Downlink Independent Throughput Optimisation in LoRaWAN

Bruno Citoni, Shuja Ansari, Qammer Hussain Abbasi, Muhammad Ali Imran, Sajjad Hussain
School of Engineering
University of Glasgow
Glasgow, Scotland
b.citoni.1@research.gla.ac.uk

***Abstract*—In a LoRaWAN network one of the main reasons of packet outage is the destructive interference that is caused by colliding packets. As the network operates with an ALOHA-like channel access setup, there is no easy way of preventing two or more devices transmitting at the same time, possibly generating interference to each other. Different methods are proposed in literature that can be used to decrease this chance. However, most of them require extensive use of downlink messages coupled with involved algorithms at the network side, often for only a marginal improvement in performance. In this paper we analyse some ways to optimise the Packet Delivery Ratio (PDR) of a LoRaWAN network that can be used when setting up a node or a group of nodes, do not involve downlink and can operate without knowledge of other devices in the same network. These are shown to provide a small boost in performance of maximum 10%, which is akin to that of more complex, downlink-dependant schemes, while decreasing the set up complexity considerably.**

***Index Terms*—LoRaWAN, low-power wide area network (LPWAN), Networking, Throughput, PDR, Quality-of-Service, Collision**

I. Introduction

Internet of Things (IoT) is a set of electronic systems including sensors and actuators, connected wirelessly, that exchange contextual data. IoT finds application in multiple industries, including smart homes and smart city environments. All these usages require a communication protocol that can support multiple connected devices with high scalability, high communication range, and low-power consumption.

LoRaWAN (Long-Range WAN), a low-power wide area network protocol (LPWAN), has gathered attention in recent years as a prominent technology by satisfying these fundamental IoT requirements. In a LoRaWAN network, end devices are typically organised in a star-of-stars topology, having no direct connection with any one gateway. Each gateway forwards to the network server the data it receives from all devices within range. Devices can send data to the network via uplink and receive data via downlink. Downlink requires the data to come through a gateway in range as node-to-node communication is not available with standard LoRaWAN. The same duty cycle limitations that apply to nodes also apply to gateways, so downlink messages need to be sparse. This is also because downlink decreases the overall Quality-of-Service (QoS) of a network by creating additional interference and preventing gateways to receive uplink messages while transmitting. These limits also preclude the use of LoRaWAN for time-critical, low-latency applications, as well as limiting the context each node can have of the network it is a part of. Being organised in a star-of-stars topology means that nodes can not discover other node devices if not through a downlink message from a gateway. To overcome these scalability limits hindering the performance of LoRaWAN, a number of configurable radio parameters can be set to control the maximum achievable communication range, power consumption, and data rate. Among these, Spreading Factor (SF), Coding Rate (CR), Bandwidth (BW) and Transmission Power all have a direct influence on the chance of packets being successfully received and decoded in a LoRaWAN network.

While analysing the scalability of this technology in literature, these parameters are often updated on end devices via periodic downlink messages. This allows the network server to collect data from all connected devices, including their current Signal-to-Noise Ratio (SNR) and Received Signal Strength Indicator (RSSI). This information is processed using different algorithms and then the network informs each device of how [1]–[4] and possibly when [5]–[8] to send the next packet.

While these studies work well in theory, the full impact and the feasibility in terms of power consumption, additional interference and network complexity of the extensive amount of downlink messages required are often disregarded in these simulations.

In this paper, we analyse and test different ways to increase the quality of a network that would be easier to implement in a real-world deployment. By varying transmission parameters on the end devices without relying on downlink information, we start to develop the idea of context unaware and downlink independent throughput optimisation.

The rest of this paper is structured as follows: first, we formalise the collision behaviour in a LoRaWAN network in Section II and discuss related literature in Section III. We then describe the system setup for the simulations in Section IV and the results in Section V. Finally, we summarise the outcomes of this work and conclude the paper in Section VI.

