Towards an Open Source Implementation of the IEEE 802.15.4 DSME Link Layer

Maximilian Köstler*, Florian Kauer†, Tobias Lübkert‡ and Volker Turau§
Institute of Telematics, Hamburg University of Technology
Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany
*maximilian.koestler@tuhh.de, †florian.kauer@tuhh.de, ‡tobias.luebkert@tuhh.de, §turau@tuhh.de

***Abstract*—Reliable wireless solutions for large-scale automation are a major challenge today. The IEEE 802.15.4 standard forms the basis for many open and proprietary implementations. To reflect current state-of-the-art techniques, the IEEE has amended standard 802.15.4 with new MAC-layers such as TSCH, which resembles WirelessHART, and the Deterministic and Synchronous Multi-Channel Extension (DSME). This paper introduces *openDSME*, our implementation of IEEE 802.15.4 DSME. DSME aims at preventing packet collisions through slot reservation in networks where conventional CSMA/CA is not reliable enough. In this document, we will outline core features of DSME and *openDSME*, and present details of our implementation. Additionally, current research efforts on connected topics will be highlighted.**

***Index Terms*—Wireless Mesh Networks, IEEE 802.15.4 Link Layer, Reliability**

I. Introduction

Employing wireless networks in industrial environments helps to reduce costs and increases flexibility. Especially the installation costs can be reduced significantly [1]. The IEEE 802.15.4 standard has gained a lot of attention since it promises to allow for energy-efficient and low-cost wireless sensor and actuator networks. However, communication over a shared medium leads to the problem of message collisions when using contention based channel access methods such as Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA). Such collisions decrease the reliability of the network, especially in very large and dense networks [2]. Since reliability, however, is a fundamental requirement for industrial applications it is essential to provide a better coordination of the transceivers to decrease the number of packet collisions. The recent IEEE 802.15.4-2015 standard offers multiple possibilities to use Time Division Multiple Access (TDMA) schemes to access a range of channels. A promising extension presented in this standard is DSME that provides TDMA over multiple hops using a distributed slot allocation mechanism. We develop *openDSME* as an implementation of this standard including the associated services and interfaces[1]. This allows for the evaluation of DSME under realistic conditions. Instead of providing just another full-stack operating system for wireless sensor networks, *openDSME* provides convenient adaption layers to plug it into existing stacks for seamless evaluation of DSME in existing hard- and software environments. These layers have already been created for OMNET++/INET [3] as well as for CometOS [4].

[1]The development of *openDSME* takes place at http://www.opendsme.org/

A. Limitations of Standard IEEE 802.15.4

In standard IEEE 802.15.4 [5], devices primarily communicate via CSMA/CA. Additionally, communication with the network's Personal Area Network (PAN) coordinator may be managed using previously reserved slots, called Guaranteed Time Slots (GTSs). In mesh networks, however, the utilisation of such slots would only be possible for a few devices situated in the vicinity of the PAN coordinator.

In dense mesh networks with many devices, the available shared medium limits the maximum number of participants, since from a certain number of devices upwards, no further collision-free messages can be sent. For such networks, a better spatial reuse is required. IEEE 802.15.4 DSME attempts to offer a solution for these problems. It extends the use of GTSs to all devices in the network and opens multiple channels for slot allocation. Of course, this makes coordinating the slots and avoiding inconsistencies in the slot schedules much harder. Because of this, DSME provides its own handshake protocol to allow for distributed slot allocation. Since devices mainly broadcast messages during this procedure, nodes in the neighbourhood can overhear the handshake and will not use the respective slots for their communication.

