

Vergleichbarkeit von Ansätzen zur Netzwerkanalyse in drahtlosen Sensornetzen

Joachim Wilke

Institut für
Telematik
Universität Karlsruhe (TH)
wilke@ira.uka.de

Zinaida Benenson
Institut für Informatik
Universität Mannheim
zina@uni-mannheim.de

Frank Werner

Institut für
Theoretische Informatik
Universität Karlsruhe (TH)
werner@ira.uka.de

Simon Kellner
Institut für Betriebs- und
Dialogsysteme
Universität Karlsruhe (TH)
kellner@ira.uka.de

Markus Besthorn

Institut für Programmstrukturen
und Datenorganisation
Universität Karlsruhe (TH)
besthorn@ira.uka.de

Erik-Oliver Blaß
Networking & Security
EURECOM France
blass@eurecom.fr

Kurzfassung

In diesem Dokument wird die Vergleichbarkeit verschiedener Netzwerk-Simulatoren, -Emulatoren und theoretischer Analyse in drahtlosen Sensornetzen untersucht. Die Ergebnisse dieses im Rahmen des ZeuS-Forschungsprojektes durchgeführten Experiments zeigen die Stärken und Schwächen der verschiedenen Ansätze auf.

1. EINLEITUNG

Im Rahmen des ZeuS-Forschungsverbundes¹, einem Projekt zur Erforschung von Grundlagen zuverlässiger Kommunikation in drahtlosen Sensornetzen, werden an den beteiligten Instituten Simulationen und Evaluierung mit verschiedensten Werkzeugen durchgeführt. Neben einer Netzwerksimulation mit GloMoSim kommen auch die Eigenentwicklung KSNSim sowie der Emulator des TinyOS-Betriebssystems TOSSIM zum Einsatz. Zusätzlich wurde in einer formalen, theoretischen Analyse der Konnektivitätsgraph verschiedener untersuchter Topologien durch Markovketten modelliert und mittels Prism [6] analysiert. Um die Vergleichbarkeit der so auf unterschiedliche Weise gewonnenen Ergebnisse auch über die einzelnen Teilprojekte hinaus zu gewährleisten, wurde ein *gemeinsames Experiment* durchgeführt. Der folgende Bericht dokumentiert dieses Experiment und stellt eine Auswertung der Ergebnisse und der daraus gewonnenen Erkenntnisse dar.

Nach einem Überblick über verwandte Forschungsarbeiten in Kapitel 2 wird der Versuchsaufbau des durchgeführten Experiments in Abschnitt 3 beschrieben. Es folgt mit Abschnitt 4 eine Diskussion der Besonderheiten der einzelnen eingesetzten Werkzeuge. Eine Analyse und Zusammenfassung der Ergebnisse in Kapitel 5 schließen diese Arbeit ab.

2. VERWANDTE ARBEITEN

Cavin et. al haben 2002 bereits einige MANET-Simulatoren verglichen [4]. Die dort verwendeten Simulatoren *OPNET Modeler*, *NS-2* und *GloMoSim* überschneiden sich jedoch nur in einem Fall mit den bei uns verwendeten Werkzeugen. Weiterhin beschränkt sich diese Arbeit nicht auf reine Simulatoren, sondern greift mit TOSSIM auch auf einen Emulator und mit der theoretischen Analyse formale Ansätze auf.

3. VERSUCHSBESCHREIBUNG

Untersucht werden im Folgenden verschiedene Parameter die sich in einem Netzwerk durch einfaches *probabi-*

listisches Fluten einer Nachricht N bestimmen lassen: der Energieverbrauch, die Anzahl versendeter bzw. empfangener Nachrichten, die Wahrscheinlichkeit, dass einzelne Knoten die Nachricht erhalten haben und besondere Eigenschaften/Messgrößen die das betrachtete Programm unterstützt, abhängig von der Simulationsumgebung (siehe Abschnitt 4). Die Topologien wurden so gewählt, dass die Auswirkung von probabilistischem Fluten auf symmetrische und unsymmetrische Sensoranordnungen (Abbildung 1c) untersucht werden kann.

In allen betrachteten Topologien bekommt Knoten A von der Basisstation die Nachricht N und leitet diese mit Wahrscheinlichkeit $p = 1$ weiter. Bei allen übrigen Knoten wird beim Empfang eines Paketes „gewürfelt“ (d.h., eine Zufallszahl $0 < p_Z \leq 1$ wird gezogen) und entsprechend dem Ausgang ($p_Z < p$) das Paket mit der Wahrscheinlichkeit p an alle Nachbarknoten weitergeleitet.

