Philipp Hurni

Traffic-Adaptive and Link-Quality-Aware Communication in Wireless Sensor Networks

Abstract: This paper is a summary of the main contributions of the PhD thesis published in [1]. The main research contributions of the thesis are driven by the research question how to design simple, yet efficient and robust *run-time adaptive* resource allocation schemes within the communication stack of Wireless Sensor Network (WSN) nodes. The thesis addresses several problem domains with contributions on different layers of the WSN communication stack.

The main contributions can be summarized as follows: First, a a novel run-time adaptive MAC protocol is introduced, which stepwise allocates the power-hungry radio interface in an *on-demand* manner when the encountered traffic load requires it. Second, the thesis outlines a methodology for robust, reliable and accurate *software-based* energy-estimation, which is calculated at network runtime on the sensor node itself. Third, the thesis evaluates several Forward Error Correction (FEC) strategies to adaptively allocate the correctional power of Error Correcting Codes (ECCs) to cope with timely and spatially variable bit error rates. Fourth, in the context of TCP-based communications in WSNs, the thesis evaluates distributed caching and local retransmission strategies to overcome the performance degrading effects of packet corruption and transmission failures when transmitting data over multiple hops. The performance of all developed protocols are evaluated on a self-developed real-world WSN testbed and achieve superior performance over selected existing approaches, especially where traffic load and channel conditions are suspect to rapid variations over time.

Dr. Philipp Hurni: E-Mail: hurni@iam.unibe.ch

1 Introduction

WSNs are autonomous networks consisting of a large number of inexpensive small electronic devices and are equipped with sensors to measure a wide range of environmental conditions (e.g., temperature, acceleration, relative humidity, pressure, oxygen concentration). For decades, measurements of physical values were carried out manually with bulky and costly equipment, a tedious, time-consuming and error-prone procedure. The convenient and automated manner of being able to gain real-time measurement data from all kinds of distant locations and in a high resolution is the key advantage that drives research and innovation on WSN technologies today.

2 Traffic-Adaptive Medium Access Control

Sensor nodes should consume as little energy as possible, since real-world sensor networks need to remain operable in an area of interest over several days, weeks or even years, given an initial battery charge, e.g., of two AA batteries. However, when keeping the radio and the onboard sensors permanently active, a typical sensor node drains out of energy after not much more than a couple of days. In the past decade, mechanisms have been developed to prolong the time a node can live on an initial energy charge, especially by designing energy-conserving communication protocols on the MAC and routing layer. Instead of attempting to balance between energy consumption and the provision of suitable Quality of Service (QoS) parameters, most of today's Energy-Efficient Medium Access (E^2-MAC) protocols have been designed with the only goal of minimizing the energy conservation. As a result, most of these protocols are able to deliver little amounts of data with a low energy footprint, however, introducing severe restrictions with respect to the achievable throughput and latency, and totally failing to adapt to varying traffic loads and changing requirements of the imposed traffic load. The gain in energy-efficiency hence comes at the cost of severely restrained maximum throughput, as well as massively increased end-to-end packet latency.

The first main contribution of the PhD thesis begins with thoroughly examining the design space of the six most frequently cited E^2-MAC protocols and their performance under variable traffic load. We then developed a tri-partite metric to measure and quantify the *traffic adaptivity*, formalizing a notion for the property of *traffic adaptivity*, for which a measurable definition has not yet existed before. Applying this metric to the selection of MAC protocols conveyed that the WiseMAC [2] yet achieves the best adaptivity under variable load, a conjecture that is also supported by the comparative study [3]. With the introduction of the *Maximally Traffic-Adaptive MAC (MaxMAC)* protocol [4], we specifically target at alleviating the performance degrading impact of most of today's E^2-MAC protocols with respect to traffic adaptivity. Relying on best practices of a decade of E^2-MAC protocol research, we base our investigations on adaptable protocol mechanisms on the most widely applied class of asynchronous random-access and preamble-sampling based MAC protocols. While MaxMAC operates with a low energy footprint at low traffic, it is able to dynamically adapt to sudden changes in the network traffic load at run-time. It integrates established design principles of asynchronous preamble-sampling based MAC protocols with novel run-time traffic adaptation techniques to allocate the costly radio transceiver truly in an *on demand* manner.

