Increasing Throughput and Efficiency of LoRaWAN Class A

Roman Trüb and Lothar Thiele

Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
Email: {roman.trueb, lothar.thiele}@tik.ee.ethz.ch

***Abstract*—The number of Internet of Things (IoT) devices is expected to increase significantly in the next few years due to the availability of low cost IoT hardware and new application scenarios. As a result, many more wireless IoT devices will share the unlicensed frequency bands. Coordinated channel access is required to increase the efficiency of the frequency spectrum usage. In this paper, we propose two extensions to Long Range Wide Area Network (LoRaWAN) Class A, the *TDMA* and *Burst* scheme, in order to increase the channel utilization and system throughput. Our calculations show that the proposed schemes can provide more than 60% throughput compared to 18% provided by the pure ALOHA scheme used in the current specifications of LoRaWAN. We verify the feasibility of the schemes with an implementation and measurements on eight LoRaWAN end-devices and one gateway.**

Keywords–LoRa; LoRaWAN; TDMA; Burst.

I. Introduction

Wireless Internet of Things (IoT) devices are used for collecting data from environment, industrial monitoring, tracking goods and more. The availability of cheap hardware components for wireless IoT devices and the emerging low-power, long-range communication hardware accelerate the deployment of many more IoT nodes. A significant increase of the number of wireless IoT devices is therefore expected in the next few years.

A large number of battery powered devices makes the frequent maintenance of individual devices infeasible. Therefore, low power requirements of IoT devices are of great importance. Furthermore, it is important to use shared limited resources efficiently, e.g., the frequency spectrum.

In the last few years, different Low Power Wide Area Network (LPWAN) technologies have emerged to connect remote wireless IoT devices to the Internet. Many of these technologies have in common that they trade throughput for increasing the range. In this paper, we focus on Long Range Wide Area Network (LoRaWAN) [1], which is currently one of the most promising LPWAN technologies. What is commonly referred to as *LoRa*, consists of two components: (1) LoRa modulation, a physical (PHY) layer, and (2) LoRaWAN, the corresponding Media Access Control (MAC) layer. The LoRa modulation uses Chirp Spread Spectrum (CSS) with different spreading factors (SFs). We focus on the Class A variant of LoRaWAN since it is best suited for low power end-devices and is widely used. LoRaWAN Class A uses the pure ALOHA protocol to access the channel. This limits the channel utilization to a maximum of 18%. In this paper, we use LoRaWAN Class A as a basis and investigate alternative schemes that allow to use the channel more efficiently while minimizing the additional resource demand in terms of on-air-time.

The development of such a scheme involves the following two main challenges. LoRaWAN Class A does not provide a synchronization on the end-devices. Furthermore, LoRa messages exhibit a long on-air-time, which is problematic given the duty cycle limit of 1%, which is enforced by law in Europe for the corresponding unlicensed EU868 frequency band around 868 MHz.

Based on the analysis of the current channel access scheme and the limitations of the LoRaWAN MAC layer, we propose two schemes to increase the channel utilization for certain use cases. Our analysis shows that the proposed schemes can provide more than 60% throughput compared to 18% throughput of the pure ALOHA scheme used in the current specifications of LoRaWAN.

With this paper, we make the following contributions:

- We identify concepts and strategies to extend LoRaWAN Class A to use the channel more efficiently than the original specification without spending a disproportional amount of resources.
- We propose two schemes, *TDMA* and *Burst*, which provide more throughput and are more efficient in specific use cases and which require only small modification of the LoRaWAN Class A layer.
- We evaluate the proposed schemes with calculations as well as implementations on real LoRaWAN hardware.

We start with discussing related work in Section II and providing relevant background information about the LoRa technology in Section III. In Section IV we analyze suitable transmission protocols. Then, we present our proposed schemes in Section V. We compare the considered schemes with calculations in Section VI. Section VII describes our implementation with real LoRaWAN development hardware and Section VIII provides an evaluation of the implementation. Finally, we conclude the paper in Section IX.

II. Related Work

Adelantado *et al.* give an overview of the limits of LoRaWAN [2]. They investigate the influence of the number of end-devices and also consider the duty cycle limit which is imposed by European regulations [3]. Augustin *et al.* provide an overview of the LoRa modulation and the LoRaWAN MAC layer [4]. The study includes an analysis of the channel capacity of LoRaWAN. Vejlgaard *et al.* investigated the impact of interference on coverage and capacity of the LoRaWAN and the SigFox system [5]. Morin *et al.* investigate the power consumption and the corresponding device lifetime of different

 ISBN: 978-1-61208-676-7

IoT schemes including LoRaWAN [6]. Kim *et al*. propose a dual-channel scheme based on LoRaWAN to allow the data of different categories being delivered with different priorities [7]. Phung *et al*. analyze the packet delivery of LoRaWAN, including acknowledged and not acknowledged Class A transmissions as well as Class C transmissions [8]. Reynders *et al*. propose to use coarse-grained scheduling of transmission power, SF, and time in LoRaWAN networks [9]. Beacons are used for time synchronization. Polonelli *et al*. investigate the use of the slotted ALOHA protocol on top of LoRaWAN [10]. In addition, they propose a simple request-reply based time synchronization, which is similar to the time synchronization used in this work.

To the best of our knowledge, the closest related work is the work of Gu *et al*. [11]. They propose a data network with separated control and data plane. For the control plane, they use LoRaWAN. The data plane is based on a multi-hop ZigBee network. Similar to our work, they add synchronization to LoRaWAN in order to use a Time Division Multiple Access (TDMA) based scheme. In contrast to the work of Gu *et al*., we do not use a separate control and data plane, we analyze the possibilities for different applications scenarios in general and in addition propose an *Burst* scheme that is advantageous in terms of aggregated throughput and channel use.

III. LoRa Technology

Two components of the LoRaWAN technology can be distinguished: (1) the LoRa modulation and (2) LoRaWAN. In this section, we will discuss the aspects of both layers which are relevant for this work and how to compute the time on air of a LoRa packet.

A. LoRa Modulation (PHY Layer)

The LoRa modulation is the physical layer. It is based on CSS modulation. Similar to the concept of Direct Sequence Spread Spectrum (DSSS), this modulation uses a large spectral bandwidth to improve the robustness. A common bandwidth setting for LoRaWAN is 125 kHz. In addition, the payload information can be distributed over different amounts of time by selecting different *spreading factors (SFs)*. Increasing the SF increases the time needed to send one byte, but also increases the probability of successful transmission with a given Signal-to-noise Ratio (SNR) and therefore increases the feasible range. This allows to trade throughput for range. The physical layer of LoRa has a payload size between 0 and 255 bytes and comprises a Forward Error Correction (FEC) with 4 different coding rates.

LoRa modulation is used on the sub-1 GHz ISM/SDR frequency bands, e.g., the 915 MHz band in North- and South America or 868 MHz and 433 MHz bands in Europe. Those bands do not require a license and are therefore shared with a large range of other devices which use different modulation schemes. Depending on the region, international regulations restrict the use of these bands in different ways. In Europe for example, there are limits on the transmit power and the duty cycle of each transmitting device is limited to 1 % for large parts of the 868 MHz band.