Um präzise Daten und eine gute Vergleichbarkeit zu erhalten, soll eine möglichst geringe relative Standardabweichung in den einzelnen Ergebnissen erzielt werden. Dazu sollten mindestens 200 Durchläufe für jede Konfiguration durchgeführt werden.

3.1 Topologiewahl

Das Experiment wurde mit unterschiedlichen Topologien durchgeführt. Für diese Arbeit wurden drei Topologien (Abb. 1) ausgewählt, die deutliche Unterschiede bezüglich Aufbau, Symmetrie und Vernetzung zeigen.

Die Kanten innerhalb der Topologien repräsentieren den Konnektivitätsgraph (*Routing-Baum*), also den Weg, den die Nachricht N im Netz zurücklegen soll. Die Distanz zwischen den Knoten, für die eine Verbindung (Kante im Konnektivitätsgraph) existiert, soll höchstens 10 Meter sein. Die zu flutende Nachricht N beträgt exakt 56 Byte (maximale Paketgröße in TinyOS). Mit Beginn der Simulation befinden sich alle Knoten im eingeschalteten Zustand und warten auf Nachrichten. Die Position der einzelnen Sensorknoten werden für alle Topologien in einer Szenariobeschreibung definiert, so dass diese allen Experimenten als Ausgangsbasis zur Verfügung stehen. Neben den Topologien wird auch die Wahrscheinlichkeit p einer Weiterleitung variiert, wobei $0,2 \leq p \leq 1$ in Schritten von je 0,2 simuliert wird. Abhängig von den Möglichkeiten der verwendeten Werkzeuge werden weitere Parameter spezifiziert. Alle wichtigen Simulationsparameter sind in Tabelle 1 zusammengefasst.

¹<http://www.zeus-bw-fit.de>

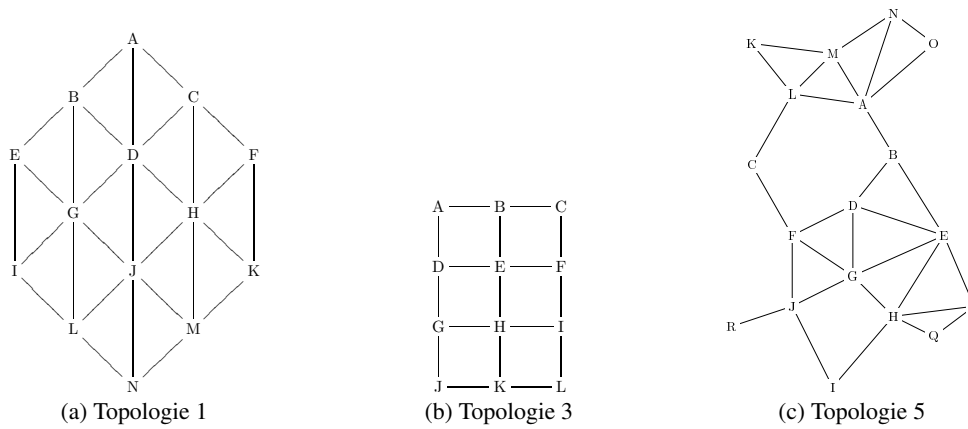


Figure 1: Verwendete Topologien

Nachrichtenlänge	$l = 56$ Bytes [5]
Weiterleitungswahrscheinlichkeiten	$p \in \{0,2; 0,4; 0,6; 0,8; 1,0\}$
MAC	CSMA/CA
Sendereichweite	$r = 10$ Meter
Höhe der Antenne	$h = 1,5$ Meter
Datenrate	$R = 38400$ Bit/s [5]
Stromverbrauch	
Funk	$I_{Funk} = 16$ mA [5]
CPU	$I_{CPU} = 5$ mA [1, 3]
Energieverbrauch	(siehe Abschnitt 3.2)
Nachricht versenden	$P_m = 185,2$ μ As
Nachricht empfangen	$P_m = 42$ μ As

Table 1: Simulations-Parameter

3.2 Energie

Die wohl interessanteste Metrik in Sensornetzen, den Energieverbrauch, mit unterschiedlichen Simulatoren zu erfassen, ist eine Herausforderung für sich. Einerseits soll der simulierte Verbrauchswert nahe an der Realität liegen, andererseits ist beim Einsatz verschiedener Werkzeuge darauf zu achten, dass die Ergebnisse vergleichbar bleiben: Da einige der eingesetzten Simulatoren nur auf Netzwerkebene arbeiten, bleibt als gemeinsame Datenbasis leider nur Paketempfang und -versand übrig. Daher wurden diese Operationen auf realen Knoten (MICAz) vermessen.