We evaluated MaxMAC in the OMNeT++[5] network simulator and with a real-world prototype implementation on the MSB430 nodes [6]. We first compared MaxMAC's against a selection of existing E^2-MAC protocols in simulation. By applying the developed metric to network simulation results of MaxMAC, we showed that the developed protocol mechanisms succeed in reaching a high *traffic adaptivity*. We then continued to study the real-world feasibility of the MaxMAC protocol by developing a real-world protocol prototype, and compared it against two other wireless channel MAC protocols: the WiseMAC protocol, and ScatterWeb[2] OS' energy-unconstrained CSMA. The experimental prototype-based evaluation relies on a series of small to medium benchmarking experiments and several experiments with different traffic patterns and networking conditions, carried out on the testbed facilities at University of Bern.

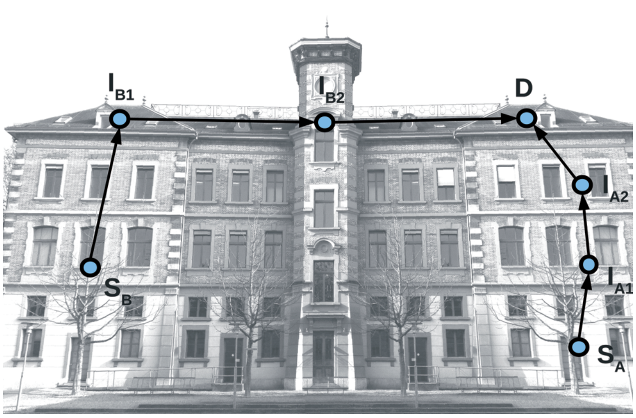

Fig. 1: Indoor Distributed Testbed Scenario.

In the following, we briefly tackle one example of the many examined scenarios: we evaluated the behavior of the three protocols CSMA, WiseMAC and MaxMAC with variable and contending traffic from different areas of the network using the entire V-shaped network consisting of 7 MSB430 nodes depicted in Figure 1. The sink node D is located in the top right corner and receives periodic *alive* status messages from nodes S_A, I_{A1}, S_B *and* I_{B1} each 20s. At some point, the nodes sense events and start sending 100 packets at a rate of 2 packets/s. There are overlaps of duration of 10s where two nodes are concurrently attempting to send their packets towards the sink. Since experimental evaluations of WSN mechanisms usually have a high variation, we ran the described experiment with 20 independent runs for each of the three MAC protocols. Figures 2, 3 and 4 depict the rate of received packets by the sink node D from each of the four source nodes. WiseMAC obviously manages well to deliver its periodic *alive* status messages to the sink. However, it suffers from major packet loss when the nodes have to transmit the 100 payload messages at a rate of 2 packets/s. Since packets have to be forwarded across multiple hops, the rather limited channel contention mechanism of WiseMAC and the hidden node problem lead to high packet losses. These are mainly caused by collisions and buffer overflows after failed transmission attempts. The rate of successfully delivered packets from the nodes S_A, I_{A1}, S_B, I_{B1} therefore does not exceed 1 packet/s on average, with the major share of packets being lost. In contrast to WiseMAC, CSMA and MaxMAC succeed in delivering the periodic *alive* status messages, but also the major share of the 100 packets which are triggered at a higher rate. The small time periods where two nodes are delivering their series of packets at the same time is managed best by CSMA. MaxMAC's rate of received packets reaches a slightly lower maximum throughput, and also tends to drop some packets when only one event is being handled.

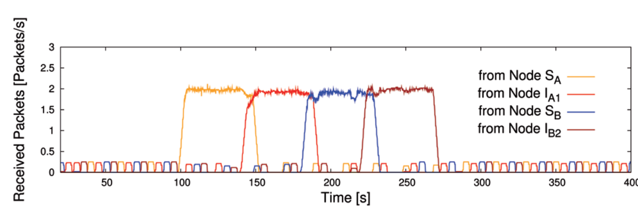

Fig. 2: CSMA: Packet Reception Rate from Nodes *SA*, I_{A1}, S_B and I_{B1}.

