Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling

Brecht Reynders, *Graduate Student Member, IEEE*, Qing Wang, *Member, IEEE*, Pere Tuset-Peiro, *Senior Member, IEEE*, Xavier Vilajosana, *Senior Member, IEEE*, and Sofie Pollin, *Senior Member, IEEE*

***Abstract*—Providing low power and long range (LoRa) connectivity is the goal of most Internet of Things networks, e.g., LoRa, but keeping communication reliable is challenging. LoRa networks are vulnerable to the capture effect. Cell-edge nodes have a high chance of losing packets due to collisions, especially when high spreading factors (SFs) are used that increase time on air. Moreover, LoRa networks face the problem of scalability when they connect thousands of nodes that access the shared channels randomly. In this paper, we propose a new MAC layer—RS-LoRa—to improve reliability and scalability of LoRa wide-area networks (LoRaWANs). The key innovation is a *two-step lightweight scheduling*: 1) a gateway schedules nodes in a coarse-grained manner through dynamically specifying the allowed transmission powers and SFs on each channel and 2) based on the coarse-grained scheduling information, a node determines its own transmission power, SF, and when and on which channel to transmit. Through the proposed lightweight scheduling, nodes are divided into different groups, and within each group, nodes use similar transmission power to alleviate the capture effect. The nodes are also guided to select different SFs to increase the network reliability and scalability. We have implemented RS-LoRa in NS-3 and evaluated its performance through extensive simulations. Our results demonstrate the benefit of RS-LoRa over the legacy LoRaWAN, in terms of packet error ratio, throughput, and fairness. For instance, in a single-cell scenario with 1000 nodes, RS-LoRa can reduce the packet error ratio of the legacy LoRaWAN by nearly 20%.**

***Index Terms*—Fairness, LoRa, LoRaWAN, NS-3, reliability, scalability, simulation.**

I. INTRODUCTION

WITH the evolutions of wireless communication, we can now embed tiny wireless transceivers into any "thing." These great advances are enabling us to build the Internet of Things (IoT). To realize IoT, many technologies have been proposed, such as narrowband (NB)-IoT [1], long range (LoRa) [2], Sigfox [3], among others. These technologies have the capability of connecting massive numbers of devices in large areas and target a lifetime of ten years simply powered by batteries.

Among these technologies, LoRa offers a compelling mix of long range and low power consumption data transmission [2]. It operates on the 868/915-MHz ISM bands that are available worldwide. It can support a communication range of 5 and 15 km in urban and suburban areas, respectively, comparable to the existing cellular networks. A single LoRa gateway can connect thousands of devices to the Internet. LoRa itself is a physical (PHY) layer technology built around the chirp spread spectrum modulation. There are several solutions for the MAC layer. Among them, LoRa wide-area network (LoRaWAN) [4] is the most popular one. Although LoRa&LoRaWAN[1] has attracted increasing attentions from both industry (Orange, BT, KPN, SK Telecom, etc.) and academia [5]–[8] during the past years, it still faces several challenges.

A. Challenge 1—Scalability

Scalability is of key importance in LoRaWAN due to its long range and thousands of devices may reach a given gateway simultaneously. However, LoRaWAN lacks such a scalability because it uses random access MAC protocol combined with a large time-bandwidth product [9]. Thus, the capture effect reduces greatly the performance of the nodes that are far away from the gateway. The collisions between the packets transmitted by nodes that are considerably close to the gateway and the packets from the nodes that are far away will always result in a loss of the packets transmitted by the nodes that are relatively further from the gateway [10]. In other words, connecting thousands of devices with LoRaWAN creates serious challenges for network scalability which is even more challenging due to the capture effect.

B. Challenge 2—Interference From Other Co-Located Networks

ISM bands are extensively used by other wireless technologies, both NB and broadband. These make coexistence mechanisms important. In that regard, a LoRaWAN gateway is supposed to reply to confirmed frames with acknowledgments. If the gateway cannot acknowledge a frame due to

Manuscript received December 1, 2017; revised February 8, 2018; accepted February 24, 2018. Date of publication March 13, 2018; date of current version June 8, 2018. This work was supported in part by the IWT CELTIC O&O Project ASUA/RoCCS, in part by the FWO SBO SAMURAI, and in part by the Spanish Ministry of Economy and the FEDER Regional Development Fund under SINERGIA Project TEC2015-71303-R. *(Corresponding authors: Qing Wang; Sofie Pollin.)*

B. Reynders, Q. Wang, and S. Pollin are with the Department of ESAT, KU Leuven, 3001 Leuven, Belgium (e-mail: brecht.reynders@kuleuven.be; qing.wang@kuleuven.be; sofie.pollin@kuleuven.be).

P. Tuset-Peiro and X. Vilajosana are with the Computer Science, Multimedia and Telecommunications Department, Universitat Oberta de Catalunya, 08016 Barcelona, Spain (e-mail: peretuset@uoc.edu; xvilajosana@uoc.edu).

Digital Object Identifier 10.1109/JIOT.2018.2815150

[1]In the rest of this paper, LoRa&LoRaWAN is referred as LoRaWAN, for simplicity.

duty cycle regulations or a lost acknowledgment due to external interference, nodes will retransmit their packets as long as it is deemed necessary. The standard behavior for frame retransmissions is to switch to more robust spreading factors (SFs), which increases the channel occupation and leads to an increase in collisions, deteriorating overall network reliability.

C. Challenge 3—Restrictions on Duty Cycle

Another challenge faced by LoRaWAN is the restrictions on the duty cycle. This restriction makes the resources of the downlink channel more valuable, *discouraging any fine-grained scheduling* of nodes performed by gateways. Fine-grained scheduling, as used in cellular networks, requires lots of downlink channel resources.

To tackle these challenges, we propose *RS-LoRa*, a novel MAC layer to improve reliability and scalability of LoRaWAN. We design a *distributed and two-step lightweight scheduling method* based on the received signal strength at both nodes and gateways. The *first-step* is performed by gateways. A gateway schedules nodes within its cell in a coarse-grained manner by dynamically specifying the allowed received signal strength and SFs for each channel. The *second-step* is performed by nodes. Based on the coarse-grained scheduling information provided by the gateway, nodes decide on their own the transmission power, SF, and on which channel and when to schedule their data transmission. Through this scheduling approach, the packet collisions in the network can be reduced greatly. Therefore, the reliability, scalability, and capture effect of the network can be improved largely.

The contributions of this paper are summarized as follows.

1) *Design:* We design the RS-LoRa, a new MAC protocol to improve the performance of LoRaWAN. The key component is the proposed distributed and two-step lightweight scheduling (Section III).
2) *Energy Efficiency Analysis:* We analyze the energy efficiency of RS-LoRa and legacy LoRaWAN. Our numerical results show that RS-LoRa can improve the energy efficiency of a network that has many nodes, even though RS-LoRa introduces additional cost on regularly listening to the channel (required by our lightweight scheduling) (Section IV).
3) *Implementation:* We implement RS-LoRa in NS-3 as well as other necessary components to compose a complete LoRa network. This complete implementation allows us to evaluate the performance of RS-LoRa extensively. For comparison, we also implement the legacy LoRaWAN in NS-3 (Section V).
4) *Evaluation:* We evaluate the performance of RS-LoRa in single-cell and multicell scenarios, with different number of nodes and traffic load. Our extensive simulations show that RS-LoRa can reduce the packet error rate (PER) of legacy LoRaWAN by 20% when there are many nodes. The results also demonstrate that RS-LoRa can improve the network scalability of legacy LoRaWAN (Section VI).

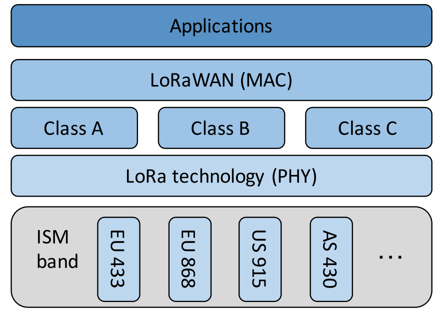

Fig. 1. Network stack of a normal LoRa node.

II. Background and Related Work

A. Background

The LoRaWAN network consists of two parts: 1) gateways and 2) nodes. Gateways are responsible for connecting all LoRaWAN nodes within their coverage. Furthermore, a gateway can communicate with several nodes simultaneously. These gateways are similar to the base stations in cellular networks. The network stack of LoRaWAN nodes is shown in Fig. 1. The PHY layer is the LoRa technology that operates on different ISM bands, depending on the regions where LoRaWAN nodes are deployed. Depending on the application scenario, LoRaWAN nodes can be configured into three different types.

