Exploiting Rateless Codes and Cross-Layer Optimization for Low-Power Wide-Area Networks

Gonglong Chen, and Wei Dong*
College of Computer Science, Zhejiang University.
Alibaba-Zhejiang University Joint Institute of Frontier Technologies.
Email: {*chengl, dongw*}@*emnets.org*

***Abstract*—Long communication range and low energy consumption are two most important design goals of Low-Power Wide-Area Networks (LPWAN), however, many prior works have revealed that the performance of LPWAN in practical scenarios is not satisfactory. Although there are PHY-layer and link layer approaches proposed to improve the performance of LPWAN, they either rely heavily on the hardware modifications or suffer from low data recovery capability especially with bursty packet loss pattern. In this paper, we propose a practical system, eLoRa, for COTS devices. eLoRa utilizes rateless codes and jointly decoding with multiple gateways to extend the communication range and lifetime of LoRaWAN. To further improve the performance of LoRaWAN, eLoRa optimizes parameters of the PHY-layer (e.g., spreading factor) and the link layer (e.g, block length). We implement eLoRa on COTS LoRa devices, and conduct extensive experiments on outdoor testbed to evaluate the effectiveness of eLoRa. Results show that eLoRa can effectively improve the communication range of DaRe and LoRaWAN by 43.2% and 55.7% with packet reception ratio higher than 60%, and increase the expected lifetime of DaRe and LoRaWAN by 18.3% and 46.6%.**

I. Introduction

Low-Power Wide-Area Networks (LPWANs) are emerging communication technologies. They offer wireless communications over long range and have power and cost advantages than other traditional wireless networks. Among many LPWANs (e.g., LoRaWAN [1], NB-IoT [2], SigFox [3] and etc.), LoRaWAN is a technology that has attracted many research interests [4–11].

Long communication range and low energy consumption are two most important design goals of LoRaWAN. However, many prior works [4, 5, 12, 13] have revealed that the performance gap of LoRaWAN between the practical scenarios and the theory is still very large. For example, Augustin *et. al.* [13] and Adwait *et. al.* [4] have shown that in a typical outdoor deployment of LoRaWAN, the effective communication range is only 650m (with packet reception ratio higher than 60%) considering the hardware platform stated in [4, 13].

Recently, many research works have been proposed to improve the performance of LoRaWAN. The PHY-layer approach, e.g., Charm [4], Choir [5], extends the communication range by improving packet reception ratio based on LoRa signal characteristics. Nevertheless, they rely heavily on the hardware modifications to perform sophisticated signal processing, which is not directly accessible on Commercial Off-The-Shelf (COTS) devices. The application-layer approach, e.g., DaRe [6], provides data recovery in LoRaWAN based on the forward error correction code (FEC). However, the error correction capability is highly related to the packet loss patterns (see Section III). For example, given the fixed coding rate, the consecutive packet loss pattern would easily exceed the correction capability of DaRe, resulting in large retransmission overhead and energy consumption.

To address the above limitations, we propose a practical system, eLoRa, for COTS devices. eLoRa has two important features: 1) eLoRa exploits rateless codes that decouple error recovering units from communication units, i.e., eLoRa transmits large packets which contain multiple blocks with configurable block length. eLoRa performs rateless codes on blocks to cope with errors in unreliable links. The use of rateless codes allows decoding with multiple gateways, resulting in longer communication range and lifetime of LoRaWAN. 2) eLoRa jointly optimize the parameters of the PHY-layer (e.g., spreading factor) and the link layer (e.g., block length) to further improve the performance of LoRaWAN. eLoRa carefully models the cross-layer parameters that have most impact on LoRaWAN. For example, the spreading factor (SF) indicates how many chips are spreading out for one symbol (i.e., 2^{SF} chips are encoded as SF bits for one symbol). A larger SF denotes more chips are encoded for one symbol, and thus increases the transmission reliability. However, it lowers the data rate and increases the energy consumption. eLoRa provides an optimization framework to optimize the cross-layer parameters.

We implement eLoRa on Dragino LoRa Shield [14] and Dragino LG01 gateway [14], and evaluate its performance in different environments. Results show that eLoRa achieves a highly accurate network model (e.g., absolute error of 0.98% for reliability estimation). To demonstrate the effectiveness of eLoRa in real scenarios, we implement eLoRa in a real-deployed testbed. Results show that eLoRa can effectively improve the communication range of DaRe and LoRaWAN by 43.2% and 55.7% with packet reception ratio higher than 60%, and increase the expected lifetime of DaRe and LoRaWAN by 18.3% and 46.6%.

*Corresponding author. We gratefully acknowledge our shepherd Saurabh Bagchi and the ICNP reviewers for their insightful comments. This work is supported in part by the National Science Foundation of China under Grant No. 61772465, and the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under Grant No. LR19F020001.

978-1-7281-2700-2/19/$31.00 2019

The contributions of this paper are summarized as follows:

- We design a practical system, eLoRa, for COTS devices to extend communication range and lifetime of LoRaWAN. eLoRa provides two application level parameters that can be specified by network operators (e.g., communication range and lifetime). eLoRa can automatically optimize the parameters based on the monitored network states.
- We jointly consider the parameters of the PHY-layer (e.g., spreading factor) and the link layer (e.g., block length) and propose a cross-layer optimization framework. eLoRa utilizes rateless codes and jointly decoding with multiple gateways to improve the performance of LoRaWAN without hardware modifications.
- We implement and evaluate eLoRa on COTS devices. Extensive experiments in the real world testbed show that eLoRa outperforms the state-of-art in terms of lifetime and communication range.

The rest of this paper is organized as follows. Section II introduces the related work and background of LoRaWAN. Section III presents two motivating examples. Section IV shows the design of eLoRa. Section V presents the evaluation results. Section VI concludes this paper and discusses future research directions.

II. Related Work

Long Range Wide-Area Networks (LoRaWAN). LoRaWAN uses chirp spread spectrum (CSS) for transmitting data considering the requirements of low power, hardware simplicity, and robustness under multi-path and narrow-band interference [1]. LoRaWAN transceivers can operate between 137 MHz to 1020 MHz with licensed bands included, however, they are often deployed in ISM bands (e.g., China: 779MHz and 433MHz, EU: 868MHz and 433MHz). The LoRa physical layer may be used with any MAC layer, however, LoRaWAN is the currently proposed MAC. LoRaWAN operates in a simple star topology. To improve communication efficiency, LoRaWAN provides an adaptive data rate (ADR) mechanism [1]. Within the maximum retransmission times, LoRaWAN gradually reduces the data rate from 11kbps to 0.25kbps as the retransmission time increased.

Performance Optimization for LoRaWAN. Charm [4] enhances the coverage of LoRaWAN and the battery life of client devices through multiple gateway combination. It exploits the observation that the weak signals from clients can be identified through filtering the signal patterns of LoRa modulation. Then by coherently combining weak signals received across multiple gateways, the underlying data can be decoded successfully with high probability. Choir [5] is a collision decoding approach in LoRaWAN exploiting hardware imperfections. It exploits the observation that signals from the two transmitters are likely to experience a small frequency offset due to the hardware imperfections. Using this offset, collided signals can be decoupled and decoded. All the above two approaches rely heavily on the hardware modifications on LoRa devices because of the procedures of complex signals at the PHY-layer. The above two approaches can be directly applied in eLoRa to further improve the performance and therefore eLoRa is orthogonal to them. On the other hand, eLoRa is a software based system and can be directly deployed in COTS LoRa devices.