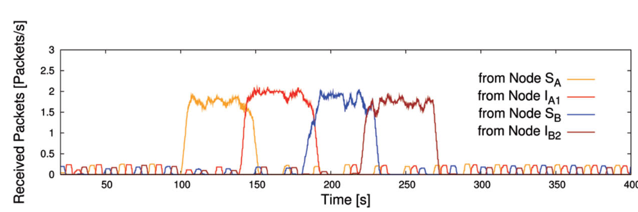

Fig. 3: MaxMAC: Packet Reception Rate from Nodes S_A, I_{A1}, S_B and I_{B1}.

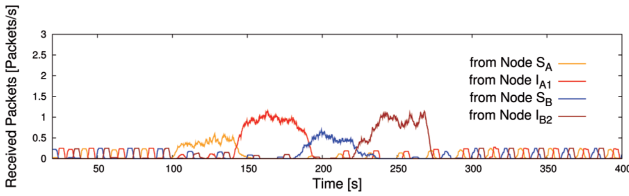

Fig. 4: WiseMAC: Packet Reception Rate from Nodes S_A, I_{A1}, S_B and I_{B1}.

All our experimental evaluations demonstrated that MaxMAC reaches its goal of being clearly distinguishable from existing preamble-sampling based approaches by reaching nearly the same throughput and a similarly low latency as energy-unconstrained CSMA, while still exhibiting the same energy-efficiency during periods of sparse network activity. The MaxMAC protocol hence succeeds in combining the advantages of energy unconstrained CSMA (high throughput, high PDR, low latency) with those of classical E^2-MAC protocols (high energy-efficiency). Like most contention-based MAC protocols, MaxMAC is a general-purpose protocol, and does not rely on assumptions that are cumbersome to achieve (e.g, rigid time-synchronization across the entire network). It can hence be applied without changes in scenarios where constant low-rate traffic is expected, and where in most cases B-MAC and X-MAC are being used today.

3 Software-based Energy-Estimation

While commonly used networking metrics such as packet delivery rate, source-to-sink latencies or maximum throughput can easily be determined in real-world WSN testbeds, measuring the power consumption of sensor nodes is much harder: costly high-resolution digital multimeters or cathode-ray oscilloscopes need to be hooked to the nodes to sample the varying low currents and voltages.

In the second main contribution of the PhD thesis [1], we develop a reliable and robust methodology for *software-based* energy estimation, cf. [7].

We implemented a simple *Three States Model* that estimates the energy consumption of a WSN nodes using the three states of the transceiver *receive*, *transmit* and *sleep*, a model which is used in most studies on WSN MAC protocols. To find optimal values for the estimation model parameters, we used a multivariate Ordinary Least Squares (OLS) regression model. By comparing physical measurements with the software-based energy-estimations computed on the nodes themselves, we observed that the model achieves a mean absolute estimation error (MAE) in the range of 3% and more compared to the *real* physically measured energy consumption. We then investigated further means to improve the estimation accuracy by enhancing the model basing on three state variables. By taking into account the transceiver switches and integrating them into our model, we could indeed significantly improve the estimation accuracy. We could reduce the MAE and its standard deviation (denoted as $\mu \pm \sigma$) of the software-based energy estimations to 1.13% ± 1.15%. Figure 6 illustrates the enhanced model's concept of a node's current draw. We refer to this model as *Three States Model with State Transitions* hereafter, as specified in equation (E1).

$$E = P_{rcv}T_{rcv}+P_{tx}T_{tx}+P_{slp}T_{slp}+\alpha s_{rcv+}\beta s_{tx+}\gamma s_{slp} \quad \text{(E1)}$$

According to this enhanced model, the energy consumed by an arbitrary node is a function of the total time it has its radio transceiver in the three different states (denoted as T_{rcv}, T_{tx}, T_{slp}) and the three adjustment terms αs_{rcv}, βs_{tx}, and γs_{slp}. The parameters α, β, γ compensate for the transceiver switches to the states *receive*, *transmit* and *sleep*. Applying even more sophisticated parameter calibration of per-node *and* per-protocol calibration using the multivariate OLS model has been shown to reduce the mean absolute error and standard deviation to as few as only 0.42% ± 0.72% across the four evaluated wireless channel MAC protocols S-MAC, T-MAC, WiseMAC, and IEEE 802.11-like CSMA.