1) *Class A:* A node wakes up and transmits its message to the gateway when a message is ready. After sending out this message, the node waits for a short period to receive its pending messages from the gateway. After this reception, the node goes back to sleep mode. This class consumes the least amount of energy.
2) *Class B:* Except for sending messages on demand as with class A, a node operating at class B is also scheduled to receive messages from the gateway.
3) *Class C:* With this configuration, a node keeps listening to the channel to check for messages from the gateway. Nodes operating at class C consume the most energy.

LoRaWAN uses *SFs* for rate adaptation. In the specification of LoRa, the SF can be selected from SF7 to SF12.[2] A lower SF leads to a higher transmission rate and shorter transmission time, but requires a higher SNR.

1) MAC Protocol: While LoRa communications are typically achieved by sacrificing bit rate or by robust modulation techniques at PHY, low power operation mainly depends on the MAC protocols. In addition, the MAC protocol also determines network scalability, organizes downlink/uplink transmissions, and node mobility. LoRaWAN uses a very simple MAC protocol: ALOHA, i.e., random access. When a node has a message to transmit, it selects an SF, channel, and transmission power, and transmits its message with these parameters. After its transmission, the node schedules two time slots for receiving downlink transmissions. The first slot is at least 1 s later than the end of the transmission, while the second slot is exactly 1 s after the first reception slot. The gateway can

[2]In this paper, the values for SF are the values used in the LoRa community rather than the typical definition of SF: chip rate over symbol rate.

select either one of them to transmit its downlink message to this node. If the gateway selects the first slot, it uses the same parameters as the node. Otherwise, more reliable parameters can be used. If there is no downlink message for that node, the gateway ignores these two slots.

2) Acknowledgments: The messages sent by nodes can be acknowledged by a gateway or not. In the former case, a gateway is obliged to acknowledge uplink messages. In the latter case, a gateway sends the acknowledgments at will. However, even without acknowledgments, gateways have to respond some received messages to confirm working connections between them and the nodes. When multiple gateways receive a message, one of them will respond that node.

3) MAC Commands: Gateways also send control information to nodes. This control information is called MAC commands in the LoRaWAN standard. An example of a MAC command is the *LinkAdrReq*. This MAC command requests a change in available channels, SFs, and transmission powers that should be used by the receiving nodes. Each requesting command needs to be acknowledged by the receiving nodes with the corresponding answer.

B. Related Work

Here, we summarize the state-of-the-art research that is most related to our proposed RS-LoRa.

Significant contributions have been done in our community investigating the scalability of LoRa networks, especially the scalability of LoRaWAN MAC protocol [5], [9]–[13]. They all conclude that *it is difficult to scale LoRaWAN networks.* Georgiou and Raza [9] and Haxhibeqiri *et al.* [11] studied the uplink traffic in LoRaWAN and show that the packet delivery ratio decreases exponentially with the number of nodes in the network. This finding is what we expect for ALOHA-based networks. In [9] and [11], only one SF is considered and the LoRa symbols are assumed to be orthogonal. However, different SFs do effect each other, as shown in [13]. Higher SFs even have a negative impact on network performance [10], where the authors also show that acknowledgments in LoRaWAN do not scale. After several retransmissions, nodes switch to higher SFs, but only increase their probability of collision. Furthermore, downlink traffic is studied in [12]. The authors find that the transmission of downlink messages can corrupt the uplink packets at the gateway. Therefore, downlink traffic, e.g., acknowledgments, should be sent with care. Motivated by the above state-of-the-art, in this paper, we propose the RS-LoRa to *improve the scalability of LoRaWANs* through a novel lightweight scheduling at gateways and a distributed self-scheduling at the nodes. To the best of our knowledge, RS-LoRa is the first MAC protocol built upon LoRaWAN MAC that targets at *improving* the latter's scalability.

An effort has been done in [7] to provide more reliability for nodes that are far from the gateway, i.e., solve the capture effect. This paper presented that when the gateway assigns the SFs, the network performance could be improved significantly over the legacy LoRaWAN where nodes pick their preferred SFs. According to the presented derivations, the ideal portion of traffic using one SF is the bit rate corresponding with that SF divided by the sum of the bit rates of all SFs. The power consumption in this scheme, however, is high when a node is assigned the highest SF.

The main reason for the low performance of LoRaWAN is the ALOHA access of the channel. Therefore, some authors proposed to completely synchronize the LoRa network [14]. Theoretically, it would significantly improve the network by assigning slots to each node (i.e., fine-grained scheduling). For large networks, such as LPWANs, it is very challenging to assign each node a slot without violating the duty cycle restrictions. So, low-power distributed queuing (LPDQ) [15] has been proposed for LPWANs [16]. It is a simple MAC protocol where nodes send small packets in contention slots to request channel access. If these requests arrive at a gateway without collisions, nodes are granted a slot to transmit their data packets, which is considerably longer. Applying LPDQ directly on LoRaWANs, however, is impractical. LoRa packets are typically small, yet vary a lot in size: from a few milliseconds to up to 2 s. Thus, defining slot boundaries creates a complex schedule or wastes a lot of resources waiting idle.

Another two approaches targeting at LoRa communication are Ingenu [17] and Weightless-SIG [18]. The former divides time into two slots in the 2.4-GHz ISM band, one for uplink and the other one for downlink. Nodes select a random subslot to transmit their data with a random phase. This approach, however, is impossible in the 868-MHz band as the duty cycle is considerably smaller. The Weightless protocol [18] exploits TDMA/FDMA and does work in the 868-MHz ISM band. First, the gateway sends a beacon including the scheduling of all the nodes. Then, the nodes transmit their data in the scheduled slots. Weightless uses FHSS to enable maximal duty cycle for the gateway. When using the full 868-MHz ISM band, this protocol can use up to 2400 s of downlink per hour, or a duty cycle of 67%. This is more than enough to fulfill the needs for long range communications. In LoRaWAN, however, bandwidths are wider, resulting in less potential channels over the entire band and a lower duty cycle.

Besides, numerous MAC protocols have been proposed for wireless sensor networks, as surveyed in [19]. The main goal of these MAC protocols is to provide low-power multihop communications between the sensor nodes and a central sink, while reducing the latency as much as possible. However, these protocols do not apply well for LPWANs. First, nodes in LPWANs are heterogeneous because a gateway is powered and has more capabilities than any low power nodes in the network. Second, unlike wireless sensor networks, in LPWAN there is no multihop communications because low-power nodes in LPWAN can communicate with a gateway directly. Finally, an LPWAN targets at connecting significantly more nodes than a wireless sensor network. A wireless sensor network only has a limited amount of peers in their network, so polling them on a regular basis is feasible. But polling devices within LPWANs will waste too much resources and thus should be avoided. The only feature in our proposed RS-LoRa that is relevant to wireless sensor networks is the synchronization. However, a considerable amount of downlink resources is required to realize this function properly. Also, a full synchronization consumes more power. In RS-LoRa, we avoid these requirements

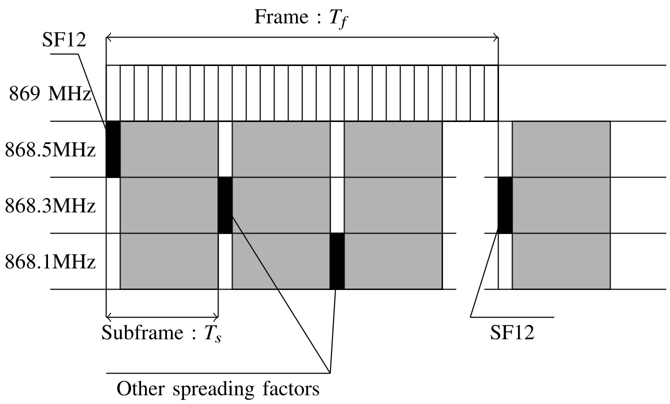

Fig. 2. Illustration of the channel assignment. ■ represents beacons and ▭ denotes uplink/downlink slots. The beacons from a gateway are transmitted at different channels alternately to increase reliability by reducing collisions with beacons from other neighboring gateways.

through a novel lightweight scheduling that will be presented in Section III.

III. System Design

In this section, we present the system model and design of our proposed RS-LoRa.