DaRe [6] recovers the lost data in LoRaWAN using the redundant information calculated from previous frames. However, given the fixed coding rate, DaRe suffers from low efficiency under different packet loss patterns. For the example shown in the Section III, when bursty packet loss occurs more retransmissions are needed to recover the lost packets. Different from DaRe, eLoRa utilizes the rateless code (e.g., LT code [15]) that can automatically achieve a proper bit rate for the given link. Note that the decoding delay of eLoRa may be larger than DaRe. However, LoRaWAN applications are usually delay non-sensitive. For example, the uplink transmission limitation of LoRaWAN application is 30 seconds on-air-time per day per device [16], indicating that we should carefully design the packet recovery mechanism for LoRaWAN applications such that LoRaWAN packets can be successfully decoded at the receiver side within the very limited on-air-time. Comparing with aggressively dealing with more packets with lower packet reception ratio by DaRe, it may be more appropriate to carefully recover enough packets with acceptable delay.

Martin *et. al.* [8] develop a link probing based approach to automatically adjust the parameters for LoRa transmissions. The parameter selection is purely based on empirical study which may not perform well in practical scenarios. Thiemo *et. al* [9] propose to use directional antenna and multiple gateways to deal with the inter-network interference. Simulation results show that the performance of LoRaWAN can be improved, however, it requires heavy deployment and modification cost on all of the end-devices. Different from this approach, eLoRa is a software based approach and is thus more efficient.

LoRaSim [10] is a simulator that captures low level LoRa communication behaviours such as capture effect and packet collisions. Floris *et. al* [11] propose another simulator to study the packet delivery ratio in a large scale simulated LoRaWAN. Different from the above two simulators that only empirically study the impact of parameters on LoRaWAN's performance, our system not only models LoRaWAN in a cross-layer manner, but also provides a jointly optimization scheme to improve the transmission performance of LoRaWAN.

Performance Optimization for wireless sensor networks (WSN). LoRaCP [7] reduces collisions in WSN by leveraging LoRa's capability to transmit control messages over one-hop out-of-band. It is an application of LoRa for WSN. We believe that eLoRa can also benefit the transmission efficiency of control messages in LoRaCP. SYNAPSE++ [17] improves the efficiency of data dissemination by optimizing the degree distribution of LT code [15]. DLT [18] provides long-distance transmissions in WSN by parallelizing the Gaussian Elimination decoding of LT code. pTunes [19] is a framework for runtime adaptation of low-power MAC protocol parameters. Using the information about the current network state, pTunes automatically determines optimized

MAC parameters whose performance meets the requirements specification. Different from pTunes that optimizes parameters using single gateway, eLoRa also combines multiple gateways to further improve the performance of LoRaWAN.

Although eLoRa utilizes several technologies from WSN, eLoRa is significantly different from them in two aspects. First, we perform a detailed modeling of LoRaWAN PHY-layer, which is quite different from WSN PHY-layer, e.g., the chirp spread spectrum for LoRa and the direct-sequence spread spectrum for ZigBee. Second, for long-range communications in LoRaWAN, we need to carefully design the system to adapt to the complicated environments, while short-range communications are typical scenarios in WSN.

III. Motivation

In this section, we present an example to show the benefits of using rateless code, and how the performance can be further improved through cross-layer optimization.

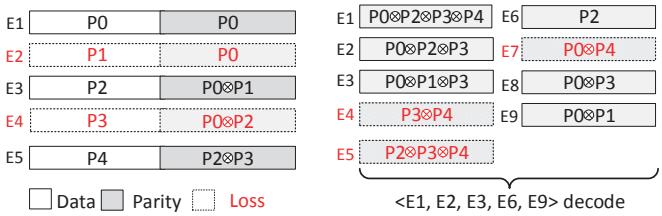
A. Benefits of Using Rateless Code

DaRe. In certain LoRaWAN data collection scenarios [6], the data sensed by LoRa devices are needed to be transmitted to the gateway in a real time manner. To this end, Marcelis *et. al.* [6] propose an application level coding approach DaRe. It encodes data packets with the redundant information that is calculated from previous data packets. When encoded packets are lost, DaRe tries to recover the lost data packets using the redundant information.

However, it is possible that DaRe is unable to recover the data packets given the fixed coding rate, resulting in large retransmission overhead. For example, suppose there are five data packets with 40 bytes each to be transmitted to the gateway, then a typical encoding pattern of DaRe packets is shown in Fig. 1(a). A DaRe encoded packet (e.g., E4) is concatenated with a data packet (e.g., P3) and a parity packet (e.g., $P0 \otimes P2$), resulting in the coding rate $R = 0.5$. The parity packet is calculated by randomly XORing two data packets (e.g., degree $d = 0.66$) from previous three data packets (e.g., sliding window $W = 3$). Although there are two DaRe encoded packets lost (e.g., $PRR = 0.6$), DaRe can still recover the lost data packets (e.g., P1 and P3) from the correctly received packets. For example, P1 can be recovered from E1 and E3, while P3 can be derived from E3 and E5. Therefore, for the packet loss pattern in Fig. 1(a), to reliably transmit $5 \times 40 = 200$ data bytes, $10 \times 40 = 400$ bytes are needed.

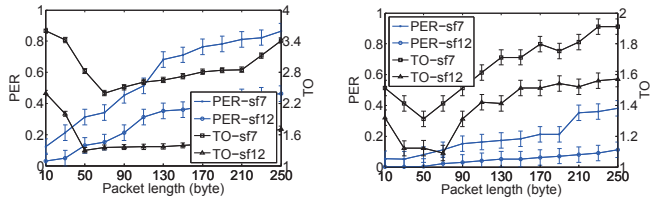
However, given the manually configured coding rate, when other packet loss patterns occur (e.g., consecutive lost packets), additional packets are needed. For example, when packets E4 and E5 are lost (e.g., the consecutive packet loss pattern), data packets P3 and P4 can not be recovered from the redundancy and more redundant packets are needed (e.g., $2/0.6 \approx 3$ extra retransmission packets). Therefore, $10 \times 40 + 3 \times 40 = 520$ bytes are needed.

Rateless code. For a given link, the rateless code can automatically achieve a proper bit rate. To this end, a packet is first split into multiple blocks (say k data blocks). The



(a) DaRe (b) LT code

Fig. 1: Motivating example. Given five data packets with 40 bytes each to be transmitted. Two DaRe encoded packets and three LT encoded packets are lost. Dashed boxes are lost packets, solid boxes are correct packets.



(a) Low SNR scenarios (-8dB). (b) High SNR scenarios (5dB).

Fig. 2: Impact of parameters on LoRa performance.

rateless code then encodes the block with d data blocks which are randomly chosen from the k data blocks. When receiving an encoded block successfully, the receiver can get an index vector I of data blocks that are encoded. The vectors are accumulated to form a matrix G. The k data blocks can be recovered using Gaussian Elimination (GE) once the rank of G is k [20]. By carefully designing the distribution of d, it is highly possible to using $k+\delta$ correctly received blocks to recover k data blocks [20], where δ can be extremely small.

