

# Sending multiple packets per guaranteed time slot in IEEE 802.15.4 DSME: Analysis and evaluation

Florian Meyer<sup>1</sup> | Ivonne Mantilla-González | Volker Turau<sup>1</sup>

Institute of Telematics, Hamburg  
University of Technology, Hamburg,  
Germany

## Correspondence

Volker Turau and Florian Meyer, Institute of Telematics, Hamburg University of Technology, Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany.  
Emails: turau@tuhh.de;  
fl.meyer@tuhh.de

Coping with bursty traffic is a common yet challenging task in the industrial Internet of Things (IoT). For example, 6LoWPAN is a standard that defines the integration of LoWPAN with IPv6, by fragmenting large IPv6 packets into several smaller MAC-layer packets. Therefore, it is necessary to envision message delivery mechanisms, which provide support for highly varying traffic. In this paper, we analyze sending multiple packets per guaranteed time slot (GTS) in IEEE 802.15.4 DSME to alleviate traffic during the contention-access period (CAP) and increase the reliability in scenarios with bursty traffic. The evaluation shows that increasing parameter  $SO$  extends the network throughput beyond default operating conditions and also provides overprovisioning beneficial for delivering sporadic messages. A comparison with the transmission of a single packet per GTS demonstrates a reduction of the total number of transmitted CAP messages by 99% while increasing the packet reception ratio by 48% for bursts with 20 packets.

## KEYWORDS

guaranteed time slot, IEEE 802.15.4 DSME, multiple packets per GTS, network congestion

## 1 | INTRODUCTION

Universal addressability and connectivity of devices are fundamental pillars of the IoT. For example, the Internet Engineering Task Force proposed the 6LoWPAN<sup>1</sup> standard to achieve addressability and interoperability of heterogeneous MAC-layer protocols with common Internet standards. This standard defines the integration of IPv6 into Low-Power Wireless Personal Area Networks (LoWPAN). Such integration is achieved by fragmenting large IPv6 packets into several smaller MAC-layer packets. This means, a short burst of MAC-layer packets is generated for each sent IPv6 packet. In general traffic patterns with bursts are common in IoT applications, they impose stringent requirements on the MAC-layer protocol in terms of reliability and adaptability to time-varying traffic.<sup>1,2</sup>

In 2015, the IEEE 802.15.4 standard was extended for applications with such strict reliability demands by new MAC-layer protocols: *Deterministic and Synchronous Multi-channel Extension* (DSME) and *Time-Slotted Channel Hopping* (TSCH). Both protocols schedule time and frequency slots, thus eliminating packet collisions due to interferences within the network. We believe that the adaptability of DSME to time-varying traffic and topology changes provides the support needed to realize IoT applications, which experience the above-described dynamic traffic patterns. The management of bursty traffic places a high demand of slot negotiations that needs to be served almost at the same time.

[Correction added on 7 December 2020, after first online publication: Florian Meyer was designated as corresponding author].

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. *Internet Technology Letters* published by John Wiley & Sons, Ltd.

Although the DSME slot (de)allocation process allows reliable data transmissions, vulnerabilities of CSMA/CA affect the slot management. Therefore, it is necessary to envision mechanisms that alleviate CSMA traffic and also provide responsiveness of the network without diminishing its performance.

We conceive two ways to address this congestion problem: Firstly, negotiating multiple GTS in one handshake and secondly, using the entire GTS to send several packets in a single slot. The first option requires far-reaching modifications of the allocation handshake to incorporate information about additional slots to negotiate. Currently, there is no mechanism known that can allocate several slots in bounded time. The second option considerably lowers the load caused by slot allocations since the number of required GTS is significantly reduced. In this paper, we analyze the second option by considering time-varying traffic. This study is made in terms of CSMA traffic, packet reception ratio (PRR), the average number of allocated GTS, latency, and energy consumption. Simulations are performed using OMNET++<sup>3</sup> and openDSME.<sup>4</sup>