Um den Eigenschaften der Energieressource Rechnung zu tragen, wurde die der Batterie entnommene Ladung mit Hilfe eines NI-USB-6210 [9] gemessen. Dem Empfang einer 56 Byte großen Nachricht wurde ein Ladungswert von 42μ As und eine Dauer von $1,85$ ms im Durchschnitt zugeordnet, gemessen vom Beginn des Empfangs am CC2420 bis zur Benachrichtigung der Anwendung. Der Versand eines Pakets derselben Größe dauert hingegen im Durchschnitt $8,34$ ms und benötigt dabei $0,185$ mAs an Ladung, gemessen vom Zeitpunkt, an dem die Anwendung den Sendeauftrag erteilt, bis zur Versandbestätigung der TinyOS Netzwerkschicht.

Nicht berücksichtigt werden hier Energiewerte für den Mikroprozessor und das An-/Ausschalten des CC2420, da diese für den Vergleich der Werkzeuge nicht relevant sind. Der Stromverbrauch beim Lauschen auf Pakete sowie zusätzlicher Energieaufwand bei Kollisionen bleibt ebenfalls unberücksichtigt, da die notwendigen Zahlen nicht von allen verglichenen Werkzeugen geliefert wer-

den. Deshalb stellen die Energiewerte aus den Simulatoren sehr kleine untere Schranken für den Energieverbrauch von MICAz Knoten dar, die in der Praxis nicht erreicht werden können. Die damit in verschiedenen Simulatoren errechneten Werte lassen sich jedoch gut vergleichen.

4. AUSWERTUNG

Im Folgenden werden die einzelnen Experimente und die Besonderheiten der einzelnen Werkzeuge vorgestellt.

4.1 GloMoSim

Die Implementierung des Versuchs erfolgte als Protokoll auf Anwendungsebene. Die von GloMoSim bereitgestellten Implementierungen der niedrigeren Netzwerkschichten wurden unverändert übernommen und lediglich in ihrer Parametrisierung gemäß den Vorgaben angepasst.

Die Knotenplatzierung wurde direkt aus den einheitlich definierten Szenariobeschreibungen übernommen und in der Konfiguration mit *NODE-PLACEMENT-FILE* referenziert. Die Signalausbreitung wurde mit dem in GloMoSim vorhandenem *TWO-RAY-Modell* simuliert, welches ein *PROPAGATION-LIMIT* von -111 dBm vorsieht. Die Parameter zur Funkschnittstelle wurden entsprechend der MicaZ-Spezifikation angepasst. Insbesondere wurde das *SNR-BOUNDED* Modell zum Empfang von Datenpaketen ausgewählt, welches Signale über dem *RADIO-RX-SNR-THRESHOLD* von 10 dB fehlerfrei zustellt und sonst verwirft. Als *MAC-PROTOCOL* kommt *CSMA-CA* zum Einsatz. Die Simulationsergebnisse sind Mittelwerte von je 1000 Durchläufen mit variierenden *SEEDs* des GloMoSim Zufallsgenerators.

4.2 TOSSIM

TOSSIM ist ein Simulator, der von TinyOS bereitgestellt wird. Sein Radiomodell basiert auf der Signalverstärkung (*gain*), die jeweils zwischen zwei Knoten festgelegt wird. Da die Signalstärke mit der Entfernung abnimmt, ist *gain* immer negativ. In diesem Experiment haben wir die *gains* mit Hilfe der von TOSSIM bereitgestellten Hilfsmittel generiert. Als Eingaben dienen die Koordinaten der Knoten und mehrere Umgebungsparameter (*path-loss exponent*, *noise floor*, *white Gaussian noise* usw.), die dem Tutorial [12] entnommen wurden.

