Technische Universität Braunschweig

Reliable End-to-End Data Transmission in Wireless Sensor Networks

Von der Carl-Friedrich-Gauß-Fakultät

der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines

Doktoringenieurs (Dr.-Ing.)

genehmigte Dissertation
(kumulative Arbeit)

von **Wolf-Bastian Pöttner**
geboren am **09.05.1983**
in **Wolfsburg**

Eingereicht am: **17.12.2013**
Disputation am: **19.03.2014**
1. Referent: **Prof. Dr.-Ing. Lars Wolf**
2. Referent: **Prof. Cormac J. Sreenan**

(2014)

»You can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future.« Steve Jobs

Acknowledgements

I thought long and hard about ways to show gratitude to the people who supported me while working on this dissertation. The truth is, that I am unable to write anything that is even remotely expressing how important the friendship, love, patience and loyalty of my partner, family, friends, peers and colleagues is to me. Thank you all very much!

Kurzfassung

Drahtlose Sensornetzwerke (englisch: "Wireless Sensor Networks", "WSNs") werden bereits heute in einer Vielzahl von Anwendungen eingesetzt und auch in Zukunft wird die Zahl der Anwendungen für derartige Netzwerke zunehmen. Allerdings sind WSNs typischerweise weniger zuverlässig als die drahtgebundenen Netzwerke des täglichen Lebens. Dies wird insbesondere durch preisgünstige Komponenten, Funkschnittstellen mit geringem Energieverbrauch (und entsprechend niedriger Reichweite) sowie begrenzen Energiereserven verursacht. Eine steigende Zahl von Anwendungen insbesondere im Hinblick auf industrielle Szenarien setzen jedoch Netzwerke mit hoher Zuverlässigkeit voraus. In der vorliegenden Arbeit wird daher das Ungleichgewicht zwischen verfügbarer Netzwerktechnologie und Anwendungsanforderungen untersucht und Lösungsansätze präsentiert. Um das breite Spektrum der möglichen Anwendungen bestmöglich abzudecken, werden zwei scheinbar gegensätzliche Netzwerktypen untersucht; beiden gemein ist jedoch die Technologie der drahtlosen Sensornetzwerke.

So genannte echtzeitfähige drahtlose Sensornetzwerke (englisch: "real-time WSNs") sind in der Lage der Anwendung gewisse Garantien im Hinblick auf die Zuverlässigkeit und Pünktlichkeit der Datenübertragung zu geben. Aus diesem Grund können die genannten Netze gut in industriellen Anwendungen wie Prozessüberwachung sowie Steuer- und Regelkreisen eingesetzt werden. In der vorliegenden Arbeit wird der Entwurf einer Architektur für echtzeitfähige drahtlose Sensornetzwerke basierend auf WSN-Komponenten gezeigt. Das Netz verwendet dabei einen TDMA-basierten Medienzugriff mit fest zugeordneten Zeitschlitzen. Weiterhin werden Mechanismen gezeigt, um die Zeitschlitzzuordnungen (Zeitpläne) programmatisch zu erzeugen. Die berechneten Zeitpläne sind in der Lage mit einer vordefinierten Unzuverlässigkeit der zugrundeliegenden drahtlosen Verbindung zu Recht zu kommen. Darüber hinaus wird in der Arbeit ein Ansatz diskutiert mit der die Sendeleistung so angepasst wird, dass Interferenz zwischen zwei Netzen verringert werden kann. Dank der starken Basis durch das EU FP7 Projekt GINSENG können Ergebnisse aus einer Testinstallation in einer produktiven industriellen Anlage in Portugal gezeigt werden. Weiterhin wurden Experimente in Bürogebäuden sowie Laboratorien durchgeführt. Der Entwurf der Testinstallation in Portugal wird näher beleuchtet und praktische Erfahrungen werden geschildert. Die Ergebnisse belegen, dass die programmatisch erzeugten Zeitpläne die Zuverlässigkeit signifikant erhöhen und gleichzeitig die Interferenzen deutlich reduzieren. Sie belegen weiter, dass die Anpassung der Sendeleistung auf Änderungen der Eigenschaften der drahtlosen Verbindung reagiert um die Interferenzen so weiter senken.

Im Gegensatz zu den vorgenannten Netzen zeigen verzögerungstolerante drahtlose Sensornetzwerke (englisch: "delay tolerant WSNs") potentiell eine hohe Verzögerung und die darauf aufbauenden Anwendungen müssen in der Lage sein mit dieser umzugehen. Zuverlässige Datenübertragung ist in solchen Netzwerken sichergestellt, solange die zur Verfügung stehende Netzwerkkapazität ausreicht. Zu diesem Zweck wird in dieser Arbeit die Technologie von verzögerungstoleranten Netzen auf drahtlose Sensornetzwerke übertragen und angepasst. Die Ausarbeitung zeigt die prinzipielle Machbarkeit dieses Ansatzes und präsentiert eine Umsetzung für WSN-Komponenten. Ebenfalls wird das passende Gegenstück auf dem PC gezeigt, um eine nahtlose Integration von verzögerungstoleranten drahtlosen Sensornetzwerken mit herkömmlichen verzögerungstoleranten Netzwerken zu ermöglichen. Auf dieser Basis werden Kommunikationsparadigmen mit niedrigen Kosten im Bezug auf die übertragenen Protokollinformationen präsentiert und experimentell verifiziert. Die Leistung eines verzögerungstoleranten drahtlosen Sensornetzwerks wird exemplarischen anhand einer Beispielanwendung gezeigt.

Zusammenfassend zeigt die vorliegende kumulative Dissertation, dass zuverlässige Datenübertragung sowohl in echtzeitfähigen als auch in verzögerungstoleranten drahtlosen Sensornetzwerken sichergestellt werden kann, wenn einige vertretbare Annahmen erfüllt werden können.

Abstract

Wireless Sensor Networks (WSNs) are used in a variety of applications today and will be used in even more applications in the future. However, WSNs tend to be less reliable than the wired networks we commonly work with mainly due to cheap hardware, low-power radios (with limited range) and limited energy supplies. An increasing number of applications especially in the context of industrial scenarios require a highly reliable network. In this thesis, the mismatch between available network technology and application demand is investigated and solution approaches are given. To cover the wide spectrum of possible applications as good as possible, two seemingly opposed types of networks are examined; both share the foundation of WSN.

So-called real-time wireless sensor networks guarantee certain reliability and timeliness of transported messages and are well suited for industrial applications such as plant monitoring and closed loop control. This thesis outlines the design of a real-time networking architecture relying on WSN hardware and TDMA medium access control with fixed slot assignments. Furthermore, mechanisms to programmatically create said TDMA schedules are presented, wherein the schedules can cope with a predefined level of unreliability for the underlying wireless links. Also, mechanisms to adapt the transmission power are discussed to reduce the inter-network interference of neighbouring networks. Based on the strong foundation of the EU FP7 project GINSENG, experimental results from a testbed in a live industrial facility in Portugal in addition to more results from office environments and laboratory experiments are presented. Insight into the design of said industrial testbed including experiences when working in such an environment are outlined as well. The results show that programmatically determined TDMA schedules significantly improve reliability while reducing interference considerably. The results also show that adapting the transmission power to changes of the wireless link further reduces interference.

On the contrary, delay-tolerant wireless sensor networks potentially exhibit a high delay and applications have to be able to handle those. Reliability in said networks is ensured, if network capacity allows. For this purpose, DTN technology known from PC-grade implementations is adapted to WSN networks. In this thesis, the feasibility of said approach is demonstrated and implementations for WSN nodes as well as appropriate adapters for the PC are shown to enable seamless integration between delay-tolerant wireless sensor networks and common delay-tolerant networks. On this basis, the necessary low-overhead communication paradigms are presented and evaluated. The performance of a delay-tolerant wireless sensor network as well as its network capacity is shown using an exemplary use case.

The cumulative dissertation at hand shows that reliability can be achieved in real-time and in delay-tolerant wireless sensor networks given a number of reasonable assumptions.

List of Publications

2014

- [1] Wolf-Bastian Pöttner, Hans Seidel, James Brown, Utz Roedig, and Lars Wolf. Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3), August 2014. accepted for publication

2013

- [2] Tony O'Donovan, James Brown, Felix Büsching, Alberto Cardoso, José Cecilio, Jose Manuel do Ó, Pedro Furtado, Paulo Gil, Anja Jugel, Wolf-Bastian Pöttner, Utz Roedig, Jorge Sa Silva, Ricardo Silva, Cormac J. Sreenan, Vasos Vassiliou, Thiemo Voigt, Lars Wolf, and Zinon Zinonos. The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences. *ACM Transactions on Sensor Networks (TOSN)*, 10(1), November 2013

- [3] Wolf-Bastian Pöttner, Daniel Willmann, Felix Büsching, and Lars Wolf. All Eyes on Code: Using Call Graphs for WSN Software Optimization. In *Eighth IEEE Workshop on Practical Issues in Building Sensor Network Applications 2013 (IEEE SenseApp 2013)*, Sydney, Australia, October 2013

- [4] Wolf-Bastian Pöttner, Daniel Willmann, Felix Büsching, and Lars Wolf. Regression Testing Framework for WSNs. In *Demonstrations of the IEEE Conference on Local Computer Networks 2013 (LCN-Demos 2013)*, Sydney, Australia, October 2013

- [5] Sebastian Schildt, Wolf-Bastian Pöttner, Felix Büsching, and Lars Wolf. RATFAT: ReAl-Time FAT for Cooperative Multitasking Environments in WSNs. In *5th Workshop on Performance Control in Wireless Sensor Networks 2013 (PWSN 2013) in conjunction with IEEE DCOSS 2013*, Cambridge, USA, May 2013

- [6] Wolf-Bastian Pöttner and Lars Wolf. Probe-based Transmission Power Control for Dependable Wireless Sensor Networks. In *The 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013 (IEEE DCoSS 2013)*, Cambridge, USA, May 2013

- [7] Ulf Kulau, Felix Büsching, Wolf-Bastian Pöttner, and Lars Wolf. Demo: A Node's Life - Increasing WSN Lifetime by Dynamic Voltage Scaling. In *The 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013 (IEEE DCoSS 2013)*, Cambridge, USA, May 2013

2012

- [8] Wolf-Bastian Pöttner, Felix Büsching, Georg von Zengen, and Lars Wolf. Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks. In *The Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2012)*, Las Vegas, Nevada, USA, October 2012

- [9] Georg von Zengen, Felix Büsching, Wolf-Bastian Pöttner, and Lars Wolf. An Overview of μDTN: Unifying DTNs and WSNs. In *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, September 2012

- [10] Felix Büsching, Wolf-Bastian Pöttner, Dieter Brökelmann, Georg von Zengen, Robert Hartung, Karsten Hinz, and Lars Wolf. A Demonstrator of the GINSENG-Approach to Performance and Closed Loop Control in WSNs. In *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, June 2012. Best Demo Award

- [11] Sebastian Schildt, Wolf-Bastian Pöttner, and Lars Wolf. Contiki Ring File System for Real-Time Applications. In *4th Workshop on Performance Control in Wireless Sensor Networks 2012 (PWSN 2012) in conjuction with IEEE DCOSS 2012*, pages 364 –371, May 2012

- [12] Sebastian Schildt, Wolf-Bastian Pöttner, Oliver Ohneiser, and Lars Wolf. NASDI - Naming and Service Discovery for DTNs in Internet Backbones. In *Proceedings of the 5th International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, Mobilware 2012*, 2012

- [13] Sebastian Schildt, Till Lorentzen, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. Free-riding the BitTorrent DHT to improve DTN connectivity. In *Proceedings of the seventh ACM international workshop on Challenged networks*, CHANTS '12, pages 9–16. ACM, 2012

- [14] Wolf-Bastian Pöttner and Lars Wolf. Flow Control Mechanisms for the Bundle Protocol for the Bundle Protocol in IEEE 802.15.4 Low-power Networks. In *Proceedings of the seventh ACM international workshop on Challenged networks (CHANTS)*, CHANTS '12, pages 65–68. ACM, 2012

2011

- [15] Wolf-Bastian Pöttner, Sebastian Schildt, Daniel Meyer, and Lars Wolf. Piggy-Backing Link Quality Measurements to IEEE 802.15.4 Acknowledgements. In *The 4th International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN-Fall 2011)*, Valencia, Spain, October 2011

- [16] Wolf-Bastian Pöttner, Johannes Morgenroth, Sebastian Schildt, and Lars Wolf. An Empirical Performance Comparison of DTN Bundle Protocol Implementations. Informatikbericht 2011-08, Technische Universität Braunschweig, September 2011

- [17] Wolf-Bastian Pöttner, Johannes Morgenroth, Sebastian Schildt, and Lars Wolf. An Empirical Performance Comparison of DTN Bundle Protocol Implementations. In *ACM MobiCom 2011 Workshop on Challenged Networks (CHANTS'11)*, Las Vegas, Nevada, USA, September 2011

- [18] Wolf-Bastian Pöttner, Lars Wolf, José Cecilio, Pedro Furtado, Ricardo Silva, Jorge Sa Silva, Amancio Santos, Paulo Gil, Alberto Cardoso, Zinon Zinonos, Jose Manuel do Ó, Ben McCarthy, James Brown, Utz Roedig, Tony O'Donovan, Cormac J. Sreenan, Zhitao He, Thiemo Voigt, and Anja Jugel. WSN Evaluation in Industrial Environments First results and lessons learned. In *3rd Workshop on Performance Control in Wireless Sensor Networks 2011 (PWSN 2011) in conjuction with IEEE DCOSS 2011*, June 2011

- [19] Sebastian Schildt, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. IBR-DTN: A lightweight, modular and highly portable Bundle Protocol Implementation. *Electronic Communications of the EASST*, 37:1–11, January 2011

- [20] Michael Doering, Wolf-Bastian Pöttner, Tobias Pögel, and Lars Wolf. Impact of Radio Range on Contact Characteristics in Bus-based Delay Tolerant Networks. In *Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS 2011)*, pages 195–202, Bardonecchia, Italy, January 2011

2010

- [21] Michael Doering, Tobias Pögel, Wolf-Bastian Pöttner, and Lars Wolf. A New Mobility Trace for Realistic Large-Scale Simulation of Bus-based DTNs. In *ACM MobiCom 2010 Workshop on Challenged Networks (CHANTS 2010)*, Chicago, USA, September 2010
- [22] Wolf-Bastian Pöttner, Oliver Wellnitz, and Lars Wolf. QoS-AODV6E: An Energy-Balancing QoS Routing Scheme for WSNs. In *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS2010)*, Kassel, Germany, June 2010
- [23] Wolf-Bastian Pöttner and Lars Wolf. IEEE 802.15.4 Packet Analysis with Wireshark and off-the-shelf Hardware. In *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS2010)*, Kassel, Germany, June 2010

2007

- [24] Sven Lahde, Michael Doering, Wolf-Bastian Pöttner, Gerrit Lammert, and Lars Wolf. A Practical Analysis of Communication Characteristics for Mobile and Distributed Pollution Measurements on the Road. *Wireless Communications and Mobile Computing*, 7(10):1209–1218, December 2007

2006

- [25] Sven Lahde, Michael Doering, Wolf-Bastian Pöttner, and Gerrit Lammert. Mobile and Distributed Measurement of Air Pollution in Metropolitan Areas Using Car2X Techniques. In *Proceedings of 3nd Symposium on Informationssysteme für mobile Anwendungen (IMA)*, Braunschweig, October 2006

Contents

1 Introduction

Wireless Sensor Networks (WSNs) are wireless networks consisting of WSN nodes that often operate with limited energy resources such as batteries. WSN nodes are designed around a Microcontroller Unit (MCU) and a wireless transceiver and often feature sensors and storage capabilities. WSNs are usually deployed to serve a specific goal such as monitoring the environment, measuring the structural integrity of buildings or bridges, keeping an eye on forests to allow early detection of wood fires or monitoring industrial facilities. WSNs are well suited for those kinds of applications because of their energy efficient operation, which results in a long lifetime on batteries as well as the cheap hardware allowing the installation of numerous nodes. Additionally the wireless communication capabilities of WSNs are well-suited for emerging and ad-hoc applications in which installation of infrastructure is either too expensive or not possible at all.

Unfortunately, a commonality of many publications in the WSN context is the assumption of rather unreliable networks. WSNs are often considered to be comparably unreliable [26, 27] mainly due to three facts:

1. Wireless links are inherently unreliable and even more so with low-power transmitters found in today's WSN hardware.

2. WSN hardware is cheap and is assumed to have a high hardware failure rate leading to node outages.

3. Limited energy supplies lead to the problem of nodes eventually dying due to depleted batteries.

However, many applications require a certain level of reliability when it comes to data transmission in the network. To investigate and eventually overcome this issue, a significant amount of publications exist that are concerned with robust networking mechanisms handling the inherent unreliability that we find in today's WSNs. While the definition of reliability cannot be generalized, in the networking context reliability is often defined as the fraction of data packets that are delivered in contrast to those that have been sent. In networks in which data packets have an associated deadline after which the content is worthless, the definition of reliability is often extended to only count data packets that have been delivered before the deadline has elapsed. Mechanisms for reliable end-to-end data transmission in the following two specific classes of WSNs are presented, discussed and evaluated in this thesis: Real-Time WSNs primarily for industrial process automation and delay-tolerant WSNs primarily for environmental data collection.

Real-time networks can be used for industrial applications, as they are able to provide hard guarantees regarding reliability (and closely connected also timeliness). Such networks are generally based on Time Division Multiple Access (TDMA) technology: Time is divided into timeslots and slots are allocated to specific senders and receivers. This allows giving hard guarantees as to when a node will be able to access the shared wireless medium, which is a necessary prerequisite for real-time network operation. Usually, a sequence of TDMA slots is periodically repeated and is referred to as schedule. However, to deal with lost packets opportunities for retransmissions have to be incorporated into the TDMA schedule. This cumulative dissertation discusses results stemming from the context of the real-time WSN research project GINSENG including a testbed in a live oil refinery in Portugal. Building on top of GINSENG's results, two mechanisms are presented and evaluated that aim at ensuring reliable data transmission. 1) The network topology is programatically provisioned to use reliable links. The resulting computed TDMA schedule allows for retransmissions if packets get lost.