## 2 | DESCRIPTION OF DSME

As depicted in Figure 1, DSME divides time into *superframes* (SF), each consisting of 16 time slots. The first of these slots is used to transmit beacons containing network and time information while the remaining slots are split into a *contention-access period* (CAP) with eight and a *contention-free period* (CFP) with seven slots. The CFP is further subdivided into *guaranteed time slots* (GTS) which are spread over time and frequency and grant exclusive access to the shared medium. In the 2.4 GHz band, 16 channels can be used, with 5 MHz channel spacing and 62.5 ksymbol/s each (corresponding to a bit rate of 250 kb/s). During the CAP, nodes exchange control messages via CSMA/CA using a single channel. One or more SF form a multi-superframe (MSF), and a *beacon interval* (BI) contains one or more MSF. The *beacon order* (BO) and the *multisuperframe order* (MO) determine the number of MSF in a BI, which is calculated as  $2^{BO-MO}$ . The structure of a MSF is defined by the *superframe order* (SO) and the *multisuperframe order* (MO). The parameter SO determines the duration of a slot and thus the duration of a superframe,  $D_{SF}$ . The duration of a MSF,  $D_{MSF}$ , is specified by MO. Both durations are multiples of 15.36 ms:  $D_{SF} = 15.36 \times 2^{SO}$  and  $D_{MSF} = D_{SF} \times 2^{MO-SO}$ . Hence, a MSF consists of  $2^{MO-SO}$  superframes. A schedule of allocated GTS is repeated every MSF. To transmit a packet with the maximum MAC Protocol Data Unit of 127 bytes,  $SO \geq 3$  is required.<sup>5</sup> Hence, the choice of SO is relevant for the analysis of sending multiple packets per GTS, as it determines not only the maximum number of packets to be delivered per GTS but also implicitly the length of the CAP in the dataframe structure of DSME.

GTS (de)allocations follow a 3-way handshake, that is done in a distributed manner on a per-hop basis. The handshake, is performed in the CAP and is therefore prone to contention.<sup>6</sup> As exemplified in Figure 2, three messages, a unicast *GTS Request* and two broadcasts *GTS Response* and *GTS Notify*, are exchanged between nodes A and B respectively. The GTS negotiation guarantees a common “collision-free” selection of the tuple (channel, superframe, GTS number). The *collision-free* assumption holds as long as neighbors of A and B detect slot inconsistencies by checking their own GTS schedules. A detection of slot inconsistencies triggers the deallocation of a target GTS.

For this work, an open-source implementation of IEEE 802.15.4 DSME, OpenDSME, is used. The *Scheduling module* is the feature of main interest, because it determines how many and in which combination (channel, superframe, GTS number) slots should be allocated. To this end, openDSME provides the *TPS algorithm*, a traffic-aware scheduling algorithm, which updates the GTS schedule at the beginning of every MSF based on exponential moving average of the rate of incoming packets.<sup>5</sup>