A. System Model

In RS-LoRa, the whole available bandwidth is divided into one *synchronous downlink* channel and several *asynchronous uplink/downlink channels*, as depicted in Fig. 2. We assume there are I asynchronous uplink/downlink channels. The set of the asynchronous channels is denoted as $\mathcal{I} = \{1, 2, \ldots, I\}$, *all structured in the unit of frames*. All the frames have the same structure and duration, each occupying T_f seconds. A frame is further divided into *subframes*, each lasting for T_s seconds. A subframe starts with a beacon transmitted by the gateway, followed by $\mathcal{I}$ uplink/downlink slots during which nodes can transmit their packets in an ALOHA manner. Each beacon carries information about how to use the uplink/downlink slots in that corresponding subframe. This information specifies the available channels and SFs that can be used on each channel. For the ith channel, we denote the set of the available SFs as $\mathcal{S}_i$. Nodes harness this *lightweight scheduling* information to further select the channel and SF for their data transmission. After that, a node selects a random time offset and transmit its packet at that offset. If a gateway successfully receives that packet, i.e., no collision occurs, it can choose whether or not to acknowledge that node, as in the legacy LoRaWAN.

Note that in our system, the asynchronous uplink/downlink channels are framed, but the access in each subframe is still ALOHA. Although a completely slotted approach can improve the network reliability, we choose to keep the access within each subframe *unslotted* as it allows low power consumption for nodes that have a very low demand on throughput. Such nodes can sleep for a long time as synchronization is not critical. Then, upon wake up, these nodes only need to deal with possible de-synchronization such that their transmissions do not overlap with any beacon.

B. Gateway

In RS-LoRa, the proposed lightweight scheduling is coordinated by gateways. In this section, we first present the beacons, followed by the first-step of the lightweight scheduling and the packet structure of beacons.

1) Beacons: A beacon has two purposes: a) synchronizing the nodes and b) carrying downlink information. Nodes listen to the latest beacon once they wake up and then adapt their clocks to the gateway. Beacons can also carry small data messages, e.g., MAC commands, to reduce the amount of downlink messages.

In RS-LoRa, beacons from each gateway are transmitted on different channel to increase reliability against both collisions with beacons from other gateways and interference from other networks operating at the same frequency. Each new beacon is transmitted on the next channel, iterating over all available channels. Furthermore, to increase the reception probability of beacons, *different SFs* are applied to transmit beacons. The first beacon of each frame is sent using SF12, to ensure that it is received by all nodes, even by the nodes at the cell edge. Other beacons can be transmitted using different SFs.

However, increasing the number of SFs to transmit beacons also increases the time spent by nodes to join the network. To have a fast network joining process in channel hopping networks, Vogli *et al.* [20] proposed to let nodes rebroadcast beacons. These rebroadcasts can be picked up faster since they occur more frequently. However, it is costly for low power networks to rebroadcast these beacons. In RS-LoRa, nodes follow the joining process specified in the legacy LoRaWAN. The gateway sends the required information about the schedule and the next beacon upon each accepted joining request. Extra information can be added to the downlink messages for nodes to quickly (re)synchronize with the network.

2) Lightweight Scheduling at the Gateway (the First-Step): The gateway uses lightweight scheduling to coordinate uplink transmissions from nodes. Instead of scheduling each node with the concrete transmission power, SF, and channel, our proposed lightweight scheduling method *schedules each channel by specifying its allowed transmission powers and SFs for each subframe.* Then, nodes select their own transmission parameters based on this high-level scheduling information.

a) Transmission power: The gateway specifies the allowed transmission power of nodes by restricting its received signal strength (RSS) of the uplink traffic. That is, each beacon from the gateway carries information about the allowed RSS of each channel for that subframe. Through this, the gateway can control the transmission power of each node to some extent, and has the capability to minimize the network *capture effect.* The capture effect needs to be minimized for those nodes that are far from the gateway. Otherwise, their uplink traffic can easily collide with the traffic that is transmitted with any SFs from nodes near the gateway. We can also use the allowed RSS values in the beacons to control the traffic of a specific channel. By modifying the thresholds of RSS, we can increase/decrease the amount of traffic, resulting in lower/higher reliability for that specific channel. Thus, the gateway can also dedicate a channel to the nodes that are very far from the gateway by allowing a low RSS on a specific

TABLE I
STRUCTURE OF BEACONS (I IS THE NUMBER OF THE USED CHANNELS AND Y REPRESENTS THE LIST OF CARRIED MAC COMMANDS)

Fields	Version	PacketType	GatewayID	FrameID	SubFrameID	Length	ListSF	ListRSS	LengthofMAC	MAC commands
Bits	0:1	2:5	8:23	24:31	32:39	40:47	$6 \times I$	$10 \times I$	8 bits	Y

channel, in order to improve the quality of service of these nodes.

b) Spreading factors: Determining the allowed SFs for each channel is important. The more allowed SFs, the higher achieved channel reliability as packets with different SFs can be received simultaneously. The reason is that less collisions occur with more used SFs. However, more SFs could result in longer messages and more energy consumption. Therefore, the reliability and the energy consumption should be balanced. In [7], we have shown that the lowest packet error ratio can be achieved when p_s, the probability of using an SF s, equals: $p_s = [R_s/(\sum_{s' \in \mathcal{S}_i} R_{s'})]$, where R_s and $R_{s'}$ denote the achieved data rates with the SFs s and s', respectively. Let η_d be the ratio between P^d_{legacy} and P^d_{RS}, where the latter two denote the packet delivery ratios under the legacy LoRaWAN (only with the SF7) and the RS-LoRa (a set of selected SFs), respectively. Then η_d can be written as

$$\eta_d = P^d_{\text{legacy}}/P^d_{\text{RS}} = e^{2\lambda(1-p_{s7})}. \tag{1}$$

Notice that η_d is bigger than 1. So, the network performance improves with more SFs, i.e., when p_{s7} is smaller. However, the average energy consumption on transmission, denoted as E, also increases

$$E = \sum_{s \in \mathcal{S}_i} p_s P_t \frac{L}{R_s} = \frac{|\mathcal{S}_i| P_t L}{\sum_{s \in \mathcal{S}_i} R_s} \tag{2}$$

where $|\mathcal{S}_i|$ denotes the cardinality of $\mathcal{S}_i$ or, in other words, the number of allowed SFs on channel i, P_t is the transmission power, and L the packet length. Equation (2) shows that the average energy consumption increases with more and especially higher SFs. The numerator increases linearly with the number of SFs, while the denominator scales significantly slower. As a result, η_e, which denotes the ratio of the energy consumptions under LoRaWAN and RS-LoRa, can be written as

$$\eta_e = \frac{P_t L/R_{s7}}{E} = \frac{\sum_{s \in \mathcal{S}_i} R_s}{|\mathcal{S}_i| R_{s7}}. \tag{3}$$

As the denominator increases faster with the number of SFs, η_e is smaller than one. From (1) and (3), the gateway determines the allowed SFs for each channel to satisfy the current network preferences. That is a tradeoff between the energy consumption and the network reliability. The energy consumption is further investigated in detail in Section IV.

3) Structure of Beacons: The packet structure of beacons is shown in Table I. The first field represents the version of the protocol, which occupies 2 bits. Next field, the 4-bit *PacketType*, specifies the type of the current packet. The next field is *GatewayID*. It provides spatial information to the nodes and allows the nodes to synchronize with different gateways, especially for mobile nodes. Next, the fields *FrameID* and *SubFrameID*, give nodes the information about the timing in the network. Their values are updated in each new beacon. The length of current packet is specified in field *Length* that takes 8 bits. The following two fields, *ListSF* and *ListRSS*, carry information about the lightweight scheduling of the current subframe. They specify the allowed SFs and RSSs for each asynchronous channel. The remaining two fields list the carried MAC commands that are reused from LoRaWAN.

C. Nodes

In LoRaWAN, all the nodes are allowed to choose any SF and transmission power freely. Therefore, nodes are eager to select the lowest possible SF because it brings the shortest time-of-flight for transmitting a packet, and thus, minimizes their power consumption. However, the probability of packet collision increases greatly when all nodes use the same SF, especially with a large number of nodes in the network close to the gateway. In RS-LoRa, we propose that *each node* uses the information carried by beacons to determine its transmission parameters, namely the channel, SF, and transmission power. This is the second-step of the proposed lightweight scheduling that is performed at each node in a distributed manner. By doing this, RS-LoRa can alleviate the capture effect and reduce the probability of packet collisions. Algorithm 1 presents the details on determining transmission parameters at each node.