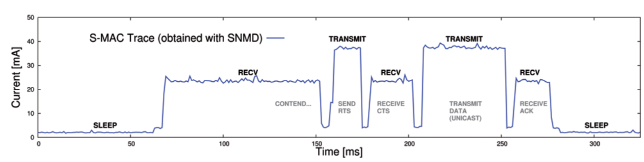

Fig. 5: Current Draw of *physically* measured node.

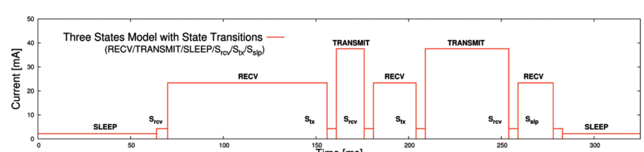

Fig. 6: Current Draw as modeled by our software-based energy estimation basing on the *Three States Model with State Transitions*.

4 Link-Quality-Aware Adaptive Forward Error Correction (FEC) Strategies

WSN nodes are typically preconfigured with the most crucial parameter settings at compile-time, hence much

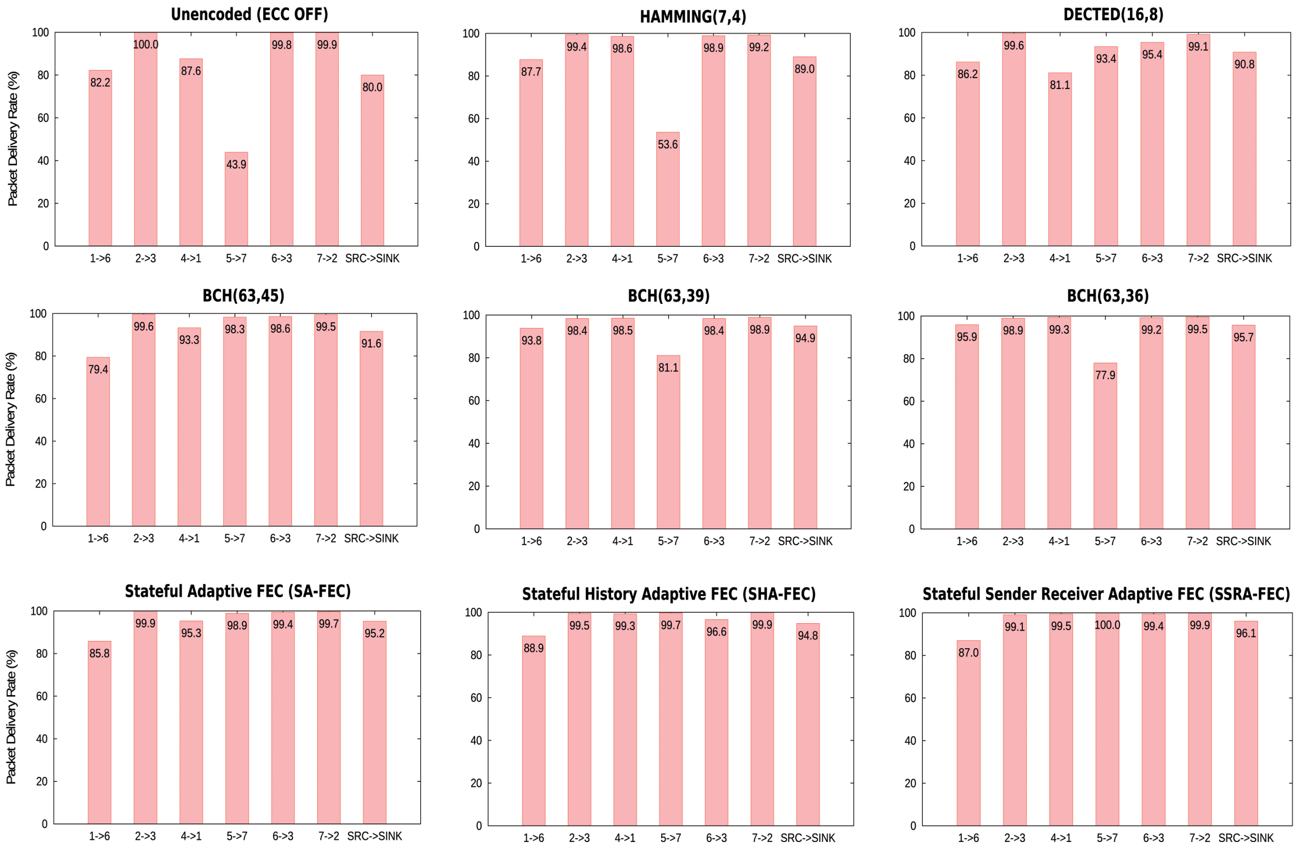

Fig. 7: PDRs per Link and Overall Source-to-Sink PDR.

before the actual network deployment. This can result in suboptimal performance in case the actually encountered environment differs much from the conditions expected during planning. When deciding to apply FEC, a crucial design question consists in selecting the appropriate Error Correcting Code (ECC). While a too weak code might not be able to correct many errors, a too strong code would waste precious time and energy for encoding/decoding and transmitting additional parity data.

The third main contribution of the PhD thesis [1] addresses the challenges related to the inherently unreliable wireless channel in WSNs by exploring the potential of run-time adaptive Forward Error Correction (FEC) schemes. We have implemented eight different ECCs and have proposed three run-time adaptive FEC strategies, which react to deteriorating link quality by allocating the correctional power of ECC codes in an *on demand* manner. The concept applied in this context shares many similarities to that of MaxMAC, cf. Section 2. The parameter adaptation algorithm follows a similar finite-state-machine-based model, where each state describes a certain set of parameters. Input variables such as the success of previous transmissions then define the state transitions, allocating less computational power when the link quality is good and unencoded transmissions are successfully acknowledged, and more sophisticated coding when link qualities are deteriorated.