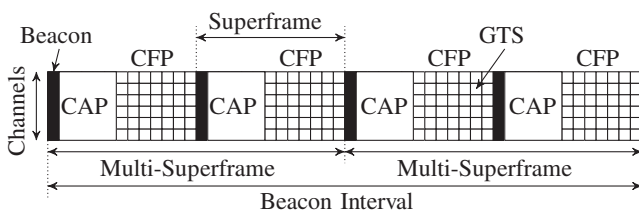


FIGURE 1 Frame structure of IEEE 802.15.4 DSME

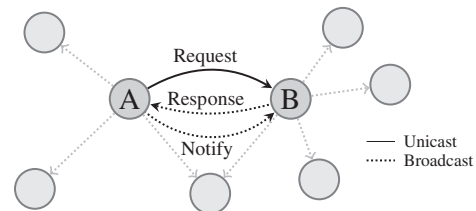


FIGURE 2 Distributed GTS allocation handshake of DSME

### 3 | RELATED WORK

An analysis of the influence of parameters  $SO$  and  $MO$  in DSME was made by F. Kauer.<sup>5</sup> Explicitly, it is stated that a small value of  $SO$ , that is, slots with short length, increases throughput for transmission of single small-size packets. Contrary, setting up a larger value of  $SO$  improves the lifetime of networks with low traffic, since nodes can turn off their transceivers while packet transmission is no longer required. Another alternative to exploit slots with large length is to send multiple packets per GTS. This option is also discussed by Kauer, but not formally analyzed, because it is considered that incrementing the value of  $SO$  to double the number of transmitted packets per GTS, will end up with the same throughput as a single packet transmission mechanism with the original  $SO$  and twice the number of GTS. Therefore, only transmissions of a single packet per GTS are considered in Kauer's work.

Vallati et al performed simulations by using the network simulator Cooja to analyze the GTS allocation process.<sup>7</sup> Their work aims to select CSMA/CA parameters that minimize the number of contention access periods to complete the network setup procedure. The DSME implementation used in their analysis delivers four packets per GTS. Results evidence the advantage of sending multiple packets per GTS in contrast to a single packet per GTS.

A proposal for sending multiple packets per time slot has been implemented for wireless body area sensor networks (WBASNs) based on IEEE 802.15.4.<sup>8</sup> In the work, a frame aggregation mechanism is proposed to provide high performance in patient monitoring systems. To this end, traffic pattern analysis and quality of service (QoS) requirements (eg, high reliability, low energy consumption, restricted delay, etc.) are used to determine resources required by sensor nodes in such systems.

### 4 | THEORETICAL CONSIDERATIONS

Bursty traffic leads to a temporarily increased number of GTS negotiations and thus to increased congestion of the CAP. As a result, a significant number of GTS allocations fail, resulting in larger queues and ultimately packet loss.<sup>9</sup> This can be counteracted by transmitting multiple packets in a single GTS, which reduces the number of required GTS and hence the number of CAP messages for GTS negotiations. For the number of transmissions per GTS, a best-case and a worst-case estimation can be given. According to the IEEE 802.15.4 standard,<sup>6</sup> an acknowledgment (ACK) during a CFP is transmitted at earliest a turnaround time after the last symbol of the data frame is received (best-case). In the worst-case, a receiver should wait for an ACK a maximum waiting time, which amounts to a unit backoff period and a turnaround time. Therefore, the number of transmissions,  $\gamma$ , per GTS is bounded by

$$\frac{S_{GTS}}{2P + 54 \text{ symbol} + \text{IFS}} \leq \gamma \leq \frac{S_{GTS}}{2P + 34 \text{ symbol} + \text{IFS}} \quad (1)$$

where  $S_{GTS} = 60 \cdot 2^{SO}$  is the number of symbols per GTS,  $P$  is the payload of the packet in bytes (2 symbols per byte), and  $\text{IFS} = 40$  symbols when  $P > 18$  Bytes and  $\text{IFS} = 12$  symbols, otherwise. The number of 34 and 54 symbols include the overhead of the physical layer packets, the payload of the ACK and the minimum and maximum ACK wait duration, respectively. TPS estimates the number of required GTS based on the lower bound obtained from this expression.

Table 1 shows that the number of packets per GTS increases exponentially for an increasing  $SO$  and a payload of 127 bytes. That is because by enlarging  $SO$ , the duration  $D_{GTS}$  of a GTS doubles, as given by  $D_{GTS} = 0.96 \text{ ms} \cdot 2^{SO}$ . However, this can lead to more than twice as many transmitted packets. For example, two packets can be transmitted for  $SO = 4$ , while up to five packets can be transmitted for  $SO = 5$ . The reason is that the duration of a frame transmission, including the acknowledgment, does not exactly match the duration of a GTS. The unused time at the end of GTS increases with an increasing  $SO$  until it is large enough to transmit an extra packet. Consequently, the potential throughput of the network increases by transmitting multiple packets per GTS and larger values of  $SO$ . Since the number of packets per GTS increases exponentially, the number of CAP messages is assumed to decrease exponentially as well, especially because fewer collisions occur and fewer packets have to be retransmitted. On the other hand this can result in larger GTS queues and thus a larger packet delay, since packets can not be transmitted immediately upon arrival. Instead, they are transmitted in big chunks of packets in few GTS. Also the probability of having overprovisioning increases for higher values of  $SO$ , because  $\gamma$  is larger and sending  $M$  additional packets could result in the allocation of a new GTS and overprovisioning for at least  $\gamma - M$  additional packet transmissions. This reduces delays and avoids queue drops because packets are transmitted immediately upon a burst and queued packets can be (re)transmitted faster.