This ensures packet delivery even in harsh industrial environments. 2) Coexistence with neighbouring networks as often found in industrial settings is made possible by reducing inter-network interference. Due to the large number of nodes that are installed in close proximity in industry environments, reducing inter-network interference is an important factor to further improve reliability.

In contrast, delay-tolerant networks cannot give any guarantees regarding delay but provide best-effort data forwarding and delivery. These networks enable data delivery in situations, in which traditional network protocols fail to deliver any data. Delay Tolerant Networks (DTNs) achieve this by temporarily storing data packets on nodes in the network to overcome intermittent connectivity and unreliability of links. DTNs have the potential to be inherently reliable if the capacity of the network in terms of storage and link capacities is not exceeded. As a significant amount of WSN deployments are used to collect long-term statistical data, delay is of minor importance but reliability is a concern. To deal with these kinds of applications, this thesis outlines how the paradigms of DTNs can be applied to WSNs. On the one hand this allows for reliable network operation while on the other hand using standard DTN technology further allows for interoperability with existing DTNs. Interoperable networks enable seamless integration of delay-tolerant WSNs and DTNs.

The two network types mentioned above represent two very different classes of networks: One class providing hard real-time guarantees (from the class of real-time WSNs) while the other class is a delay-tolerant network (from the class of best-effort WSNs). While real-time and delay-tolerant networks are only examples, most WSN applications can be mapped into either of these two categories. Applications that cannot be mapped into one of those two classes are often based on best-effort networks but still have certain expectations towards the network delay and are out of the scope of this thesis. This thesis shows that reliable end-to-end data transmission can be ensured for both network classes given some assumptions that are outlined within the document. However, although both networks are realized using standard WSN hardware, the employed network and software architecture are quite different.

1.1 Publications

The cumulative dissertation at hand makes several contributions in the area of WSN research including novel approaches for TDMA schedule dimensioning among others. The individual contributions are contained in the following 11 scientific publications both in conference proceedings and journals. Those 11 publications are contained in this document and are listed (with their respective page number) in the following:

1. Tony O'Donovan, James Brown, Felix Büsching, Alberto Cardoso, José Cecilio, Jose Manuel do Ó, Pedro Furtado, Paulo Gil, Anja Jugel, Wolf-Bastian Pöttner, Utz Roedig, Jorge Sa Silva, Ricardo Silva, Cormac J. Sreenan, Vasos Vassiliou, Thiemo Voigt, Lars Wolf, and Zinon Zinonos. The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences. *ACM Transactions on Sensor Networks (TOSN)*, 10(1), November 2013 (on page 49)

2. Wolf-Bastian Pöttner, Lars Wolf, José Cecilio, Pedro Furtado, Ricardo Silva, Jorge Sa Silva, Amancio Santos, Paulo Gil, Alberto Cardoso, Zinon Zinonos, Jose Manuel do Ó, Ben McCarthy, James Brown, Utz Roedig, Tony O'Donovan, Cormac J. Sreenan, Zhitao He, Thiemo Voigt, and Anja Jugel. WSN Evaluation in Industrial Environments First results and lessons learned. In *3rd Workshop on Performance Control in Wireless Sensor Networks 2011 (PWSN 2011) in conjuction with IEEE DCOSS 2011*, June 2011 (on page 51)

3. Wolf-Bastian Pöttner, Hans Seidel, James Brown, Utz Roedig, and Lars Wolf. Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3), August 2014. accepted for publication (on page 53)

4. Sebastian Schildt, Wolf-Bastian Pöttner, and Lars Wolf. Contiki Ring File System for Real-Time Applications. In *4th Workshop on Performance Control in Wireless Sensor Networks 2012 (PWSN 2012) in conjuction with IEEE DCOSS 2012*, pages 364 –371, May 2012 (on page 55)

5. Sebastian Schildt, Wolf-Bastian Pöttner, Felix Büsching, and Lars Wolf. RATFAT: ReAl-Time FAT for Cooperative Multitasking Environments in WSNs. In *5th Workshop on Performance Control in Wireless Sensor Networks 2013 (PWSN 2013) in conjunction with IEEE DCOSS 2013*, Cambridge, USA, May 2013 (on page 57)

6. Wolf-Bastian Pöttner and Lars Wolf. Probe-based Transmission Power Control for Dependable Wireless Sensor Networks. In *The 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013 (IEEE DCoSS 2013)*, Cambridge, USA, May 2013 (on page 59)

7. Wolf-Bastian Pöttner, Sebastian Schildt, Daniel Meyer, and Lars Wolf. Piggy-Backing Link Quality Measurements to IEEE 802.15.4 Acknowledgements. In *The 4th International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN-Fall 2011)*, Valencia, Spain, October 2011 (on page 61)

8. Felix Büsching, Wolf-Bastian Pöttner, Dieter Brökelmann, Georg von Zengen, Robert Hartung, Karsten Hinz, and Lars Wolf. A Demonstrator of the GINSENG-Approach to Performance and Closed Loop Control in WSNs. In *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, June 2012. Best Demo Award (on page 63)

9. Wolf-Bastian Pöttner, Felix Büsching, Georg von Zengen, and Lars Wolf. Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks. In *The Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2012)*, Las Vegas, Nevada, USA, October 2012 (on page 65)

10. Wolf-Bastian Pöttner and Lars Wolf. Flow Control Mechanisms for the Bundle Protocol for the Bundle Protocol in IEEE 802.15.4 Low-power Networks. In *Proceedings of the seventh ACM international workshop on Challenged networks (CHANTS)*, CHANTS '12, pages 65–68. ACM, 2012 (on page 67)

11. Georg von Zengen, Felix Büsching, Wolf-Bastian Pöttner, and Lars Wolf. An Overview of μDTN: Unifying DTNs and WSNs. In *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, September 2012 (on page 69)

1.2 Contributions

The contribution of this cumulative dissertation in general and of the research papers stated in the previous section in particular can be summarized as follows:

- **Programmatically determined TDMA schedules for real-time networks**: The first contribution is an approach for computing TDMA schedules based on application requirements and link properties while ensuring timely and reliable delivery of data and reducing inter-network interference at the same time. This includes a mechanism for data gathering in the field and subsequent processing. Based on this information an algorithm is discussed to calculate a TDMA schedule that takes link reliability into account and reduces inter-network interference significantly. To make this computationally feasible, a heuristic is presented. The approach is verified using experiments in industrial as well as office environments with results given in the thesis and in the related publications.

- **Transmission Power Control for real-time networks**: The second contribution is an online transmission power control algorithm that lowers the transmission power to the necessary minimum while keeping the link reliability within certain bounds. The presented approach is a transmission power control algorithm based on receiver signal strength feedback that adjusts transmission power to meet a specific receiver signal strength target. Furthermore, a probing mechanism is used to determine the target. Additionally, a feedback scheme is discussed to feed the receiver signal strength back to the sender without incurring overhead. The approaches are experimentally verified in controlled environments and results are discussed in the thesis and in the related publications.

- **WSN testbed support network in a live industrial facility**: The third contribution is a support network for the GINSENG testbed within a live industrial facility in Portugal that allows to remotely access and use the testbed for experiments. The approach contains detailed descriptions of the infrastructure that is needed for remote experimentation in a live oil refinery and further includes deployment experiences. In addition, mechanisms such as connecting the real-time WSN to the back end network over a serial connection and mechanisms to locate performance bottlenecks in the system are described within the thesis as well.

- **Real-time file systems for cooperative multitasking environments**: The fourth contribution are mechanisms to allow for large-scale long-term data storage on WSN nodes running a cooperative multitasking operating system with a real-time application. The thesis contains two radically different approaches for data storage either in the on-board flash of nodes or on Secure Digital (SD) cards. The unique characteristic of the approaches is the compatibility with real-time tasks even in cooperative multitasking environments. The file system approaches are experimentally verified and results are given in the thesis and in the related publications.

- **Bundle Protocol Convergence Layer for IEEE 802.15.4-based networks**: The fifth contribution is an analysis and specification of how the de-facto standard protocol in delay-tolerant networking can be used on resource-constrained WSN nodes. The contribution includes experimental evidence that today's WSN nodes can handle the Bundle Protocol (BP). The performance of several flow control mechanisms for the Convergence Layer (CL) is experimentally evaluated and the overhead of the BP is compared to IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN). Results are given in this thesis and in the related publications.

- **Hardware-independent Bundle Protocol implementation for WSN nodes**: The sixth contribution is an implementation of the de-facto standard protocol in delay-tolerant networking that is based on Contiki OS [28] and can be used on a variety of hardware platforms including the Cooja simulator. The thesis outlines the software design of the implementation and provides insight into the expected performance. The evaluation in a delay-tolerant real-world experiment in an office environment is presented in this thesis and the related publications.

1.3 Outline

This extended overview of the cumulative dissertation is meant to be a summary of the publications listed in section 5.1. A full list of publications by the author is given on page *xi* of this document. Those publications that are considered an integral part of this work are contained in the dissertation and start on page 47. The thesis contains explicit references to those papers to highlight their context and the relation between the publications.

The remainder of this thesis is structured as follows: Section 2 gives an overview over the terminology used in this thesis and gives a more formal definition of WSNs, WSN nodes, reliability and timeliness. In the following Section 3 real-time WSNs are introduced and a use-case is presented. Furthermore, measures to ensure reliable data transmission are introduced, discussed and evaluated.

In Section 4 delay-tolerant WSNs are introduced using an exemplary use-case. Unique challenges are discussed and mechanisms to ensure reliability are presented. Before the research papers are presented starting on page 47, a summary of this extended overview is given in Section 5.

2 Terminology

WSNs are part of an active area of research. In the early days of WSN research, energy efficient hardware and algorithms that help to reduce the energy consumption of the hardware were the primary concern. This lead to a plethora of publications about Medium Access Control (MAC) mechanisms and to various different node architecture approaches. However, the focus of research has moved on to higher-level network protocols and interconnection between different network types. The Internet of Things (IoT) is the new buzzword, thereby combining WSN hard- and software to a network that is fully integrated into the Internet.

Due to the fast pace of innovation in the WSN context, finding fixed and exact definitions for certain terms is impossible. To pave common ground, this chapter gives definitions for certain terms used throughout this thesis and the appended publications. Those definitions are by no means complete or conclusive and hence are limited to the scope of this thesis.

2.1 Wireless Sensor Network

WSNs are groups of at least two stations. In the literature, references of several hundreds or even thousands stations can be found, whereas such deployments are rare in practice. In the WSN context, stations are referred to as nodes. Nodes in a WSN collaborate to achieve a common goal and form a network. For greater spatial coverage and freedom of the nodes, wireless communication is used for the links between the nodes. Many WSNs sense data in their environment and transport said data to one or multiple sinks in the network. Sinks are often regular nodes that are connected to a Personal Computer (PC) via the serial port (or a Universal Serial Bus (USB) connection) for data retrieval. Sinks may also be more powerful nodes that combine the wireless communication capabilities of a regular node with the computational capabilities of a PC.

Many WSNs in the literature use multi-hop data delivery in which data is not delivered within a single hop but routed over multiple nodes in between. This allows geographical coverage to exceed the wireless communication range of a single node. Furthermore, some publications assume WSNs to be self-organizing in nature. Others claim a fixed and sometimes pre-defined physical and logical topology. Routing is generally an application-specific problem. Routing Protocol for Low power and Lossy Networks (RPL) [29] is the new de-facto standard in WSN routing when running the 6LoWPAN [30] IP stack.

2.2 WSN Nodes

Typical WSN nodes are computing platforms designed for this very purpose. Due to the expected high number of nodes deployed in a specific setting, nodes are designed to be cheap. Typically, a sensor node is composed of the following units:

- *Processing Unit*: a microprocessor that is able to execute binary code and is often referred to as Central Processing Unit (CPU). The CPU also contains volatile Random-access Memory (RAM).

- *Wireless Communication Unit*: a wireless transceiver that allows sending and receiving transmissions.

- *Sensor Unit*: a sensor allows collecting analog information about the environment and provides a digital representation of said phenomenon. Analogue sensors are digitized using Analog-to-Digital-Converters (ADCs).

- *Storage Unit*: means of storing arbitrary information on the nodes. This includes the program in persistent Read-only Memory (ROM) and potentially additional flash memory.

- *Energy Supply*: provides power to the different units of the node.

Nodes are expected to run unattended for extended time periods of multiple days, weeks, months or even years. With energy supplies often provided by batteries, these long lifetimes can only be realized when using low-power hardware and appropriate software. Due to the low price, WSN nodes are assumed to show rather high hardware failure rates leading to either a partial or full malfunction of the node.

2.2.1 State of the Art

On almost all WSN platforms that are widely used at the moment the CPU is a low-power microcontroller referred to as MCU. Several products are on the market with differences in maximum clock rate, energy consumption, ROM and RAM. Newer nodes are equipped with more powerful and energy efficient ARM microprocessors. In general, the available computing power on nodes is significantly below a standard PC or smartphone. Commercial wireless transceivers are often packet-based radios conforming to IEEE 802.15.4 [31]. Packet-based radios realize the Physical (PHY) as well as parts of the MAC layer on the chip and expose a command set that allows transmitting and receiving full IEEE 802.15.4 radio frames. Older, stream-based radios left more processing to the CPU. Most chips operate in the 2.4 GHz band of the IEEE 802.15.4 specification and some offer a range of over 200 m [32]. In most deployments, standard batteries are the primary energy source. Some deployments augment this with solar-power [33] to enhance the lifetime of the batteries. Many nodes feature on-board flash chips to allow for permanent, large-scale data storage. Sizes of several megabytes are the order whereas some nodes feature SD card slots to further enhance the available space.

The Tmote Sky [34] is a well-known WSN node based on a 16 bit Texas instruments (TI) MCU with 48 kB ROM and 10 kB RAM. The node can be clocked at up to 8 MHz and features the TI CC2420 [35] packet-based IEEE 802.15.4 radio. The node has several sensors and 1 MB of serial data flash on-board.

Inexpensive Node for General Applications (INGA) [32] is a more recent WSN node development featuring an 8 bit Atmel MCU with 128 kB ROM, 16 kB RAM that can be clocked at up to 16 MHz. The node features the Atmel AT86RF231 [36] packet-based IEEE 802.15.4 radio, 2 MB of on-board flash as well as several sensors.

In the remainder of this thesis, the Tmote Sky as well as INGA are used as WSN reference platforms.

2.3 Reliability

The definition of *Reliability* depends on a specific application and context and cannot be generalized. However, in networking reliability often refers to the fraction of unique data that is delivered on a given path (or connection) through a network in its original state (without being corrupted). If all data is delivered, the reliability of this particular path is 100 %. Corrupted or lost data reduces the reliability. In WSNs, data is transported in frames or packets, whereas both terms are used interchangeably in the thesis at hand. The Packet Reception Rate (PRR) (as well as Packet Delivery Ratio (PDR)) is defined as the fraction of delivered packets compared to the total packets as indicated in equation 2.1.

In this context, unique packets are the number of outgoing packets whereas retransmissions of the same packet count only once. Similarly, the receiver counts multiple copies of incoming packets only once as well. The Packet Loss Rate (PLR) is the inverse of the PRR.

$$PRR = \frac{\text{Number of unique, unmodified packets received}}{\text{Number of unique packets sent}} \tag{2.1}$$

On a given link in a wireless network, the wireless channel in terms of frame loss or corruption in the air primarily influences the PRR. We refer to this kind of reliability as "hop-by-hop reliability". In a multi-hop network, multiple links are concatenated and the chances for packet loss add up. In this case, we define "end-to-end reliability" as the PRR of all data packets between sender and receiver. Furthermore, effects of higher layers such as queuing effects come into play. The application usually cannot distinguish between different reasons for packet loss. Most applications require a certain minimum reliability (often specified as minimum PRR) in order to work properly. Too high packet loss in the network may lead to malfunction of the application or even to unpredictable behaviour.

Certain network protocols send multiple copies of packets on multiple paths through the network to increase the likelihood of successful packet delivery. By using independent paths, potentially devastating events such as the premature death of a node can be circumvented. In such networks, a packet counts as delivered when the first copy of the packet arrives at the destination. Copies arriving at a later point in time do not have an influence on the PRR.

Definition 2.1. Reliability in WSNs is the ratio of unique, non-corrupted data packets that are delivered to the destination.

2.3.1 Timeliness

Certain applications can only make use of data, when said data is fresh enough. Old data is possibly outdated and may not be of any use. An example is a closed loop controller that cannot act on data that is too old. Delay is usually defined as the time between sending a packet and its delivery to the destination. In this thesis, end-to-end delay in the WSN always refers to application-layer delay. In a network, this time is typically dominated by the forwarding delay within the network. In Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) based wireless networks, a significant and unpredictable amount of time is needed for channel access because multiple nodes are competing for the channel.

An application that requires end-to-end data delivery within a certain maximum delay will count packets that arrive after the deadline as packet loss. For such time critical applications, the definition of reliability is extended to:

Definition 2.2. Reliability in real-time WSNs is the fraction of unique, non-corrupted data packets that are delivered to the destination within the deadline.

2.4 WSN Traffic Classes

In a network, traffic can be classified - among other things - by the required timeliness of the transported data. As we have seen before, certain applications require data to arrive before a certain deadline; otherwise the transported data becomes useless. Such real-time applications can be further subdivided into two classes. Applications with hard real-time requirements tolerate no or very little late packets and rely on the network to deliver a high fraction of data in time. Application scenarios include industrial monitoring and control but can also be found in various other areas. Applications with soft real-time requirements are able to tolerate a higher fraction of packets to be lost or to arrive late. While the service provided by the application may be degraded during loss periods, it can still fulfil its core duties. Examples are forest fire or structural health monitoring. If individual samples get lost or arrive late, such applications continue to work.