Among the existing light-weight rateless codes [15, 20], we choose LT code [15] because of the light-weight coding operation (e.g., XOR) and the high decoding efficiency optimized by prior works [17, 18]. Fig. 1(b) presents an example of using LT code to encode the five data packets (40 bytes each) to be transmitted to the gateway. Note that we treat the packet as the encoded block for illustration purpose. Although three packets are lost (e.g., $PRR \approx 0.667$), the LT code can still recover the five data packets by accumulating six encoded packets. In total, $(3+6) \times 40 = 360$ bytes are needed to reliably transmit $5 \times 40 = 200$ data bytes. Comparing with DaRe, LT code can significantly reduce the transmitted bytes from 520 bytes to 360 bytes at most.

Therefore, we can achieve more efficient transmission by automatically achieve a proper bit rate with LT code, while DaRe relies heavily on manual configurations which may result in large transmission overhead. Note that the latency of DaRe may be lower than LT code (e.g., LT code needs to wait until all five data packets are ready to perform encoding). However, in most LoRaWAN scenarios [16] that are insensitive to the latency due to the low duty cycle limitations (e.g., $<1\%$), LT code is more suitable than DaRe.

B. Benefits of Parameter Optimization

Now that we have improved the error correction capability through LT code, we investigate how to further improve the performance by optimizing the cross-layer parameters. We first conduct empirical experiments to study the impact on LoRa transmissions without any codes considering two representative parameters from the PHY-layer (e.g., spreading factor) and link layer (e.g., block length). The spreading factor (SF) denotes the amount of chips per symbol [21]. The larger the SF is, the more reliable the LoRa transmission is, however, resulting in longer packet on-air-time. We measure the PER and the normalized transmission overhead (denoted as TO) per useful received byte for each packet length and SF. A low TO value indicates a high goodput [22].

We place an end-device and a gateway at a distance of 100m in an outdoor scenario. The transmission power is varied at the end-device to result in different SNR (e.g., -8dB and 5dB) at the gateway. We change the value of SF and the packet length while setting other parameters as default in LoRaWAN [1] (e.g., 125kHz bandwidth and 4/5 coding rate). Each experiment is repeated 20 times.

Fig. 2 shows the impact of packet length and SF on the PER under different SNRs. We can find that in low SNR scenarios, the packet length has a greater impact on PER. For example, when packet length changes from 80 bytes to 40 bytes, the PER is reduced from 0.4 to 0.3. On the other hand, SF also impacts the PER (e.g., PER is reduced from 0.4 to 0.2 when enlarging SF from 7 to 12). Note that the shortest packet length may not produce the optimal TO due to more overhead incurred from the packet header. Given the same example shown in Fig. 1, through the cross-layer parameter optimization (e.g., enlarging SF to 12 and the packet length to 50 bytes) we can reduce the transmitted bytes from 360 to 250. Although we have achieved more efficient transmission, however, resulting in more energy consumption due to longer transmission time caused by the larger SF (e.g., LoRa data rate is reduced from 5kbps to 0.2kbps when enlarging SF from 7 to 12). Similarly, a smaller block length leads to a more reliable transmission, however, resulting in a larger CRC overhead because the number of blocks is increased. Therefore, we need to design a cross-layer optimization framework that jointly consider the high level requirements (e.g., lifetime, communication range and latency).

IV. Design

In this section, we present overall procedures, design details, and the network model of eLoRa.

A. Overview

Fig. 3 presents the overview of eLoRa architecture. **At the cloud side,** it receives the same packets from multiple gateways to assemble a new packet that contains most correct blocks. Then the packet is recovered using LT code with high probability. eLoRa provides an attractive feature that network operators can define application level requirements (e.g., lifetime and communication range). eLoRa will periodically check whether the requirements are violated, and automatically carry out the cross-layer parameter optimization based on the network model and the monitored network states. To ensure the requirements are not violated, eLoRa periodically carries out the cross-layer parameter optimization based on the network model and the monitored network states. The optimized parameters are then transmitted to the specific end-device over the gateway. **At the gateway side,** it relays the received data to the cloud via Internet or transmits the parameter settings to the end-device over LoRa links. **At the end-device side,** when receiving a packet from the network layer, it splits the packet into blocks and performs LT coding over the blocks using the optimized parameters. The end-device also updates the parameters (e.g., block size x) upon the commands from the gateway.

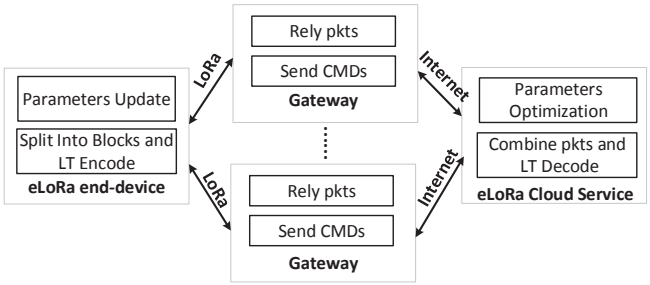


Fig. 3: Overview of eLoRa architecture.

Algorithm 1: Retransmission protocol at the end-device.

```
1  T_bf is the back off time;
2  α is used to determine the number of merged packets;
3  case ACK received do
4      clear up pkt in the sender buffer sbf;
5      state = next_pkt;
6  case NAK received do
7      extract the num. of blocks CB from NAK;
8      generate CB blocks using LT code;
9      send the encoded pkt after T_bf and start ACK timer;
10 case ACK timer times out do
11     retransmit pkt after T_bf;
12 case pkt received from network layer do
13     put pkt into the buffer bf;
14     if len(sbf) ≥ len(pkt)*α and state == next_pkt then
15         generate len(sbf)/x data blocks with x bytes each;
16         generate len(sbf)/x encoded blocks using LT code;
17         send the pkt with encoded blocks and start ACK
             timer;
18         state = send;
19     else
20         move pkt from bf into the send buffer sbf;
```

B. Key Procedures at The End-device and The Cloud

Algorithm 1 shows the key procedures at the end-device side. To improve the transmission efficiency, eLoRa will first accumulate the packets received from the network layer into a send buffer sbf until it is large enough to be sent (e.g., packet size is at least larger than $\alpha * len(pkt)$). Where α can be used

Algorithm 2: Retransmission protocol at the cloud side.

```
1  G_n is the number of gateways receiving pkts from
     end-device n;
2  M is the communication distance; L is the latency;
3  case pkt received from gateway do
4      put pkt into the receive buffer rbf;
5      if G_n packets are received then
6          combining most correct blocks in pkt;
7          pkt2 = lt_decode(pkt);
8          if pkt2 is correct then
9              reply ACK and deliver pkt2 to the network layer;
10         else
11             reply NAK with the num. of incorrect blocks
                 CB;
12 case constraint violation do
13     find optimized parameters;
14     if no solution then
15         decrease M or increase L until find the solution;
16     else
17         send parameter settings;
```

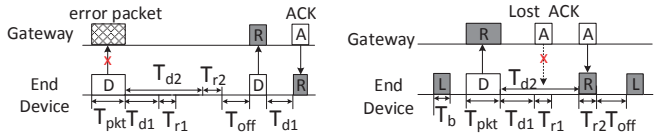
to determine the number of merged packets. Then eLoRa splits the packet into multiple blocks to perform the LT encode. When there is a NAK received, eLoRa will generate CB new LT encoded blocks for retransmissions.

Algorithm 2 presents the key procedures at the cloud side. The network operator can adjust four high level parameters: the communication range M, the lifetime T, the reliability R and the latency L. M or T can be selected as the optimization goal while others are treated as the constraints. eLoRa periodically checks the network states and performs cross-layer optimizations when constraints are not satisfied. When there are no proper solutions under current network states, to best fit the requirements, eLoRa gradually relaxes the constraints until the optimized parameters are found (e.g., decreasing M or increasing L). To perform LT decoding, eLoRa infers the seed using the similar way in [18] that from the packet id and the offset of the block.