II. LoRaWAN Overview

The main radio transmission parameters of LoRaWAN that can be altered to modify the protocol's performance are:

- Spreading factor (SF), which relates to the number of frequency chirps that are used to modulate the signal. Larger

This research was funded by UK EPSRC grant number EP/R512266/1.

spreading factors increase the SNR and communication range, at the cost of slower transmission and longer Time-on-Air (ToA) for each packet.
- Bandwidth (BW), which is the spectrum over which the chirp spreads.
- Coding Rate (CR), which is used to perform forward error correction techniques.

These three parameters have an influence on the bit rate, which in turn has an influence on the Time-on-Air (ToA) of a packet. ToA is the time it takes a packet to be transmitted on a wireless channel.

The ToA of a packet has a strong correlation with the probability of it being correctly received. If increasing SF and consequently the ToA allows the packet to be received by a gateway more easily, it also increases the chance of collision with other packets, the chance of saturating a receiver's demodulation paths and the chance to go above the allowed duty cycle limits [9].

A. Collision behaviour

LoRaWAN uses an ALOHA-like channel access strategy. Devices are able to transmit at any moment, thus increasing the chance of collision between packets. To try and overcome this issue, LoRa provides frequency hopping capabilities as well as semi-orthogonality between spreading factors. With up to 8 channels enabled, each device can attempt to transmit using a randomly selected central frequency for every transmission. As long as multiple packets arriving simultaneously at a gateway were sent on different channels, they will not collide. This choice is mostly open to the network designer. The only necessity in the EU is that the three channels at 868.10, 868.30, 868.50 MHz are always enabled and that the duty cycle of 1% is respected in each utilised sub-band.

Similarly, a packet can be sent using one of six available SFs, ranging from 7 to 12. Packets that overlap will surely collide if their SF (and channel) are the same, and have a chance to collide, due to the quasi-orthogonality of the spreading factors even when having different SF. On top of this, because of the capture effect, a packet can still be received over the co-SF interference if it is received with a certain power difference with respect to the interference [10].

Unlike channel selection, the spreading factor choice is not as "free", as the SF needs to be selected among those that will grant the packet sufficient range to be received at the nearest gateway. This means that, depending on the network size, nodes near gateways can possibly use any spreading factor, while those farthest away may only have the choice of SF12, as the highest possible receiver sensitivity, achieved by using the highest possible SF, might be needed to cover the distance between the two devices [9].

To sum up, a packet collision will occur when multiple signals overlap in time, were sent with the same SF (and, to a lesser extent, other SFs too) and on the same frequency channel, as shown in Fig.1. When colliding with the same SF, the strongest packet has a chance to be received despite the collision, thanks to the capture effect.

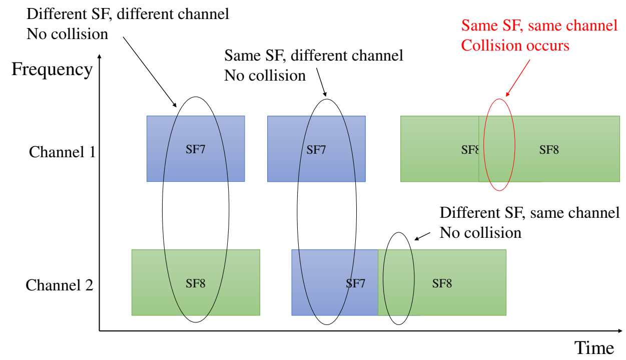

Fig. 1: Visual representation of collision behaviour assuming 2 frequency channels and 2 spreading factors. Despite time overlap, only the last instance causes a collision

III. Literature review

Much effort has been made to increase LoRaWAN QoS by minimising the chance of packet collision in literature. These works are generally focused on 4 main topics:

- Hardware based schemes
- Channel Access schemes
- Adaptive Data Rate (ADR) based schemes
- Downlink independent schemes

In [11] the authors developed Choir, a system that aims to disentangle and decode interfering transmissions at a single-antenna gateway by leveraging hardware imperfections of low-cost LP-WAN clients. It also expands the range of each node by exploiting the correlation of the sensed data. Overall, Choir improves the analysed network's throughput by 6.84× and expands communication range of the devices by 2.65×. Charm [12] is another similar system that allows LoRaWAN gateways to combine their received signals in the cloud, to try and decode signals that are too weak to be decoded by any individual gateway. These methods clearly increase complexity and cost of network deployments.