The algorithm works as follows. When a node n needs to transmit a message, it first looks for an available channel. It can acquire this necessary information by listening to the latest beacon. Node n then determines which channel to use (lines 3–9). To select the most suited channel, the algorithm searches over all the available channels and then selects the one that has the highest target RSS which is lower than the estimated RSS from this node to the gateway. This estimation can be derived from the RSS of the beacons from the gateway.

If a channel is found, a random SF is selected (line 11). Note that a higher SF consumes more time on air and should be used less, and vice versa. Therefore, in RS-LoRa, *different SFs are selected with different probabilities.* Let $\mathcal{S}_i$ denote the set of allowed SFs for the channel i (carried in the beacons). Then the probability of selecting an SF of s is defined as follows:

$$p_s = \frac{R_s}{\sum_{s' \in \mathcal{S}_i} R_{s'}} \quad \forall s \in \mathcal{S}_i \tag{4}$$

where R_s and $R_{s'}$ are the bit rate achieved with the SF s and s', respectively, and we have $\sum_{s \in \mathcal{S}_i} p_s = 1$.

Based on the determined SF and target RSS at the gateway, the node calculates the required transmission power (line 12). An increase of SF by 1 brings an increase of 2.5 dB in the sensitivity of the gateway, according to the datasheet of LoRa transceiver [21]. Thus, the transmission power can be reduced when a higher SF is used.

When no channel is determined (line 13), it implies that node n is very close to the gateway. In this case, the lowest

Algorithm 1 Determine Transmission Parameters at Each Node

Input: P_b: RSS of the received beacon at node n;
$\mathcal{I}$: channel set; T_s: duration of each subframe;
$\mathcal{S}_i$: set of allowed SF for channel i, $\forall i \in \mathcal{I}$;
P'_i: target uplink RSS for channel i, $\forall i \in \mathcal{I}$;
Output: Transmission parameters: C_n: selected channel;
S_n: selected SF; P_n: transmission power; T_n: offset time.

```
1:  P_tmp ← 0                         # P_tmp: a temporary variable
2:  Flag ← false    # Flag: denoting if a channel is selected
3:  for i ∈ I do
4:    if P_tmp < P'_i < P_b then
5:      C_n ← i                              # Select channel
6:      P_tmp ← P'_i
7:      Flag ← true
8:    end if
9:  end for
10: if Flag = true then
11:   S_n ← randomly choose a SF from S_{C_n} with different
      probabilities, refer to Eq. (4)          # Select SF
12:   P_n ← P_tmp + P_pathloss − 2.5S_n + P_offset    # Calculate
      transmission power
13: else
14:   S_n ← 7                        # Select the lowest SF
15:   P_n ← 0 dBm
16:   C_n ← arg max_{i∈I} P'_i
17: end if
18: T_p ← time-of-flight of the packet with selected SF S_n
19: T_n ← rand(0, T_s − T_p) # Select an offset time randomly
```

SF is selected because it has the minimal power consumption and suffices to reduce the amount of destructive interference (lines 14–16).

Finally, the node selects a time for transmission (lines 18 and 19). There are no requirements for this value except that the message cannot interfere with the beacons. Even when this node would use a channel that is not used to transmit the next beacon by its own gateway, the transmitted message could still collide with beacons from neighboring gateways.

IV. Energy Consumption Analysis

An important property of LPWANs is low power consumption. While RS-LoRa improves reliability and scalability of the legacy LoRaWANs through lightweight scheduling, it introduces additional energy consumption. In this section, we analyze the energy consumptions for both legacy LoRaWANs and our proposed RS-LoRa, and will show from numerical analysis that the additional energy does not necessarily result in a worse energy efficiency. Without loss of generality, we assume that all packets sent by nodes have the same length. We make the following definitions.

1) *Packet Arrival Rate:* Denoted as λ. It represents the traffic load of the network. It is defined as the average number of packets that arrives to the network within a period T, where T is the period on transmitting a packet with SF7.
2) *Energy Consumption on Transmitting Each Packet:* Denoted as E_d. It is the energy consumption for transmitting a packet in both legacy LoRaWAN and RS-LoRa with SF7.
3) *Energy Efficiency:* The energy efficiency is defined as the amount of information arriving at a gateway divided by the energy to transmit that same information. In other words, the energy efficiency is the packet delivery ratio over the average energy cost of transmitting one packet.

We distinguish between two type of transmissions: *acknowledged transmission* and *unacknowledged transmission*. In networks where ACKs are used for acknowledgment (acknowledged transmission), the energy consumption is calculated by including retransmissions of each packet. In a network with unacknowledged transmissions, each message is repeated once.

A. Legacy LoRaWANs

In legacy LoRaWANs, the energy consumption at each node is mainly spent on the transmission of packets. Thus, ALOHA is most energy efficient since no overhead is introduced on listening to the channels.

1) Unacknowledged Transmission: To calculate the energy efficiency, it is enough to consider the packet delivery ratio. When λ increases (more traffic in the network), the amount of collisions also increases. Let p denote the probability of transmitting a packet successfully. It can be written as follows:

$$p = e^{-2\lambda}. \tag{5}$$

The packets that are successfully transmitted consume the default amount of energy, i.e., E_d. Then the overall energy efficiency in unacknowledged legacy LoRaWAN, denoted as $\eta^{\text{un}}_{\text{legacy}}$, can be written as follows:

$$\eta^{\text{un}}_{\text{legacy}} = \frac{p}{E_d} = \frac{e^{-2\lambda}}{E_d}. \tag{6}$$

Equation (6) shows that the energy efficiency in unacknowledged transmissions depends exponentially on the packet arrival rate in the network.

2) Acknowledged Transmission: The energy consumption with the acknowledged transmission is more complex. In the LoRaWAN standard, up to eight retransmissions are allowed. In our analysis, we treat the retransmitted packets as "new" packets, meaning that we will have a higher new packet arrival rate. Let λ' represent this new packet arrival rate, then

$$\lambda' = \lambda \frac{1 - \left(1 - p'\right)^8}{p'} = \lambda \frac{1 - \left(1 - e^{-2\lambda'}\right)^8}{e^{-2\lambda'}} \tag{7}$$

where p' is the probability of a successful packet transmission in the network with acknowledged transmissions. To calculate the new packet arrival rate, this equation needs to be iterated a few times, as more traffic generates more collisions and as a result, even more traffic. Let $\eta^{\text{ack}}_{\text{legacy}}$ denote the overall energy efficiency with the acknowledged transmissions. To derive $\eta^{\text{ack}}_{\text{legacy}}$, the packet delivery ratio has to be divided by the energy consumption. This energy consumption can be estimated by $(\lambda'/\lambda)E_d$. The final energy efficiency can be calculated as

$$\eta_{\text{legacy}}^{\text{ack}} = \frac{1-(1-p')^8}{\frac{\lambda'}{\lambda}E_d} = \frac{e^{-2\lambda'}}{E_d}. \tag{8}$$

Note that the LoRaWAN standard suggests to use more reliable SFs when a packet is retransmitted, resulting in higher energy consumption. Here, we do not take this into account.

B. Proposed RS-LoRa

Similarly, the overall energy efficiency in RS-LoRa can be calculated. However, there are some differences. First, in RS-LoRa, the allowed transmission power and SFs are controlled in a coarse-grained manner by the proposed lightweight scheduling. Since RS-LoRa controls the transmission power and use more SFs, the transmissions can only collide with those using the same SF. As less nodes use the same SF in RS-LoRa, the resulted packet delivery ratio will increase. Second, since we use different SFs, transmissions with higher SFs consume more time and energy. Besides, nodes need to listen to the beacons when they have packets to send, which brings additional energy consumption.