C. Network Model and Basic Notations

We consider a typical LoRaWAN network that the end-device transmits packets to the gateway through one-hop [16]. There are three classes defined in LoRaWAN [1], i.e., Class A, Class B and Class C. By default, all three classes should implement the basic procedure of Class A. Class A and Class B are low power MAC protocols, enabling duty cycle to save the end-device's energy. eLoRa builds on two representative low power MAC protocols of LoRaWAN, i.e., Class A and Class B [1]. Note that we do not model the Class C because it keeps the end-device's radio on and consumes too much energy, which is rarely used in practical scenarios.

Fig. 4(a) shows the basic procedure of LoRaWAN Class A. For the end-device, it wakes up every T_p and sends packets to the gateway (i.e., the uplink transmission) if any. Once the uplink transmission is done, the end-device must



(a) LoRaWAN Class A. (b) LoRaWAN Class B.

Fig. 4: Two typical LoRaWAN classes.

wait a specific time T_{d1} and then open two short receive windows (i.e., windows length T_{r1} and T_{r2}) for the potential packets from the gateway (i.e., the downlink transmission). The interval between two short receive windows is $T_{d2} - T_{d1}$ as shown in Fig. 4(a). The length of the first window is fixed (i.e., set to one second by default), whereas the length of the second window can be modified by sending MAC commands by the gateway. If there is no downlink transmission arrived at any two of receive windows, the end-device will go to sleep for T_{off}. When a gateway want to send downlink packets, it must keep on listening until a uplink packet is received correctly. The gateway can send multiple downlink packets by setting the FPending bit to one in the packet header, while the end-device can only send or resend one packet every T_p. T_{pkt} denotes the time for transmitting a packet and T_{ack} denotes the time for transmitting an ACK.

Fig. 4(b) shows the basic procedure of LoRaWAN Class B. In addition to the two receive windows after the uplink transmission as defined in Class A, the end-device in Class B also opens another listening window lasting for T_b every T_p to receive downlink packets. The extra window is synchronized by the gateway with a factor of T_p time.

There are mainly four runtime-adjustable parameters in the LoRaWAN PHY-layer: spreading factor s (SF), coding rate c (CR), bandwidth b (BW) and transmission power P_t [21]. SF determines how many chips are spreading out in a given bandwidth for one symbol (i.e., 2^{SF} chips are encoded as SF bits for one symbol). It can be set between 7 and 12. A higher SF denotes more chips are encoded for one symbol, and thus increases the receiver sensitivity and the range of the signal. However, it lowers the data rate and therefore increases the transmission duration and energy consumption. In a typical LoRa deployment, BW can be set to 125 kHz, 250 kHz or 500 kHz. A wider bandwidth increases the data rate. CR is the amount of FEC (e.g., Hamming code [21]) that is applied to the message to protect it against interference. Higher CR makes the message longer and therefore increases the time on air. Transmission power can be varied from 5dBm to 23dBm (on rf95 [23]).

Table 1 summaries the notations we use to denote network states and protocol-dependent quantities.

D. Application Level Metrics

In a typical data collection scenario with static LoRa devices, we simultaneously consider two important application level metrics of real-world LoRaWAN applications [16]: lifetime T, communication range M.

Lifetime. Similar to prior works [22, 24], we first analyze the lifetime in terms of its duty cycle, i.e., the fraction of

Table 1: Notations used in this paper.

Notation	Meaning
n	The current end-device
S	The set of end-devices in the network
G_n	The number of gateways that can receive packets from the end-device n
H	Header size in one packet
T_{d1}, T_{d2}, T_{r1}, T_{r2}, T_p, T_{off}, T_{la}, T_r	Class A parameters (see Fig. 4(a) and Section IV-E)
T_{d1}, T_{d2}, T_{r1}, T_{r2}, T_p, T_{off}, T_{lb}, T_b, T_r	Class B parameters (see Fig. 4(b) and Section IV-E)
U_{tn}	Useful data rate at the end-device n (bytes/h)
x_n	Block length at the end-device n
d_{nj}	The distance between the gateway j and the end-device n
$T_{pkt}(x)$	Time to transmit one packet with x bytes
T_{ack}	Time to transmit one ACK
s, c, b, P_t	LoRa PHY-layer parameters, spreading factor (s), coding rate (c), bandwidth (b), transmission power (P_t)
α	The number of data packet needed to be merged for the LT encode.
$\boldsymbol{\nu}_n$	Parameters to be optimized and used at the end-device n, $\{s, c, b, P_t, x_n, \alpha\}$
N	The maximum retransmission time.
F_{tnb}	The block transmission rate at end-device n
F_{tnp}	The packet transmission rate at end-device n

time that is used in transmission mode (D_{tn}) and receiving mode (D_{rn}). For the Dragino LoRa Shield [23], the current of radio in transmission mode is 120mA (with transmission power at 23dBm) and in receiving mode is 16.6mA, while the current of MCU is only 3.5mA. Then the duty cycle of node n is calculated as $D_n = D_{tn} + D_{rn}$. We want to minimize D_n to the meet the application level requirements set by the network operator. Given a battery capacity Q, we define the node lifetime T_n of end-device n as [19]:

$$T_n = Q/(D_{tn} \cdot I_{tx} + D_{rn} \cdot I_{rx} + D_{in} \cdot I_i), \tag{1}$$

where I_{tx}, I_{rx}, and I_i are the current draws of the radio in transmitting, receiving, and idle mode. $D_{in} = 1 - D_{tn} - D_{rn}$.

Communication range. We let M_n denote the distance between the end-device n and the gateway, and R_n denote the packet reception ratio of a gateway for the end-device n. The distance of every end-device is recorded at the initialization step at the gateway. The network operator gives the minimum communication distance d, our system would try to ensure the PRR of all nodes within the range d is higher than the PRR requirement r. It has a default value (e.g., 60% [12]) in our system and can be adjusted by the network operator.

Then we can express the optimization problem in our system in the following general form:

$$\begin{aligned} \min / \max \quad & A_1(c) \\ s.t. \quad & A_2(c) <, > C_1, \end{aligned} \tag{2}$$

where A_i is one among $\{T_n, M_n, R_n\}$ and C_i is the requirement. We also consider the transmission latency L_n in the constraint. For example, given the PRR requirement r, the lifetime requirement t and the latency requirement l, the optimization problem is:

$$\begin{aligned} \max \quad & M_n \\ s.t. \quad & T_n > t, R_n > r, L_n < l, n \in S. \end{aligned} \tag{3}$$

Note that eLoRa can only optimize one goal at a time, e.g., the communication range or lifetime, while other metrics are treated as the constraints. To maximize the communication range, we first transform the above problem into: maximize the R_n while meet $T_n < t$ and $L_n < l$. After optimization, we choose the end-device n with the maximum distance from all end-devices satisfying $R_n > r$, as the optimized communication range. In the following, we present the details of modeling the application level metrics and relevant requirements.

E. Metric Modeling

We define the optimization parameters as the set $\boldsymbol{\nu}_n$, which consists of the number of data packets α to be merged, the block size x and all other PHY-layer parameters (e.g., spreading factor s, coding rate c, bandwidth b and transmission power P_t). Then we present the relationship among $\boldsymbol{\nu}_n$ and high level metrics, i.e., lifetime T, reliability R, and latency L.

