toolkit for development of rapid application, protocol and system prototypes is Motescope [22]. Motescope and its successors sMote, and Omega [25] have been gradually extended and upgraded with recent nodes to be able to test larger applications. Utilizing the TinyOS toolchain, including the popular *Serial Forwarder*, the testbed can be used as a bunch of nodes, attached serially to a local PC.

S-WiNeTest [18] and SensoNet [19] base upon the same architecture. While SensoNet uses the testbed for general protocol development and evaluation, S-WiNeTest focuses on networking functionality. S-WiNeTest is a very small testbed, built out of 8 MICA nodes attached to PCs. It has only limited possibilities to deploy new applications, but allows to change the routing protocol used on the sensor nodes without reprogramming the nodes. The testbed also offers tools for packet injection into the sensor network using a preprogrammed injector node and adaption of the sensing rate.

In [20], the authors present a testbed to benchmark the performance of network management protocols in wireless sensor networks. Unlike other approaches, the testbed itself belongs to the management system and doesn't build an additional transparent layer to evaluate applications on the nodes. Some MICA2 nodes are used in conjunction with an FPGA board as a central base station. The base station is connected with a PC, serving as a client with user interface.

SANDbed [48]–[51] is a testbed designed for the development and optimization of energy efficient protocols and applications. Therefore, it offers distributed energy measurement capabilities to analyze power consumption simultaneously on every sensor node with a high temporal resolution. Especially MAC-layer protocol mechanisms have been accurately analyzed with SANDbed.

Similar to SANDbed, TWIST [9]–[11] has been used to develop MAC-layer protocols. Besides tools for protocol debugging and robustness evaluations, TWIST also focuses on power consumption measurements. In contrast to distributed measurements of protocol mechanisms, the energy measurement approach in TWIST targets on a general network lifetime estimation.

III. WSN Testbed Architectures

Besides meeting the necessary requirements for the research focus, designing a testbed needs some architectural consideration. Some infrastructure components are widespread in current testbeds. These components are used to realize a structured control of the node hardware, a systematic approach to deploy experiments and collect the results for evaluation. In this section, we discuss the typical infrastructure components (Fig. 2) of current architectures in details.

A. Backbone

The choice of backbone is a major design decision when building a testbed. Basically, the commonly used backbones can be divided into wired and wireless backbones.

Most of the discussed testbeds use wired backbones to connect their sensor nodes, testbed-specific hardware and central components like databases or control servers.

1) Wired Backbones: Ethernet and USB infrastructures, already exist in most of the research labs. These backbones are the typical basis for many current sensor network testbeds. While USB is chosen for its plug-and-play and power supply capabilities, Ethernet provides high transmission rates and a huge variety of existing tools. The main drawback of a wired infrastructure is a comparatively static testbed deployment. Mobile nodes are difficult to include in existing wired networks resulting in additional infrastructure components, effort and complexity. Wired testbed infrastructures do not scale well and are expensive compared to wireless infrastructures [26]. Adding several sensor nodes or changing the network topology for experiment needs is far less laborious in wireless testbeds. In outdoor scenarios wired infrastructure has drawbacks, due to the laborious setup. The high infrastructure deployment costs in relation to the deployment costs of the actual sensor network are also a reason for wireless backbones in outdoor scenarios.

2) Wireless Testbed Backbones: The IEEE 802.11 WLAN variants are most popular for wireless backbones in testbed architectures. Individual sensor nodes are connected to gateways via IEEE 802.15.4 or wired connections. The authors in [2] propose a wireless backbone using ultrasound, and in [29] Bluetooth is used.

Generally, testbeds using a wireless backbone emphasize their improved topological flexibility and support of mobile nodes. These advantages gained by decreasing robustness and an increasing complexity for programming and managing the sensor nodes. Programming nodes can cause situations with unresponsive nodes when the programmed software is erroneous or has not been received completely. In difference to wired testbeds, wireless testbeds often use the same communication channel to sensor nodes during programming and testing.

To avoid a node or an entire sensor network being completely unresponsive, either a second independent communication channel can be used. Alternatively, a *golden image* [24] can be used in combination with *watchdogs* or *grenade timers* [105]. A golden image is a simple, but extensively tested application, providing basic connectivity. It is permanently stored on a sensor node in a non-accessible part of the memory to prevent overwriting by programming. While a watchdog timer is used to prevent locked-up nodes, a grenade timer can be used to prevent unresponsive programs allocating all available resources without locking-up. When such a timer fires, the currently running application will be overwritten by the preinstalled golden image.

Providing an energy supply for sensor nodes and additional hardware is far more difficult with wireless backbones. Powering sensor nodes with batteries or solar cells is common for such testbeds but is also leading to limited control over the power source for experimenters. Changing batteries in large deployments can also be a serious maintenance task for testbed administrators.

Another issue of wireless backbones is interfering backbone traffic and application traffic in the sensor network under test. To prevent such an issue either traffic on the backbone and in the network under test must be timely decoupled as done in [26] or the sensor nodes can be instrumented for management and measurement tasks as in [13], [18], [24]. Timely decoupling has drawbacks for the logging, debugging and management capabilities during a running experiment. Contrariwise, instrumentation of sensor nodes for management and measurement tasks can have a major impact on the experiment results.

3) Avoiding Any Backbones: Simple testbed designs without any backbones are also possible to build. In [56], the authors decided to build a *backchannel free* testbed, meaning during experiments no controlling instances can gather any result. In consequence, the sensor nodes have to be instrumented for logging debugging and programming. The network under test also must be instrumented for code exchange and other management tasks.

A general consideration when choosing a backbone for sensor testbeds should be the amount of data to deal with. Especially, the size of the deployed network and the amount

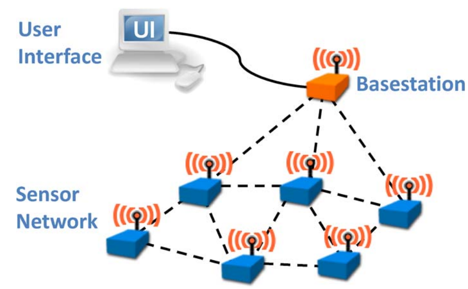

Fig. 3. Example topology of a testbed with two tiers.

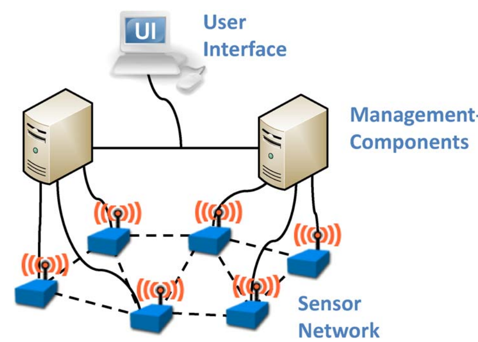

Fig. 4. Example topology of a three-tier testbed with management capabilities.

of data for measurements, management, logging, debugging and experimentation results has to be taken into account. An example of throughput problems on the backbone can be found in the discontinued JAWS-DSN [29]. Latency could also be a problem in testbed backbones for example if central instances try to trigger time-critical events during an experiment. Nevertheless, latency seems to be no big issue in existing testbed literature.