Our three developed FEC adaptation strategies are referred-to as Stateful Adaptive FEC (SA-FEC), Stateful History Adaptive FEC (SHA-FEC) and Stateful Sender Receiver Adaptive FEC (SSRA-FEC) hereafter. SA-FEC selects the ECC for the next transmission to the specified destination according to the success of the last one. If the last transmission of an ECC packet has been successful, SA-FEC selects the next less powerful ECC. The second strategy SHA-FEC maintains a variable denoting the currently used ECC *and* a history of entries representing the recent past transmissions. In case a transmission succeeds and an ACK is received, SHA-FEC stores an integer value representing the next *lower* ECC into the history, since it assumes that a lower ECC would have sufficed as well. In case the transmission fails and no ACK is received, SHA-FEC stores a value representing the next *higher* ECC, assuming that more correctional power is necessary. The third strategy SSRA-FEC extends SHA-FEC by taking into account an additional history containing *receiver* information into the ECC selection process. Given that a packet payload consists of n blocks b_1, b_2, … b_n, the receiver calculates for each block the number of occurred and corrected errors $e_1, e_2, \ldots e_n$ during the decoding process. If all the blocks could be correctly decoded, the receiver sends $e_{max} = max\ (e_1, e_2, \ldots e_n)$ back to the sender in the ACK frame.

We have examined the different implemented ECCs as well as the adaptive FEC strategies in a series of experiments, of which the majority were conducted again on the distributed testbed facilities of University of Bern.

In the following, we outline the results of one particular experiment scenario, where each sensor node except the sink node generates and sends packets to its gateway node towards the sink node. We evaluated 1000 packets generated on each node for each run, and 20 runs for each configuration. The total amount of transmissions within the network amounts to roughly 2 transmissions per second, resulting in a channel utilization of roughly 14% across the entire testbed. With this level of channel utilization, interferences due to other ongoing transmissions are likely to occur, which may render the application of ECC to be a valuable countermeasure.