B. State of the Art

Research on a combination of TDMA and Frequency Division Multiple Access (FDMA) as proposed for DSME has already been performed in [6] and [7]. Researchers previously have pursued this concept of reservation-based communication in the context of IEEE 802.11. That work precedes the current DSME standard. [8] and [9] propose protocols that use a dedicated control channel to allocate the other channels for data transfer. Both protocols have been designed to implement IEEE 802.11 ad hoc networks. [9] requires two transceivers per device to allow parallel listening to the control channel while transmitting or receiving data. The Hop-Reservation Multiple Access (HRMA) protocol introduced in [8] only relies on a single transceiver but requires time synchronisation. Later protocols do not require a separate channel for control information [6], [7]. This change increases the maximum network throughput because every channel can be used for data transmission now. Both [6] and [7] have been based on

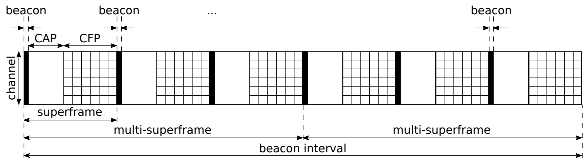

Fig. 1. Structure of an IEEE 802.15.4 DSME Beacon Interval.

IEEE 802.11 and divide time into two different windows. An Ad Hoc Traffic Indication Messages (ATIM) phase is used to negotiate slots in the subsequent communication window. For these protocols, the term slot identifies a specific channel in a portion of the TDMA phase. These slots are thereby equivalent to the GTS used in IEEE 802.15.4 DSME.

Analytically, the performance of IEEE 802.15.4 DSME has been evaluated in [10] where W.C. Jeong and J. Lee show that DSME yields higher throughput compared to CSMA/CA while maintaining a low energy consumption. This low energy consumption is a further advantage of TDMA approaches since the transceivers can be turned off during times where no reception or transmission is planned.

II. DSME Features

In DSME, as well as in standard IEEE 802.15.4, beacons are used for time synchronisation throughout the network. These beacons are broadcasted by coordinator devices in the network. Beacons divide time into Beacon Intervals (BIs) that in DSME consist of multi-superframes which are comprised of superframes. These multi-superframes were newly introduced for DSME. Fig. 1 displays the structure of such an extended BI. Every superframe is composed of a Contention Access Period (CAP) followed by a Contention-Free Period (CFP). Devices can use the CAP for data exchange, but the primary purpose of this phase is the negotiation of management messages. Association and disassociation to a network are done during this phase as well as one of the most complex operations of the protocol - slot reservation.

A. Slot Allocation and Deallocation

Each CFP consists of multiple GTSs, the exact number depends on the number of channels available. During the CAP, every device can request a slot for allocation with any other device in its direct communication range. A device starts this process by sending a *DSME GTS Request* command to the partner with which it wants to share a slot. If the target device issues a positive reply to this request, it is sent via broadcast as a *DSME GTS Response* so that all devices in the neighbourhood can update their data structures. To make this possible for all devices inside of the requester's communication range as well, it sends a *DSME GTS Notify* command to broadcast after receiving the reply.

GTSs in a neighbourhood must not be allocated more than once, to prevent messages from colliding during the CFP. Every device stores a bitmap of all available GTSs with the information if a slot is in use inside the device's communication range, to avert and detect such duplicate allocations. Additionally, all devices have to maintain a list of all GTSs currently active for themselves to make sending or receiving possible at the right point in time and to or from the right partner.

Devices have to deallocate slots which have become orphaned, are not required anymore or have been allocated twice by mistake. The deallocation procedure resembles the allocation message sequence in that a *DSME GTS request* is followed by a broadcasted response and a notify to inform all neighbours about the change.

III. openDSME Implementation

The openDSME implementation is a *C++* data link layer that can currently be executed on hardware as well as in the OMNeT++ simulation environment as introduced in [3]. It aims at providing a portable solution that can easily be ported to different hardware platforms. By using the adaptation layer, it can be combined with any network layer on top. This adaptation layer employs a simple traffic based slot reservation scheme, so it is not even required for the routing layer to make any decisions about the slot schedule, though the interfaces can support any desired interaction.