B. Hierarchy

Comparing different testbed architectures, we can identify several common infrastructure tiers. Employing this tiered structure, we categorize existing architectures.

1) Two Tiers: Every testbed relies at least on two tiers: The topmost tier provides an interface for testbed users, usually the experimenters, and optionally an interface for administrators (Fig. 3). Therefore, most of the testbeds either realize an API or some sort of specialized client with a graphical interface or access via web browser. In the bottom tier, the *devices under test* (DUTs) are located. In a WSN testbed, the DUTs consist at least of some sensor nodes with homogenous or heterogeneous hardware. Additionally, in some testbeds like the DES-Testbed [69]–[75], DUTs can also be wireless mesh nodes or mobile gadgets like smartphones or small robots [45]–[47].

2) Three Tiers: In three-tier testbeds (Fig. 4), the experiment and management networks are separated at least logically. Optionally, they can also be physically independent of each other. The additional tier is used for management and experiment control tasks [14], [15], simplification of sensor node accessibility [31], [32] or experiment data and performance handling [42].

For these tasks, the testbeds uses various supporting hardware components in the second tier. Supporting hardware are small common PCs in [48]–[51], embedded devices in [27], [40] and wireless access points or gateways in [6], [14], [15], [24], [42]. A three-tier testbed architecture is more complex to realize than a two-tier architecture, due to the additional hardware involved. It offers extended management capabilities and an increase of error robustness in the sensor network tier. Further, the implementation of evaluation data gathering, experiment support and automation on an independent tier can reduce side effects on the DUTs. For example the collection and storage of measured sensor or energy data doesn't need additional storage or computing resources on the sensor nodes for experiment evaluation.

3) Four Tiers: In some testbeds, the DUTs in the bottom tier are directly coupled with experiment supporting hardware. This type of hardware usually consists of self-developed PCBs with microcontrollers for a special testbed functionality. We do not consider this hardware as a distinct tier because of its limited impact on the general architecture. In the following, we call such components *testbed specific hardware* (TSH). In most of the cases, TSH provides optional features supporting experiments on some or all DUTs. For example in WiLab [40] an *Environment Emulator* is attached to TMote Sky sensor nodes as TSH to emulate sensor inputs and different energy harvesting power sources. In TWIST [9]–[11], TSH is used to measure the energy consumption and return it to the nodes to evaluate energy-aware applications. SANDbed [48]–[51] developed the *SNMD* as complex TSH for energy measurements, power source emulation and remote management functionalites like reprogramming DUTs. In WSNTB [34], the *Reset Device* is another example for TSH. The Reset Device is used to reset the sensor nodes for an experiment as demanded by the experimenters. Flocklab [27] presents an different type of TSH by using additional sensor nodes coupled with the DUTs to observe and monitor running experiments. Meanwhile [28] special designed embedded computers are used for monitoring infrastructure in FlockLab. SWARMS [21] doesn't need TSH per DUT, but utilizes *Cluster Controller* devices which host a scripted process called *Node-Mate* for each DUT.

As described in Section III, most of the testbeds use an Ethernet or IEEE 802.11 backbone. Usually, sensor nodes cannot be attached directly to the management backbone but offer external connectors for USB (e.g., TelosB nodes or SunSPOTs) or proprietary connectors for serial interfaces (e.g., MICA-family nodes). Hence, some kind of adapter or supporting hardware is needed to integrate sensor nodes into the testbed infrastructure. Adapters are commonly realized as TSH and are not considered a separate tier.

In this taxonomy, four-tier testbeds add an additional access tier between the DUT tier and the upper tiers (Fig. 5). The access tier is realized with additional hardware for gateways to allow remote access to the DUTs, i.e., from other testbed clusters or the Internet without involving the upper testbed tiers. In testbeds with different deployment sites or federations as described in Section V, remote access to the hardware can be required. In such a case, the remote testbed's services on upper layers (e.g., result gathering and experiment configuration) can be replaced by own services.

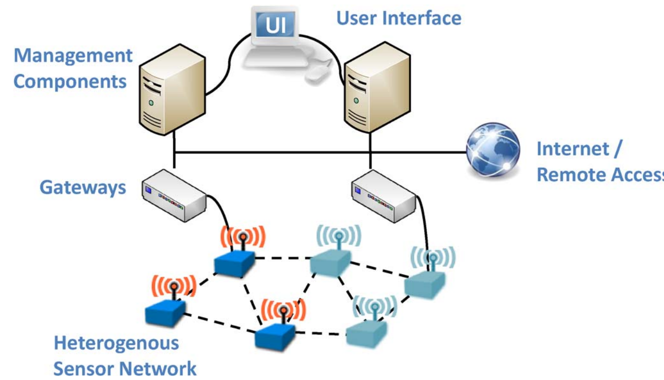

Fig. 5. Example topology of a four-tier testbed with heterogeneous DUTs and Internet gateways.

In some experiments DUTs have to communicate with other services (e.g., NTP for time synchronization) and devices in the Internet. A complex *Internet of Things* scenario where gateways are already intended in real deployments is a typical use-case for four-tier testbeds. Bridges in the access tier can translate between different networks in the DUT and testbed core tiers.

Four-tier architectures are not very common in literature. With S-WineTest [18], the authors present a four-tier testbed with an additional *Core Access Network*. The Core Access Network connects the core testbed network or the Internet on the upper tier with the senor network on the lower tier. In Trio [24], the fourth tier is realized as network of gateways which is used to add support of outdoor deployments and provides Internet connectivity for sensor nodes.

C. Management Systems and Software

A common practice to enable administration of large numbers of nodes in a testbed is to utilize a management system. Such a management system is typically similar to or based on conventional network management systems, but also handles some unique testbed specific tasks.

The management tasks in a testbed comprise capabilities for resource allocation and experiment deployment. The choice of appropriate node subsets for an experiment, the distribution of software images and updates including the programming of sensor nodes and also the control of their operational status are such management tasks. Also, topology control and options to interact with the sensor nodes in real-time are important management qualities during experiment preparation and execution phases. Providing a well-regulated access to sensor nodes is a basic management concept, that is important for debugging, logging and error handling. It is also the essential basis for any multi-user usage of a testbed. Further, scheduling techniques for multiple experiments of different users are required to enable parallel usage or batch processing. Finally, the management must observe and monitor the testbed to detect and handle errors, visualize the network status and experimentation environment dynamically.