However, in WSNs many network stacks are not able to give any guarantees regarding timeliness of data. This often incurs that the applications that are realized on top either have no requirements towards timeliness or try to manage problems connected to late data on their own. In general we can say that applications building on top of network stacks offering no guarantees - so-called best-effort networks - must be able to handle unpredictable network latencies. Hence those types of applications can be called delay-tolerant. In practice, a number of applications building on top of best-effort networks assume that data will not take “too long” to be delivered. However, “too long” is not defined and seems to be based on experience rather than guarantees. Those applications may stop working if data takes too long - however, this has to be considered a design fault of the application itself. For this reason, those applications and network types are out of scope of this thesis.

In the scope of this thesis, the following differentiation is used: Either traffic is real-time traffic with hard or soft requirements, or the traffic is best-effort traffic and by definition delay-tolerant.

3 Real-Time Wireless Sensor Networks

WSNs can be used in a multitude of applications in various scenarios and some of those require the network to fulfil real-time requirements. Especially in applications where data loss (or data arriving late) causes potentially hazardous consequences, real-time guarantees are necessary. This applies for example to forest fire detection [37], monitoring of dangerous goods [38], target tracking [39] and industrial monitoring and control [40]. In general, widespread WSN deployment in industrial context requires a significantly higher level of trust in the system compared to traditional, best-effort WSN projects.

For a wireless network to fulfil the requirements posed by those kinds of applications, the network stack has to guarantee certain aspects of network reliability and timeliness. In general, those networks are referred to as real-time networks because the delay is upper-bounded. Although such guarantees are a major requirement especially for industrial applications, not many solutions are available. This is why the GINSENG project was started in 2008. From a networking perspective to create a network that can be considered to be real-time capable, performance control is necessary. Performance control refers to the ability of individual elements of the network stack to ensure a certain performance so that in the end the application requirements are met.

As outlined in Section 1, a significant amount of research leading to the majority of publications included in this thesis stem from the GINSENG project. Hence, the GINSENG project is introduced and discussed in detail in this chapter. The context of the project allowed to work on applications with real-world constraints as found in industry. GINSENG is one of the first projects to produce a real-time capable end-to-end WSN solution that is fully published in scientific publications. Compared to industrial approaches such as ISA 100.11a [41], WirelessHART [42], IEEE 802.15.4e [43] and WIA-PA [44] all important building blocks of GINSENG are publicly available in scientific papers. In the following section, the application scenario in an oil refinery in Portugal is shown. After that, the GINSENG system is presented in detail and some results are discussed. Building on top of GINSENG, the following section outlines how to achieve end-to-end reliability in a GINSENG network by using centralized, offline schedule calculation and local, online transmission power control. While the coming sections are all based on the GINSENG system, the presented solutions can be generally applied to a variety of TDMA-based networks.

The discussion in this chapter is primarily based on the following publications: [1, 2, 5, 6, 10, 11, 15, 18, 23]. Those publications are part of this cumulative dissertation and hence are included in this document. An overview can be found in Section 1, whereas the publications can be found starting on page 47.

3.1 Application Scenario

The Petrogal Oil refinery in Sines, Portugal is a complex industrial facility covering over 3 200 000 m^2. A significant number of physical processes related to the production of various products such as gasoline, Jet fuel and many others have to be monitored and controlled. Tasks such as leakage detection, measurement of product flow and fluid level detection in storage tanks are performed using over 35,000 sensors and actuators in the field. At the moment those are connected to control rooms using wires. Sensors are generally sampled in the order of every few seconds and resulting measurement values are transported to the control rooms. Based on several visits to the refinery, core requirements of the applications currently realized in the field have been collected and documented in the public

GINSENG project deliverable D1.3 [45].

The refinery is classified in different Atmosphére Explosibles (ATEX) zones to prevent ignition of a potentially explosive atmosphere. Deploying electronic equipment in those areas requires certification to ensure safe operation. Uncertified equipment may only be used in the open if contained in a certified ATEX box and if heat dissipation is limited.

The refinery currently uses three systems for monitoring and control of the physical processes: 1) The indicatory system shows recent measurements on displays in the control room and allows operators to monitor the refinery operation. The wired system allows defining setpoints and triggers alarms whenever readings reach or exceed the setpoints. The acceptable delay for such readings is 3 s, whereas 99 % of the measurement values have to be delivered to the control room in time. 2) The semi-automatic control system includes actuators in the field as well and allows operators to issue actuation commands to those. The requirements towards delay and reliability are comparable to the indicatory system whereas data flows in two directions: from the sensors to the control room and from the control room to the actuators. 3) The automatic control system realizes control loops by issuing actuation commands based on sensor input. Operators can monitor sensor values and modify parameters of the control loops. The delay requirements are tighter compared to the previous two use cases. A delay of 1 s from sensors to the control room and another 1 s vice versa is acceptable, whereas the reliability requirement remains unchanged. In practice, actuators are located physically close to the corresponding sensors because they belong to the same physical process.

To adapt the refinery to changing needs in terms of output products, physical reorganization happens regularly. Adapting the control networks is costly because laying wires in environments with potentially explosive atmospheres is expensive. The operators of the refinery are therefore keen to transform the existing control networks into wireless networks to enhance flexibility and reduce cost for maintenance operations. However, the wireless networks have to guarantee the required level of reliability and timeliness.

3.2 The GINSENG System

GINSENG is a EU-funded 7th Framework Programme for Research and Technological Development (FP7) project that aims at researching so-called performance-controlled WSNs. The goal is to provide an end-to-end solution that collects information in the field, transports it to in-field sinks and processes it in a middleware component to finally allow high-level decisions based on streams of data. To achieve this goal, GINSENG relies on a fixed physical topology and uses a static logical tree topology on top of that. A key feature of the GINSENG approach is to prove the developed solutions in a real industrial setting, namely the oil refinery in Sines, Portugal.

The basis of the GINSENG WSN is a multi-hop network with a TDMA MAC layer called GinMAC [46]. By using exclusively assigned slots, GinMAC avoids collisions within the network by design. GINSENG uses a logical tree topology on top of the physical topology of nodes and a corresponding TDMA schedule based on the logical topology. The logical topology is rooted at the sink node, which is also the source for time synchronization. Time synchronization packets are broadcast down the tree to eventually reach all leaf nodes. Upstream measurement data destined for the sink node is sent in unicast packets, wherein each packet is sent in one TDMA slot. A TDMA slot in GinMAC contains the original packet transmission as well as an acknowledgement from the receiver. A node that is parent of other nodes in the logical tree topology needs upstream slots for its own packets as well as slots to forward packets coming from its children.

A TDMA slot in GINSENG typically has a length of $D = 10$ ms. A TDMA schedule $S = \{s_1, s_2, s_3, \ldots, s_n\}$ is composed of multiple slots $s_{1,\ldots,n}$ and is repeated regularly. The duration of one cycle (so-called epoch) of the schedule can be calculated as $E = |S| \cdot D$. GinMAC has four different types of slots: basic slots, additional slots, probe slots and unused slots. *Basic slots* are used to allow a node to forward one packet towards the sink or the sink to send one packet towards one actuator.

Additional slots are used for retransmissions if a packet in a basic slot has not been acknowledged. *Probe slots* are used to send probe packets to another node. *Unused slots* are appended to the schedule to allow for time-consuming processing tasks or to reduce the duty cycle. To ensure timely delivery of data from nodes to the sink, the epoch duration E has to be less or equal to the application delay requirement T in the network. With a typical slot length of $D = 10$ ms, GinMAC is able to perform 100 transmissions per second within the whole network. Due to exclusive slot use, the transmissions must be divided between all nodes in the network.

TDMA networks with exclusive slot usage must trade-off the delay requirements of the application with the maximum number of nodes that can be supported to ensure $E \leq T$. Since parent nodes need additional slots to forward data, having a long hierarchy (high tree height) drastically decreases the number of nodes the network can support. For a typical maximum delay in the order of 1 s to 3 s, a maximum network size of 24 nodes [1] has to be assumed.

From a network perspective, the GINSENG scenario requires to deploy nodes at each location where a sensor or actuator is installed in the field. The deployment locations are defined by the physical processes and cannot be modified to benefit the network. Additional relay nodes may be installed to provide better network connectivity. Power supply in the refinery can either be done via batteries (for nodes that are either mobile or otherwise hard to reach) or via cable infrastructure. Although cables are inflexible, power often has to be installed in the field in any case because industry-class sensors and actuators cannot be run on standard AA batteries.

The radio environment in the refinery can be characterized to be harsh [47]. While the use of portable electronic equipment such as mobile phones is largely prohibited in the open, large metal structures lead to short ranges and multi-path propagation. Furthermore, interference stemming from motors and other industrial equipment leads to further diminished propagation characteristics and to subsequently low ranges and high packet loss rates.

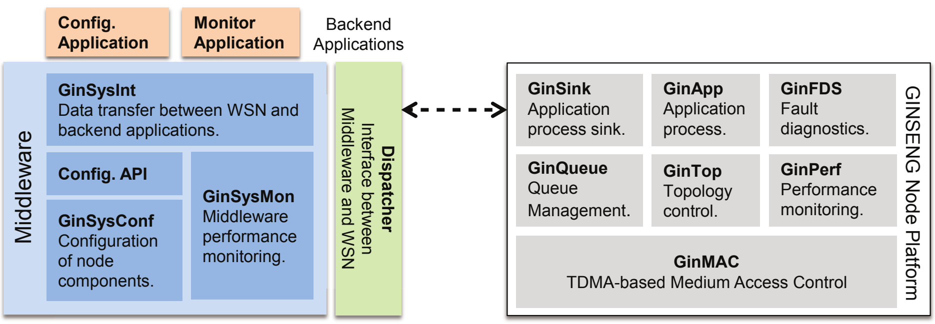

Figure 3.1: GINSENG architecture with components in the back end (left) and on the node (right). Figure originally from [2], reproduced here with modifications.

On top of GinMAC, various other components are put into place (see Figure 3.1) to handle the hurdles of industrial environments such as the refinery. In the scope of this thesis, mainly the following components are of interest:

- *GinTop* - GINSENG Topology control: responsible for local handover decisions for mobile nodes.
- *GinPerf* - GINSENG Performance Debugging: collects statistical and debugging information and transport them to the sink.
- *GinApp* - GINSENG Application Process: samples and sends data from sensors and execute actuator commands.
- *GinSink* - GINSENG Sink Process: runs on the sink node, handles communication with the sink PC and distributes time information in the network.

- *Dispatcher*: Middleware component running on the sink PC connected to the sink node by wire to exchange data with the WSN.

The GINSENG solution has been accepted for publication in “ACM Transactions on Sensor Networks” [2] (see page 49) in which more details on the individual components and the overall architecture are given. The paper provides deeper insight into the application scenario for GINSENG and details how a typical GINSENG deployment looks like and how experiments in the Sines refinery have been performed. The GINSENG solution has been evaluated in two WSNs in the refinery and mostly fulfils the application requirements in terms of maximum end-to-end delay as well as end-to-end packet loss ratio. However, a single tree-branch showed significant packet loss after one day and even more packet loss after three days of measurement. The paper further discusses related efforts for wireless networks in an industry context.

The author of this thesis worked as part of the GINSENG project and was involved in many areas of the project. The contribution of the author for this particular publication includes the following key elements: Significant contributions to GinPerf, GinSink and the Dispatcher (see next subsections for details). Work on the testbed in the refinery especially on the remote reprogramming system (see Section 3.2.3). Processing, filtering and graphing the measurement results obtained in the Sines refinery in Portugal.

3.2.1 GinPerf

One of the cornerstones of GINSENG is to provide a WSN with controlled performance thus requiring performance debugging mechanisms to locate and pinpoint problems in the network. The following information regarding GinPerf is contained in the public GINSENG project deliverables D2.4 [48] and D2.5 [49] and is partially reproduced here for completeness of this thesis.

Each node keeps a Management Information Base (MIB) in its RAM that contains information about its neighbours, the links to neighbours, node statistics and other information. That information is sent out to the sink node on a regular basis, whereas GinPerf messages have a lower priority than regular application messages. The collected information allows pinpointing individual links that are experiencing high link losses. This helps to identify which links should be either replaced or improved (if possible). To allow a visual representation of the collected information, a monitor application running on a PC in the back end network displays the collected information. The application allows a detailed look into the network and helps to quickly locate the source of a problem.

Monitoring the energy consumption of a WSN is crucial because some nodes may run on batteries and have limited lifetime. Contiki’s software-based online energy estimation [50] allows estimating the current energy consumption on a node without hardware support. Using this technology, all nodes periodically send a packet containing their own energy readings to the sink node. The monitor mentioned above allows operators to inspect energy consumption of each node. Furthermore, due to the predictable energy consumption pattern of TDMA networks, the lifetime of nodes can be easily predicted. By continuously comparing the prediction with the real measurements, misbehaving nodes can be located. In fact, one node in the Sines refinery showed alarmingly high energy consumption during one experiment and was replaced to prevent a complete hardware failure.

3.2.2 GinSink and Dispatcher

The following information regarding GinSink and the Dispatcher is contained in the public GINSENG project deliverable D2.4 [48] and is partially reproduced here for completeness of this thesis.

GinSink is the application running on the sink node of the GINSENG network. In all current GINSENG deployments, the sink node is a standard Tmote Sky [34] node with a USB port. The USB port is connected to the sink PC running the dispatcher component, thereby allowing two-way communication between GinSink and dispatcher. The dispatcher exposes several Transmission Control

Protocol (TCP) ports offering data coming from the WSN in different formats. This allows components running in the back end (see Figure 3.1) to receive and send information from and to the WSN in raw and Extensible Markup Language (XML) format as well as in Java object format.

Incoming data from the network is received by GinMAC, handed over to GinSink and forwarded to the dispatcher. Incoming data from the dispatcher is in turn handed over to GinMAC to be forwarded to nodes in the network. However, timing on the sink node is extremely critical to avoid interfering with the TDMA operation. This is especially important because the sink node is the source of time synchronization for the whole network. Thus, operations on the serial port must be completed within a single GinMAC slot.

The serial connection between GinSink and the dispatcher runs at $460\,800$ baud, which allows transferring a 120 byte GINSENG packet within 2.6 ms [23]. To protect against corrupted or missing symbols on the line, Cyclic Redundancy Check (CRC) is used for each data and control message. Synchronization between dispatcher and GinSink is established using Serial Line IP (SLIP) [51]. Since timing is critical on the sink node, a master-slave concept is used in which the sink node is the master. To ensure that the dispatcher does not send packets at inappropriate times (during radio activity for example), a software-based flow control is employed. All packets between dispatcher and GinSink contain sequence numbers and are acknowledged by the receiver after verifying the CRC. The sender uses a sliding window approach with fixed timeouts and retransmissions to ensure reliable data transmission.

GinMAC notifies GinSink whenever enough time until the next active TDMA slot is left. GinSink performs operations (such as sending a packet to the dispatcher) until either no more operations are pending (because the queue is empty) or until the next TDMA slot is scheduled. If no operations are left and there is still time, GinSink notifies the dispatcher by means of a Transmission On (XON) packet that there is opportunity to transmit data to the node. XONs contain the time that is left until the next active TDMA slot and also the amount of space that is left in the buffer. The dispatcher will then send data packets going to the WSN to GinSink and wait for acknowledgements using the sliding window approach.

Since GINSENG TDMA schedules are provisioned with back-to-back transmission slots, packets have to be buffered on the sink node until time for serial transmission is available. GinSink features a multi-packet buffer that holds packets until an acknowledgement from the dispatcher has been received (packets going to the back end network) or until GinMAC has been accepted the packet (packets going to the WSN).

Determining the end-to-end delay in a GINSENG network is crucial, both for the end user and for scientific evaluation. Since sensor nodes operate in their local time domain, synchronization and translation between the WSN and the PCs has to be established. Multiple PCs can be synchronized using Network Time Protocol (NTP) [52] and GinMAC synchronizes GINSENG nodes. Time synchronization between nodes and PCs is established by the dispatcher who timestamps incoming packets from GinSink upon reception. Since all packets contain the GinMAC internal creation as well as reception timestamps, the dispatcher calculates the offset between the PC and WSN. The offset is calculated for each incoming packet and smoothed using Exponentially Weighted Moving Average (EWMA). To account for the unpredictable delay on the serial connection, the sink node measures the time between sending out a data packet and receiving the Acknowledgement (ACK). This measurement is fed back to the dispatcher, smoothed via EWMA and taken into consideration for the time synchronization.

The dispatcher interprets incoming packets from the WSN. Packets with a well-defined format are transformed into XML and raw values are converted into a standard unit measure. Furthermore, energy consumption of individual nodes is calculated by the dispatcher and the end-to-end packet loss is computed as well.

(a) Refinery testbed overview

(b) ATEX enclosure with external antenna

Figure 3.2: GINSENG testbed in live oil refinery

3.2.3 GINSENG Testbed

Technical details on a preliminary version of the GINSENG system are published in "Proceedings of the 3rd International Workshop on Performance Control in Wireless Sensor Networks" [18] (see page 51) whereas an update on the actual layout of the final testbed is given in [2] (see page 49).

One cornerstone of the GINSENG approach is to prove components in a real industrial setting. Therefore, a WSN testbed has been installed in the Sines refinery in Portugal (see Figure 3.2) to be able to perform long-term experiments on site. The testbed consists of a total of 26 nodes; whereas the nodes have been divided into two separate WSNs with 12 sensors and 1 sink each. Most nodes are connected to actual sensors in the refinery thus sampling live data from the production process. Some nodes are connected to actuators that are installed in the refinery but do not actuate anything for safety reasons. The remaining nodes are not connected to sensors or actuators and send random data to the sink. Please note that from a network perspective it does not make a difference whether a node sends live or random data.