Next, the channel access schemes are modifications to the ALOHA mechanism that LoRaWAN uses and is at the root of its scalability problems. [7] proposed using a Slotted-ALOHA approach rather than the pure ALOHA for LoRaWAN. This new framework is built over the existing firmware and is based on creating time slots for uplink and downlink for all devices, which are also time synchronised. [8] performs a similar improvement by developing a new MAC layer, RS-LoRa, that aims to improve scalability and reliability by two-step lightweight scheduling implemented through downlink beacons synchronising and informing nodes. Similar ideas, exploiting TDMA (time division multiple access) and timeslot scheduling are developed in [5], [6]. Crucially, all the proposed advancements in this category require the use of downlink frames to instruct the nodes as to when to transmit their next packet and with what parameters. Ultimately, they require a considerable amount of planning and additional resources to what it would normally be needed to simply connect a node to a LoRaWAN network.

In a similar vein, a number of works try to reach the same goal by improving the standard ADR mechanism that is present in both the network server and on the end devices. In [13], the authors developed FADR, a Fair Adaptive Data Rate algorithm designed to alleviate the near-far problem and grant a fairer chance to be received to each node, regardless of its position. [14] develop EARN, a greedy ADR mechanism taking into account the effect of the coding rate on the packet collision, and leveraging the capture effect to increase survivability of packets. [15] developed a contention-aware ADR, capable of dynamically calculating the optimal SF distribution for the network, which is then sent to each node via downlink to increase the network throughput. [1]–[4] all present similar concepts, analysing the existing ADR and proposing improvements that, like the standard algorithm, require downlink capabilities as well as increased network complexity.

Finally, there are downlink independent ways to try and increase throughput. Among these, there are re-transmission schemes, which involve transmitting the same packet multiple times during a single transmission interval, to maximise the chance that at least one transmission is received. One such method is proposed and validated through numerical simulations in [16]. To the best of our knowledge, no other work tries to achieve an increased QoS by simply varying the transmission interval, start time and channel hopping algorithm of node devices.

IV. System Model

A MATLAB simulation has been developed to recreate the transmission behaviour of LoRaWAN and explore the collision probability exclusively. This involves making sure no packet failure is possible for any reason but packet collision. Firstly, no node can ever be out of range, by selecting the size of network deployment based on the maximum range of a device operating with SF12 and only subjected to the deterministic log-distance path loss as laid out by [17]. Similarly, the loss due to the limited number of parallel reception paths on a standard receiver is ignored, assuming that the gateway can decode an infinite number of packets simultaneously.

We consider the Packet Delivery Ratio (PDR) as the ratio of the number of packets that were received correctly (i.e. that did not collide) over the total amount of packets sen in percentage.

By knowing in advance the full radio parameters of all the nodes in the network at any given time it would be trivial to create a schedule that ensures no packet will ever collide. By careful selection of SFs, channel frequencies and start times, such a network would yield a theoretical 0% collision rate (before factoring in external interference) even without downlink.

As we are aiming to test methods that are feasible in a real-world deployment, we will not consider this option and instead decide on which metrics and information a single node is aware of, without the access to downlink transmission. We can assume that each node would always know its own settings, including SF, Tx interval and start up time.

We then decided to split the use cases in two: one involving a full network of our design and another where a single device is connected to an already existing network that we don't have any information about, akin to simply turning on a single node. We then tried to find ways to select the parameters we are in control of to maximise the PDR of the network in the first case, and of our lone device in the second.

In both scenarios, the initial network behaviour is as follows. 1000 nodes are uniformly distributed around a central gateway in a circle with radius 6473 m. Propagation is modelled according to the log-distance propagation model:

$$PL = PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g, \qquad (1)$$

with:
PL_0 (dB) is the path loss at the reference distance d_0
d is the distance between receiver and transmitter;
d_0 is the reference distance;
γ is the path loss exponent;
X_g is a variable used to model slow and fast fading.

A radius of 6473 m ensures that all devices will be within range with the parameters used.

Each node transmits 10 bytes of data with 13 bytes of mandatory LoRaWAN header, giving Time-On-Air for a SF12 packet of 1482.8 ms.

The time between transmissions (Tx Interval) is set for all devices to 300 s, to avoid issues due to the 1% duty cycle restrictions. Each node is set to send a total of 20 packets during a simulation run.

Spreading Factors are allocated based on a node's distance from the gateway, with each device operating on a fixed SF which is the lowest it can possibly adopt. No downlink transmissions or uplink re-transmissions are taken into account. The network also operates in ALOHA fashion, hence ignoring the capture effect and the semi-orthogonality of the spreading factors.

Each node will also have a random start time which is between 0 and Tx Interval, in order to make sure that all devices will have performed 20 transmissions by the end of the simulation. When modifying the Tx Interval, each device will only transmit a maximum of 20 packets.

Finally, each device will perform a random channel hopping selection among those that are free to transmit given the aforementioned duty cycle limitations.

V. Simulation Results

First, network-wide simulations are executed. These assume full control of the network of 1000 devices while applying the same configuration to each node. We assume no other devices would be connected to our network and no external interference would be present. While in a real-world urban deployment there is no control over these factors, we believe it is a safe assumption for more rural use cases. Then, we repeated the analysis focusing on a single device, placed at a distance of 1000 m from the gateway, while the rest of the network operates as described in Section IV. Each configuration is run 10 times and the obtained PDR results are then averaged over the 10 runs, with numerical results reported in Tables I, II and III.

A. Variable Tx Interval

For the variable Tx Interval analysis we set up 5 alternatives to the default network, represented by the 'Fixed' case. Here the Tx Interval is set to 300 s for each node in the network. A good way to decrease the chance of time overlap is to increase the Tx Interval. It is straight-forward that two packets lasting 1 s sent from different devices at a random point during a year have a lower chance to collide than if those two packets were to be sent in a 10 s window.

The 6 methods we used to vary the Tx Intervals are:

- Fixed: The Tx Interval for all transmissions on each device is set to 300 s.
- Fastest: The Tx Interval for all transmissions on each device is set to the fastest possible without going over the duty cycle limit.
- Event Based Short: The Tx Interval of each transmission on each device is set to be between the fastest the node can transmit and twice that value.
- Event Based Long: The Tx Interval of each transmission on each device is set to be between the Fixed interval and twice that value (300 and 600 seconds).
- Fixed Dynamic Low Q: The Tx Interval of each transmission on each device is equal to the Fixed interval plus or minus a tenth of itself.
- Fixed Dynamic High Q: The Tx Interval of each transmission on each device is equal to the Fixed interval plus or minus 1/30, 1/20, 1/10 or 1/5 of itself.

In the network case, the only appreciable gain over the 'Fixed' case, an increase of 2.87% is obtained by slightly increasing the Tx Interval and randomly picking a value between 300 s and 600 s ('Event Based Long'). At the same time, reducing the Tx Interval in both the 'Fastest' and 'Event Based Short' cases lowers the overall PDR. Acting on a single device, no configuration can yield a benefit over just letting the node transmit with a fixed time of 300 s.