The packet delivery probability can be calculated as follows:

$$p_{\text{rs}} = e^{-2\lambda p_{s7}} \tag{9}$$

where p_{s7} is the percentage of packets that use SF7, as shown in (4). The packet delivery probability is not calculated for other SFs as they are distributed to ensure that all the packets with different SFs have the same probability of reception [10]. Note that p_{rs} changes the arrival rate under acknowledgments, which can be written as follows:

$$\lambda'_{\text{rs}} = \lambda\frac{1-(1-p'_{\text{rs}})^8}{p'_{\text{rs}}} = \lambda\frac{1-\left(1-e^{-2\lambda'_{\text{rs}}p_{s7}}\right)^8}{e^{-2\lambda'_{\text{rs}}p_{s7}}}. \tag{10}$$

Equation (10) is similar to (7), except that the packet delivery ratio is adapted with the new packet delivery ratio p'_{rs}. More SFs also increase the average energy consumption of a packet. Thus, the new average energy consumption, denoted as $E_{d,\text{rs}}$, has to be weighted with the corresponding data rates

$$E_{d,\text{rs}} = \sum_{s\in\mathcal{S}_i} P_s\frac{E_d R_{b,s7}}{R_{b,s}} = \frac{|\mathcal{S}_i|E_d R_{b,s7}}{\sum_{s\in\mathcal{S}_i} R_{b,s}} \tag{11}$$

where $R_{b,s}$ is the data rate achieved with SF s, $s \in \mathcal{S}$ and $R_{b,s7}$ is the datarate achieved with SF 7. From (11), it is clear that the average energy consumption on transmitting each packet in RS-LoRa is higher than that in legacy LoRaWAN. However, it should be noted that in RS-LoRa the allowed SFs are determined by the gateway based on the network preference. The gateway can tradeoff between network reliability and energy consumption (see Section III-B). Thus, the energy consumption in RS-LoRA could be reduced by only including SF7.

Furthermore, E_b is the energy consumption for listening for beacons and synchronizing to the beacon structure. This additional energy consumption is added directly to the average energy consumption of transmitting a packet. The beacon is assumed to be received without problem such that only one beacon per transmission is required. Finally, we can derive the energy efficiencies for unacknowledged and acknowledged transmissions as

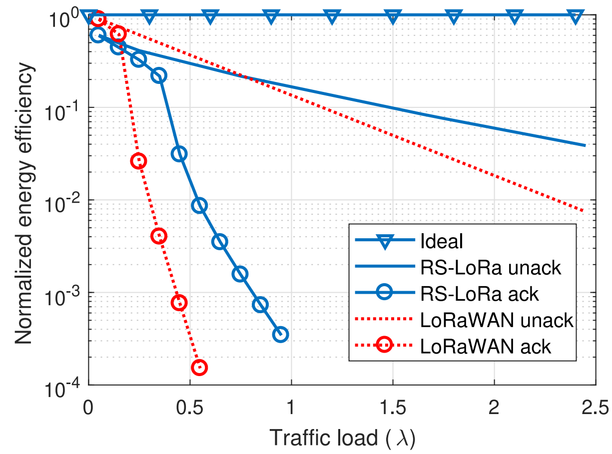

Fig. 3. Energy efficiency of the uplink transmissions under different traffic load.

$$\eta_{\text{rs}}^{\text{un}} = \frac{e^{-2\lambda p_{s7}}}{E_b + E_{d,\text{rs}}} \tag{12}$$

$$\eta_{\text{rs}}^{\text{ack}} = \frac{e^{-2\lambda'}}{E_b + E_{d,\text{rs}}}. \tag{13}$$

C. Numerical Results

The numerical results of the energy efficiencies under legacy LoRaWAN and our proposed RS-LoRa are shown in Fig. 3. The results are normalized to the energy efficiency of the *ideal* scenario where all the packets are transmitted only once, i.e., without any collisions. And also the energy consumption for listening to beacons E_b is assumed to be equal to half the energy consumption for transmitting one packet. This assumption is valid since listening consumes a lot less power and beacon messages tend to be smaller than actual data packets. From Fig. 3, it is clear that the acknowledged traffic is not energy scalable, that is, the energy consumption increases quickly with the increase of traffic load. Also, notice that there is a bend in the curve. This bend is the point where the acknowledged traffic cannot reliably transmit the traffic anymore. From there, retransmissions are just increasing the traffic on the channel, and not the reliability. We can also observe that the RS-LoRa consumes more energy when the traffic loads are small, but the gap becomes narrower with the increase of traffic load. With higher traffic loads, the energy efficiency of RS-LoRa is even better than that of the legacy LoRaWAN. This benefit is a result of the collision reduction of RS-LoRa. More results from our simulations will be presented in Section VI. To conclude, we state that the energy overhead of RS-LoRa is small, and the reduced collision cost leads overall to better energy-scalability.

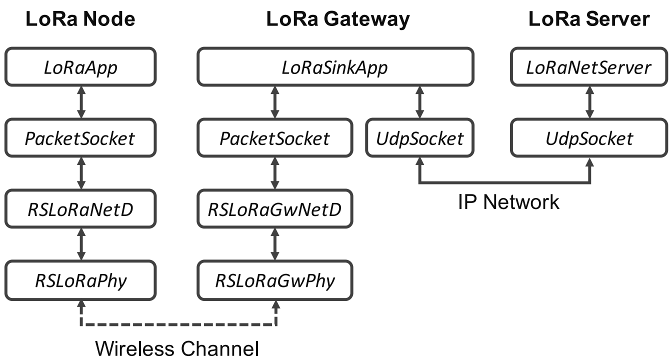

Fig. 4. Implementation architecture of RS-LoRa in NS-3.

V. Implementation in NS-3

In this section, we present the implementation of RS-LoRa in NS-3.[3] The implementation architecture is shown in Fig. 4. Four layers are present: 1) *PHY layer*; 2) *data link layer*; 3) *transport layer*; and 4) *application layer.* Next, we present them in detail. The implementation details of the transport layer is omitted due to its similarities to other types of communication.

A. Physical Layer

Two different PHY layers have been implemented: 1) PHY for nodes and 2) PHY for gateways.

1) PHY for Nodes (RSLoRaPhy): At the nodes, *RSLoRaPhy* keeps track of the noise on all channels and records which SFs have been used. It also checks if similar LoRa patterns are being used, for example, the chirp rate of a signal with SF10 and bandwidth 125 kHz is identical to the chirp rate of a signal with SF11 and bandwidth 250 kHz. Whenever a new packet arrives, *RSLoRaPhy* calculates the signal to noise ratio of this packet. With each new interfering signal that arrives at this PHY layer, the bit errors are calculated up to that point using the formulas presented in [8]. When a packet is successfully received from the channel, it will be forwarded to the data link layer for further processing. On the other direction, when a packet is ready for transmission (forwarded from the data link layer to *RSLoRaPhy*), *RSLoRaPhy* will immediately try to send it. When this transmission fails for any reason, the data link layer will be notified.

2) PHY for Gateways (RSLoRaGwPhy): A gateway is able to *receive multiple packets simultaneously*. Thus, a different PHY layer is required for gateways. However, since all the parameters are identical to those used in the PHY layer for nodes, we implement *RSLoRaGwPhy* based on the *RSLoRaPhy* by adapting the latter to the specific requirements for gateways. The main difference between them falls into the receiving part. Unlike *RSLoRaPhy* for the nodes, *RSLoRaGwPhy* keeps a list of all incoming signals and tracks the bit errors for each signal individually. All incoming signals contribute to the noise for any other incoming signal. Furthermore, when a transmission is initiated by the gateway (downlink), this self-interfering signal is also added to the noise sources because we assume that the gateways do not have full-duplex capabilities. In our implementation, a gateway can track up to eight simultaneous signals. This threshold is the maximal number of signals that can be processed simultaneously by a gateway [21]. However, we also check if a signal is being received correctly. If not, we drop this signal and thus another additional new incoming signal can be tracked.

[3]The code can be found online. [Online]. Available: https://github.com/networkedsystems/lora-ns3

B. Data Link Layer

Two different data link layers are implemented: one for nodes and the other one for gateways.

1) Data Link Layer for Nodes (RSLoRaNetD): At this layer, a node first needs to receive beacon messages. The RSSs of these beacons are stored and averaged within a predefined window to estimate the closest gateway. Then *RSLoRaNetD* parses the content of the beacon from that gateway. In our current implementation, a node listens to all beacons. If no data is waiting to be transmitted, the node will sleep until the next beacon. Otherwise, *RSLoRaNetD* executes Algorithm 1 to determine channel, SF, transmission power, and offset time for transmission. Then it wakes up at the offset time and starts transmitting with the determined parameters. Next, two receiving slots are reserved. *RSLoRaNetD* will listen to these slots for an acknowledgment. If MAC commands are carried in this acknowledgment, then they will be executed by *RSLoRaNetD*. When a receiving slot overlaps with a beacon message, that slot will be discarded. If the gateway does not acknowledge a message, *RSLoRaNetD* will keep transmitting that message. This is different from legacy LoRaWAN where the SF will be increased incrementally. It is up to the application layer to detect any failure in communication and act upon it.