Operating a testbed in a productive industrial facility is challenging because certain regulations have to be met. Furthermore, installing non-certified hardware in potentially explosive atmosphere requires nodes to be contained in so-called ATEX boxes to comply with the regulations. Since operating a laptop in the open is not allowed, manually reprogramming the testbed requires removing the node from an ATEX box, carrying it to a portable office container, flashing it and putting it back into the ATEX box. To allow efficient use of the testbed and to enable coordinated experiments, remote access to the testbed for log collection and node flashing is essential.

An embedded PC[1] running Linux is contained inside each ATEX box and provides power and USB connectivity to the node. Since the ATEX boxes are 150x150 mm in size, the Linux boards have to be rather small and the heat dissipation has to be low. The hardware is put into the ATEX box alongside the Tmote Sky node (see Figure 3.3) and connected to a central switch using standard Ethernet cabling. Power is received over the Ethernet cable using non-standard Power over Ethernet (PoE). Since the distance to the portable office where the Ethernet switch is located is larger than 100 m for some nodes, the Ethernet connection exposes up to 50 % packet loss. To cope with the loss, a TCP connection is used to communicate between the central testbed server and the embedded PCs.

[1] Technologic Systems TS-7500 `http://www.embeddedarm.com/products/board-detail.php?product=TS-7500`

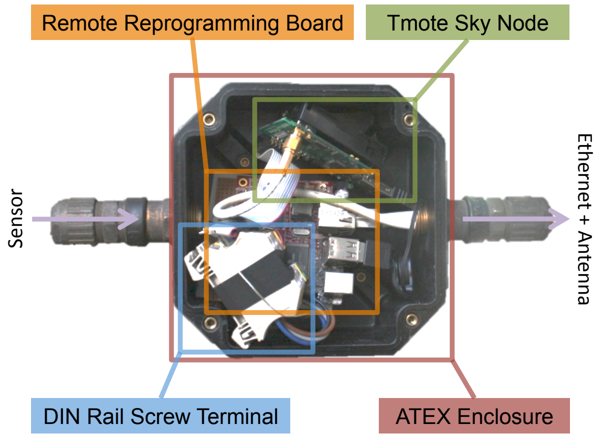

Figure 3.3: Tmote Sky node installed in ATEX box with remote reprogramming support

The non-standard PoE uses the unused 4 wires in the Ethernet cable to transport 24 V to the ATEX boxes where a step-down converter produces 5 V to feed the TS-7500 board. Standard PoE could not be used because commercial modules to produce 5 V from PoE would not fit into the ATEX boxes.

A daemon running on the embedded PC collects log output from the serial ports of the nodes and transfers the logs to the central server. The embedded PCs are time synchronized to the testbed server to produce log messages coming from the nodes with precise timestamps. The testbed server distributes flash images to the embedded PCs via Secure Copy (SCP) to be flashed onto the nodes. The embedded PCs flash the images autonomously and report back the result to the testbed server. Furthermore, the server allows resetting all nodes either individually or at once to allow for coordinated starts of experiments.

To allow all partners of the project access to the testbed, accessibility over the Internet is key. Since the refinery does not have network infrastructure installed in the vicinity of the portable office a 3G cellular uplink was used. To work around problems with Native Address Translation (NAT) in typical 3G cellular networks, the testbed server proactively established a Virtual Private Network (VPN) connection to a server located in Braunschweig, Germany. This computer was the gateway for GINSENG partners to access the testbed.

3.3 Reliability in Real-Time Wireless Sensor Networks

A GINSENG network (or in general any TDMA network) operates just fine when no frames are lost in the air. In practice however, frame loss on radio links is not an unusual phenomenon [53]. We have to assume that frame loss in the air cannot be avoided. Generally, losses are often caused by interference. Such interference can be caused by other networking technologies on the same band, by the same networking technology using the same channel or even by broadband interferers such as industrial equipment. Predicting interference is not a trivial problem and out of scope of this thesis.

To overcome interference (and resulting packet loss), the sender has to retransmit lost frames in the hope that the reason for interference is short lived. In IEEE 802.15.4-based networks, the sender can request an explicit ACK frame for outgoing data frames. This is the normal procedure for unicast frames and allows the sender to determine whether a data packet has reached its single-hop destination or not. Please note that the ACK frame can be lost as well thus leading to a seemingly lost data packet (false negative) whereas only the ACK did not arrive at its destination. In a TDMA network with fixed slot assignments and exclusive slot usage, packet loss in the air may lead to end-to-end packet loss if no retransmission opportunities are available. If a radio frame is not acknowledged, the sender has to resend the frame within the current epoch to meet the deadline. Consequently, the epoch has to include additional slots that can be used to retransmit packets. In effect, the expected number of retransmissions has to be known when the schedule is constructed. However, to include enough slots for retransmissions into the schedule, the number of expected losses has to be known in advance. Section 3.4 outlines, how this can be done.

Interference with the same networking technology on the same channel is a special problem that can be solved more efficiently than by pure retransmissions. Such interference occurs for example, when multiple networks are spread out in geographical proximity. In industrial facilities such as the Sines refinery, 35,000 sensors and actuators have to be connected wirelessly. With up to 24 nodes (23 sensors and 1 sink) per network, at least 1,521 networks have to be installed in parallel. The following approaches are possible to avoid or reduce interference between those networks:

- **Synchronize networks** in the time domain to avoid overlapping slot usage. However, this approach costs time and will lead to a longer epoch duration or less nodes in a network.
- Spread networks on **multiple channels**. However, with 1,521 networks and 16 channels in IEEE 802.15.4, channel reuse cannot be avoided and interference will occur.
- **Negate exclusive slot usage** and make nodes within a single network use slots at the same time. This requires extensive knowledge of the interaction between links to avoid collisions and is subject of existing research [54].
- **Reducing the transmit power** reduces interference range and can help to reduce interference.

In practice, synchronizing networks significantly increases system complexity due to inter-network coupling and eventually prolongs the epoch duration. This also increases the minimum end-to-end delay guarantees that can be given. Breaking up the exclusive use of TDMA slots increases system complexity significantly and makes the system harder to predict. Therefore, reducing the transmission power is a viable option as it increases system complexity only slightly. Furthermore, a network can be spread onto multiple channels to further reduce interference. A combination of using multiple channels and reducing transmission power is a viable combination that is pursued in this thesis.

The following section discusses an approach to calculate a TDMA schedule for real-time networks based on previously collected data about link reliability. In addition, section 3.6 shows how purely local information can be used to further optimize transmission power.

3.4 Calculating TDMA Schedules for Real-Time Wireless Sensor Networks

The work presented in this section has been accepted for publication in "ACM Transactions on Sensor Networks" [1] (see page 53).

The goal is to produce TDMA schedules that are reliable, respect the requirements of the application and minimize interference at the same time. The process of obtaining a schedule for a network is threefold and starts after deploying nodes in the field. 1) First of all, data about the worst-case packet losses of links has to be collected. To do so, links are measured multiple times to determine their loss behaviour. 2) With this information, the schedule calculation algorithm produces a logical topology as well as the corresponding schedule. Since the schedule is based on measurement results, it is merely the best schedule based on the results and may deviate from the absolute best schedule. 3) The calculated schedule is then flashed onto the nodes and the network can start operating. Since calculating the best schedule for networks of reasonable size is computationally infeasible, a heuristic to speed up the process is discussed as well. The whole process is detailed in the following subsections.

The approach discussed in this section is based on the following assumptions: The physical locations of all nodes are fixed and do not change. Furthermore, the traffic patterns of all nodes are fixed and known in advance. If patterns are changing, an upper bound for the number of outgoing packets per epoch can be given. Finally, the approach shall not require knowledge about the environment to enable non-expert users to produce a TDMA schedule. Those assumptions are generally valid over extended time periods. If physical node locations change due to plant reorganization, the schedule calculation process has to be restarted.

3.4.1 Burst Behaviour of Links

To calculate a TDMA schedule including slots for retransmissions requires knowledge about the loss behaviour of a link. Research has shown [55] that packet losses typically occur in bursts. Using statistical metrics such as Expected Transmission Count (ETX) or PRR to characterize a link do not take burst length into account. This may lead to a link with high statistical reliability but still a significant number of packets in a row can get lost with potentially severe consequences. Since traditional TDMA schedules are often laid out with all slots of a node back-to-back, a burst loss may impact all of those thus leading to end-to-end packet loss.

It has been shown [55] that burst length is a much better way of characterizing the reliability of a given link. The burst behaviour of a link (so-called burstiness) can be expressed with B_{max} and B_{min}. B_{max} is the maximum number of unsuccessful transmission in a row and B_{min} is the minimum number of successful transmissions after a burst loss on a specific link. The burstiness can be measured by sending a number of so-called probe packets [56] over a link. The sender collects the ACK responses for the probes in a pattern of 1s (ACK received) and 0s (no ACK received). Based on this pattern, B_{max} and B_{min} can be computed.

For a node to successfully transmit P outgoing frames given the link burstiness of B_{min} and B_{max}, a maximum of n_{slots} as expressed in equation 3.1 is necessary. Essentially, B_{max} slots are necessary to overcome a potential loss burst. Then, up to B_{min} packets can be successfully transferred. If a link shows a burst behaviour of $B_{min}/B_{max} = 1/1$ that means that every other packet may get lost. For each outgoing packet, two slots would have to be scheduled to overcome possible packet loss.

$$n_{slots} = \lceil \frac{P}{B_{min}} \rceil \cdot B_{max} + P \tag{3.1}$$

Link quality changes over time and so do burst sizes. However, the worst-case description of a link in terms of B_{max} and B_{min} is generally accurate over long time periods [55]. To determine the worst-case burstiness for a link, multiple measurements at different points in time have to be performed (see Section 3.4.6).

3.4.2 Data Collection

To be able to calculate the best schedule for a given set of nodes, all links between those nodes have to be characterized in terms of their burstiness. To minimize the interference with neighbouring networks during productive operation of the network, the transmission power on all links shall be kept at the necessary minimum. Since the burstiness is heavily dependent on transmission power, different power levels must be considered during data gathering. Therefore, sender and receiver nodes as well as the transmission power of the sender define a link in the scope of this thesis. In a network with N nodes and M radio transceiver power levels, a total of $L = N \cdot (N-1) \cdot M$ links have to be probed.

Probing a link is done by sending a sequence of P probe packets. For each sequence, the burstiness in terms of B_{min}/B_{max} is calculated. Probing a link multiple times at different times of the day allows capturing time-varying characteristics of the burstiness of the link. When a link is probed multiple times, the worst-case burstiness is used as representative for this link to ensure provisioning for the worst-case losses that have been observed.

Probing is done with nodes running the GinMAC software with a TDMA schedule that includes primarily probe slots. Beacons that are broadcast by the sink, whereas nodes in the field rebroadcast them, establish time synchronization. Sender and receiver pairs as well as transmission power are determined in a round-robin fashion based on the number of the TDMA epoch. Resulting patterns of B_{min}/B_{max} values can either be stored on the nodes and collected later on or sent to the sink via the wireless network. While the former solution requires a file system solution that does not interfere with the TDMA schedule, the latter solution requires an initial schedule to transport data to the sink. After all links have been probed and data is persistently stored on the nodes, data from all nodes is collected at a central point. Details on two approaches for data storage on nodes can be found in Section 3.5.

3.4.3 Calculation Algorithm

Based on the burstiness measurements in the previous section a TDMA schedule is calculated. While the physical topology of nodes is fixed, the logical tree topology is unknown and has to be calculated as well. A schedule is then calculated based on this logical topology, the burstiness of links and the traffic pattern of nodes. The algorithm discussed in the following is an exhaustive search through all possible topologies and corresponding schedules.

A valid topology has to form a tree rooted at the sink and contain all nodes in the network. Furthermore, the sink must be reachable from all nodes within H hops and nodes must not have more than C children. The latter two conditions are for practical reasons to avoid inaccurate time synchronization and nodes using most of the time and their energy for forwarding. A valid schedule is based on a valid topology, allows each node to transmit the necessary number of messages to the sink within an epoch, contains retransmission slots according to B_{max} and B_{min} and has an epoch length $E \leq T$. The optimal schedule is the schedule among the valid schedules that minimizes the energy signature ϵ (see following section for details). Since the optimal schedule requires knowledge of the real worst-case burstiness of all links, it is unlikely to be found in practice. Therefore, the best schedule is the schedule in the obtained data set that minimizes ϵ. Finally, the found best schedule is the best schedule found by the heuristic, which may not be as good as the best schedule.

A set containing all possible topologies is calculated using a recursive algorithm that iterates through all outgoing links of all nodes. For a network with N nodes and M power levels, $((N-1) \cdot M)^N$ combinations are possible whereas invalid topologies are discarded. For each valid topology a schedule is calculated. For each node, the number of upstream packets is summed up (based on its own outgoing packets plus packets from the children) and converted into transmission slots (see Section 3.4.1) while taking the burstiness of the upstream link into account. If a node has child nodes, additional downstream slots are added to the schedule. All schedules that are invalid (for example because the epoch E is too long) are discarded. Remaining schedules are compared to the previously known best schedule. The new schedule replaces the existing schedule, if its energy signature ϵ is lower. For two schedules with equal ϵ, the shorter epoch E is considered to be better.

While this algorithm finds the best schedule with respect to the collected data set, the complexity is $O((M \cdot N)^N)$ for N nodes and M power levels. In a network with $N = 6$ nodes and $M = 32$ power levels, the calculation takes 5.67 h on a powerful up-to-date computer. Clearly, this is computationally infeasible for networks of reasonable size. A heuristic to speed up the calculation process is shown in Section 3.4.5.

3.4.4 Energy Signature ϵ

Comparing two schedules requires a metric to pick the better schedule. The goal is to produce a schedule that is reliable while meeting the application delay requirements and minimizing interference. All schedules calculated with the algorithm stated above fulfil the former two criteria. Therefore the better schedule with respect to the stated goal is a schedule that reduces interference.

Reducing interference between neighbouring networks can be achieved by reducing the transmission power of each network. A lower transmission power reduces the range of the transmission [57] and correspondingly the spatial coverage. This in turn lowers the interference with neighbouring networks because those networks receive weaker interfering signals compared to packets sent at the maximum transmission power. This is emphasized by measurements [1] that show a lower number of received packets and at lower power levels when reducing the transmission power of the sender.

$$\epsilon = \sum_{i=0}^{k} M_i \cdot D \tag{3.2}$$

ϵ is calculated by summing up the transmission power M_i used in all k slots multiplied with the slot duration D as shown in equation 3.2. Minimizing ϵ reduces the transmission power of all nodes in the network while taking the number of transmission slots into account. When comparing two schedules, the schedule with lower ϵ is considered to be better. The energy signature is a very rough approximation for the interference because the power and position of nodes is not taken into account. However, given practical considerations, the approximation is reasonable [1].

3.4.5 Heuristic

A heuristic must be used to limit the search space for the schedule calculation algorithm and to reduce the time needed for computing the found best schedule. By limiting the maximum number of outgoing links per node to T_L the complexity can be reduced. Instead of taking $(N-1) \cdot M$ links per node into account, only T_L outgoing links with $T_L < (N-1) \cdot M$ are considered. This reduces the complexity to $O((T_L)^N)$. To ensure a high schedule quality, only links that are unlikely to be used in the final schedule should be discarded.

In a first step, bad outgoing links are pruned by discarding links on which no packet could be transmitted at all or links with B_{max} above a threshold $T_{B_{max}}$. In the second step, the remaining links are sorted by ascending transmission power, by ascending B_{max} and by descending B_{min}. This ensures that low-power links with high reliability are sorted to the front of the link list. For the schedule calculation, only the first T_L outgoing links of the list for each node are considered. The exponential dependency on the number of nodes N is irrelevant in practice because networks are limited to 24 nodes.

Measurements in the refinery have shown that $T_L = 5$ is a reasonable value. A comparison based on the refinery data set has shown that the schedules that have been calculated with the heuristic have a slightly longer epoch and a higher energy signature ϵ compared to schedules that were calculated without excluding links. When computing a schedule for 6 nodes (the maximum that is computationally feasible without heuristic) based on the refinery data set, the schedule computed without heuristic has $E_B = 14$ slots and $\epsilon_B = 3.06\,\mu$Ws whereas the schedule computed with heuristic has $E_H = 18$ slots and $\epsilon_H = 3.81\,\mu$Ws. However, compared to the reduction in time required to calculate the schedule (5.67 h without heuristic compared to 0.14 s with heuristic) the difference is acceptable. In a network with $N = 13$ nodes, computing a schedule with the heuristic takes 196.6 s.

3.4.6 Required Probing Effort

To probe a link, P probe packets are sent between two nodes and B_{max} and B_{min} are calculated based on the collected patterns. Sending more probes on a link increases the probability to capture a worst-case burst loss. In the Sines oil refinery, burstiness has been measured with $P = 40$ probes per epoch over a period of 23.4 h, which allowed to probe each link in at least 16 epochs. To find out how much probing is necessary, the real worst-case burstiness of a link would have to be known. Given the very low end-to-end packet loss of $0.01\,\%$ in the refinery, the observed worst-case is assumed to be a good approximation of the real worst-case. Using more probes will characterize the burstiness of links more accurately. With a $T_{B_{max}}$ threshold using more probes will exclude an increasing number of links from the schedule calculation. By truncating the collected patterns after P' probes, then calculating the burstiness and comparing it to the observed worst-case, the fraction of links that are erroneously taken into consideration can be determined for a given data set. The results from the refinery show that using $P' = 5$ probes and $T_{B_{max}} = 4$ leaves 74.1 % of links available for schedule calculation. Increasing P' to 20 brings the fraction of links down by $1.76\,\%$ while adding another 20 probes eliminates another $0.49\,\%$ of the links. Having $1\,\%$ of the links classified incorrectly is reasonable and requires $P' = 14$ probes.

Link quality changes over time and so does burstiness. To determine the minimum time necessary to collect data until the worst-case burstiness is observed, a window of length a can be shifted through the b probe responses collected per link with $a \leq b$. The minimum window a containing the worst-case burstiness for all links in any case (any offset) is the minimum amount of time that has to be probed to observe the worst-case. The assumption is that environments normally show a certain periodicity within which the behaviour of links is repeated (think of day/night or weekday/weekend cycles). In the refinery, all 20.8 h long windows out of the 23.4 h of collected data show the worst-case burstiness for all links. In one of the office environments (see Section 3.4.7 for details), the minimum window is 141.44 h out of 152.5 h of probing.