B. Variable Start Times

Knowing the exact number of devices in a network, all operating with periodic Tx Intervals, the chance of collision can be reduced to 0 by making sure that

$$Tx\ Interval > ToA_{MaxSF} \times\ number\ of\ devices, \quad (2)$$

and by assigning a start time to each device that is $ToA_{MaxSF}\ \times\ number\ of\ devices$ apart from any other. Even when it is not possible to increase the Tx Interval to such extent (our standard network would require a Tx Interval of 1400 s) it can still be beneficial to distance the transmissions as such.

The 5 methods we used to vary the devices' start time are:

- Standard: each device starts at a random time that is between 0 and Tx Interval. This ensures that they will all perform the same amount of uplink transmissions during the simulation time and it models somebody plugging the devices at any time within a single cycle.
- Slotted Long: each node is assigned a start time that depends on its ID and is at a distance in time greater than $ToA_{MaxSF}\ \times\ number\ of\ devices$ from the next transmission. It is important to note that the Tx Interval of each device is in this case raised according to 2 to be greater than that distance.
- Slotted Short: each node is assigned a start time that depends on its ID and is at a distance in time greater than $ToA_{MaxSF}\ \times\ number\ of\ devices$ from the next transmission. The Tx Interval is set to the standard 300 s.
- SF Based: nodes are set to start transmitting at random times that are the same for each different Spreading Factor.
- Unison: each node starts transmitting at 1 s.

In the network scenario, 'Slotted Long' performed the best, achieving the theoretical 100% PDR, according to eq. 2. More interestingly, by slotting transmissions in the same way while keeping the Tx Interval to 300 s ('Slotted Short'), there is still a marginal improvement of 2.12% in performance over the 'Random' assignment. For a single device scenarios, most of the schemes are equivalent in theory, as the slotting makes sense only if it is a concerted effort made by all devices at once. Applying a specific start time to a single device is equivalent to selecting a random start time, albeit with different time boundaries. 'Slotted Long' still provides the best outcome at 98.5% due to the increased Tx Interval.

C. Channel hopping schemes

Finally, we tested different methods of deciding the next frequency channel to transmit on. This is usually selected randomly. A check is then performed to make sure that transmitting on the channel would not violate the duty cycle limits. If it does not, transmission goes ahead, otherwise another random channel is selected and checked. If there are no channels available, the transmission is cancelled until the next one.

We tested the standard random scheme against various sequences of the 8 available channels. Provided the Tx Interval is long enough, each channel can be chosen liberally without restrictions. For these tests, we kept the standard Tx Interval of 300 s.

- Random: devices use the standard random hopping described above.
- Static Sequence Low Q: Each device uses one of 10 sequences, which are random permutations of the available channels.
- Static Sequence High Q: Each device uses one of 1000 sequences, which are random permutations of the available channels.
- Dynamic Sequence: Each device is assigned a permutation of the available 8 channels. This is then changed into another permutation every time the device reaches the end of the previous one.

TABLE I: Average PDR over 10 simulation runs for different Tx Interval Schemes

	Fixed	Fastest	Event Based Short	Event Based Long	Fixed Dynamic Low Q	Fixed Dynamic High Q
Network-Wide	89.23%	69.61%	86.95%	92.10%	89.03%	89.34%
Single Device	94.5%	92.5%	88.17%	93.94%	87%	87.84%

TABLE II: Average PDR over 10 simulation runs for different Start Time Schemes

	Standard	Slotted Long	Slotted Short	SF Based	Unison
Network-Wide	89.29%	100%	91.41%	89.15%	0%
Single Device	96.5%	98.5%	95%	98%	97%

TABLE III: Average PDR over 10 simulation runs for different Channel Hopping Schemes

	Random	Static Sequence Low Q	Static Sequence High Q	Dynamic Sequence	Sequence by SF
Network-Wide	89.20%	83.55%	88.82%	88.99%	53.49%
Single Device	91%	98.5%	98.5%	94.5%	93.5%

- Sequence by SF: Each device is assigned a random permutation of the channels based on its own spreading factor.