Figure 7 depicts the Packet Delivery Rate (PDR) bars of the examined ECC and the adaptive FEC approaches. The figure depicts for each examined setting the PDRs of the different links and the overall source-to-sink PDR. When comparing with the case of unencoded transmissions (top left corner of Figure 7), one can clearly see that the application of FEC made transmissions along the error-prone links more reliable, especially on $S_A \rightarrow I_{A1}$. The application of ECCs has clearly paid off with respect to alleviating the deteriorating impact of the lossy links $I_{B1} \rightarrow I_{B2}$ and $S_A \rightarrow I_{A1}$: the improvement in the achieved PDR is up to 15% of the total generated packets (cf. Figure 7, DECTED and BCH). Comparing the results of Hamming (7,4), DECTED(16,8) and the BCH-variants with the adaptive approaches, we can conclude that the adaptive FECs achieved astonishingly good results. The three strategies SA-FEC, SHA-FEC and SSRA-FEC outperformed almost every other static and network-wide setting of any of the implemented ECC codes. Since the adaptive schemes have only employed FEC on weak links and in periods of elevated BER, the majority of packets could be sent unencoded, which may also have led to fewer interference due to shorter transmission times, compared to static FEC settings. The major advantage of the adaptive approaches is the *on-demand* nature of using the correctional power of ECCs: with simple state-based concept, the adaptive approaches have managed to reach the same or better PDRs. Since the application of ECC comes at the cost of time and hence energy spent for encoding and decoding, ECCs should be limited to weak and error-prone links and/or time periods where the link quality suffers from deteriorating influences.

5 TCP Optimizations for Wireless Sensor Networks

TCP/IP has been shown to perform rather poorly [9] [10] in WSNs with multiple hops, due to the unreliable nature of the wireless channel (higher bit error rates and packet loss), particular properties of and interactions with the underlying wireless channel MAC protocols (exponential backoff mechanisms, hidden node and exposed node problem), and the design of the TCP congestion control mechanisms.

In the final contribution of the PhD thesis [1], we experimentally evaluate the performance of TCP/IP across multiple hops in WSNs, cf. [8] for more details. We take up basic concepts of distributed TCP caching and local retransmissions proposed in [9] [10] and self-developed extensions of the latter, and implement them – in contrast to the studies [9] [10] – in a MAC-layer independent manner into our *Caching and Congestion Control (cctrl)* module. The *cctrl* module developed within this experimental study is a modular add-on for the µIP stack [11] of the Contiki OS [12], which implements and augments the distributed caching and local retransmission features proposed in DTC [9] and TSS [10]. On top of the basic retransmission mechanism, we further designed and implemented additional extensions for our *cctrl* module. These extensions consist in monitoring the channel for activity on the link layer (activity monitoring) in order to defer retransmissions and in splitting the TCP connection in parallel streams (dual-connections). We show that our contribution is able to significantly increase the end-to-end throughput across multiple hops in various real-world WSN topologies. We study the performance of *cctrl* using three different E^2-MAC protocols (X-MAC, LPP, ContikiMAC) and Contiki's energy-unconstrained CSMA variant NullMAC. We test our implementations in an indoor wireless sensor node testbed in two scenarios where data has to be transmitted reliably across routes of increasing length.

Figure 8 is an excerpt from the results of numerous experiments conducted on our distributed testbed laboratories. In general, we encountered a rather high variability among the results of the different examined protocol configurations. Some protocols (e.g., NullMAC, LPP) generally reacted positively to the local retransmission scheme, whereas others (i.e., ContikiMAC) performed rather worse.

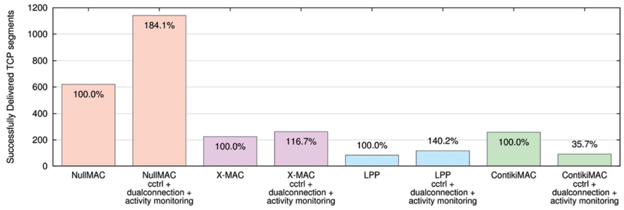

Fig. 8: PDRs per Link and Overall Source-to-Sink PDR.