Lifetime modeling. To model the lifetime of LoRa end-device, we first need to accurately model the duty cycle, i.e., $D_n = D_{tn} + D_{rn}$. As shown in Fig. 4, the communication procedures of LoRaWAN classes are different, resulting in different duty cycle modeling.

For **LoRaWAN Class A**, it consists of the time used in transmitting packets (D_{tn}) and receiving ACK/NAK from the gateway (D_{rn}). When ACK/NAK is not received, the end-device needs to listen for T_{d2} and T_{r2}:

$$D_{na} = D_{tn} + (1 - D_{rn}) \frac{T_{d2} + T_{r2}}{T_p} + D_{rn} \frac{(T_{d1} + 2 \cdot T_r + T_{d2})}{2 \cdot T_p}, \tag{4}$$

where $T_p = T_{pkt} + T_{off} + T_{la}$. $T_{la} = T_{d2} + T_{r2}$ when there is no ACK/NAK received. On the other hand, we assume that probability of successfully receiving ACK/NAK in the first receive window T_{r1} and the second receiving window T_{r2} is equal, and T_r is the time for receiving ACK/NAK from the gateway, then $T_{la} = 0.5 \cdot (T_{d1} + T_r) + 0.5 \cdot (T_{d2} + T_r)$. D_{tn} can be further divided into the time faction in transmitting LT encoded blocks and packet headers.

$$D_{tn} = F_{tnb} T_{pkt}(x_n) + F_{tnp} T_{pkt}(H), \tag{5}$$

where F_{tnb} is the rate of transmitting blocks (given in Eq. (11)), and F_{tnp} is the rate of transmitting packets which may contain multiple blocks (given in Eq. (13)). D_{rn} consists of the time faction for receiving ACK/NAK from the gateway.

$$D_{rn} = F_{tnp} PRR_{ack}(\boldsymbol{\nu}_n) T_{ack}. \tag{6}$$

For **LoRaWAN Class B**, its difference from Class A is the additional listening window T_b for receiving packets from the gateway.

$$D_{nb} = D_{tn} + (1 - D_{na}) \frac{T_b}{T_p} + D_{na}, \tag{7}$$

where $T_p = T_{pkt} + T_{off} + T_{la} + T_{lb}$. Where T_{la} is the same with class A. $T_{lb} = T_b$ when there is no packets received, and otherwise T_{lb} is the total time for successfully receiving packets from the gateway.

Reliability modeling. It is expressed as the packet reception ratio from the end-device n to the gateway j. It depends on parameter vector $\boldsymbol{\nu}_n$.

$$R_n = \frac{\alpha \cdot U_{tn}}{x_n \cdot N_{tnb}(\boldsymbol{\nu}_n)}, \tag{8}$$

where U_{tn} denotes there is one data packet with U_{tn} bytes generated per hour, and $N_{tnb}(\boldsymbol{\nu}_n)$ denotes the expected time of retransmitting LT encoded blocks.

Latency modeling. We define the latency as the time needed for the gateway successfully receiving a packet from the end-devices. When packet merging is enabled (e.g., $\alpha > 1$), the latency is defined as the time for successfully receiving the first generated packet. Then the latency consists of the time for transmitting or retransmitting packets, and the time for waiting new packets and backoff T_{wait}.

$$L_n = N_{tnb}T_{pkt}(x) + N_{tnp}T_{pkt}(H) + (N_{tnp} - 1)\cdot T_{wait}, \quad (9)$$

where N_{tnp} is the expected number of transmitting packet that may contain multiple LT encoded blocks (given in Eq. (14)). T_{wait} for LoRaWAN Class A consists of the time in listening when no ACK/NAK is received, waiting new packets and backoff.

$$T_{waita} = (1 - PRR_{ack})(T_{d2} + T_{r2}) + T_{bf} + \alpha \cdot T_D, \quad (10)$$

where T_D is the time for generating a new packet from network layer. For LoRaWAN Class B, there are additional listening windows opened for T_b and therefore $T_{waitb} = T_{waita} + T_b$.

F. Link Layer and PHY-Layer Modeling

In this subsection, we will present the building blocks for modeling the previous metrics. We denote F_{tnb} as the block transmission rate of an end-device that depends on the block length x and the number of merged packets α.

$$F_{tnb} = N_{tnb}(\boldsymbol{\nu}_n)\cdot\alpha\cdot U_{tn}/x_n, \quad (11)$$

where $N_{tnb}(\boldsymbol{\nu}_n)$ is the expected number of block transmissions given the parameter vector $\boldsymbol{\nu}_n$ and can be expressed as:

$$N_{tnb}(\boldsymbol{\nu}_n) = \min(\sum_{i=k}^{\infty} i\cdot BRR_{lt}(m,k,\boldsymbol{\nu}_n), N), \quad (12)$$

where $BRR_{lt}(m,k,\boldsymbol{\nu}_n)$ is the block reception rate using LT code, i is the number of transmitted blocks, m is the number of blocks received successfully, and $m = i\cdot BRR_g$. BRR_g is the block reception rate using multiple gateways (Eq. (15)). k is the number of useful blocks and is equal to $\alpha\cdot U_{tn}/x_n$.

We denote F_{tnp} as the rate of transmitting packets that may contain multiple LT encoded blocks from an end-device.

$$F_{tnp} = N_{tnp}(\boldsymbol{\nu}_n)/\alpha, \quad (13)$$

where $N_{tnp}(\boldsymbol{\nu}_n)$ denotes the expected number of packet transmissions that may contain multiple LT encoded blocks:

$$N_{tnp}(\boldsymbol{\nu}_n) = \log_{(1-BRR_{lt}(m,k,\boldsymbol{\nu}_n))}\left(1 - \frac{N_{tnb}(\boldsymbol{\nu}_n)\cdot x_n}{PRR_{ack}\cdot\alpha\cdot U_{tn}}\right). \quad (14)$$

Given the number of gateways G_n that can receive the packets from the end-device n, then BRR_g is defined as when the block is correctly received at any one of the gateway, then it can be directly fed into the LT decoder:

$$BRR_g(\boldsymbol{\nu}_n, G_n) = 1 - \prod_{j=1}^{G_n}(1 - BRR_{raw}(\boldsymbol{\nu}_n, d_{nj})), \quad (15)$$

where d_{nj} denotes the distance between the end-device n and the gateway j, and $BRR_{raw}(\boldsymbol{\nu}_n, d_{nj})$ denotes the block reception rate without LT decoding:

$$BRR_{raw}(\boldsymbol{\nu}_n, d_{nj}) = (1 - B(\boldsymbol{\nu}_n, d_{nj}))^{8(x+1)}, \quad (16)$$

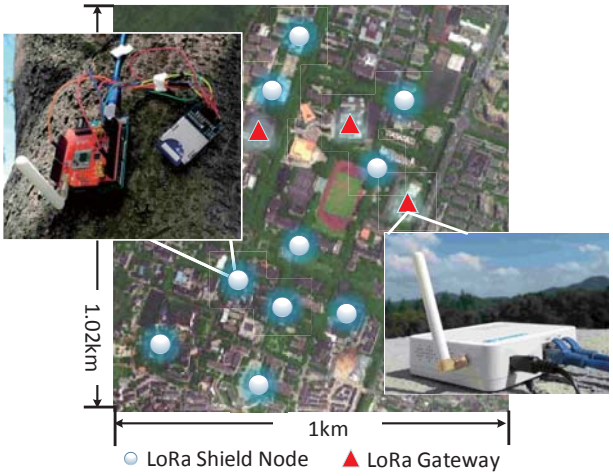


Fig. 5: Outdoor testbed setup.

where $B(\boldsymbol{\nu}_n, d_{nj})$ is the bit error rate given the PHY-layer parameters [21] and can be expressed as [25]:

$$B(\boldsymbol{\nu}_n, d_{nj}) = Q\left(\frac{\log_{12}(s)}{\sqrt{2}}\frac{E_b}{N_0}\right), \quad (17)$$

where $\frac{E_b}{N_0}$ is the signal to noise ratio per bits, and it is related to the SF s, the CR c [21] and the distance d_{nj}, the transmission power P_t. The relationship can be derived based on the existing path loss model [10].