	$SO = 3$	$SO = 4$	$SO = 5$	$SO = 6$	$SO = 7$
$\gamma$	1	2	5	10-11	22-23

**TABLE 1** Upper and lower bounds of the numbers of transmittable packets per GTS for different values of  $SO$

Parameter	$SO$	$MO$	$BO$	$\tau$	capacity $Q_{CA}$	capacity $Q_{CF}$
Values	{3, ..., 6}	9	9	16s	40	40

**TABLE 2** Setup of DSME, traffic generator and TPS scheduler parameters used in the evaluation

## 5 | SCENARIO DESCRIPTION

To evaluate the performance of sending multiple packets per GTS in multi-hop networks under bursty traffic, a grid topology with  $4 \times 4$  nodes is considered. Table 2 summarizes the simulation setup. The parameter  $SO$  bounds the maximum number of packets that can be sent in a GTS and  $MO$  determines the period,  $D_{MSF}$ , in which a GTS schedule is updated by TPS (eg,  $D_{MSF} = 7.8464$  seconds for  $MO = 9$ ). We fixed  $BO$  to 9 and thus BI equals 7.8464 seconds. In this evaluation, each node has a MAC queue capacity of 80 packets, which is split in a CAP queue,  $Q_{CA}$ , and a GTS queue,  $Q_{CF}$ , used to store messages to be sent in CAP and CFP respectively. Transmitting nodes generate  $\delta$  packets per burst. The packet generation interval  $\tau$  follows a normal distribution with a mean,  $\mu = 16$  seconds, and a SD  $\sigma = 0.5$  second. This is,  $\tau$  is chosen in a way that it is on average at least twice as long as  $D_{MSF}$  for  $MO = 9$ , to allow for the allocation and deallocation of GTS.

A static shortest path routing is used, in which a bijective pairing is implemented to guarantee all outer nodes of the grid communicate with a unique and fixed chosen partner and the path lengths between communicating nodes is at least 2. Inner nodes act as relays, that is, they receive and forward packets. Therefore, six communication pairs (ie, 6 routes) are active per run: Two routes with path length 3 and 4 with path length 2. A total of 20 runs is conducted and results are shown with a 95% confidence level.

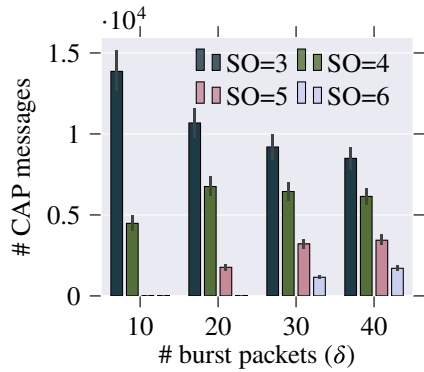
## 6 | EVALUATION

As predicted in Section 4, simulations yield an exponentially increasing number of transmitted packets per GTS for an increasing  $SO$ . However, the actual number of packets transmitted in a GTS can be highly unbalanced, for example, with  $\delta = 16$ ,  $SO = 6$  and two allocated GTS, about 11 packets are transmitted in the first GTS and 5 in the second, resulting in an average of 8 packets per GTS.