The DSME layer itself can be implemented on any software platform that offers a scheduling service and basic hardware interfaces to the transceiver. The Media Access Control (MAC) is fully implemented in software, including the execution of channel access backoffs for the CAP and the generation of acknowledgements.

On the upside, this allows for a very flexible implementation. Especially under consideration of faulty hardware MAC layers [11], this is a large bonus and furthermore allows for easy adaptation to other platforms. On the downside, for hardware with little computational power, timing issues become very relevant. DSME poses real time requirements for delays such as the maximum wait time for an Acknowledgement (ACK). Since the IEEE 802.15.4 standard does not explicitly

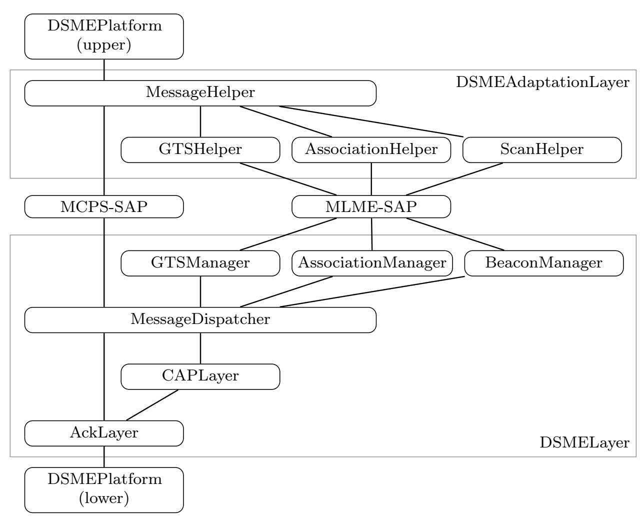

Fig. 2. Structure of the openDSME Link Layer Implementation

regard the processing time, efficient and timely data processing by the software is essential. However, the ACK layer was successfully tested on an Atmel 8-bit ATmega with a clock frequency of $16\,\mathrm{MHz}$.

A. *Structure and Interfaces*

Fig. 2 shows the general structure of the openDSME link layer and how it fits into a network stack. The implementation consists of two major parts. The DSME layer closely implements the DSME protocol as described in [5]. Secondly, the DSME adaption layer implements functionality that is out of the scope of the IEEE 802.15.4 standard but which is required for the integration with upper layers. This includes the previously discussed slot reservation, association to a network and channel scanning. Higher layers have to decide when to request and grant or deny GTSs. Also, the decision-making for network association and address reservation are outside of the scope of the standard.

Since existing network layers as 6LoWPAN [12] or geographic routing layers usually do not care for some or all of these actions, we provide the adaption layer that takes and delivers messages to and from a higher layer and performs management tasks transparently. One topic of our current research is the automatic GTS allocation and deallocation performed by this layer. This improvement allows openDSME to be used in combination with virtually any higher layer. However, it is also possible to bypass this adaption layer and to communicate directly with the DSME layer.

The higher layer sends all instructions to the DSME layer via two interfaces called Service Access Points (SAPs). The MAC Common Part Sublayer (MCPS) SAP is responsible for sending and receiving messages that contain actual payload passed from a higher layer, including the possibility to queue messages for transmission. Management tasks are controlled via the MAC Sublayer Management Entity (MLME) SAP.

It is possible to use DSME on top of every IEEE 802.15.4 Physical Layer (PHY) implementation. It only requires the basic operations of sending and receiving frames, Clear Channel Assessment (CCA) and the possibility to switch channels between those operations.

IV. Current Work

Following the initial implementation, several approaches are currently being pursued. In [13] we perform a formal analysis of the DSME GTS reservation process and propose means to detect inconsistencies faster. The UPPAAL tool environment for verification of systems of timed automata is used to model the allocation and deallocation of GTSs. This model is used to proof that DSME resolves all inconsistencies in a bounded time span.