3.4.7 Results

The schedule calculation approach has been evaluated in the refinery as well as in two office environments. The respective baseline for comparison is GinMAC with a static, handpicked schedule that has been used for previous long-term experiments [2]. Handpicking a schedule was the only option available before using computed schedules and involves expert knowledge, knowledge about the environment and a limited number of link measurements. Handpicked schedules always use the maximum transmission power setting.

In the refinery, a network with $N = 13$ nodes has sent one packet per node per second towards the sink during a 55 minute experiment. The end-to-end packet loss for all nodes in the network has been summed up and analysed over time. While $1.53\,\%$ of the packets got lost when using the handpicked schedule, only $0.01\,\%$ of the packets got lost when using the computed schedule despite significantly lower transmission power levels (down from level 31 to levels 2 - 4 on the CC2420 [35] radio transceiver).

Another experiment lasting 308 h in a wisebed [58] installation in the Lübeck office environment and involving $N = 9$ nodes was used to evaluate long-term schedule stability. In this experiment, $0.38\,\%$ of all packets were lost when using the computed schedule. No handpicked schedule was used as baseline since the goal was to show the long-term stability of the computed schedule. The highest transmission power level was 14 (compared to level 31 as default). While the loss rate is significantly higher than in the previous experiment, the application requirements of $99\,\%$ packet delivery rate were still met.

To investigate the impact on network interference, 14 nodes in a wisebed installation in Braunschweig have been used in an experiment lasting 2 h. Besides a network of 13 nodes, an observer was placed at the fringe of the network to observe incoming packets and their power levels. When using the handpicked schedule with $E = 78$ slots, the observer received 27722 transmissions with most packets being received at -45 dBm, -75 dBm and -90 dBm. The computed schedule has $E = 170$ slots but the observer received only 19183 transmissions with the majority at -80 dBm. These results confirm that lowering the transmission power of a TDMA schedule reduces range of the network and interference with neighbouring network accordingly.

3.5 Persistent Real-Time Data Storage on WSN Nodes

Persistently storing data on nodes can be done on the on-board flash chips or optional SD card slots. However, traditional WSN file systems such as Contiki's COFFEE [59] are not made for real-time environments such as a TDMA network. Contiki is a cooperative multitasking runtime system and hence does not preempt tasks. If certain operations take longer than a TDMA slot, a node may miss the next active TDMA slot and loose synchronization. The main problem of COFFEE is the unpredictable timing behaviour. Since certain write operations take more than two orders of magnitude longer than the average (up to 711 ms compared to the average of 0.43 ms) [11], using COFFEE in a real-time environment would require to schedule a time that is three orders of magnitude larger than the average write time. While this is clearly not desirable, it is often not even possible due to the TDMA epoch length.

3.5.1 Ring-based Data Storage for Collected Data

To overcome this, Ring File System (RFS) [11] is a file system for flash media in real-time cooperative multitasking environments and is optimized for continuous data logging and en-bloc retrieval. Flash memory is organized into pages and sectors whereas pages can be written at once and only larger sectors can be erased at once. RFS has been published in "Proceedings of the 4th International Workshop on Performance Control in Wireless Sensor Networks" (see page 55). RFS exposes the same Application Programming Interface (API) as COFFEE and stores data in First In First Out (FIFO) buffers of linked flash pages. Copying of data is avoided whenever possible; potentially longer-running operations can be scheduled by the user at appropriate times. RFS optionally allows consuming data while reading it, just like a FIFO.

RFS organizes data in multiple rings that can be considered to be the equivalent of files. Multiple flash pages are combined into an RFS block and an RFS file consists of linked RFS blocks. File system meta data is stored in one sector thus speeding up meta data operations such as opening a file. One empty sector is always kept in reserve for garbage collection. Garbage collection is either executed when needed or when the user explicitly triggers it at appropriate points in time. Since garbage collection moves data from one physical sector to another one, a virtual sector mapping is applied.

Evaluation on a Tmote Sky node has shown that RFS shows a maximum write time for blocks of 8 bytes of 6.47 ms with an average of 0.42 ms (the median is 0.37 s) whereas COFFEE showed a maximum of 711 ms and an average of 0.43 ms (the median is 0.24 ms). So while COFFEE is slightly faster in the average case, the worst-case is significantly better when using RFS. The performance for reading 98 bytes (payload of one GinMAC frame) is a maximum read time of 1.46 ms for COFFEE (average of 1.25 ms) and 1.95 ms for RFS (average of 1.49 ms). The performance of RFS is lower due to the mapping of virtual sectors, which eventually allows having non-contiguous files.

3.5.2 FAT-based Data Storage for Collected Data

Storing data about the burstiness of links on the on-board flash of nodes is convenient. However, the flash may not be sufficient if truly large-scale storage is required. Typical WSN nodes have on-board flash capacities in the order of some megabyte. Furthermore, retrieving information stored in the flash of the node can either be done via specialized firmware over the serial port or over the air. While the former requires collecting nodes, the latter is limited by the throughput of the wireless interface that is potentially shared between many nodes. Hence, storing data on comparably cheap SD cards using a PC compatible file system such as FAT solves both issues.

The paper presenting RATFAT has been published in "Proceedings of the 5th Workshop on Performance Control in Wireless Sensor Networks 2013" [5] (see page 57). RATFAT is an implementation of the FAT [60] file system for WSN nodes running Contiki. RATFAT is optimized for SD card and supports SD High Capacity (SDHC) cards in both FAT16 and FAT32 format. RATFAT has an API that is compatible to the COFFEE API but supports only the 8.3-naming scheme. RATFAT uses a single-page buffer in memory and can mount either Master Boot Record (MBR) or non-MBR SD card volumes.

RATFAT ensures real-time compliance by splitting up file system operations into multiple atomic operations. After each atomic operation, RATFAT decides to either perform the next operation (if time allows) or yield the MCU. Atomic operations are SD block reads and writes. Those operations use the Serial Peripheral Interface (SPI) bus to which the SD card is connected and interruption of those would lead to unpredictable behaviour. On the INGA node running on 8 MHz with a transcend microSD card, the length of the atomic operations is 12 ms. To decide whether time for the next operation is left, RATFAT looks at pending timer events. Also, RATFAT has a maximum execution time (so-called slot size) after which the MCU is yielded in any case.

Applications that want to access the FAT file system use a real-time API. Each call to the API queues a file system request to the RATFAT queue and returns immediately (within 0.12 ± 0.03 ms). Applications receive events when a file system request has been finished and can query the state of each pending request in the meantime. Data for each request is buffered in RAM, so the application does not need to maintain its own buffer while the request is being processed. The RATFAT File System Process (FSP) consumes the FIFO queue and processes the requests in sequence. To keep its internal state, the FSP has a stack that is saved on context switches (in contrast to the Contiki protothreads [61]).

On a 2 GB microSD card from transcend, COFFEE regularly takes 20 ms to write 8 bytes but may take up to 5500 ms occasionally. Standard RATFAT (without real-time support) usually takes below 1 ms for 8 byte writes with occasional writes taking up to 71 ms. The throughput is up to 10.9 kiB/s in standard mode and drops to 6.0 kiB/s in real-time mode. Comparing standard RATFAT with real-time RATFAT shows that the latter can support a real-time process with an invocation frequency of up to 64 Hz while the former misses deadline when invoking the process at more than 16 Hz.

3.6 Probe-based Transmission Power Control

The work discussed in the present section has been published in “Proceedings of the 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013” [6] (see page 59).

Calculated TDMA schedules as discussed in section 3.4 are static and optimized for the observed worst-case burstiness that occurred on the involved links during the data collection period. However, in practice links will rarely see the worst-case because wireless link conditions are changing over time. Hence the network uses a transmission power that is higher than necessary in many situations. Since transmitting at higher power than necessary leads to inter-network interference that is higher than necessary (see Section 3.3), the transmission power should be kept as low as possible at all times.

During periods in which links do not show their worst-case burstiness it is possible to reduce the transmission power (and hence the interference) below the static value found in the schedule.

The primary goal is to send packets at the necessary minimum transmission power to fulfil the burstiness assumptions on which the TDMA schedule is based. Since some nodes may be powered by batteries, the secondary goal is to minimize the energy consumption of all nodes in the network while ensuring that packets arrive at their destination in time.

Measurements in the refinery [6] have shown that the burstiness of links changes with the receiver signal strength. The burstiness as well as the statistical packet loss rate increases with decreasing receiver signal strength values. However, due to fluctuations in the measured values, a model function cannot be given. Since the burstiness changes over time, continuous measurement of the burstiness for different receiver signal strength values is required.

To pursue the stated goals, a two-stage approach is used in this thesis. A continuous attenuation-based Transmission Power Control (TPC) approach is used to ensure a defined receiver signal strength at the receiver as discussed in the coming section. However, to stay within the specified burstiness assumptions a probe-based approach for receiver signal strength target selection is incorporated and discussed in Section 3.6.2.

3.6.1 Attenuation-based Transmission Power Control Algorithm

For the sender to send packets with a transmission power that produces a certain receiver signal strength requires knowledge about the attenuation of the path. The attenuation A can be calculated as $A = \frac{P_r}{P_t}$ based on the received signal strength P_r and the transmission power P_t. If the sender wants to achieve a certain receiver signal strength target T, the necessary transmission power P_t' can be calculated as $P_t' = \frac{T}{A}$.

For each outgoing packet, the sender calculates the necessary transmission power and transmits the packet at the calculated power P_t'. The receiver feeds the receiver signal strength P_r back to the sender in the link-layer ACK and the sender calculates the link attenuation A. While assuming that no sudden changes of the channel attenuation occur, the sender smoothens multiple attenuation measurements using EWMA. Unacknowledged packets are assumed to have $P_r = 100$ dBm, which is well below the receiver sensitivity of the CC2420 radio chip [35].

While this approach allows modifying the transmission power to meet a receiver signal strength target T, the following section shows how to determine such goal given a certain link burstiness goal.

3.6.2 Probe-based Receiver Signal Strength Target Selection

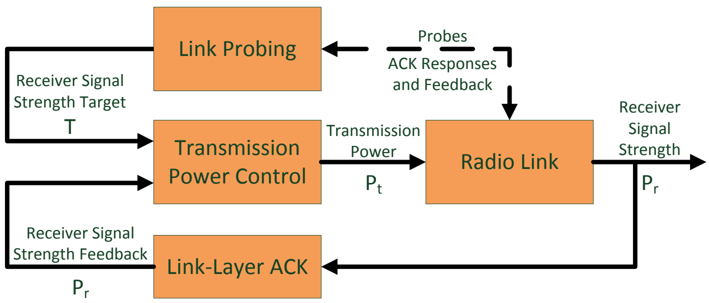

Figure 3.4: Probe-based Transmission Power Control approach

Link burstiness changes with receiver signal strength and over time, so that a fixed receiver signal strength target is no option. Repeated link probing has to be used to continuously update the receiver signal strength target. Appending shared probe slots to the TDMA schedule allows continuous probing of links. Those slots are distributed in a round-robin fashion among all child nodes and are used to probe the link to the parent at a random transmission power level. The receiver sends ACKs including the receiver signal strength and the sender smoothens the readings to obtain the average receiver signal strength for a series of probes. The sender further calculates the burstiness for this series and stores a tuple of average receiver signal strength, B_{min} and B_{max}.

In a network with $N = 14$ nodes, transceivers with $M = 32$ power levels and an epoch duration of $E = 1$ s, probing each link at each power level takes $(N - 1) \cdot M \cdot E = 416$ s. Probing is done continuously and values are stored in a ring buffer to churn out old results. The receiver signal strength target T is determined by looking for the lowest receiver signal strength value at which the burstiness of the link was equal to or below the burstiness assumed when crafting the schedule for this particular link. The relationship between the attenuation-based TPC and the probe-based receiver signal strength target selection is visualized in Figure 3.4.

3.6.3 Results

Measurements have been performed in the basement of a university office building with 14 nodes placed less than 1 m apart using a GinMAC schedule laid out for $B_{min} = B_{max} = 1$. The baseline comparison is a handpicked schedule at maximum power and the attenuation-based TPC with a fixed target of $T = -60$ dBm. The retransmissions in the network increased from $0.08\,\%$ for the two baseline approaches to $0.09\,\%$ when using the combination of probing and attenuation-based TPC. The average range calculated with the 2-ray ground reflection model [57] was reduced from 199.5 m with the handpicked schedule to 46.3 m when using the purely attenuation-based TPC with fixed target. With the probe-based TPC approach, the estimated range goes down to 34.6 m and 35.5 m for 4 and 8 probe slots per epoch respectively. Since range converts into interference range and then into interference, this is a noticeable effect in terms of network reliability when operating two networks on close proximity. The median energy expenditure summed up for all nodes in experiments each lasting 190 minutes is 42.48 mJ for the baseline schedule, 37.85 mJ for attenuation-based TPC, 39.01 mJ for 4 probe slots and 41.58 mJ for 8 probe slots, both in combination with attenuation-based TPC.

3.6.4 Receiver Signal Strength Feedback in Link-Layer Acknowledgements

Feedback about the signal strength at the receiver is required for the attenuation-based TPC to measure the attenuation of a link. Such feedback can be given at no cost in terms of time or energy when recycling three unused bits in the Frame Control Field (FCF) of the IEEE 802.15.4 ACK frame header. How this can be achieved has been published in “Proceedings of the 4th International Workshop on Wireless Sensor, Actuator and Robot Networks” [15] (see page 61).

Receiver signal strength feedback contained in link-layer ACK frames is a challenge in IEEE 802.15.4-based wireless networks. The IEEE 802.15.4 specification [31] requires ACK frames to be received within $864\,\mu$s ($macAckWaitDuration = 54$ symbols in the 2.4 GHz band) after the last symbol of the corresponding data frame has been received (so-called ACK time). To meet the stringent timing requirements, all modern IEEE 802.15.4-compliant packet-based transceivers generate and send the ACK frames fully automated in hardware. Code running on the MCU is not involved in this process. To modify the FCF of the ACK, the AutoAck feature of the transceiver has to be disabled and the ACK must be generated and send in software running on the MCU.

To generate ACKs in software, reading the frame from the receive buffer after the complete frame has been received (standard procedure in most drivers) is too slow. Hence, the frame has to be read out in blocks while it is still being received. After reading the first 4 bytes, the ACK can be generated and put into the transmit buffer. In a read loop, the remaining bytes of the packet are copied to the RAM of the MCU. Whenever possible, a complete block is read. Otherwise, a single byte is read over the SPI bus. While SPI is purely byte-oriented, reading blocks of data reduces the processing overhead in the MCU and is faster. The ACK is transmitted after the Frame Check Sequence (FCS) has been successfully verified.

To provide receiver signal strength feedback within the ACK, three unused bits in the FCF are used. However, to be able to generate the ACK early on in the receiving process, the receiver signal strength has to be known early as well. Therefore, the RSSI_VAL register of the CC2420 is queried while generating the ACK.

In a laboratory experiment with two Tmote Sky nodes, a measurement of the ACK time for different block sizes and frame lengths shows that shorter frames take longer to acknowledge. This is because reading bytes over SPI is significantly faster than receiving them over the air. Hence, the MCU can make up time that was lost while generating the ACK while reading the frame. Longer frames yield more potential for making up time and hence the ACK can be sent out quicker. Block sizes of 4 and 6 bytes show the lowest average ACK time, with 6 bytes being slightly faster. Comparing the ACK time to hardware-generated ACKs reveals that the software is $140\,\mu$s to $225\,\mu$s slower but still within the requirements of the IEEE 802.15.4 standard. The rate of lost or unrecognised ACKs was measured to be 0.00 % when using hardware-generated ACKs and 0.01 % to 0.08 % when using software generated ACKs due to Contiki's cooperative multitasking environment. The rate of unacknowledged unicast packets when using the GINSENG TDMA layer is 0.022 % to 0.032 % when using hardware ACKs and 0.052 % to 0.063 % when using the software-generated ACKs.

3.7 Demonstration / Outreach

The GINSENG approach has been demonstrated in the booth of the state of Lower Saxony at the world's largest ICT event CeBIT in Hannover, Germany. The same demonstrator was submitted as a demo to the "Ninth International Conference on Networked Sensing Systems" [10] (see page 63) and won the best demo award.

The demonstrator is a vertical 2.5 m^2 plate of acrylic glass with a replication of a very much simplified version of an industrial process. The demonstrator visualizes the industrial context of GINSENG by demonstrating its key functionality. Furthermore, the demonstrator allows visitors to interact with the process that contains tanks, pipes, valves and pumps. Despite two motorized vales, the process contains a manual valve that visitors can use to modify process parameters. The demonstrator contains several wireless control loops that are closed in the sink node. Performance Debugging visualizes the performance of the network, both in terms of timeliness and end-to-end packet loss. The GINSENG middleware receives data from the network via the Dispatcher and visualizes process parameters on a screen.

3.8 Conclusion

GINSENG is a system for wireless real-time monitoring and control of industrial facilities based on WSN hardware and provides guarantees in terms of reliability and timeliness of data transport. In this work, the GINSENG application scenario as well as the testbed in a live oil refinery on Portugal has been outlined and a system for remote experimentation on the testbed has been shown. Furthermore, an overview over the GINSENG stack (including performance debugging and the coupling between the WSN and back end systems) has been given and results from several experiments conducted in the refinery have been presented. The results confirm that the general approach of GINSENG can fulfil industrial requirements.

The original GINSENG system is based on a manual approach for TDMA schedule determination that involves expert knowledge. The contribution of the author of this thesis is a solution for automatic schedule calculation. The system can further guarantee that no end-to-end packet loss occurs if the behaviour of links stays within values observed during an initial measurement period. To leverage additional potential for optimization, a localized dynamic transmission power control approach is further part of the contribution.

To allow for long-term data collection in real-time environments, two approaches to persistent data storage have been presented. On the one hand there is a ring file system for continuous real-time data logging. On the other hand, a FAT file system for compatibility with standard PC operating systems has been presented. Also an approach for reception quality feedback piggybacked onto link-layer ACKs has been discussed. The GINSENG system has further been demonstrated at the CeBIT in Hannover.

4 Delay-Tolerant Wireless Sensor Networks

In the Internet, we are used to continuous end-to-end paths between two nodes and mostly use TCP and User Datagram Protocol (UDP) for data transmission. Those protocols are ubiquitous today because they “just work” and are good enough for the job at hand. However, if the round-trip time in a network increases significantly, the protocols or the applications on top of them will usually fail. Originally from interstellar communication, DTNs are built around the concept of message-oriented, hop-by-hop data transmissions and on-node storage to overcome intermittent connectivity and potentially long delays.

In WSN research, a surprisingly high number of publications actually use DTN technology, often without explicitly saying so. ZebraNet [62], SeNDT [63], LUSTER [64], Seal-2-Seal [65] and Vineyard Computing [66] are just some examples. Unfortunately, all of those publications and many more are based on proprietary protocols and are not interoperable. Furthermore, implementations are often purpose-built and highly optimized and therefore cannot be reused. However, the history of the Internet has taught us that eventually standard protocols take over if resources allow. This eases implementation hurdles as well as it allows interoperability and seamless network integration and is hence desirable.

The number of DTN-related WSN publications implies that a need for DTN technology in WSNs exists. In the scope of this thesis, WSNs employing DTN technology are referred to as Delay Tolerant Wireless Sensor Networks (DTWSNs). In this section, two exemplary use-cases for delay-tolerant communication in a WSN are discussed and DTWSNs are introduced. Furthermore, the question of how to achieve reliability in such networks is discussed and a DTN implementation for Contiki is presented. This is accompanied by a discussion of flow control issues when two nodes make contact.

The discussion in this chapter is primarily based on the following publications: [8, 9, 14] (see pages 65, 69 and 67 respectively).

4.1 Application Scenarios

An overview paper containing information about the application scenario and more details in this section has been published in “Proceedings of the Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems” [8] (see page 65).

A typical example for a DTWSN use-case is the “Data Elevator” [8] in which a temperature sensor is installed in a remote location (the roof of a 15 story building). The temperature readings shall be collected in a laboratory in the 3rd floor of the building where they are processed to obtain statistical long-term information. Installing wires is out of the question and direct wireless communication is equally impossible because of steel reinforced concrete structures. However, the elevator of the building regularly moves through all floors of the building, whereas the movement cannot be predicted or influenced. In any case, the elevator is an ideal Mobile Ubiquitous LAN Extension (MULE) that can be exploited to transport the temperature readings. While the delay (between sampling and delivery) of the data is of minor importance (hours and even days are acceptable), all readings should eventually arrive in the laboratory for statistical reasons.

Another DTWSN use-case is a Delay Tolerant Wireless Body Area Network (DTWBAN) [67] in which a person is equipped with a 3-axis accelerometer. To detect a fall of an elderly person living alone at home, the node continuously samples accelerometer data and sends the information to a nearby base station. The received data is analysed and an automated emergency message is send if a fall is detected. In today's systems, if the wireless link is obstructed (which happens regularly in a DTWBAN), data gets lost and a potential fall may not be detected. Other patients need gait analysis to assess the individual fall risk of that person and to take appropriate countermeasures before a fall occurs. For this use-case, the accelerometer is regularly sampled and data is written to an SD card. The card is regularly put into a PC for offline analysis. Both use-cases share in common that a complete data set (no lost packets) is crucial for analysis. Gait analysis can handle a significant delay because data is used for long-term analysis. Fall detection cannot handle delay of many hours but delays of several seconds or even minutes are manageable, when reliability can be guaranteed.

4.2 Delay Tolerant Wireless Sensor Network

DTN [68] is a generic architecture for challenged networks to deal with intermittent connectivity, high bit-error rates and large delays. Originally coming from interstellar communication [69], DTNs are used in many different network types. The "store, carry and forward" approach not only allows overcoming long and potentially unpredictable network disruptions but also allows leveraging existing physical movement in the environment to move data through the network. The de-facto standard in delay-tolerant communication is the BP [70]. The BP is an address-centric, message-based protocol that is designed to form an overlay network over existing Internets. Data is transported in self-contained messages of almost arbitrary size that are called bundles, which expire after the lifetime has elapsed. So-called CLs are used to transport bundles over heterogeneous underlay technologies such as TCP and UDP over Internet Protocol (IP). The BP is flexible and is composed of variable length header fields (so-called Self-Delimiting Numeric Values (SDNVs)) and optional extension blocks. Addressing is either done via string-based Endpoint Identifiers (EIDs) in Uniform Resource Identifiers (URI) [71] format or via SDNVs in Compressed Bundle Header Encoding (CBHE) [72]. Routing is also part of the bundle layer and is often a modification of flooding [73] in which multiple copies of a bundle take different paths through the network. The BP optionally supports end-to-end ACKs for end-to-end reliability. Furthermore, custody transfer allows making individual nodes responsible for the delivery or forwarding of a bundle.

Running the BP on WSN nodes creates a number of unique challenges: The low computational resources of the nodes, small RAM and ROM sizes as well as comparably small radio frame sizes, tight energy budgets of nodes and inherently unreliable radio links. The following sections discuss how these challenges can be solved on off-the-shelf WSN hardware.

4.3 Reliability in Delay Tolerant Wireless Sensor Networks

Reliable end-to-end data delivery in a DTWSN is an important requirement posed by various application scenarios. While the bundle protocol features end-to-end reliability features such as delivery reports, using end-to-end retransmissions significantly increases network load and is hence not desirable. Instead, reliable hop-by-hop delivery of bundles has to be used to make sure that a node only deletes a bundle when another node has received it. The BP defines custody transfer allowing the custodian of a bundle to hand over custody for a bundle to another node. Eventually, custody is handed over to the destination of the bundle. While custody transfer ensures that a bundle is not simply deleted, hardware failures and the like cannot be compensated. Furthermore, the BP specification contains many blind spots with regard to custody transfer. It is rarely used in practice and often argued about[1]. Consequently, neither end-to-end retransmissions nor custody transfer are efficient ways to establish reliability in a DTWSN.

Reliability between two nodes can be achieved by designing the CL in a way that packet loss is detected and compensated. However, even if reliable hop-by-hop transfer is established, bundle loss

[1] http://www.ietf.org/mail-archive/web/dtn-interest/current/msg04337.html

may still occur due to one of the following reasons:

- *Bundle corruption* could invalidate data. The BP specification leaves it up to the CLs to ensure uncorrupted bundle transport.
- *Bundle storage is insufficient* and bundles have to be discarded. In this situation, either the network is overloaded (see the discussion on network capacity in Section 4.4.2) or the routing protocol is aggressively sending bundle copies (which essentially overloads the network).
- *Nodes reboot* and loose non-persistently stored bundles. This can be easily overcome by requiring persistent storage of bundles on all nodes.
- *Hardware failure* makes nodes unusable and thus the bundles stored on those nodes inaccessible. Since hardware failure of cheap WSN hardware cannot be avoided, bundle copies and multi-path delivery have to be employed.
- *Network splits* may not loose bundles permanently but make them undeliverable. While those bundles are technically not lost, they will not be delivered either. With multi-path delivery, this problem can be overcome.
- *Too short bundle lifetimes* may expire bundles on their way to the destination. Careful network capacity and bundle lifetime planning can help to avoid this issue.

Bundle delivery reliability can be ensured if all of those problems can be neglected. While flood routing can solve the problem of multi-path delivery, the following sections deal with CL design (see Section 4.4), flow control issues (see Section 4.4.1), BP implementation design (see Section 4.5), network capacity (see Section 4.4.2) and provide experimental results (see Section 4.6).

4.4 IEEE 802.15.4 Convergence Layer

Information about the CL has been published in “Proceedings of the Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems” [8] (see page 65).

The IEEE 802.15.4 CL bridges the gap between the BP and the IEEE 802.15.4 MAC. Existing layers 3 and 4 in the network stack are avoided to minimize overhead and bundles are transported directly inside IEEE 802.15.4 MAC frames with a 1 byte CL header. This means that the CL has to realize certain functions of layers 3 and 4.

IEEE 802.15.4 data frames have a maximum size of 127 bytes while a typical payload size is 116 bytes. The 1 byte CL header is used for dispatching, multiplexing and sequencing. Bundles exceeding 115 bytes are split up into multiple segments transmitted to the next hop and reassembled there to avoid the significant overhead of BP fragmentation. Flow control in the CL is done via explicit ACK messages as shown in the next section. Negative Acknowledgement (NACK) messages are sent out to signal either permanent or temporary (due to overload) rejection of a single bundle. Rejecting one segment of a bundle rejects the whole bundle.

When comparing the communication overhead, BP over IEEE 802.15.4 has a total header length (including IEEE 802.15.4) of 31 bytes in a typical setting, which leaves 96 bytes of payload for a bundle transmitted in a single radio frame. UDP over 6LoWPAN is the de-facto standard for communication in WSNs and has a total header length of 25 bytes in a typical setting, which leaves 102 bytes of payload in a single radio frame. For implementation reasons, the 6LoWPAN implementation in Contiki only supports 95 bytes of payload before starting to fragment. This shows that the communication overhead for BP over IEEE 802.15.4 is slightly higher but manageable when compared to UDP over 6LoWPAN.

The BP has a header of variable length by using SDNVs that are computationally expensive to encode and decode. However, comparing the time it takes to parse one incoming UDP/6LoWPAN packet and a bundle shows that the two are comparable. For smaller payload sizes, 6LoWPAN is slightly faster, but for payloads of 70 bytes parsing of both protocols takes almost the same time. From a computational perspective, those two protocols are comparable.

To enable seamless integration of nodes using the IEEE 802.15.4 CL into a "conventional" DTN, the CL has been implemented for the Linux-based BP implementation IBR-DTN [19]. IEEE 802.15.4 radios can be attached via USB or SPI using the linux-zigbee[2] stack.

4.4.1 Flow Control

A discussion of the flow control issues for the IEEE 802.15.4 CL has been published in "Proceedings of the seventh ACM international workshop on Challenged networks" [14] (see page 67).

IEEE 802.15.4 uses stop-and-wait as flow control. A sender sends out a data frame and waits for the link-layer ACK. If no ACK is received in time, the frame may be retransmitted. However, as shown in Section 3.6.4, the link-layer ACKs are generated by the radio hardware. The software running on the MCU is not involved in this process. Consequently, an ACK only confirms reception and not successful processing of a packet. To avoid that fast senders overrun slow receivers, flow control is necessary. Especially with flash storage taking up to 5 ms to program a single flash page on the Tmote sky, a sender can easily send packets faster than a receiver can store them.

In DTWSNs, low communication overhead as well as low energy consumption is very important. However, for a BP CL throughput is another important metric to make good use of potentially short contacts. The goal is to maximize throughput while minimizing overhead and thereby energy consumption. The time between the ACK of a packet and the beginning of the next data frame is called t_D and should not be lower than the processing necessary on the sender (t_S) and on the receiver (t_R). Formally, the best throughput can be achieved when $t_D = max(t_R, t_S)$.

The following four approaches have been evaluated on Tmote Sky nodes:

- *Fixed Delay* waits a fixed time between two frames. This minimizes the energy consumption (no additional overhead, no retransmissions) but shows a low throughput.
- *Application-layer ACKs* are send after an incoming frame has been processed (and stored). This turns out to be optimal in terms of throughput but has significant overhead for the additional signalling.
- *Receiver-Feedback + Estimation* is based on a processing time estimation piggybacked onto the link-layers ACKs. This is among the optimal approach in terms of overhead. Practical limitations reduce the throughput to be slightly below the application-layer ACK approach.
- *TCP-inspired* is inspired by TCP congestion control and increases the bundle rate as long as link-layer ACKs are received. The receiver stops sending ACKs, when buffers are overrunning and the sender then decreases the sending rate. This approach shows medium performance and low overhead.

All measurements have been conducted with COFFEE-based bundle storage on a Tmote Sky as this represents the worst-case in terms of time needed at the receiver. Storages based on RAM, RATFAT or RFS would show a significantly higher throughput. The results show an application-layer throughput of up to 507.3 bytes/s for a payload size of 80 bytes when using the receiver-feedback approach, whereas the application-layer ACK approaches reaches up to 547.1 bytes/s. Please note that the low absolute throughput is caused by COFFEE taking up to 225.8 ms to create a file for a received bundle. The total amount of transferred bytes is 55 000 bytes for the receiver-feedback approach and 60 000 bytes for the application-layer ACK approach when sending 500 bundles with 80 bytes payload each. This means that the total overhead is 15 000 bytes (or 30 bytes/bundle) for

[2] `http://sourceforge.net/apps/trac/linux-zigbee/`

the receiver-feedback approach whereas the application-layer ACK approach has a total overhead of 20 000 bytes or 40 bytes/bundle.

Although the receiver-feedback approach was expected to be the best trade-off between overhead and throughput, a practical limit on the size of the feedback information make the time estimation too coarse to be of use. To avoid sending too early, the sender has to be conservative and this reduces throughput. Hence, the best approach is using the explicit application-layer ACKs for the IEEE 802.15.4 CL.

4.4.2 DTN Capacity

A discussion of the capacity of DTWSNs has been published in “Proceedings of the Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems” [8] (see page 65).

Even with proper CL design and flow control mechanisms, bundles in a DTWSN can get lost if the available storage space is insufficient. While flow control and CLs can help to prevent nodes from flooding other nodes with bundles, a too high network load has to be avoided to prevent bundle loss. To inspect the capacity of a DTWSN, a capacity model has been published in [8].

In a DTN without multiple paths between source and destination, the capacity of the network is essentially the minimum capacity of all links. The link capacity is determined by the duration and frequency of contacts as well as the bundle throughput. More, longer or higher throughput contacts mean more potential for exchanged data. Consequently, the data path between two neighbours of a DTN can be modelled as a bottleneck that can only transmit a certain amount of bundles per time interval.

The data elevator scenario introduced in Section 4.1 uses a sampling interval of 300 s for temperature data. Given a throughput of 72 bundles/s between two nodes, a storage capacity of up to 100 bundles and the contacts observed during two days, the model predicts a packet loss of 0 %. However, decreasing the sampling interval to 240 s produces an end-to-end bundle loss of 1.39 % due to overrunning buffers. With hypothetically infinitive storage capacity and a sampling interval as low as 10 s all bundles would still be delivered because the capacity of the links is not the bottleneck.

This shows that storage capacity is of major importance when setting up a DTWSN.

4.5 μDTN

μDTN is a BP implementation for Contiki OS and has been published in “Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze” [9] (see page 69).

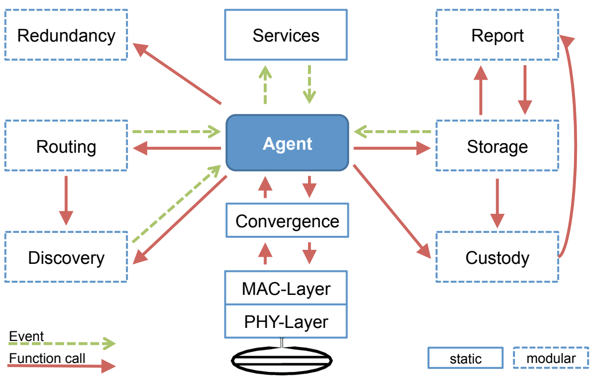

Figure 4.1: Block diagram of μDTN’s architecture

μDTN is designed and implemented for Contiki OS and therefore runs on many different hardware platforms. The modular architecture allows configuring modules at compile time, making the design flexible and efficient. During implementation, memory efficiency was considered more important than throughput because RAM is tightly limited on WSN platforms. μDTN is implemented as a Contiki protothread [61] and supports bundles in CBHE-format only. It is designed around the IEEE 802.15.4 CL and uses a 1 byte header inside the radio frame for signalling and dispatching. μDTN's different modules are shown in Figure 4.1 and the most important modules are described in the following.

The "Agent" is the central component that handles event-based communication between modules, applications (services) and the underlying layers. "Discovery" is done via the IEEE 802.15.4 CL in DTN IP Neighbor Discovery (IPND) format [74] via periodic broadcasts. "Routing" is done via constrained flooding, where bundles are send to a maximum of n neighbours and deleted afterwards. Bundles are not send back to the originator and to the peer the bundle was received from. Bundle "Storage" can be done either in volatile but fast RAM or in persistent but slow flash. A storage implementation based on FAT (see Section 3.5.2) is in preparation. The "Redundancy" module determines if bundles are duplicates and discards multiple copies of the same bundle. The "Convergence" layer is responsible for radio communication via the Contiki network stack. The "API" is event-based and allows multiple Contiki protothreads to register for incoming bundles and to send outgoing bundles.

4.6 Performance Results

Evaluating a DTN implementation without assuming a specific application scenario is primarily focused on hop-by-hop throughput [17, 16]. As seen in Section 4.4.2, the throughput between nodes is one of the influencing factors for overall network capacity. In a lab setting, μDTN produced a throughput of 74.34 bundles/s with 80 bytes of payload on a single-hop link between two nodes when using RAM storage. The round-trip time was measured to be 23.13 ms in the same setting. In a simulated store, carry and forward scenario with two nodes and a MULE in between, μDTN delivered 1000 bundles without bundle loss.

The data elevator use-case has been evaluated using INGA nodes installed on the roof, on the top floor, in the elevator cabin and next to the elevator on the 3rd floor. A Linux-based node running IBR-DTN has been installed in a laboratory on the 3rd floor and temperature measurements have been taken every 300 s. Results from a day on the weekend and a workday show no bundle loss while the storage of one node filled to 80 out of a possible 100 bundles at one point. The maximum end-to-end delay was measured to be 11 930 s on the workday and 23 591 s on the weekend. Those high delays occur primarily during night time when no one is using the elevator. On the workday, 59.03 % of the bundles have been delivered within 5 minutes while 50 % of the bundles have been delivered within 25 minutes on the weekend.

4.7 Conclusions

A significant number of WSN research projects are using DTN technology without explicitly saying so. Clearly, Delay Tolerant Wireless Sensor Networks (DTWSNs) are needed for certain WSN projects in which delay is secondary but reliability is of high importance. In this work, two DTWSN application scenarios have been described and the challenges towards reliable end-to-end data transmission have been outlined. Furthermore, design considerations for a BP implementation and an IEEE 802.15.4-based CL have been discussed. Experimental results show the potential of DTNs in WSN context by delivering 100 % of the bundles by means of an elevator as data MULE.

5 Conclusions

WSN technology is used in various application scenarios and reliable end-to-end data transmission is a concern for most applications. In real-time networks, data has to be delivered before a deadline as well as with a minimum PDR to enable applications to work within their specifications. In delay-tolerant networks, delay is not a concern but PDR is important. Within this thesis, both network types have been discussed in detail and approaches to ensure reliable network operation have been presented.

GINSENG was an EU FP7 research project that focused on performance controlled WSNs for industrial applications. Based on the GINSENG use-case, the basic system design for a reliable real-time WSN has been discussed in this thesis. GINSENG is able to fulfil application requirements given a TDMA schedule that is able to handle the specific loss behaviour of the employed wireless links. Even in harsh conditions found in typical industrial facilities, more than 99 % of the data packets can be delivered in time with the proper schedule.

The contribution of this thesis spans over multiple building blocks of the GINSENG system. This is not limited to parts of the software on the nodes but also includes a *real-time capable serial interconnection* between the WSN and the back end network. Also the *remote reprogramming facilities* of the GINSENG testbed in a live industrial facility in Portugal are part of the contribution and allow running experiments remotely. Furthermore *performance debugging* allows pinpointing problems in the network.

The main contributions however are the mechanisms to calculate TDMA schedules and how transmission power (and closely connected inter-network interference) can be controlled. By gathering data about link burstiness and by *programmatically determining TDMA* schedule for real-time networks, both reliability and timeliness of data delivery can be ensured. Taking transmission power into account allows *reducing interference* between neighbouring networks to *improve reliability*. To leverage the potential of localized, short-term variations in link characteristics, *online transmission power control* is employed. Probing links at various transmission power levels ensures that reliability and timeliness can be maintained while further reducing inter-network interference. *Real-time file system support* for WSN nodes as well as *link-quality feedback in link-layer ACKs* are important building blocks in the software and networking stacks of the employed WSN Operating Systems that help to realize such reliable networks.

The presented solutions have been evaluated in a refinery as well as in university laboratories. The experiments have shown that programmatically determining TDMA schedules after initial data gathering *ensures reliable network operation* and allows *reducing interference between neighbouring networks* significantly. The approach *reduces energy consumption* slightly. Transmission power control makes use of temporary optimization potential and allows *reducing interference even further* using purely local information.

Delay Tolerant Wireless Sensor Networks are realized by adapting standard DTN protocols to the WSN environment. Using the original BP in conjunction with an efficient *IEEE 802.15.4 Convergence Layer* is the basis for the contribution and exposes little additional overhead compared to established WSN protocols. The *Bundle Protocol implementation μDTN* for WSN nodes and the respective counterpart on the PC side that is designed as an IBR-DTN component enable interoperability between DTWSNs and traditional DTNs. *CL flow control* is necessary to complement the existing mechanisms to ensure reliable and fast bundle transfer. A *DTWSN capacity model* allows estimating the maximum network capacity based on contacts and node abilities.

Simulations and experiments in an office building show that *DTWSNs can deliver all data* if the network capacity is not exceeded. Furthermore, the hop-by-hop *throughput of µDTN is comparable* to less reliable standard protocols such as UDP over 6LoWPAN. The *overhead is slightly higher* but the functionality significantly more advanced.

5.1 Future Work

Using the work presented in this thesis, schedules for real-time WSN can be calculated and potential for local optimization can be used. However, there is substantial potential for future work. The TDMA schedule could be recalculated regularly to cope with small and larger changes of the environment. After a certain amount of time, another topology may be better suited for the changing environmental conditions. A schedule management framework could be developed to continuously probe links during network operation and to deploy new schedules when necessary. Furthermore, different optimization criteria can be applied depending on the specific environment or the requirements of the application. Transmission power control is currently done on the basis of receiver signal strength, whereas signal to noise ratio is another important metric that could be considered in the future.

While the foundation for interoperable delay-tolerant wireless sensor networks have been presented in this thesis, significant potential for future work exists. The presented convergence layer will be published as an Internet Engineering Task Force (IETF) draft to allow others to comment and improve. Furthermore, node discovery can be realized in a more energy-efficient way by using scheduling techniques. Among the existing approaches DISCO [75] is well-known and PDS [76] is claimed to be optimal. Also, employing BP header compression approach similar to the 6LoWPAN approach can reduce the communication overhead of the bundle protocol even further. To optimize bundle storage on nodes both in terms of throughput but also capacity, a RAM and flash-based hybrid storage seems to be a viable alternative. Finally, to reduce network load, opportunistic in-network data aggregation can help to reduce the number of bundles. Also, adapting sampling intervals based on network conditions or remaining storage capacity may be investigated.

Bibliography

[1] Wolf-Bastian Pöttner, Hans Seidel, James Brown, Utz Roedig, and Lars Wolf. Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3), August 2014. accepted for publication.

[2] Tony O'Donovan, James Brown, Felix Büsching, Alberto Cardoso, José Cecilio, Jose Manuel do Ó, Pedro Furtado, Paulo Gil, Anja Jugel, Wolf-Bastian Pöttner, Utz Roedig, Jorge Sa Silva, Ricardo Silva, Cormac J. Sreenan, Vasos Vassiliou, Thiemo Voigt, Lars Wolf, and Zinon Zinonos. The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences. *ACM Transactions on Sensor Networks (TOSN)*, 10(1), November 2013.

[3] Wolf-Bastian Pöttner, Daniel Willmann, Felix Büsching, and Lars Wolf. All Eyes on Code: Using Call Graphs for WSN Software Optimization. In *Eighth IEEE Workshop on Practical Issues in Building Sensor Network Applications 2013 (IEEE SenseApp 2013)*, Sydney, Australia, October 2013.

[4] Wolf-Bastian Pöttner, Daniel Willmann, Felix Büsching, and Lars Wolf. Regression Testing Framework for WSNs. In *Demonstrations of the IEEE Conference on Local Computer Networks 2013 (LCN-Demos 2013)*, Sydney, Australia, October 2013.

[5] Sebastian Schildt, Wolf-Bastian Pöttner, Felix Büsching, and Lars Wolf. RATFAT: ReAl-Time FAT for Cooperative Multitasking Environments in WSNs. In *5th Workshop on Performance Control in Wireless Sensor Networks 2013 (PWSN 2013) in conjunction with IEEE DCOSS 2013*, Cambridge, USA, May 2013.

[6] Wolf-Bastian Pöttner and Lars Wolf. Probe-based Transmission Power Control for Dependable Wireless Sensor Networks. In *The 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013 (IEEE DCoSS 2013)*, Cambridge, USA, May 2013.

[7] Ulf Kulau, Felix Büsching, Wolf-Bastian Pöttner, and Lars Wolf. Demo: A Node's Life - Increasing WSN Lifetime by Dynamic Voltage Scaling. In *The 9th IEEE International Conference on Distributed Computing in Sensor Systems 2013 (IEEE DCoSS 2013)*, Cambridge, USA, May 2013.

[8] Wolf-Bastian Pöttner, Felix Büsching, Georg von Zengen, and Lars Wolf. Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks. In *The Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2012)*, Las Vegas, Nevada, USA, October 2012.

[9] Georg von Zengen, Felix Büsching, Wolf-Bastian Pöttner, and Lars Wolf. An Overview of μDTN: Unifying DTNs and WSNs. In *Proceedings of the 11th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN)*, Darmstadt, Germany, September 2012.

[10] Felix Büsching, Wolf-Bastian Pöttner, Dieter Brökelmann, Georg von Zengen, Robert Hartung, Karsten Hinz, and Lars Wolf. A Demonstrator of the GINSENG-Approach to Performance and Closed Loop Control in WSNs. In *Networked Sensing Systems (INSS), 2012 Ninth International Conference on*, June 2012. Best Demo Award.

[11] Sebastian Schildt, Wolf-Bastian Pöttner, and Lars Wolf. Contiki Ring File System for Real-Time Applications. In *4th Workshop on Performance Control in Wireless Sensor Networks 2012 (PWSN 2012) in conjuction with IEEE DCOSS 2012*, pages 364 –371, May 2012.

[12] Sebastian Schildt, Wolf-Bastian Pöttner, Oliver Ohneiser, and Lars Wolf. NASDI - Naming and Service Discovery for DTNs in Internet Backbones. In *Proceedings of the 5th International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, Mobilware 2012*, 2012.

[13] Sebastian Schildt, Till Lorentzen, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. Free-riding the BitTorrent DHT to improve DTN connectivity. In *Proceedings of the seventh ACM international workshop on Challenged networks*, CHANTS '12, pages 9–16. ACM, 2012.

[14] Wolf-Bastian Pöttner and Lars Wolf. Flow Control Mechanisms for the Bundle Protocol for the Bundle Protocol in IEEE 802.15.4 Low-power Networks. In *Proceedings of the seventh ACM international workshop on Challenged networks (CHANTS)*, CHANTS '12, pages 65–68. ACM, 2012.

[15] Wolf-Bastian Pöttner, Sebastian Schildt, Daniel Meyer, and Lars Wolf. Piggy-Backing Link Quality Measurements to IEEE 802.15.4 Acknowledgements. In *The 4th International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN-Fall 2011)*, Valencia, Spain, October 2011.

[16] Wolf-Bastian Pöttner, Johannes Morgenroth, Sebastian Schildt, and Lars Wolf. An Empirical Performance Comparison of DTN Bundle Protocol Implementations. Informatikbericht 2011-08, Technische Universität Braunschweig, September 2011.

[17] Wolf-Bastian Pöttner, Johannes Morgenroth, Sebastian Schildt, and Lars Wolf. An Empirical Performance Comparison of DTN Bundle Protocol Implementations. In *ACM MobiCom 2011 Workshop on Challenged Networks (CHANTS'11)*, Las Vegas, Nevada, USA, September 2011.

[18] Wolf-Bastian Pöttner, Lars Wolf, José Cecilio, Pedro Furtado, Ricardo Silva, Jorge Sa Silva, Amancio Santos, Paulo Gil, Alberto Cardoso, Zinon Zinonos, Jose Manuel do Ó, Ben McCarthy, James Brown, Utz Roedig, Tony O'Donovan, Cormac J. Sreenan, Zhitao He, Thiemo Voigt, and Anja Jugel. WSN Evaluation in Industrial Environments First results and lessons learned. In *3rd Workshop on Performance Control in Wireless Sensor Networks 2011 (PWSN 2011) in conjuction with IEEE DCOSS 2011*, June 2011.

[19] Sebastian Schildt, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. IBR-DTN: A lightweight, modular and highly portable Bundle Protocol Implementation. *Electronic Communications of the EASST*, 37:1–11, January 2011.

[20] Michael Doering, Wolf-Bastian Pöttner, Tobias Pögel, and Lars Wolf. Impact of Radio Range on Contact Characteristics in Bus-based Delay Tolerant Networks. In *Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS 2011)*, pages 195–202, Bardonecchia, Italy, January 2011.

[21] Michael Doering, Tobias Pögel, Wolf-Bastian Pöttner, and Lars Wolf. A New Mobility Trace for Realistic Large-Scale Simulation of Bus-based DTNs. In *ACM MobiCom 2010 Workshop on Challenged Networks (CHANTS 2010)*, Chicago, USA, September 2010.

[22] Wolf-Bastian Pöttner, Oliver Wellnitz, and Lars Wolf. QoS-AODV6E: An Energy-Balancing QoS Routing Scheme for WSNs. In *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS2010)*, Kassel, Germany, June 2010.

[23] Wolf-Bastian Pöttner and Lars Wolf. IEEE 802.15.4 Packet Analysis with Wireshark and off-the-shelf Hardware. In *Proceedings of the Seventh International Conference on Networked Sensing Systems (INSS2010)*, Kassel, Germany, June 2010.

[24] Sven Lahde, Michael Doering, Wolf-Bastian Pöttner, Gerrit Lammert, and Lars Wolf. A Practical Analysis of Communication Characteristics for Mobile and Distributed Pollution Measurements on the Road. *Wireless Communications and Mobile Computing*, 7(10):1209–1218, December 2007.

[25] Sven Lahde, Michael Doering, Wolf-Bastian Pöttner, and Gerrit Lammert. Mobile and Distributed Measurement of Air Pollution in Metropolitan Areas Using Car2X Techniques. In *Proceedings of 3nd Symposium on Informationssysteme für mobile Anwendungen (IMA)*, Braunschweig, October 2006.

[26] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A Survey on Sensor Networks. *Communications Magazine, IEEE*, 40(8):102–114, 2002.

[27] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler. Lessons from a Sensor Network Expedition. In *Wireless Sensor Networks*, volume 2920 of *Lecture Notes in Computer Science*, pages 307–322. Springer Berlin Heidelberg, 2004.

[28] Adam Dunkels, Björn Grönvall, and Thiemo Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In *Proceedings of the First IEEE Workshop on Embedded Networked Sensors (Emnets-I)*, Tampa, Florida, USA, November 2004.

[29] J. Hui and JP. Vasseur. The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams. RFC 6553 (Proposed Standard), March 2012.

[30] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. RFC 4919 (Informational), August 2007.

[31] IEEE. 802.15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). *IEEE Standard for Information technology*, 2007.

[32] Felix Büsching, Ulf Kulau, and Lars Wolf. Architecture and Evaluation of INGA - An Inexpensive Node for General Applications. In *Sensors, 2012 IEEE*, pages 842–845, Taipei, Taiwan, October 2012. IEEE.

[33] Peter Corke, Philip Valencia, Pavan Sikka, Tim Wark, and Les Overs. Long-Duration Solar-powered Wireless Sensor Networks. In *Proceedings of the 4th workshop on Embedded networked sensors*, EmNets '07, pages 33–37, New York, NY, USA, 2007. ACM.

[34] Moteiv Corporation. Tmote Sky Datasheet. `http://www.snm.ethz.ch/snmwiki/pub/uploads/Projects/tmote_sky_datasheet.pdf`, 2006.

[35] Texas Instruments Incorporated. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. `http://www.ti.com/lit/gpn/cc2420`, 2013.

[36] Atmel. AT86RF231, Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE, SP100, WirelessHART and ISM Applications. `http://www.atmel.com/dyn/resources/prod_documents/doc8111.pdf`, 2009.

[37] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-Time Forest Fire Detection with Wireless Sensor Networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1214–1217, 2005.

[38] Esteban Egea-López, Javier Vales-Alonso, Alejandros S. Martínez-Sala, Joang García-Haro, Pablo Pavón-Mariño, and M. Victoria Bueno Delgado. A wireless sensor networks MAC protocol for real-time applications. *Personal Ubiquitous Comput.*, 12(2):111–122, January 2008.

[39] Tian He, Pascal Vicaire, Ting Yan, Liqian Luo, Lin Gu, Gang Zhou, Radu Stoleru, Qing Cao, John A. Stankovic, and Tarek Abdelzaher. Achieving Real-Time Target Tracking Using Wireless Sensor Networks. In *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS '06, pages 37–48, Washington, DC, USA, 2006. IEEE Computer Society.

[40] M. Hanssmann, Sokwoo Rhee, and Sheng Liu. No Wiring Constraints. *Industry Applications Magazine, IEEE*, 15(4):60–65, 2009.

[41] International Society of Automation. ISA-100.11a-2009 Wireless systems for industrial automation: Process control and related applications. Available from the International Society of Automation, 2009.

[42] HART Communication Foundation. HART Communication Protocol Specication, HCF SPEC 13 Revision 7.1. Available from the HART Communication Foundation, June 2008.

[43] IEEE. Wireless Personal Area Networks: Proposal for Factory Automation. IEEE Proposed Standard 802.15.4-15/08/0571r0, August 2008.

[44] Tang Zhong, Cheng Mengjin, Zeng Peng, and Wang Hong. Real-time Communication in WIA-PA Industrial Wireless Networks. In *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, volume 2, pages 600–605. IEEE, 2010.

[45] Vasos Vassiliou, Zinon Zinonos, Christiana Ioannou, Marios Koutroullos, Vasco Pereira, Jian Li, Wolf-Bastian Pöttner, Ricardo Silva, James Brown, and Ben McCarthy. Final GINSENG Architecture, Scenarios and Quality of Service Measures. GINSENG Deliverable D1.3 `http://www.ibr.cs.tu-bs.de/projects/ginseng/deliverables/D1_3.pdf`, August 2010.

[46] Petcharat Suriyachai, James Brown, and Utz Roedig. Time-Critical Data Delivery in Wireless Sensor Networks. In *Distributed Computing in Sensor Systems*, volume 6131 of *Lecture Notes in Computer Science*, pages 216–229. Springer Berlin Heidelberg, 2010.

[47] Thanh-Dien Tran, Ricardo Silva, David Nunes, and Jorge Sa Silva. Characteristics of Channels of IEEE 802.15.4 Compliant Sensor Networks. *Wireless Personal Communications*, pages 1–16, 2011.

[48] Tony O'Donovan, Wolf-Bastian Pöttner, Zhitao He, Thiemo Voigt, and Cormac Sreenan. Refined Methods and Tools for Performance Debugging. GINSENG Deliverable D2.4 `http://www.ibr.cs.tu-bs.de/projects/ginseng/deliverables/D2_4.pdf`, February 2011.