As for the block reception rate $BRR_{lt}(m,k,\boldsymbol{\nu}_n)$ of LT code, because we utilize the Gaussian Elimination to decode LT encoded blocks, then it turns to calculate the probability of receiving m blocks with k ranks (e.g., m successfully received blocks, k data blocks). It can be derived similar with [20, 26] and we do not provide further details in this paper. The packet transmission time T_{pkt} is a function of packet length, the SF, the CR and the BW. It can be calculated according to [21].

G. Cross-layer Optimization

Applying the optimization problem in Eq. (2) to Class A and Class B leads to a mixed-integer nonlinear program (MINLP) with non-convex objective and constraint functions. To solve it efficiently, we use the ECLiPSe constraint programming system [27]. Its high-level programming paradigm allows for a succinct modeling of our optimization problem. We solve the optimization problem at the cloud side and send the parameter update command to the gateway to distribute the settings to the end-devices.

V. Evaluation

In this section, we introduce the testbed setup and present the evaluation results of eLoRa.

A. Experimental Setup and Metrics

Testbed. As shown in Fig. 5, to evaluate the performance of eLoRa in practical scenarios, we build a LoRaWAN testbed in a 1.02km × 1km campus with 10 LoRa Shield nodes and three LoRa gateways. Gateways are placed on the roof to achieve a wider coverage. Each gateway connects to the Internet via the wired connection, so that we can access the received data

Table 2: Parameter settings. Class A and Class B are two typical device types of LoRaWAN.

	Class A		*Class B*		*Specified Range*
	S1	S2	S3	S4	
T_{cc}	1%	10%	5%	15%	-
U_{tgn}(bytes/h)	100	150	100	150	80∼260
s	9	7	9	7	7∼12
c	4/8	4/5	4/8	4/5	4/5, 4/6, 4/7, 4/8
b(kHz)	250	125	250	125	125∼500
P_t(dBm)	20	14	20	14	5∼23
G_n	3	1	3	1	1∼3

Table 3: Absolute errors of estimating high level metrics.

	Class A			*Class B*		
	S1	S2	S1-eLoRa	S3	S4	S3-eLoRa
δ(R)[%]	0.14	-0.86	0.21	2.98	-1.25	0.45
δ(T)[year]	0.05	0.15	-0.23	0.14	-0.15	0.11
δ(L)[h]	0.12	0.14	0.2	-0.05	0.12	0.13

through the web application. We combine the data received from multiple gateways to jointly decode the corrupted packets with LT code [17]. Each LoRa client is equipped with a 16000mAh portable charger and a 4GB TF card to record the necessary information (e.g., transmitted packets) as the ground truth. LoRa clients are placed on the tree.

Metrics. We use the two application level metrics introduced in Section IV (e.g., communication range M and lifetime T) to evaluate the overall performance of eLoRa. The communication range is recorded as the longest distance from the end-device to the closet gateway [4] while meeting the reliability requirement R (e.g., $R > 60\%$ [12]). To measure end-device's lifetime, we modify the chip driver (i.e., rf95 [23]) of Dragino LoRa Shield to record the fractions of time the radio is in receiving, transmitting, and idle mode. Then, we compute projected lifetimes using Eq. (1) and current draws from the rf95 data sheet [23], assuming batteries constantly supply 2000mAh at 3V. The lifetime is estimated based on the energy consumption recorded within one week, instead of actually running out of the battery.

Requirements. We consider a data collection scenario [16] in LoRaWAN that maximize end-device's lifetime while providing a certain communication range.

$$\begin{aligned} \max \quad & T_n \\ s.t. \quad & M_n > 600m, R_n > 60\%, L_n < 5h, n \in S \end{aligned} \tag{18}$$

eLoRa solves Eq. (18) at run-time to determine the optimized parameters. If there are no solutions, eLoRa picks the maximized T_n with the other constraints decreased step by step (e.g., decreasing communication range with 100m per step).

Methodology. We compare eLoRa with two existing approaches: 1) the default LoRaWAN [1]; 2) the LoRaWAN with data recovery code DaRe [6]. The parameter settings of default LoRaWAN are shown in Table 2. We note that $T_{cc} = T_{pkt}/T$, where T_{pkt} actually denotes the time for transmitting and retransmitting the packets. We implement eLoRa and DaRe as the 2.5 layer on top of the LoRaWAN respectively. For Class A and Class B, the approaches are denoted as A-DaRe, B-DaRe, A-eLoRa(wG), and B-eLoRa(wG) (parameter settings S1 and S3 in Table 2 are used for Class A and Class B). For DaRe we set the coding rate $R = 0.5$, window size $W = 8$ and degree $d = 0.83$ according to [6]. We craft an eLoRa

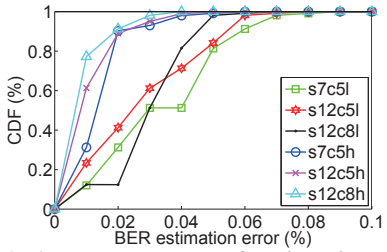


Fig. 6: Absolute errors of estimating BER.

version without the combinations of multiple gateways, e.g., A-eLoRa(woG) and B-eLoRa(woG). We evaluate the model accuracy in Section V-B and the overall performance of eLoRa in Section V-C.

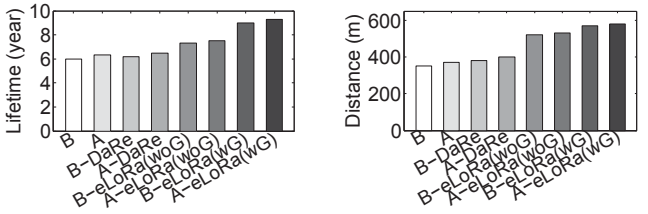
We also evaluate the detailed performance of eLoRa under static conditions and dynamic conditions in Section V-D. We use the parameter settings shown in Table 2, and fix one of the parameters while varying other parameters. We use the USRP [28] to generate controllable interference patterns on the sub-1 GHz band. To generate dynamic link qualities scenarios, we place the USRP near the gateway to interfere the packet reception. The interference fraction is computed as the ratio of the interference duration and the total duration (i.e., 100 minutes). Smaller interference fraction indicates a more frequently changing channel. The experiments are conducted 10 times and the results are averaged.

The running overhead of the optimization depends largely on the search space of the parameters, i.e., v_n is vector contains the parameters to be optimized as stated in Section IV (s, c, b, P_t, x_n, α). Note that the search space of the PHY-layer parameters are relatively fixed, e.g., the range of the spreading factor s is from 6 to 12 with the step one, and the bandwidth b is within 125kHz, 250kHz, and 500kHz. Then the optimization overhead is highly related to the range of α and the resolution of the enumerating block size x_n. We fix other parameters while varying the range of α or the resolution of x_n to evaluate the running overhead when optimizing the lifetime, e.g., the resolution of x_n is set to five bytes, and the maximum α is set to 13. The results of optimizing the communication range are similar. The requirements are the same to the Eq. 18. The results are shown in Section V-E.

