WIP: Collaborative Approaches to Mitigate Links of Variable Quality in LoRa Networks

Henrik Rosenberg
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
henrik.rosenberg@alumni.tu-clausthal.de

Andreas Reinhardt
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

Abstract—**LoRa has become a de-facto standard for long-range communication networks with low data rate requirements. Despite the robustness of its employed chirp spread spectrum modulation, however, LoRa links covering long distances can be prone to errors and losses. A common reaction to such link conditions is to increase the symbol durations and thus their chance of a correct reception. In this paper, we follow an alternative approach to mitigate lossy links by adding redundant information to LoRa transmissions. We investigate the potentials of both local and collaborative methods to increase the network's overall data delivery. By means of simulation studies and a practical evaluation, for which we have implemented the designed techniques on actual LoRa hardware devices, we show that significant improvements are possible in a real-world deployment.**

I. INTRODUCTION

Over the past years, LoRa has established itself as a widely used protocol for long-range wireless communication. Its definition of physical layer functionalities is complemented by LoRaWAN [1], which specifies the required features on the link layer and above, in order to accommodate long-range data transfers. LoRaWAN networks are designed to be set up in star topologies, in which one or more *gateways* act as the destination for the traffic from all deployed *end-devices*. Gateways are always-on devices that can receive traffic at any time. As a result thereof, individual LoRa devices can operate on low energy budgets by keeping their wireless transceivers in a low-power sleep state unless they are transmitting frames or expecting downlink messages. Such end-devices, used to periodically transmit small data items (e.g., sensor readings) to the gateway, are termed *Class A* devices and constitute the device type under investigation in this work.

A fundamental difference between LoRaWAN and other networking technologies for embedded sensing devices (e.g., IEEE 802.15.4) is that LoRaWAN end-devices are commonly not designed to accommodate multi-hop traffic. While this renders sophisticated MAC protocol mechanisms unnecessary, given that end-devices can simply wake up at any time to transmit their data, it imposes the limitation that end-devices cannot overhear traffic from other end-devices. Techniques widely used in multi-hop networks, such as repeating data on behalf of neighboring nodes, are thus inapplicable by default.

Unless LoRa networks deviate from this rule (e.g., by using repeaters or forwarding gateways [2, 3]), end-devices can only detect the correct arrival of their messages at the gateway through acknowledgments, and re-transmit data in absence of such confirmations.

Setting up a LoRa network is hence generally preceded by an empirical assessment of the link qualities [4], in order to ensure connectivity of all end-devices to at least one gateway. Attempts at improving the reliability of LoRa links under the presence of interference and fluctuating link characteristics are presented in both the LoRaWAN specification [1] and related work. The former states that end-devices can negotiate different LoRa parameters with the gateway (through the exchange of *LinkCheck* or *LinkADR* messages) in an attempt to increase reliability. The authors of [5] present a methodological study on the effects of local coding. The introduced latency and added information gains of introducing message redundancy were investigated in more depth in [6]. Local coding is also applied in [7], where lossy data compression mechanisms and data recovery schemes based on reinforcement learning are being applied to determine optimal transmission policies.

In this paper, we study possibilities to overcome links with (temporarily) lossy characteristics in a computationally inexpensive way. We consider both the local operation (i.e., retransmitting previously sent data in subsequent transfers) as well as a collaborative approach in which co-located nodes, able to overhear each other's traffic, combine overheard messages with their own data. Through simulation studies and real-world tests, we demonstrate the viability of the presented mechanisms and the ensuing data reception rate improvements.

II. A STUDY ON LoRa LINK VARIABILITY

Links in LoRa networks are generally characterized by a high reliability, which is largely a result of the careful probing of channel conditions and suitable node locations prior to the deployment of end-devices. With typical Frame Reception Rates (FRRs) on the order of 95 % to 99 % [8, 9], packet losses can be easily compensated for by simply retransmitting lost messages. The underlying phenomenon for message losses, namely the link variability in LoRa networks, has seen less scientific consideration, however. To investigate this issue in more depth, we have set up and run a practical experiment to monitor the variability of LoRa links across our university
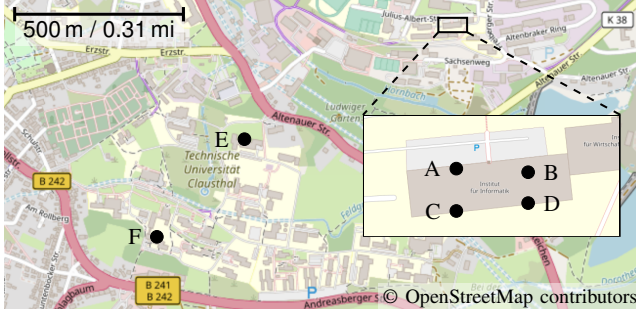
Fig. 1. Topology overview of the campus LoRa test network.

TABLE I
AVERAGE FRR VALUES OBSERVED DURING OUR EXPERIMENT RUNS.

Sender	Receiver					
	A	B	C	D	E	F
A	—	100%	100%	100%	3%	0%
B	100%	—	100%	100%	100%	22%
C	100%	100%	—	100%	94%	9%
D	100%	100%	100%	—	0%	19%
E	0%	13%	100%	0%	—	96%
F	0%	0%	40%	1%	97%	—

campus. The LoRa network was composed of six LoRa end-devices[1], labeled A–F in Fig. 1. In order to track the devices' FRRs, each of the LoRa end-devices was instrumented with a Raspberry Pi single-board computer, interfaced to the university's campus Ethernet for remote access. End-devices were configured to transmit packets at a rate of $1\,\mathrm{frame/min}$ in an attempt to collect as many samples as possibly without jamming the wireless channel. The LoRa spreading factor was set to SF 12 with a transmit power of $20\,\mathrm{dBm}$ for maximal range, in conjunction with a forward error connection coding rate of 4/5 and bandwidth of $125\,\mathrm{kHz}$. All experiments were setup to have a duration that is a multiple of $24\,\mathrm{h}$ to average out periodical interferences that follow a diurnal pattern. All nodes were operated on external power supplies, and did not turn their radio off during our experiments. Note that we intentionally did not seek to optimize node locations to achieve perfect FRRs. Rather than this, we deliberately accepted below-maximum reception rates in order to observe the link variability better.

The resulting FRR values between each pair of end-devices are given in Table I. Largely differing frame reception rates can be observed, and some links (e.g., between nodes B and E) are highly asymmetric. It is also apparent that co-located nodes A–D feature perfect FRRs between each other, yet their capabilities of delivering data to nodes E and F differ strongly. Essentially, if F were the destination for data from nodes A–D, the partially low FRRs clearly highlight the need for additional means to recover as many data as possible.

III. MESSAGE REDUNDANCY IN LORA NETWORKS

The stark link variabilities observed in Sec. II have motivated us to investigate ways to increase the reliability of LoRa traffic. Retransmissions of data messages represent the simplest form of attempting greater Data Reception Rates (DRRs). However, both legal constraints (limiting a node's maximum duty cycle) and the uncertainty whether a re-transmitted message will experience more favorable channel conditions limit the efficacy of this method. To overcome the aforementioned limitations, we consider three possible types of adding redundancy in this study; all of which have

[1]Arduino Uno devices with a Dragino LoRa Shield (based on the HopeRF RFM95W transceiver and operating in the $868\,\mathrm{MHz}$ band).

been selected to keep the required computational power and memory demand for buffering previous/overheard frames low.

Local Data Retransmissions (LDR): An end-device appends its most recently sent frame to its next transmitted frame, regardless of whether it had been acknowledged or not. This method mimics the behavior of LoRa networks in which a best-effort delivery is targeted through sending each frame twice, and serves as our baseline for comparison.

Neighbor Data Retransmissions (NDR): An end-device appends its most recently overheard frame (i.e., both header and payload) to its next transmitted frame.

Combined Data Retransmissions (CDR): An end-device appends a bitwise exclusive-OR (XOR) combination of its most recently overheard frame and its most recently sent frame to its next transmitted frame. In other words, it combines both messages through the application of Network Coding (NC).

The aforementioned options imply that at least two senders are close enough to each other be overhearing each others' frames. As visible in Table I, this is the case between several end-devices in our campus scenario. However, LoRa *Class A* end-devices are generally configured to access the channel at random, without trying to overhear each other's transmissions. As such, LoRa devices are generally unaware of each other's transmissions. This only makes the application of LDR easily possible because it only relies on buffering the most recently transmitted frame. In order to run redundancy methods that rely on the overhearing of a node's neighbors, the nodes must be enabled to do so. In our tests, we have thus kept all LoRa end-devices in an always-on configuration to meet this requirement. We wish to point out two methods that to allow nodes to duty-cycle their LoRa transceivers, however. Nodes can either indicate their wake-up interval via a newly added header field (cf. Sec. III-A) or, alternatively, a TDMA-based channel access can be used [10]. We leave a dedicated investigation of such methods for future work, as they are beyond the scope of this work-in-progress paper.

A. Required Modifications to LoRa Frames

Besides the regular LoRaWAN Frame Header (FHDR), as defined in [1], a new auxiliary header is needed in order to inform the receiving device about the presence of (possibly encoded) data beyond the actual frame content. As such, we insert a Redundancy Header (RHDR) into the existing frame structure after the FHDR, as shown in Table II. Its

Frame part	Frame Header (FHDR)			Redundancy Header (RHDR)			Frame Payload	Redundant Data (RDat)	
Field type	DevAddr	FCtrl	FCnt	ROpts	PayloadLen	TNextTX *(optional)*	application specific	RFHDR	RPayload
Length (bytes)	4	1	2	1	1	3 *(if present)*	*e.g. 10*	7	*e.g. 10*

presence is indicated by means of a *RFU* ("reserved for future usage") bit in the MAC Header. The fields in the RHDR indicate whether a coded message is appended, as well as the type of coding that has been applied (i.e., LDR, NDR, or CDR). Moreover, the length of the regular frame payload is given, such that the receiver can compute the frame offset at which the Redundant Data (RDat) begins. Optionally, the timing of the next transmission can be included in RHDR, so neighboring nodes can adjust their next wake-up interval accordingly for overhearing. Lastly, RDat contains both the FHDR and the payload, both of which are encoded as defined in RHDR. We would like to note that frame payload and RPayload do not necessarily need to be of the same size.

B. Decoding Received Messages

Through extracting the type of redundancy from the RHDR, a gateway can easily decode incoming transmissions in case of LDR and NDR, where RDat is transmitted in plain. When a frame was sent using CDR (i.e., as the XOR combination of two frames), at least one of the input frames to the XOR operator is needed in full to decode the corresponding other frame. Often, the previously transmitted frame (whose *FCnt* is one less than the current value) of the same end-device is known to the receiver, so it can reconstruct the encoded frame of the sender's neighbor. Only when both a station's previous frame and its neighbor's most recently sent frame are unknown to the gateway, their data remains unrecoverable.

IV. EVALUATION

As follows, we confirm the viability of the presented approaches through simulation studies and practical experiments, again based on the LoRa campus network introduced in Sec. II.

A. Simulation Study

Let us first consider a simple three-node topology, in which we assume that two of the nodes, labeled i and j, are spatially co-located and can overhear each other's transmissions. By varying the FRRs between both nodes and the gateway g, we determine under which channel conditions the addition of redundant message parts is beneficial. The results of our simulation studies are shown in Figs. 2a to 2c, in which the FRRs of both senders are varied along the axes, and the overall DRR at the gateway is indicated by means of the plot's shading. It can be observed that the impact of the methods to add redundancies depends on the FRRs of the senders to the gateway. If both end-devices already have very good connectivity to the gateway, all approaches perform equally well, chiefly because there is not much room for improvement. Retransmitting overheard data (NDR) primarily excels when one node has an excellent connection to the gateway, while the other one does not. A remarkable insight, however, is that repeating own frames (LDR) is *never* better (in terms of the DRR) than encoding overheard messages into an end-device's own transmissions (i.e., applying NDR) in this scenario.

The use of CDR, shown in Fig. 2c, shows the greatest area in which high DRRs can be observed. At the same time, however, it also highlights the principal disadvantage of CDR: When both end-devices have a poor link quality towards gateway, many of the encoded messages cannot be correctly decoded due to not knowing the message they have been encoded with. This can be seen as well in the direct comparison between CDR and NDR in Fig. 2d. Only in the top right quadrant of the plot, CDR leads to more data being received by the gateway. As such, it becomes clear why knowledge of each end-devices FRR to the gateway is crucial to determine which redundancy-adding method (if any at all) is most effective.

The *break-even line* (in which both NDR and CDR perform equally well) can be approximated by a circle, centered around the top right corner of the diagram (where $FRR_{i,g} = FRR_{j,g} = 100\%$) with a radius of 0.669. This simplification allows the decision which redundancy method to apply to be made on individual low-power LoRa devices, by inserting their own $FRR_{i,g}$ and the neighbor's $FRR_{j,g}$ to compute the distance to the upper-right corner:

$$d^2 = (1 - FRR_{i,g})^2 + (1 - FRR_{j,g})^2$$

If $d < 0.669$, CDR is being used; otherwise, the end-device applies NDR only. Learning about a neighbor's FRR to the gateway could, e.g., be realized by a periodic gateway broadcast containing all FRRs it has been keeping track of.

B. Practical Experimentation Results

Next, we evaluate the efficacy of the redundancy adding methods using our practical experimental setup, composed of six nodes as described in Sec. II. For each experimental run, two of the nodes act as the senders i and j. The remaining four end-devices in the network have been configured to receive all wireless traffic for analysis; they can thus be thought of as gateways. Given that senders were selected to be spatially co-located and thus capable of overhearing each other, the nodes can work together using one of the introduced methods. For different pairs of sending devices and gateways, we show the observed results in Fig. 3. We did not plot node combinations where both senders had an almost perfect link to the gateway or none at all. We also left out ones where the senders are not co-located or very similar results are already plotted. For every practical experiment run, the bars represent the observed FRRs from both end-devices i and j to the gateway g, as well as the predicted DRR (according to the simulations in Sec. IV-A) and the DRRs measured in practice.
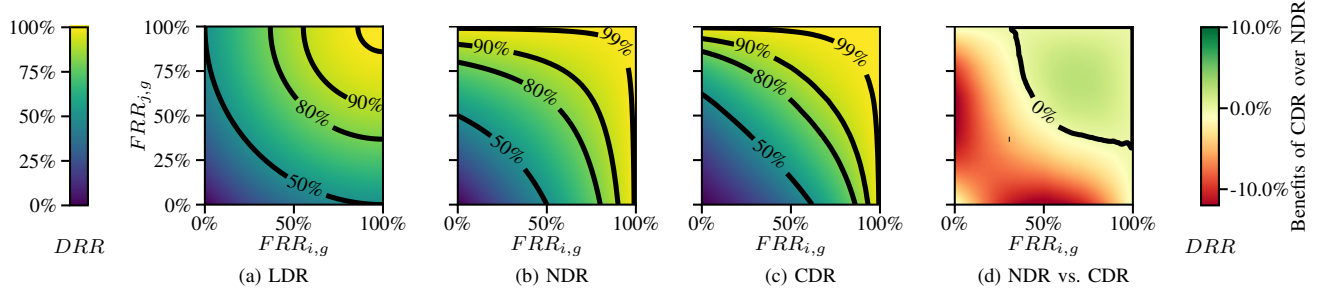
Fig. 2. Simulation results. FRRs of the two senders i and j to the gateway g on the x- and y-axes; DRR at the gateway indicated by color shading.
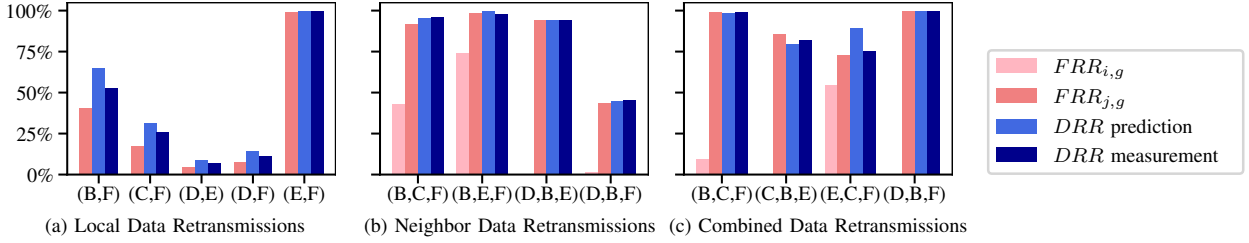