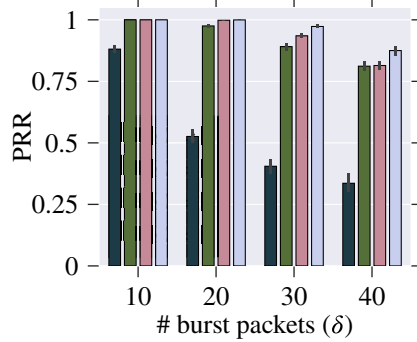
As fewer GTS need to be allocated, fewer CAP messages have to be transmitted for  $SO \geq 4$  when compared to a single packet transmission per GTS ( $SO = 3$ ). This is illustrated in Figure 3. The effect is further enhanced by TPS, which deallocates GTS cautiously. For example, only one GTS is required for  $SO = 6$  and  $\delta = 10$  so that there is almost no CAP traffic. It may seem counter-intuitive at first that the number of CAP messages decreases for  $SO = 3$  with an increasing  $\delta$ , while it increases for higher values of  $SO$ . The reason is that for  $SO = 3$  contention increases with increasing  $\delta$  as more nodes try to (de)allocate GTS. Therefore, nodes go into backoff more often and as long as not all required GTS are (de)allocated according to the incoming traffic rates, queues at the MAC layer (ie,  $Q_{CA}$  and  $Q_{CF}$ ) overflow and the number of GTS handshakes during the CAP decreases, resulting in a lower PRR. On the other hand,  $SO = 6$  performs best for all  $\delta$ . For example, in case of  $\delta = 10$ , the traffic in the CAP is reduced by 99% in comparison to a single packet transmission per GTS ( $SO = 3$ ).

The mentioned queue drops are also reflected in the packet reception ratio (PRR) as shown in Figure 4. For transmissions of a single packet per GTS ( $SO = 3$ ), the PRR decreases exponentially for an increasing  $\delta$  to a value of 33.6% at  $\delta = 40$ . Contrary, higher values of  $SO$  result in a PRR above 80% for  $\delta \leq 40$ , with  $SO = 6$  performing best. That is because fewer GTS are required and it is more likely that all GTS negotiations can be performed and no packets will be dropped due to a full queue. Therefore, it is possible to transmit about 30 packets more per burst while maintaining the same PRR, by choosing  $SO = 3$  instead of  $SO = 6$ . Additionally, it should be noted that lost packets are expected for  $\delta = 40$  for any  $SO$ , since it exactly matches the capacity of the GTS queue. For  $\delta < 8$  no packet loss was observed.

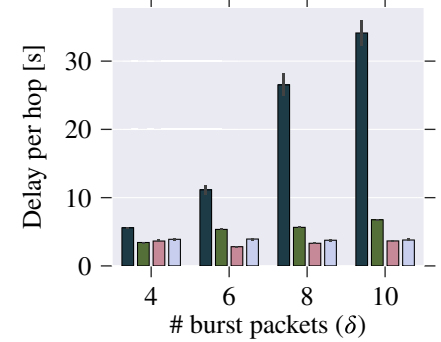
As mentioned in Section 4, the packet transmission delay should increase for an increasing  $SO$  as packets are delayed and sent in a single slot. This is, however, disproved by Figure 5. Here the delay per hop for  $SO = 3$  is significantly higher than the delay for other values of  $SO$ . This is true for two reasons: on the one hand, more GTS have to be allocated for



**FIGURE 3** Total number of transmitted CAP messages for different slot lengths ( $SO$ ) and packets per burst ( $\delta$ )