B. LoRaWAN (MAC Layer)

LoRaWAN [1] specifies the MAC layer which is used together with the LoRa modulation. The specification comprises three different types of devices which form a star-of-star topology. At the core, there are one or multiple network servers (NSs), which implement the back-end with the interface to applications in the Internet. Multiple gateways (GWs) are connected to the network server by the Internet Protocol via Ethernet. Each gateway connects multiple end-devices (EDs) via LoRa wireless links to the network server. Three default frequencies are used for end-devices to join a network, data transmissions and fallback. Additional frequencies can be configured manually.

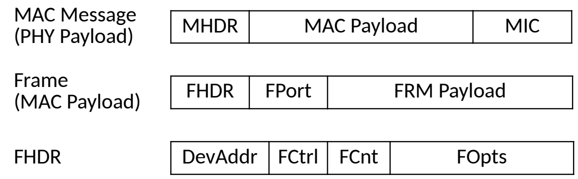

Figure 1. Frame structure of a MAC layer message.

The gateways simply forward messages from the end-devices to the network server and vice versa. A gateway can receive (uplink) messages on different frequencies and with different spreading factors simultaneously. Common gateways feature 8 frequency channels. In contrast to the multi-channel reception, the gateways usually only support to send on a single frequency and a single spreading factor. Most of the available gateways do not support duplex mode, i.e., they cannot receive while transmitting. The network servers manage the connections to the end-devices, keep a state of each end-device and remove duplicate messages originating from different gateways.

LoRaWAN messages are transmitted as payload of a LoRa PHY message. The message structure of data packets is depicted in Figure 1. MHDR is the header of the MAC message. It contains the message type and LoRaWAN version. The MAC payload contains the LoRaWAN frame. The MIC is a message integrity code, which is calculated over MHDR and MAC payload.

The LoRaWAN frame consists of a frame header (FHDR), frame port (FPort) and the frame payload, i.e., the application data. The frame port is a number which specifies which application the data is intended for. The frame header contains the device address (DevAddr), a frame control field (FCtrl) which contains information about the state of the connection, a frame counter value (FCnt), and zero or more MAC layer commands (MAC commands) in the FOpts field.

A LoRaWAN message with 50 bytes of frame payload needs a time-on-air of 176 ms for spreading factor SF7 or 3548 ms for SF12. Accordingly, a device is allowed to send a maximum of 204 messages with SF7 or 10 messages with SF12 in one hour due to the duty cycle limit. This limitation holds for both end-devices and gateways.

The LoRaWAN protocol is divided into three classes. **Class A** provides simple unsynchronized two-way communication between end-devices and network server with focus on the uplink. Downlink messages can be sent only following an uplink message in the so called *receive windows*, which are depicted in Figure 2. Therefore, the downlink throughput and latency are severely limited. **Class B** enabled devices support all features of Class A. In addition, the gateways periodically send beacons to synchronize the end-devices. This allows to

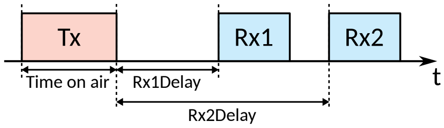

Figure 2. In LoRaWAN Class A end-devices only listen for packets during the defined receive windows.

schedule additional downlink receive windows. With **Class C**, the transceiver of each end-device is constantly turned on, i.e., the end-devices are either receiving or sending at any point in time.

Class B needs additional hardware such as a GPS receiver to keep the gateways globally synchronized since they need to transmit the beacon. Furthermore, Class B is inflexible since all end-devices need to use the same synchronization interval. Currently not many deployed end-devices implement LoRaWAN Class B. Class C is only feasible for devices which have extensive power supply available, which is not the case for scenarios considered in this paper. Because of these reasons, we focus on Class A in this work.

C. Time on air

In order to calculate the on-air-time of a LoRaWAN packet, we use the $\mathrm{toa}()$ function (1) given in the SX1276 datasheet [12]. It depends on the number of payload symbols (2) and the symbol duration (3).

$$\mathrm{toa}(PL) = (n_{\mathrm{preamble}} + 4.25 + n_{\mathrm{pl}}) \cdot T_{\mathrm{sym}} \tag{1}$$

$$n_{\mathrm{pl}} = 8 + \max\left(\left\lceil \frac{8\cdot(\mathrm{PL}+13)-4\mathrm{SF}+28+16\mathrm{CRC}-20\mathrm{IH}}{4\cdot(\mathrm{SF}-2\mathrm{DE})} \right\rceil (\mathrm{CR}+4), 0\right) \tag{2}$$

$$T_{\mathrm{sym}} = \frac{2^{\mathrm{SF}}}{\mathrm{BW}} \tag{3}$$

PL is the number of frame (i.e. application) payload bytes. We adapted the formula such that it is valid for application layer payload by adding 13 bytes which correspond to the LoRaWAN overhead under the assumption of not sending any MAC commands in the FOpts field. SF is the spreading factor. We always enable the CRC ($\mathrm{CRC} = 1$) and the header ($\mathrm{IH} = 0$, *i.e.* implicit header off). We do not make use of the low data rate optimization ($\mathrm{DE} = 0$) and use a coding rate of 4/5 ($\mathrm{CR} = 1$). For the remaining parameters we use the LoRaWAN default values for the EU868 ISM band according to the LoRaWAN standard [1], [13]: the number of preamble symbols $n_{\mathrm{preamble}} = 8$, and bandwidth of the LoRa modulation $\mathrm{BW} = 125\,\mathrm{kHz}$ (default for data rates DR0 - DR5).

IV. Transmission Protocols

In this section, we discuss all channel access and synchronization schemes we consider and mention the restrictions implied by LoRaWAN Class A. Then, we describe the considered schemes to increase the channel utilization, including our two proposed schemes *TDMA* and *Burst*.

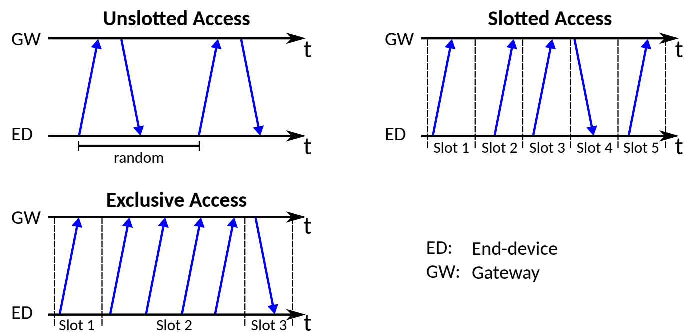

Figure 3. Basic channel access schemes.

A. Channel Access Schemes

Figure 3 provides an overview of the basic channel access schemes considered in this paper. The message exchange between an end-device and the gateway consists of uplink messages that are directed from end-device to gateway and downlink messages from gateway to end-device. Note that in LoRaWAN Class A the time between uplink and downlink messages is fixed, see Figure 2. Depending on the downlink queue in the network server and whether the uplink requests an acknowledgment, the network server transmits a downlink in the receive window or not. In other words, not every uplink message is necessarily followed by a downlink message.

Unslotted Access: End-devices can send messages anytime. Due to this uncoordinated access of the channel, there is a relatively high probability of colliding transmissions.

Slotted Access: The time is partitioned into slots of a fixed length. The end-devices are allowed to access the channel only at the beginning of a slot. This reduces the probability of collisions in comparison to the unslotted protocol. However, the clocks of the end-devices and the network need to be synchronized. In addition, all messages need to fit into the same time slot length.

Exclusive Access: A scheduler determines a time-driven schedule, which defines the assignment of devices to time intervals and frequencies to each device. The resulting schedule, with mutually exclusive channel accesses, precludes message collisions. End-devices need to be synchronized and to receive and store information, which determines the time interval in which they are allowed to access which channel.

B. Time Synchronization

The slotted access and exclusive access scheme require end-devices to be time synchronized. Two options that we consider are shown in Figure 4. The selection is based on the opportunities of the LoRaWAN Class A standard, i.e., sending beacons from the gateways is not possible as downlink messages can only be sent as answer to a previous uplink message.

Request: An end-device and the network server exchange dedicated messages via a gateway to synchronize the clock of the end-device.

Piggy-Back: The request for a time-synchronization is part of a regular data message, which is sent to the network server via a gateway. Synchronization information, like a timestamp, is then part of a downlink message sent from the network

 ISBN: 978-1-61208-676-7

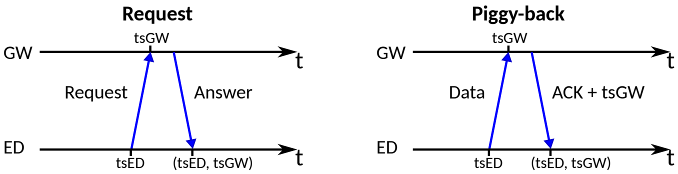

Figure 4. Basic synchronization schemes.

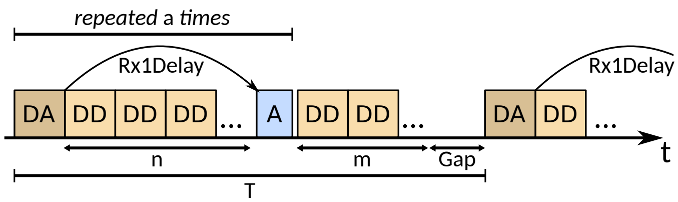

Figure 5. *TDMA* scheme with interleaved acknowledgment/sync packets.

server via a gateway to the end-device. This may be either an acknowledgment or a data downlink message.

C. Considered Transmission Schemes

We will study the following four protocols. They are selected as they represent different extreme solutions of a wide spectrum of possible schemes. Many generalizations and combinations of these four basic schemes are possible.

Pure ALOHA: This protocol has been proposed in [14] and uses the unslotted access scheme and therefore does not require synchronization.

Slotted ALOHA: This scheme has been proposed in [15]. It combines piggy-back synchronization with a partition of the channel in fixed time slots. Depending on the drift of the clock of end-devices, not all uplink messages need be answered by a synchronization message, e.g., a timestamp from the network server. These synchronization messages from the gateway are also subject to message collision.

TDMA: We propose the *TDMA* scheme which takes into account the LoRaWAN Class A specifics. In our scheme, the end-devices repeatedly send data packets D according to a fixed TDMA schedule, see Figure 5. In order to achieve synchronization between the clocks of the end-devices, special data messages (denoted as DA) are answered by acknowledgment messages (denoted by A) from the network server, which include synchronization information. The required rate of the synchronization messages depends on the clock-drift of the end-devices and the required time synchronization accuracy. There are several obvious options on determining such a TDMA schedule and sending it to the end-devices via downlink messages. We propose one possible schedule in Section V-B3.

Burst: For application with non-critical latency demands, we propose a new *Burst* scheme depicted in Figure 6. Multiple messages are aggregated and sent together in a burst. The scheme uses two different LoRa channels (two different frequencies): The *request channel* for coordinating the transmission of burst data messages and to perform time synchronization and the *burst channel* for transmitting bursts of uplink data messages. In order to send a burst, the end-device first needs to request a burst transmission slot from the network server by sending a burst request message on the

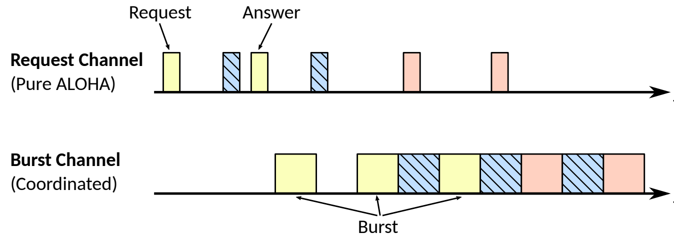

Figure 6. *Burst* scheme with on-demand synchronization with 3 nodes (light, hatched, dark).

request channel. A scheduler on the network server determines which end-device is allowed to send in which time slots.

V. Analysis

In this section, we describe and analyze the considered schemes in order to increase the channel utilization. First, we start defining the model and metrics to compare the different schemes.

A. Model and Metrics

1) Basic Model Assumptions: We suppose that our communication scenario consists of N end-devices and K gateways. We further assume that all end-devices and gateways can reach any other end-device or gateway directly. In general, we assume that every end-device connected to the network generates data of size D with a fixed period T.

Time-On-Air: The on-air-time of a LoRa transmission can be calculated, see Section III-C. We use t_D for the on-air-time of any data uplink packet and t_A for any non-data packet (this includes requests, answer, synchronization, and coordination packets).

Clock Drift: After synchronization, the clocks in the network server and the end-devices will drift apart. We denote the time difference between previous time synchronization and the time when the clock of the end-device is accessed as Δt. The maximal absolute value of the time difference between the end-device and the network server at this point in time is then modeled as

$$\tau(\Delta t) = \tau_0 + \Delta t \cdot \tau_1 \quad (4)$$

where τ_0 is the synchronization error due to the synchronization protocol between the network server and the end-device and τ_1 denotes the clock-drift of the end-device. In order to account for the clock inaccuracy, we expand the actual time-on-air t_D and t_A by a safety margin and define the expanded time for data packets as $s_D = t_D + 2 \cdot \tau(\Delta t_{\max})$ where $\Delta t_{\max}$ is the maximum time between time synchronization updates. We define s_A accordingly.

Duty Cycle Limit: The LoRaWAN standard limits the time, a device is on air, i.e., sending a message. In this paper, we focus on LoRaWAN EU868 and therefore the European regulations (ETSI EN 300 220-1 [3]) apply which enforce a the duty cycle limit of $L = 0.01$ for each device. This also applies to the gateways. According to the regulations, the time interval which is considered to evaluate the adherence to the duty cycle limit is $I = 3600\,\text{s}$. This constraint in terms of duty cycle L and measurement interval I strongly restricts the design space of efficient LoRaWAN based protocols.

2) Metrics for Comparison: In the following, we list the four metrics we are interested in to compare the considered transmission schemes.

The **success probability** P_{succ} is defined as the probability that an attempt of an end-device to transmit a data packet to the network server is successful, i.e., there is no colliding transmission.

We define the **throughput** S as the average accumulated time of successful data message transmissions from all end-devices relative to the total time. If there are in total $M_{\mathrm{succ}}(\Delta t)$ successful message transmissions from any end-device in a time interval of length Δt, then

$$S = \lim_{\Delta t \to \infty} \frac{M_{\mathrm{succ}}(\Delta t) \cdot t_D}{\Delta t} \tag{5}$$

where t_D is the time on air for transmitting a data message.

The **device time utilization** W_d for a specific device d is the average accumulated time the device is (successfully or unsuccessfully) transmitting relative to the total time. For example, if there are $M_d(\Delta t)$ message transmissions from this specific end-device in a time interval of length Δt, then

$$W_d = \lim_{\Delta t \to \infty} \frac{M_d(\Delta t) \cdot t_D}{\Delta t} \tag{6}$$

where t_D is the time on air for transmitting a data message. We denote the device time utilization of an end-device by W_{ED} and the the device time utilization of an gateway by W_{GW}.

The **send efficiency** E is the average accumulated time all end-devices are transmitting data packets which are successfully received relative to the total time all devices are transmitting. For example, if there are $M_{\mathrm{succ}}(\Delta t)$ successful data transmissions, $M_{\mathrm{unsucc}}(\Delta t)$ unsuccessful transmissions, and $M_{\mathrm{sync}}(\Delta t)$ synchronization messages in a time interval of length Δt, then

$$E = \lim_{\Delta t \to \infty} \frac{M_{\mathrm{succ}} \cdot t_D}{(M_{\mathrm{succ}}(\Delta t) + M_{\mathrm{unsucc}}(\Delta t)) \cdot t_D + M_{\mathrm{sync}}(\Delta t) \cdot t_A} \tag{7}$$

B. Considered Transmission Schemes in Detail

In this section, we explain the considered transmission schemes in more detail and provide the corresponding performance analysis.

1) Pure ALOHA Scheme: The LoRaWAN Class A scheme uses the pure, i.e., unslotted, ALOHA scheme for channel access. If the channel access attempts from end-devices are assumed to be Poisson distributed with an average of G accesses per packet time, then the performance metrics are known to be [14]

$$S = G \cdot \mathrm{e}^{-2G} \qquad P_{\mathrm{succ}} = \mathrm{e}^{-2G} \tag{8}$$

In (8), S corresponds to our definition of throughput and G is the number of access attempts per packet time (access rate). The maximum throughput is about $S_{\max} = 18.4\%$, which is achieved for $G = 1/2$. This leads to a corresponding success probability of about $P_{\mathrm{succ,max}} = 0.37$.

In other words, even a low maximal throughput comes with a low success probability. One intuitive measure to increase the success probability is to use positive acknowledgment and retransmission. But this leads to additional channel accesses due to the re-transmitted packets and the acknowledgment for each correctly requested packet. Finally, acknowledgment packets are also subject to collisions. This means we can increase the success probability only by reducing the channel accesses, i.e., by reducing the throughput.

In reality, the duty cycle limit restricts the number of feasible operating points of the pure ALOHA scheme. In our analysis, we want to achieve a throughput S defined by N, t_D and T. We use (8) to calculate $G = \frac{N \cdot t_{\mathrm{Tx}}}{T}$ from S in (9). The actual time-on-air t_{Tx} for each pure ALOHA end-device to generate a total throughput of S is always larger then the theoretical send time required without collisions ($t_{\mathrm{Tx}} > t_D$).

$$S = \frac{N \cdot t_D}{T} \quad \overset{G(S)}{\Rightarrow} \quad G \overset{!}{=} \frac{N \cdot t_{\mathrm{Tx}}}{T} \tag{9}$$

With this, we can determine the end-device time utilization W_{ED} (see (10)). The device time utilization of the gateway is 0 since no transmissions are acknowledged and no synchronization is preformed. The send efficiency is determined by $E = \frac{S}{G}$.

$$W_{\mathrm{ED}} = \frac{m \cdot t_{\mathrm{Tx}}}{T} = \frac{G}{N} \overset{!}{\leq} L \qquad W_{\mathrm{GW}} = 0 \overset{!}{\leq} L \tag{10}$$

Due to retransmissions, a delay has to be accounted for this scheme.

2) Slotted ALOHA Scheme: As in the case of pure ALOHA, the throughput analysis for slotted ALOHA is well known and established [15]:

$$S = G \cdot \mathrm{e}^{-G} \qquad P_{\mathrm{succ}} = \mathrm{e}^{-G} \tag{11}$$

If we assume perfect synchronization with no overhead and no interference on the frequency band (i.e. all transmitted packets are received successfully if they are not overlapping in time), the maximum throughput is about $S_{\max} = 36.8\%$, which is achieved for $G = 1$, where G is the access rate. This would be an improvement by a factor of 2 in comparison to the pure ALOHA protocol in terms of maximally achievable throughput. The success probability remains unchanged with about $P_{\mathrm{succ,max}} = 36.8\%$ for this operating point.

But as we know, the scheme requires the end-devices to be synchronized. Therefore, the actual throughput of a slotted ALOHA system is lower than S and the success probability is lower than P_{succ}. In addition, the rate of acknowledgment packets transmitted by the gateways is limited by the duty cycle limit L. This fact constrains the possible design space for slotted ALOHA further.

3) TDMA Scheme: We propose the *TDMA* scheme which is depicted in Figure 5. The end-devices send a data packet (DD or DA) of fixed size D according to a Time Division Multiple Access (TDMA) schedule. The schedule repeats periodically with period T. In each period, all N end-devices send exactly one data packet. Only a small number $a < N$ of all transmissions are acknowledged (DA packets) in each period in order to keep the overhead low and to comply with the duty cycle limit. The period between synchronization of a particular end-device is $\frac{T \cdot N}{a}$ on average. The acknowledgment A is used to transfer a timestamp from the gateway to the end-device. The acknowledgment slots of multiple periods are evenly distributed to the participating end-devices such that the end-devices are alternately synchronized. In Figure 5, the case for $a = 1$ is depicted. All devices, including the gateways, send on a single frequency.

The LoRaWAN standard specifies the `Rx1Delay` time between the end of the data packet and the corresponding acknowledgment in order to allow the server to react to the received message and to transmit a reply, see Figure 2. During this interval, $n = \left\lfloor \frac{\texttt{Rx1Delay}}{t_D} \right\rfloor$ not acknowledged transmissions are scheduled. This sequence of DA, DD ..., A is repeated a times. The rest of the period is used to schedule m not acknowledged transmissions from the remaining $m = N - a \cdot (n + 1)$ end-devices.

Due to the ETSI duty cycle limit, not all combinations of (N, t_D, T) are possible. The relation for the *TDMA* scheme between payload size D (indirectly given as on-air-time with safety margin s_D) and the period T is given by (12). The application payload and the on-air-time of a packet is linked by the $toa()$ function described in Section III-C.

The relation of the components of one *TDMA* schedule period is given in (12). A non-zero gap (Gap) allows the scheme to be suitable for combinations of parameters which do not exactly fill the *TDMA* schedule. Gap is determined by parameters in (12) (N, t_D, t_A and a).

The constraints due to the duty cycle limit based on the device time utilization are given in (13). The device time utilization due to downlink messages from the network server can be distributed to K gateways.

$$T = (\texttt{Rx1Delay} + s_A) \cdot a + \left(N - \left\lfloor \frac{\texttt{Rx1Delay}}{s_D} \right\rfloor \cdot a \right) \cdot s_D + \text{Gap} \tag{12}$$

$$W_{\text{ED}} = \frac{t_D}{T} \overset{!}{\leq} L \qquad W_{\text{GW}} = \frac{a \cdot t_A}{T} \overset{!}{\leq} L \cdot K \tag{13}$$

L corresponds to the duty cycle limit, see Section V-A1. The send efficiency is $E = \frac{N \cdot t_D}{N \cdot t_D + a \cdot t_A}$.

In the case of no packet loss, the maximum time a clock of an end-device is not synchronized is $\Delta t_{\max} = \left\lceil \frac{N \cdot T}{a} \right\rceil$ and the *TDMA* scheme provides a success probability of $P_{\text{succ}} = 1$ since none of the transmissions can be overlapping due to the TDMA schedule.

4) Burst Scheme: As a second scheme, we propose the *Burst* scheme, which is depicted in Figure 6. In contrast to the *TDMA* scheme, in the *Burst* scheme the end-devices are not continuously synchronized. The end-device synchronizes their clock to the network server before sending a burst. Synchronizing for sending a single packet would implicate a large overhead. Therefore, in general the end-devices aggregate data and send multiple packets bundled together in a *burst*.

The proposed scheme uses two different channels: The *request channel* for handling requests and synchronization and the *burst channel* for transmitting the bursts. The request channel is uncoordinated and uses pure ALOHA, whereas the burst channel is coordinated by a scheduler on the network server. There is no explicit acknowledgment but it would be possible to acknowledge the complete transmission of the last burst in the following burst answer.

In order to send a burst, the end-device first needs to request a burst transmission and obtain synchronization by sending a burst request message on the request channel to the network server. The network server maintains a schedule of all scheduled transmissions. With a burst answer transmitted on the request channel, the network server sends a timestamp for synchronization and the information when the end-device is allowed to send the individual packets of the burst. Then the end-devices synchronizes its clock and stands by for sending the burst packets in the designated slots.

The scheduler running on the network server makes sure that no burst can collide on the burst channel. If there are no suitable slots available in the following time interval of length Δt which defines the safety margin $\tau(\Delta t)$, the network server can deny an access to the burst channel. In this case, the end-device tries to request the burst channel again after a backoff time. The backoff time is increased exponentially with every denial. This prevents overloading the request channel if there are simultaneous requests from many nodes.

The LoRaWAN specifications require the end-device to wait with sending a new message until the receive window of the previous message has passed no matter whether the end-device sent a confirmed or unconfirmed message. For this reason, a single burst transmission, which consists of multiple packets, needs to be sent as individual packets with gaps of at least `Rx1Delay` in between.

For the analysis of the performance of this scheme, we assume that the average period between two burst transmissions of a single device is T. The accumulated duration of a burst t_{Burst} and the relation to the burst period T are given in (14). The device time utilization and the corresponding limits are given in (16). Since the messages are sent in bursts, they are not evenly distributed over time, we need an additional constraint, (15), that ensures that the absolute maximal transmitting time within $I = 3600\,\text{s}$ is not exceeded.

$$\begin{aligned} t_{\text{Burst}} &= (n_B + n_G) \cdot s_D \\ T &= N \cdot t_{\text{Burst}} \end{aligned} \tag{14}$$

$$n_B \cdot t_D + t_A \overset{!}{\leq} L \cdot I \tag{15}$$

$$W_{\text{ED}} = \frac{n_B \cdot t_D + t_A}{T} \overset{!}{\leq} L \qquad W_{\text{GW}} = \frac{t_A}{t_{\text{Burst}}} \overset{!}{\leq} L \cdot K \tag{16}$$

n_B indicates the number of LoRaWAN data packets that are sent in a single burst, $n_G \geq 0$ is the number of slots which are unused (gap) following a burst transmission. The unused slots are, in some cases, necessary to keep the aggregated on-air-time (W_{GW}) below the allowed limit L on the side of the gateway. In the *Burst* scheme, the number of nodes is not restricted by the duty cycle limit, as it is the case with the *TDMA* scheme, but depends on the size of the bursts. The success probability of sending the bursts is $P_{\text{succ}} = 1$ since the synchronization and scheduling of the network server makes sure that no transmissions overlap. The probability for a collision on the request channel is supposed to be small since the aggregated on-air-time for acknowledgments by the gateway is limited to $L = 1\%$. The throughput of the

TABLE I. INPUTS AND OUTPUTS OF THE CALCULATIONS.

Input	
D	Application payload size
T	Period with which an end-device sends the application payload
N	Number of participating end-devices
K	Number of gateways
–	LoRaWAN and LoRa modulation parameters

Output	
–	Feasibility
E	Send efficiency
W_{ED}, W_{GW}	Device time utilization
P_{succ}	Probability of a transmission to arrive

entire scheme is determined as $S = \frac{n_B \cdot t_D}{t_{\mathrm{Burst}}+2\cdot t_A}$ and the send efficiency is given by $E = \frac{n_B \cdot t_D}{n_B \cdot t_D + 2\cdot t_A}$.

5) Comparison of Slotted ALOHA with TDMA: If we compare slotted ALOHA and the *TDMA* scheme, it is obvious that the minimum synchronization overhead is the same (unless out-of-band synchronization mechanisms are used). In both schemes, all nodes need to be continuously synchronized within the required precision such that packets can be sent inside a time slot. The only difference in overhead is the assignment of each node to slots which exists in the *TDMA* scheme but not in the slotted ALOHA scheme. However, this assignment can be pre-configured (e.g. based on the node's ID), which means that the overhead of the slot assignment of the *TDMA* scheme is negligible. Since the *TDMA* scheme provides a significantly better success probability ($P_{\mathrm{succ}} = 1$) it is always beneficial to use *TDMA* instead of slotted ALOHA. Because of this reason, we omit the slotted ALOHA scheme in our calculations which follow next.

VI. CALCULATIONS

In order to compare the three remaining considered schemes, we calculate the previously described metrics.

A. Calculation Model

We numerically evaluate the access schemes described in Section V to obtain the metrics and the feasibility of the considered schemes at different design points. The most important input and output quantities of the calculation are listed in Table I.

The main input of each scheme consists of three parameters, application payload size D in bytes, period between transmissions T of a single end-device in seconds and the number of end-devices N. These three parameters describe the requested throughput. By evaluating the equations of the considered schemes, we determine which areas of the design space are feasible and which scheme provides the best send efficiency.

Further parameters for the calculations are the spreading factor SF, plus further LoRa modulation parameters, which are described in Section III-B and are kept constant for all calculations. In addition, there are parameters which are relevant only for a subset of the schemes: The number of gateways K which is relevant for the *TDMA* and *Burst* scheme, the number of acknowledged transmission a in a *TDMA* period. Another parameter for the calculation are clock offset and drift values which are relevant for the *TDMA* and *Burst* scheme. Based on measurements in Section VIII-A, we assume a clock offset τ_0 of 15 ms and a clock drift of 20 ppm. The application payload D size is a discrete parameter by definition. In order to keep

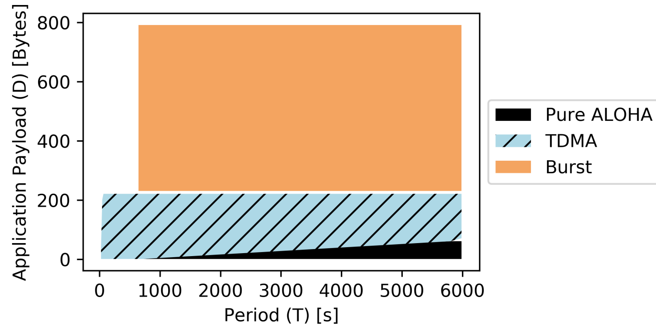

Figure 7. Feasible combinations of period T and application payload D (fixed SF=7 and N=100). The color/hatching indicates the scheme with highest send efficiency E.

the calculations tractable, the application payload D has been sampled with a step size of 10 bytes, the period T with a step size of 20 s. In our model, we consider it infeasible to send multiple packets in a sequence, except for the *Burst* scheme. This limits the application payload D in the *TDMA* scheme to the maximum payload of a LoRaWAN packet (which is 222 bytes for SF7).

B. Calculation Results

In this section, we discuss the results from our calculations in terms of feasibility, send efficiency and throughput.

1) Feasibility and Efficiency of the Schemes: We investigate which scheme is most efficient for a given throughput defined by N, D, and T. In Figure 7 we show an exemplary plot for SF7 and $N = 100$ with 1 gateway. The non-white areas represent the feasible combinations of input parameters. The shading of the area indicates which of the scheme has the best efficiency E. Please note that our assumptions of the model limit the feasibility.

The calculations show that the *TDMA* scheme is feasible and efficient in a large range of periods and payload sizes. For very large periods and small payload size, i.e., low requested throughput, the pure ALOHA scheme has highest send efficiency since the overhead for the continuous synchronization is large compared to the data that should be transmitted. The *Burst* scheme is not feasible for small periods. For lower periods, the device time on the end-device W_{ED} would exceed the duty-cycle limit. For SF7 even for very short periods, the *TDMA* scheme is feasible and provides higher send efficiency. This is expected since the pure ALOHA needs approximately 4 failed transmission to send 1 successful message in the best case of $G = 0.5$. The *TDMA* scheme only needs to send synchronization messages for a fraction of all nodes in each period. The calculations for different spreading factors and different number of end-devices N yield similar insights.

2) Maximum Throughput: In this section, we investigate the maximum achievable throughput of each scheme.

In Figure 8, the throughput S for different number of end-devices N in the case of using only a single gateway is depicted. The *TDMA* scheme can provide significantly higher throughput values compared to the pure ALOHA scheme. However, the maximum throughput for the *Burst* scheme is comparable to the pure ALOHA scheme when using only one gateway.

 ISBN: 978-1-61208-676-7

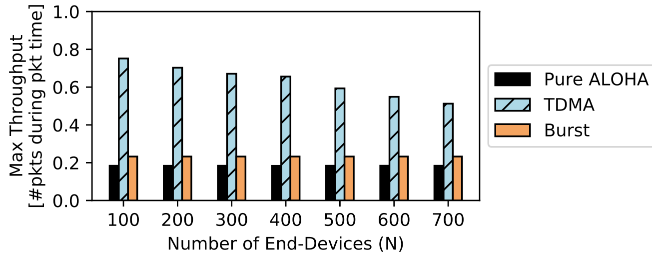

Figure 8. Maximum throughput S for different combinations of period T and application payload D in relation to N (SF=7 and 1 gateway).

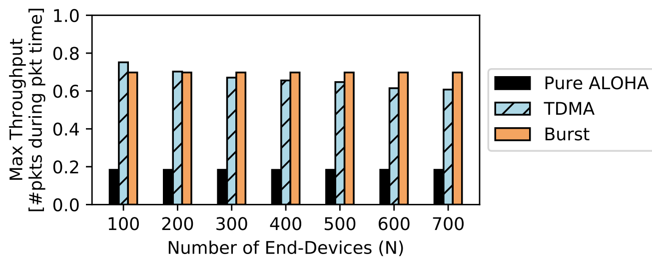

Figure 9. Maximum throughput S for different combinations of period T and application payload D in relation to N (SF=7 and 3 gateways).

If we increase the number of available gateways, we can increase the maximum throughput of the *TDMA* and the *Burst* scheme. In Figure 9, we show the maximum throughput in case of 3 gateways. The calculations show that especially the *Burst* scheme profits from the additional gateways. For the *TDMA* scheme it helps only if the number of end-devices is large. With 3 gateways and SF=7 and less than 300 end-devices, the schemes can provide a throughput up to 70% whereas the pure ALOHA scheme only provides up to 18% throughout.

As shown in Figure 9, the maximum *TDMA* throughput decreases with increasing number of end-devices whereas the maximum *Burst* throughput is constant. For the *TDMA* scheme the overhead to keep nodes synchronized grows with the number of devices. For the *Burst* scheme, the overhead of handling burst requests on the gateway can be kept constant for any N by increasing the period. However, the transferred data per end-device decreases with increasing number of end-devices.

C. Selection of Transmission Scheme

Finally, we will provide guidelines that help to select an appropriate transmission scheme based on the analysis in the previous sections. An overview in the form of a decision tree is given in Figure 10.

In the pure ALOHA scheme, the throughput is constrained by the collisions ($P_{\text{succ}} \leq \frac{1}{2\text{e}} = 18.4\%$) which are accepted in order to not require a time synchronization. The *TDMA* and *Burst* scheme have a success probability of $P_{\text{succ}} = 1$ but comprise an overhead due to the necessary time synchronization. The collisions of pure ALOHA and the synchronization overhead of the *TDMA* and *Burst* schemes reduce the send efficiency. In addition, lack of synchronization increases the necessary safety margin $\tau(\Delta t)$. A larger safety margin influences whether a requested throughput can be achieved by

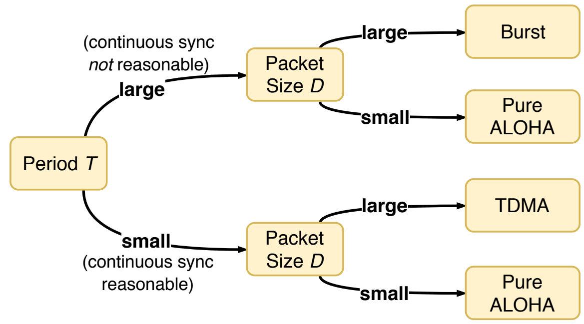

Figure 10. Decision tree for selecting channel access / synchronization scheme.

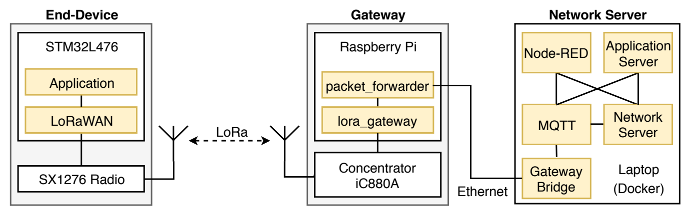

Figure 11. LoRaWAN setup.

a certain scheme but does not influence the send efficiency directly.

If an application requests only a very low throughput (i.e. T large and D small), the pure ALOHA scheme is suitable and provides good send efficiency $E = S/G$, which is large compared to the send efficiency of the synchronized schemes.

For larger requested throughputs there are two cases. If the period T should be small, continuous synchronization of all end-devices is reasonable and therefore the *TDMA* scheme is suitable. If the period T should be large, continuous synchronization is not necessarily reasonable and therefore the *Burst* scheme is more suitable. In certain cases of small period and small packet size, the *TDMA* scheme is not feasible due to the duty cycle limitations. This holds especially for larger spreading factors. In this case, the pure ALOHA is the only option.

VII. Implementation

In this section, we explain the implementation on real LoRaWAN development hardware, which is used to demonstrate the feasibility of the two proposed schemes. The implementation is based on the framework of Polonelli *et al.* [10], which implements the basic mechanism for time synchronization on LoRaWAN development hardware. We extended the framework by a mechanism to enforce frequency channels, using the history of synchronization messages for time synchronization, and the implementation of our proposed *TDMA* and *Burst* scheme.

A. Setup

An overview of the setup which is used in this work is depicted in Figure 11.

1) Network Server: For the network server, we use LoRaServer [16], which is an open source implementation of the corresponding LoRaWAN specifications. We run the different components of the LoRaServer inside different Docker [17]

 ISBN: 978-1-61208-676-7

containers on a Lenovo ThinkPad T460s laptop (Intel Core i7-6600U, 2.60 GHz, 19 GiB RAM). The LoRaServer project consists of 3 main parts: the *gateway bridge*, the *network server* block and the *application server*. The gateway bridge is responsible for communicating with the gateway. The network server block implements the LoRaWAN MAC-layer on the server side. The application server manages different user applications and provides a web-interface. The received and transmitted messages are exchanged via the open-source Eclipse Mosquitto MQTT broker [18]. Furthermore, we use a Node-RED [19] container to implement network flows.

2) Gateway: Our LoRaWAN gateway consists of a Raspberry Pi 2 and the iC880A concentrator. This setup supports the simultaneous reception of messages with all spreading factors (SF7 - SF12) on 8 different frequencies. The gateway is connected to the laptop via Ethernet. The software running on the gateway is the *lora_gateway* and *packet_forwarder* from the Lora-net reference project on GitHub [20].

3) End-Device: Each end-device consists of an STM32L476RG microcontroller developer board (Nucleo-L476RG) combined with an *mbed* SX1276 868 MHz LoRa shield. The software running on the end-devices is based on *LoRaMAC-node* from the *Lora-net* reference project on GitHub [20] and the framework of Polonelli *et al*. [10].

B. Time Synchronization

Our proposed schemes require synchronized clocks across end-devices and network server. By default this is not supported with the LoRaWAN MAC layer. Therefore, we add a custom time synchronization to LoRaWAN. The implementation of the time synchronization is based on the implementation described in the work of Polonelli *et al*. [10] and is similar to the scheme used by Gu *et al*. [11].

The implemented synchronization scheme is depicted in Figure 4. This scheme is based on a pair of corresponding timestamps. The end-device takes a local timestamp right before sending a synchronization request. The gateway receives this request and immediately takes the corresponding global timestamp (tsGW). This timestamp is then sent to the end-device in the following synchronization answer message. One such pair of timestamps can be used to calculate the offset between the local and global clocks. This offset is used in the *Burst* scheme. For the *TDMA* scheme, this procedure is repeated, which leads to multiple synchronization pairs. From multiple synchronization pairs, the clock offset and drift are calculated, which can then be used to convert the local timestamp on the end-device to the global time of the network.

C. Enforcing Frequencies

By default, LoRaWAN end-devices choose pseudo-randomly one of the 3 default frequencies to sending LoRa packets. However, our schemes use a fixed assignment of channels to end-devices and in the *Burst* scheme no uncoordinated requests are tolerated in the coordinated burst channel. Therefore, we enforce the end-devices to use the assigned frequency. To implement this, we configure the channel mask on the end-devices such that all but one configured LoRa channels are disabled at any point in time. For this, we make a request to the request/confirm based MAC Information Base (MIB) interface of the LoRaWAN MAC layer.

D. TDMA Implementation

The *TDMA* scheme, proposed in Section V-B3, requires an implementation of a bootstrap mechanism to obtain synchronization and to send messages according to a schedule. The implementation uses the piggy-back synchronization method described in Section VII-B. The end-device re-calculates the offset and drift value each time a new synchronization pair is obtained. The obtained values are used to provide timestamps from a virtual clock, which is synchronized to the clock of the network server due to the synchronization scheme. After boot up and joining the LoRaWAN network, the virtual clock is not yet synchronized and therefore the end-device requests the first synchronization point by sending a dedicated synchronization request message on one of the 3 default LoRaWAN channels. This bootstrap mode is at the same time used for fallback in case the end-device looses synchronization in the case of not receiving a synchronization answer.

In general, a scheduler on the network server can send a schedule to the end-device in the synchronization answer. For the implementation of our experiments, we define the TDMA schedule statically. The individual end-devices determine their send slots based on the current time and the end-device's ID, similar to the implementation of Gu *et al*. [11]. After the bootstrap phase or after sending a data packet, an end-device obtains the current timestamp and together with the ID it determines the time to transmit the next packet and the transmission type (DA, DD or A). In the time between the transmissions, the end-device is put into sleep mode. Timers that are based on the virtual clock are configured to wake the end-device up and send the transmission scheduled by the TDMA schedule.

E. Burst Implementation

The implementation of the *Burst* scheme requires a logic on the end-device to send messages on the assigned channel at the assigned point in time, a scheduler on the network server, and a mechanism to prevent transmitting while receiving a burst data message on the same gateway.

When the end-device wants to send a burst, a synchronization procedure as described in Section VII-B is initiated. In addition to the timestamp, the network server sends the time difference between timestamp and start time of the burst.The end-device then uses the timestamp to synchronize the virtual clock, configures a timer to wake up when the burst should start, and goes into sleep mode. The end-device then alternately sends packets and sleeps in between until all packets of the burst are transmitted. After completing a burst, the end-device sleeps until the start of the next burst. In the implementation for our experiments, we use a configurable interval between sending bursts. The length of this interval is randomized to ensure that the pattern of burst requests changes over time.

The scheduler is implemented as stateful Node-RED flow function. The schedule consists of time slots of size s_D, which are assigned to end-devices which request to send a burst. With this, the scheduler guarantees that the allocated burst patterns do not overlap.

Most of the commercially available LoRaWAN gateways (including the one used for the experiments in this paper) are not capable of full-duplex, i.e., they cannot transmit while receiving. This causes the gateway to miss burst data packets if it sends an answer to a burst request at the same time. We mitigate this problem, by not answering burst requests, if the burst answer would overlap with scheduled burst data packet. Please note, that this problem has no influence on the throughput if two or more gateways are used. For the case of two gateways for example, one gateway can be configured to only receive and the other one to only transmit. The gateways then can periodically switch roles in order to distribute the accumulated transmit time such that the device send time utilization of the gateways is kept below the duty-cycle.

VIII. Evaluation

We perform experiments to verify that our implementation behaves as expected and to demonstrate that our proposed schemes work on real hardware.

A. Synchronization Accuracy

First, we examine whether our assumptions of clock accuracy for the calculations are in accordance with values measured on real hardware. For this, the synchronization accuracy which can be achieved with the implementation is measured. The synchronization implementation is based on the framework of Polonelli *et al.* [10].

We measure the offset between two periodically synchronized end-devices located in the same room. For this, both end-devices synchronize every 10 s with the gateway. In order not to interfere with each other, one of the end-devices runs with a configured offset of 5 s relative to the other end-device. An experiment with 200 transmissions over a time interval of 30 minutes with spreading factor SF7 has been conducted. The the measured offset is is in the range ± 0.0123 s. From these measurements we conclude that $\tau_0 = 15$ ms, which we use in the calculation in Section VI, is a good upper bound for the offset between two synchronized end-devices.

B. Evaluation of the TDMA Scheme

In order to verify that the implementation performs as expected from the calculations, we run the *TDMA* scheme and measure the packet delivery ration (PDR). For the evaluation, we determined a feasible configuration which satisfies all the constraints in Section V-B3 with $N = 64$, an application payload of $D = 50$ bytes, a period $T = 30$ s, and $a = 1$ acknowledged transmission per period. This configuration would ideally lead to a throughput of $S = 25.2\%$ with 1 gateway.

We perform measurements with the mentioned configuration with 8 end-devices and 1 gateway, which are located in the same room. The slots for the remaining 56 end-devices are unused, i.e., no device sends anything during this time. In order to verify the synchronization implementation, we artificially increase in our experiment the synchronization rate such that in every period exactly one of the 8 participating end-devices sends a synchronization request. The measurements includes 253 periods, which corresponds to 127 minutes or 2024 uplink packets.

The resulting packet-delivery-ratio (PDR) for each end-device is plotted in Figure 12. The mean PDR over all 8 end-devices is 99.75%. A probable cause for not receiving all packets successfully is interference from other LoRa transmissions on the same frequency. With this, we demonstrate that an implementation of the described TDMA discipline on top of LoRaWAN is feasible and that the *TDMA* scheme can be implemented on LoRaWAN development hardware.

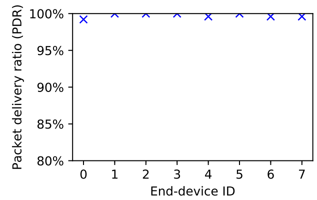

Figure 12. Packet delivery ratios (PDR) of the *TDMA* evaluation.

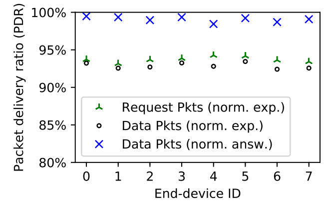

Figure 13. Packet delivery ratios (PDR) of the *Burst* evaluation.

C. Evaluation of the Burst Scheme

In an evaluation experiment, we run the *Burst* scheme implementation to verify that it performs as expected from the calculations. For this, we use the following configuration: $N = 8$, $n_B = 4$, $T = 60.02$ s (95 slots), SF7, $D = 222$ bytes. This leads to a throughput of $S = 31.0\%$. This configuration uses an artificially decreased period T in order to test the system under high load. This means, a single physical device represents multiple virtual devices.

In our experiments, we use the setup of Section VIII-B with 8 end-devices and 1 gateway. The use of 1 gateway means that some burst requests are not answered the first time because the gateway does not send a burst answer while receiving a burst data packet. Therefore, we expect that the amount of positive answers to the the burst requests is smaller than 100%.

The results of the measurements are shown in Figure 13. The star shows the number of burst requests, the circle shows the number of burst data messages of each end-device. Both values are normalized by the expected number of burst requests or data messages, which is calculated using the values of the configuration. The average number of expected bursts from each end-device is 204.96 (absolute number). On average the end-devices received 93.92% of the expected burst requests and 93.04% of the expected burst data packets As expected, both values are lower than the PDR of the *TDMA* evaluation since a significant amount of burst requests is not answered by the network server to allow proper reception of burst data messages. The cross represents the number of received burst data packets normalized by the number of unique burst requests that have been answered by the network server. On average the end-devices received 99.07% of all burst data packets which followed on a answered burst request. This number is comparable with the PDR of the *TDMA* evaluation. In summary, the results show that the implementation of the *Burst* scheme is feasible.

IX. Conclusion

In this paper, we first analyze the existing LoRaWAN Class A scheme. Based on this analysis, we then propose the *TDMA* and the *Burst* scheme, which allow to use the channel more efficiently. The *TDMA* scheme uses a single frequency channel and requires the end-devices to remain synchronized. The *Burst* scheme uses two frequency channels, aggregates data to be sent and requires the devices to be synchronized only when sending a burst. With calculations, we investigate in which scenarios the proposed schemes are advantageous compared to the current version of the LoRaWAN specifications, which uses pure ALOHA. Our analysis shows that

 ISBN: 978-1-61208-676-7

the proposed schemes can provide more than 60% throughput compared to 18% provided by the pure ALOHA scheme used in the current specifications of LoRaWAN. With experiments with eight end-devices and one gateway, we demonstrate that the proposed scheme can be implemented on real LoRaWAN development hardware and that only small modifications of the LoRaWAN layer are required.

Interesting future work is the implementation and evaluation with multiple gateways. As discussed in Section VII-E, this would allow to distribute the acknowledgments such that more packets can be acknowledged and the system still complies to the duty cycle limit. Another useful extension is the use of multiple frequency channels with an agreed pseudo random channel hopping sequence for the transmissions to make the schemes more resilient against narrow band interference.

ACKNOWLEDGEMENTS

We would like to thank Emanuele Bedeschi, Tommaso Polonelli, and Davide Brunelli for providing the framework which implements time synchronization on LoRaWAN hardware.

REFERENCES

[1] “LoRaWAN 1.1 Specification,” LoRa Alliance, 2017.

[2] F. Adelantado et al., “Understanding the limits of LoRaWAN,” IEEE Communications Magazine, vol. 55, no. 9, 2017, pp. 34–40.

[3] “EN 300 220-1 - V3.1.1,” European Telecommunications Standards Institute (ETSI), 2017.

[4] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, “A study of LoRa: Long range & low power networks for the internet of things,” Sensors, vol. 16, no. 9, 2016, p. 1466.

[5] B. Vejlgaard et al., “Interference impact on coverage and capacity for low power wide area IoT networks,” in Wireless Communications and Networking Conference (WCNC), 2017 IEEE. IEEE, 2017, pp. 1–6.

[6] É. Morin, M. Maman, R. Guizzetti, and A. Duda, “Comparison of the device lifetime in wireless networks for the internet of things,” IEEE Access, vol. 5, 2017, pp. 7097–7114.

[7] D.-Y. Kim and S. Kim, “Dual-channel medium access control of low power wide area networks considering traffic characteristics in IoE,” Cluster Computing, vol. 20, no. 3, 2017, pp. 2375–2384.

[8] K.-H. Phung, H. Tran, Q. Nguyen, T. T. Huong, and T.-L. Nguyen, “Analysis and assessment of LoRaWAN,” in Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom), 2018 2nd International Conference on. IEEE, 2018, pp. 241–246.

[9] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, “Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling,” IEEE Internet of Things Journal, 2018.

[10] T. Polonelli, D. Brunelli, and L. Benini, “Slotted ALOHA Overlay on LoRaWAN – a Distributed Synchronization Approach,” in IEEE International Conference on Embedded and Ubiquitous Computing (EUC 2018).

[11] C. Gu, R. Tan, X. Lou, and D. Niyato, “One-Hop Out-of-Band Control Planes for Low-Power Multi-Hop Wireless Networks,” arXiv preprint arXiv:1712.06056, 2017.

[12] “SX1276/77/78/79 Datasheet,” Semtech Corporation, 2017.

[13] “LoRaWAN 1.1 Regional Parameters,” LoRa Alliance, 2017.

[14] N. Abramson, “THE ALOHA SYSTEM: another alternative for computer communications,” in Proceedings of the November 17-19, 1970, fall joint computer conference. ACM, 1970, pp. 281–285.

[15] L. G. Roberts, “ALOHA packet system with and without slots and capture,” ACM SIGCOMM Computer Communication Review, vol. 5, no. 2, 1975, pp. 28–42.

[16] “LoRa Server, open-source LoRaWAN network-server,” https://www.loraserver.io/, [retrieved: September, 2018].

[17] “Docker,” https://www.docker.com/, [retrieved: September, 2018].

[18] “Eclipse Mosquitto,” https://mosquitto.org/, [retrieved: September, 2018].

[19] “Node-RED,” https://nodered.org/, [retrieved: September, 2018].

[20] “LoRa network ,” https://github.com/Lora-net, [retrieved: September, 2018].