B. Model Validation

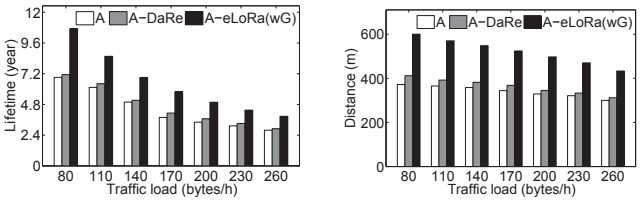
Fig. 6 shows the absolute error of estimating BER in terms of SNR, SF and CR. The label $sxcy[l,h]$ denotes the parameter settings of SF x, CR y and with low SNR l (-10dB) or high SNR h (2dB). Results show that the error is 0.25% on average under different SNR, SF and CR. We note that the estimating error under high SNR (e.g., 0.156%) is a little bit smaller than under low SNR (e.g., 0.35%). Because under low SNR the communication link is unreliable and more unpredictable factors may manifest, resulting a bit higher estimating error.

Table 3 presents the accuracy of estimating high level metrics (i.e., reliability R and lifetime T) under different parameter settings shown in Table 2. In addition to the static parameter settings, we also perform each parameter setting with eLoRa enabled. We let eLoRa to adapt the cross-layer



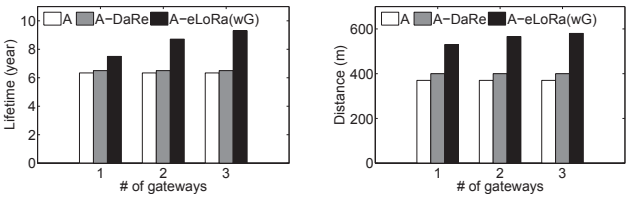
(a) Lifetime. (b) Communication range.

Fig. 7: Overall performance comparison.



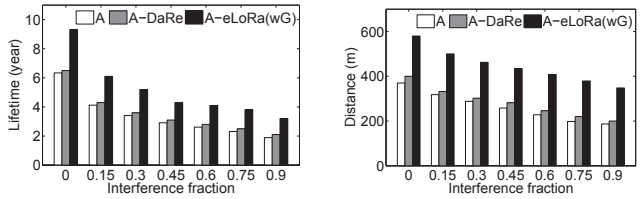
(a) Lifetime. (b) Communication range.

Fig. 9: Impact of traffic load.



(a) Lifetime. (b) Communication range.

Fig. 8: Impact of the number of gateways.



(a) Lifetime. (b) Communication range.

Fig. 10: Impact of dynamic interference.

parameters every 20 minutes. We see that eLoRa achieves high accuracy in reliability (0.98% errors on average), lifetime (0.138 years errors on average) and latency (0.126 hours errors on average) under different parameter settings.

C. Overall Performance

Fig. 7 shows the overall performance improvements of eLoRa comparing with DaRe and LoRaWAN. DaRe and eLoRa are used as the 2.5 layer protocol of LoRaWAN MAC respectively, and are denoted as A+DaRe, B+DaRe, A+eLoRa/wG, and B+eLoRa/wG. We also craft a eLoRa version without the combinations of multiple gateways, e.g., A+eLoRa/woG and B+eLoRa/woG. Results show that eLoRa achieves the best performance over other approaches for both the Class A and the Class B. For example, when combining multiple gateways, eLoRa increases the communication range and the expected lifetime of the default LoRaWAN by 55.7% and 46.6% (both are averaged in Class A and Class B). Note that the performance improvement of DaRe for LoRaWAN is subtle (e.g., 8.1% and 2.5% improvements in terms of communication range and lifetime) due to the default conservative parameter settings and the low decoding efficiency in the pattern of bursty packet loss. Different from DaRe, even without the combination of multiple gateways, eLoRa achieves better performance than the default LoRaWAN by 43.2% and 18.3% in terms of the communication range and lifetime. Because eLoRa utilizes rateless codes to recover the lost packets under any patterns (unlike DaRe that is not robust to the bursty packet loss), and further optimizes the performance by adjusting the parameters from both the PHY-layer (e.g., SF) and link layer (e.g., block length). A-eLoRa(wG) achieves longer lifetime than B-eLoRa(wG) by 3.4% on average. Because in Class B more idle listening windows are opened for receiving packets from the gateway, resulting in more energy consumption.

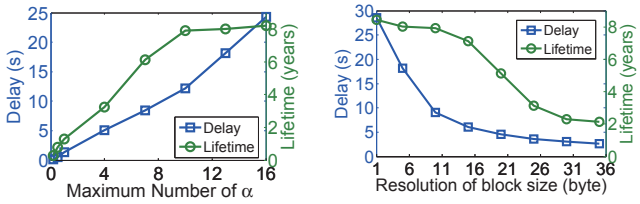
Note that the limited communication range we achieved in this experiment is due to the hardware constraints of LoRa Shield and LoRa gateway [14]. For example, the gain of the antenna we used is only 3dBi [14]. With high-gain antenna, it is possible to achieve at least 5km communication range.

D. Detailed Performance

We evaluate the detailed performance improvement of eLoRa in two conditions: static condition where the link quality keeps stable and dynamic condition where the link quality is changed dynamically.

1) Static condition: Fig. 8 shows the impact of how the communication range and lifetime vary with the increasing number of gateways. We can see that the improvement of eLoRa keeps steady. With more gateways eLoRa can achieve better performance than LoRaWAN and DaRe, e.g., 45.3% and 43.1% improvements on average over DaRe in terms of the communication range and lifetime when using three gateways. Note that the original design of DaRe and LoRaWAN without any modifications do not support the combination of multiple gateways, therefore their performance keeps the same. Even with little modifications, they can only rely on receiving a completely correct whole packet to improve the performance. In eLoRa, correct sub-packets from multiple gateways are utilized and jointly decoded. Therefore, fewer packets are needed and better performance is achieved.

Fig. 9 shows the impact of traffic load on the overall performance. Results show that eLoRa still achieves the best performance over LoRaWAN and DaRe with the increasing traffic load. Specifically, with smaller traffic load eLoRa achieves larger performance improvement, e.g., increasing the communication range and lifetime of DaRe by 45.6% and 50.3% when traffic load is 80 bytes/h. We note that eLoRa can meet the communication range requirement when the traffic load is 80 bytes/h and achieve the longest distance and lifetime at larger traffic load, while DaRe and LoRaWAN can not meet the communication range requirement all the time. Because with the use of rateless coding and cross-layer



(a) Impact of α. (b) Impact of block size x_n.

Fig. 11: Overhead of performing optimization.

Table 4: Impact of optimizing the lifetime.

R>60%, L<5h								
M (m)	200	400	600	800	1000	1200	1400	1600
Opti_T	7.13	5.34	4.02	NA	NA	NA	NA	NA
M>600m, L<5h								
R (%)	20	30	40	50	60	70	80	90
Opti_T	8.5	8.02	7.76	5.89	4.23	2.13	NA	NA
M>600m, R>60%								
L (h)	3	5	7	9	11	13	15	17
Opti_T	NA	6.2	7.4	8.73	9.31	10.02	12.22	12.23

parameter optimizations, eLoRa can adapt the parameters to network traffic load.

2) Dynamic condition: Fig. 10 depicts the overall performance under dynamic link qualities. We see that eLoRa can achieve the longest distance and lifetime over Class A and DaRe under different interference patterns. Specifically, eLoRa improves the communication range and lifetime of DaRe by 74% and 52.2% at most frequently changing interference. Because eLoRa can adapt the parameters to the link quality changes timely and therefore keeps the high performance. While for LoRaWAN and DaRe, their performance is drastically degraded when the interferer is active due to their fixed parameter settings.

E. Overhead of Optimization

Fig. 11 presents the overhead of performing optimizations considering the parameter searching space. Fig. 11(a) shows that the running time increases when the range of α is larger. When the range is bigger than ten, the increase of the optimized lifetime becomes small, which indicates that setting α to ten can satisfy most optimization scenarios. The overhead is thus up to 12.13s. Fig. 11(b) shows that with the coarse-grained resolution of x_n the running time is small. We note that when the resolution is lower than ten bytes, the optimization amount of increased lifetime is small (about 5.1% increase). Therefore, in our evaluation, the resolution of x_n is selected to ten bytes and the running overhead is up to 9.06s. The optimization is multiple order of seconds, but considering that the common uplink transmission limitations of LoRaWAN applications are 30 seconds on-air-time per day per device [16], the overhead is negligible. Besides, the optimization only performed when the requirements are violated which happens rarely.

F. Impact of Optimization Constraints

Table 4 and Table 5 show the impact of constraint settings on the optimization results. Developers can choose proper constraint settings from the Table when deploying eLoRa on practical scenarios. Although the concrete parameters may differ because of the utilized hardware, the trend is the same. Note that in this experiment the constraint is strictly fixed and there will be no results when constraints can not be satisfied. Table 4 shows the impact of varying constraints of communication distance (M), packet reception ratio (R) and transmission latency (L) on the optimization results of the lifetime (T). While Table 5 presents the results of optimizing M while varying other three constraints, T, R and L. The results of the above two tables both show that it is more likely to have the optimization solutions when the constraints are more relaxed. For example, when optimizing the lifetime T, setting M to be bigger lower than 600m, R to be lower than 60%, and L to be bigger than five hours. It is highly possible to optimize T to four years at least.

Table 5: Impact of optimizing the communication range.

R>60%, L<5h								
T (year)	2	5	8	11	14	17	20	23
Opti_M	822	613	423	198	NA	NA	NA	NA
T>6 years,L<5h								
R (%)	20	30	40	50	60	70	80	90
Opti_M	1322	1123	978	659	345	NA	NA	NA
T>6 years,R>60%								
L (h)	3	5	7	9	11	13	15	17
Opti_M	NA	234	432	578	765	912	1078	1298

VI. Conclusion

In this paper, we propose a practical system, eLoRa, for COTS devices. It utilizes rateless codes and jointly decoding with multiple gateways to extend the communication range and lifetime of LoRaWAN. To further improve the performance of LoRaWAN, eLoRa optimizes parameters from the PHY-layer (e.g., spreading factor) and the link layer (e.g, block length). eLoRa provides an attractive feature that network operators can define application level requirements (e.g., lifetime and communication range). eLoRa periodically checks whether the requirements are violated, and automatically carries out the parameter optimization based on the network model and the monitored network states. We implement eLoRa on COTS LoRa devices, and conduct extensive experiments on outdoor testbed to evaluate the effectiveness of eLoRa. Results show that eLoRa can effectively improve the communication range of DaRe and LoRaWAN by 43.2% and 55.7% with packet reception ratio higher than 60%, and increase the expected lifetime of DaRe and LoRaWAN by 18.3% and 46.6%.

In the future, there are multiple directions to explore. First, we would like to optimize the degree distribution of LT code given the multiple gateway combination approach. Second, we would like to design a better gateway deployment strategy to cover more LoRa clients.

References

[1] "LoRaWAN 1.1 specification," https://www.lora-alliance.org/resource-hub/lorawantm-specification-v11.
[2] "NB-IoT," http://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete.
[3] "Sigfox," https://www.sigfox.com.
[4] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: Exploiting Geographical Diversity

Through Coherent Combining in Low-power Wide-area Networks," in *Proc. of ACM/IEEE IPSN*, 2018.
[5] R. Eletreby, D. Zhang, and et. al., "Empowering Low-Power Wide Area Networks in Urban Settings," in *Proc. of ACM Sigcomm*, 2017.
[6] P. Marcelis, V. Rao, and R. Prasad, "DaRe: Data Recovery through Application Layer Coding for LoRaWAN," in *Proc. of ACM/IEEE IoTDI*, 2017.
[7] G. Chaojie, T. Rui, L. Xin, and N. Dusit, "One-Hop Out-of-Band Control Planes for Low-Power Multi-Hop Wireless Networks," in *Proc. of IEEE INFOCOM*, 2018.
[8] M. Bor and U. Roedig, "LoRa Transmission Parameter Selection," in *Proc. of IEEE DCOSS*, 2017.
[9] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating Inter-network Interference in LoRa Networks," in *Proc. of EWSNs*, 2017.
[10] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?" in *Proc. of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2016.
[11] F. V. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale lorawan networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.
[12] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, pp. 1–16, 2017.
[13] A. Augustin, J. Yi, and etc, "A Study of LoRa: Long Range Low Power Networks for the Internet of Things," *Sensors*, vol. 16, no. 9, 2016.
[14] "Dragino Lora Shield and Gateway," http://www.dragino.com/.
[15] M. Luby, "LT Codes," in *Proc. of IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.
[16] "LoRa Technology Is Connecting Our Smart Planet," https://www.semtech.com/technology/lora/lora-applications.
[17] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi, "SYNAPSE++: Code Dissemination in Wireless Sensor Networks Using Fountain Codes," *IEEE TMC*, vol. 9, no. 12, pp. 1749–1765, 2010.
[18] W. Du, Z. Li, J. C. Liando, and M. Li, "From Rateless to Distanceless: Enabling Sparse Sensor Network Deployment in Large Areas," *IEEE/ACM ToN*, vol. 24, no. 4, pp. 2498–2511, 2016.
[19] M. Zlmmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "pTUNES: Runtime parameter adaptation for low-power MAC protocols," in *Proc. of ACM/IEEE IPSN*, 2012.
[20] D. J. C. MacKay, "Fountain codes," *IEE Proceedings - Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.
[21] Semtech, "An1200.22: Lora modulation basics," 2015.
[22] W. Dong, C. Chen, X. Liu, Y. He, Y. Liu, J. Bu, and X. Xu, "Dynamic Packet Length Control in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1172–1181, 2014.
[23] "RFM95," http://www.hoperf.com/upload/rf/RFM95_96_97_98W.pdf.
[24] W. Dong, J. Yu, and P. Zhang, "Exploiting Error Estimating Codes for Packet Length Adaptation in Low-Power Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 8, pp. 1601–1614, 2015.
[25] B. Reynders, W. Meert, and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," in *Proc. of International Conference on Telecommunications (ICT)*, 2016.
[26] G. Ferreira, B. Jesus, J. Vieira, and A. J. Pinho, "Random block-angular matrices for distributed data storage," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.
[27] "ECLiPSe," http://eclipseclp.org/.
[28] "USRP N210," https://www.ettus.com/.