**FIGURE 4** Average packet reception ratio (PRR) for different slot lengths ( $SO$ ) and packets per burst ( $\delta$ )

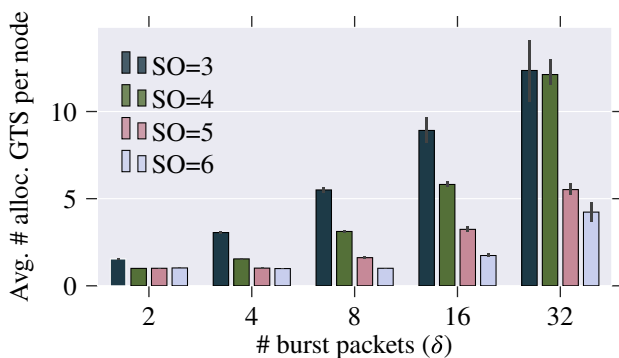


**FIGURE 5** Average delay per hop for different slot lengths ( $SO$ ) and number of packets per burst ( $\delta$ )

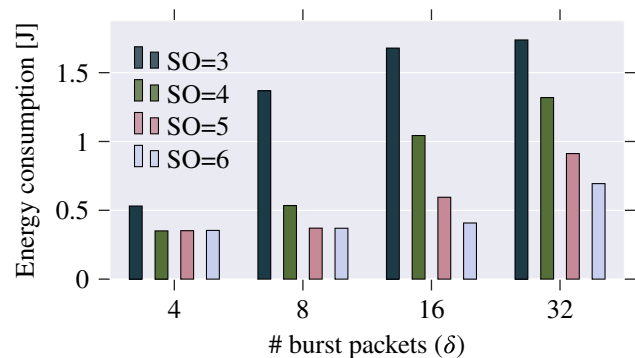
$SO = 3$ , resulting in a longer time until the schedule is completed and packets can be transmitted. On the other hand, choosing  $SO \geq 4$  results in overprovisioning, which is potentially larger for larger values of  $SO$  as indicated in Section 4. For example, for  $SO = 6$  and  $\delta = 15$ , two slots are allocated so that five extra packets can be transmitted. This reduces the length of GTS queue and therefore the packet transmission delay. For  $\delta = 6$ , 2 slots with  $\gamma = 5$  are allocated for  $SO = 5$  and 1 slot with  $\gamma = 10$  is allocated for  $SO = 6$ . Although the overprovisioning is four packets in both cases, the  $D_{MSF}$  is longer for  $SO = 6$ , resulting in a higher expected time until the next allocated GTS and thus in an increased delay.

Figure 6 shows the average number of allocated GTS for transmitting nodes for different values of  $\delta$  and  $SO$ . The larger the  $SO$ , the more packets can be sent per GTS and the fewer GTS must be allocated. On the other hand, a too large  $SO$  results in an overprovisioning. For example, it can be seen that  $SO = 5$  is optimal for  $\delta = 4$ , since  $SO = 4$  requires two more allocated GTS and  $SO = 6$  implies an overprovisioning of 6-7 packets. Moreover, the average number of allocated GTS is higher than expected, for example, for  $SO = 3$  and  $\delta = 8$ , one would expect an average of 4 allocated GTS instead of about 5.7 GTS, since the number of required GTS is either 0 or 8. However, TPS keeps one remaining slot per established link, increasing the average number of allocated GTS. For large bursts for  $SO = 3$  with  $\delta \geq 32$ , nodes reach their own slot (de)allocation rate limits given the increasing amount of slots to negotiate per CAP. For the evaluated burst size, this limit is not reached for configurations with  $SO > 3$ .

At last, it should be noted that sending multiple packets per GTS is not beneficial for all types of applications. CAPs are longer for larger values of  $SO$ , potentially resulting in a considerable fraction of it not being utilized. This idle time can be counterproductive for energy-harvesting devices, since nodes have to stay active during the whole CAP. However, the remaining time in the CAP gives room for sending other types of control messages (eg, link quality indicator) relevant for the management and maintenance of the network. Figure 7 shows the average total energy consumption per node. The energy consumption for a single packet per GTS ( $SO = 3$ ) is highest because there are more GTS negotiations in the CAP during which the transceiver is active. Additionally, it can be seen that the energy consumption stays constant



**FIGURE 6** Average number of allocated GTS per node for different slot lengths ( $SO$ ) and packets per burst ( $\delta$ )



**FIGURE 7** Average total energy consumption per node for different slot lengths ( $SO$ ) and packets per burst ( $\delta$ )

for  $SO = 6$  until  $\delta = 16$ . Here, the nodes need to allocate 2 GTS instead of 1, resulting in a higher energy consumption. For  $\delta = 1$  higher values of  $SO$  result in a higher energy consumption because the receiver has to stay active during the whole GTS. Finally, the evaluation assumes that all nodes generate similar amounts of traffic. If the generation rates of individual nodes vary considerably, a lot of time might be wasted during GTS for large  $SO$ . In that sense, it is necessary to carefully define the value of  $SO$  that optimally fits the traffic demand of the network. Energy consumption during idle time of a GTS can be reduced by the transmitter announcing the end of the communication to the receiver. Then, both communication partners can turn off their transceivers for the remaining part of the current GTS.

## 7 | CONCLUSION

In this paper, we consider the transmission of multiple packets per GTS to reduce the number of slot allocations and to increase the reactivity of DSME networks under bursty traffic patterns, for example, 6LoWPAN applications. A theoretical estimation is given and the results are verified by simulation. The simulations show that changing  $SO = 3$  to  $SO = 6$  results in a higher PRR in a grid topology with 16 nodes for different burst sizes. It also increases the GTS length and the number of transmittable packets per GTS from 1 to 11 and reduces the number of GTS (de)allocation messages by up to 99%. At the same time, it enlarges the length of the CAPs significantly, leaving CAPs at the end of a MSF unused most of the time. Therefore, finding an algorithm to tune the number of CAPs according to the traffic demand of the network is an essential next step, especially because DSME can currently either use all CAPs or a single CAP per MSF (CAP-reduction).

## ACKNOWLEDGMENT

This research is partially supported by the German Academic Exchange Service (DAAD). Open access funding enabled and organized by Projekt DEAL.

## ORCID

Florian Meyer  <https://orcid.org/0000-0002-0901-9408>

Volker Turau  <https://orcid.org/0000-0001-9964-8816>

## REFERENCES

1. Shelby Z, Bormann C. *6LoWPAN: The Wireless Embedded Internet*. Vol 43. Chichester: John Wiley & Sons; 2011.
2. Hui J, Culler D, Chakrabarti S. 6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture. *White paper*. 2009.
3. Varga A, Hornig R. An overview of the OMNeT++ simulation environment. In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems workshops*. Marseilles, France: ICST (Institute for Computer Science, Social-Informatics and Telecommunications Engineering); 2008:60-69.
4. Kauer F, Köstler M, Turau V. openDSME: reliable time-slotted multi-hop communication for IEEE 802.15.4. Virdis A, Kirsche M, *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*. Berlin: Springer; 2019:451-467.
5. Kauer F. Scalable Wireless Multi-Hop Networks for Industrial Applications [PhD thesis]. Hamburg University of Technology, Hamburg; 2019.
6. IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*. 2016.
7. Vallati C, Brienza S, Palmieri M, Anastasi G. Improving network formation in IEEE 802.15.4e DSME. *Comput Commun*. 2017;114:1-9.
8. Akbar M, Yu H, Cang S. IEEE 802.15.4 frame aggregation enhancement to provide high performance in life-critical patient monitoring systems. *Sensors*. 2017;17(2):241.
9. Meyer F, Mantilla-González I, Kauer F, Turau V. Performance Analysis of the Slot Allocation Handshake in IEEE 802.15.4 DSME. Palatella M, Scanzio S, Coleri S, *Ad-Hoc, Mobile, and Wireless Networks*. Cham: Springer; 2019:102-117.

**How to cite this article:** Meyer F, Mantilla-González I, Turau V. Sending multiple packets per guaranteed time slot in IEEE 802.15.4 DSME: Analysis and evaluation. *Internet Technology Letters*. 2021;4:e167. <https://doi.org/10.1002/itl2.167>