TOSSIM simuliert Umgebungsrauschen und Interferen-

```

global a : [0..1] init 1;

module SensorA
  // Local State
  // (0=idle, 1=forward, 2=drop, 3=sleep)
  la: [0..3] init 0;

  [] (a=1) & (la=0) -> pf: (la'=1) & (b'=1) &
                        (d'=1) & (c'=1)
                        + 1-pf: (la'=2);
  [] (a=1) & (la=2) -> (la'=2);
endmodule

```

Figure 2: Prism Modell eines Moduls, das einen Sensor-knoten repräsentiert

zen mit Hilfe des *Closest Pattern Matching (CPM)* Verfahrens [8]. CPM braucht als Eingabe sogenannte *noise traces*, die zum Teil mit TinyOS mitgeliefert werden. Die von uns verwendete mitgelieferte Datei beinhaltet die *noise traces* der Meyer Library an der Stanford University [11]. Das Rauschen in dieser Umgebung ist hoch wegen vieler WLAN Access Points. Somit wird der Empfang der Pakete durch den *gain* zwischen dem Sender und dem Empfänger, die Interferenzen mit anderen Nachrichten, und durch das Umgebungsrauschen bestimmt.

Da sich das obige Modell zum Empfang von Datenpaketen erheblich von den Modellen in allen anderen von uns verwendeten Simulatoren unterscheidet, ist nicht klar, wie und ob in TOSSIM die Modellparameter so gewählt werden können, dass sie in etwa zum Beispiel denen von GloMoSim entsprechen. Außerdem kann man die Sendereichweite in TOSSIM im Gegensatz zu anderen Simulatoren nicht ändern, da diese Eigenschaft noch nicht implementiert wurde. In unseren Experimenten lag die Sendereichweite zwischen 10 und 18 Metern.

4.3 Formale Betrachtung

Die theoretische Analyse wird mit Markov-Ketten durchgeführt, wodurch sich die auftretenden Wahrscheinlichkeiten mit hoher Präzision (ohne Standardabweichung) berechnen lassen. Die Modellierung und Evaluation erfolgt mit Prism [6] für jede Topologie getrennt. Die berechneten Größen lassen sich als untere Schranke für die zu messenden Werte verstehen und betrachten den optimalen Fall, in dem Kollisionen, Sendewiederholungen und andere störende Effekte ausgeschlossen werden. Auf die Bit-Übertragungs- (*physical*) und die Sicherungsschicht (*data link*) wird bei der Modellierung verzichtet.

Jeder Knoten wird als Modul (Abbildung 2) mit jeweils einer lokalen Variable modelliert, die den jeweiligen Zustand für *Idle*, *Forward* und *Sleep* repräsentiert. Um die Modellkomplexität zu reduzieren² wird auf eine explizite Darstellung des Sender-/Empfängermodells verzichtet. Dennoch konnte für Topologie 5 keine Erreichbarkeit auf einem Rechner mit 32GB RAM wegen Speicherüberlauf berechnet werden. Die Modellierung sieht für die Kommunikation zwischen den Knoten das Setzen eines globalen Nachrichten-Bits vor, um so den Paketempfang zu betrachten.

Das so konstruierte Modell entspricht dem Konnektionsgraphen für die Topologien, wie sie Abbildung 1 zeigt.

²bei einer Netzgröße von 14 Knoten wächst das Modell auf $124 \cdot 10^6$ Zustände und $1,5 \cdot 10^9$ Transitionen

Hierzu wird keine Sendereichweite definiert, sondern die Konnektion aus der Topologievorgabe verwendet. Ein Paketverlustmodell wird nicht betrachtet, denn wie durch Experimente [2] gezeigt wurde, lässt sich bei einer Sendeleistung von $3,99 \cdot 10^{-17}$ mW ($-163,98$ dBm) eine *Bit Error Ratio* von 10^{-6} messen, die für die vorliegenden Experimente vernachlässigbar klein ist.

4.4 KSNSim

Der JAVA basierte *Karlsruhe Sensor Networking Simulator (KSNSim)* wurde mit dem Ziel entwickelt, Mechanismen zur relationalen Anfrageverarbeitung in Sensornetzen schnell und ohne Spezifikation von Netzwerk-, Energie- oder Betriebssystemparametern auf ihre Funktionstüchtigkeit zu testen. Ein weiteres Entwurfsziel bei der Entwicklung des Simulators ist die Schnittstellenkompatibilität mit dem auf den Sun SPOT [10] verwendeten Java SDK, so dass entwickelte Programme ohne Portierungsaufwand auch in einem realen Sensornetz aus Sun SPOT Knoten genutzt werden konnten. Ziel war es also nicht, drahtlose Kommunikation möglichst realitätsgetreu nachzubilden oder besonders ausgefeilte Kommunikationsmechanismen bereitzustellen.