The results reveal that the *cctrl* module can definitely increase the throughput of TCP across multi-hop WSNs in many of the examined situations. However, due to the differences in the examined MAC protocols algorithms, the impact of the proposed *cctrl* extensions varied heavily, such that it was impossible to find a single *cctrl* configuration that maximizes the throughput with all evaluated MAC layers. We observed that the results depend not only on MAC protocol characteristics, but also on the network topology and the presence of interfering traffic. The *cctrl* module managed to increase the end-to-end TCP throughput of NullMAC by up to 84% across routes with 2–6 hops lengths. X-MAC and LPP tended to exhibit improvements, whereas ContikiMAC's performance was significantly degraded. The best results with respect to the energy-efficiency (measured as radio on-time per TCP segment) were achieved with NullMAC in combination with our *cctrl* module. If a large portion of data has to be transmitted to a certain node across multiple hops, it is a better strategy to let the entire route temporarily operate without duty-cycling the radio, because the net radio on-time per transmitted TCP segment is lower than with any existing E^2-MAC protocol. This observation basically confirms and justifies the MaxMAC concept, which proposes to temporarily switch to CSMA if the encountered load conditions can not be handled anymore without major packet loss when sticking to the periodic radio duty-cycling pattern.

References

1 Philipp Hurni, "Traffic-Adaptive and Link-Quality-Aware Communication in Wireless Sensor Networks," *PhD Thesis submitted and accepted at University of Bern, Switzerland,* December 2011.

2 A. El-Hoiydi and J. D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks." International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS), Turku, Finland, July 2004, pp. 18–31.

3 K. Langendoen and A. Meier, "Analyzing MAC Protocols for Low Data-Rate Applications," *ACM Transactions on Sensor Networks (TOSN), New York, USA,* vol. 7, no. 2, pp. 1–40, August 2010.

4 P. Hurni and T. Braun, "MaxMAC: a Maximally Traffic-Adaptive MAC Protocol for Wireless Sensor Networks." European Conference on Wireless Sensor Networks (EWSN), Coimbra, Portugal, February 2010, pp. 289–305.

5 A. Varga, "The OMNeT++ Discrete Event Simulation System." European Simulation Multiconference (ESM), Prague, Czech Republic, June 2001, pp. 319–324. [Online]. Available: http://www.omnetpp.org.

6 M. Baar, E. Koeppe, A. Liers, and J. Schiller, "The ScatterWeb MSB-430 Platform for Wireless Sensor Networks." SICS Contiki Workshop, Kista, Sweden, March 2007.

7 P. Hurni, B. Nyffenegger, T. Braun, "On the Accuracy of Software-based Energy Estimation Techniques." European Conference on Wireless Sensor Networks (EWSN), Bonn, Germany, February 2011, pp. 49–64.

8 P. Hurni, U. Bürgi, T. Braun, and M. Anwander, "Performance Optimizations for TCP in Wireless Sensor Networks." European Conference on Wireless Sensor Networks (EWSN), Trento, Italy, February 2012.

9 A. Dunkels, J. Alonso, T. Voigt, and H. Ritter, "Distributed TCP Caching for Wireless Sensor Networks." Mediterranean Ad-Hoc Networks Workshop (Med-Hoc-Net), Bodrum, Turkey, June 2004, pp. 13–28.

10 T. Braun, T. Voigt, and A. Dunkels, "TCP Support for Sensor Networks." Wireless On demand Network Systems and Services (WONS), Obergurgl, Austria, January 2007, pp. 162–169.

11 A. Dunkels, "Full TCP/IP for 8-Bit Architectures." International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, USA, May 2003, pp. 85–98.

12 A. Dunkels, B. Groenvall, and T. Voigt, "Contiki – a Lightweight and Flexible Operating System for Tiny Networked Sensors." IEEE Workshop on Embedded Networked Sensors (EmNets), Tampa, Florida, November 2004, pp. 455–462. [Online]. Available: {http://www.sics.se/contiki/}.

Dr. Philipp Hurni: Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland