DSME-LoRa: Seamless Long-range Communication between Arbitrary Nodes in the Constrained IoT

JOSÉ ÁLAMOS, PETER KIETZMANN, and THOMAS C. SCHMIDT,
HAW Hamburg, Germany
MATTHIAS WÄHLISCH, Freie Universität Berlin, Germany

Long-range radio communication is preferred in many IoT deployments, as it avoids the complexity of multi-hop wireless networks. LoRa is a popular, energy-efficient wireless modulation but its networking substrate LoRaWAN introduces severe limitations to its users. In this article, we present and thoroughly analyze DSME-LoRa, a system design of LoRa with IEEE 802.15.4 Deterministic Synchronous Multichannel Extension (DSME) as a MAC layer. DSME-LoRa offers the advantage of seamless client-to-client communication beyond the pure gateway-centric transmission of LoRaWAN. We evaluate its feasibility via a full-stack implementation on the popular RIOT operating system, assess its steady-state packet flows in an analytical stochastic Markov model, and quantify its scalability in massive communication scenarios using large-scale network simulations. Our findings indicate that DSME-LoRa is indeed a powerful approach that opens LoRa to standard network layers and outperforms LoRaWAN in many dimensions.

CCS Concepts: • **Computer systems organization → Sensor networks**; • **Networks → Link-layer protocols;**; *Network performance analysis*

Additional Key Words and Phrases: Internet of Things, wireless, LPWAN, MAC layer, network experimentation

ACM Reference format:
José Álamos, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2022. DSME-LoRa: Seamless Long-range Communication between Arbitrary Nodes in the Constrained IoT. *ACM Trans. Sensor Netw.* 18, 4, Article 69 (November 2022), 43 pages.
https://doi.org/10.1145/3552432

1 INTRODUCTION

LoRa is a popular wireless modulation for the IoT that is robust against interference and Doppler effect. It uses narrowband Chirp Spread Spectrum modulation to achieve long-range transmission (km) with low power consumption (mW). LoRa operates in unlicensed spectra and therefore is subject to regional regulations that shall prevent a saturation of the spectrum. In Europe, the ETSI

This work was supported in part by the German Federal Ministry for Education and Research (BMBF) within the project *PIVOT: Privacy-Integrated design and Validation in the constrained IoT*.

Authors' addresses: J. Álamos, P. Kietzmann, and T. C. Schmidt, Department Informatik, HAW Hamburg, Berliner Tor 7, Hamburg, 20099, Germany; emails: {jose.alamos, peter.kietzmann, t.schmidt}@haw-hamburg.de; M. Wählisch, Institut für Informatik, Freie Universität Berlin, Takustr. 9, Berlin, 14195, Germany; email: m.waehlisch@fu-berlin.de.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1550-4859/2022/11-ART69 $15.00
https://doi.org/10.1145/3552432

EN300.220 standard [17] limits the transmission duty cycle to 0.1%, 1%, or 10%, depending on the sub-band.

LoRaWAN was designed as an upper layer for LoRa that provides Media Access Control and Internet communication between LoRa end devices and end-user applications. LoRaWAN is a cloud-based Media Access Control (MAC) layer for LoRa that organizes Physical Layer (PHY) configurations and MAC schedules and routes traffic between end devices and end-user applications. The LoRaWAN architecture consists of three components: an Application Server, which contains the application logic; a Network Server, which coordinates access to the media between nodes and routes traffic between the Application Server and End Devices; and Gateways, which act as the backbone of the LoRaWAN network. The architecture prevents peer-to-peer communication between end devices, which operate without network layer.

LoRaWAN defines three operational classes (modes) that show a tradeoff between downlink delay and power consumption. With class A, the reception of downlink packets is only possible during a short interval after an uplink transmission. Consequently, class A devices exhibit a high downlink latency, but the highest power efficiency. With class C, the end devices are always listening. The downlink latency is consequently lowest at the cost of a high energy consumption. In class B, beacon-synchronized end devices wake up periodically to be able to receive data. This class provides a good tradeoff between downlink latency and power consumption.

LoRaWAN imposes a series of limitations, which make it impractical for scenarios with heterogeneous communication patterns. To increase the LoRa versatility as well as its efficiency, we propose the usage of IEEE 802.15.4 DSME as a MAC layer for LoRa. DSME is a flexible MAC layer introduced in the 802.15.4e revision (2012) that provides communication in contention-access, as well as contention-free (time/frequency slots).

In this article, we want to answer the research question of how LoRa can be integrated with such a flexible MAC layer and how this stack performs in various settings. In particular, we want to show how LoRa end nodes can be opened up for hosting various network layers such as standard IP or data-centric adaptations [32]. The contributions of this article are as follows:

(1) We present DSME-LoRa, a system design of LoRa with IEEE 802.15.4 **Deterministic Synchronous Multichannel Extension (DSME)** as a MAC layer.
(2) We evaluate in this work the performance of DSME-LoRa on real hardware, based on a DSME-LoRa implementation [5] on the popular IoT operating System RIOT [7].
(3) We propose a novel analytical stochastic model to predict transmission delay and throughput for DSME slotted transmission.
(4) We perform a large-scale simulation of DSME-LoRa nodes to assess the scaling behavior of our proposed solution.
(5) Based on the evaluation and model results, we derive preferred mappings for implementing different transmission patterns, with a balance tradeoff of energy consumption and transmission delay.

The remainder of this article is structured as follows: We outline the shortcomings of the current LoRaWAN system along with a problem statement in Section 2. The relevant background on low-power radio communication is summarized in Section 3. Section 4 presents our DSME-LoRa system design, which we evaluate on real hardware in Section 5. We develop an analytical stochastic model in Section 6, from which we predict the per-packet performance for the slotted transmission. Peer-wise communication in large ensembles of LoRa nodes is subject to a simulation study in Section 7. In Section 8, we discuss design decisions and options for optimization. Finally, we review related work in Section 9 and give a conclusion and outlook in Section 10. The

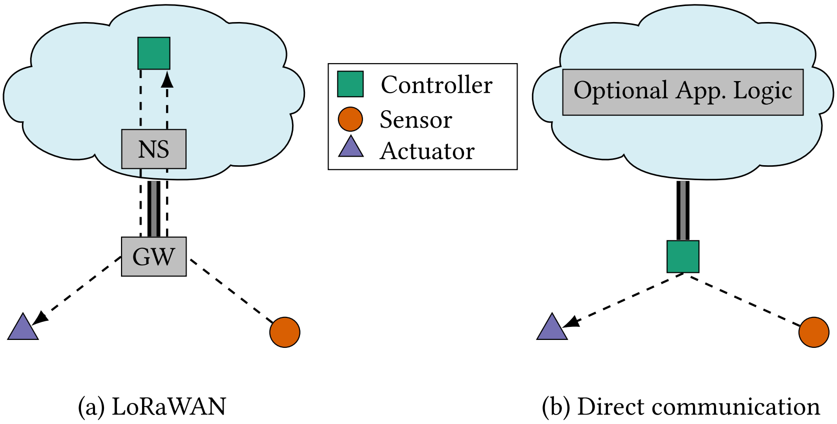


Fig. 1. Control traffic of smart lighting scenario in LoRaWAN (left) through a gateway (GW) and direct communication (right).

Appendix provides a supplementary figure (Section A) and lists a table of abbreviations (Section B) that we use throughout this article.

2 PROBLEM STATEMENT

The common LoRaWAN architecture adds rigid constraints to long-range networking, which hinder many IoT deployments. Its centralized design facilitates uplink-oriented applications but challenges data sharing and the creation of distributed applications.

We argue that direct communication between LoRa devices overcomes these limitations, while it still enables reliable communication in long-range deployments or harsh environments. To further motivate the need for direct communication, we analyze an IoT control scenario for smart lighting.

Sanchez-Sutil et al. [20] design a LoRa system for smart regulation of street lights (see Figure 1(b)) and propose an architecture with illumination-level devices (sensors), which transmit sensor data every minute. A gateway for street lights system (controller), which acquire illumination data from sensors, transmit control messages to actuators and send measurement data from street lights to the cloud. Operating and monitoring devices for street lights (actuators) control light level and transmit electrical measurements to the controllers. The authors deploy several scenarios (up to 64 actuators) with all devices in LoRa wireless reach.

A LoRaWAN implementation of such system may move controller logic to the cloud application and use LoRaWAN to transmit data between sensors and actuators (see Figure 1(a)). However, this approach has the following disadvantages:

(*i*) Traffic between controller and actuator is forced through LoRaWAN gateways. If controllers transmit unicast control data every minute to all actuators, then nearby gateways will forward 64 downlink packets per minute. Even if the LoRa devices and the LoRaWAN Network Server agree on the fastest downlink data rate, a single gateway scenario will render 7% duty cycle. Because LoRaWAN gateways are half-duplex, packets received during downlink transmission are lost. Therefore, such a deployment requires at least two dedicated gateways to enable a Data Extraction Rate (DER) ≥99%. In regions with duty cycle regulations such as EU868, even more gateways are required to prevent additional packet losses as a result of downlink budget depletion. Adding more gateways addresses these problems, but it increases deployment costs and it is not always practical.

(*ii*) The LoRaWAN infrastructure prevents the deployment of edge devices and blindly forwards sensor data to the cloud infrastructure. To further motivate the usage of edge devices, consider a deployment in the city of London (≈2.8 million street lights). If sensors are on par with actuators and transmit every minute, then the cloud infrastructure receives 1.5 trillion LoRaWAN messages per year, which artificially leads to a cost explosion in cloud infrastructure.

(*iii*) Devices with poor LoRaWAN wireless coverage increase transmission time on air to improve link budget. This increases energy consumption [36], which reduces life cycle of nodes.

(*iv*) In many remote areas, cellular networks are the only uplink options for LoRaWAN gateways. Poor Internet connectivity will lead to packet loss at the gateway, which threatens the versatility of the control system.

The proposed smart lighting topology (see Figure 1(b)) overcomes the limitations of the LoRaWAN architecture. Instead of using a centralized controller in the cloud, the system implements low-cost controllers that run the control logic in a distributed way. Therefore, downlink traffic is distributed between many controller devices instead of aggregating at a few gateways. This effectively reduces downlink stress. Because sensors and controllers are likely in wireless reach, sensors can transmit using a fast data rate, which facilitates battery-powered operation. Controllers can transmit preprocessed data at a lower rate, which effectively reduces cloud transmissions and infrastructure costs. The system does not interrupt control operation in case of intermittent connectivity at controllers. In addition, controllers are free to implement caching strategies to reduce packet loss on intermittent Internet uplinks.

In light of this use case, we argue that a DSME MAC should perform better than the proposed system for two reasons: (*i*) DSME enables multichannel time-slotted communication, in contrast with the single channel approach of the system. This enables concurrent collision-free communication without special hardware requirements (e.g., LoRa concentrator). Therefore, controllers may be implemented with low cost components, while still maintaining high Packet Reception Ratio (PRR). (*ii*) DSME offers powerful built-in features such as device discovery and security mechanisms, which facilitate deployment and secure operation.

We follow the DSME-LoRa direction in the remainder of this article to foster flexible long-range node-to-node communication.

3 BACKGROUND ON LOW-POWER RADIOS

3.1 IEEE 802.15.4 with DSME MAC

The **Deterministic and Synchronous Multichannel Extension (DSME)** initiates a beacon-synchronized superframe structure that consists of a beacon slot, a Contention Access Period (CAP) and a Contention Free Period (CFP). End devices can choose to communicate during CAP or CFP. During CAP, devices transmit using Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) in a common channel. During CFP, end devices transmit in a dedicated time-frequency slot. The CFP is divided in the time domain into seven multichannel slots, namely, Guaranteed Time Slot (GTS). Each GTS is divided in the frequency domain into the number of available channels in the channel page (usually 16). DSME supports both peer-to-peer and cluster tree topologies. Similar to traditional IEEE 802.15.4, there are three device roles: Personal Area Network (PAN) coordinator, regular coordinators, and child devices. Devices can transmit confirmed messages, where the MAC layer retransmits frames in case of missed ACK frame. We summarize the configuration parameters in Table 1 and introduce in the reminder of this section.

Network formation. The PAN coordinator is the device in charge of defining the superframe structure. For this purpose, the device will transmit *enhanced beacons* periodically. The transmission of *enhanced beacons* always occurs during a beacon slot, and the period is a multiple of the superframe duration. Devices that want to join the DSME network perform a scanning procedure to detect *enhanced beacons*. When the scanning procedure succeeds, the joining device sends an

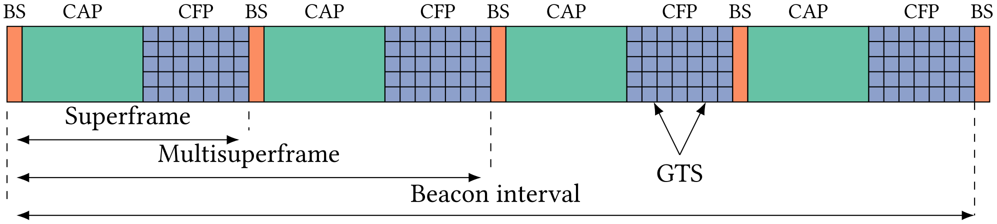


Fig. 2. Overview of the DSME multisuperframe transmission resources.

Table 1. List of MAC Configuration Parameters for the DSME MAC

Parameter	**Description**
`macMinBE`	Minimum backoff exponent
`macMaxBE`	Maximum backoff exponent
`macMaxCsmaBackoff`	No. of access attempts before declaring channel failure
`macMaxFrameRetries`	No. of frame retransmissions.
`macSuperframeOrder (SO)`	Describes the length of the superframe slot
`macMultisuperframeOrder (MO)`	Describes the length of the multisuperframe
`macBeaconOrder (BO)`	Describes the length of the beacon interval
`macCapReduction`	Opt. replace CAP with CFP in all superframes except first
`macRxOnWhenIdle`	Opt. keep receiver on during CAP

association request to the coordinator. The association finishes when the coordinator acknowledges with a positive association reply.

The DSME network can be extended natively by adding more coordinators. In such case, the coordinator will emit *enhanced beacons* using the same period as the PAN coordinator but in a different beacon slot offset. This ensures multiple coordinators can share the same area without risk of beacon collisions. In the event of beacon collisions (i.e., two coordinators started emitting beacons in the same slot offset), DSME provides a native mechanism to resolve collisions. To be able to switch the common channel and PHY properties, the standard defines the PHY-OP-SWITCH mechanism in which neighbor devices are instructed to switch to a different PHY configuration on reception of a dedicated MAC command. This allows dynamic switching between data rates, modulations, and frequency bands.

Superframe structure. Superframes merge into a multisuperframe structure as visualized in Figure 2. DSME supports a CAP reduction mode in the multisuperframe structure (`macCapReduction`), in which the CAP period is replaced by 8 CFP additional slots in all superframes except the first. For example, a configuration with four superframes per multisuperframe exposes 28 GTS (448 unique time-frequency slots). With CAP reduction, the same structure exposes 52 GTS (832 unique time-frequency slots).

DSME defines three parameters to describe the superframe structure, namely, Superframe Order (SO), Multisuperframe Order (MO), and Beacon Order (BO) (compare Table 1). The superframe order defines the slot duration as: `aBaseSuperframeDuration` $\cdot T_{Symbol} \cdot 2^{SO}$, where `aBaseSuperframeDuration` = 60 symbols, as per standard. A small superframe order, which leads to a shorter superframe duration, offers shorter latencies at the cost of higher energy consumption and smaller payload. SO = 3 enables the transmission of standard 127 bytes 802.15.4 frames. The multisuperframe order, together with the superframe order, define the number of superframes per multisuperframe as $2^{(MO-SO)}$. Higher multisuperframe orders lead to higher GTS resources with the cost of higher latencies. Finally, the beacon order sets the beacon interval to $2^{(BO-MO)}$

Table 2. Number of Available GTS with and without CAP Reduction; and Multisuperframe Duration (T_{msf}) for Varying Multisuperframe Orders (MO), with a Superframe Order (SO) of 3 and Symbol Time 1 ms

Multisuperframe order (MO)	Number of GTS (w/o CAP reduct.) [#]	Number of GTS (w/CAP reduct.) [#]	Duration $T_{msf}[s]$
3	7	7	7.68
4	14	22	15.36
5	28	52	30.72
6	56	112	61.44
7	112	232	122.88

multisuperframes. Higher beacon orders lead to higher beacon intervals, which extend the number of potential coordinator devices at the cost of longer association time. These three parameters must comply with $0 \leq SO \leq MO \leq BO \leq 14$. We summarize the number of available GTS and the multisuperframe duration for different multisuperframe orders for the case SO=3 in Table 2.

CSMA/CA transmissions. On schedule of a CSMA/CA transmission, the MAC queues the packet in the CAP queue and performs slotted CSMA/CA, aiming to avoid collisions while accessing the common channel. The CSMA/CA algorithm requires four parameters displayed in the first four rows in Table 1. On transmission the MAC aligns to the backoff period, which occurs every 20 symbols since the start of the CAP, and waits a random number of backoff periods between 0 and $2^{\texttt{macMinBE}}$. In case the duration of the remaining portion of the CAP is shorter than the required backoff periods, the MAC waits for the next CAP period and continues its countdown accordingly. The MAC then performs a series of clear channel assessments (at least two), each one at the beginning of a backoff period. On failure, the MAC doubles the backoff period (below $2^{\texttt{macMaxBE}}$) and the CSMA/CA algorithm retries until it succeeds or the MAC runs out of CSMA/CA attempts (`macMaxCsmaBackoff`). When successfully accessing the channel, the MAC transmits the frame and (optionally) waits for an ACK frame. If the MAC expects an ACK frame and does not receive it, then the MAC repeats the CSMA/CA procedure until it runs out of CSMA/CA attempts or retransmissions (`macMaxFrameRetries`).

During CAP the MAC can transmit both unicast and broadcast frames. To minimize the energy consumption on constrained devices, the MAC offers the `macRxOnWhenIdle` configuration parameter to turn off the receiver during CAP. This does not affect outgoing transmissions, but prevents the MAC from receiving frames. To transmit frames to these constrained devices, the standard defines the *indirect transmission* mechanism. A coordinator queues frames scheduled with *indirect transmission* and appends the target address to the next beacon. A constrained device that finds its address in the beacon polls the coordinator with a data request command and waits for an ACK frame with the subsequent data frame.

GTS transmission. End devices that require communication with other devices during CAP need to negotiate one or more GTS with the target device. DSME provides a native mechanism to negotiate slots—in contrast to Time Slotted Channel Hopping (TSCH). DSME GTS are unidirectional (RX or TX) and only support unicast frames. When a device **A** wants to allocate one or several slots with device **B** (coordinator or child), it sends a DSME-GTS request frame during CAP to device **B**. In case the device accepts the slot, it replies with a DSME-GTS response frame indicating success. Finally, device **A** broadcasts a DSME-GTS notify frame to indicate the other node in reach about the new slot allocation. Alternatively, a device can allocate a slot during the association procedure by sending a DSME Association Request command. On schedule of GTS transmission, the MAC queues the packet in the CFP queue, which divides into multiple FIFO queues, one for each destination device among the allocated GTS resources. GTS transmissions support two channel diversity modes, namely, channel adaptation and channel hopping. In channel adaptation mode, a source device may allocate GTS in a single channel or in different channels based on the knowledge of the channel quality. The source device requests channel quality information to a destination device using the DSME Link Report MAC command. Thereby devices agree on a different channel if the channel quality is poor. In channel hopping mode, each GTS hops over a predefined sequence of channels.

DSME supports message priority for GTS transmissions. On the occurrence of a valid GTS, the MAC layer transmits first the frames with high priority and then regular frames, providing a class-based service differentiation.

The 802.15.4e amendment introduces the group ACK feature, in which a coordinator receiving data from multiple senders transmits one group ACK frame to all nodes in a single slot of a multisuperframe. The latest versions of the standard do not include this feature, but we discuss its potential use cases for reducing time on air in Section 8.3.

3.2 LoRa Modulation

The LoRa modulation utilizes the chirp spread spectrum technique to transmit data over the wireless channel. This technique defines a linear frequency modulated symbol, namely, chirp, which utilizes the entire allocated bandwidth spectrum. As a result, the LoRa signal is robust against interference and multi-path fading and enables transmission ranges of kilometers, depending on the PHY configuration. An interesting property of the LoRa modulation, namely, the capture effect, allows to successfully decode a frame under collision if the power difference with the colliding frames is large enough.

LoRa relies on two PHY parameters, namely, bandwidth and spreading factor, which define the symbol duration. A higher symbol duration renders better receiver sensitivity, which increases transmission range at the cost of higher time on air and lower PHY bit rate. A third parameter, code rate, defines the redundancy bits encoded in the LoRa transmission. Similarly, the code rate trades off transmission range with time on air.

The LoRa PHY frame consists of a preamble, used to synchronize the transceiver to the frame; an optional LoRa PHY header, which encodes payload length, forward error correction code rate, and the presence of a payload CRC at the end of the PHY packet; a payload, which contains the PSDU; and an optional payload Cyclic Redundancy Check (CRC). The LoRa preamble defines a sync word at the end, with the purpose of isolating networks of LoRa devices. For example, LoRaWAN sets the sync word to 0×34 for public networks and 0×12 for private networks.

LoRa devices are subject to regional Sub-GHz regulations that impose restrictions on the transmission of LoRa frames. These restrictions can be categorized in (*i*) duty cycle restrictions, in which a transmitter may not exceed a maximum time over an observation period (usually 1% of time over an hour); (*ii*) dwell time restrictions, in which the transceiver may not exceed a maximum time on a single channel; and (*iii*) channel restriction, in which the device must switch channels on consecutive transmissions or transmit over a minimum number of channels.

LoRa transceivers can decode signals below the noise floor, which renders energy detection mechanisms such as RSSI impractical for detecting the presence of signals on the air. To circumvent this problem, common LoRa transceivers implement a Channel Activity Detection (CAD) mechanism to note the presence of a LoRa preamble signal.

An interesting feature of LoRa transceivers, which has not been exploited by LoRaWAN, is Frequency-hopping spread spectrum (FHSS) transmission. This feature allows to repeatedly switch carrier frequencies during radio transmission, aiming to reduce interference and avoid interception.

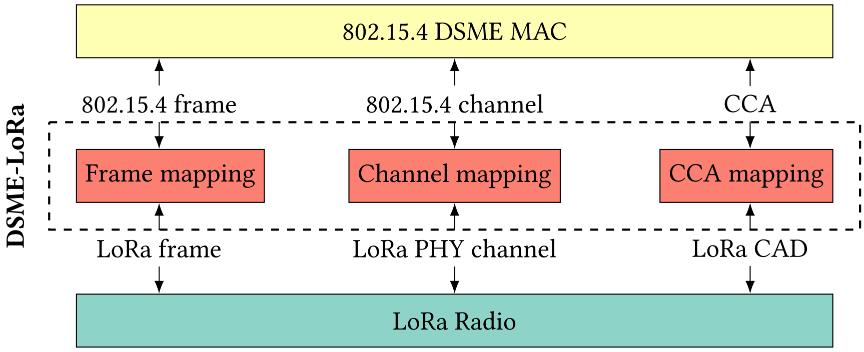


Fig. 3. DSME-LoRa architecture overview.

4 DSME-LORA SYSTEM DESIGN

To operate LoRa radios below DSME, we define a DSME-LoRa adaptation , based on our original work [4], that maps 802.15.4 MAC operations to LoRa operations. The adaptation layer performs three tasks: (*i*) mapping of 802.15.4 channels to LoRa PHY channels; (*ii*) conversion of 802.15.4 frames to LoRa frames; (*iii*) implementation of 802.15.4 Clear Channel Assessment (CCA) on top of the LoRa device. Figure 3 depicts the system architecture.

Channel mapping. The adaptation layer maps 802.15.4 channels to LoRa PHY channels. For this work, we define a channel page with 16 LoRa channels (see Table 3) in the EU868 [17] region. Note that the channel page may define more than 16 channels, as long as the channel information fits in the MAC frames that control GTS allocation.

In the EU868 region the duty cycle of a band limits the time on air of a transmitting device to a percentage within one-hour observation period. For 1% and 10% bands, the duty cycle limits the cumulative time on air to 36 s and 360 s, respectively. For example, devices in a 1% band cannot transmit a frame if the active send time exceeds 36 s during the last hour. Note that the duty cycle is measured per band and not per channel. If a device transmits in multiple channels on a same band, then the duty cycle of each channel adds up to the duty cycle of the band.

All channels utilize the same PHY configurations: spreading factor 7, bandwidth 125 kHz, and code rate 4/5, which results in a PHY bit rate ≈5.5 kbps and a symbol time of ≈1 ms. We choose these settings to provide a balanced tradeoff between transmission range, time on air, and throughput.

Note that different sets of LoRa PHY settings can be encoded using different channel pages. Thereby, devices can agree on different channel pages, using the PHY-OP-SWITCH feature Section 3.1, to increase transmission range or increase the channels for concurrent PHY communication. We will investigate the feasibility of this proposal in future work.

Defining LoRa PHY channels for other regions is viable, albeit challenging. We further discuss this situation in Section 8.3.

We define one channel inside the *g3* band (10% duty cycle) and 15 channels inside the *g* band (1% duty cycle) with 200 kHz channel spacing. To relax duty cycle restrictions, we utilize the 10% band channel for beacon transmissions, CAP channel, and GTS transmissions. The remaining channels are used exclusively for GTS transmissions.

Since the proposed channels overlap with LoRaWAN channels, we define the synchronization word of the preamble to 0×17 to avoid decoding of LoRaWAN frames. Furthermore, we include the LoRa PHY header and payload CRC described in Section 3.2.

Frame mapping. On frame transmission, the adaptation layer calculates and appends a checksum to the MAC frame and passes the frame to the LoRa transceiver. On frame reception, the layer

Table 3. DSME-LoRa PHY Channel Definition, EU868 Frequency Band Information, and Purpose

Channel	Band	Frequency [MHz]	Duty Cycle [%]	Purpose
11–25	*g*	863.00–868.00	1	GTS transmission
26	*g3*	869.40–869.65	10	Beacon transmission, CSMA/CA transmission, GTS transmission

receives the LoRa frame from the transceiver and calculates the frame checksum. On success, the layer dispatches the frame to the MAC layer. To transmit full 127 bytes 802.15.4 frames, we set the superframe order to 3. The adaptation layer defines the MAC symbol time to 1 ms, which is in line with the LoRa symbol time for the channel configuration. With this superframe order configuration and symbol time, the superframe slot duration resolves to 0.48 s. Hence, the superframe duration (16 superframe slots) is 7.68 s. We leave the multisuperframe and beacon order configuration to the application.

CCA mapping. On CCA requests from the MAC layer, the adaptation layer maps to the LoRa CAD feature, which detects the presence of a LoRa preamble on the air. On successful detection, the layer reports channel busy to the MAC layer. Otherwise, the layer assumes the channel is free and reports clear channel.

4.1 DSME-LoRa Implementation

The integration of DSME-LoRa on real hardware imposes a series of challenges: (*i*) long time on air of LoRa requires a long superframe slot duration, which results in long beacon intervals. IoT devices are prone to clock drift from cheap crystals, which increases the chances of desynchronization between child devices and coordinators; (*ii*) common LoRa transceivers do not add a mechanism to timestamp frame reception, which is required to synchronize time between neighbor; (*iii*) DSME accesses the transceiver based on interrupts during critical operations. This faces concurrent access with hardware serial peripheral buses (e.g., SPI) and limits the responsiveness on real-time operating systems; (*iv*) common LoRa hardware platforms are constrained and have low memory resources. This serves a common LoRaWAN stack. In contrast, DSME requires additional memory due to the complexity of the MAC.

We integrate *openDSME* into the RIOT network stack (GNRC), which provides a generic messaging interface (GNRC Netapi), a centralized packet buffer (GNRC Pktbuf), and a packet dispatch registry (GNRC Netreg). RIOT provides a high-level platform timer API and a hardware abstraction layer for 802.15.4 devices. We further extend *openDSME* to support the *macRxOnWhenIdle* mode (Table 1) to turn the transceiver off during CAP and save energy when CAP is not used. Figure 4 presents the system integration of DSME-LoRa in RIOT and our contributions.

GNRC Netif DSME. Implements the GNRC network interface for DSME. This allows, via the GNRC Netapi: (*i*) transmission of DSME frames; (*ii*) configuration of the DSME MAC (device role, static slot allocation); (*iii*) scanning and association procedures. We use the DSME Adaptation Layer, a convenience API provided by *openDSME*, to implement the MAC layer logic below the network interface. The interface dispatches the incoming frames via the GNRC Netreg. To minimize memory consumption, we utilize the callback extension of GNRC Netreg as an alternative to the default IPC implementation. This saves an additional thread for reception, which heavily reduces RAM utilization.

DSME Message. Implements the **DSME Message interface (IDSMEMessage)**, which abstracts the packet representation. We use the GNRC Pktbuf to implement the packet representation. This

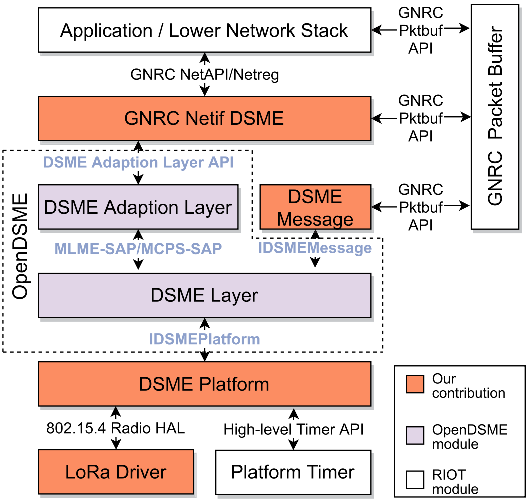


Fig. 4. DSME-LoRa integration into the networking subsystem of RIOT.

approach has advantages on memory consumption: (*i*) centralized storage prevents data duplication; (*ii*) the scattered packet representation of GNRC Pktbuf allows appending chunks of data to a packet without memory reallocation; (*iii*) GNRC Pktbuf supports allocation with *malloc*. This facilitates the operation of DSME in the same memory pool as *openDSME*, which bases on heap.

DSME Platform. Implements the **DSME Platform interface (IDSMEPlatform)**, which defines the platform abstraction layer of *openDSME*. The interface implements the access to the LoRa transceiver on top of the 802.15.4 Radio HAL. It further implements the access to timer functionalities of the operating system. Thereby, we configure the high-level timer to use the real-time timer peripheral, aiming to mitigate the effect of clock drift due to long beacon intervals. We delegate the processing of transceiver interrupts and system timers to the RIOT scheduler to avoid concurrent access to the system bus between the transceiver and operating system. The implementation reconfigures the symbol time of the MAC layer to 1 ms (LoRa) in compliance with Section 4.

LoRa Driver. Implements a 802.15.4 compatible driver for the LoRa transceiver (SX1272/SX1276). The driver implements the three components of the DSME-LoRa Adaptation Layer (Section 4), namely, channel mapping, frame mapping, and CCA mapping. To timestamp frame reception, we calculate the time difference between the packet reception interrupt (*RxDone*) and the valid header interrupt (*ValidHeader*). We use this time difference to calculate the exact reception timestamp of the frame.

As a result of these design decisions, our DSME-LoRa implementation consumes ≈108 kB of ROM and ≈12 kB of RAM on ARM Cortex-M0 CPU, which is enough for common LoRa hardware platforms.

5 EVALUATION ON REAL HARDWARE

We evaluate the DSME-LoRa implementation (see Section 4.1) in a peer-to-peer topology with source devices (TX-only) and sink devices (RX-only), as depicted in Figure 5. During our experiments, each source device transmits data with exponentially distributed interarrival times to a single sink. We vary the number of source devices (N) and the average transmission interval.

Our results include the transmission delay (time between packet schedule and successful reception), time on air, and energy consumption for transmissions during CAP and CFP. For the CAP,

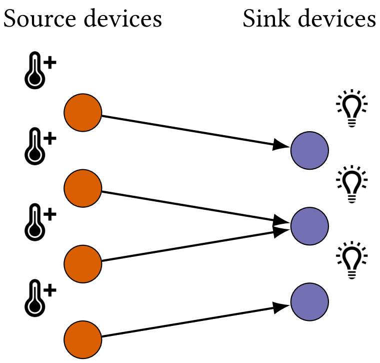


Fig. 5. Topology of a DSME-LoRa network with source and sink devices. One sink device can receive from multiple sources.

Fig. 6. Deployment of LoRa platforms (B-L072z-lrwan1) on the FIT IoT-LAB testbed [3].

we further analyze the impact of CSMA/CA with CAD using different backoff parameters. We also evaluate the impact of cross-traffic between coexistent DSME-LoRa and LoRaWAN networks and the effect of interference on DSME-LoRa.

5.1 Experiment Setup

Testbed deployment. We conduct our experiments in the *Saclay* site of the FIT IoT-LAB testbed, which supplies 25 LoRa boards (B-L072z-lrwan1). These are distributed in a 12 m by 12 m room, as shown in Figure 6. The B-L072z-lrwan1 platform consists of an ARM Cortex-M0 CPU, which runs at 32 MHz, provides 192 kB of ROM/20 kB of RAM, and contains a SX1276 LoRa transceiver. The testbed contributes a *serial_aggregator* tool that aggregates all UART output of the deployment and adds a timestamp. We add logging to our measurement firmware for packet schedule, transmission, reception, and MAC queue lengths and use this information to calculate transmission delay, PRR, and time on air.

Multisuperframe structure. We configure DSME to one superframe per multisuperframe, which results in a multisuperframe duration of T_{msf}=7.68 s. This configuration exposes 7 GTS over

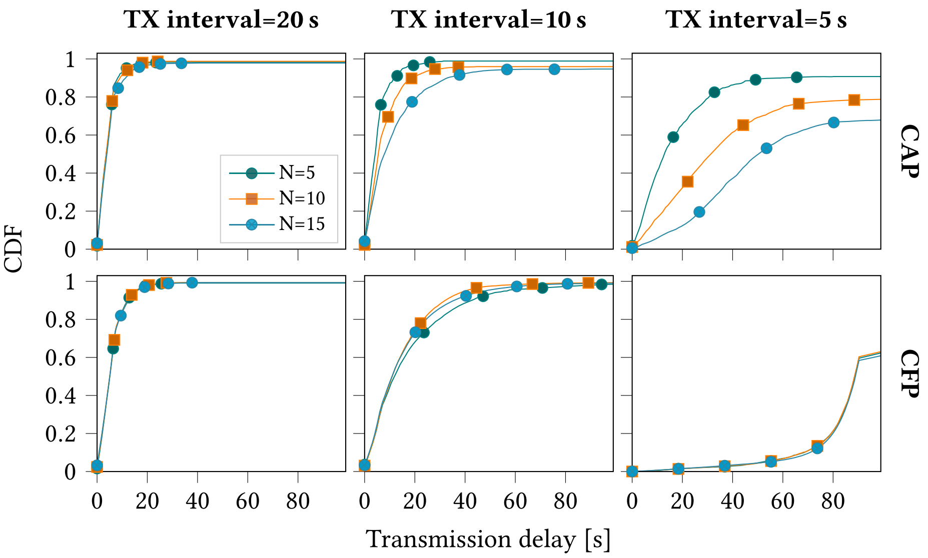


Fig. 7. Transmission delay for unconfirmed transmissions during the CAP (CSMA/CA) and CFP (GTS) with a varying number (N) of source devices and transmission intervals.

16 channels, which enables 112 unique time-frequency slots. We use a beacon interval of two multisuperframes, which results in a beacon period of 15.36 s.

Network topology. A variable number of source devices transmits data to three sink devices using direct communication (gateway-less). This mapping accommodates solely GTS transmissions on the proposed multisuperframe configuration. During bootstrap, a random sink is assigned to each source device. We use static allocation for GTS. With that, we imitate GTS allocation during device association with the DSME Association Request command (see Section 3.1). We deploy an extra device that operates as the PAN coordinator, which establishes the superframe structure by transmitting *enhanced beacons*. Although any sink or source device may operate as the PAN coordinator, we opt for this approach to simplify the deployment.

MAC configurations. If not mentioned otherwise, then we configure the CSMA/CA backoff parameters to `macMinBE=7`, `macMaxCsmaBackoff=5`, `macMaxBE=8` (see Table 1), which are close to the maximum values, to cope with long time on air. Section 5.3 further compares these values to 802.15.4 default values. In agreement with the 802.15.4 standard, we set the maximum number of retransmissions to `macMaxFrameRetries=4`. We utilize the channel hopping mode for GTS transmissions.

5.2 Data Transmission in CAP and CFP

Figure 7 shows the distribution of transmission delay for unconfirmed transmissions during CAP (CSMA/CA) and CFP (GTS) for different network sizes and transmission interval. The intersections between each curve and the right axis reflect the PRR.

CSMA/CA transmission. The transmission delay increases with the network size and lower transmission intervals. In both cases, the on-air traffic increases, which leads to enhanced wireless interference. As a result, the CCA procedure faces more often a busy channel, which increments the number of CSMA/CA backoff periods per transmission. This effect increases the delay between packet schedule and the actual transmission and causes a higher transmission delay. Similarly, the

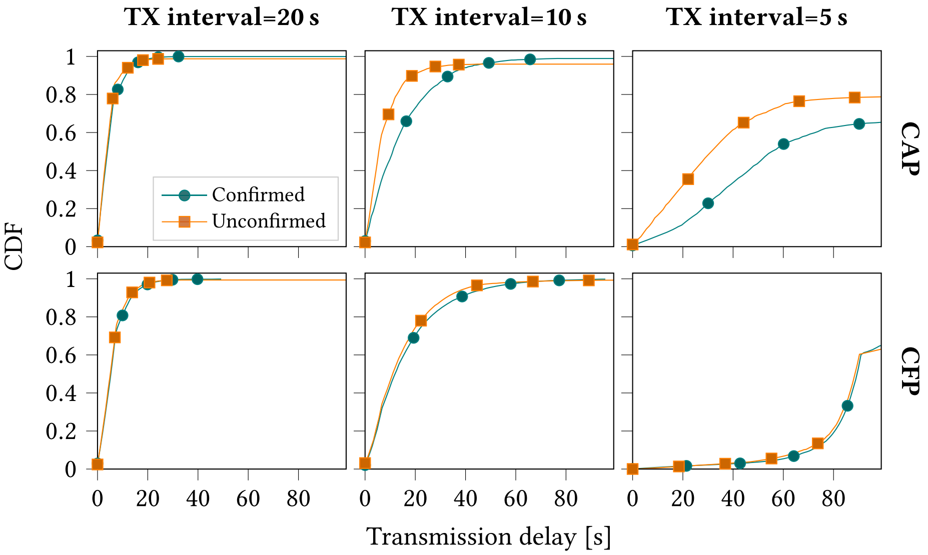


Fig. 8. Comparison of transmission delays for confirmed and unconfirmed transmissions during CAP (CSMA/CA) and CFP (GTS) for 10 source devices and varying transmission intervals.

PRR decreases with the network size and lower transmission intervals. Multiple CCA failures enhance the probability of exceeding the maximum number of CCA retries. The MAC drops the packet in such case, hence, it decreases the PRR. Note that due to inaccuracies of the CCA procedure, the MAC transmits a fraction of packets even when the channel is busy, which decreases the PRR marginally. We further analyze this effect in Section 5.3. Higher transmission delays increase the CAP queue stress, since packets have to be buffered until they are actually sent. Hence, a fraction of packet losses occur due to CAP queue overflows. The stressed scenario (TX interval=5 s) reflects this situation, in which an increasing number of senders decrease the reception ratio.

GTS transmission. The transmission delay increases with a lower transmission interval during CFP, but it does not vary with the network size. In contrast to the CSMA/CA scenario, each source device transmits data during a dedicated time slot that repeats every T_{msf} ≈7.68 s. In the advent of packet queuing for a particular sink device, the last queued packet delays until the MAC transmitted all preceding frames. As a consequence, a lower transmission interval (TX interval=5 s) increases the transmission delay by increasing the average queue occupation—introducing MAC queue stress. This situation explains a larger transmission delay in CSMA/CA than in GTS, most notable in scenarios with TX interval=5 s/10 s, even with high backoff exponent configuration for CSMA/CA. Note that the network size only affects the number of allocated time slots during one multisuperframe. Consequently, the network size does not affect the transmission delay as long as a sufficient number of GTS in the multisuperframe structure exist.

Effect of retransmissions. Figure 8 compares the delay and PRR for confirmed and unconfirmed transmissions during CAP and CFP. Confirmed frames under transmission stay in the MAC queue until the reception of a valid ACK frame. In case of packet loss, the MAC retransmits a pending frame until reception of the ACK frame or running out of retransmission attempts. Following our preceding measurements, we vary transmission intervals in a network of 10 source devices. Confirmed transmissions during CAP reveal a higher transmission delay in relaxed scenarios (TX interval=10 s/20 s), i.e., 90% of confirmed packets finish within 40 s, whereas the same amount of unconfirmed packets finish in less than 20 s. Therefore, the reception ratio increases from 95% to

100% with confirmed traffic. In the stressed scenario (TX interval=5 s), the transmission delay of the confirmed scenario increases as well, while the PRR decreases in comparison to the unconfirmed scenario. Two causes are worth stressing: (*i*) retransmissions increase the on-air traffic, which leads to collisions and a high number of CCA failures; (*ii*) frames in retransmit occupy the MAC queue for a longer time and are dropped occasionally due to CAP queue overflow.

In the CFP, confirmed packets improve the PRR by only ≈0.5% to achieve 100% success. Similar to the CSMA/CA scenario, frame retransmissions increase the probability of packet reception, however, since GTS transmissions are exclusive, retransmits are barely required. Therefore, the contribution of frame retransmissions to MAC queue stress is negligible. Only a few retransmitted frames slightly increase the transmission delay. This effect is notable in the scenario with TX interval=10 s. In contrast to the stressed CSMA/CA scenario, the queue load in the stressed GTS scenario (TX interval=5 s) with confirmed transmissions is similar to unconfirmed transmissions. Hence, the PRR does not decrease any further.

5.3 Effect of CAD under Different CSMA/CA Configurations

Effect on collisions. We evaluate the collision avoidance capabilities of the CAD feature for transmissions during the CAP and compare to the ALOHA protocol, i.e., randomized delay before transmissions. Thereby, we utilize two timing parameters sets for CSMA/CA with CAP and the initial ALOHA backoff.

(1) **High BE:** our default choice (compare Section 5.1) with `macMinBE=7`, `macMaxCsmaBackoff=5`, `macMaxBE=8`.
(2) **Standard BE:** 802.15.4e standard values (for radios that operate in the 2.4 GHz band) with `macMinBE=3`, `macMaxCsmaBackoff=4`, `macMaxBE=5`.

Figure 9 displays the fraction of packets that face a clear channel, collide, are dropped due to maximum number of CSMA/CA reattempts or queue overflow. Naturally, the latter options do not occur using ALOHA. In the ALOHA scenario (Figure 9, left), the results show that collisions increase with a lower transmission interval. Due to higher traffic on air, the chances of packet collision increase up to 38% with 10 nodes and TX interval=5 s in the standard BE scenario (Figure 9, bottom left). Note that scenarios with standard BE CSMA/CA settings show higher collision rates than scenarios with high BE settings (Figure 9, left bottom vs. top). The higher backoff exponent increases the initial TX delay, hence, the average transmission interval, and thereby reduces the probability of collisions. The number of transmitted packets stays constant (100%) in all ALOHA scenarios, regardless of interference or busy channel. This is because the MAC always assumes a free channel at the end of the backoff period and unconditionally transmits the frame.

The number of collisions in the CSMA/CA CAD scenario (Figure 9, right), is smaller than in the ALOHA scenario and increases at a lower rate with a lower transmission interval. In contrast to ALOHA, the number of transmitted packets decreases with a lower transmission interval. This is the positive effect of channel sensing indicated by CCA failures, which avoids sending during ongoing transmissions on the channel. Packets that are delayed due to CCA remain queued and the MAC drops a pending frame if CSMA/CA runs out of retries. This decreases the number of transmitted frames in a stressful scenario (CSMA/CA standard BE, TX interval=5 s). It is worth noting that CAD is affected by inaccuracies; it detects a clear channel if two nodes start CSMA/CA about simultaneously. As a result, a fraction of packets collides despite CCA. Similar to the ALOHA scenario, high BE CSMA/CA settings trigger fewer collisions than standard BE settings. A reduction in transmission rate due to higher backoff delays relaxes the channel, though, a negative side effect is additional CAP queue load, which increases packet losses due to occasional queue overflows (CSMA/CA high BE, TX interval=5 s).

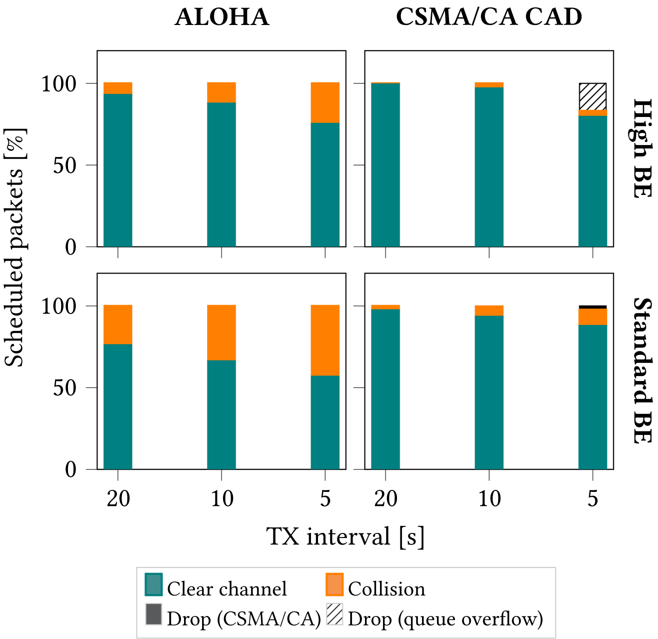


Fig. 9. Proportion of CAP packets that face a clear channel on transmission, collide, or are dropped by CSMA/CA. Results are separated into high BE and standard BE CSMA/CA configurations with and without CAD for 10 source devices and varying transmission intervals.

Effect on transmission delay. The results in Figure 10 show that the transmission delay in the CSMA/CA scenario increases with decreasing TX intervals. Increased channel access failures with CSMA/CA delay the transmissions (until CCA reports clear channel). Hence, the average transmission delay increases. Increasing TX intervals with ALOHA do not affect the transmission delay, since sending is independent of the channel state. ALOHA therefore suffers from wireless interference. In all cases, CSMA/CA leads to a higher PRR than ALOHA transmission. Although CSMA/CA with CAD reduces the proportion of transmitted packets, the number of non-transmitted packets (which avoided a collision) is smaller than collisions upfront. As a result, the PRR increases. Scenarios with high BE CSMA/CA settings reveal higher packet reception ratios, as a result of reduced collisions and higher transmission delay as a result of higher backoff delay.

Effect on retransmissions. Figure 11 analyzes the effect of using CAD when enabling confirmable traffic and retransmissions. We compare the PRR and the average number of retransmissions per packet (Figure 11(b)) for both the CSMA/CA with CAD and ALOHA scenarios. The PRR decreases with higher TX intervals and CSMA/CA CAD measurements outperform ALOHA. The effect is most notable in the stressed scenario (TX interval=5 s), where the difference amounts to ≈8%. Retransmissions remain rare with CSMA/CA, which indicates that losses are mainly caused by avoided transmissions in stressed cases. Nevertheless, CSMA/CA outperforms ALOHA in terms of reception ratio. In contrast, nodes retransmit every packet up to 1.5× (on average) using ALOHA, without improving packet reception. This useless amount of retransmissions demonstrates the advantage of CSMA/CA with CAD.

5.4 Time on Air and Duty Cycle Compliance

Transmission time with LoRa radios is limited by band regulations. We analyze the on-air time with respect to duty-cycle compliance. As depicted in Section 5.1, we set up topologies in which a

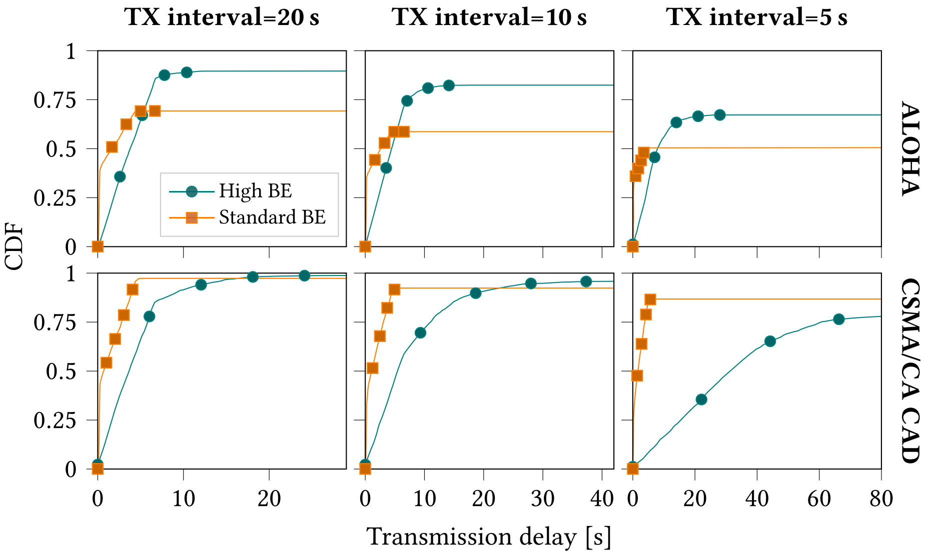
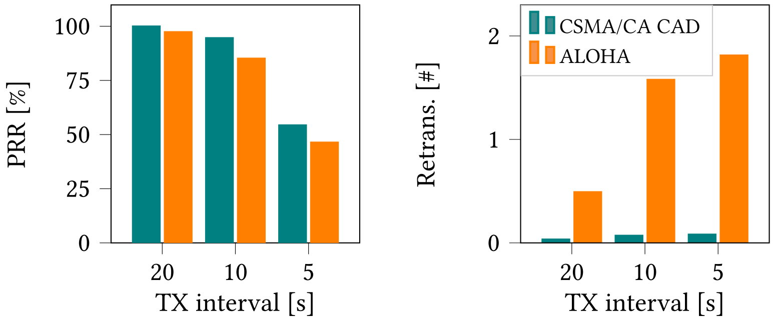


Fig. 10. Comparison of transmission delays between CAP transmissions with CSMA/CA CAD and ALOHA for 10 source devices, with varying transmission intervals and CSMA/CA backoff exponent (BE) configurations.



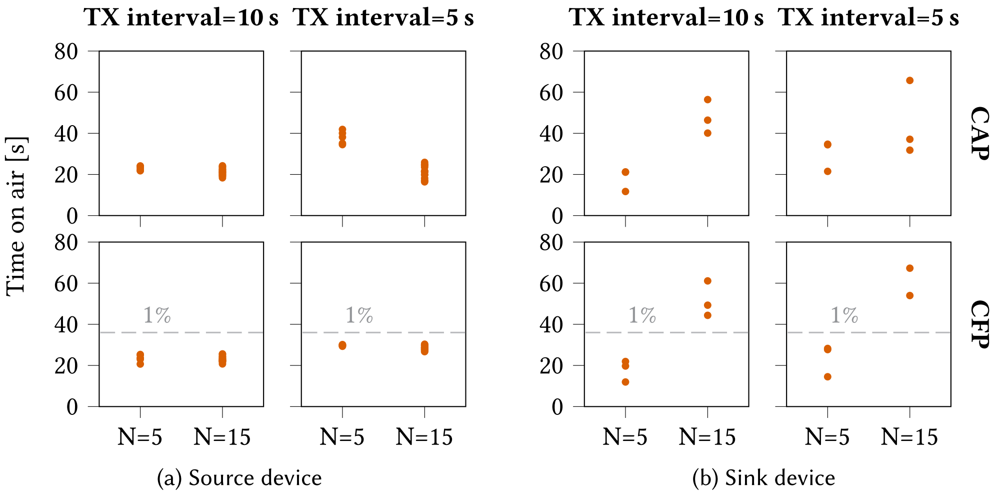
(a) Packet reception ratio

(b) Retransmissions per frame

Fig. 11. Packet reception ratio (left) and average retransmissions per frame (right) for 15 source devices and varying transmission intervals.

variable number of source devices sends data to three sink devices. The assignment is uniformly random. Sink devices, in turn, reply with an ACK to every incoming packet. Note that a data frame contains 27 bytes of data, whereas the ACK frame contains only 5 bytes. Due to the LoRa PHY frame overhead, however, the ACK packet takes ≈31 ms on air, which is around half of the data frame (≈67 ms). We set up mostly stressed TX intervals to foster duty cycle violations and present our results of the time on air per node in Figure 12. Thereby, the dashed gray line indicates the maximum on air time to comply with a 1% band occupation during CFP. For the CAP, we utilize a 10% of the band (see Section 4), which is not visible on this scale.

CSMA/CA transmission. The time on air increases with a lower transmission interval as a result of a higher transmission rate on the MAC (Figure 12(a), top left to right). For the N=5 case, the time almost doubles from ≈20 s to 40 s with half the TX interval. Note that we apply a 10% band to the CAP, hence, on-air times of up to 360 s per node do comply with duty cycle restrictions.



(a) Source device (b) Sink device

Fig. 12. Time on air of data frames on source devices (left) and ACK frames on sink devices (right) compared to 1% band limitations (gray line). Frames are sent in CAP (CSMA/CA) or CFP (GTS), and we vary the number of source devices (N) and transmission intervals.

Our results further show that the average time on air of source devices decreases in larger networks (N increases). The effect is most notable in the scenario with TX interval=5 s. In that scenario, CCA fails more often due to higher on-air traffic, which persuades the MAC to drop a fraction of data frames, either due to exceeded CSMA/CA attempts or overflowed CAP queue (see Section 5.3). The case with N=15 source devices does not vary on air time with varying TX intervals due to these MAC drops.

In the case of sink devices, the time on air increases with a lower transmission interval, similarly to source devices. In contrast, though, the time on air also increases with the network size. ACK packets are sent in response to every incoming source device frame and do not utilize CSMA/CA. Due to our topology choice, a sink device has to return multiple ACK packets to satisfy all its assigned source devices. Consequently, a higher number of source devices leads to a higher ACK frame transmission rate per sink device, which increases the time on air up to 60 s for scenarios with N=15 source devices. This is still in line with 10% restrictions. Note that the random source-sink assignment leads to a different number of source devices per sink on each scenario, which introduces variations between time on air measurements across sink devices.

Sink devices send multiple ACK packets back-to-back, contrasting a "simultaneous" channel access of source devices that introduced MAC dropping. Overall, increasing transmission rates have a less severe impact on node duty cycles than increasing number of nodes that try to access the medium during the same time period.

GTS transmission. The average time on air of source devices increases with a lower TX interval, which is in agreement with CSMA/CA transmissions (Figure 12(a), bottom left to right). Our CFP assignment with one GTS per source-sink link, however, limits the effective TX interval to T_{msf}=7.68 s in our multisuperframe configuration as described in Section 5.1. Note that this configuration preserves duty cycle compliance natively. Sending a 67 ms long packet every 7.68 s results in 31.4 s active send time per hour, which is below the 1% regulation mark of 36 s per device and hour. Conversely, we intentionally chose a very stressful measurement setup with TX interval=5 s.

Similar to the relaxed CSMA/CA scenario, the average time on air of sink devices increases with a lower TX interval (Figure 12(b), bottom left to right) due to an increased frame rate. Increasing the

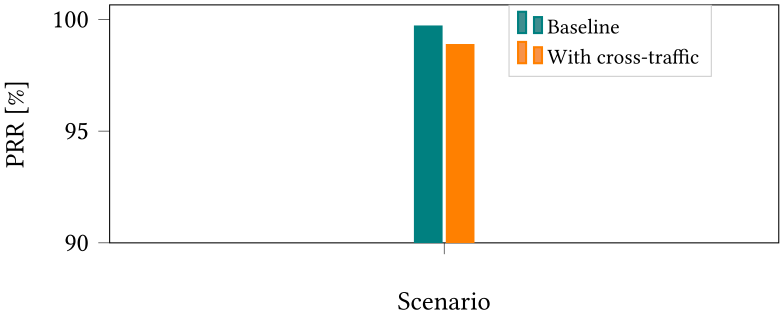


Fig. 13. Comparison of PRR for 10 DSME-LoRa source devices with and without cross-traffic from a LoRaWAN network with 10 class A devices. All devices transmit unconfirmed frames with 16 bytes payload and uniformly distributed interarrival times between 7 and 13 s. DSME-LoRa devices transmit during CFP (GTS).

network size, in contrast to CSMA/CA, further increases the on-air times of sink devices. Observe that the time on air exceeds 1% of duty cycle in scenarios with N=15. The reasons for this are threefold: (*i*) Due to our topology choice, each sink device has to confirm five source packets on average in the N=15 scenario. This amplification burdens the link budget of a single sink device. Hence, we deliberately violate the duty cycle regulations by our experiment setup. (*ii*) Sink devices only send ACK frames back by back and without CSMA/CA. Hence, sink devices transmit 100% of the scheduled ACK frames. (*iii*) GTS transmissions utilize guaranteed resources, which increases the reception ratio and decreases losses in comparison to CSMA/CA transmissions. As a result, the number of transmitted ACK frames is in line with the number of transmitted data frames, regardless of the network size.

5.5 Coexistence with LoRaWAN

The proposed DSME-LoRa PHY channels overlap with channels of LoRaWAN networks. Therefore, we are interested in the effects of LoRaWAN cross-traffic in DSME-LoRa networks. For this analysis, we focus only on cross-traffic between GTS transmissions and LoRaWAN traffic for two reasons: (*i*) The common CAP channel does not overlap with standard LoRaWAN uplink channels. (*ii*) LoRaWAN downlink traffic is typically transmitted using a higher spreading factor and thereby does not collide with DSME-LoRa packets.

For the evaluation, we deploy DSME-LoRa and LoRaWAN networks simultaneously and measure the PRR of the DSME-LoRa network. We compare these values against the same DSME network without cross-traffic.

For the LoRaWAN network, we set up 10 nodes with class A transmissions and DR5 (spreading factor 7, bandwidth 125 kHz). The deployment uses a single 8-channel TTN [11] LoRaWAN gateway, available in the testbed. The DSME-LoRa network consists of 10 source devices with the topology and configuration in agreement with Section 5.1. All devices transmit 16 bytes payload using unconfirmed transmissions and uniformly distributed interarrival times between 7 and 13 s.

Figure 13 shows the PRR of the isolated DSME-LoRa network (baseline) and the network with LoRaWAN cross-traffic. With DSME cross-traffic, the PRR reduces ≈0.7%. From all available DSME-LoRa channels, only seven overlap with TTN LoRaWAN channels, which means 56.25% of transmissions are collision-free. A fraction of the remaining packets is transmitted concurrently with DSME-LoRa transmissions, which reflect the PRR reduction. The LoRaWAN traffic does not collide with DSME beacons and, therefore, device desynchronization as a result of collision between LoRaWAN packets and DSME-LoRa beacons is negligible. Even though LoRaWAN traffic degrades

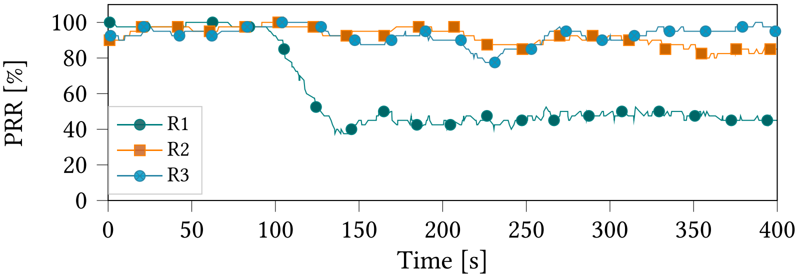


Fig. 14. Evolution of PRR for three replicas of a DSME-LoRa deployment with 10 source devices, unconfirmed GTS transmissions, and TX interval=10 s, under heavy interference on the common channel.

PRR as a result of concurrent transmissions on shared channels, the cross-traffic does not prevent normal operation of the DSME-LoRa network. We conclude that DSME-LoRa traffic is compatible with standard LoRaWAN uplink traffic.

5.6 Effect of Interference in Common Channel

Section 5.5 confirmed that DSME-LoRa networks tolerate channel interference in GTS channels. However, the results do not reflect tolerance to noise in the common channel used for CAP and beacon transmissions. While common LoRaWAN deployments in the EU868 region do not transmit in the 10% band (CAP channel) using the same PHY settings as DSME-LoRa, the LoRaWAN network server does not prevent the configuration of a downlink channel using spreading factor 7, which may cause LoRaWAN frames to collide with DSME-LoRa frames. Since synchronization to the DSME superframe structure relies on beacons, we evaluate whether DSME-LoRa can operate under noise in the common channel. We focus only on GTS transmission, because the effect of noise during CSMA/CA transmissions has already been analyzed in Section 5.2.

We generate a harsh environment by deploying five jammer devices that send 127 bytes payload data in the common channel with uniformly distributed interarrival times between 1 s and 6 s, next to the DSME-LoRa network with 10 source devices. The interarrival time of packets and the MAC configurations of the DSME-LoRa network are identical to the network in Section 5.5.

Figure 14 shows the moving average PRR over time for three replicas (R1, R2, and R3). The PRR of all replicas oscillates around 92% before T=100 s. After T=100 s, the PRR in R3 reduces to ≈50% and does not recover. Similarly, around T=270 s the PRR in R2 drops to ≈89%. To understand these results, observe that the common channel is included as one of the transmission channels in CFP. Therefore, GTS transmissions in the common channel are likely to collide with traffic from the jammers. Assuming $\frac{1}{16}$ of GTS vulnerable transmissions (i.e., one channel gets jammed), around 94% of transmission are collision-free. This reflects the lower average PRR before T=100 s. The PRR drops in R2 and R3 are caused by the desynchronization of two sink devices and a source device, respectively, as a result of beacon loss. When the MAC misses a number of consecutive beacons (4 by default), the device disassociates and ignores all transmission requests and GTS reception slots. Therefore, incoming and outgoing packets are simply discarded.

To summarize, interference on a single channel reduces the efficiency of GTS transmissions but does not prevent normal operation. Interference on the common channel, however, increases beacon loss, which desynchronizes devices from coordinators. While raising the threshold of consecutive missed beacons can delay desynchronization on sporadic interference, it cannot solve the problem. Additionally, long time on air and long range of LoRa frames make wireless attacks plausible, in which an attacker blocks beacon reception through channel jamming. We discuss potential solutions to this problem in Section 8.5.

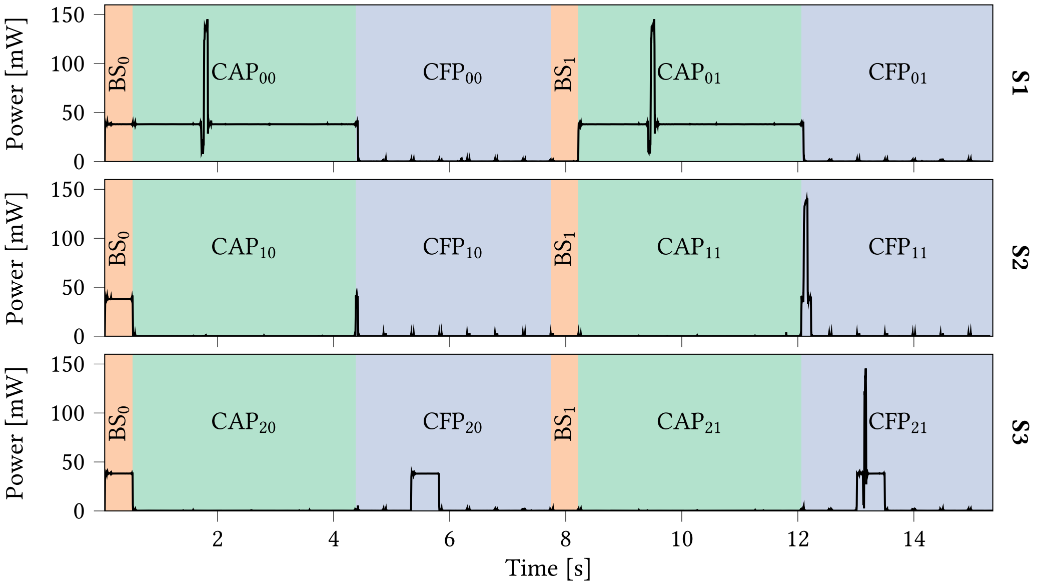


Fig. 15. Power consumption during one beacon interval with two multisuperframes separated into CSMA/CA transmissions (top), GTS transmissions with transceiver off during CAP (middle), and GTS receptions with transceiver off during CAP (bottom).

5.7 Energy Consumption

We evaluate the power consumption on the target board using a digital multimeter (Keithley DMM7510 7 1/2). Therefore, we sample the current consumption at 100 kHz and provide the board with an externally stabilized voltage supply. Our analyses are separated into passive and active consumption. Passive consumption includes the maintenance of the superframe structure without data transmission. Active consumption, in contrast, includes the transmission and reception of data and ACK frames in different superframe periods. Hence, the total consumption of a node consists of both passive and active components. Figure 15 represents the passive power consumption over time during one beacon interval and three traffic options. It is noteworthy, though, that we exemplarily include active TX/RX spikes in the plot for presentative reasons.

(1) **S1** enables transmission during CAP. This requires both sender and receiver to enable the transceiver during that period (Figure 15, top).
(2) **S2** disables the CAP to save power and represents the case for sending data during one GTS (Figure 15, middle).
(3) **S3** is similar to (2), however, it displays data reception during one GTS (Figure 15, bottom).

Figure 16 represents active power consumption for the sender and receiver of a frame with CSMA/CA (used in the CAP) as well as without channel sensing (in the CFP). Table 4 integrates the power over dedicated intervals and presents the energy consumption for passive (top part) and active (bottom part) actions. In the remainder of this section, we will first analyze passive and active components separately. We then evaluate the total energy consumption and present our results in Table 5. In all measurement configurations, we set the transmission interval to TX interval=20 s and the payload size to 16 bytes.

Passive consumption (Table 4 top). During Beacon Slot $(BS)_0$ the MAC turns the transceiver on for the duration of the beacon slot (0.48 s) to receive the beacon from its coordinator. The energy for

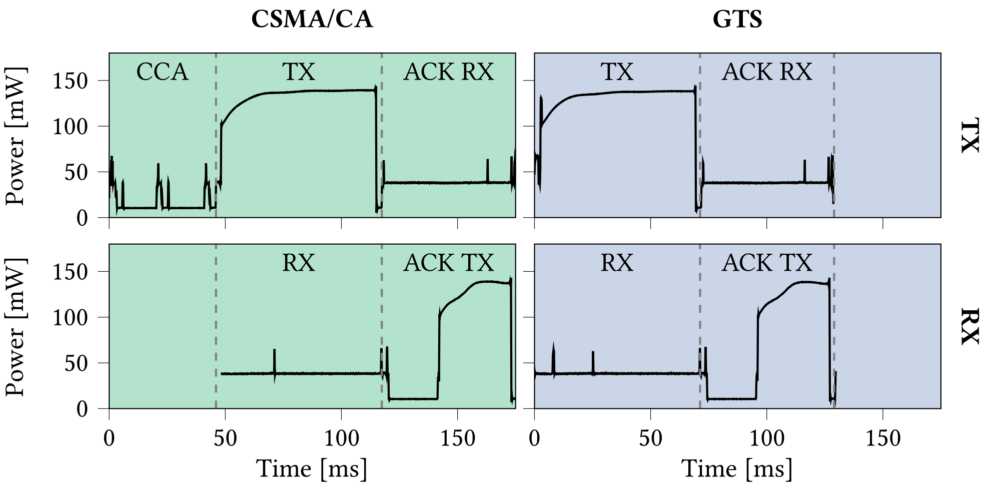


Fig. 16. Power consumption for transmission and reception of one packet with CSMA/CA (left) and in a guaranteed time slot (right).

Table 4. Energy Consumption for Each Superframe Period

	Period	State	Additional Description	Energy [mJ]
Passive	BS	RX	Beacon synchronization	18.28
		RX off	Inactive beacon slot	0.08
	CAP	RX idle	*macRxOnWhenIdle=1*	146.12
		RX off	*macRxOnWhenIdle=0*	0.50
	CFP	TX idle	Single GTS allocation	2.20
		RX idle	Single GTS allocation	19.31
		RX off	No GTS allocation	0.69
Active	CAP	TX	Single CSMA/CA TX	12.04
		RX		6.44
	CFP	TX	Single GTS TX	11.27
		RX		6.44

Passive (top) components relate the maintenance of the superframe structure under varying transmissions options. Active (bottom) components relate to the actual data transmission. *State* relates to the following components idle – device is ready for operation w/o ongoing transmissions; off – device is not operable to save energy; TX/RX – transmission/reception of frames.

beacon synchronization is 18.28 mJ for CPU processing, listening, and receiving (RX). During BS_1 the MAC repeats the superframe structure and begins with a new inactive beacon slot, reserved for beacon collision avoidance (see Section 3.1). The MAC keeps the transceiver off (RX off) during that time and the energy consumption reduces to 0.08 mJ.

The MAC switches to the CAP after a beacon slot. In the S1 scenario (Figure 15, top), the transceiver stays idle listening during CAP_{00} for 3.84 s (RX idle, *macRxOnWhenIdle=1*), which consumes 146.12 mJ. This high consumption shows the need for battery powered devices to turn the transceiver off during CAP.

The S2 scenario (Figure, 15 middle) reflects that the transceiver is turned off (RX off, *macRxOnWhenIdle=0*) during CAP_{10}, as it reduces the consumption during CAP to 0.5 mJ, only for

Table 5. Passive and Active Energy Consumption per Beacon Interval [mJ], for the CSMA/CA Scenario S1 and both GTS Scenarios S2 & S3

	S1		**S2**		**S3**	
Period	Energy [mJ]	Prop. [%]	Energy [mJ]	Prop. [%]	Energy [mJ]	Prop. [%]
BS	18.36	5.68	18.36	45.15	18.36	28.74
CAP						
passive	292.23	90.45	0.99	2.43	0.99	1.55
active	11.12	3.44	0.00	0.00	0.00	0.00
CFP						
passive	1.38	0.43	4.40	10.82	38.62	60.45
active	0.00	0.00	16.91	41.59	5.92	9.27
Total	**323.09**	**100**	**40.66**	**100**	**63.89**	**100**

maintenance purposes (i.e., timers, interrupts, etc.). This makes the node, however, unavailable for packet reception during that period. The CFP follows the CAP (T=4.32 s) and the MAC switches to slot mode. Without a GTS allocation, the transceiver stays off (RX off) and a system wake-up for internal housekeeping requires 0.69 mJ, which is similarly low as the sleep mode of the CAP. In the presence of an allocated GTS TX slot in the CFP, the MAC turns the transceiver on (TX idle) before the GTS to prepare the next transmission. An empty transmission queue triggers the immediate shutdown of the transceiver to save energy. This situation reflects the power peak in CFP_{10} (T=4.32), which consumes no more than 2.20 mJ and can be mitigated by slot deallocation. An actual GTS transmission is visible in CFP_{11}.

Scenario S3 (Figure 15, bottom) presents the corresponding consumption in CFP_{20} to receive during one GTS. Here, the MAC enables the transceiver during one full GTS duration (RX idle), which requires 19.31 mJ and provides a frugal alternative to the CAP receiver. An actual GTS reception on top of the baseline is displayed in CFP_{21}.

Active consumption (Table 4 bottom). At the beginning of CSMA/CA transmission (Figure 16, top left), the MAC waits for the duration of the backoff period and performs three consecutive CCA measures, which consumes 0.73 mJ. On clear channel, the transceiver loads the frame and performs the frame transmission (TX), which consumes 9.07 mJ, followed by ACK frame reception at 2.23 mJ. In total, this makes ≈12.04 mJ for CSMA/CA transmission during CAP.

The CSMA/CA receiver (Figure 16, bottom left) consumes 2.75 mJ for the bare frame; however, awaiting the turnaround time and sending the ACK back consumes additional 3.69 mJ. Hence, pure receiving (RX) requires 6.44 mJ, which appears low compared to transmission. It, however, requires an active CAP, which consumes >20 times more energy (see Table 4).

On transmission during GTS (Figure 16, top right), the MAC loads the frame into the transceiver buffer without a preceding CCA, immediately transmits, and receives an ACK. This total consumption of 11.27 mJ outperforms the CSMA/CA sender slightly. Note that the device turns off the transceiver if the MAC queue is empty, which further reduces the passive CFP consumption occasionally.

Similar to the reception during CSMA/CA, the reception during GTS (Figure 16, bottom right) turns the receiver on for the duration of the frame, delays, and transmits the ACK frame, which leads to the same consumption. In contrast, however, GTS receivers can turn the transceiver off during CAP.

Total energy consumption. Table 5 presents the total energy consumption and proportions during BS, CAP, and CFP, for the three scenarios in Figure 15. We normalize the consumption to

Table 6. List of Variables Used in Our Model for DSME-LoRa

Variable	Description
$L(t)$	Packets in queue at time t
T_n	Time at the end of the slot n
L_n	Queue length at T_n, equivalent to $L(T_n)$
S	Elapsed time since the end of the last slot ($0 \leq S < T_{msf}$)
$N(t, s)$	Number of packets scheduled between t and $t + s$
λ	Transmission schedule rate
T_{msf}	Duration of a multisuperframe
ρ	System utilization, defined as $\lambda \cdot T_{msf}$
D	Number of neighboring devices
$L_d(t)$	Number of queued packets with destination to neighbor $d \in [0, D)$ by definition, $L(t) = \sum_{d=0}^{D-1} L_d(t)$

one beacon interval and present average values from 10 measurements. Results are separated into passive and active operations in alignment with the preceding micro analysis. All three scenarios, unsurprisingly, consume the same amount of energy (18.36 mJ) for maintaining the beacon slot. In the S1 scenarios, over 90% of the consumed energy accounts to passive CAP consumption—for keeping the radio on—whereas only ≈3.5 % is used for sending. Since the CFP is not actively used, the overhead remains small (<0.5%). We also conducted measurements at the receiver side in this scenario, but the variation remains negligible. As the results do not contribute to additional insights, we excluded these experiments. The overall consumption for scenarios S2 and S3 behaves similar. The CAP remains unused to save energy. Thus, only the passive component consumes 1 mJ (<2.5%). The GTS scenarios save about 290 mJ over S1. The main proportion is spent in the CFP. Active sending in a GTS (≈17 mJ) is more expensive than receiving (≈6 mJ). Conversely, turning on the transceiver for a GTS duration (≈39 mJ) consumes four times more energy than scheduling a transmission slot without sending (≈11 mJ). In total, this leads to a 50% higher consumption of the GTS receiver scenario S3. Comparing S1 to S2 and S3, a total consumption of 323.09 mJ reveals a notable overhead to the GTS alternatives, which require five to eight times less energy. Stress during CAP (see Section 5.3) can further worsen the energy excess for CSMA/CA reattempts and retransmits. Conversely, S1 scenarios (*i*) are required to associate slots in S2 and S3 in a real-world deployment and (*ii*) enable sporadic transmissions without association overhead. In practice, however, mixed scenarios can build a compromise to facilitate moderate consumption in flexible deployments.

6 ANALYTICAL STOCHASTIC MODEL

An important measure for the feasibility of our solution is its performance under continuous network load. To evaluate this, we introduce an analytical stochastic model, which allows to calculate the stationary probability distributions of the MAC queue length at an arbitrary time and for transmissions during CFP (GTS). Our symbols and nomenclature are summarized in Table 6.

The temporal evolution of the MAC queue at a DSME-LoRa device is visualized in Figure 17. Packets arrive randomly over time and are added to the queue. At the end of an arbitrary slot T_n, packets are transmitted and removed from the queue. Correspondingly, the number of queue entries at the end of slot n is L_n, while $N(T_n, S)$ packets are added in the time span S after T_n. In a homogeneous process, $N(T_n, S)$ is proportional to S.

6.1 A Markov Queuing Process

We model our DSME-LoRa transmission system as a simple Markov queuing process, For this, we make the following simplifying assumptions:

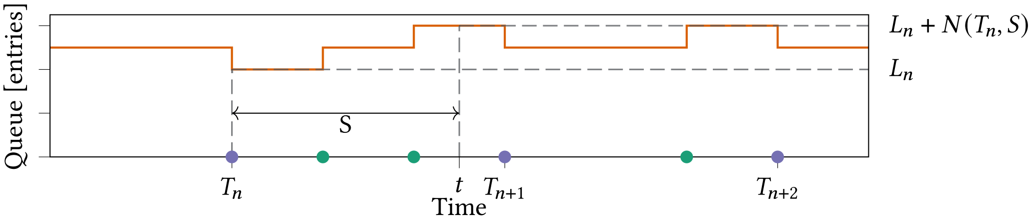


Fig. 17. Qualitative evolution of the MAC queue over time: Occupation grows on packet scheduling events (green) and reduces at slot occurrences (purple).

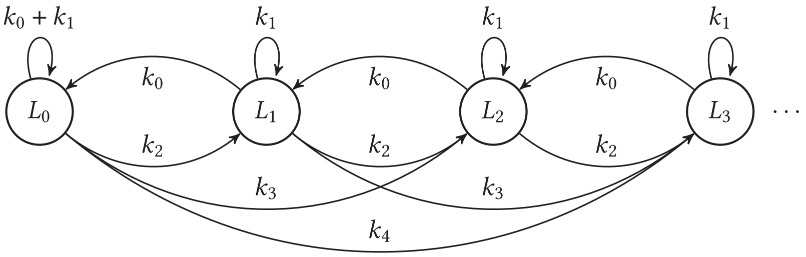


Fig. 18. Embedded Markov chain: The queue occupation L_i remains constant between slots if only one packet arrives (k_1). It increases by $i - 1$ for k_i packet arrivals and decreases for k_0.

(1) packets arrive independently
(2) exponentially distributed interarrival times between scheduled packets with $\rho < 1$ ($\lambda < \frac{1}{T_{msf}}$)
(3) MAC queue has unlimited capacity
(4) unacknowledged transmissions and 100% packet reception ratio
(5) transmission time on air is neglected.

We start by considering transfer to only one neighbor ($D = 1$). Thereafter, we extend the model under moderate conditions to scenarios with $D > 1$. We further analyze this situation in Section 6.4.

Our Markov queuing model is shown in Figure 18. The state of the queue L_i is reduced by one at the end of every time slot. Packets arrive randomly during any time interval s in the queue of the system and follow a Poisson process with parameter λs. For a complete multisuperframe time, let us denote

$$k_i = P\{N(T_n, T_{msf}) = i\} = \frac{\rho^i \cdot e^{-\rho}}{i!}$$

the probability of i packets arriving during one multisuperframe ($\rho = \lambda \cdot T_{msf}$ denotes the arrival intensity, i.e., system utilization). Then the transition arcs of the Markov matrix P are defined by:

$$P[i,j] = \begin{cases} k_0 & \text{if j-i= -1} \qquad (1) \\ k_0 + k_1 & \text{if i=0, j=0} \qquad (2) \\ k_{j-i+1} & \text{if j} \geq \text{i} \qquad (3) \\ 0 & \text{otherwise.} \qquad (4) \end{cases}$$

The probability of a queue reduction corresponds to no packet arrival (Equation (1)), of a growth by (j-i) to j-i+1 packet arrivals (Equation (3)), and a constant initial condition to either none or one packet arriving (Equation (2)).

6.2 Queue Length

For calculating the actual queue occupation, we note that the number of queued packets at an arbitrary time is $L(t) = L_n + N(t-S, S)$, as seen in Figure 17.

Consequently, the distribution of queue length is given as:

$$\begin{aligned} P\{L(t) = i\} &= P\{L_n + N(t-S,S) = i\} \\ &= \sum_{j=0}^{i} P\{L_n = i - j, N(t-S,S) = j\} \\ &= \sum_{j=0}^{i} P\{L_n = i - j\} \cdot P\{N(t-S,S) = j\}. \end{aligned} \tag{5}$$

We first derive a result for $P\{N(t-S,S) = i\}$. Note that $P\{N(t-S,S) = i \mid S = s\}$ is a Poissonian with parameter $(\lambda \cdot s)$ and S is uniform in $(0, T_{msf})$. Therefore, we can calculate $P\{N(t-S,S) = j\}$ via the law of total probability:

$$\begin{aligned} P\{N(t-S,S) = j\} &= \int_0^\infty P\{N(t-S,S) = j | S = s\} \cdot P\{S = s\} ds \\ &= \int_0^{T_{msf}} \frac{(\lambda s)^j e^{-\lambda s}}{j!} \cdot \frac{1}{T_{msf}} ds = \frac{1}{\rho} \cdot \Gamma(j+1, \rho), \end{aligned} \tag{6}$$

where $\Gamma(j, x) = \frac{\int_0^x t^{j-1} e^{-t}}{\int_0^\infty t^{j-1} e^{-t}}$ is the regularized lower incomplete gamma function.

For the calculation of $P(L_n = i)$, observe that L_i, $\forall i \in [0, \infty)$ is a Markov chain (Figure 18), for which we search the stationary distribution. We also observe that the Markov chain is ergodic (positive recurrent and aperiodic). Thus, the stationary distribution $\pi_i = \lim_{n \to \infty} P(L_n = i)$ exists and complies with $P^T \vec{\pi} = \vec{\pi}$. The calculation of a closed-form analytical solution for π_i is not trivial. We describe a detailed numeric procedure to calculate the vector in Section A.

Combining the stationary distribution of the Markov chain and Equation (6) into Equation (5) leads to the distribution of queue length:

$$P\{L(t) = i\} = \sum_{j=0}^{i} \pi_{i-j} \frac{\Gamma(j+1, \rho)}{\rho}.$$

It is possible to calculate the average queue length directly. Observe that $E(L(t)) = E(L_n) + E(N(S))$, where $E(L_n) = \sum_{i=0}^{\infty} i\pi_i$ and $E(N(S)) = \int_0^{T_{msf}} \rho \frac{1}{T_{msf}} dt = \frac{\rho}{2}$.

Therefore,

$$E(L(t)) = \sum_{i=0}^{\infty} i\pi_i + \frac{\rho}{2}. \tag{7}$$

6.3 Transmission Delay

We now calculate the distribution of transmission delay in multiples of T_{msf} from the distribution of queue length:

$$P\{W \le nT_{msf}\} = P\{L(t) \le n - 1\}.$$

As an example, the fraction of packets with transmission delay less than one multisuperframe is $P\{L(t) \le 0\} = \pi_0 \frac{1 - e^{-\rho}}{\rho}$.

Little's Law [41] $L = \lambda W$ calculates the average number of queued items (L) using the arrival rate (λ) and average waiting time W. We use the result to calculate the average transmission delay directly:

$$W \quad = \quad \frac{1}{\lambda}E(L(t)) = \frac{1}{\lambda}\left(\sum_{i=0}^{\infty} i\pi_i + \frac{\rho}{2}\right). \tag{8}$$

6.4 Allocation of Multiple GTS

So far, the model assumes only one neighbor device ($D = 1$) and the allocation of only one slot. The model, however, is still valid for $D > 1$ if each target device allocates only one slot per multisuperframe. In such case, the MAC utilizes the queue as multiple independent FIFO sub-queues (L_d). As a result, the model is valid for each sub-queue and the distribution of the total queue length is:

$$P\{L(t) = i\} = P\left\{\sum_{d=0}^{D-1} L_d(t) = i\right\}.$$

Note that the average queue length is the sum of all average sub-queue length and the average transmission delay the fraction of the average queue length and the total schedule rate (Little equation).

The proposed formulas, however, cease to hold if the MAC allocates more than one slot to the same neighbor. Nevertheless, $D = 1$ sets the worst-case scenario for transmission delay and queue length over these scenarios.

6.5 Validation of the Model

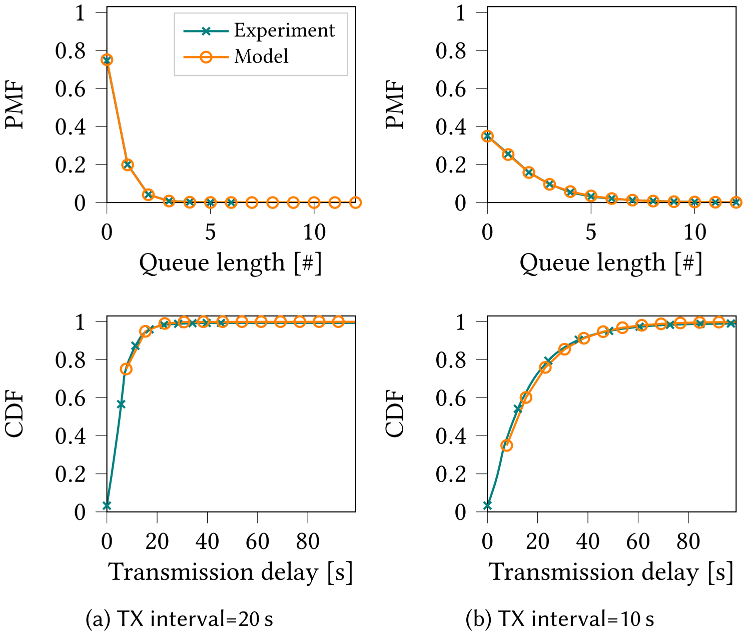
We validate the model accuracy for the distribution of queue length and transmission delay. We chose the GTS transmission scenario with the highest rate of transmitted packets, namely, with 15 source devices, minimize the effect of the transient queue. We do not include the scenario with TX interval=5 s, because it is shorter than the multisuperframe duration (7.68 s). In such case the model does not converge. For the calculation of theoretical results, we clipped the Markov matrix to 100 elements.

In Figure 19, we validate our model by comparing to experimental results (see Section 5). Figure 19(a) compares the probability mass function of the queue length at packet schedule between the results of the experiments and the model. The model predicts the distribution of queue length with more than 99.99% of accuracy. In the relaxed scenario, the probability of more than five elements in the queue is $6.17 \cdot 10^{-5}$, which is consistent with the observation that the queue does not exceed this value. Similarly, the model predicts the transmission delay with an accuracy of 99.99%, as seen in Figure 19(b).

The small variations between the experiment results and the model are due to the effect of the transient queue and a small fraction of packet losses. The former effect mitigates either with a bigger network size or with a longer experiment run.

7 SIMULATION STUDY: ASSESSMENT OF LARGE-SCALE ENSEMBLES

We proceed to evaluate the performance of DSME-LoRa for a larger networks using the *INET [28]/ OMNeT++ [63]* based on our simulation environment [4]. The simulator utilizes the radio module of *FLoRa* [60] and the OMNeT++ adaptation of *openDSME* [31], namely, *inet-dsme*, for the MAC implementation. We extend *inet-dsme* to enable DSME communication over the LoRa radio, as shown in Figure 20. We reuse the traffic generator application of *inet-dsme*, namely, *PRRTrafGen*, which bases on the *IpvxTrafGen* traffic generator module of INET. We utilize the *nextHop* module of INET to resolve L3 address from packets into the destination MAC address.



(a) TX interval=20 s

(b) TX interval=10 s

Fig. 19. Validation of the analytical stochastic model with experimental results. Comparison of distributions of the queue length and transmission delay for varying transmission intervals.

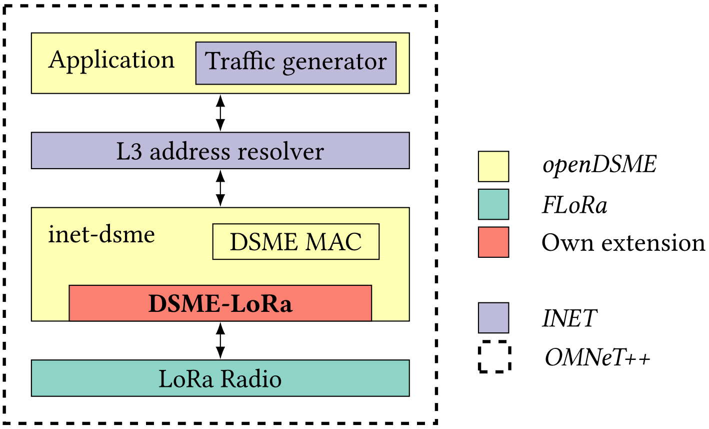


Fig. 20. DSME-LoRa simulation environment and our contribution.

7.1 Validation of the Simulation Environment

To validate our simulator, we first compare simulation results with real-world measurements on hardware (see Section 5). In particular, Figure 21 compares simulation results and our experiments conducted in Section 5.2. In CSMA/CA transmissions, a fraction of collided packets are successfully decoded (capture effect). The fraction of packets varies between the simulation and the experiment, which lead to a different number of retransmissions and dropped frames by the MAC. In GTS transmissions, the collision-free transmission renders high reception ratio for MAC transmissions in the simulation. In the experiment, in practice, a tiny fraction of transmitted frames is lost, which increases the number of retransmissions. Overall, the results of the simulator converge with the

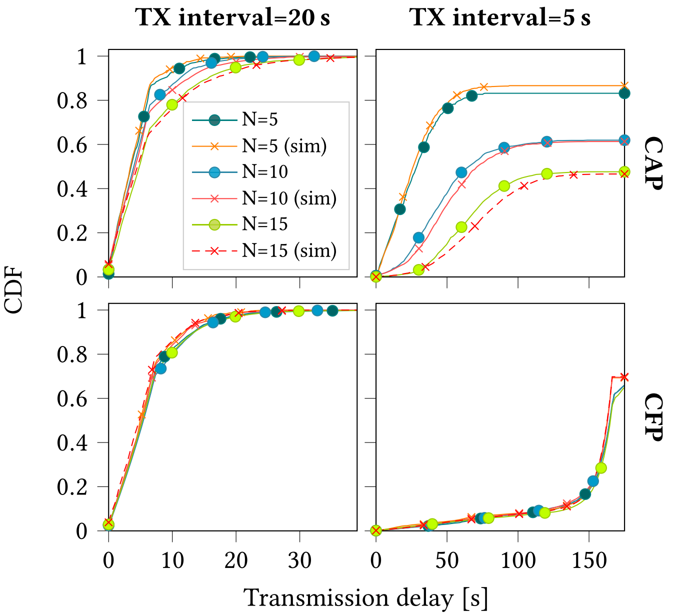


Fig. 21. Comparison of transmission delays from simulations and experiments for confirmed transmissions during CAP (CSMA/CA) and CFP (GTS). We vary the number (N) of source devices and the transmission interval.

experiments results. Differing behavior between the physical channel and the simulation channel model explain small variations.

7.2 Large-scale Peer-to-peer Communication

We evaluate the transmission delay and packet reception ratio of confirmed CSMA/CA and GTS transmissions, for varying network sizes (N = 100 and N = 300) and varying transmission interval (Section 5.1). In agreement with Section 5.1, we use high backoff exponent settings to minimize packet collision. To accommodate one slot for every source device during CFP, we configure the multisuperframe order to 5, which renders 28 GTS and a multisuperframe duration of 30.72 s (Table 2). Figure 22 presents our results.

CSMA/CA transmission. Our results show that the small network (N=100) renders a 95% packet reception ratio in the relaxed scenario (Figure 22, top left). The high on-air traffic due to the network size increases packet collisions and CCA failure rates, as analyzed in Section 5. As a result, a fraction of packet is lost. With half of the transmission interval (Figure 22, bottom left), the small network reduces the packet reception ratio to ≈60% as a result of the higher on-air traffic.

Scenarios with big networks render an even higher on-air time, which reflects ≈38% packet reception ratio in the relaxed scenario (Figure 22, top left, N=300) and ≈14% in the stressed scenario (Figure 22, bottom left). This concludes that CSMA/CA is not reliable for large-scale deployments.

Observe that the transmission delay of the majority frames does not exceed 10 s even in the stressed scenario (Figure 22, bottom left, N=300). This value reflects the worst-case transmission (maximum CSMA/CA retries and maximum frame retransmissions). The delay in the worst case is lower than the TX interval in both CAP scenarios. Therefore, the stress in the CAP queue is low, hence the transmission delay.

GTS transmission. In the relaxed scenario (Figure 22, top right), the transmission delay hits the maximum value at ≈120 s in both network sizes as a result of the delay of queued packets. As

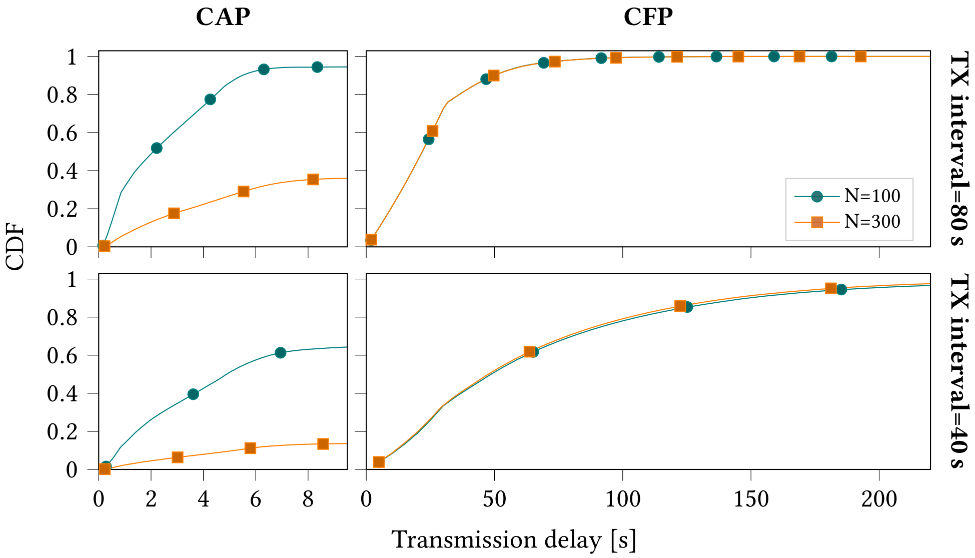


Fig. 22. Comparison of transmission delays for relaxed and stressed scenarios, during CAP (CSMA/CA) and CFP (GTS), for confirmed transmissions and a varying number (N) of nodes.

per Section 5, the transmission delay does not vary with the network sizes, because all devices have equal GTS resources (one slot per multisuperframe). In contrast, in the stressed scenario (Figure 22, bottom right) the transmission delay, as a result of the higher MAC queue stress, hits the maximum at ≈500 s (not shown in the subfigure). Similar to the relaxed scenario, the transmission delay does not vary with the network size.

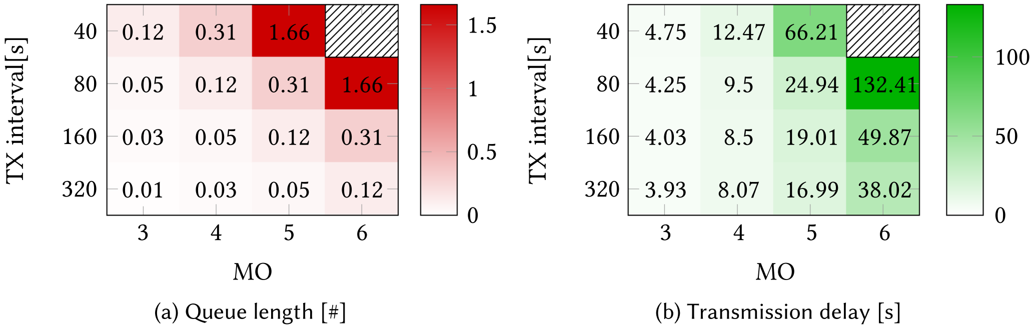
The packet reception ratio hits ≈100% in all GTS scenarios as a result of the slot allocation. The results reflect the robustness of GTS transmissions over CSMA/CA, which make it suitable for large-scale scenarios. We further analyze this in Section 8.1.

7.3 Impact of the Multisuperframe Duration

We analyze the effect of the multisuperframe duration on the average queue length and transmission delay on transmissions during CFP. Figure 23 shows the results of the analytical stochastic model (Section 6) for queue length (left) and transmission delay (right) of unconfirmed transmissions during CFP, for different multisuperframe configurations and transmission intervals. The results show that the average queue length increases with the multisuperframe order. An increment of one multisuperframe order duplicates the number of superframes per multisuperframe, ergo, the multisuperframe duration. With a fixed transmission interval, this situation increases the system utilization ($\rho = \lambda \cdot T_{msf}$), which reflects the higher queue length. The queue lengths in the diagonals are equal as a result of equal system utilization.

Observe that variations in the queue length, as a result of variations in the system utilization, are higher as the system utilization approaches 100%, as depicted in Figure 24. This reflects in the higher queue utilization in the upper right corner (Figure 23(a)). Transmissions with TX interval=40 s and multisuperframe order of 6 (T_{msf}=61.44 s) are out of the convergence region, hence, they are not shown in the figure.

Figure 23(b) reflects that the transmission delay is the product between the transmission interval and the average queue length, as seen in Section 6.3. The increase in queue length on varying multisuperframe reflects in the increased transmission delay on higher multisuperframe orders.



(a) Queue length [#]

(b) Transmission delay [s]

Fig. 23. Queue length 23(a) and transmission delay 23(b) for varying transmission intervals and multisuperframe orders. Invalid configurations are marked with hatches.

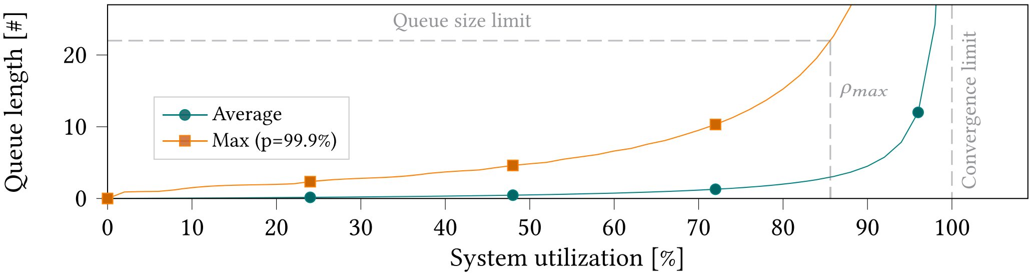


Fig. 24. Average and maximum queue length for varying system utilization cases (ρ).

Note that equal queue lengths reflect different transmission delays as a result of the longer multisuperframe duration.

We use the model to calculate the worst-case scenario of queue length for a given system utilization (Figure 24). We define the worst-case scenario as the maximum queue length with a confidence of 99.9%. The results show that the queue length in the worst-case scenario increases linearly until a system utilization of 60%, from where the queue length grows exponentially. The queue length exceeds the maximum of *openDSME* (22 frames) at ρ_{max}=85.6%. We use this value to calculate the throughput (λ_{max}) for each multisuperframe order, given that $\rho_{max} = \lambda_{max} \cdot T_{msf}$. We compare the throughput results from the model against simulation environment results (Table 7). The model shows the maximum throughput (401.24 packets/hour) with MO=3. Observe that an increment in MO halves the maximum throughput (Table 7, second column, top to bottom). This is required to maintain the maximum system utilization (85.6%), because an increment in MO duplicates the multisuperframe duration (Table 2). The model shows a deviation of less than 0.02% with respect to the simulation.

8 DESIGN DISCUSSIONS

Based on the evaluation results and the analytical stochastic model, we can now discuss optimal transmission patterns for different scenarios and tradeoffs between different superframe configurations. We also compare design options to comply with local regulations and improve the energy consumption. Finally, we discuss DSME-LoRa operation under cross-traffic.

8.1 Analysis on Data Transmission

Our evaluation reveals different properties for CSMA/CA and GTS transmissions. On the one hand, CSMA/CA transmissions in low on-air traffic show lower transmission delays than GTS

Table 7. Comparison of Throughput (λ_{max}) Using Results from the Analytical Stochastic Model and Simulation, for GTS Transmission with 85.6% System Utilization (Maximum Queue Length), for Single GTS Allocation, and Varying Multisuperframe Orders

Multisuperframe Order	**Throughput [$\frac{packets}{hour}$]**		**Error [%]**
	Model	Simulation	
3	401.24	401.25	$2.5 \cdot 10^{-3}$
4	200.58	200.62	$1.9 \cdot 10^{-2}$
5	100.29	100.31	$1.9 \cdot 10^{-2}$
6	50.15	50.16	$2.5 \cdot 10^{-3}$

transmissions and similar packet reception ratio. In high on-air traffic scenarios, CSMA/CA failures and packet collisions increase transmission delays and sharply reduce packet reception, which render CSMA/CA transmission unusable for periodic communication in large-scale networks. Confirmed transmissions improve the packet reception ratio for CSMA/CA, but increases transmission delay.

On the other hand, GTS transmissions admit ≈ 100% packet reception ratios and transmission delays for transmission intervals below the maximum system utilization, (85.6% of the multisuperframe duration; see Section 7.3). The delay of GTS transmission depends only on the MAC queue length at the moment of packet scheduling. For applications that require class-based service differentiation, priority level of DSME transmissions has potential to reduce transmission delay of high-priority messages. The network size does not affect transmission delay nor packet reception ratio for devices in a network that allocate a GTS. In contrast to CSMA/CA, unconfirmed transmissions in GTS perform similar to confirmed transmissions. Hence, we recommend to use confirmed transmission in GTS only for high-priority data.

Due to the deterministic behavior and very high packet reception ratio, GTS transmission is a better alternative than CSMA/CA transmissions for reliable large-scale unicast communication. However, CSMA/CA transmissions are still important for two reasons: (*i*) CSMA/CA supports broadcast frame transmissions. This makes CSMA/CA transmissions effective for application where a small group of devices broadcasts data to multiple receivers, such as firmware update scenarios. We leave the evaluation of broadcast transmissions for future work. (*ii*) The CAP is used for transmission of MAC commands required for association and slot allocation.

To optimize CSMA/CA transmissions, we recommend the usage of small CSMA/CA backoff exponent settings for scenarios with low on-air traffic, aiming to reduce transmission delay. However, we recommend a high backoff exponent to increase the packet reception ratio for higher on-air traffic to increase PRR.

In common LoRaWAN deployments, the addition of gateways increases PRR by exploiting the capture effect (see Section 3.2). On frame collision, a fraction of LoRaWAN gateways can still recover a frame if the power difference with the colliding frames is large enough. Due to the gateway-less nature of DSME-LoRa, it is not possible to increase PRR of unicast frames by adding more receivers. However, the capture effect can improve delivery of broadcast frames, in which a group of devices successfully decode the broadcast frame despite collision. For example, devices at a close distance to a coordinator may still successfully decode beacons under LoRa cross-traffic interference.

We show that CAD improves the performance of CSMA/CA transmissions by reducing collisions, which effectively increases PRR and reduces frame retransmissions (Section 5.3). The latter not only reduces the time on air of devices, but reduces energy consumption. We believe it is

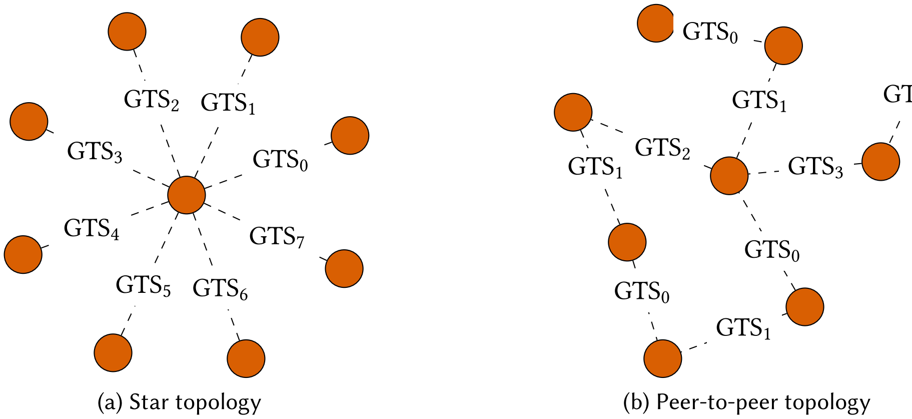


Fig. 25. GTS assignment between neighbor devices for star topology (left) and peer-to-peer topology (right).

possible to reduce collisions even further by utilizing a more sophisticated CSMA/CA mechanism such as LMAC [21].

8.2 Selection of Multisuperframe Configuration

Our evaluation shows that smaller multisuperframe orders decrease the delay of GTS transmissions. However, this reduces the GTS resources (Table 2). Although each GTS defines multiple unique frequency slots, a device can only allocate one frequency slot per GTS. This limits the number of GTS links per device to the number of GTS in the multisuperframe structure, which in turn favors cluster-tree and peer-to-peer topologies over star topologies.

Consider the two example topologies from Figure 25 with nine devices. In the star topology (Figure 25(a)) each child device allocates a transmission GTS with the coordinator. In the peer-to-peer topology, each device allocates a transmission GTS with another device of the network. The star topology requires 8 GTS, one per child device, to establish all links. Hence, the superframe structure requires at least two superframes per multisuperframe, which sets the multisuperframe duration to at least 15.36 s.

However, the peer-to-peer topology (Figure 25(b)) allows transmitting frames in the same GTS, using different channels. As a result, only 4 GTS are needed to schedule all transmissions. Therefore, a configuration with one superframe per superframe suffices, which sets the multisuperframe duration to 7.68 s. Under the same data transmission rates, the peer-to-peer topology reflects shorter transmission delays than the star topology as a result of the shorter multisuperframe duration. In contrast to the star topology, the peer-to-peer topology does not use all available GTS resources, which allows to further extend the network.

For scenarios with more than two superframes per multisuperframe, the CAP reduction mechanism offers a solution to extend the GTS resources, in which the CAP of all superframes in a multisuperframe, excluding the first, is replaced by 8 GTS. However, this reduces the CAP time of a multisuperframe, which stresses CSMA/CA transmissions and thereby challenges dynamic GTS allocation. We will analyze the impact of CAP reduction on slot allocation in future work.

8.3 Compliance with Regional Regulations

Regions with duty cycle restrictions. We show in Section 5.4 that unconfirmed data transmission to neighboring nodes does not stress the time on air resources of the network in regions with duty cycle restrictions, because transmissions do not require an intermediate forwarder (in

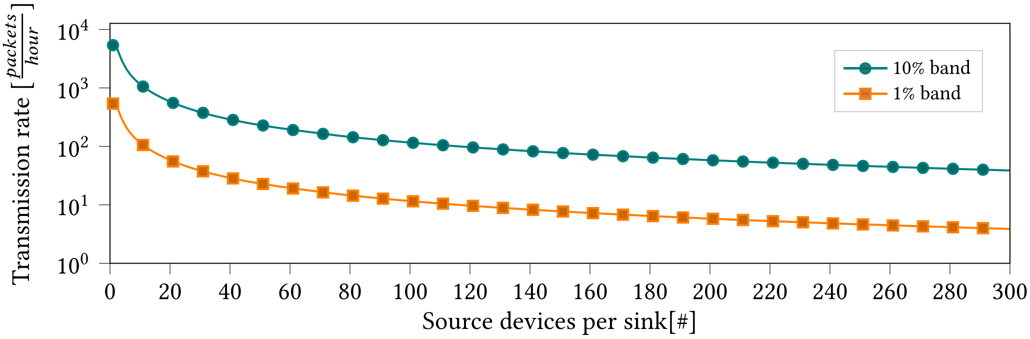


Fig. 26. Maximum transmission rate of a source device that transmits confirmed data (16 bytes payload) to a single sink (star topology) for varying number of source devices in the network and duty cycle.

contrast to LoRaWAN). This makes DSME-LoRa suitable for scenarios, in which a series of devices communicate directly with one or more sink devices (see Section 2). However, confirmed transmissions stress time on air resources of sink devices (Section 5.4). It is therefore crucial to limit the transmission interval for scenarios, in which a sink device receives packets from multiple source devices to ensure compliance with duty cycle restrictions.

We show in Figure 26 the theoretical limits of transmission rate per source device as a function of source devices that transmit to a single sink (star topology). We assume that source devices do not retransmit data (100% packet reception ratio on first transmission). For example, a star topology with 10 source devices allows transmission of ≈ 100 packets per hour in the 1% band and 1,000 packets per hour in the 10% band without exceeding the duty cycle restriction. However, a network with 115 source devices allows in the 1% and 10% bands transmission rates of 10 and 100 packets per hour, respectively.

CSMA/CA transmissions benefit from the 10% band. Nevertheless, the majority of channels in CFP belong to the 1% band, which restricts data transmissions on source devices and ACK frames on sink devices. We propose two potential solutions to overcome these limitations: (*i*) Use the group ACK feature (Section 3.1), which restricts ACK transmission to one common ACK frame per multisuperframe. Group ACKs do not contribute to better performance over regular ACK [45], but reduce the time on air utilization of sink devices by removing the dependency to the number of source devices; (*ii*) distribute the DSME-LoRa channels in more bands. For example, channels 11–25 (see Table 3) can be arranged into 12 channels in the *g* band, 2 channels in the *g1* band (868.0–868.6 MHz), and 1 channel in the *g4* band (869.7–870.0 MHz). If GTS transmissions distribute evenly among channels, then this allows for ≈20% additional transmissions per device.

Regions with dwell time or channel hopping. Dwell time requirements (e.g., in US902–928) can be easily addressed by restricting payload size, however, the channel hopping requirement (e.g., in US902–928 and CN779–787) is incompatible with single channel communication during CAP. While, on the one hand, limiting CAP transmissions in these regions is not an option, because CAP is required for slot allocation and MAC control traffic, enabling multichannel CAP, on the other hand, would require devices to listen on multiple channels (e.g., by using a LoRa concentrator). Although feasible, it increases deployment costs.

We argue that FHSS transmissions (see Section 3.2) can enable transmissions during CAP for these regions. In this regard, the channel number may dictate a unique FHSS sequence. This addresses the problem of channel hopping and dwell time, since transmissions are spread among different carrier frequencies. Thereby, the transmission time per channel is reduced. Still, it degrades

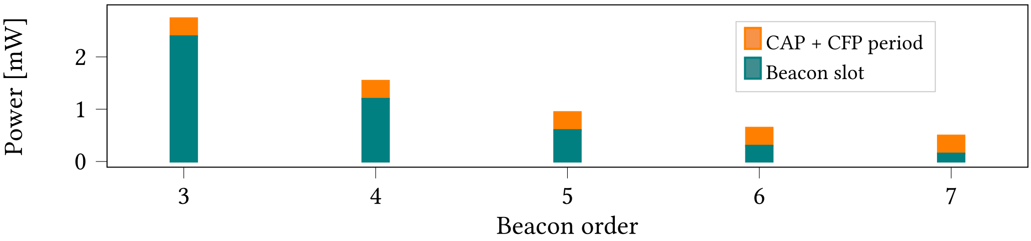


Fig. 27. Passive consumption of superframe structure with transceiver off during CAP (*macRxOnWhenIdle=0*) for varying beacon order.

CCA performance, because CAD can detect only one carrier frequency at a time. Therefore, the CCA implementation requires a different strategy. We will address this problem in future work.

8.4 Energy Considerations

On standard deployments, the passive consumption of CAP is 146.12 mJ per superframe (19.02 mW) and therefore not a good option for battery-powered devices. To overcome this problem, we proposed to turn off the CAP in battery-powered devices, as shown in Section 5.7. Although this prevents frame reception, the indirect transmission feature (Section 3.1) provides a mechanism to communicate with a device with the receiver off during CAP. On the other side, a device can still turn on the transceiver during CAP to transmit data to other devices. This allows, for example, to trigger GTS RX allocation from battery-powered device.

We show in Section 5.7 that the beacon period has a high impact in the energy consumption. A way to improve this situation is to configure a higher beacon order, which results in a higher beacon interval and therefore reduces the passive consumption of the beacon slot. We estimate the power consumption from the measurements in Section 5.7 and present the base consumption (superframe structure) for different beacon order configurations in Figure 27.

Noteworthy: The beacon order has a high impact on the energy consumption. In the scenario with BO=3, the total passive consumption hits 2.73 mW, in which the beacon period consumes ≈87%. On the contrary, in scenarios with BO=7 the power consumption is 0.49 mW, in which the beacon period consumes ≈30% of the energy. Note that BO=7 renders the beacon period to ≈122.88 s—in line with the duration of LoRaWAN class B beacon period (128 s).

Inherently, higher beacon intervals have two potential problems: The scanning procedure takes longer, which increases the energy consumption during association, and the devices synchronize to their neighbors less often, which potentially leads to desynchronization due to clock drifts. Adding more coordinators mitigates the longer association times, because the frequency of beacons increases. The use of real time clocks, available in common LoRa target platforms, mitigates desynchronization issues (Section 4.1).

To illustrate the tradeoff between transmission delay and energy consumption, we present in Table 8 the energy consumption and lifetime of a DSME-LoRa sender node for different multisuperframe orders. We assume exponentially distributed interarrival times with TX interval=15 m. The beacon interval is set to 122.28 s (BO=7)—in line with LoRaWAN class B beacon interval— and assume the device keeps the transceiver on for two beacon intervals to associate with a single coordinator. We also estimate the voltage regulator efficiency to be 90%. For the lifetime estimation, we assume the device operates on a battery of 2,800 mAh capacity—in line with common off-the-shelf AA alkaline batteries. We utilize the model (Section 6) to estimate transmission delay.

MO=7 renders the lowest power consumption (0.38 mW) and allows ≈3 years of operation. However, it also depicts the highest average latency (≈71 s). Note that the delay can increase up to the

Table 8. Comparison of Average Transmission Delay, Power, and Lifetime for a DSME-LoRa Sender Device with TX Interval = 15 m and BO = 7, for Varying Multisuperframe Order

Multisuperframe Order	Delay [s]	Power [mW]	Lifetime [y]
3	3.87	0.58	1.82
4	7.81	0.47	2.24
5	15.9	0.42	2.53
6	32.97	0.39	2.71
7	71.16	0.38	2.81

beacon duration (122.28 s) if the packet is scheduled right after the boundary of the GTS. If the use case does not tolerate high delays, then the device may opt for a lower multisuperframe order. Please observe that (*i*) the energy consumption decreases with an increase of multisuperframe order, and (*ii*) the energy consumption decreases at a lower rate on higher multisuperframe order. For example, a reduction from MO=7 to MO=6 increases consumption only by 0.01 mW, while a reduction from MO=4 to MO=3 increases 0.11 mW. This effect occurs because a decrement in MO duplicates the number of GTS per beacon interval. Recall that a scheduled GTS-TX consumes 2.2 mJ even if there is no transmission (see Section 5.7), which reflects the increase in energy consumption. A device with MO=3 renders an average delay of 3.87 s and lifetime of ≈2 years. If transmission delay is not critical, then a device can extend the lifetime up to 1 year by setting MO=7. Although MO beyond 7 is possible, it increases the beacon interval, which challenges device synchronization. In general, the energy footprint of DSME-LoRA for uplink-oriented applications is higher than LoRaWAN, considering an equivalent LoRaWAN class A device can operate >10 years running on batteries.

Finally, we present potential optimizations for *openDSME* and the integration into RIOT, aiming to reduce power consumption: (*i*) Avoid turning on the transceiver on a TX GTS if the MAC queue is empty, which reduces the energy consumption by ≈1.5 mJ per superframe (≈0.20 mW) for the allocation of one GTS TX slot. This is possible with a minor change in the GTS management routines of *openDSME*. (*ii*) Turn off the transceiver in the beacon slot right after the reception of the beacon, which potentially reduces ≈16 mJ in the receiving beacon slot. (*iii*) Use CAD to detect the preamble of a LoRa frame at the beginning of a GTS RX and proceed to reception if CAD succeeds, instead of keeping the transceiver idle listening for the duration of the slot (0.48 s). For example. three CAD attempts consume 0.73 mJ, as analyzed in Section 5.7. Under this scenario, the passive consumption of the CFP reduces by 18.58 mJ per superframe (2.41 mW).

8.5 Analysis of Cross-traffic

We show in Section 5.5 that DSME-LoRa can coexist with LoRaWAN in the EU868 region, because LoRaWAN traffic neither interferes with DSME MAC control traffic nor with beacons. Therefore, concurrent DSME-LoRa and LoRaWAN communication only degrades PRR. As long as the common channel does not overlap with LoRaWAN channels, the coexistence of DSME-LoRa and LoRaWAN in regions other than EU868 is feasible.

The channel hopping mode in GTS allows communication despite heavy interference on a single channel but degrades PRR as a result of a fraction of packets being transmitted in the noisy channel. To overcome this problem, a device may transmit confirmed messages. Thereby, the MAC will perform the retransmission on a channel with better quality. An alternative solution is to use the channel adaptation mode of DSME-LoRa (see Section 3.1), in which the source and target device agree on a different channel if the channel quality is poor. Although *openDSME* implements the channel adaptation mode, it does not implement the required MAC command (DSME Link Report) to request channel quality information. Therefore, there is no way to infer channel quality and agree on a different channel.

Poor channel quality in the common channel challenges device synchronization (see Section 5.6), which prevents normal operation of the DSME-LoRa network. While this is also a problem for standard DSME, the long time on air and long range of LoRa packets represents a security thread for DSME-LoRa networks. Attackers may desynchronize devices by jamming the channel during beacon transmissions.

To address this problem, coordinators may request children devices to switch to a channel with better quality using the PHY-OP-SWITCH mechanism (see Section 3.1). This requires the coordinator device to estimate the channel quality, for example, by keeping track of the failed CCA attempts during CSMA/CA transmissions. However, the MAC control frames required by the PHY-OP-SWITCH mechanism are sent during CAP, which challenges packet delivery under noisy conditions. Also, a device may detect good channel quality during CAP even if an attacker jams only beacon frames. An alternative solution is to transmit frames using the FHSS, as analyzed in Section 8.3. Thereby, packet transmissions can tolerate noise in a single channel by relying on forward error correction mechanisms on the LoRa PHY. To prevent selective jam attacks, packets can be transmitted with a pseudorandom FHSS sequence shared by all devices. We will analyze this proposal in future work.

9 RELATED WORK

9.1 IEEE 802.15.4 Standards TSCH and DSME

The 802.15.4 MAC modes TSCH and DSME [26] have been analyzed [34], modeled [10, 30], and simulated [6, 16, 29]. The results indicate that TSCH obtains lower latency and higher throughput for small networks (<30 nodes). DSME outperforms TSCH for higher duty cycles and an increasing number of nodes. Kauer et al. [31] introduce *openDSME*, an implementation that is available for *OMNeT++* and as a portable C++ library. We utilize *openDSME* in our work. The authors compare simulated performances to real-world measurements—which are on par—and further investigate group ACKs [45] that do not contribute to better performance over direct ACKs. Vallati et al. [62] find inefficiencies in DSME network formation and provide countermeasures, however, we move network formation to future work. Improvements on the QoS of DSME networks were proposed by Kurunathan [33]. Similar to the IETF standard solution IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) [64] for IPv6 over TSCH, Kurunathan et al. present **RPL (Routing Protocol for Low power and Lossy Networks)** over DSME [35]. The IETF further provides an applicability statement [9] for RPL in metering use cases and proposes DSME as a MAC. The IEEE 802.15 working group, in contrast, introduces “Low-Energy Critical Infrastructure Monitoring” in the *w*-amendment [27], which adds long-range radios that operate in the sub-GHz band. These networks are primarily defined to operate in star topologies, which supports up our topology choice of a single-hop DSME network.

9.2 Analysis of LoRaWAN

Existing LoRaWAN [43] networks are susceptible to collisions [18, 49] as well as energy depletion [46]. Liando et al. [40] provide real-world measurements of LoRa and LoRaWAN and explore the impact of transmission parameters of the chirp spread spectrum modulation. They thereby identify optimization potentials for the medium access layer. Slabicki et al. [60] contribute *FLoRa*, a LoRa simulator for *OMNeT++*, and improve the **adaptive data rate (ADR)** mechanism of LoRaWAN. We utilize *FLoRa* in our simulations. Rizzi et al. [53] and Leonardi et al. [39] show that slight modifications of the LoRaWAN MAC already improve performance metrics of class A deployments, which are centered around the concept of uplink packets from an end node. LoRaWAN, however, poses a severe challenge on downlink traffic due to band limitations [17, 55] in the sub-GHz band and contention with unpredictable uplink packets [47]. Vincenzo et al. [65] propose countermeasures to that problem by adding multiple gateways and a gateway selection mechanism. This decreases losses but adds deployment cost.

LoRaWAN class B (see Section 1), though barely deployed, provides periodic downlink slots (unlike class A&C) and multicast capabilities [44] through these slots. Elbsnir et al. [15] confirm that class B decreases downlink latency and loss over class A. Ron et al. [54] derive an optimal class B configuration to trade waiting time with energy consumption, and Pasetti et al. [50] design a single-gateway class B LoRaWAN network for 312 LoRa nodes. Unfortunately, a practical evaluation is missing. Operating in class B, however, suffers from scalability issues [19, 59]. Despite, class B still burdens the gateway duty cycle and requires an infrastructure network, hence, it is not an option for long-range node-to-node communication.

9.3 New Protocols for LoRa

The IETF standardized a compression [22] scheme for LoRa networks. Similarly, Perešíni et al. [51] introduce a slim packet format with a new LoRa link layer to reduce effective payload. Gonzalez et al. [23] motivate the development of a new LoRa MAC and present LoRa PHY configurations to define logical channels, which assists frequency- and time division multiple access protocols. We apply these considerations in our work.

Cotrim et al. [12] provide a classification for multi-hop LoRaWAN networks. Enabling multi-hop with long-range radios is a common desire [1, 8, 56, 61]. New designs of time-slotted LoRa protocols [71] have been analyzed with simulations [38, 69] and practical deployments [70]. Haubro et al. [25] present an adaptation of the 802.15.4 TSCH mode [26] for LoRa. Their real-world measurements show the applicability of 802.15.4 MAC layers for long-range communication, however, the experiment deployment consists of only three nodes and limited traffic. The analysis of duty-cycle compliance remains open. In contrast, we focus on LoRa and the 802.15.4 DSME mode in our work and aim to fill in the gap of a large-scale deployment that further includes duty-cycle analyses.

Several MAC and PHY approaches have been analyzed to overcome the problem of concurrent LoRa communication. Xu et al. [68] propose S-MAC, an adaptive scheduling mechanism for Low Power Wide Area Network (LPWAN) that exploits the fact many LPWAN applications transmit period uplink data. Devices with the same spreading factor and known transmission interval are grouped and assigned a unique carrier frequency to minimize intergroup frame collisions. The approach brings a 4× throughput improvement for periodic uplink communication, but does not address downlink limitations of LoRaWAN (see Section 2). The authors of References [48, 52] present experimental results for contention-based media access with LoRa. Kennedy et al. [48] explore CSMA/CA with CAD. Results show that listen before talk performs better than ALOHA in dense deployments, which motivated our efforts of using CSMA/CA with CAD in the contention access period of a DSME frame (see Section 5.3). Gamage et al. [21] propose LMAC, an improved CSMA/CA protocol, and evaluate on a testbed the design of three advancing versions of the protocol. Results indicate that the approach brings 2.2× goodput improvement and 2.4× reduction of energy consumption. We motivate the LMAC approach for future work to reduce collisions during CAP transmissions (see Section 5.3). There have been multiple proposals to resolve LoRa frames collisions at the physical layer [58, 66, 67]. Evaluations of those mechanisms on software-defined radio show a clear improvement of throughput and overall network capacity, but add hardware complexity and extra deployment cost in comparison to common off-the-shelf LoRa nodes.

Little work analyzes alternative communication pattern over LoRa. Lee et al. [37] propose gateway-driven requests. This approach follows a request-response pattern and indicates performance benefits over producer-driven ALOHA. Similarly, the authors of References [13, 14, 42] deploy information-centric networking over LoRa radios, which is a data request-driven protocol. Their work showed, however, the need for a proper LoRa media access layer.

10 CONCLUSIONS AND OUTLOOK

In this work, we exposed the problems of LoRaWAN for node-to-node communication and motivated the usage of IEEE 802.15.4 DSME over LoRa, which opens LoRa to general networking. We summarized the DSME mappings for the EU868 region with system integration into the operating system RIOT and presented a comprehensive evaluation of DSME-LoRa on an IoT testbed. The results revealed that CSMA/CA transmissions during the contention access period provide a good tradeoff between transmission delay and packet reception ratio for networks with low traffic and a few nodes. On the contrary, GTS transmissions show about 100% packet reception ratio and predictable transmission delays for networks with higher network size and higher traffic. We could show that under the limits of available GTS resources, these performance metrics do not degrade with the network size. The results also confirmed that coexistence between LoRaWAN and DSME-LoRa is possible. Nevertheless, noise in the common channel affects normal operation of the network due to beacon loss.

Our findings confirmed that the Channel Activity Detection feature of LoRa radios is a powerful clear channel detection mechanism for CSMA/CA and effectively reduces the number of retransmissions ≈15 times in scenarios with moderate traffic. We evaluated the effect of CSMA/CA backoff exponent settings and could show that higher values mitigate frame collisions during CAP. The evaluation evidenced that direct communication between devices facilitates compliance with regional duty cycle regulations. We also confirmed that with optimal MAC configurations, DSME-LoRa offers a passive consumption of less than 1 mW. Based on a novel analytical stochastic model, we calculated average queue length in the MAC for slotted transmission, from which we estimated the transmission delays. Validation of the model with data from the experiments using real IoT hardware showed an accuracy of 99.99%. We also evaluated DSME-LoRa for larger network sizes using a well-known simulation environment and confirmed our experimental findings. We evaluated the effect of the MAC configuration and utilized the model to optimize throughput for each configuration. From the evaluation results, we built an overview of transmission patterns and configurations aiming to provide a good tradeoff between transmission delay, time on air, and energy consumption, which led to proposing changes in the MAC implementation for improving energy consumption.

There are three future directions of this research. First, recent IETF concepts of the 6TiSCH and IPv6 over LPWAN working groups should be adopted while taking advantage of built-in features of DSME to enable IPv6 over DSME-LoRa. Second, studying dynamic slot allocation between DSME-LoRa nodes can foster deployment experience for real-world scenarios. Third, the study of suitable network layers on top of DSME-LoRa and its performance under massive industrial deployment [24] shall open a new direction of LoRa-centric research.

Availability of software and reproducibility. We strongly support reproducible research [2, 57] and utilize open source software and open testbed platforms. All of our work is intended for public release. The code of the software components (implementation of DSME-LoRa on RIOT, simulation environment), the implementation of the analytical stochastic model, documentation, datasets, and related tools are available on GitHub at https://github.com/inetrg/tosn-dsmelora22.

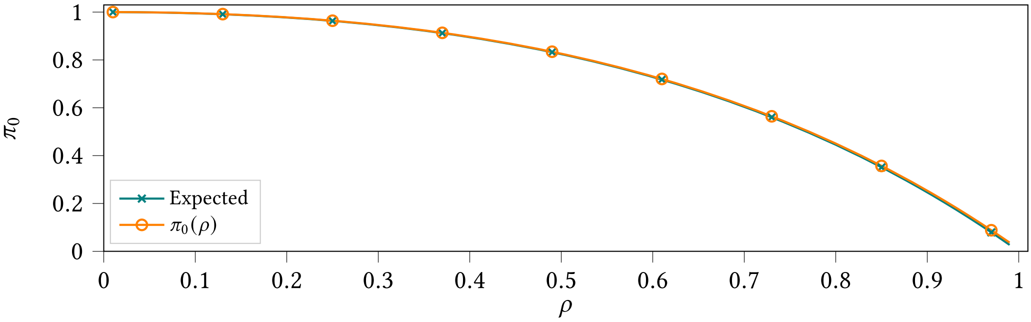


Fig. 28. Comparison between the expected and the polynomial approximation of the distribution π_0 for varying system loads ρ.

APPENDICES

A NUMERICAL CALCULATION OF STATIONARY MARKOV DISTRIBUTION

In this Appendix, we want to explain how to evaluate the station Markov distribution. For this, let us define $\vec{x}$ as any eigenvector of P to the value of 1. Note that:

$$\vec{\pi} = \frac{\vec{x}}{\sum_{i=0}^{\infty} x_i}. \tag{9}$$

The equation system that describes the eigenvectors is:

$$\begin{aligned}(k_0 + k_1)x_0 + k_0 x_1 &= x_0\\ k_2 x_0 + k_1 x_1 + k_0 x_2 &= x_1\\ k_3 x_0 + k_2 x_1 + k_1 x_2 + k_0 x_3 &= x_2\\ &\cdots\\ k_0 x_{n+1} + \sum_{i=0}^{n} k_{n+1-i} x_i &= x_n.\end{aligned}$$

Which resolves to:

$$\begin{aligned}x_1 &= x_0 \frac{(1 - k_0 - k_1)}{k_0}\\ x_i &= x_0 \frac{(1 - k_1)x_{i-1} - \sum_{j=0}^{i-2} k_{i-j} x_j}{k_0}, \forall i \in [2, \infty].\end{aligned}$$

We set $x_0 = 1$ and calculate $x_1, x_2, x_3 \ldots x_N$ with N a big enough number. We then obtain $\vec{\pi}$ using Equation (9). This method is not practical, because it requires the calculation of all vector members. Therefore, we propose to approximate π_0 with a polynomial function. The proposed polynomial function converges to the expected values of π_0, as depicted in Figure 28.

We calculate π_0 using the former method for different ρ and $N = 500$. We then fit $f(x) = ax^4 + bx^3 + cx^2 + dx + e$ accordingly. The result of the fit procedure produces the polynomial function:

$$\pi_0(\rho) = -0.24\rho^4 - 0.21\rho^3 - 0.55\rho^2 + 0.01\rho^1 + 1.$$

B GLOSSARY

6TiSCH IPv6 over the TSCH mode of IEEE 802.15.4e.

BO Beacon Order.
BS Beacon Slot.

CAD Channel Activity Detection.
CAP Contention Access Period.
CCA Clear Channel Assessment.
CFP Contention Free Period.
CRC Cyclic Redundancy Check.
CSMA/CA Carrier Sense Multiple Access/Collision Avoidance.

DER Data Extraction Rate.
DSME Deterministic Synchronous Multichannel Extension.

FHSS Frequency-hopping spread spectrum.

GTS Guaranteed Time Slot.

LPWAN Low Power Wide Area Network.

MAC Media Access Control.
MO Multisuperframe Order.

PAN Personal Area Network.
PHY Physical Layer.
PRR Packet Reception Ratio.
PSDU Physical Service Data Unit.

RSSI Received Signal Strength Indicator.

SO Superframe Order.

TSCH Time Slotted Channel Hopping.

REFERENCES

[1] Andrea Abrardo and Alessandro Pozzebon. 2019. A multi-hop LoRa linear sensor network for the monitoring of underground environments: The case of the medieval aqueducts in Siena, Italy. *Sensors* 19, 2 (2019), 402.

[2] ACM. 2017. Result and Artifact Review and Badging. Retrieved from http://acm.org/publications/policies/artifact-review-badging.

[3] Cedric Adjih, Emmanuel Baccelli, Eric Fleury, Gaetan Harter, Nathalie Mitton, Thomas Noel, Roger Pissard-Gibollet, Frederic Saint-Marcel, Guillaume Schreiner, Julien Vandaele, and Thomas Watteyne. 2015. FIT IoT-LAB: A large scale open experimental IoT testbed. In *IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE Press, Piscataway, NJ, 459–464.

[4] Jose Alamos, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2021. DSME-LoRa—A flexible MAC for LoRa. In *29th IEEE International Conference on Network Protocols (ICNP'21)*. IEEE, Piscataway, NJ. DOI: https://doi.org/10.1109/ICNP52444.2021.9651945

[5] Jose Alamos, Peter Kietzmann, Thomas C. Schmidt, and Matthias Wählisch. 2022. WIP: Exploring DSME MAC for LoRa—A system integration and first evaluation. In *23rd IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, Piscataway, NJ.

[6] Giuliana Alderisi, Gaetano Patti, Orazio Mirabella, and Lucia Lo Bello. 2015. Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic process automation scenarios. In *13th International Conference on Industrial Informatics (INDIN'15)*. IEEE, Piscataway, NJ, 948–955.

[7] Emmanuel Baccelli, Cenk Gündogan, Oliver Hahm, Peter Kietzmann, Martine Lenders, Hauke Petersen, Kaspar Schleiser, Thomas C. Schmidt, and Matthias Wählisch. 2018. RIOT: An open source operating system for low-end embedded devices in the IoT. *IEEE Internet Things J.* 5, 6 (Dec. 2018), 4428–4440. Retrieved from http://dx.doi.org/10.1109/JIOT.2018.2815038.

[8] Maite Bezunartea, Roald Van Glabbeek, An Braeken, Jacques Tiberghien, and Kris Steenhaut. 2019. Towards energy efficient LoRa multihop networks. In *International Symposium on Local and Metropolitan Area Networks (LANMAN'19)*. IEEE, Piscataway, NJ, 1–3.

[9] N. Cam-Winget, J. Hui, and D. Popa. 2017. *Applicability Statement for the Routing Protocol for Low-power and Lossy Networks (RPL) in Advanced Metering Infrastructure (AMI) Networks*. RFC 8036. IETF.

[10] Nikumani Choudhury, Rakesh Matam, Mithun Mukherjee, and Jaime Lloret. 2020. A performance-to-cost analysis of IEEE 802.15.4 MAC With 802.15.4e MAC modes. *IEEE Access* 8 (2020), 41936–41950.

[11] TTN Community. 2022. The Things Network. Retrieved from https://www.thethingsnetwork.org/.

[12] Jeferson Rodrigues Cotrim and João Henrique Kleinschmidt. 2020. LoRaWAN mesh networks: A review and classification of multihop communication. *Sensors* 20, 15 (2020), 4273.

[13] Anthony Dowling, Lauren Huie, Laurent Njilla, Hong Zhao, and Yaoqing Liu. 2021. Toward long-range adaptive communication via information centric networking. *Intell. Converged Netw.* 2, 1 (2021), 1–15.

[14] Anthony Dowling, Yaoqing Liu, Lauren Huie, and Kang Chen. 2021. Building an information-centric and LoRa-based sensing platform for IoT. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE Press, Piscataway, NJ, 1–6.

[15] Houssem Eddin Elbsir, Mohammed Kassab, Sami Bhiri, and Mohamed Hedi Bedoui. 2020. Evaluation of LoRaWAN class B efficiency for downlink traffic. In *16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'20)*. IEEE, Piscataway, NJ, 105–110.

[16] Atis Elsts. 2020. TSCH-Sim: Scaling up simulations of TSCH and 6TiSCH networks. *Sensors* 20, 19 (2020), 5663.

[17] European Telecommunications Standards Institute. 2006. *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods*. Technical Report ETSI EN 300 220-1 V2.1.1. IEEE, Sophia Antipolis, France.

[18] Guillaume Ferre. 2017. Collision and packet loss analysis in a LoRaWAN network. In *25th European Signal Processing Conference (EUSIPCO'17)*. IEEE, Piscataway, NJ, 2586–2590.

[19] Joseph Finnegan, Stephen Brown, and Ronan Farrell. 2018. Evaluating the scalability of LoRaWAN gateways for class B communication in ns-3. In *IEEE Conference on Standards for Communications and Networking (CSCN'18)*. IEEE, Piscataway, NJ, 1–6.

[20] F. Sanchez-Sutil and A. Cano-Ortega. 2022. Smart regulation and efficiency energy system for street lighting with LoRa LPWAN. *Sustain. Cities Societ.* 83 (2022).

[21] Amalinda Gamage, Jansen Christian Liando, Chaojie Gu, Rui Tan, and Mo Li. 2020. *LMAC: Efficient Carrier-sense Multiple Access for LoRa*. Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/3372224.3419200

[22] O. Gimenez and I. Petrov. 2021. *Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN*. RFC 9011. IETF.

[23] Nicolas Gonzalez, Adrien Van Den Bossche, and Thierry Val. 2018. Specificities of the LoRa physical layer for the development of new ad hoc MAC layers. In *17th International Conference on Ad Hoc Networks and Wireless (AdHoc-Now'18)*. Springer, Cham, Switzerland, 163–174.

[24] Cenk Gündogan, Peter Kietzmann, Martine S. Lenders, Hauke Petersen, Michael Frey, Thomas C. Schmidt, Felix Shzu-Juraschek, and Matthias Wählisch. 2021. The impact of networking protocols on massive M2M communication in the industrial IoT. *IEEE Trans. Netw. Serv. Manag.* 18, 4 (Dec. 2021), 4814–4828. DOI: https://doi.org/10.1109/TNSM.2021.3089549

[25] Martin Haubro, Charalampos Orfanidis, George Oikonomou, and Xenofon Fafoutis. 2020. TSCH-over-LoRA: Long range and reliable IPv6 multi-hop networks for the internet of things. *Internet Technol. Lett.* 3, 4 (2020), e165.

[26] IEEE 802.15 Working Group. 2016. *IEEE Standard for Low-Rate Wireless Networks*. Technical Report IEEE Std 802.15.4™–2015 (Revision of IEEE Std 802.15.4-2011). IEEE, New York, NY.

[27] IEEE 802.15 Working Group. 2020. *IEEE Standard for Low-Rate Wireless Networks–Amendment 2: Low Power Wide Area Network (LPWAN) Extension to the Low-Energy Critical Infrastructure Monitoring (LECIM) Physical Layer (PHY)*. Technical Report IEEE Std 802.15.4™–2020w (Amendment to IEEE Std 802.15.4-2020). IEEE, New York, NY.

[28] INET Authors. 2021. INET Framework - An open-source OMNeT++ model suite for wired, wireless and mobile networks. Retrieved from https://inet.omnetpp.org/.

[29] Wun-Cheol Jeong and Junhee Lee. 2012. Performance evaluation of IEEE 802.15.4e DSME MAC protocol for wireless sensor networks. In *1st IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT'12)*. IEEE, Piscataway, NJ, 7–12.

[30] Iacob Juc, Olivier Alphand, Roberto Guizzetti, Michel Favre, and Andrze Dudaj. 2016. Energy consumption and performance of IEEE 802.15.4e TSCH and DSME. In *the IEEE Wireless Communications and Networking Conference (WCNC'16)*. IEEE, Piscataway, NJ.

[31] Florian Kauer, Maximilian Köstler, and Volker Turau. 2018. *Reliable Wireless Multi-hop Networks with Decentralized Slot Management: An Analysis of IEEE 802.15.4 DSME*. Technical Report arXiv:1806.10521. Open Archive: arXiv.org.

[32] Peter Kietzmann, Jose Alamos, Dirk Kutscher, Thomas C. Schmidt, and Matthias Wählisch. 2022. Long-range ICN for the IoT: Exploring a LoRa system design. In *21st IFIP Networking Conference*. IEEE Press, Piscataway, NJ.

[33] Harrison Kurunathan. 2021. *Improving QoS for IEEE 802.15.4e DSME Networks*. Doctoral Dissertation. Faculty of Engineering, University of Porto. Retrieved from https://hdl.handle.net/10216/132005.

[34] Harrison Kurunathan, Ricardo Severino, Anis Koubaa, and Eduardo Tovar. 2018. IEEE 802.15.4e in a Nutshell: Survey and performance evaluation. *IEEE Commun. Surv. Tutor.* 20, 3 (2018), 1989–2010.

[35] Harrison Kurunathan, Ricardo Severino, Anis Koubaa, and Eduardo Tovar. 2020. Symphony: Routing aware scheduling for DSME networks. *SIGBED Rev.* 16, 4 (Jan. 2020), 26–31.

[36] Yandja Lalle, Mohamed Fourati, Lamia Chaari Fourati, and João Paulo Barraca. 2021. Routing strategies for LoRaWAN multi-hop networks: A survey and an SDN-based solution for smart water grid. *IEEE Access* 9 (2021), 168624–168647.

[37] Huang-Chen Lee and Kai-Hsiang Ke. 2018. Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation. *IEEE Trans. Instrum. Meas.* 67, 9 (2018), 2177–2187.

[38] Luca Leonardi, Filippo Battaglia, Gaetano Patti, and Lucia Lo Bello. 2018. Industrial LoRa: A novel medium access strategy for LoRa in Industry 4.0 applications. In *44th Annual Conference of the IEEE Industrial Electronics Society (IECON'18)*. IEEE Press, Piscataway, NJ, 4141–4146.

[39] Luca Leonardi, Lucia Lo Bello, Filippo Battaglia, and Gaetano Patti. 2020. Comparative assessment of the LoRaWAN medium access control protocols for IoT: Does listen before talk perform better than ALOHA? *Electronics* 9, 4 (2020), 553.

[40] Jansen C. Liando, Amalinda Gamage, Agustinus W. Tengourtius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *Trans. Sensor Netw.* 15, 2 (Feb. 2019), 16. DOI: https://doi.org/10.1145/3293534

[41] John D. C. Little. 1961. A proof for the queuing formula: $L = \lambda W$. *Oper. Res.* 9, 3 (1961), 383–387.

[42] Yaoqing Liu, Laurent Njilla, Anthony Dowling, and Wan Du. 2020. Empowering named data networks for ad-hoc long-range communication. In *Wireless and Optical Communications Conference (WOCC'20)*. IEEE, Piscataway, NJ, 1–6.

[43] LoRa Alliance – Technical Committee. 2017. *LoRaWAN 1.1 Specification*. Technical Report. LoRa Alliance. Retrieved from https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf.

[44] LoRa Alliance – Technical Committee. 2018. *LoRaWAN Remote Multicast Setup Specification v1.0.0*. Technical Report. LoRa Alliance. Retrieved from https://lora-alliance.org/sites/default/files/2018-09/remote_multicast_setup_v1.0.0.pdf.

[45] Florian Meyer, Phil Malessa, Jan Niklas Diercks, and Volker Turau. 2022. Are group acknowledgements worth anything in IEEE 802.15.4 DSME: A comparative analysis. In *5th Conference on Cloud and Internet of Things, CIoT 2022*. IEEE, 114–121.

[46] Konstantin Mikhaylov, Radek Fujdiak, Ari Pouttu, Voznak Miroslav, Lukas Malina, and Petr Mlynek. 2019. Energy attack in LoRaWAN: Experimental validation. In *14th International Conference on Availability, Reliability and Security (ARES'19)*. ACM, New York, NY, 1–6.

[47] Konstantin Mikhaylov, Juha Petäjäjärvi, and Ari Pouttu. 2018. Effect of downlink traffic on performance of LoRaWAN LPWA networks: Empirical study. In *29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'18)*. IEEE, Piscataway, NJ.

[48] Morgan O'Kennedy, Thomas Niesler, Riaan Wolhuter, and Nathalie Mitton. 2020. Practical evaluation of carrier sensing for a LoRa wildlife monitoring network. In *19th IFIP Networking Conference*. IEEE Press, Piscataway, NJ, 10–18.

[49] Charalampos Orfanidis, Laura Marie Feeney, Martin Jacobsson, and Per Gunningberg. 2019. Cross-technology clear channel assessment for low-power wide area networks. In *16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS'19)*. IEEE Computer Society, Washington, DC, 199–207.

[50] Marco Pasetti, Emiliano Sisinni, Paolo Ferrari, Stefano Rinaldi, Alessandro Depari, Paolo Bellagente, Davide Della Giustina, and Alessandra Flammini. 2020. Evaluation of the use of class B LoRaWAN for the coordination of distributed interface protection systems in smart grids. *J. Sensor Actuat. Netw.* 9, 1 (2020), 13.

[51] Ondrej Perešíni and Tbor Krajčovič. 2017. More efficient IoT communication through LoRa network with LoRa@FIIT and STIOT protocols. In *11th International Conference on Application of Information and Communication Technologies (AICT '17)*. IEEE, Piscataway, NJ, 1–6.

[52] Conduc Pham. 2018. Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks. In *Wireless Communications and Networking Conference (WCNC'18)*. IEEE, Piscataway, NJ, 1–6.

[53] Mattia Rizzi, Paol Ferrari, Alessandra Flammini, Emiliano Sisinni, and Mikael Gidlund. 2017. Using LoRa for industrial wireless networks. In *13th International Workshop on Factory Communication Systems (WFCS'17)*. IEEE Press, Piscataway, NJ, 1–4.

[54] Dara Ron, Chan-Jae Lee, Kisong Lee, Hyun-Ho Choi, and Jung-Ryun Lee. 2020. Performance analysis and optimization of downlink transmission in LoRaWAN class B mode. *IEEE Internet Things J.* 7, 8 (2020), 7836–7847.

[55] Martijn Saelens, Jeroen Hoebeke, Adnan Shahid, and Eli De Poorter. 2019. Impact of EU duty cycle and transmission power limitations for sub-GHz LPWAN SRDs: An overview and future challenges. *EURASIP J. Wirel. Commun. Netw.* 219 (2019), 219–251.

[56] Benjamin Sartori, Steffen Thielemans, Maite Bezunartea, An Braeken, and Kris Steenhaut. 2017. Enabling RPL multihop communications based on LoRa. In *13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'17)*. IEEE Computer Society, Washington, DC, 1–8.

[57] Quirin Scheitle, Matthias Wählisch, Oliver Gasser, Thomas C. Schmidt, and Georg Carle. 2017. Towards an ecosystem for reproducible research in computer networking. In *ACM SIGCOMM Reproducibility Workshop*. ACM, New York, NY, 5–8.

[58] Muhammad Osama Shahid, Millan Philipose, Krishna Chintalapudi, Suman Banerjee, and Bhuvana Krishnaswamy. 2021. Concurrent interference cancellation: Decoding multi-packet collisions in LoRa. In *ACM SIGCOMM Conference (SIGCOMM'21)*. Association for Computing Machinery, New York, NY, 503–515. DOI: https://doi.org/10.1145/3452296.3472931

[59] Yonatan Shiferaw, Apoorva Arora, and Fernando Kuipers. 2020. LoRaWAN class B multicast scalability. In *19th IFIP Networking Conference*. IEEE Press, Piscataway, NJ, 609–613.

[60] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. 2018. Adaptive configuration of LoRa networks for dense IoT deployments. In *IEEE/IFIP Network Operations and Management Symposium (NOMS'18)*. IEEE Press, Piscataway, NJ, 1–9.

[61] Steffen Thielemans, Maite Bezunartea, and Kris Steenhaut. 2017. Establishing transparent IPv6 communication on LoRa based low power wide area networks (LPWANS). In *Wireless Telecommunications Symposium (WTS'17)*. IEEE, Piscataway, NJ, 1–6.

[62] Carlo Vallati, Simone Brienza, Maurizio Palmieri, and Giuseppe Anastasi. 2017. Improving network formation in IEEE 802.15.4e DSME. *Comput. Commun.* 114, C (Dec. 2017).

[63] András Varga. 2003. The OMNeT++ Discrete Event Simulation System. Retrieved from https://omnetpp.org/.

[64] X. Vilajosana, K. Pister, and T. Watteyne. 2017. *Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration*. RFC 8180. IETF.

[65] Valentina Di Vincenzo, Martin Heusse, and Bernard Tourancheau. 2019. Improving downlink scalability in LoRaWAN. In *IEEE International Conference on Communications (ICC'19)*. IEEE, Piscataway, NJ.

[66] Xianjin Xia, Ningning Hou, Yuanqing Zheng, and Tao Gu. 2021. PCube: Scaling LoRa concurrent transmissions with reception diversities. In *27th Annual International Conference on Mobile Computing and Networking (MobiCom'21)*. Association for Computing Machinery, New York, NY, 670–683. DOI: https://doi.org/10.1145/3447993.3483268

[67] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: Parallel decoding for LoRa transmissions. In *17th Conference on Embedded Networked Sensor Systems (SenSys'19)*. Association for Computing Machinery, New York, NY, 192–204. DOI: https://doi.org/10.1145/3356250.3360024

[68] Zhuqing Xu, Junzhou Luo, Zhimeng Yin, Tian He, and Fang Dong. 2020. S-MAC: Achieving high scalability via adaptive scheduling in LPWAN. In *IEEE Conference on Computer Communications*. 506–515. DOI: https://doi.org/10.1109/INFOCOM41043.2020.9155474

[69] Gokcer Yapar, Tuna Tugcu, and Orhan Ermis. 2019. Time-slotted ALOHA-based LoRaWAN scheduling with aggregated acknowledgement approach. In *25th Conference of Open Innovations Association (FRUCT'19)*. IEEE, Piscataway, NJ, 383–390.

[70] Dimitrios Zorbas, Khaled Abdelfadeel, Panayiotis Kotzanikolaou, and Dirk Pesch. 2020. TS-LoRa: Time-slotted LoRaWAN for the industrial Internet of Things. *Comput. Commun.* 153 (2020), 1–10.

[71] Dimitrios Zorbas and Xenofon Fafoutis. 2021. Time-slotted LoRa networks: Design considerations, implementations, and perspectives. *IEEE Internet Things Mag.* 4, 1 (3 2021), 84–89.

Received 18 March 2022; revised 25 June 2022; accepted 8 July 2022