Das Netzwerkmodell zur Simulation von drahtloser Kommunikation ist daher so einfach wie möglich: Sensoren werden an explizit definierten Punkten in einem virtuellen Raum platziert und haben eine Sendereichweite r . Als Kommunikationsmodell dient ein Unit-Disk-Graph mit Radius r . Alle Knoten, die einen Abstand größer als r zum Absender einer Nachricht haben, werden von der Kommunikation nicht beeinflusst.

Auf Grund der extrem einfachen Modellierung der Kommunikation ist es daher vor allem zu erwarten, dass die Anzahl der empfangenen Nachrichten überschätzt wird, da keinerlei Paketverlust simuliert wird. Diese erwartete Überschätzung der empfangenen Nachrichten bewirkt eine Überschätzung der Anzahl erreichter Knoten und des Energiebedarfes.

5. ZUSAMMENFASSUNG

In Abbildung 3 werden die Ergebnisse des Experiments zusammengefasst. Alle Diagramme verwenden die selbe Legende, die in Abbildung 3d dargestellt ist.

Abbildung 3a zeigt die Resultate des Energieverbrauchs, den die verschiedenen Werkzeuge für die ausgewählten Topologien errechnet haben. Trotz fehlender vollständiger Übereinstimmung, sind Parallelen erkennbar, etwa bei der Reihenfolge der Topologien bezüglich ihres Energieverbrauchs. Einem direkten Vergleich der Energieverbräuche stehen zudem die in einigen Werkzeugen fehlenden Möglichkeiten, Kollisionen und Paketwiederholungen mit einzubeziehen, entgegen.

Weitere Details sind durch Aufschlüsselung der Ergebnisse in versendete und empfangene Pakete in den Abbildungen 3b und 3c erkennbar. Die Unterschiede im Energieverbrauch sind demzufolge hauptsächlich auf Differenzen bei der Zahl der empfangenen Pakete zurückzuführen, was auf die teilweise unterschiedlichen Sendereichweiten (und damit veränderte Konnektivitätsgraphen) und Paketverlustmodelle zurückzuführen ist.

Letztendlich gibt es deutliche Unterschiede bei der Zahl der erreichten Knoten (Abbildung 3d), die neben den

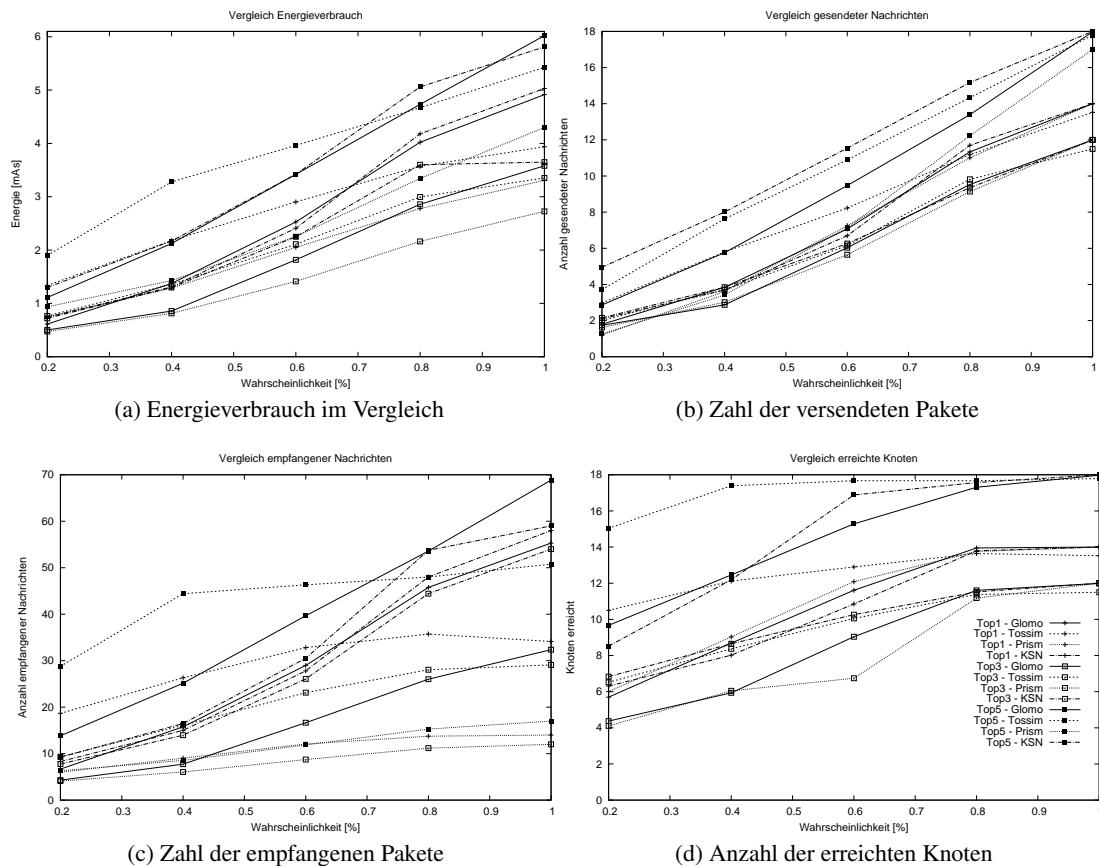


Figure 3: Evaluationsergebnisse

bisher genannten Ursachen auch auf starke Varianzen der Ergebnisse bei geringem p zurückzuführen ist.

Die Ergebnisse zeigen insgesamt, dass eine grundsätzliche Vergleichbarkeit schon jetzt gegeben ist. Deckungsgleiche Ergebnisse werden jedoch durch die stark unterschiedliche Flexibilität und Konfigurierbarkeit der verwendeten Werkzeuge sowie das stark vereinfachte Energiemodell verhindert.

Um realistische Werte des Energieverbrauchs ermitteln zu können, ist die Energieabschätzung zur Laufzeit und Energiemessungen der Knoten in einem Testbett [7] interessant. Das wird ermöglicht durch eine Erweiterung des verwendeten Betriebssystems (TinyOS) bzw. durch eigene Hardware zur Knotenverwaltung. Beide Ansätze werden gegenwärtig implementiert und in zukünftige Publikationen einfließen.

6. REFERENCES

- [1] Atmel Corporation. ATMEL ATmega128 datasheet. http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf, 2002.
- [2] E.-O. Blaß. *Sicherer, aggregierender Datentransport in drahtlosen Sensornetzen*. Dissertation, Universitätsverlag Karlsruhe, Karlsruhe, Germany, June 2007. ISBN: 978-3-86644-142-2.
- [3] E.-O. Blaß, L. Tiede, and M. Zitterbart. An energy-efficient and reliable mechanism for data transport in wireless sensor networks. In *Proceedings of International Conference on Networked Sensing Systems*, pages 211–216, Chicago, USA, May 2006. ISBN 0-9743611-3-5.
- [4] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of MANET simulators. In *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 38–43, New York, NY, USA, 2002. ACM.
- [5] Crossbow Inc. RADIO, RF concepts, and TOS radio stack, 2006. http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/PresentationOverheads/Day1_Sect06_RFConcepts.pdf.
- [6] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A Tool for Automatic Verification of Probabilistic Systems. In H. Hermanns and J. Palsberg, editors, *Proc. 12th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.
- [7] S. Kellner, M. Pink, D. Meier, and E.-O. Blaß. Towards a Realistic Energy Model for Wireless Sensor Networks. In *5th Ann. Conf. on Wireless on Demand Network Systems and Services (WONS 2008)*, pages 97–100, 2008.
- [8] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In *IPSN '07: Proc. 6th Int. Conf. on Inf. Processing in Sensor Networks*, pages 21–30, New York, NY, USA, 2007. ACM.
- [9] NI USB-6210 USB Multifunction DAQ. <http://sine.ni.com/nips/cds/view/p/lang/de/nid/203189>.
- [10] SUN Microsystems Inc., Small Programmable Object Technology (SPOT).
- [11] TinyOS Tutorial. Lesson 11: TOSSIM. <http://docs.tinyos.net>.
- [12] M. Zuniga. Building a network topology for TOSSIM. <http://www.tinyos.net>.