Among the proposed schemes applied network-wide, the channel hopping is the one that yields the least benefits, partly because, without knowledge of other devices, it remains a random process. The fact that sequences make sure that the channels are more equally utilised, without the chance of a channel being picked randomly multiple time in quick succession, does not improve the PDR in any considerable way. When applied to a single device, any sequence proves to be more effective than the random channel hopping algorithm, increasing PDR by 7.5%.

VI. Conclusion and Future Work

In this paper we investigated downlink independent and mostly context-unaware methods to try and increase the PDR of both a network and a single device without having to resort to downlink messages from the network server. Some techniques proposed in this paper grant a relatively small increase in performance, no greater than 10%. However, this is at a very low cost of programming the different behaviour on the network's end devices compared to the increased complexity and energy consumption of traditional, downlink-reliant schemes. For future works this idea will be expanded to feature combinations of these methods applied to groups of nodes making up different classes for performance comparison. A mathematical characterisation of each of these methods is also planned to be included as part of a longer study in future work.

References

[1] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-demand LoRa: Asynchronous TDMA for energy efficient and low latency communication in IoT," *Sensors (Switzerland)*, vol. 18, no. 11, pp. 1–22, 2018.

[2] R. Marini, W. Cerroni, and C. Buratti, "A Novel Collision-Aware Adaptive Data Rate Algorithm for LoRaWAN Networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2670–2680, 2021.

[3] J. Finnegan, R. Farrell, and S. Brown, "Analysis and Enhancement of the LoRaWAN Adaptive Data Rate Scheme," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7171–7180, 2020.

[4] M. Slabicki, G. Premsankar, and M. Di Francesco, "Adaptive configuration of lora networks for dense IoT deployments," *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, pp. 1–9, 2018.

[5] J. Finnegan, R. Farrell, and S. Brown, "Lightweight Timeslot Scheduling through Periodicity Detection for Increased Scalability of LoRaWAN," *Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020*, pp. 8–15, 2020.

[6] R. Trüb and L. Thiele, "Increasing throughput and efficiency of lorawan class a," 2018.

[7] T. Polonelli, D. Brunelli, and L. Benini, "Slotted ALOHA Overlay on LoRaWAN - A Distributed Synchronization Approach," *Proceedings - 16th International Conference on Embedded and Ubiquitous Computing, EUC 2018*, pp. 129–132, 2018.

[8] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, 2018.

[9] B. Citoni, S. Ansari, Q. H. Abbasi, M. A. Imran, and S. Hussain, "Impact of Inter-Gateway Distance on LoRaWAN Performance," *Electronics*, vol. 10, no. 18, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/18/2197

[10] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, 2018.

[11] R. Eletreby, D. Zhang, S. Kumar, and O. Yagan, "Empowering low-power wide area networks in urban settings," *SIGCOMM 2017 - Proceedings of the 2017 Conference of the ACM Special Interest Group on Data Communication*, pp. 309–321, 2017.

[12] 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," *Proceedings - 17th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2018*, pp. 60–71, 2018.

[13] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN," *19th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2018*, pp. 6–14, 2018.

[14] J. Park, K. Park, H. Bae, and C. K. Kim, "EARN: Enhanced ADR with Coding Rate Adaptation in LoRaWAN," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 873–11 883, 2020.

[15] S. Kim and Y. Yoo, "Contention-aware adaptive data rate for throughput optimization in LoRaWAN," *Sensors (Switzerland)*, vol. 18, no. 6, 2018.

[16] A. Hoeller Jr, R. D. Souza, O. L. A. López, H. Alves, M. de Noronha Neto, and G. Brante, "Exploiting Time Diversity of LoRa Networks Through Optimum Message Replication," *Iswcs 2018*, pp. 1–5, 2018.

[17] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," *IEEE International Conference on Communications*, 2017.