2) Data Link Layer for Gateways (RSLoRaGwNetD): While *RSLoRaNetD* is similar to ALOHA, the *RSLoRaGwNetD* for gateways follows a strictly slotted approach. *RSLoRaGwNetD* invokes *RSLoRaGwPhy* to listen to the channel all the time. When an uplink message is received, *RSLoRaGwNetD* reserves two slots for a potential downlink transmission. The message from the node, including some transmission parameters, is forwarded to the upper layers for further processing. Then *RSLoRaGwNetD* starts a timer and waits for the decision from the upper layers. When the timer expires and acknowledgments or other data messages have arrived for the scheduled node, *RSLoRaGwNetD* first checks if there are any incoming messages on the current channel or any outgoing packets. If the answer is negative, the message is transmitted. If another message is currently being received, *RSLoRaGwNetD* will try to avoid demolishing the incoming message by waiting for 1 s and then transmitting the acknowledgment/message in the next reception slot. Note that beacons will be scheduled irrespective of the amount of uplink messages on the channel. LoRa MAC commands are also added to the packets in this layer. Currently, only the *LinkAdrReq* and *LinkAdrAns* MAC commands are implemented. However, a base class is provided that allows easy implementations of other MAC commands.

C. Application Layer

We also implement several applications for different objects: *LoRaApp* for the nodes, *LoRaSinkApp* for the gateways, and *LoRaNetServer* for the server that controls the whole network.

1) LoRaApp: This application is a dummy application that creates configurable data and sends it to *LoRaNetServer*. It has a configurable port to simulate different types of traffic. We use this application at each node to simulate its behaviors.

2) LoRaSinkApp: This application runs at each gateway and implements two functions.

1) Forward data received from the nodes to the backend server.
2) Forward data from the backend server to the nodes.

LoRaSinkApp acts as the bridge between nodes and the backend server. Therefore, it acts as a translator between LoRaWAN messages and the common Internet traffic.

3) LoRaNetServer: This is the most intelligent application that acts as the backend of RS-LoRa. For the packet transmitted by a node, *LoRaNetServer* could receive the duplicated packets from several gateways and selects the most suitable gateway to acknowledge the corresponding node. It runs several services, which are called apps here. Examples include the power and SF allocation, as well as storing and forwarding of data to the correct remote destination. It is important to mention that *LoRaNetServer* collects information about all packets. This includes the RSSs of the packets at each gateway and their time of arrival. These information will be used for the future lightweight scheduling.

For comparison, we also implement the legacy LoRaWAN in NS-3. The PHY layer and application layer are identical to what we implement for the proposed RS-LoRa. The data link layer is different in the sense that no beacons are transmitted and there is no lightweight scheduling. Only recently, a similar implementation of legacy LoRaWAN is reported in [22]. In this paper, we implement both RS-LoRa and legacy LoRaWAN from scratch. The implementation details of legacy LoRaWAN are omitted in this paper.

VI. Performance Evaluation

In this section, we first present the simulation setup and then present the performance evaluations under different scenarios.

A. Simulation Setup

In total, four different scenarios are evaluated. These scenarios are discriminated by different number of gateways and MAC protocols. The number of gateways is set to either one or seven. The locations of the gateways are shown in Fig. 5.

1) *Single-Cell Scenarios:* We only use one gateway, i.e., only enable the gateway GW_1. The investigated protocols are the legacy LoRaWAN and our proposed RS-LoRa.
2) *Multicell Scenarios:* We use seven gateways, i.e., enable all the gateways shown in Fig. 5 and increase the radius. The investigated protocols are LoRaWAN and RS-LoRa.

The gateways are located at a height of 30 m, while the nodes are placed one meter above the ground. Nodes transmit a packet to the network server (via gateways) every two minutes.

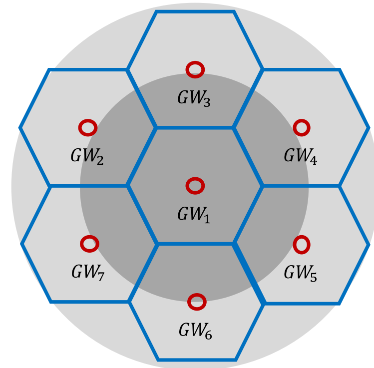

Fig. 5. Location of the gateways in the simulation. The main gateway (GW_1) is placed in the middle, used in both *single-cell scenarios* and *multicell scenarios*. The six surrounding gateways are only used in *multicell scenarios*. The gray circles mark the regions where the nodes are randomly distributed, where "dark-gray" and light-gray mark the regions for single-cell and multicell scenarios, respectively.

TABLE II
Parameters Setup in the Simulations

Parameter	Value	Unit
Number of nodes	100 to 3500	/
Packet length (excluding header)	51	bytes
T_f	10	minute
T_s	1	minute
Maximal distance to the GW_1	1000 or 1500	meter
Distance between GWs	1000	meter
Average inter-packet interval	120	second

The nodes are distributed in the gray area (see Fig. 5). The locations of the nodes follow Poisson distribution.[4] We use the Okunura–Hata model for urban areas together with Rayleigh fading (Nakagamim model with default parameters) as the path loss model. These models bring a high path loss. Therefore, small LoRa cells are expected, being consistent to the reality in urban areas where it is more likely to have small cells due to the dense existence of LoRa gateways. Table II lists the setup of other important parameters in the simulation.

Next, we present the details of the simulation results.

B. Single-Cell Scenarios

We first present the results under the single-cell scenarios.

1) Reliability and Scalability: We use PER to demonstrate the performance of reliability and scalability. The evaluation results of the PER under RS-LoRa and legacy LoRaWAN are shown in Fig. 6. The simulation creates different number of nodes (100, 500, 1000) for each scenario. From the results we can first observe that at most of the distances, the PERs under RS-LoRa are lower than those under LoRaWAN, especially at cell edge. For example, when 1000 nodes are randomly distributed in the network, RS-LoRa can reduce the PER from 43% to 29% at the cell edge. The reasons behind this are as follows.

[4]Other distributions have also been considered, and the results are similar.

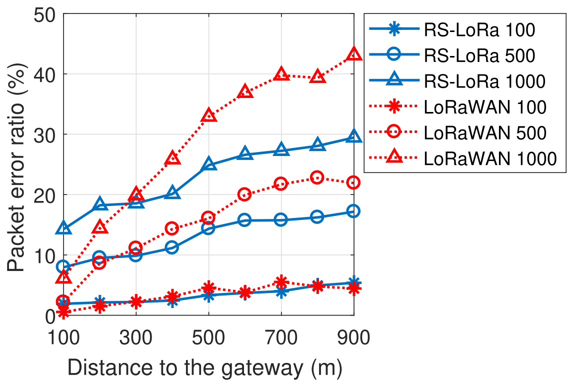

Fig. 6. Evaluation results (packet error ratio) under the single-cell scenarios. "RS-LoRa X" represents the scenario where there are X nodes in the network and the used MAC protocol is RS-LoRa.

TABLE III
AVERAGE PERs UNDER RS-LoRa AND LoRaWAN

No. of Nodes	PER w/ LoRaWAN	PER w/ RS-LoRa	Gain
100	3.4%	3.3%	2.9%
500	15.4%	13.1%	14.9%
1000	28.7%	23.0%	19.8%

1) The lightweight scheduling in RS-LoRa can reduce the capture effect by assigning different channels for nodes that are at different distances from the gateway.
2) Nodes randomly select an SF from the allowed SFs to reduce potential packet collisions.

This decrease of PER increases network reliability, which can further improve the network scalability (i.e., given the same threshold on PER, RS-LoRa can serve more nodes than legacy LoRaWAN).

With less nodes in the network, for example 100 nodes, the PER decrease achieved by RS-LoRa over LoRaWAN is smaller. This smaller decrease is due to the fact that packet collisions occur less in smaller networks (i.e., the probability of having concurrent transmissions is low). Also, for nodes that are close to the gateway, they suffer from a higher, albeit small, packet error ratio in RS-LoRa. They see in an increase of collisions since all nodes close to the gateway are using the same channel, while in LoRaWAN, they could choose from three different channels. Even though, our proposed RS-LoRa can improve the overall network performance largely, compared to legacy LoRaWAN. This claim can be supported by Table III where we show the average PERs of the network under RS-LoRa and LoRaWAN. We can observe that our RS-LoRa can improve the PER by 19.8% on average when there are 1000 nodes, demonstrating RS-LoRa's advantage of improving the network reliability and scalability.

2) Throughput: We report the overall throughputs achieved with RS-LoRa and LoRaWAN under different number of nodes. The results are shown in Fig. 7. We observe that with RS-LoRa, the network throughput can be increased from 2.2 to 2.5 Kb/s when there are 1000 nodes in the network. This is because that RS-LoRa improves the network reliability. The gain decreases when there are less nodes in the network.

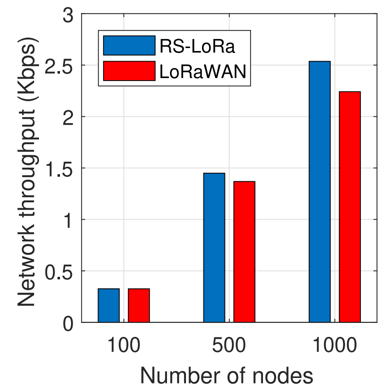

Fig. 7. Evaluation results (throughput) under the single-cell scenarios.

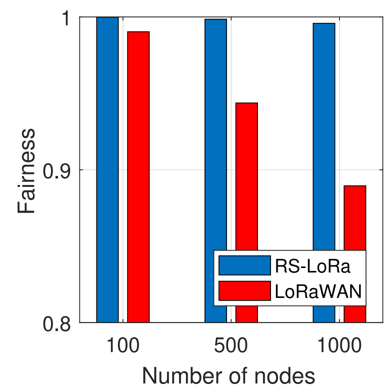

Fig. 8. Evaluation results (fairness) under the single-cell scenarios.

3) Fairness: As presented in Section III, RS-LoRa can reduce the capture effect by the proposed lightweight scheduling. In the simulation, we also exploit this aspect by investigating the fairness among nodes. The results are shown in Fig. 8. Again, we observe that when there are a large number of nodes in the network (e.g., 1000 nodes), we can improve the network performance by increasing the fairness from 89% to 99.6%, compared to the legacy LoRaWAN. This gain also decreases when there are less nodes in the network.

C. Multicell Scenarios

We continue to present the results under multicell scenarios. We consider the seven gateways as depicted in Fig. 5. Two different number of nodes are simulated: 1000 and 3500 nodes. Under each scenario, the nodes are randomly distributed in the "light-gray" area of Fig. 5.

1) Reliability and Scalability: We also use PER as the metric to demonstrate the system reliability and scalability. Evaluation results are shown in Fig. 9. First, we can observe that when there are more nodes in the network, i.e., 3500 nodes, RS-LoRa outperforms LoRaWAN at most of the distances. The gain is large especially at the cell edge, for example, the PER can be reduced from 42% to 20% by RS-LoRa when the nodes are 1400 m away from GW_1. The reason behind this has been explained in Section VI-B. Second, the PER decreases when nodes are around 900 m away from GW_1, when the nodes are near gateways $GW_2, GW_3, \ldots, GW_7$.

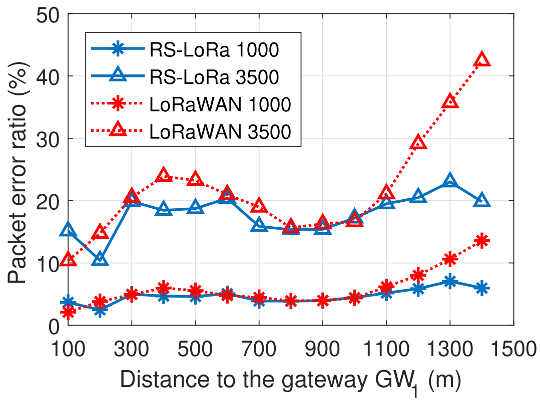

Fig. 9. Evaluation results (packet error ratio) under the multicell scenarios. RS-LoRa X denotes the scenario where there are X nodes in the network and the used MAC protocol is RS-LoRa.

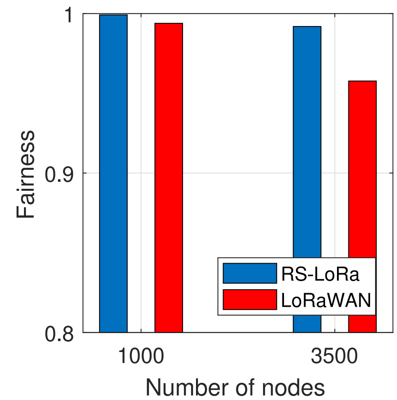

Fig. 11. Evaluation results (fairness) under the multicell scenarios.

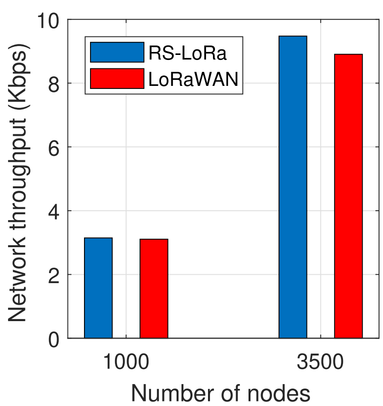

Fig. 10. Evaluation results (throughput) under the multicell scenarios.

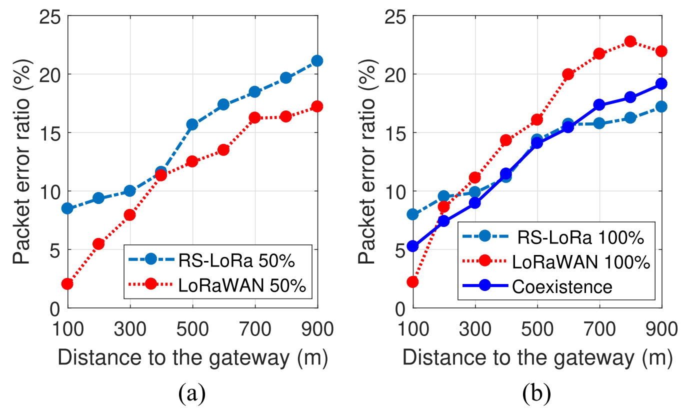

Fig. 12. PER in a network where both RS-LoRa nodes and legacy LoRaWAN nodes communicate with the same gateway (single-cell). (a) Coexistence in one network ("RS-LoRa 50%" and "LoRaWAN 50%," respectively, represent that 50% of the total nodes run the RS-LoRa protocol and the LoRaWAN protocol). (b) Comparison with: 1) all nodes run RS-LoRa ("RS-LoRa 100%") and 2) all nodes run LoRaWAN (LoRaWAN 100%). "Coexistence": (average PER of) RS-LoRa 50% + LoRaWAN 50%.

TABLE IV
AVERAGE PERs UNDER RS-LoRa AND LoRaWAN

No. of Nodes	PER w/ LoRaWAN	PER w/ RS-LoRa	Gain
1000	6.8%	5.6%	17.7%
3500	23.7%	18.8%	20.7%

It should be noticed that LoRaWAN performs better than RS-LoRa when nodes are very close to the gateways, so in between the gateways RS-LoRa performs better.

When there are less nodes in the network, e.g., 1000 nodes, the PER performances under RS-LoRa and LoRaWAN are similar. The reason has also been explained in the previous section: there are few collisions when the number of nodes in the network is small. In our multicell scenario, collisions are also resolved by different gateways that could be reached. Overall, our proposed RS-LoRa outperforms LoRaWAN, supported by Table IV where the average PERs of the network under RS-LoRa and LoRaWAN are presented. RS-LoRa improves the PER by 20.7% on average in a network with 3500 nodes, demonstrating RS-LoRa's advantage on improving the network reliability and scalability of LoRaWANs.

2) Throughput: Fig. 10 shows the evaluation result of throughput under multicell scenarios. Similar to the result of reliability, the throughput under RS-LoRa outperforms that under LoRaWAN due to less packet collisions.

3) Fairness: The evaluation results of the fairness are shown in Fig. 11. We can observe that RS-LoRa brings higher fairness. For example, when there are 3500 nodes served by the multiple gateways, this fairness under LoRaWAN can be increased from 95% to 99.6% by RS-LoRa.

D. Backward Compatibility

Experiments are also carried out to show the proposed RS-LoRa's backward compatibility with legacy LoRaWAN. In the experiments, 500 nodes are randomly distributed around the central gateway GW_1 (single-cell). Half of these nodes runs the legacy LoRaWAN protocol while the other half runs our proposed RS-LoRa protocol. The results of the PER performance in this coexisting network are shown in Fig. 12(a). We observe that the PER of RS-LoRa nodes is *higher* than that of LoRaWAN nodes. This is due to that RS-LoRa nodes use the transmission power and SFs guided by the gateway for an optimized global network performance. LoRaWAN nodes, however, are not restricted by these guideline and can choose the best configurations to suit their own needs. Thus, RS-LoRa messages are more prone to have collisions, especially with LoRaWAN nodes closer to the gateway. Moreover, the results demonstrate an interesting property of the proposed RS-LoRa: it is not aggressive when sharing the medium with other ALOHA access technologies in the same band. This shows as well that RS-LoRa is adaptive to realistic situations and as a hidden effect, does not degrade the performance of LoRaWAN when coexisting but in the contrary limiting its own capabilities to not interfere with LoRaWAN.

As a result, we also present the comparison of this coexistence scenario with scenarios when all the nodes run RS-LoRa or LoRaWAN, as shown in Fig. 12(b). We can clearly observe that at most of the positions *the global network performance is indeed improved with the existence of RS-LoRa nodes.* The extreme case is when all the nodes run the RS-LoRa protocol.

For the full convergence between legacy LoRaWAN nodes and RS-LoRa nodes, it is also interesting to see how legacy LoRaWAN Class B devices could be implemented with the proposed RS-LoRa. Many similarities can be found between RS-LoRa and the class B specifications. Their beacons both provide time and location information albeit that RS-LoRa shrinks this information to a minimum. However, since the default channel is not touched in RS-LoRa, the class B nodes could run as specified in the LoRaWAN specifications, or their beacons could be replaced by the proposed beacons of RS-LoRa. In the latter case, a translation layer needs to be designed to translate the data to the more elaborated data of the legacy beacons. Also, the timings need to be changed. The interval between class B beacons should then be $3T_s$, where T_s is the duration of a subframe.

VII. Conclusion

This paper designed, implemented, and evaluated RS-LoRa, a novel MAC protocol to improve reliability and scalability of LoRaWAN. A two-step lightweight scheduling was proposed to divide nodes into groups where similar transmission powers are used in each group to reduce the capture effect. The nodes were guided by the gateway's coarse-grained scheduling to use different SFs to enable simultaneous transmissions and thus reduce packet collisions. As shown by our simulation in NS-3, RS-LoRa improved the performance of the legacy LoRaWAN in terms of packet error ratio, throughput, and fairness with a reasonable energy efficiency. Our results also show that RS-LoRa can coexist with legacy LoRaWAN nodes due to the fairness mechanism provided in RS-LoRa. Going forward, we will continue to optimize the network performance and energy consumption of RS-LoRa.

References

[1] (2016). *3GPP Low Power Wide Area Technologies*. [Online]. Available: https://goo.gl/DaUHkV

[2] Semtech. (2017). *LoRa—Semtech*. [Online]. Available: http://iot.semtech.com

[3] (2017). *Sigfox*. [Online]. Available: https://www.sigfox.com/

[4] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent. (2015). *LoRaWAN Specification*. [Online]. Available: http://www.lora-alliance.org

[5] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," in *Proc. ACM EWSN*, Graz, Austria, 2016, pp. 361–366.

[6] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proc. ACM MSWiM*, 2016, pp. 59–67.

[7] B. Reynders, W. Meert, and S. Pollin, "Power and spreading factor control in low power wide area networks," in *Proc. IEEE ICC*, Paris, France, 2017, pp. 1–6.

[8] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication," in *Proc. IEEE SCVT*, Mons, Belgium, 2016, pp. 1–5.

[9] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.

[10] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *Proc. IEEE ICT*, Thessaloniki, Greece, 2016, pp. 1–6.

[11] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, 2017, Art. no. E1193.

[12] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in lorawan based lpwa networks?" in *Proc. IEEE GLOBECOM*, Singapore, 2017, pp. 1–6.

[13] K. Mikhaylov, J. Petajajarvi, and J. Janhunen, "On LoRaWAN scalability: Empirical evaluation of susceptibility to inter-network interference," in *Proc. IEEE EuCNC*, Oulu, Finland, 2017, pp. 1–6.

[14] F. Adelantado *et al.*, "Understanding the limits of LoRaWan," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017.

[15] P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso-Zarate, L. Alonso, and X. Vilajosana, "LPDQ: A self-scheduled TDMA MAC protocol for one-hop dynamic low-power wireless networks," *Pervasive Mobile Comput.*, vol. 20, pp. 84–99, Jul. 2015.

[16] K. Zhang and A. Marchiori, "Crowdsourcing low-power wide-area IoT networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Kailua-Kona, HI, USA, 2017, pp. 41–49.

[17] (2017). *Ingenu*. [Online]. Available: http://ingenu.com

[18] Weightless-SIG. (2015). *Neul's Weightless-N*. [Online]. Available: http://www.weightless.org

[19] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of MAC protocols in wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 101–120, 1st Quart., 2013.

[20] E. Vogli, G. Ribezzo, L. A. Grieco, and G. Boggia, "Fast join and synchronization schema in the IEEE 802.15.4e MAC," in *Proc. IEEE Workshop (WCNC)*, New Orleans, LA, USA, 2015, pp. 85–90.

[21] Semtech. (2017). *SX1272 LoRa Modem*. [Online]. Available: https://goo.gl/Rc5Q7v

[22] F. V. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017.

Brecht Reynders (GS'17) received the B.Sc. and M.Sc. degrees in electrical engineering from KU Leuven, Leuven, Belgium, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree focusing on multitechnology routing.

His current research interests include long range communication, distributed computing, and network modeling.

Qing Wang (S'13–M'17) received the B.E. and M.S. degrees from the University of Electronic Science and Technology, Chengdu, China, in 2008 and 2011, respectively, the M.S. degree from the University Carlos III of Madrid, Madrid, Spain, in 2012, and the Ph.D. degree from IMDEA Networks Institute, Madrid, in 2016.

He is currently a Post-Doctoral Associate with KU Leuven, Leuven, Belgium. His current research interests include visible light communication, Internet of Things, and device-to-device communication.

Pere Tuset-Peiro (M'12–SM'18) received the B.Sc. and M.Sc. degrees in telecommunications engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2007 and 2011, respectively, and the Ph.D. degree in network and information technologies from the Universitat Oberta de Catalunya, Barcelona, in 2015.

He is an Assistant Professor with the Computer Science, Multimedia and Telecommunications Department, Universitat Oberta de Catalunya. He has authored or co-authored over 20 journal and conference publications and holds 7 international patents.

Dr. Tuset-Peiro was a recipient of the Second Best Demo Award of IEEE INFOCOM and IEEE SENSORS in 2014 and 2017.

Xavier Vilajosana (M'13–SM'15) received the M.Sc. degree in computer science from the Universitat Politècnica de Catalunya, Barcelona, Spain, and the Ph.D. degree in computer science from the Universitat Oberta de Catalunya, Barcelona.

He is a Computer Science Engineer, the co-founder of Worldsensing, Barcelona, Spain, and OpenMote Technologies, Barcelona. He is currently a Principal Investigator with the Wireless Networks Research Laboratory, Universitat Oberta de Catalunya, Barcelona. From 2012 to 2014, he was a Visiting Professor with the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, holding a prestigious Fulbright fellowship. In 2008, he was a Visiting Researcher with France Telecom Research and Development Laboratories, Paris, France. He has been one of the main promoters of low power wireless technologies, co-leading the openWSN.org initiative at UC Berkeley, and promoting the use of low power wireless standards for the emerging Industrial Internet paradigm. He also contributed to the industrialization and introduction of low power wide area networks to urban scenarios through Worldsensing. He is contributing actively at the IETF 6TiSCH WG. He has extensive experience in distributed systems, wireless networks and industrial networks. He has authored different Internet drafts, as part of his standardization activities for low power industrial networks. He holds 30 patents and has authored over 40 high impact journal publications and over 50 International conference contributions.

Sofie Pollin (S'03–M'06–SM'14) received the Ph.D. degree (Hons.) from KU Leuven, Leuven, Belgium, in 2006.

From 2006 to 2008, she was involved with research on wireless communication, energy-efficient networks, cross-layer design, coexistence, and cognitive radio with the University of California at Berkeley, Berkeley, CA, USA. In 2008, she returned to imec, Leuven, to become a Principal Scientist with the Green Radio Team. She is currently an Associate Professor with the Electrical Engineering Department, KU Leuven. Her current research interests include networked systems that require networks that are ever more dense, heterogeneous, battery powered, and spectrum constrained.

Prof. Pollin is a BAEF and Marie Curie Fellow.