[49] Tony O'Donovan, James Brown, Wolf-Bastian Pöttner, and Zhitao He. System Evaluation. GINSENG Deliverable D2.5 `http://www.ibr.cs.tu-bs.de/projects/ginseng/deliverables/D2_5.pdf`, February 2012.

[50] Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, and Zhitao He. Software-based On-line Energy Estimation for Sensor Nodes. In *Proceedings of the 4th workshop on Embedded networked sensors*, EmNets '07, pages 28–32, New York, NY, USA, 2007. ACM.

[51] J.L. Romkey. Nonstandard for transmission of IP datagrams over serial lines: SLIP. RFC 1055 (INTERNET STANDARD), June 1988.

[52] D. Mills, J. Martin, J. Burbank, and W. Kasch. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905 (Proposed Standard), June 2010.

[53] Jerry Zhao and Ramesh Govindan. Understanding Packet Delivery Performance In Dense Wireless Sensor Networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, New York, NY, USA, 2003. ACM.

[54] Sinem Coleri Ergen and Pravin Varaiya. TDMA Scheduling Algorithms for Wireless Sensor Networks. *Wirel. Netw.*, 16(4):985–997, May 2010.

[55] Sirajum Munir, Shan Lin, Enamul Hoque, S. M. Shahriar Nirjon, John A. Stankovic, and Kamin Whitehouse. Addressing Burstiness for Reliable Communication and Latency Bound Generation in Wireless Sensor Networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pages 303–314, New York, NY, USA, 2010. ACM.

[56] James Brown, Ben McCarthy, Utz Roedig, Thiemo Voigt, and Cormac J. Sreenan. BurstProbe: Debugging Time-Critical Data Delivery in Wireless Sensor Networks. In *Proceedings of the 8th European conference on Wireless sensor networks*, EWSN'11, pages 195–210. Springer-Verlag, 2011.

[57] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[58] Geoff Coulson, Barry Porter, Ioannis Chatzigiannakis, Christos Koninis, Stefan Fischer, Dennis Pfisterer, Daniel Bimschas, Torsten Braun, Philipp Hurni, Markus Anwander, Gerald Wagenknecht, Sandor P. Fekete, Alexander Kröller, and Tobias Baumgartner. Flexible Experimentation in Wireless Sensor Networks. *Commun. ACM*, 55:82–90, January 2012.

[59] Nicolas Tsiftes, Adam Dunkels, Zhitao He, and Thiemo Voigt. Enabling Large-Scale Storage in Sensor Networks with the Coffee File System. In *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2009)*, San Francisco, USA, April 2009.

[60] Microsoft Corporation. Microsoft EFI FAT32 File System Specification. `http://msdn.microsoft.com/en-US/windows/hardware/gg463084`, December 2000.

[61] Adam Dunkels, Oliver Schmidt, Thiemo Voigt, and Muneeb Ali. Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded Systems. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, pages 29–42, New York, NY, USA, 2006. ACM.

[62] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. *SIGOPS Oper. Syst. Rev.*, 36(5):96–107, October 2002.

[63] Paul McDonald, Dermot Geraghty, Ivor Humphreys, Stephen Farrell, and Vinny Cahill. Sensor Network with Delay Tolerance (SeNDT). In *Proceedings of 16th International Conference*

on Computer Communications and Networks, 2007. ICCCN 2007., pages 1333 –1338, August 2007.

[64] Leo Selavo, Anthony D. Wood, Qing Cao, Tamim I. Sookoor, H. Liu, Aravind Srinivasan, Yafeng Wu, Woochul Kang, John A. Stankovic, D. Young, and J. Porter. LUSTER: Wireless Sensor Network for Environmental Research. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, SenSys '07, pages 103–116. ACM, 2007.

[65] Anders Lindgren, Cecilia Mascolo, Mike Lonergan, and Bernie McConnell. Seal-2-Seal: A Delay-Tolerant Protocol for Contact Logging in Wildlife Monitoring Sensor Networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, October 2008.

[66] Jenna Burrell, Tim Brooke, and Richard Beckwith. Vineyard Computing: Sensor Networks in Agricultural Production. *Pervasive Computing, IEEE*, 3(1):38 – 45, January 2004.

[67] Felix Büsching, Maximiliano Bottazzi, Wolf-Bastian Pöttner, and Lars Wolf. DT-WBAN: Disruption Tolerant Wireless Body Area Networks in Healthcare Applications. In *the International Workshop on e-Health Pervasive Wireless Applications and Services (eHPWAS'13)*, Lyon, France, October 2013.

[68] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking Architecture. RFC 4838 (Informational), April 2007.

[69] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 27–34. ACM, 2003.

[70] K. Scott and S. Burleigh. Bundle Protocol Specification. RFC 5050 (Experimental), November 2007.

[71] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (INTERNET STANDARD), January 2005. Updated by RFC 6874.

[72] S. Burleigh. Compressed Bundle Header Encoding (CBHE). RFC 6260 (Experimental), May 2011.

[73] Andrew S. Tanenbaum and David J. Wetherall. *Computer Networks*. Pearson Prentice Hall, 2010.

[74] Daniel Ellard and Dan Brown. DTN IP Neighbor Discovery (IPND). *IRTF Draft*, 2010.

[75] Prabal Dutta and David Culler. Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 71–84, New York, NY, USA, 2008. ACM.

[76] Jó Ágila Bitsch Link, Christoph Wollgarten, Stefan Schupp, and Klaus Wehrle. Perfect Difference Sets for Neighbor Discovery: Energy Efficient and Fair. In *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, ExtremeCom '11, New York, NY, USA, 2011. ACM.

List of Acronyms

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks

ACK Acknowledgement

ADC Analog-to-Digital-Converter

API Application Programming Interface

ATEX Atmosphére Explosibles

BP Bundle Protocol

CBHE Compressed Bundle Header Encoding

CL Convergence Layer

CPU Central Processing Unit

CRC Cyclic Redundancy Check

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

DTN Delay Tolerant Network

DTWBAN Delay Tolerant Wireless Body Area Network

DTWSN Delay Tolerant Wireless Sensor Network

EID Endpoint Identifier

ETX Expected Transmission Count

EWMA Exponentially Weighted Moving Average

FCF Frame Control Field

FCS Frame Check Sequence

FIFO First In First Out

FP7 7th Framework Programme for Research and Technological Development

FSP File System Process

IETF Internet Engineering Task Force

INGA Inexpensive Node for General Applications

IoT Internet of Things

IP Internet Protocol

IPND DTN IP Neighbor Discovery

MAC	Medium Access Control
MBR	Master Boot Record
MCU	Microcontroller Unit
MIB	Management Information Base
MULE	Mobile Ubiquitous LAN Extension
NACK	Negative Acknowledgement
NAT	Native Address Translation
NTP	Network Time Protocol
OS	Operating System
PC	Personal Computer
PDR	Packet Delivery Ratio
PHY	Physical
PLR	Packet Loss Rate
PoE	Power over Ethernet
PRR	Packet Reception Rate
RAM	Random-access Memory
RFS	Ring File System
ROM	Read-only Memory
RPL	Routing Protocol for Low power and Lossy Networks
SCP	Secure Copy
SD	Secure Digital
SDHC	SD High Capacity
SDNV	Self-Delimiting Numeric Value
SLIP	Serial Line IP
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TI	Texas instruments
TPC	Transmission Power Control
UDP	User Datagram Protocol
URI	Uniform Resource Identifiers

USB	Universal Serial Bus
VPN	Virtual Private Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language
XON	Transmission On

Publications

The publications that are part of this cumulative dissertation are omitted in this PDF document due to copyright reasons. However, open-access links to the PDF documents as well as DOIs are given for all publications on the following pages.

Paper 1: The GINSENG System for Wireless Monitoring and Control: Design and Deployment Experiences

Status:	**appeared in ACM Transactions on Sensor Networks Volume 10, Issue 1, Nov. 2013**
DOI:	10.1145/2529975
URL:	`https://www.ibr.cs.tu-bs.de/papers/ginseng_journal.pdf`

Abstract

Today's industrial facilities such as oil refineries, chemical plants, and factories rely on *wired* sensor systems to monitor and control the production processes. The deployment and maintenance of such cabled systems is expensive and inflexible. It is, therefore, desirable to replace or augment these systems using wireless technology, which requires us to overcome significant technical challenges. Process automation and control applications are mission-critical and require timely and reliable data delivery, which is difficult to provide in industrial environments with harsh radio environments. In this paper we present the GINSENG system which implements performance control to allow us to use wireless sensor networks for mission-critical applications in industrial environments. GINSENG is a complete system solution that comprises on-node system software, network protocols, and back-end systems with sophisticated data processing capability. GINSENG assumes that a deployment can be carefully planned. A TDMA-based MAC protocol, tailored to the deployment environment is employed to provide reliable and timely data delivery. Performance debugging components are used to unintrusively monitor the system performance and identify problems as they occur. The paper reports on a real-world deployment of GINSENG in an especially challenging environment of an operational oil refinery in Sines, Portugal. We provide experimental results from this deployment and share the experiences gained. These results demonstate the use of GINSENG for sensing and actuation and allow an assessment of its ability to operate within the required performance bounds. We also identify shortcomings that manifested during the evaluation phase, thus giving a useful perspective on the challenges that have to be overcome in these harsh application settings.

Paper 2: WSN Evaluation in Industrial Environments First results and lessons learned

Status:	**appeared in Proc. of PWSN 2011**
DOI:	10.1109/DCOSS.2011.5982178
URL:	`https://www.ibr.cs.tu-bs.de/papers/poettner-pwsn2011.pdf`

Abstract

The GINSENG project develops performance-controlled wireless sensor networks that can be used for time-critical applications in hostile environments such as industrial plant automation and control. GINSENG aims at integrating wireless sensor networks with existing enterprise resource management solutions using a middleware. A cornerstone is the evaluation in a challenging industrial environment — an oil refinery in Portugal. In this paper we first present our testbed. Then we introduce our solution to access, debug and flash the sensor nodes remotely from an operations room in the plant or from any location with internet access. We further present our experimental methodology and show some exemplary results from the refinery testbed.

Paper 3: Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks

Status:	**accepted for publication in ACM Transactions on Sensor Networks Volume 10, Issue 3**
DOI:	-
URL:	`https://www.ibr.cs.tu-bs.de/papers/poettner-tosn-2014.pdf`

Abstract

Wireless sensor networks for industrial process monitoring and control require highly reliable and timely data delivery. To match performance requirements specialised schedule based medium access control (MAC) protocols are employed. In order to construct an efficient system it is necessary to find a schedule that can support the given application requirements in terms of data delivery latency and reliability. Furthermore, additional requirements such as transmission power may have to be taken into account when constructing the schedule. In this paper we show how such schedule can be constructed. We describe methods and tools to collect the data necessary as input for schedule calculation. Moreover, due to the high complexity of schedule calculation, we also introduce a heuristic. We evaluate the proposed methods in a real-world process automation and control application deployed in an oil refinery and further present a long-term experiment in an office environment. Additionally, we discuss a framework for schedule life-cycle management.

Paper 4: Contiki Ring File System for Real-Time Applications

Status:	**appeared in Proc. of PWSN 2012**
DOI:	10.1109/DCOSS.2012.57
URL:	`https://www.ibr.cs.tu-bs.de/papers/schildt-pwsn12-rfs.pdf`

Abstract

Wireless Sensor Nodes often need to store or cache measurement or operational data into some non-volatile memory. Many applications using low power sensor nodes are based on Contiki, which provides the COFFEE filesystem for data storage. As COFFEE introduces occasional long and hard to predict write delays, it can not be used in applications which have real time demands, such as sensor networks operating in industrial environments.

In this paper we present an efficient implementation of a flexible Ring File System for Contiki, that can be used in real-time applications. Read and write times are predictable and adherence to Contiki's filesystem API allows it to be used as a drop-in replacement for COFFEE in many applications.

Paper 5: RATFAT: ReAl-Time FAT for Cooperative Multitasking Environments in WSNs

Status:	**appeared in Proc. of PWSN 2013**
DOI:	10.1109/DCOSS.2013.70
URL:	`https://www.ibr.cs.tu-bs.de/papers/schildt-pwsn13-ratfat.pdf`

Abstract

Today, many sensor nodes are equipped with a microSD slot to provide a cost effective way to store large amounts of data. When using the FAT file system, data collected by a node can be easily read by a PC without the need for any special software or communication protocol. While several FAT implementations for microcontrollers do exist, they are not suited for real-time applications. For a WSN in an industrial scenario where a node needs to run a closed loop control program, logging to a non-real-time capable persistent storage system is not an option.

In this paper we present RATFAT, an efficient implementation of a flexible real-time capable FAT file system for Contiki, that can be used for applications requiring real-time guarantees.

Paper 6: Probe-based Transmission Power Control for Dependable Wireless Sensor Networks

Status: **appeared in Proc. of DCOSS 2013**
DOI: 10.1109/DCOSS.2013.71
URL: `https://www.ibr.cs.tu-bs.de/papers/poettner-dcoss2013-tpc.pdf`

Abstract

Dependable Wireless Sensor Networks for industrial process automation, monitoring and control applications have high demands regarding network reliability in terms of timely delivery and packet loss. In large industrial facilities, interference between neighboring networks is a major problem that can be mitigated by reducing the radio transmission power to the necessary minimum. However, reliability of the network should not be compromised even in spite of packet loss bursts.

In this paper we aim at keeping the link within certain burstiness bounds while still minimizing the transmission power. We present evaluation results showing that the burstiness changes with the receiver signal strength. Based on this observation we present two mechanisms for dependable wireless sensor networks and study the performance in our testbed.

Paper 7: Piggy-Backing Link Quality Measurements to IEEE 802.15.4 Acknowledgements

Status:	**appeared in Proc. of WiSARN-Fall 2011**
DOI:	10.1109/MASS.2011.92
URL:	`https://www.ibr.cs.tu-bs.de/papers/poettner-wisarnfall2011.pdf`

Abstract

In this paper we present an approach to piggyback link quality measurements to IEEE 802.15.4 acknowledgement frames by generating acknowledgements in software instead of relying on hardware support. We show that the software-generated ACKs can be sent meeting the timing constraints of IEEE 802.15.4. This allows for a standard conforming, energy neutral dissemination of link quality related information in IEEE 802.15.4 networks. This information is available at no cost when transmitting data and can be used as input for various schemes for adaptive transmission power control and to assess the current channel quality.

Paper 8: A Demonstrator of the GINSENG-Approach to Performance and Closed Loop Control in WSNs

Status:	**appeared in Proc. of INSS 2012**
DOI:	10.1109/INSS.2012.6240572
URL:	`https://www.ibr.cs.tu-bs.de/papers/buesching-inss2012.pdf`

Abstract

The overall goal of GINSENG is a wireless sensor network that will meet application-specific performance targets, and that will be proven in a real industry setting where performance is critical.

This demonstrator shows some of the key outcomes of the GINSENG project such as energy-efficient wireless real-time closed loop control, performance debugging, integration with ERP systems and usability on heterogeneous hardware. Furthermore, it allows the user to interact with the system.

Paper 9: Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks

Status:	**appeared in Proc. of MASS 2012**
DOI:	10.1109/MASS.2012.6502520
URL:	`https://www.ibr.cs.tu-bs.de/papers/poettner-mass2012.pdf`

Abstract

Delay Tolerant Networking (DTN) enables transfer of data where conventional network protocols fail to deliver data because no continuous end-to-end connectivity is available. While the Bundle Protocol (BP) has been established as the standard DTN protocol in many application areas, Wireless Sensor Networks (WSN) often use proprietary protocols with a subset of the BP features.

In this paper we use an exemplary application to demonstrate how the BP can be beneficial for many WSN-based projects. We show, how low-power sensor nodes can transport bundles by exploiting existing movement in the environment. More importantly we show, how 8-bit WSN nodes can seamlessly interact with standard BP implementations running on standard PCs.

Our application, a sensor that is installed on the roof of a 15-story building, is using an elevator to transport bundles carrying measured values to our lab. We analytically compare the BP to existing protocols for WSNs, evaluate our application scenario and give insight into principal limitations.

Paper 10: Flow Control Mechanisms for the Bundle Protocol in IEEE 802.15.4 Low-power Networks

Status:	**appeared in Proc. of CHANTS 2012**
DOI:	10.1145/2348616.2348630
URL:	`https://www.ibr.cs.tu-bs.de/papers/poettner-chants2012.pdf`

Abstract

Wireless Sensor Networks are networks formed by wireless sensor nodes that generally feature a low-power microcontroller and an IEEE 802.15.4 radio. Most Delay-Tolerant Wireless Sensor Networks in literature use proprietary protocols that are specifically designed for a single purpose. In this paper we explore, how the Bundle Protocol can be used on top of the IEEE 802.15.4 PHY and MAC layers, while avoiding overhead for additional layers in between. To pursue this, a Convergence Layer has to realize certain tasks that usually are dealt with in layers 3 and 4 of the OSI 7-layer protocol stack. We argue that flow control has to be an integral ingredient of the Convergence Layer and compare the performance of four different mechanisms using our Bundle Protocol implementation for Contiki.

Paper 11: An Overview of μDTN: Unifying DTNs and WSNs

Status:	**appeared in Proc. of FGSN 2012**
DOI:	-
URL:	`https://www.ibr.cs.tu-bs.de/papers/bueesching-fgsn2012.pdf`

Abstract

Wireless Sensor Networks nowadays are deployed in miscellaneous fields of application. Some use cases – especially those which include mobility – would benefit from a communication protocol that handles connection losses by design instead of assuming a continuous end-to-end connection. In this work, we describe the implementation of a Delay-Tolerant Network protocol for Wireless Sensor Networks, that is based on the Bundle Protocol specifications. The Bundle Protocol specifies a “store, carry and forward” network protocol that can be used in scenarios with changing network topologies and unstable links. With μDTN, we present a Delay-Tolerant Network implementation for Contiki OS which is compatible to Contiki’s network stack and is able to handle disruptions without packet loss. Furthermore, it is interoperable with Bundle Protocol implementations running on Linux.