Fig. 3. Experimental results. Labels below each group of bars specify the involved nodes (based on the topology shown in Fig. 1) in the form ($[i,]j,g$). In each of the experiments, all sending end-devices were configured to transmit at least 7200 frames.

Fig. 3a, which serves as the baseline for our comparison, already indicates that a simple retransmission of each outbound frame already has a positive effect on the DRR over sending them only once. Through the application of NDR and CDR, much greater improvements could be observed. This can be attributed to the strong FRR imbalance between nodes i and j and the gateway. As such, a frequent observation in experiments using NDR was that the frames from one of the senders almost never reached the gateway. In these cases, the added redundancy could be shown to achieve an enormous increase in the DRR (Figs. 3b and 3c). Even when both senders face similar channel conditions to the gateway, our experimental results also demonstrate that adding redundancy still increases the DRR. When only one sender can reliably reach the gateway, CDR provides the greatest improvements, as almost all XOR combinations can be decoded. The application of NDR and/or CDR has thus been demonstrated to be a viable method to increase the network's DRR.

V. SUMMARY AND CONCLUSIONS

Adding redundancy to transmitted data is a well established approach to enable the recovery of data from messages that have experienced collisions or packet losses. In this work, we have investigated the positive impact when LoRa end-devices encode previously sent data and/or frames of immediate neighbors into their own transmissions. Through simulation studies and practical experiments, we have demonstrated that significant reliability improvements can be accomplished. This way, the network can effectively recover from (temporarily) poor links to the gateway. Overall, we could show to that significantly improved DRRs in real-world LoRa networks can be reached, and can thus be considered a key enabling element for

more reliable and robust long-range communication between embedded sensing devices in the future.

REFERENCES

[1] "LoRaWAN specification v1.1." [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v11

[2] O. Flauzac, J. Herard, F. Nolot, and P. Cola, "A low power LoRa-LoRaWAN relay function with a single input, single output device," in *International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2020.

[3] M. H. Dwijaksara, W. Sook Jeon, and D. G. Jeong, "Multihop gateway-to-gateway communication protocol for LoRa networks," in *IEEE International Conference on Industrial Technology (ICIT)*, 2019.

[4] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology," in *14th International Conference on ITS Telecommunications (ITST)*, 2015.

[5] P. J. Marcelis, V. S. Rao, and R. V. Prasad, "DaRe: Data recovery through application layer coding for LoRaWAN," in *IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2017.

[6] M. Sandell and U. Raza, "Application layer coding for IoT: Benefits, limitations, and implementation aspects," *CoRR abs/1706.01271*, 2017.

[7] C. Pielli, C. Stefanovic, P. Popovski, and M. Zorzi, "Joint retransmission, compression and channel coding for data fidelity under energy constraints," *CoRR abs/1706.09183*, 2017.

[8] 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.

[9] J. Petäjäjärvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen, and J. Iinatti, "Evaluation of LoRa LPWAN technology for indoor remote health and wellbeing monitoring," *International Journal of Wireless Information Networks*, vol. 24, no. 2, 2017.

[10] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the industrial internet of things," *Computer Communications*, vol. 153, 2020.