First simulation results indicate that for large enough transmission intervals DSME can guarantee reliable transmission without any packet loss. On the other hand, DSME increases the end-to-end delays for rare or non-recurring data transmissions since GTS reservation takes up time, while the reliability is already high in those situations. Currently, the approach of sending those messages via CSMA/CA in the CAP is being evaluated to allow for dynamic adaption to the current traffic load.

Furthermore, we registered the problem of bottlenecks emerging in the neighbourhood of gateway devices. A solution to this would be to distribute more gateways throughout the network, but this opposes the purpose of deploying Wireless Sensor Networks (WSNs) for cost saving in the wiring. Since every deployed gateway increases the total cost, the number of these sinks should be kept as small as possible without harming network performance and stability. To reduce the number of required sinks, each of those can be fitted with multiple transceivers to allow parallel communication and better utilisation of the available GTSs. These Multi-Transceiver Gateways are currently being implemented.

To ensure the applicability of our implementation for real-world applications, we want to employ DSME on actual hardware. For this reason, *openDSME* has been consequently designed with hardware compatible interfaces. Running a complete protocol stack with DSME as data link layer in a large-scale testbed is scheduled within the near future.

V. Conclusion

openDSME is the first open implementation of the IEEE 802.15.4 DSME data link layer. It offers a reliable point to point communication service even for large scale networks and can outperform pure CSMA/CA. Our modular design allows DSME to be used on many hardware platforms with existing system software. First results now drive us to test the completed *openDSME* layer on actual hardware while we pursue multiple ideas to improve and verify our existing implementation.

References

[1] J. Colpo and D. Mols, “No strings attached,” *Hydrocarbon Engineering*, vol. 16, Nov. 2011.

[2] F. Meier and V. Turau, “An Analytical Model for Fast and Verifiable Assessment of Large Scale Wireless Mesh Networks,” in *11th International Conference on the Design of Reliable Communication Networks (DRCN)*, Mar. 2015.

[3] A. Varga *et al.*, “The OMNeT++ discrete event simulation system,” in *Proceedings of the European simulation multiconference (ESM2001)*, vol. 9, no. S 185. sn, 2001, p. 65.

[4] S. Unterschütz, A. Weigel and V. Turau, “Cross-platform protocol development based on omnet++,” in *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 278–282.

[5] *Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Standard 802.15.4, 2015.

[6] J. So and N. H. Vaidya, “Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver,” in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2004, pp. 222–233.

[7] Zhang, Jingbin and Zhou, Gang and Huang, Chengdu and Son, Sang H and Stankovic, John A, “TMMAC: An energy efficient multi-channel mac protocol for ad hoc networks,” in *Communications, 2007. ICC’07. IEEE International Conference on*. IEEE, 2007, pp. 3554–3561.

[8] Z. Yang and J. Garcia-Luna-Aceves, “Hop-reservation multiple access (hrma) for ad-hoc networks,” in *INFOCOM’99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 194–201.

[9] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Parallel Architectures, Algorithms and Networks, 2000. I-SPAN 2000. Proceedings. International Symposium on*. IEEE, 2000, pp. 232–237.

[10] W.C. Jeong and J. Lee, "Performance evaluation of IEEE 802.15. 4e DSME MAC protocol for wireless sensor networks," in *Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on*. IEEE, 2012, pp. 7–12.

[11] A. Weigel and V. Turau, "Hardware-Assisted IEEE 802.15.4 Transmissions and Why to Avoid Them," in *Internet and Distributed Computing Systems: 8th International Conference, IDCS 2015, Windsor, UK, September 2-4, 2015. Proceedings*. Cham: Springer International Publishing, 2015, pp. 223–234. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-23237-9_20

[12] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 78–82.

[13] F. Kauer, M. Köstler, T. Lübkert, V. Turau, "Formal Analysis and Verification of the IEEE 802.15.4 DSME Slot Allocation," 2016, unpublished.