The remaining part of this subsection exemplarily reviews selected management approaches as used in the discussed testbeds, whereas in Section IV we discuss how essential management tasks are realized to support the experimentation process.

1) Conventional Network Management: An example for a conventional network management system is the DES-Testbed [69]–[75]. It utilizes *SNMP* as a management system with a focus on network configuration and monitoring. The management console *DES-Portal* provides a management interface on a central testbed control server. SANDbed [48]–[51] is also based on a conventional network management architecture, called *WBEM*. SANDbed's management covers the testbed network and hardware, but also provides experiment management to configure the parameters of individual test runs like energy measurements and simulated environments. It differs by the management of the DES-Testbed by its distributed approach with independent management sites. Each site is observed by a small management server and user client controlling the management system.

2) Management Scripts: In many of the early testbeds, self-developed management scripts replace some functional aspects of a complex management system or architecture. Various scripts developed for Motelab [3] are still in use and have been adapted in later works like Kansei [14], [15] and Indriya [57]. The scripts are mainly used for scheduling and to provide access to the nodes. Another testbed that primarily relies on parallel scripting, but also offers a central GUI for configuration is TWIST [9]–[11]. Script based testbed approaches often lack on flexibility because they use a central and hierarchical architecture and support only one specific node hardware and operating system.

3) Auction Based Management: Auction based management for sensor network testbeds has been researched in Mirage [77]. The project focuses on a time-balanced experiment scheduling by letting different users bid on resources for their projects with a virtual currency. The average bid prices are adapted depending on the individual requirements for testbed and node resources, priority and experiment runs during peak demand times. To our knowledge, auction based management has not been implemented in subsequent testbed deployments.

4) Management on Sensor Nodes: Testbed management is realized in Trio [24] directly on sensor nodes with the *Nucleus* management system. Nucleus is developed as a successor of *SNMS*, a protocol for the management of wireless sensor networks after deployment in their target environment. Nucleus can monitor the state of the nodes and track software images through the network. To avoid problems with failures and non-responding nodes, Nucleus uses an elaborated approach with golden images, watchdog and grenade timers [105] (cf. Section III-A).

5) Testbed Interconnections: The management of different interconnected testbeds increases the demand of uniform systems for administration. Especially, capabilities to authenticate and authorize different users, the support of heterogeneous hardware platforms and smart topology control mechanisms are required. For this purpose, TARWIS [82]–[84] has been designed to manage the *WISEBED* interconnection of testbeds. It is based on a web services architecture and uses *Shibboleth* for authentication and authorization.

Moreover, many hybrid management solutions were developed for specific sensor testbeds. For example, the ATMA management framework [78] utilizes IEEE 802.11 wireless mesh network to allow out-of-band monitoring, configuration and topology control. ATMA's out-of-band management approach led to the development of additional tools like an interference meter. Communication between a central *manager server* and the sensor nodes is realized with wireless mesh nodes for routing. Some mesh nodes, called *agents*, are located near the sensor nodes to execute management commands.

IV. Experimentation Support

Some aspects to simplify experimentation are shared in all discussed testbeds, except several testbeds developed just for a single experiment or application. In this section, we discuss these primarily experimentation focused aspects. We also present the usual practice to handle these aspects in current testbeds.

A. Node Power Source Control

Same as in a real deployment, nodes in testbeds of cause have to be supplied with power. There are several possibilities to power sensor nodes in a testbed.

1) Batteries: Batteries are a common way to power sensor nodes. However they are quite impractical in a testbed. Experiments are supposed to run many times and the nodes often stay turned on, even if they are not currently used for experiments. Due to this fact, the energy of the batteries is exhausted very quickly, making the testbed unusable for a while. The batteries have to be exchanged or recharged. Exchanging batteries on a large number of nodes is very time consuming. For automated recharge there is additional HW on every testbed node needed. Furthermore, doing automated recharge and using the nodes at the same time equals to just powering the nodes with a continuous power source, which is needed for recharge anyway. Experiments that require realistic behavior of the power source, e.g exhaustion and voltage decrease, cannot be done during recharge.

2) Node Connectors: Many sensor node platforms are equipped with various interfaces supposed for programming, debugging and connecting the nodes to base station gateways. The USB interface is widely used among available platforms. In most cases, the USB voltage is used for power supply while the node is connected and USB power is available. So it is convenient, using USB to connect the nodes to the infrastructure and use it for power supply. Unfortunately, potentially required additional HW for interfaces like USB, e.g., voltage regulators or serial converters, may significantly affect the energy consumption of the node. Such impacts may distort energy measurement results if provided, making them not realistic compared to a deployment with typical WSN power sources, e.g., batteries.

3) Harvesting: Energy harvesting, e.g., with solar cells, is a typical approach in WSNs to extend network lifetime and to reach the desired autonomy. Energy harvesters do not directly power the device, but are mostly used with rechargeable batteries or capacitors. Especially in backbone-less WSN testbeds, it is convenient to use harvesters for powering nodes. In GNOMES [1] solar cells were used together with two batteries. One battery was used for power supply while the other is being charged by the solar cell. In [37] an outdoor WSN testbed was realized similarly.

4) Voltage Regulators: Voltage regulators are the most comfortable solution for power sources in testbeds. For automation of experiments it is desirable to control the power supply of the testbed nodes. There are three reasons to use voltage regulators for power control.

a) *Voltage adaption*: Simplest approach is to use voltage regulators for voltage adaptation to meet the requirements of the sensor node platform. In many cases the nodes are powered over USB, so the USB voltage of 5 V has to be reduced to a more common supply voltage in WSNs, i.e., 3.3 V.
b) *Remote on/off switch*: A more advanced power control of testbed nodes is the ability to remotely turn them on and off. This allows a controlled node life-cycle during an experiment and can be used for simulation of node malfunctions, battery depletion or even attacks in security scenarios. Such a possibility is implemented in TWIST [11].
c) *Power source emulation*: The most flexible solution for power control in testbeds are programmable voltage regulators. With programmable power sources it is possible to emulate the behavior of power sources supposed for real deployments, e.g., batteries or various energy harvesters. For this purpose the voltage is regulated depending of the actual power consumption of the nodes. Doing so, it is possible to simulate effects from real power sources in WSNs, e.g., battery depletion or day time dependent voltage output of solar cells. Emulation of power sources enables flexible experimentation with different virtual power sources and thus simplifies the selection of an adequate power source for real deployments of the observed WSN application. Furthermore, this method improves evaluation of energy aware WSN applications, that take available energy resources into account and react accordingly to fluctuating voltages of the power source. WiLab [40], FlockLab [28] and SANDbed [49]–[51] are known examples for testbeds with programmable regulated power sources.

B. Experiment Conduction and Result Collection

Experiments can either be conducted in a scripted and automated way triggering events and measurements in a predetermined sequence. Alternatively, some testbeds allow to conduct experiments interactively with experimenters changing experiment related settings on the sensor nodes or injecting packets in the network during run-time.

While the automated approach supports the repeatability under similar conditions and an extensive logging of experiment inputs, an accurate experiment plan is needed in advance. Scheduling of experiments is also an essential requirement for multi-user testbeds. Testbeds supporting scheduling usually rely on scripted experiment execution. Some examples using this approach are SignetLab [16], [17], DES-Testbed [69]–[75] or Flocklab [27].

The interactive approach simplifies debugging of deployed software in sensor networks and allows a faster comprehension of experiment related correlations. A major drawback is often the fragmentary documentation of user triggered inputs during the experiments. Run-time interaction is realized e.g., in MoteLab [3], S-WiNeTest [18] and NetEye [35].

Regardless of automated or interactive experiment conduction, the results must be collected to analyze and evaluate the experiment outcome afterward. In research literature we can identify five different approaches to collect experiment results.

1) Central Database: The most widespread approach uses a central (SQL) database on a dedicated server which is deployed in the testbed backbone. Central SQL databases are usually chosen to get a complete and retrievable view of the testbed status covering all involved nodes at a single point in network. Each node's status and results during the experiment runs is stored in the database using database transactions to assure consistency. A transaction has to be triggered by the sensor node itself or a proxy entity in the testbed that is able to exchange messages with the database.

A database server is easy to realize because already existing software can be reused with many available tools. Besides result collection, a central database can be involved in further management tasks, e.g., experiment configuration, deployment, scheduling and user accounting. Furthermore, central databases are commonly used to store and organize the data of (multiple) experiments like the applications to deploy, setup configurations and parameters, input data or control scripts for experiment conduction. Some testbed examples with a central database for data gathering and furthermore for experimentation support can be found in [3], [14], [15], [35], [45]–[47].

However, a central database can be inappropriate in a roomy deployment with distributed nodes due to the involved overhead for communication during experiment runs. A central database can also turn out as a single point of failure in case high amount of measurement data causes performance problems in the network next to the database.

2) Distributed Approach: To adapt the benefits of a database solution to the distributed WSN approach and according testbed environments, some researchers like [49] suggest distributed result collection on local management nodes. These management nodes are usually small PC-like devices within the testbed, providing storage for result data in subareas of the testbed network. Nevertheless, this approach relies on additional tools like user clients or coordinating entities. The tools support the experimenters gathering results from distributed management nodes and maintaining a consistent status.

The authors of [42] realize distributed result collection, with results of elapsed experiments archived in local databases on intermediate gateway nodes. A central server is only used to initiate queries on the result data.

3) Hybrid Approach for Result Collection: A hybrid approach for result collection can be found in Flocklab [27]. Each sensor node is coupled with an observer node which is gathering the results during experiment runs in a local SQLite database and transfers its data periodically to a central server where users can access their bundled results.

4) Instrumenting Sensor Nodes: In the Motel [56] testbed, result collection and data-logging is done directly on sensor nodes. Therefore, a modular run-time management software called *FLEXOR* is deployed on the nodes. FLEXOR allows experiment task handling and backchannel-free experimentation besides the control of robot platforms coupled with the sensor nodes. While this approach may ease experimentation in small testbeds, extracting and analyzing correlated experiment results gets cumbersome if hundreds of nodes are involved. Another issue is the limited amount of memory and energy available on the nodes preventing full coverage of long-term experiments.

5) Result Collection During Experiment Run-Time: Some of the testbeds like MoteLAB [3], Motescope [22], Trio [24], KonTest [33], NetEye [35] explicitly support the TinyOS toolchain. TinyOS provides the popular *SerialForwarder* tool which is often used in testbed environments to allow experimenters to connect to the serial ports of sensor nodes with experiment specific tools. This enables direct access to the nodes for result collection during experiment run-time without a complex result collection system.

V. Trends and Future Work

Comparing earlier and first generation testbeds with current research works, we identified the following trends.

Wireless backbones: Modern sensor network testbeds try to increase their flexibility and application range. To conduct WSN experiments in an environment, comparable to the real target environment, many projects use wireless backbones. This allows more dynamical topology changes and outdoor deployments. Many testbeds started with indoor deployments and extended parts of their deployments into outdoor regions [12]–[15], [56].

"Nomadic" testbeds: Another approach to combine real environments with controlled experimentation are the *nomadic* testbeds [45]–[47]. Nomadic testbeds are based on the idea to use portable testbeds and rapidly deploy them in real environments.

Mobile nodes: Mobile robots with scripted motion models are very popular in sensor network testbeds. Many current projects use mobile nodes or plan to add such nodes to the testbed. Mobile nodes offer more flexibility for experiment design to solve the common problem of static testbed topologies. In future testbed research we will get an increasing number of mobile nodes.

Energy measurements: Many researchers already struggled with difficulties in energy efficient protocol design using simulations only. Especially the influence of wireless channel characteristics is not sufficiently covered by simulations. Some more recent testbeds try to measure the energy consumption of the nodes during experiments. They also validate simulation results and gather experience on energy efficient protocol and application development. Future research will benefit by energy inspection of commonly used algorithms and protocol mechanisms, currently analyzed in theory only.

Repeatable experiments: Just like simulation experiments, testbed experiments must be repeated to gain statistically significant results. Repeatable experiments are also needed to compare different experiment parameters or to increase the knowledge about impacting factors. Using real environments in testbeds can reveal design flaws and problems not covered in simulations with artificial environments. Realizing repeatability of experiments is challenging, due to the fact that testbeds use real hardware and a real, dynamically changing environment. The experiments still run in a artificial testbed with differences to the target environment of the WSN. The key for experiment repetition under comparable conditions, is to control the entire experimentation environment and the course of events as far as possible. Controlling the experimentation environment includes preparations for synchronous measurements and start/stop triggering on multiple nodes, time accurate event signaling and data logging. Furthermore, to measure comparable sensor readings in multiple experiment runs, the environment context, radio and media characteristics and disturbances, energy supply and the current testbed setup have to be considered.

Management software: Experimentation with real hardware can be a very cumbersome process. Necessary decisions for efficient experiments comprehend the scenario and the experiment conduction. This also involves preplanned events, devices in the testbed to interact with, result collection and interpretation. Latest testbeds deploy management systems to simplify research on real sensor networks by controlling hardware in conjunction with the experiments. These management systems tend to handle even large experiment data-sets during deployment and result gathering. Testbeds increasingly profit by experiment scheduling and scalability support because a manual setup of the nodes is becoming dispensable.

Powerful data evaluation tools: The huge amount of data generated by testbed experiments is worthless, if no tool support is given to evaluation and visualization. Many testbed researchers are aware of the lack of appropriate tools and working on their own tools. Currently, the provided tools are non-generic and tailored to a specific testbeds. To compare results of different testbeds, future tools tend to be more generic and can be shared between testbeds.

Interconnection of testbeds: WSN researchers can draw conclusions on a higher level of abstraction by overcoming the limitations of a distinct testbed environment and topology. Most research institutions can not afford to build up multiple testbeds with different hardware and environment conditions to compare experiment results. That is why an increasing number of institutions tend to form testbed federations to repeat their experiments with slightly different conditions. Shared nodes are also used to perform experiments with heterogeneous hardware on a larger scale. Realizing federations is no simple task, just connecting the individual testbeds via the Internet. Instead, sharing resources and forming links between the nodes in different testbeds entails virtualization of hardware and communication media. Experimenting with a combination of real and virtualized components is a non-trivial challenge and should be supported with a common API for all participating testbed partitions. Testbed federations are used by experimenters from different locations requiring additional arrangements for user management, authentication and resource reservation, allocation or scheduling. The European testbed federation *Wisebed* [59]–[61] and the interconnection of the Kansei and GENI testbeds to *KanseiGenie* [62] currently addressing this issues. With WHYNET [58] multiple distributed testbeds, supporting different physical technologies and protocol stacks, e.g., for *Ultra Wide Band* and *Software Defined Radios*, have already been interconnected. The project influenced the research on current wireless sensor testbed federations as a successor.

VI. Conclusion

In this survey, we presented an overview of WSN testbeds and discussed the underlying research foci and design considerations. We identified common aspects and requirements for experimentation and analyzed current solutions and testbed architectures. Finally, we conclude with some lessons learned and point out common open research and development issues for WSN testbeds in general.

Existing research work is very limited regarding energy measurement in testbeds. Accordingly, the energy consumption of protocols in an sensor network has not been researched in depth. Instead, only a few measurements on single nodes have been made in the community.

Accounting, authentication and security are concerns in every testbed used by distinct users. Some approaches to deal with these topics already exist for individual testbeds, but the ongoing trends to build federations urges to unify the different approaches. Researchers could profit by deploying their experiments on multiple testbeds with different characteristics but common access interfaces.

Experimenting with many nodes quickly creates scalability challenges and a huge amount of data for logging, debugging and measurement output. Currently, there is a lack of tools dealing (semi-)automatically with the amount of data and supporting the researchers to evaluate their projects. To experiment with really large sensor networks, simulations will still be the better choice due to the costs for hardware and maintenance.

A testbed can be seen as a large experimentation tool. To build an appropriate tool and a corresponding capable architecture, knowledge about the desired requirements is needed. We noticed that many existing testbeds have been developed without a clear definition of the requirements. Especially in some early testbed projects, the lack of exactly defined requirements contributed to abortion of the research efforts.

The testbed environment and the target sensor network environment might differ tremendously. Nevertheless, repeatability and validation of experiments is one of the most important concerns for significant experiment results. Developing a testbed to support repeatability and experiment validation leads to the question: How to control the different aspects of the experiment environment? Currently, the question has been touched only by a few projects but still needs to be covered in depth by future research. Especially, dealing with environment measuring sensors, we can not simply fade out a testbed's environment. Today, many testbeds can already adapt their topology and the communication range of the nodes to be compliant with an examined application.

References

[1] E. Welsh, W. Fish, and J. Frantz, "GNOMES: A testbed for low power heterogeneous wireless sensor networks," in *Proc. ISCAS*, May 2003, vol. 4, pp. IV-836–IV-839.

[2] J. Blumenthal, M. Handy, and D. Timmermann, "SeNeTs—TEST and validation environment for applications in large-scale wireless sensor networks," in *Proc. 2nd IEEE INDIN*, Jun. 2004, pp. 69–73.

[3] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: A wireless sensor network testbed," in *Proc. 4th Int. Symp. IPSN*, Apr. 2005, pp. 483–488.

[4] A. Arora *et al.*, "Exscal: elements of an extreme scale wireless sensor network," in *Proc. 11th IEEE Int. Conf. Embedded RTCSA*, Aug. 2005, pp. 102–108.

[5] D. Jia, B. H. Krogh, and C. Wong, "TOSHILT: Middleware for hardware-in-the-loop testing of wireless sensor networks," Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep., 2005, white paper.

[6] Tutornet: A Tiered Wireless Sensor Network Testbed. [Online]. Available: http://enl.usc.edu/projects/tutornet/

[7] D. Johnson *et al.*, "Demo abstract: Emulab's wireless sensor net testbed: True mobility, location precision, and remote access," in *Proc. 3rd Int. Conf. Embedded networked SenSys*, Nov. 2005, pp. 306–306.

[8] D. Johnson *et al.*, "Mobile Emulab: A robotic wireless and sensor network testbed," in *Proc. 25th IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[9] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, "TWIST: A Scalable and Reconfigurable Wireless Sensor Network Testbed for Indoor Deployments," Telecommunication Networks Group, Technische Universität Berlin, Berlin, Germany, Tech. Rep. TKN-05-008, Nov. 2005.

[10] S.-F. Li, V. Handziski, A. Köpke, M. Kubisch, and A. Wolisz, "A wireless sensor network testbed supporting controlled in-building experiments," in *Proc. 12th Sensor Kongress*, May 2005. [Online]. Available: http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/sensors2005_v3.pdf

[11] A. Köpke and A. Wolisz, "Measuring the node energy consumption in USB based WSN testbeds," in *Proc. 28th ICDCS*, Jun. 2008, pp. 333–338.

[12] T. Schmid, H. Dubois-ferrière, and M. Vetterli, "SensorScope: Experiences with a wireless building monitoring sensor network," in *Proc. 1st Workshop REALWSN*, Jun. 2005, pp. 13–17.

[13] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "Wireless sensor networks for environmental monitoring: The SensorScope experience," in *Proc. IEEE Int. Zurich Seminar Commun.*, Mar. 2008, pp. 98–101.

[14] A. Arora, E. Ertin, R. Ramnath, M. Nesterenko, and W. Leal, "Kansei: A high-fidelity sensing testbed," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 35–47, Mar./Apr. 2006.

[15] E. Ertin *et al.*, "Kansei: A testbed for sensing at scale," in *Proc. 5th Int. Conf. IPSN*, 2006, pp. 399–406.

[16] R. Crepaldi, A. Harris, A. Scarpa, A. Zanella, and M. Zorzi, "SignetLab: Deployable sensor network testbed and management tool," in *Proc. 4th Int. Conf. Embedded Netw. SenSys*, Nov. 2006, pp. 375–376, demo.

[17] R. Crepaldi, S. Friso, A. F. Harris, III, A. Zanella, and M. Zorzi, "The design, deployment, analysis of signetLab: A sensor network testbed and interactive management tool," in *Proc. 1st Int. Workshop WiNTECH*, 2006, pp. 93–94.

[18] L. Galluccio, A. Leonardi, G. Morabito, A. Panto, and F. Scoto, "S-WiNeTest: Remote access to a sensor testbed," in *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*, F. Davoli, S. Palazzo, and S. Zappatore, Eds. New York, NY, USA: Springer-Verlag, 2006, ser. Signals and Communication Technology, pp. 215–231.

[19] SensoNet Project—Testbed for Sensor Networks, 2004. [Online]. Available: http://www.ece.gatech.edu/research/labs/bwn/Sensor/testbed.html

[20] J. Abdo, N. Georganas, and H. Mouftah, "Test-bed for sensor network management protocols," in *Proc. 23rd Biennial Symp. Commun.*, May/Jun. 2006, pp. 312–315.

[21] C. Gruenwald, A. Hustvedt, A. Beach, and R. Han, "SWARMS: A Sensornet wide area remote management system," in *Proc. 3rd Int. Conf. TridentCom*, May 2007, pp. 1–10.

[22] UC Berkeley Motescope, 2007–2008. [Online]. Available: http://smote.cs.berkeley.edu/motescope/

[23] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler, "Energy metering for free: Augmenting switching regulators for real-time monitoring," in *Proc. Int. Conf. IPSN*, Apr. 2008, pp. 283–294.

[24] P. Dutta *et al.*, "Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments," in *Proc. 5th Int. Conf. IPSN*, Apr. 2006, pp. 407–415.

[25] Motescope & Omega Testbeds, University of California, Berkeley 2007. [Online]. Available: https://www.millennium.berkeley.edu/wiki/sensornets

[26] T. Dimitriou, J. Kolokouris, and N. Zarokostas, "Sensenet: A wireless sensor network testbed," in *Proc. 10th ACM Symp. MSWiM*, 2007, pp. 143–150.

[27] J. Beutel *et al.*, "Poster abstract: The FlockLab testbed architecture," in *Proc. 7th ACM Conf. Embedded Networked SenSys*, Nov. 2009, pp. 415–416.

[28] R. Lim *et al.*, "Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proc. 12th Int. Conf. IPSN*, 2013, pp. 153–166.

[29] M. Dyer *et al.*, "Deployment support network—A toolkit for the development of WSNss," in *Proc. 4th EWSN*, 2007, pp. 195–211.

[30] wsCoop: Wireless Sensor Cooperation, 2007. [Online]. Available: http://www.deeds.informatik.tu-darmstadt.de/wscoop/testbed.html

[31] C. S. Metcalf, "TOSSIM live: Towards a testbed in a thread," Master's thesis, Colorado School of Mines, Golden, CO, USA, 2007.

[32] C. S. Metcalf, T. Camp, M. Colagrosso, and O. Chase, "TestbedProfiler: A validation tool for wireless sensor network testbed deployment," in *Proc. 35th IEEE Conf. LCN*, 2010, pp. 188–191.

[33] K. Iwanicki, A. Gaba, and M. van Steen, "KonTest: A wireless sensor network testbed at Vrije Universiteit Amsterdam," Vrije Universiteit, Amsterdam, Development Laboratories(DevLab), Eindhoven, The Netherlands, 2008, Tech. Rep..

[34] J.-P. Sheu, C.-J. Chang, C.-Y. Sun, and W.-K. Hu, "WSNTB: A testbed for heterogeneous wireless sensor networks," in *Proc. 1st IEEE Int. Conf. Ubi-Media Comput.*, Aug. 2008, pp. 338–343.

[35] D. Sakamuri, "NetEye: A wireless sensor network testbed," Master's thesis, Graduate School of Wayne State University, Detroit, MI, USA, 2008.

[36] J. Slipp *et al.*, "WINTeR: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments," in *Proc. 6th Annu. CNSR Conf.*, May 2008, pp. 422–431.

[37] S. Chen, Y. Huang, and C. Zhang, "Toward a real and remote wireless sensor network testbed," in *Proc. 3rd Int. Conf. WASA*, 2008, pp. 385–396.

[38] I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, "PowerBench: A scalable testbed infrastructure for benchmarking power consumption," in *Proc. IWSNE*, Santorini Island, Greece, Jun. 2008, pp. 37–44.

[39] J. He, "Design and implementation of a wireless sensor network testbed," Master's thesis, Wireless and Mobile Communication (WMC) Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, 2009.

[40] L. Tytgat *et al.*, "Demo abstract: WiLab, a real-life wireless sensor testbed with environment emulation," in *Proc. EWSN Adjunct Poster*, Feb. 2009, pp. 9–10.

[41] A. Achtzehn, E. Meshkova, J. Ansari, and P. Mahonen, "MoteMaster: A scalable sensor network testbed for rapid protocol performance evaluation," in *Proc. 6th Annu. IEEE Commun. Soc. Conf. SECON Workshops 2009*, Jun. 2009, pp. 1–3.

[42] A. Kanzaki, T. Hara, Y. Ishi, N. Wakamiya, and S. Shimojo, "X-Sensor: A sensor network testbed integrating multiple networks," in *Proc. Int. Conf. CISIS*, Mar. 2009, pp. 1082–1087.

[43] S. G. Hong, Y. B. Moon, S. J. Park, and W. W. Kim, "Wireless sensor network testbed for real-time sensor monitoring," in *Proc. 3rd Int. Conf. SENSORCOMM*, Jun. 2009, pp. 486–489.

[44] C. Weyer, C. Renner, V. Turau, and H. Frey, "A roadmap for hardware and software support for developing energy-efficient sensor networks," in *Proc. 8th GI/ITG KuVS FGSN*, Aug. 2009, pp. 67–70.

[45] O. Rensfelt, F. Hermans, C. Ferm, and A. Larzon, "Sensei-UU: A nomadic sensor network testbed supporting mobile nodes," Uppsala Universitet, SUppsala, Sweden, Tech. Rep., Oct. 2009.

[46] O. Rensfelt, F. Hermans, C. Ferm, L.-A. Larzon, and P. Gunningberg, "Sensei—A flexible testbed for heterogeneous wireless sensor networks," in *Proc. 5th Int. Conf. TridentCom*, Apr. 2009, pp. 1–2.

[47] O. Rensfelt, F. Hermans, L.-A. Larzon, and P. Gunningberg, "Sensei-UU: A relocatable sensor network testbed," in *Proc. 5th ACM Int. Workshop WiNTECH*, 2010, pp. 63–70.

[48] A. Hergenröder, J. Horneber, D. Meier, P. Armbruster, and M. Zitterbart, "Distributed energy measurements in wireless sensor networks," in *Proc. 7th ACM Conf. Embedded Netw. SenSys*, Nov. 2009, pp. 299–300.

[49] A. Hergenröder, J. Horneber, and J. Wilke, "SANDbed: A WSAN testbed for network management and energy monitoring," TU Hamburg-Harburg, Hamburg, Germany, Tech. Rep., Aug. 2009.

[50] A. Hergenröder, J. Wilke, and D. Meier, "Distributed energy measurements in WSN testbeds with a sensor node management device (SNMD)," in *Proc. 23rd Int. Conf. Archit. Comput. Syst. Workshop*, M. Beigl and F. J. Cazorla-Almeida, Eds., Feb. 2010, pp. 1–7.

[51] A. Hergenröder and J. Horneber, "Facing challenges in evaluation of WSN energy efficiency with distributed energy measurements," in *Proc. 7th IWCMC*, Jul. 2011, pp. 1004–1009.

[52] Z. Zhao, G.-H. Yang, Q. Liu, V. O. Li, and L. Cui, "EasiTest: A multi-radio testbed for heterogeneous wireless sensor networks," in *Proc. IET-WSN*, Nov. 2010, pp. 104–108.

[53] Q. Liu, Z. Zhao, and L. Cui, "Demo abstract: A versatile heterogeneous sensor networks testbed," in *Proc. 8th ACM Conf. Embedded Netw. SenSys*, Nov. 2010, pp. 387–388.

[54] T. Ducrocq, J. Vandaele, N. Mitton, and D. Simplot-Ryl, "Large scale geolocalization and routing experimentation with the SensLAB testbed," in *Proc. 7th IEEE Int. Conf. MASS*, 2010, pp. 751–753.

[55] W. Huangfu, L. Sun, and J. Liu, "A high-accuracy nonintrusive networking testbed for wireless SensorNetworks," *EURASIP J. Wireless Commun. Netw.*, vol. 2010, pp. 2:1–2:15, Feb. 2010.

[56] A. Förster *et al.*, "Poster abstract: MOTEL: Towards flexible mobile wireless sensor network testbeds," in *Proc. 8th EWSN Poster Session*, Feb. 2011, pp. 1–2.

[57] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, "Indriya: A low-cost, 3D wireless sensor network testbed," in *Proc. 7th Int. ICST Conf. TridentCom*, Apr. 2011, pp. 302–316.

[58] WHYNET: Scalable Testbed for Next Generation Mobile Wireless Networking Technologies 2003–2004. [Online]. Available: http://pcl.cs.ucla.edu/projects/whynet/

[59] S. Fischer, D. Pfisterer, S. P. Fekete, and A. Kroeller, "WISEBED—Pan-European wireless sensor network testbeds," Fachgespraech der GI/ITG-Fachgruppe Kommunikation und Verteilte Systeme, Tech. Rep., 2008.

[60] I. Chatzigiannakis, S. Fischer, C. Koninis, G. Mylonas, and D. Pfisterer, "WISEBED: An open large-scale wireless sensor network testbed," in *Sensor Applications, Experimentation, and Logistics*, vol. 29, O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, M. Gerla, H. Kobayashi, S. Palazzo, S. Sahni, X. S. Shen, M. Stan, J. Xiaohua, A. Zomaya, G. Coulson, and N. Komninos, Eds. Berlin, Germany: Springer-Verlag, 2010, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 68–87.

[61] T. Baumgartner *et al.*, "Wiselib: A generic algorithm library for heterogeneous sensor networks," in *Proc. 7th EWSN*, J. S. Silva, B. Krishnamachari, and F. L. Boavida, Eds., 2010, pp. 162–177.

[62] M. Sridharan *et al.*, "From Kansei to KanseiGenie: Architecture of federated, programmable wireless sensor fabrics," in *Proc. 6th Int. Conf. TridentCom*, May 2010, pp. 155–165.

[63] J. Elson *et al.*, "EmStar: An Environment for Developing Wireless Embedded Systems Software," Center for Embedded Networked Sensing, University of California, Los Angeles, CA, USA, Tech. Rep., 2003.

[64] D. Raychaudhuri *et al.*, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. IEEE WCNC*, Mar. 2005, vol. 3, pp. 1664–1669.

[65] M. Ott, I. Seskar, R. Siraccusa, and M. Singh, "ORBIT Testbed software architecture: Supporting experiments as a service," in *Proc. 1st Int. Conf. Tridentcom*, Feb. 2005, pp. 136–145.

[66] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proc. 11th Annu. Int. Conf. MobiCom*, 2005, pp. 31–42.

[67] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "Mint: A miniaturized network testbed for mobile wireless research," in *Proc. 24th Annu. Joint Conf. IEEE INFOCOM*, Mar. 2005, pp. 2731–2742.

[68] P. De *et al.*, "MiNT-m: An autonomous mobile wireless experimentation platform," in *Proc. 4th Int. Conf. MobiSys*, Jun. 2006, pp. 124–137.

[69] A. Zimmermann, M. Günes, M. Wenig, J. Ritzerfeld, and U. Meis, "Architecture of the hybrid MCG-mesh testbed," in *Proc. 1st Int. Workshop WiNTECH*, Sep. 2006, pp. 88–89.

[70] A. Zimmermann, M. Günes, M. Wenig, U. Meis, and J. Ritzerfeld, "How to study wireless mesh networks: A hybrid testbed approach," in *Proc. 21st Int. Conf. AINA*, May 2007, pp. 853–860.

[71] B. Blywis, F. Juraschek, M. Günes, and J. Schiller, "Design concepts of a persistent wireless sensor testbed," presented at the 7. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", Berlin, Germany, 2008.

[72] M. Güneş, B. Blywis, and J. Schiller, "A hybrid testbed for long-term wireless sensor network studies," in *Proc. IWSNE*, Santorini, Greece, Jun. 2008.

[73] M. Güneş, B. Blywis, and F. Juraschek, "Concept and Design of the Hybrid Distributed Embedded Systems Testbed," Freie Universität Berlin, Berlin, Germany, Tech. Rep. TR-B-08-10, 2008.
[74] M. Güneş, F. Juraschek, B. Blywis, Q. Mushtaq, and J. Schiller, "A testbed for next generation wireless networks research," *Special Issue PIK Mobile Ad-hoc Netw.*, vol. 34, no. 4, pp. 208–212, 2009.
[75] O. Hahm, M. Güneş, and K. Schleiser, "DES-testbed A wireless multihop network testbed for future mobile networks," presented at the 5th GI/ITG KuVS Workshop Future Internet, Stuttgart, Germany, Jun. 2010.
[76] R. N. Murty *et al.*, "CitySense: A vision for an urban-scale wireless networking testbed," Harvard University, Cambridge, MA, USA, Tech. Rep., Sep. 2007.
[77] B. N. Chun *et al.*, "Mirage: A microeconomic resource allocation system for sensornet testbeds," in *Proc. 2nd IEEE Workshop EmNets*, May 2005, pp. 19–28.
[78] K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer, "A framework for the management of large-scale wireless network testbeds," in *Proc. 1st Workshop WiNMee*, Apr. 2005. [Online]. Available: http://www.winmee.org/2005/papers/WiNMee_Ramachandran.pdf
[79] A. Milenkovic, M. Milenkovic, E. Jovanov, D. Hite, and D. Raskovic, "An environment for runtime power monitoring of wireless sensor network platforms," in *Proc.37th SSST*, Mar. 2005, pp. 406–410.
[80] X. Jiang, P. Dutta, D. Culler, and I. Stoica, "Micro power meter for energy monitoring of wireless sensor NetworksatScale," in *Proc. 6th Int. Conf. IPSN*, Apr. 2007, pp. 186–195.
[81] D. Hillenbrand, M. Mende, T. Armstrong, and J. Henkel, "Hyperion: A sensor node test bed for (high-speed) power measurements," in *Proc. IEEE/ACM Design Autom. Test Eur. Conf. Session 8: Power Manag. Exhib.*, Apr. 2007. [Online]. Available: http://ces.itec.kit.edu/~hillenbrand/papers/date-hillenbrand.pdf
[82] G. Wagenknecht *et al.*, "MARWIS: A management architecture for heterogeneous wireless sensor networks," in *Wired/Wireless Internet Communications*, vol. 5031, J. Harju, G. Heijenk, P. Langendörfer, and V. Siris, Eds. Berlin, Germany: Springer-Verlag, 2008, ser. Lecture Notes in Computer Science, pp. 177–188.
[83] P. Hurni, T. Staub, G. Wagenknecht, M. Anwander, and T. Braun, "A secure remote authentication, operation and management infrastructure for distributed wireless sensor network testbeds," *Electronic Commun. EASST*, vol. 17, pp. 1–6, 2009. [Online]. Available: http://journal.ub.tu-berlin.de/eceasst/issue/view/24
[84] P. Hurni, G. Wagenknecht, M. Anwander, and T. Braun, "A Testbed Management Architecture for Wireless Sensor Network Testbeds (TARWIS)," in *Proc. 7th EWSN*, Feb. 2010, pp. 33–35.
[85] A. Dwivedi and O. Vyas, "An exploratory study of experimental tools for wireless sensor networks," *Wireless Sensor Netw.*, vol. 3, no. 7, pp. 215–240, 2011.
[86] M. Imran, A. Said, and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks," in *Proc. ITSim*, Jun. 2010, vol. 2, pp. 897–902.
[87] L. P. Steyn and G. Hancke, "A survey of wireless sensor network testbeds," in *Proc. AFRICON*, Sep. 2011, pp. 1–6.
[88] A. Seema and M. Reisslein, "Towards efficient wireless video sensor networks: A survey of existing node architectures and proposal for a flexi-WVSNP design," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 462–486, 2011.
[89] I. Akyildiz, T. Melodia, and K. Chowdhury, "Wireless multimedia sensor networks: Applications and testbeds," *Proc. IEEE*, vol. 96, no. 10, pp. 1588–1605, Oct. 2008.
[90] M. Farooq and T. Kunz, "Wireless multimedia sensor networks testbeds and state-of-the-art hardware: A survey," in *Communication and Networking*, vol. 265, T. Kim, H. Adeli, W. Fang, T. Vasilakos, A. Stoica, C. Patrikakis, G. Zhao, J. Villalba, and Y. Xiao, Eds. Berlin, Germany: Springer-Verlag, 2012, ser. Communications in Computer and Information Science, pp. 1–14.
[91] D. Sakamuri and H. Zhang, *Handbook on Sensor Networks*. Singapore: World Scientific, 2010, ch. 11, pp. 229–262, Elements of Sensornet Testbed Design.
[92] A. Gluhak *et al.*, "A survey on facilities for experimental internet of things research," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 58–67, Nov. 2011.
[93] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simutools*, Mar. 2008, pp. 60:1–60:10.
[94] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st Int. Conf. Embedded Netw. SenSys*, Nov. 2003, pp. 126–137.
[95] R. de Paz Alberola and D. Pesch, "AvroraZ: Extending Avrora with an IEEE 802.15.4 compliant radio chip model," in *Proc. 3nd ACM Workshop PM2HW2N*, Oct. 2008, pp. 43–50.
[96] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. 2nd ACM Conf. Embedded Netw. SenSys*, Nov. 2004, pp. 188–200.
[97] 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 *Proc. 3nd ACM Workshop PM2HW2N*, Oct. 2008, pp. 35–42.
[98] C. Haas, V. Stöhr, and J. Wilke, "Realistic simulation of energy consumption in wireless sensor networks," in *Proc. 9th EWSN*, Feb. 2012, pp. 82–97.
[99] A. Hergenröder and J. Furthmüller, "On energy measurement methods in wireless networks," in *Proc. ICC*, Jun. 2012, pp. 6268–6272.
[100] J. Andersen and M. Hansen, "Energy bucket: A tool for power profiling and debugging of SensorNodes," in *Proc. 3rd Int. Conf. SENSORCOMM*, Jun. 2009, pp. 132–138.
[101] J. Furthmüller and O. P. Waldhorst, "Energy-aware resource sharing with mobile devices," in *Proc. 8th Int. Conf. WONS*, Jan. 2011, pp. 52–59.
[102] R. Fonseca, P. Dutta, P. Levis, and I. Stoica, "Quanto: Tracking energy in networked embedded systems," in *Proc. 8th USENIX Conf. OSDI*, Dec. 2008, pp. 323–338.
[103] D. Watson and M. Nesterenko, "Mule: Hybrid simulator for testing and debugging wireless sensor networks," in *Proc. 2nd Int. Workshop Sensor Actor Netw. Protocols Appl.*, Aug. 2004, pp. 67–71.
[104] G. Tolle and D. Culler, "Design of an application-cooperative management system for wireless sensor networks," in *Proc. 2nd Eur. Workshop Wireless Sensor Netw.*, Jan./Feb. 2005, pp. 121–132.
[105] F. Stajano and R. Anderson, "The grenade timer: Fortifying the watchdog timer against malicious mobile code," in *Proc. 7th Int. Workshop MoMuC*, Sep. 2000, pp. 1–5.

Jens Horneber received the Diploma in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 2008.

He is currently working as a Researcher on telematics with the Karlsruhe Institute of Technology, Karlsruhe. Since then, he has been involved in the DFG Research Training Group GRK1194 "Self-organizing Sensor-Actuator-Networks" and the development of the testbed *SANDbed*. His research interests include robust and energy-efficient communication and network management protocols in wireless sensor networks.

Anton Hergenröder received the Diploma in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 2009.

He is currently a Researcher with the Institute of Telematics, Karlsruhe Institute of Technology (KIT), Karlsruhe. Since then, he has been involved in research on wireless sensor actuator networks and the development of the testbed *SANDbed*. His research interests include energy-efficient communication and security in distributed systems and wireless sensor networks.